UNIVERSITÉ DE LORRAINE

AVERTISSEMENT

Ce document est le fruit d'un long travail approuvé par le jury de soutenance et mis à disposition de l'ensemble de la communauté universitaire élargie.

Il est soumis à la propriété intellectuelle de l'auteur. Ceci implique une obligation de citation et de référencement lors de l'utilisation de ce document.

D'autre part, toute contrefaçon, plagiat, reproduction illicite encourt une poursuite pénale.

Contact : ddoc-theses-contact@univ-lorraine.fr

LIENS

Code de la Propriété Intellectuelle. articles L 122. 4
Code de la Propriété Intellectuelle. articles L 335.2- L 335.10
http://www.cfcopies.com/V2/leg/leg_droi.php
http://www.culture.gouv.fr/culture/infos-pratiques/droits/protection.htm

Nancy-Université
Université Nancy 2

UFR mathématiques et informatique **École doctorale IAEM Lorraine**

Département de formation doctorale en informatique

L'intelligence en essaim sous l'angle des systèmes complexes : étude d'un système multi-agent réactif à base d'itérations logistiques couplées

THÈSE

présentée et soutenue publiquement le 8 décembre 2009

pour l'obtention du

Doctorat de l'université Nancy 2

(spécialité informatique)

par

Rodolphe Charrier

Composition du jury

Rapporteurs :	Jacques Ferber Jean-Louis Deneubourg
Examinateurs :	Mathias Quoy Anne Boyer
Directeur :	François Charpillet
Co-directrice :	Christine Bourjot

Laboratoire Lorrain de Recherche en Informatique et Applications
Equipe MAIA
Campus Scientifique, B.P.239 - 54506 Vandœuvre-lès-Nancy Cedex

Remerciements

Cette thèse fut l'occasion d'un travail de recherche passionnant qui trouve son aboutissement dans le présent document. J'en éprouve une réelle satisfaction intellectuelle. Rien de tout cela n'aurait vu le jour sans un contexte favorable et des personnes bienveillantes que je souhaite remercier ici.

Mes remerciements vont tout d'abord à François Charpillet qui a accepté de diriger cette thèse et m'a accueilli dans l'équipe Maia : ce fut un réel pari autant de son côté que du mien, tant le sujet, avec un peu de recul, était risqué sur le plan scientifique. L'accueil chaleureux de François, son ouverture d'esprit et son écoute, sa perspicacité et sa curiosité scientifiques ont largement contribué à la bonne réalisation de ce travail.

Remerciements également à Christine Bourjot, qui fut ma co-directrice de thèse pendant ces quelques années et qui m'a ouvert la voie des systèmes multi-agents bio-inspirés. Ses relectures attentives et ses conseils avisés, ainsi que sa bonne humeur et son optimisme ont été précieux.

Je remercie les rapporteurs de cette thèse, Jacques Ferber et Jean-Louis Deneubourg, qui m'ont fait l'honneur d'accepter de juger de mon travail, tandis que leurs propres travaux furent pour moi des sources d'inspiration et des références constantes. Je remercie également mes examinateurs Mathias Quoy et Anne Boyer. Anne Boyer n'a jamais cessé d'encourager ma démarche et de croire en mon travail, je l'en remercie doublement.

Enfin je veux remercier toute l'équipe Maia que j'ai pu côtoyer au cours de ces années, membres permanents ou doctorants. Ils m'ont aidé directement ou indirectement par leurs interactions scientifiques ou humaines à tenir la distance, et notamment ceux qui furent présents dans la dernière ligne droite, lors des phases de relecture et de préparation à la soutenance, moment crucial s'il en est. Qu'ils reçoivent ici toute ma gratitude.

A Gaby.

Table des matières

Table des figures **xi**

I Introduction **1**

1 Préambule 3

2 Problématique générale et positionnement scientifique 4

3 Modèle et méthodologie 7

4 Evaluation et pertinence du modèle 11

5 Contributions de la thèse 11

6 Plan de la thèse 12

II Intelligence en essaim, auto-organisation et systèmes complexes **15**

1 L'intelligence en essaim **19**

1.1 Naissance et définitions du concept d'intelligence en essaim. 19

1.1.1 Apparition du concept. 19

1.1.2 Autres visions de l'intelligence en essaim. 21

1.2 Inspiration biologique et caractéristiques 22

1.2.1 La métaphore biologique 22

1.2.2 Caractéristiques génériques des approches en essaim 25

1.3 De la bio-inspiration aux algorithmes emblématiques : "flocking" et colonies de fourmis 26

1.3.1 Boids de Reynolds 26

1.3.2 Le modèle de Vicsek et les systèmes particulaires 29

1.3.3 Le double pont de Deneubourg et l'algorithme fourmi de Dorigo . . . 30

1.3.4 Vers l'optimisation 34

2 Champs d'application de l'intelligence en essaim 37

2.1 Résolution de problèmes d'optimisation 37

2.1.1 Métaheuristiques et optimisation 37

2.1.2 Optimisation par essaim particulaire 39

2.1.3 Optimisation par colonies de fourmis 41

2.2 Robotique en essaim 44

3 Quelques constats sur l'intelligence en essaim 47

4 Auto-organisation : concept fondateur de l'intelligence en essaim. 51

4.1 Systémique et cybernétique 52

4.2 Synergétique 55

4.3 Mécanismes de l'auto-organisation biologique 57

4.4 Auto-organisation en intelligence artificielle 61

4.5 Bilan intermédiaire sur l'auto-organisation 63

5 Systèmes complexes pour l'intelligence en essaim. 65

5.1 Mesure de l'auto-organisation et notion de complexité 66

5.2 La science des systèmes complexes : une science fédératrice autour de l'auto-organisation 68

5.2.1 Définitions du domaine 68

5.2.2 Notre aproche : application logistique et RIC 72

5.3 L'application logistique : un modèle minimal non linéaire et chaotique 73

5.3.1 Analyse des séries temporelles de l'application logistique 73

5.3.2 Diagramme de bifurcation 79

5.3.3 Exposants de Lyapunov et chaos 79

5.3.4 Autres propriétés. 81

5.4 Réseaux d'itérations couplées : modèle archétypal de système complexe computationnel 82

5.4.1 Réseaux localement et globalement couplés 82

5.4.2 Phénomènes de synchronisation 85

5.4.3 Le modèle de gaz d'itérations couplées 88
5.5 Entropie de Kolmogorov : mesure du degré de chaos d'un système 90
5.5.1 Définition de l'entropie K. 90
5.5.2 Propriétés de l'entropie K. 90
5.5.3 Calcul de l'entropie K. 91

6 Synthèse 93

III Modèle de SMA réactif pour l'intelligence en essaim : le SMA logistique 95

7 Les approches orientées agent 99
7.1 Diverses conceptions de l'agent 99
7.1.1 De la particule à l'agent : l'agent du physicien 99
7.1.2 L'agent de l'informaticien 101
7.2 Système multi-agent réactif 102

8 Formalisation du SMA logistique 107
8.1 Principes de conception 107
8.2 Spécifications et formalisation du SMAL 109
8.2.1 Définition du système multi-agent logistique 109
8.2.2 Agents logistiques 111
8.2.3 Modèle de l'environnement 113
8.2.4 Caractéristiques des champs de l'environnement 114
8.2.5 Équations maîtresses du SMAL 118
8.2.6 Opérateurs de perception limitée. 120
8.2.7 Processus sensorimoteurs 122
8.2.8 Réactions de l'environnement 125
8.2.9 Discussion 126
8.3 Analyse du SMA logistique 127
8.3.1 Intérêt algorithmique : génération des comportements d'exploration et d'exploitation 127

8.3.2 Interprétation biologique de la modélisation chaotique 129

9 Synthèse **131**

IV Mise en oeuvre du SMA logistique pour l'intelligence en essaim **133**

10 Phénomènes d'auto-organisation **137**

10.1 Synchronisation du SMAL chaotique 137

10.2 Mesure de la stabilité 146

10.2.1 Des agents jumeaux pour étudier la stabilité des clusters. 146

10.2.2 Application au suivi des clusters de synchronisation partielle. 149

10.3 Auto-organisation de type flocking 151

10.3.1 Cas d'étude et instanciation. 152

10.3.2 Résultats de simulations et analyse. 152

10.3.3 Discussion. 156

10.4 Bilan 157

11 Mécanismes adaptatifs **159**

11.1 Flocking dans un espace avec obstacles 159

11.2 Application au flocking de robots 171

11.3 Apparition de files d'agents par adaptation 172

11.4 Application à un système proie-prédateurs vu comme problème d'optimisation 177

11.5 Synthèse 187

12 Mécanismes adaptatifs des fourmis logistiques **189**

12.1 Modélisation des fourmis logistiques 190

12.2 L'expérience du double pont revisitée 194

12.3 Application des fourmis logistiques au problème du TSP 200

12.3.1 Algorithme des fourmis logistiques pour le TSP 200

12.3.2 Simulations et résultats 205

12.3.3 Bilan 207

13 Discussion **211**

V Conclusion **213**

VI Annexes **219**

A Les systèmes dynamiques **221**

A.1 Définitions générales sur les systèmes dynamiques 221

A.2 Les systèmes dynamiques non linéaires discrets 226

A.3 Quelques éléments de théorie des bifurcations 228

A.3.1 Bifurcation noeud-col 229

A.3.2 Bifurcation trans-critique 230

A.3.3 Bifurcation par doublement de période 230

B Quelques propriétés complémentaires sur l'application logistique **231**

B.0.4 Propriétés statistiques de l'application logistique pour $r = 4$ 231

B.0.5 Universalité de l'application logistique 233

C Réseaux de neurones chaotiques **237**

C.1 Les prémisses biologiques 237

C.2 La modélisation et l'analyse numérique 239

Bibliographie **243**

Table des figures

1 Représentation schématique des divers niveaux de conception : intelligence en essaim, système mulit-agent logistique (SMAL), gaz d'itérations couplées (GIC) . 8

1.1 Règles de déplacement et d'évitement d'obstacles chez Reynolds 27

1.2 Expérience du double pont pour l'étude du fourragement chez les fourmis. Données tirées de [Deneubourg *et al.*, 1990]. 31

2.1 Principe et image de simulation 3D de l'algorithme par essaims particulaires. . . 40

4.1 Carte globale sur le positionnement et l'évolution des sciences de la complexité. Carte réalisée par Brian Castellani dans l'ouvrage [Castellani et Hafferty, 2009]. Image sous licence Creative Commons Attribution-Share Alike 3.0, téléchargée sur le Web. 53

4.2 Illustration du principe d'asservissement. 57

4.3 Illustration du principe de causalité circulaire. 57

5.1 Pôles méthodologiques des sciences des systèmes complexes selon [Shalizi, 2006] . 71

5.2 Tracé de $f(x)$ pour différentes valeurs du paramètre r, également appelée "application de premier retour". La tangente de la courbe pour $r = 3$ à l'intersection avec la première diagonale a une pente -1 : c'est le premier point de bifurcation de doublement de période. 74

5.3 Applications de premier retour (à gauche) et séries temporelles associées (à droite) pour quelques valeurs du paramètre de contrôle de l'application logistique 75

5.4 Tracé de $f(x)$ pour différentes valeurs du paramètre r, également appelée "application de premier retour". La tangente de la courbe pour $r = 3$ à l'intersection avec la première diagonale a une pente -1 : c'est le premier point de bifurcation de doublement de période. 76

5.5 Application logistique pour $r = 4$. 78

5.6 Diagramme de bifurcation de l'application logistique à différentes itérations . . . 80

5.7 Estimation de l'exposant de Lyapunov pour l'application logistique en fonction du paramètre de contrôle. 81

5.8 Schémas de principe d'un RIC localement couplé à une dimension. 84

5.9 Diagramme de bifurcation de l'application itérée $x_{n+1} = 1 - a\ x_n^2$. Diagramme obtenu après 500 itérations de l'application. 85

5.10 Diagramme de synchronisation de l'application itérée $x_{n+1} = 1 - a\ x_n^2$. D'après [Manrubia *et al.*, 2004]. 87

7.1 Schéma de la boucle globale de rétro-action agent-environnement "perception-décision-influence-réaction". 105

8.1 Représentation schématique des divers niveaux de conception : intelligence en essaim, système mulit-agent logistique (SMAL), gaz d'itérations couplées (GIC) . 107

8.2 Illustration de la notion de champ dans un espace 3D tirée de "3D HeatBugs" par Gabriel Catalin Balan sur la plate-forme de simulation MASON [Luke *et al.*, 2003].115

8.3 Application du principe exploration-exploitation aux zones du diagramme de bifurcation de l'application logistique $x_{n+1} = r\ x_n(1 - x_n)$ 128

10.1 Diagramme de bifurcation de l'application logistique $x_{t+1} = f(x_t, a) = 4\ a\ x_t\ (1 - x_t)$ après 500 itérations sur 500×500 échantillons des couples $(a, x^0) \in ([0, 1] \times [0, 1])$ 138

10.2 Schéma récapitulatif du changement de position de l'agent chaotic. 139

10.3 Images de simulations à différents pas de temps où $a_0 = 1$ et $\epsilon^0 = 0.9$ pour chacun des $N = 100$ agents. De plus le rayon de voisinage de perception est $R = 4$. . . . 141

10.4 Simulation avec $\epsilon^0 = 0.95$, et un rayon de perception $R = 4$ pour tous les agents. La transition de synchronisation se produit au pas de temps $t = 2700$. 143

10.5 Cas des agents chaotiques : nombre moyen de clusters formés en z, en fonction du couplage ϵ^0 en y et du rayon de voisinage de perception R en x pour $N = 100$ agents. 144

10.6 Déviation standard sur le nombre moyen de cluster $\bar{N}_c$ en z en finction du couplage en y et du rayon de voisinage en x pour $N = 100$ agents. 144

10.7 Méthode de calcul approché du plus grand coefficient de Lyapunov qui évite les dépassements des limites de calcul. 147

10.8 Clusters de synchronisation après environ 18000 pas de temps. La couleur rouge des agents est associée aux valeurs positives de leur coefficient de Lyapunov, la couleur verte indique les valeurs négatives. Les arcs entre agents relient les agents d'un même cluster. 150

10.9 Spectre de Lyapunov partiel et suivi de l'entropie K du cluster C_0 en fonction du temps avec $\epsilon = 0,8$. L'intervalle de temps est de 50 pas de temps à chaque évaluation. 151

10.10 Copies d'écran d'une simulation avec $N = 100$ agents de rayon de voisinage $R = 4$, et de couplage $\epsilon^0 = 0.8$ 153

10.11 Copies d'écran d'une simulation avec $N = 100$ agents de rayon de voisinage $R = 4$, et de couplage $\epsilon^0 = 1$ 153

10.12 Evolutions temporelles de l'ensemble des variables décisionnelles internes x correspondant à tous les agents en fonction du temps. Le rayon de perception est $R = 4$. 154

10.13 Cas d'agents de tout profil dynamique : nombre moyen de clusters formés en z, en fonction du couplage ϵ^0 en y et du rayon de voisinage de perception R en x pour $N = 100$ agents. 155

10.14 Spectre de Lyapunov partiel et suivi de l'entropie K du système en fonction du temps avec deux valeurs ϵ. L'intervalle de temps est de 50 pas de temps à chaque évaluation. 156

11.1 Champ de l'environnement et repérage géométrique. 161

11.2 Schéma de principe de l'évitement d'obstacles pour l'agent logistique et du calcul de la direction globale résultante. 161

11.3 Simulations dans le cas où tous les agents ont même variable interne de contrôle a. La première ligne concerne le cas $a = 0$ des agents de Vicsek et le deuxième le cas $a = 1$ des agents chaotiques. Une analyse comparative est faite sur l'angle d'ouverture du cône de perception des obstacles. 168

11.4 Simulations dans le cas où les agents ont une variable interne de contrôle a tirée aléatoirement sur $[0, 1]$. La première ligne montre l'effet du rayon de perception de synchronisation interne aux agents. La deuxième ligne montre un cas d'engorgement du groupe d'agents dans un angle de l'environnement et sa résolution. Enfin un cas de simulation avec une distribution normale des variables internes de contrôle. 169

11.5 Photos successives d'une séquence d'un flocking de 5 robots Khepera. 171

11.6 Simulation obtenue au bout de 2160 pas de temps avec : environnement 100×100, $N = 50$ agents, $\epsilon^0 = 0,20$, voisinage de perception de rayon $R = 20$ 174

11.7 Point $(\frac{1}{2}, \frac{1}{2})$ superstable sur le diagramme de bifurcation de l'application logistique $x_{t+1} = f(x_t, a) = 4\ a\ x_t\ (1 - x_t)$ 177

11.8 Diagramme de bifurcation de l'application quadratique $x^{t+1} = a\ (2\ x^t - 1)^2$, calculée avec 500 iterations sur 500 échantillons dans l'intervalle $[0, 1]$ 179

11.9 Simulation du problème proie-prédateurs avec $N = 50$ predateurs. La proie est repérésentée par un petit disque et son voisinage de "présence" est marqué dans l'environnement. Les parameters sont $\epsilon^0 = 0.2$, le masque de marquage de la proie est de taille $(11, 11)$, le rayon de perception des prédateurs est $R = 1.0$ 182

11.10 Image d'une simulation avec capture de trois proies 183

11.11 Évolution de la dynamique interne d'un prédateur capturant la proie — 10000 pas de temps sont portés sur le graphique. 184

11.12 Résultats des temps de capture pour deux séries de simulations pour deux groupes d'agents. Le temps de capture est mesuré par couleur et résulte d'une moyenne des résultats sur 5 simualtions de même paramètres. 185

11.13 Fonction d'Easom : $f_{Easom}(x_1, x_2) = -\cos(x_1)\ \cos(x_2)\ e^{-((x_1-\pi)^2+(x_2-\pi)^2)}$ 187

12.1 Diagramme de bifurcation de l'application $x^{t+1} = 1 - a(2\ x^t - 1)^2$ 192

12.2 Expérience du double pont et sa modélisation sous forme de graphe. 194

12.3 Résultats de simulation du double pont comportant 1500 fourmis logistiques. Le résultat affiché est une moyenne sur 10 simualtions. Q désigne le rapport du nombre de fourmis ayant emprunté une branche sur le total N de fourmis ayant traversé au temps t. 196

12.4 Rapport de nombres d'agents passés par un arc donné du graphe sur le nombre total qui a traversé ; avec $N = 1500$ fourmis logistiques, 6 branches, $\epsilon = 0$ et $m = 499$ 199

12.5 Tracé de quelques sigmoides d'équation $y = \dfrac{1}{1 + e^{b(x-x_0)}}$ avec $x_0 = 5$. 203

12.6 Problème TSP att48 (48 villes) de cycle hamiltonien optimal 10628. Meilleures performances de l'algorithme AS et des fourmis logistiques en fonction du nombre de fourmis utilisées. 208

12.7 Performances comparées sur 5000 pas de temps et en fonction du coefficient d'évaporation pour les deux algorithmes Ant System (AS) et fourmis logistic (LA). 209

A.1 Schémas illustant les types de stabilité d'un point fixe. 224

A.2 Les trois principaux types de bifurcation pour des applications en dimension 1. . 230

B.1 Densité de probabilité pour l'application logistique en $r = 4$. 231

B.2 Diagramme de bifurcation de l'application itérée $x_{n+1} = r \sin(\pi\ x_n)$. 233

C.1 Etude d'un réseau de neurones biologiques de la rétine de salamandre d'après [Crevier et Meister, 1998]. En haut les données biologiques et en bas la réponse du système modélisé par circuit électrique. 238

C.2 Réseau à deux neurones de Xin Wang. 240

C.3 Diagrammes de bifurcation des deux états (x, y) des neurones en fonction du paramètre de contrôle μ. 241

Première partie

Introduction

1 Préambule

Cette introduction brosse un portrait global de cette thèse intitulée **L'intelligence en essaim sous l'angle des systèmes complexes : étude d'un système multi-agent réactif à base d'itérations logistiques couplées**. Avant même de présenter de façon plus explicite une vision synthétique de ce travail, commentons dans un premier temps ce titre et évoquons pour ce faire un article important qui peut être considéré comme l'article inspirateur de cette thèse. Cet article publié en 2003 par les physiciens Shibata et Kaneko [Shibata et Kaneko, 2003] pose les fondements de la question scientifique que nous avons abordée. Cet article vient clore d'une certaine façon l'étude menée depuis une vingtaine d'années par Kaneko sur les réseaux d'itérations couplées (RIC[1]). Nous employons le terme "clore" parce qu'à notre connaissance aucun article des auteurs n'a suivi sur cette thématique depuis cette date[2]. Selon nous pourtant, cet article vient en même temps ouvrir de nouvelles perspectives et une voie de recherche à part entière, voie que nous avons explorée. Résumons en quelques lignes l'objectif de l'article.

Un réseau d'itérations couplées (RIC) est un modèle de calcul à temps discret et à espace d'état continu typique des systèmes dits complexes[3]. La version classique de ce réseau abondamment étudiée par Kaneko [Kaneko, 1993], consiste à coupler une grille de cellules dont les états internes sont donnés par une application non linéaire itérée, dont la plus élémentaire — l'application logistique[4]— calcule un polynôme de degré 2. On assimile souvent ce modèle à un automate cellulaire à états continus, l'évolution des états des cellules étant déterminée à chaque pas de temps par l'application non linéaire utilisée. L'article expose une évolution radicale du modèle RIC en proposant de casser la topologie de couplage fixe de type grille pour rendre les cellules capables de mobilité dans l'espace géométrique. Ainsi les couplages entre entités cellulaires peuvent évoluer dans le temps. Ce modèle est appelé "gaz d'itérations couplées"[5] ; il est destiné selon les auteurs à mieux comprendre le fonctionnement de certains systèmes biologiques. En particulier, trois types de phénoménologies sont ciblés implicitement par les auteurs : le comportement collectif des fourmis, celui des oiseaux et celui des bactéries. L'intelligence en essaim est explicitement évoquée, sans figurer dans les objectifs visés par le modèle précis développé dans l'article. Tout en assumant que le "gaz d'itérations couplées" ne prétend pas à ce stade modéliser ces phénomènes (car au final, l'article n'aborde pas vraiment les phénoménologies biologiques), les auteurs sont convaincus que sa simplicité de conception et la batterie d'outils mathématiques qui l'accompagnent en font un modèle abstrait permettant de mieux cerner l'émergence des propriétés macroscopiques dans ces systèmes biologiques.

Nous en sommes également convaincus et avons repris ce credo dans cette thèse. Nous nous sommes ainsi fixés le programme de recherche suivant : sur la base des principes de conception du gaz d'itérations couplées précédent, nous prolongeons l'intention de leurs auteurs à l'intelligence en essaim, en proposant le système multi-agent logistique (SMAL), entièrement nouveau dans le domaine des SMA réactifs, qui met en jeu des mécanismes génériques d'auto-organisation

[1] "Coupled Map Lattice" (CML) dans la littérature scientifique

[2] ... de l'aveu même de Kaneko, rencontré en conférence en 2007.

[3] Le chapitre 5.2 est consacré à expliciter ce qu'est ce domaine de recherche parfois difficile à cerner.

[4] Les propriétés principales de cette application sont décrites de façon détaillée dans la section 5.3, ainsi que les réseaux d'itérations couplées dans la section 5.4.

[5] "Coupled Map Gaz" (GCM).

et d'adaptation dans un cadre de formalisation mathématique simplificateur mais unifié. Cette thèse tente de vérifier que le SMAL est bien un modèle adapté à l'intelligence en essaim, en focalisant l'étude sur deux classes de phénomènes d'auto-organisation qui ont inspiré le domaine : il s'agit d'une part de la formation de groupes d'entités mobiles dans un environnement, que nous emploierons dans ce document sous son expression anglaise de "flocking", et d'autre part des comportements stigmergiques à base de phéromones de type fourmis. Nous montrerons qu'elles correspondent aux schémas typiques générés par le SMAL.

Nous retrouvons ainsi les différents mots-clés du titre de cette thèse : le cadre d'étude est l'intelligence en essaim (dans son sens informatique), que l'on aborde selon une démarche issue des systèmes complexes, au moyen d'un modèle de SMA réactif intégrant le modèle d'applications non linéaires couplées (tiré du modèle RIC de Kaneko). Il reste à préciser ce que l'on entend plus précisément sous ces divers termes. Les quelques sections de cette partie introductive ont pour but de présenter la démarche scientifique qui sous-tend ce travail, la méthodologie employée pour la mener à bien, ainsi que les principales contributions que cette thèse apporte. Il s'agit d'une part de préciser l'articulation des idées scientifiques qui ont conduit à développer ces travaux de recherche et permettre au lecteur de situer le positionnement de notre problématique scientifique. Cela est rendu d'autant plus utile que cette problématique est à la croisée de plusieurs domaines de recherche, comme le titre l'indique. Cet exercice d'équilibriste peut rendre difficile la projection d'une vision globale et cohérente de notre projet scientifique. D'autre part, nous indiquons comment s'est constitué le modèle que nous développons et quelle méthodologie d'évaluation a été engagée pour en vérifier la pertinence. Enfin, nous annonçons dès cette partie quelles contributions ce travail prétend apporter.

Cette première partie introductive a donc pour objectif de donner un premier niveau de lecture sur cette thèse afin d'en faire ressortir la ligne directrice ainsi que les points et caractéristiques essentiels.

2 Problématique générale et positionnement scientifique

Domaine de l'intelligence en essaim.

Problématique générale. En partant d'une problématique très large, nous nous intéressons dans cette thèse à la coordination d'un grand nombre d'agents informatiques conduisant à certains types de schémas organisationnels ou à la réalisation collective d'une tâche donnée. Bien que cette problématique ait capté l'attention de la communauté de l'Intelligence Artificielle (IA), le recours aux techniques "classiques" de l'IA (apprentissage, planification, théorie de la décision, approches symboliques,...) et plus généralement aux approches de type descendantes ("top-down") tendant à optimiser globalement une fonction caractéristique du problème considéré, s'avèrent difficiles à mettre en oeuvre dès que le nombre d'agents augmente ("passage à l'échelle") : explosion de la complexité algorithmique impliquant l'intractabilité des problèmes, perte des conditions d'application des théorèmes fondamentaux d'optimalité,... A partir de ce constat, des approches décentralisées voient progressivement le jour, approches fondamentalement ascendantes ("bottom-up") qui tentent de pallier l'obstacle du nombre et de la distribution des agents au prix d'une perte d'optimalité des solutions.

Certaines de ces approches se fondent sur l'observation des espèces du monde vivant qui réalisent collectivement des tâches complexes en mobilisant relativement peu de capacités cognitives individuelles, ce qui constitue un paradoxe des conceptions sur l'intelligence et un nouveau champ d'investigation scientifique. Ces approches constituent le socle de ce qui est appelé aujourd'hui l'"intelligence en essaim" [6], domaine scientifique récent et pluridisciplinaire qui s'est développé activement à partir des années 90 à l'initiative du domaine de la robotique cellulaire. C'est dans cette démarche de modélisation de l'intelligence en essaim que se situent nos travaux pour aborder la problématique globale énoncée précédemment.

Problématiques de l'intelligence en essaim. Pour comprendre ce qu'est l'intelligence en essaim de façon intuitive, le plus simple est de faire appel à la métaphore biologique : l'intelligence en essaim s'apparente à une forme collective d'intelligence que l'on observe chez un grand nombre d'espèces animales (insectes, poissons, oiseaux, mammifères,...), exhibant pour la plupart des comportements sociaux très développés et des capacités d'adaptation remarquables à leur environnement. Le fait surprenant et fascinant pour les scientifiques dans cette forme d'intelligence provient de l'émergence de phénomènes d'organisation et d'adaptation observés à une échelle macroscopique que l'on ne peut soupçonner en analysant individuellement les capacités cognitives des entités, parfois extrêmement limitées. Le cas des fourmis est devenu l'archétype de cette forme d'intelligence extrêmement sophistiquée au regard de la structure neuronale d'une fourmi individuelle. La question fondamentale ici réside dans le passage d'un système principalement réactif (la fourmi agit essentiellement selon des mécanismes de type stimulus-réponse) à un super-organisme que l'on pourrait qualifier d'intelligent (la colonie), et cela sans aucun contrôle centralisé.

La question est d'autant plus intéressante qu'elle a des avantages réels pour l'informatique et l'ingénierie en général : les technologies du web, l'optimisation, la robotique entre autres, sont des domaines d'application où l'intelligence en essaim apporte des idées nouvelles de résolution des problèmes. L'informatique s'est en effet inspiré des études faites sur l'intelligence en essaim biologique pour produire des algorithmes novateurs. L'exemple de la conception des algorithmes fourmis est révélatrice de cette bio-inspiration. En retour les algorithmes conçus en informatique sont aussi des outils d'analyse et de simulation pour les biologistes.

L'intérêt grandissant pour l'intelligence en essaim est sans doute liée aux caractéristiques qu'on lui prête : coût computationnel réduit, robustesse aux pannes, fonctionnement autonome et décentralisé (pas de contrôleur global), "émergence" des solutions globales, robustesse au passage à l'échelle. De fait l'intelligence en essaim informatique est rattachée à un courant de développement informatique baptisé "Soft Computing"[7]. Ce courant développe des approches algorithmiques qui se démarquent des approches dites "Hard Computing" dédiées à la recherche de solutions exactes ou optimales aux problèmes posés, par le fait qu'elles abordent plutôt des problèmes intractables, de type NP-complets par exemple, ou qui échappent à l'analyse mathématique. De plus ces approches ne garantissent pas l'exactitude ou l'optimalité des solutions

[6] traduction de "Swarm Intelligence"

[7] Ce courant de développement informatique recouvre également la logique floue, les réseaux de neurones artificiels, les algorithmes évolutionnaires, les approches probabilistes bayésiennes, la théorie du chaos,... [App,].

fournies mais privilégient plutôt la rapidité à trouver des solutions acceptables à coût computationnel réduit.

Les travaux de cette thèse sont tournés vers deux types de problématiques spécifiques à l'intelligence en essaim :

- Les phénomènes de déplacement de groupe de type "flocking" rencontrées dans de nombreuses espèces animales, y compris chez les êtres humains (étude des déplacements de foules humaines par exemple). Le but est avant tout la simulation, et notre modèle SMAL conduit assez naturellement vers ce type de schémas dynamiques.
- Les phénomènes stigmergiques à base de phéromones de type fourmis, dont les applications sont désormais nombreuses en optimisation combinatoire et dans le domaine des réseaux.

Nous restreignons volontairement notre étude à ces deux aspects car le modèle SMAL que nous développons permet précisément d'en apporter une traduction originale dans le champ des systèmes complexes. De plus ces deux phénomènes ont inspiré des métaheuristiques[8] pour l'optimisation comme nous l'exposons au chapitre 2.

L'intelligence en essaim vue comme un système complexe.

L'intelligence en essaim aujourd'hui est en fait abordée sous plusieurs angles scientifiques : biologique tout d'abord puisque ce domaine reste fortement bio-inspiré, informatique ensuite où il constitue une branche de l'informatique artificielle distribuée, et physique enfin par les modèles particulaires dédiés à ce domaine. Il est admis néanmoins par la communauté scientifique que toutes les caractéristiques associées à l'intelligence en essaim relèvent d'un processus plus général : l'auto-organisation. Ce processus se révèle aussi bien en physique, biologie, chimie, économie, sociologie, ... Le chapitre 4 lui est consacré et en rappelle les principales problématiques, mécanismes ou tentatives de théorisation scientifique. Cette courte revue du concept fait apparaître un manque cruel de caractérisation quantitative qui pourrait être appliquée pour détecter le degré d'auto-organisation d'un système.

Le concept de complexité tente de combler cette lacune : il vise à fournir une grandeur quantifiable de l'auto-organisation des systèmes. Une science s'est développée autour de cette problématique portant le nom de science des systèmes complexes qui a pour objectif de théoriser les phénomènes d'auto-organisation. Un système complexe est défini de façon variable selon les auteurs, mais on retrouve systématiquement une base commune : un système complexe est constitué d'un grand nombre d'entités en interaction mutuelle, et éventuellement en interaction avec un environnement, dont le comportement global émerge de façon non linéaire du niveau local. Le credo fondateur de ce champ scientifique, qui tend à devenir une science à part entière[9], vient du fait que ces problématiques de diverses natures en apparence peuvent avoir des outils de modélisation communs et des lois/mécanismes sous-jacents similaires —voire même universels—, conférant ainsi une forme d'unité à tous ces phénomènes complexes. Cependant aucune théorie "unificatrice" des systèmes complexes n'est encore en mesure de corroborer ce credo, et pour

[8] Principes algorithmiques généraux d'approximation qui peuvent s'appliquer à divers types de problèmes. cf. section 2.1.1.

[9] "Vers une science des systèmes complexes" était le sous-titre de la 4ème conférence européenne des systèmes complexes ECCS 2007

le moment, la simulation informatique, notamment à base d'agents, y reste le terrain principal d'expérimentation pour tester les modèles. Néanmoins, de nombreux modèles sont développés, puisant leur source théorique dans les mathématiques ou la physique contemporaines (théorie du chaos, modèles stochastiques markoviens,...), apportant leurs batteries d'outils d'analyse.

Cette thèse place la vision de l'intelligence en essaim, en tant que modèle mettant en jeu de multiples entités en interaction mutuelle et environnementale, dans une perspective de modélisation complexe, avec des outils de modélisation déjà éprouvés largement dans le champ des systèmes complexes. Cette démarche vise trois objectifs conjoints :

- bénéficier des résultats théoriques et grandeurs quantitatives associées aux techniques des systèmes complexes,
- dépasser les limitations des modèles actuels de l'intelligence en essaim informatique : manque de possibilité de contrôle de la dynamique, manque d'outils de mesure, manque d'explication théorique des mécanismes impliqués,
- produire un modèle plus générique permettant de rendre compte (par simulation) de plusieurs types de phénoménologies caractéristiques de l'intelligence en essaim, auparavant décrite par des modèles distincts. Nous visons notamment à réunir les phénomènes stigmergiques de type fourmis et les phénomènes de type flocking dans un même cadre de modélisation.

Cette filiation avec les systèmes complexes se traduit dans notre approche par l'utilisation d'un des modèles de calcul typiques des systèmes complexes, déjà évoqué dans le préambule : les réseaux d'itération couplées dans leur version "gaz".

3 Modèle et méthodologie

Cette section résume la démarche globale et les principes de conception qui ont mené à la proposition du système multi-agent logistique développé dans cette thèse.

Présentation schématique de l'approche de modélisation.

La figure 8.1 décrit schématiquement l'articulation de notre démarche de modélisation en trois niveaux :

- un niveau d'inspiration phénoménologique représenté par l'intelligence en essaim et ses concepts clés : la notion d'essaim, d'environnement et de boucle de rétroaction, d'auto-organisation,
- un niveau systémique correspondant au système multi-agent qui décrit les différentes parties du système, à savoir les agents, l'environnement, et les relations entre ces parties selon le schéma de principe influence-réaction,
- un niveau computationnel pour finir, occupé par notre modèle de calcul dérivé des gaz d'itérations couplées (GIC).

Ces trois niveaux entrent dans le cadre des systèmes complexes. Cette séparation des niveaux permet de rendre compte des phénomènes dynamiques dus au modèle de calcul d'une part, et de raisonner sur une conceptualisation systémique à base d'agents mieux adaptée au traitement des problèmes d'intelligence en essaim d'autre part. Le niveau agents est intermédiaire entre le

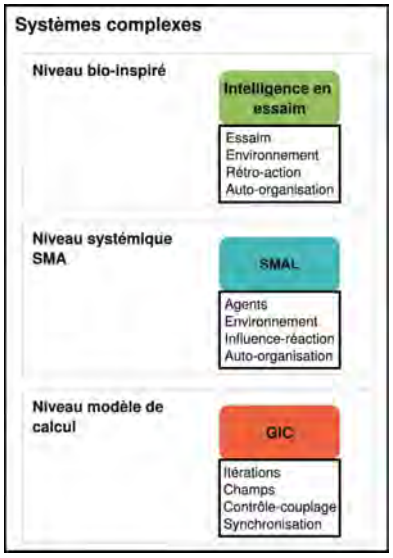


FIG. 1 – Représentation schématique des divers niveaux de conception : intelligence en essaim, système mulit-agent logistique (SMAL), gaz d'itérations couplées (GIC)

modèle de calcul et la phénoménologie ciblée. Nous en détaillons la structure particulière dans cette thèse.

Caractéristiques du modèle de calcul.

Nous considérons ici l'intelligence en essaim dans une approche dynamique relative au modèle de calcul mis en oeuvre. Notons que des approches semblables sont menées en sciences cognitives et et réseaux de neurones artificiels (consulter par exemple l'ouvrage collectif [Guillot et Daucé, 2002] sur ces approches). Précisons en quoi consiste notre approche dynamique : le nouveau modèle de SMA réactif que nous avons dénommé système multi-agent logistique (SMAL), a pour ambition de montrer explicitement dans sa modélisation comment à partir d'une formulation générique des mécanismes au niveau local peuvent apparaître des phénomènes d'auto-organisation et d'adaptation au niveau global, dans un système mettant en interaction des agents et un environnement de façon non linéaire et déterministe.

Non linéarité. La notion de non linéarité est souvent mentionnée dans cette thèse à propos de l'auto-organisation et des systèmes complexes, car elle se révèle être un élément fondamental pour expliquer l'origine complexe de ces systèmes. Cette notion de non linéarité est de fait au coeur de la conception de notre modèle. Cependant, elle n'est définie rigoureusement que dans le cadre de la théorie mathématique des systèmes dynamiques. Un rappel des principales notions utilisées dans cette thèse, associées à la théorie des systèmes dynamiques dans le cas non linéaire à temps discret ou continu, est proposé en annexe A. Notre modèle relève ainsi fondamentalement d'une approche dynamique des systèmes, afin de préciser, formaliser et localiser mathématiquement les non linéarités et d'en contrôler autant que possible les effets et la structure.

Déterminisme. D'autre part, la conception fondamentalement déterministe du modèle proposé contraste avec les approches stochastiques habituellement développées dans la modélisation de l'intelligence en essaim en informatique actuellement (nous abordons deux grandes familles du domaine au chapitre 2). Elle relève d'une démarche volontairement constructive et explicative des phénomènes étudiés ; nous reprenons en cela un des principes fondateurs de Kaneko pour aborder l'étude des systèmes complexes dans l'ouvrage [Kaneko et Tsuda, 2001a]. Il est important de rappeler que l'aspect déterministe du système n'est pas contradictoire avec son caractère imprédictible, à cause précisément de la notion de chaos déterministe. L'article [Bricmont, 1995] fait le point sur les idées associées à cette question théorique. Dans notre cadre de modélisation, le chaos déterministe sert dans un certain sens à générer de l'aléatoire de façon contrôlée dans le système.

Le système multi-agent logistique.

Notre modèle s'inscrit dans le champ des systèmes complexes : il propose et étudie un système multi-agent (SMA) inspiré d'un modèle emblématique des systèmes complexes, les réseaux d'itérations couplées (RIC) de Kaneko [Kaneko, 1993]. Comme mentionné en préambule, ces réseaux sont constitués de cellules comportant chacune une application non linéaire à temps discret que l'on itère en la couplant avec les cellules voisines.

L'étude numérique et théorique de ce types de réseau a été en grande partie menée par le physicien Kaneko depuis les années 80 lorsque l'application considérée est de type logistique. De fait toute la non linéarité du SMAL est concentrée dans l'application logistique : cette application est de la forme récurrente $x^{t+1} = 4ax^t(1 - x^t)$ où t est le temps discret, et a le paramètre de contrôle de l'application. Elle possède un grand nombre de propriétés chaotiques universelles (route vers le chaos par doublement de périodes, fenêtre de période trois, attracteur étrange,...), propriétés exposées en section 5.3. Dans le cas où plusieurs applications sont couplées en réseau, le phénomène caractéristique est la synchronisation, c'est-à-dire à une configuration où l'évolution temporelle de toutes ces applications est parfaitement identique. Cette synchronisation peut être partielle ou totale selon la topologie du réseau, la valeur du couplage, les conditions initiales.

Les RIC n'étant pas orientés agents à l'origine, même dans leur version "gaz d'itérations couplées" (GIC), le travail principal de cette thèse du point de vue de la modélisation théorique a été d'intégrer le modèle de calcul du GIC dans le paradigme multi-agent, c'est-à-dire de replacer les mécanismes de calcul propres au RIC dans une conceptualisation agent, qui intègre en particulier la boucle sensorimotrice "perception-décision-action". Les agents résultant de cette fusion sont baptisés agents logistiques du fait qu'ils utilisent des applications logistiques comme générateurs de leur dynamique décisionnelle interne.

L'autre aspect de cette modélisation porte sur l'environnement qui est composé d'un espace géométrique de base sur lequel on définit un ou plusieurs champs en fonction des problèmes. Un champ est pris au sens mathématique le plus large, c'est-à-dire comme une application renvoyant un scalaire ou un vecteur en chaque point de l'espace. Cette conception et structuration de l'environnement n'est pas présente explicitement dans les RIC / GIC et constitue un réel apport méthodologique de notre modèle. Les champs ont un rôle de médiation des interactions (indirectes) entre agents et environnement, ils mémorisent les informations du système, et devi-

ennent de fait des éléments essentiels à l'auto-organisation et à l'adaptation au sein du système multi-agent.

Le résultat de ces deux axes de conception donne lieu au système multi-agent logistique (SMAL). Le SMAL emprunte l'architecture influence-réaction de Ferber [Ferber et Müller, 1996, Ferber, 1997] destinée à englober tout type de SMA réactif sous forme d'un système dynamique générique. Nous avons de fait instancié ce système en y intégrant les concepts clés mentionnés précédemment.

La formalisation du SMA logistique fait l'objet d'une partie complète de ce document car elle est réutilisée dans l'ensemble des développements autour de ce modèle et constitue la contribution principale de cette thèse au domaine des SMA.

Des grandeurs et mesures associées au SMAL.

Un des intérêts majeurs de placer la conception des modèles dans une description à base de systèmes dynamiques déterministes réside dans la possibilité de disposer de grandeurs quantifiables, qui caractérisent la dynamique du système à différents niveaux d'analyse. Ainsi le calcul des coefficients de Lyapunov d'un système dynamique (ou au moins du coefficient maximal) donne des informations sur la stabilité du système : des coefficients négatifs indiquent une stabilité asymptotique, des coefficients positifs une dynamique chaotique. De même la détermination du spectre des coefficients de Lyapunov (l'ensemble de ces coefficients) donne accès à un calcul d'entropie, dite entropie de Kolmogorov qui indique également la stabilité sur la globalité du système et permet d'estimer l'horizon prédictif du système (temps au-delà duquel il est impossible de prévoir ce que deviendra le système). Entre la définition théorique de ces grandeurs et leur détermination expérimentale, il y a cependant beaucoup d'approximations. Ces grandeurs restent cependant des grandeurs clés pour analyser la dynamique et nous proposons dans cette thèse une méthode d'approximation pour les estimer de façon adaptée à la structure du SMAL, notamment dans le cas d'étude du flocking.

Objectifs du modèle SMAL.

Dans un premier temps, notre modèle SMAL vise à disposer d'un cadre unificateur pour les deux types de problématiques de l'intelligence en essaim ciblées par notre étude. Nous tentons de montrer que les divers changements conceptuels qu'il amène par rapport à l'existant ouvrent la voie à de nouvelles analyses et interprétations sur ces problématiques. Malgré son aspect générique, le SMAL ne prétend pas à l'universalité, mais il apporte des arguments —exposés dans cette thèse— qui prônent ainsi une vision nouvelle de la modélisation pour l'intelligence en essaim.

Dans un second temps et pour compléter cet objectif, nous envisageons le SMAL comme une métaheuristique possible pour aborder des problèmes d'optimisation, à l'instar des premiers algorithmes fourmis.

Nous confirmons ainsi que le SMAL constitue un modèle pertinent pour traiter de l'intelligence en essaim. Nous expliquons avec plus de détails dans ce qui suit commencer nous avons évalué le modèle.

4 Evaluation et pertinence du modèle

Principe de l'évaluation du SMAL.

Une fois le SMA logistique posé et formalisé, il convient de montrer concrètement comment il répond aux objectifs que l'on s'est fixés en ce qui concerne spécifiquement l'intelligence en essaim. Pour cela, nous avons suivi une méthodologie comparative : afin de vérifier l'adéquation du SMAL au cadre de l'intelligence en essaim informatique —comme notre projet l'a annoncé en préambule—, un ensemble de situations et de problèmes typiques du domaine a été abordé avec le SMAL et a été comparé aux approches existantes sur le plan de la modélisation, et/ou sur le plan des performances de réalisation lorsque cela fut possible. En fait, nos références de comparaison sont tirées des premiers modèles et algorithmes de l'intelligence en essaim, sans doute les plus emblématiques.

Méthodes d'évaluation.

Nous procédons à l'évaluation du modèle SMAL selon plusieurs axes :
- une évaluation expérimentale pour vérifier certains résultats théoriques connus,
- une validation théorique, par démonstration de quelques propriétés du modèle selon les cas d'étude traités.
- une évaluation par simulation, notamment pour les phénomènes de flocking, accompagné parfois d'une mesure sur la dynamique du système,
- une évaluation comparative et quantitative dans un véritable cas d'optimisation abordé : celui de l'optimisation par colonie de fourmis sur un problème de voyageur de commerce (TSP),
- une évaluation en contexte physique enfin par l'implémentation du SMAL pour réaliser du "flocking" de robots.

5 Contributions de la thèse

Pour clore cette partie introductive, et juste avant de présenter le plan du document, nous entendons résumer les contributions principales de cette thèse autour de notre proposition de système multi-agent logistique (SMAL) :

Un modèle générique. Le modèle SMAL figure parmi les rares propositions de modèle de système multi-agent tentant d'unifier les deux aspects traités dans cette thèse concernant l'intelligence en essaim, à savoir la modélisation de flocking et la modélisation de la stigmergie [10] à base de phéromone dans le cas des fourmis. Il démontre ainsi son aspect générique fort par sa capacité à instancier à partir d'un même schéma théorique des phénoménologies auparavant sans lien apparent et sans modèle commun. Son potentiel en tant que métaheuristique pour l'optimisation confirme cette caractéristique.

[10] Ce concept qui décrit l'effet autocatalytique de la phéromone est développé en section 4.3

Un modèle déterministe et "complexe". Le SMAL est par ailleurs le résultat d'une fusion entre un modèle de calcul typique des systèmes complexes et un schéma structurel influence-réaction générique des systèmes multi-agents réactifs. L'intérêt est de bénéficier d'un côté des méthodologies propres à la science des systèmes complexes (outils d'analyse, de mesure, visualisation...), et de l'autre du paradigme systémique que représente le système multi-agent.

Le SMAL se démarque des approches stochastiques existantes par son caractère fondamentalement déterministe et dynamique, mais restitue l'aléatoire par des non linéarités chaotiques générées par une application logistique régulant l'état interne des agents. Il se démarque également des approches particulaires à base de champs de potentiel, par le recours à la notion d'agent réactif d'une part et d'autre part à une conception de l'environnement comme composé d'un espace géométrique sur lequel sont définis divers champs pris dans leur sens mathématique le plus large. Ces champs constituent la mémoire et la structure de données manipulées par le système multi-agent.

Analyse des mécanismes. Enfin le modèle SMAL permet la conception explicite de mécanismes d'auto-organisation et d'adaptation pour l'intelligence en essaim en définissant le processus sensorimoteur perception-décision-action des agents d'un côté et les champs de l'environnement avec lesquels les agents interagissent de l'autre. De plus, nous mettons en lumière les mécanismes d'auto-organisation par la nature endogène des données manipulées par les agents, c'est-à-dire des données générées par les agents eux-mêmes, contrairement aux mécanismes d'adaptation qui manipulent des données exogènes aux agents, préexistants dans l'environnement (les entrées du système).

Le modèle ainsi posé induit les deux conclusions fondamentales suivantes :
- les phénomènes de type "flocking" sont liés aux mécanismes de couplage et de synchronisation internes aux agents,
- tandis que les phénomènes stigmergiques fourmis sont associés aux mécanismes de contrôle interne aux agents.

Ce dernier constat nous semble le résultat qui résume le mieux cette thèse, puisque le modèle SMAL rend explicite du point de vue mathématique ces deux types de phénoménologies et de mécanismes dans un seul cadre théorique.

6 Plan de la thèse

Outre cette partie introductive, le plan de cette thèse est découpé en trois grandes parties.

Partie II La partie II présente l'intelligence en essaim. Dans un premier temps, un court historique du concept d'intelligence en essaim, ses définitions et ses caractéristiques constituent le corps du chapitre 1, suivi par l'exposé des modèles fondamentaux qui ont structuré ce domaine au chapitre 1.3. Enfin, les principales applications dans le domaine de l'optimisation et de la robotique sont décrites au chapitre 2. Ce panorama dessine également les limitations de ce domaine. Pour dépasser les limitaitons évoquées, on retrace l'évolution des concepts et des approches scientifiques qui ont jalonné la pensée scientifique autour de l'auto-organisation au

chapitre 4. Cette perspective permet de mesurer les enjeux de l'intelligence en essaim sur des bases conceptuelles globales et génériques et comprendre l'intérêt d'emprunter les méthodologies issues de l'étude des systèmes complexes, exposées au chapitre 5. Cela conclue l'état de l'art des concepts et techniques actuellement rattachées à l'intelligence en essaim.

Partie III La partie III traite de la conception du système multi-agent logistique, de ses sources d'inspiration mathématiques et structurelles jusqu'à sa formalisation. Cette partie commence par le chapitre 7 sur les différentes conceptions de l'agent, avec la particularité ici de mettre en regard la conception agent des physiciens avec comme point de départ la particule, et la conception des informaticiens avec comme point de départ les modèles logiques de l'intelligence artificielle. Les systèmes multi-agents réactifs sont plus particulièrement développés, notamment à travers le modèle influence-réaction en section 7.2 puisqu'ils sont une brique essentielle pour la construction du SMA logistique formalisé au chapitre 8. Enfin, des arguments qualitatifs sur les modèles sont exposés pour justifier l'utilisation de modèles chaotiques en tant générateurs des comportements de nos agents logistiques au chapitre 8.3. Ainsi ce chapitre explique que le SMA logistique peut être considéré de source d'inspiration à la fois biologique et théorique.

Partie IV La dernière partie est consacrée aux applications du modèle SMAL ainsi qu'aux résultats d'analyse des simulations et des mécanismes. Cette partie montre la capacité du SMA logistique à traiter de problèmes de l'intelligence en essaim par comparaison aux algorithmes existants lorsque les mesures existent et sont comparables. Les cadres d'applications sont tout d'abord orientés vers la génération des phénomènes d'auto-organisation fondamentaux au chapitre 10, dans lesquels on traite essentiellement des phénomènes de flocking avec le SMAL. Puis des mécanismes d'adaptation réalisables avec le SMA logistique sont expliqués par l'exemple, du plus simple au plus compliqué, au chapitre 11. Ce chapitre s'achève sur une application à la robotique en essaim où notre modèle est implémenté pour réaliser un flocking de robots. Les algorithmes des fourmis logistiques enfin, application de nos agents logistiques aux mécanismes fourmis, font leur apparition au chapitre 12 en terminant sur l'adaptation optimisée pour le problème du voyageur de commerce.

Deuxième partie

Intelligence en essaim, auto-organisation et systèmes complexes

Cette partie présente le domaine principal de notre étude, l'intelligence en essaim, ainsi que les concepts clés qui lui sont indissociables, à savoir l'auto-organisation et les systèmes complexes.

En ce qui concerne l'intelligence en essaim, il s'agit tout d'abord d'en préciser l'origine, les différentes définitions ainsi que les problématiques qu'elle soulève, afin d'expliciter les questions qui sont au fondement de cette forme d'intelligence. En particulier, nous focalisons notre étude autour de deux familles de phénomènes/problématiques typiques du domaine, que sont les mouvements de groupes en "flocking" d'une part et les phénomènes stigmergiques restreints au cas du fourragement chez les fourmis fourmis d'autre part. Des modèles existants et fondamentaux pour ces deux aspects phénoménologiques sont présentés. Ils possèdent leurs versions algorithmiques et leurs champs d'application respectifs, en particulier en tant que métaheuristiques pour l'optimisation. Ces algorithmes sont également décrits en détail car ils constituent des éléments de référence particuliers pour notre travail tout au long de cette thèse.

Nous ne prétendons pas dresser un état de l'art de l'intelligence en essaim qui serait fatalement lacunaire tant ce domaine se ramifie aujourd'hui en divers courants d'applications, mais nous tentons surtout de faire apparaître certaines de ses limitations au travers de quelques algorithmes de référence qui engagent, de notre point de vue, à repenser la modélisation pour pallier ces limitations. Notamment, nous tentons de comprendre ce qui caractérise, et à terme pourrait unifier, les deux types de modélisation existants considérés dans cette thèse, modèle de "flocking" et modèle de fourmis.

Dans l'optique de dépasser les limitations qui auront été identifiées, et pour exposer plus clairement la voie que nous avons empruntée, nous remontons à la racine commune des phénomènes de l'intelligence en essaim qui nous mène à considérer les concepts d'auto-organisation et d'adaptation dans les systèmes. Ces concepts dépassent le cadre biologique ou informatique et sont un sujet d'interrogation scientifique depuis de nombreuses années. La difficulté à théoriser ces concepts et à en mesurer les effets dans les systèmes a conduit à repenser la démarche générale de modélisation et les grandeurs caractéristiques utilisées. Ce programme d'investigation amène à circonscrire le champ dit "des systèmes complexes" qui apporte une méthodologie de conception et la définition de grandeurs clés pour étudier l'auto-organisation sous toutes ses formes. Cela nous conduit au final à présenter plus particulièrement le modèle typique des systèmes complexes auquel nous nous référons par la suite : les réseaux d'itérations couplées.

1

L'intelligence en essaim

1.1 Naissance et définitions du concept d'intelligence en essaim.

1.1.1 Apparition du concept.

La première utilisation et la première définition du terme d'"intelligence en essaim"[11] remontent à l'article de Beni et Wang [Beni et Wang, 1989] de 1989 intitulé "Swarm intelligence in cellular robotics systems". Cet article fondateur du concept proposait une alternative à l'expression de robotique cellulaire pour décrire un groupe de robots coopérant pour accomplir une tâche ou résoudre un problème. Dans un article plus récent [Beni, 2005], Gerardo Beni revient sur cette terminologie et son extension à d'autres contextes scientifiques.

L'auteur explique ainsi que ce terme avait été choisi d'une part pour se démarquer de la désignation de robot cellulaire, relative à celle d'automate cellulaire –thématique scientifique en plein essor à cette période–, mais devenue inadaptée à la description d'unités se déplaçant, opérant et interagissant de façon asynchrone et non séquentielle. Une critique formulée en conférence sur l'inadéquation de ce terme a convaincu Beni d'utiliser finalement un nouveau terme pour désigner le comportement de ces robots, celui d'"essaim". Ainsi apparaît pour la première fois dans un autre secteur que la biologie le terme d'essaim, mais cette fois associé au concept d'intelligence, repris depuis lors par d'autres secteurs disciplinaires.

Notion d'essaim. Beni caractérise la notion d'essaim par un groupe constitué d'un nombre d'au moins une centaine d'entités, mais très inférieur aux populations de particules considérées en physique pour lesquelles des approches statistiques sont nécessaires, soit : $10^2 \leqslant N \ll \mathcal{N}_A$ où $\mathcal{N}_A$ est le nombre d'Avogadro[12]. D'autre part, les entités du groupe sont simples dans leur comportement individuel, sont quasi-identiques, et le contrôle de l'essaim se réalise en mode décentralisé. Enfin comme précisé plus haut, le séquencement des différents comportements individuels n'est pas synchrone, ce qui signifie dans ce contexte que les boucles sensorimotrices perception-décision-action des entités (en l'occurrence des robots ici) opèrent temporellement de façon indépendante –ce qui en temps continu paraît plus naturel que pour un système logiciel à temporal-

[11] "Swarm Intelligence"
[12] De l'ordre de 10^{23}

ité discrète où l'asynchronisme doit être programmé–. La métaphore de l'essaim biologique paraît évidente dans cette définition assez générique que les éthologues [Bonabeau et Theraulaz, 1994a] ont également adoptée pour désigner de façon très générale le comportement des insectes sociaux, comme le relate la section 1.2.1.

Les avantages pressentis de l'essaim de robots par rapport à un système de robots à contrôle centralisé, découlent directement de ces propriétés :

- des unités simples sont plus faciles à produire en masse et sont interchangeables, voire jettables,
- le système est plus fiable, plus robuste aux diverses pannes ou perturbations auxquelles il peut être soumis du fait de la redondance des unités,
- le système en essaim est doté d'importantes capacités d'adaptation à son environnement,
- enfin le système en essaim peut résoudre des problèmes compliqués du fait de ses multiples unités de calcul, bien au-delà des problèmes que peuvent résoudre les entités individuellement.

Ces diverses caractéristiques s'appliquent aussi bien aux "véritables" essaims étudiés en biologie, qui deviennent ainsi des instances concrètes d'un concept plus général. Cependant il s'agit de distinguer les deux approches. Les roboticiens et informaticiens se fixent des objectifs opérationnels pour améliorer la réalisation de tâches complexes par des groupes de robots – un objectif d'ingénierie en réalité – à partir de la description algorithmique des mécanismes. Les biologistes tentent avant tout de comprendre les mécanismes à l'origine des phénomènes par la production de modèles capables de reproduire de façon aussi proche que possible les comportements biologiques observés en comparant leur prédiction avec les mesures expérimentales.

Essaim et intelligence. Beni situe l'intelligence de l'essaim de robots dans sa capacité à traiter à la fois de la matière et de l'énergie comme toute machine, à la fois de l'information comme tout automate informatique. Il définit cette intelligence comme [Beni, 2005] :

> une aptitude à générer des schémas ordonnés de façon imprévisible, schémas sur lesquels une analyse est faite par chaque robot pour optimiser si besoin ses actions en vue d'atteindre un objectif collectif fixé à l'avance (formalisé par une certaine fonction).

Ainsi, le processus d'optimisation consiste à sélectionner les schémas qui répondent le mieux à l'objectif/la fonction. Dans le cas où cette aptitude est applicable à un groupe de robots –elle devient ainsi collective–, on parle d'intelligence en essaim.

C'est dans ce rapprochement paradoxal entre ordre et imprévisibilité que réside le coeur de la notion d'intelligence selon Beni. Les processus imprévisibles sont des sujets d'étude fréquents en science –bien évidemment dans le domaine des probabilités mais aussi dans des systèmes de nature déterministe (la théorie du chaos déterministe a renouvelé les conceptions à ce sujet)– mais c'est lorsqu'ils conduisent à des structures ordonnées qu'ils entrent selon Beni dans le cadre des systèmes "intelligents".

Reste néanmoins à préciser les termes de cette première définition :

- l'ordre du schéma obtenu est le résultat d'un calcul au sens informatique du terme

– l'imprévisibilité pose question à l'informaticien concernant le modèle de calcul utilisé : pour pouvoir générer de l'imprévisible, le modèle de calcul ne doit pas pouvoir être prédit et donc décrit par un autre modèle de calcul, autrement dit il doit être équivalent à une machine de Türing universelle.

1.1.2 Autres visions de l'intelligence en essaim.

Vision de l'intelligence en essaim en biologie. Comme on l'a vu précédemment, de façon paradoxale le concept d'intelligence en essaim n'est pas né directement de la biologie. En revanche la biologie foisonne d'exemples concrets qui sont autant de sources d'inspiration pour l'informatique. D'ailleurs la définition qu'en donnent Bonabeau, Dorigo et Theraulaz dans [Bonabeau *et al.*, 1999a] étend la définition de l'intelligence en essaim au-delà du contexte robotique d'origine :

Définition 1.1.1. *L'intelligence en essaim inclut toute tentative de conception d'un algorithme ou d'un dispositif visant à résoudre des problèmes de façon distribuée, inspirée du comportement collectif des insectes sociaux ou d'autres sociétés animales.*

Le message est clair : les phénomènes biologiques sont la principale source d'inspiration pour l'intelligence en essaim. Ces auteurs décrivent de plus cette forme d'intelligence comme une "intelligence collective émergeant d'un groupe d'agents simples."

Vision de l'intelligence en essaim en ingénierie de l'informatique. Kennedy et Eberhart dans la préface de [Kennedy *et al.*, 2001], tout en adhérant à la vision biologique précédente, réfutent toute notion d'agent associé à cette forme d'intelligence. Leur argument est fondé sur l'exemple des systèmes particulaires qui sont difficilement classables dans la catégorie agent car dotés selon eux de trop peu d'autonomie et de spécialisation, ce qui ne les empêche pas d'exhiber une des formes de l'intelligence en essaim. Cependant, ils restent évasifs sur la nature de l'intelligence mise en jeu. Leur dilemme provient de la distinction dans ce cas précis entre les mécanismes d'adaptation et l'intelligence. Ces deux auteurs n'osent pas se prononcer sur cette question délicate, renvoyant simplement aux réflexions de l'ouvrage [Fogel, 1995].

La préface de Merkle et Blum dans l'ouvrage [Blum et Merkle, 2008] propose une définition pragmatique de l'intelligence en essaim du point de vue informatique :

Définition 1.1.2. *L'intelligence en essaim est une discipline de l'intelligence artificielle moderne qui traite de la conception de systèmes multi-agents en vue d'applications telles que l'optimisation et la robotique*[13].

Ces auteurs n'y font plus référence à la bio-inspiration.

[13]Swarm intelligence is a modern artificial intelligence discipline that is concerned with the design of multiagent systems with applications, e.g., in op- timization and in robotics.

Notre positionnement. Parmi ces diverses conceptions, nous tenons à conserver l'aspect bio-inspiré indissociable selon nous de l'intelligence en essaim, même si les algorithmes résultants sont parfois très éloignés de leur source d'inspiration biologique.

L'autre dimension de l'intelligence en essaim se situe pour nous dans la modélisation à base d'agents affirmée à la fois par les biologistes et par certains informaticiens. Ce point de vue a plus de conséquences qu'il n'y paraît : comme Kennedy et Eberhart, nous faisons une différence entre les modélisations de type particulaire et de type agent, mais contrairement à leur point de vue nous insistons justement sur le fait que les modèles de l'intelligence en essaim doivent s'extraire des approches particulaires largement répandues dans les modèles de type flocking comme nous l'expliquerons en section 1.3. D'une part, le modèle de particule est directement issu des modèles des physiciens pour étudier la matière inerte ; ce modèle est très bas niveau du point de vue sémantique ; il a d'ailleurs l'avantage d'être simple dans sa formulation. L'utilisation de la notion de champ de potentiel lui est associée. Ce modèle est encore très extérieur à l'aspect décisionnel qui n'existe pas en tant que mécanisme interne dans la particule. Nous discutons précisément de cette question en section 7.1 à propos des "agents browniens". Ainsi nous ne nous satisfaisons pas de l'utilisation de ces modèles particulaires pour ce qui concerne le monde vivant. Seul le concept d'agent peut permettre selon nous d'embrasser l'intelligence en essaim dans toute ses dimensions.

Enfin, Merkle et Blum insistent sur les aspects applicatifs de l'intelligence en essaim. Il est vrai que les métaheuristiques obtenues par bio-inspiration sont reconnues comme particulièrement efficaces sur certains problèmes difficiles comme nous le verrons en section 2.1.1, mais ces applications ne verraient pas le jour sans l'analyse préalable des mécanismes en essaim identifiés comme responsables de cette efficacité. Ainsi une réflexion sur l'auto-organisation et l'adaptation est nécessaire ; elle doit être soutenue par une phase parfois longue de simulations informatiques et fait partie intégrante de l'intelligence en essaim. L'historique de la création des algorithmes fourmis et des algorithmes particulaires en atteste largement dans la suite de ce chapitre.

Résumons notre point de vue :

Définition 1.1.3. *Nous considérons l'intelligence en essaim (informatique) comme un domaine fortement bio-inspiré dont l'objectif est de modéliser au moyen de systèmes multi-agents, les mécanismes d'auto-organisation et d'adaptation observés dans le vivant. Ces modèles donnent lieu à terme à des algorithmes permettant de simuler des phénomènes naturels de façon logicielle (sur ordinateur) ou matérielle (avec des robots), ou bien de servir de métaheuristiques pour les problèmes de l'intelligence artificielle.*

1.2 Inspiration biologique et caractéristiques

1.2.1 La métaphore biologique

Cette section traite de l'intelligence en essaim sous l'angle de la biologie et des grands types de problématiques qu'elle aborde. La biologie est en effet une source d'inspiration forte pour le domaine de l'intelligence en essaim informatique comme nous le réaffirmons dans notre définition, qui y trouve des métaphores et des mécanismes théoriques pour initier de nouvelles idées

d'algorithmes dits "bio-inspirés". Nous listons donc en particulier trois problématiques identifiées en biologie qui se révèlent cruciales à modéliser informatiquement : la prise de décision décentralisée, l'équilibre des comportements d'exploration et d'exploitation, le déplacement collectif.

Problématiques des comportements des essaims biologiques. L'article [Beekman *et al.*, 2008] dresse un panorama synthétique des problématiques qui concernent prioritairement l'intelligence en essaim. Nous en résumons à présent les grandes lignes.

Problématique de la prise de décision décentralisée. La sociabilité des insectes va de pair avec la prise de décision collective. Cette socialisation, qui s'opère par échange d'information entre individus est la condition d'une coordination des actions individuelles pour rendre cohérent le schéma collectif obtenu. C'est le cas lors de la recherche de nourriture ou d'un nouveau lieu de nidification, déterminer le moment de la reproduction ou le niveau de séparation des rôles lorsque cela est nécessaire. Ces aspects sont très fréquents dans les sociétés fourmis ou dans les essaims d'abeilles. La décision collective résulte d'interactions individuelles locales et du contexte environnemental local [Camazine *et al.*, 2001]. Cela définit une prise de décision décentralisée dans ces sociétés animales.

On peut exhiber certains principes de fonctionnement permettant aux insectes de réaliser des tâches complexes. La recherche de nourriture utilise par exemple le recrutement individuel par contact direct d'un individu à l'autre (interaction directe) ainsi que le recrutement de masse opéré par le biais de phéromone (interaction indirecte).

Problématique de l'équilibre exploration-exploitation. Afin de garantir une flux constant de ressources disponibles, les sociétés d'insectes maintiennent un équilibre entre les phases d'exploration destinées à la découverte de nouveaux sites de l'environnement et les phases d'exploitation pour tirer profit des informations déjà disponibles. Ces deux faces de la même pièce constituent un mécanisme indispensable à la survie de la société à long terme et est à la base de l'adaptation dynamique aux changements de l'environnement.

Problématique du déplacement collectif. Le déplacement collectif que nous désignerons dans tout ce document par le terme de "flocking", est un phénomène central pour l'intelligence en essaim. Non seulement il résulte de mécanismes auto-organisationnels[14] mais il peut s'observer à plusieurs échelles, pour diverses espèces et divers niveaux d'intelligence des individus. Ainsi des similarités existent entre le vol groupé d'oiseaux, les bancs de poisson, les troupeaux d'animaux et ... les foules humaines. Comme l'ont montré des travaux récents de Cousin et Krause [Dyer *et al.*, 2008], la dynamique de foules d'humains dans certaines conditions privées de communications directes, est non seulement assimilable à d'autres comportements de groupes animaux mais est contrôlable par quelques individus qui ont connaissance d'un chemin. Les applications au contrôle des situations de panique ou de stress important sont envisagées à partir de ces études des comportements collectifs humains [Helbing *et al.*, 2000].

[14] cf. chapitre 4

Pour des espèces individuellement moins évoluées, deux configurations apparaissent : soit le groupe possède des membres plus expérimentés qui jouent le rôle de leaders pour le groupe, soit il n'y en a pas et le consensus émerge par auto-organisation spontanée ou bien du fait de caractères génétiques spécifiques (le choix de directions privilégiées serait d'origine génétique chez certaines espèces [Berthold et Querner, 1981]). Ce second phénomène semble assez similaire à des systèmes physiques qui passent d'une phase désordonnée à une phase ordonnée par une transition de phase, mais son mécanisme reste encore énigmatique.

La plupart des modèles actuels sont de fait théoriques et la simulation informatique est d'une aide précieuse pour vérifier la pertinence de ces modèles. Le déplacement des nuées de crickets d'Afrique semblent par exemple bien simulé par des modèles de déplacement du type Vicsek (cf. section 1.3.2). Ou bien encore, les simulations sur la construction de toiles chez les araignées sociales ont pu corroborer les hypothèses faites par les biologistes : l'appétence des araignées pour leur propre soie est le facteur principal de leurs actions de déplacement [Bourjot *et al.*, 2003].

Mentionnons que les études expérimentales sur les déplacements de groupes animaux sont difficiles à mettre en œuvre concrètement car les protocoles d'observation et de mesure manquent souvent de la précision qui est indispensable pour élaborer des modèles fiables. Des études récentes basées sur des reconstructions 3D à grande échelle de nuées d'étourneaux dans l'article [Ballerini *et al.*, 2008] font une percée majeure dans l'observation expérimentale des phénomènes de "flocking". Deux caméras filment en stéréovision une nuée d'oiseaux, puis par reconstruction 3D au moyen d'algorithmes de traitement d'images, la nuée est synthétisée sur ordinateur. Une fois la nuée virtualisée, les mesures sur le champ de vitesse devient possible. Le résultat principal de l'article est que le voisinage d'interaction entre oiseaux, n'est pas un voisinage métrique, i.e. fondé sur une distance fixée à l'avance, mais un voisinage topologique, prenant seulement en compte les n plus proches voisins indépendamment de leur distance (jusqu'à la limite de perception des individus). Cette précision des mesures autorise la conception de modèles plus fidèles à la réalité biologique.

Remarques finales. Les problématiques qui viennent d'être évoquées sont présentes dans les deux archétypes que nous avons choisi d'aborder et de modéliser :

- le phénomène de "flocking" ou formation de groupes animaux, que l'on associe initialement au phénomène de nuées d'oiseaux mais qui peut être étendu à d'autres phénoménologies comme les troupeaux d'animaux, les bancs de poissons, et les mouvements de foules humaines, est au coeur des problématiques du déplacement collectif et pose la question de la prise de décision décentralisée : en effet comment le choix local d'une direction commune est-il effectué ?
- les mécanismes à l'œvre dans le cas du fourragement des colonies de fourmis impliquent à la fois la prise de décisions décentralisée, selon une modalité directe (contact) ou indirecte et la problématique d'exploration-exploitation. Ce type de mécanisme est théorisé par le concept de stigmergie explicité en section 4.3 consacrée à l'auto-organisation biologique.

Notre ambition est d'intégrer les aspects de ces deux types de phénomènes dans un même modèle théorique.

1.2.2 Caractéristiques génériques des approches en essaim

Les différentes problématiques posées par les systèmes biologiques sont une source importante d'inspiration pour la conception des systèmes artificiels. Mark Millonas a tiré de la métaphore fourmi une proposition de caractérisation en cinq principes des systèmes d'intelligence en essaim dans l'article [Millonas, 1993]. Nous en résumons ci-dessous les grandes lignes.

Principe de proximité. Ce premier principe pose la nécessité d'une capacité des entités du groupe à répondre dans une localité spatiale et temporelle aux signaux/stimuli de l'environnement. Cela implique une capacité de calcul local, calcul dont l'objectif est de décider d'un comportement qui maximiserait au mieux l'utilité pour l'activité du groupe entier (par exemple le fourragement, la construction d'un nid, le déplacement de groupe,...).

Principe de qualité. Le groupe doit pouvoir répondre aussi à des critères de qualité par rapport à l'objectif recherché (qualité de la source de nourriture, de l'emplacement du nid,...). Il s'agit là d'optimiser un critère de qualité pour l'essaim.

Principe de réponse diversifiée. Le groupe doit pouvoir répondre dynamiquement aux éventuels changements de l'environnement, et donc diversifier autant que possible ses modes de fonctionnement.

Principe de stabilité. Ce principe vient compléter et limiter le précédent. Le groupe doit également maintenir une forme de stabilité afin d'éviter de basculer d'un mode de fonctionnement à l'autre au moindre changement de l'environnement, ce qui serait une perte d'énergie considérable.

Principe d'adaptabilité. Le changement de mode de fonctionnement, lorsque le mode courant n'est plus suffisamment satisfaisant, s'opère par adaptation à une nouvelle situation environnementale.

Bilan.

La conclusion tirée par Millonas sur le comportement de l'essaim [Millonas, 1993] est qu'il doit se maintenir entre un état chaotique qui génère de la diversité et un état ordonné qui permet d'exploiter une ressource ou d'achever la réalisation d'une activité collective. L'intelligence selon Millonas intervient au moment des changements de comportement du groupe pour sélectionner une meilleure ressource ou une meilleure activité au sens d'un certain critère de qualité, ce qui implique des capacités de l'essaim pour l'optimisation.

La notion d'état chaotique n'est pas vraiment explicitée dans l'article de Millonas, ni formalisée mathématiquement. Elle a un caractère d'intuition à ce niveau sans que l'on puisse l'extrapoler sur une base tangible. L'article reste proche de concepts physiques issus de la thermodynamique et reste sur une vision globale des processus.

Le manque de formalisation et d'analyse théorique au niveau local du fonctionnement de l'essaim est le grief majeur que nous faisons à propos de la modélisation en intelligence en essaim de façon générale.

1.3 De la bio-inspiration aux algorithmes emblématiques : "flocking" et colonies de fourmis

Cette section introduit les modèles qui nous serviront de socles d'étude de l'intelligence en essaim informatique. Elle vise également à montrer comment s'est opérée la "bio-inspiration" sur ces cas particulier d'étude.

Il s'agit dans un premier temps de présenter deux modèles de "flocking" : le modèle de Reynolds et ses principes algorithmiques, ainsi qu'une version particulaire proposée par Vicsek, alternative au modèle de Reynolds, plus simple et plus formalisée mathématiquement. Puis nous retraçons les étapes d'élaboration du modèle fourmi, depuis les analyses statistiques expérimentales et la loi de probabilité établies par Deneubourg, à l'algorithme emblématique des fourmis de Dorigo.

Ces exemples clés nous fournissent une base comparative pour évaluer notre propre proposition de modèle d'agents logistiques en partie IV.

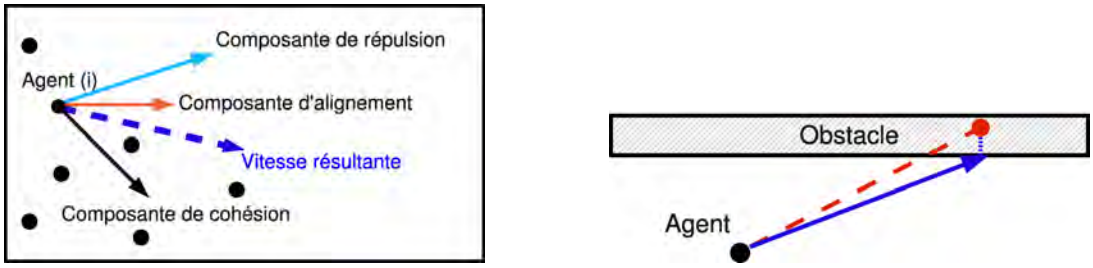
1.3.1 Boids de Reynolds

Intérêt et contributions des Boids de Reynolds. Le travail de Reynolds sur la synthèse du flocking pour les besoins de l'infographie désigné sous le nom de "Boids" de Reynolds est une des premières et plus remarquables propositions de règles permettant de simuler un groupe d'entités virtuelles informatiques se déplaçant collectivement en évitant des obstacles. Bien que peu formalisé, l'article [Reynolds, 1987] apporte un ensemble de spécifications précises tout en laissant le choix de la méthode d'implémentation. L'objectif premier de Reynolds est la simulation et la performance algorithmique, non l'explication des phénomènes décisionnels internes aux individus du groupe. L'exigence de réalisme pour la simulation donne au projet son caractère bio-inspiré : les tentatives d'évaluation par rapport à des données expérimentales restent peu nombreuses (comparaison statistique, schémas de formation,...), mais les simulations obtenues ont pu malgré tout convaincre les biologistes. En réalité ce sont les systèmes particulaires déjà existants en infographie (notamment les modèles de Reeves [Reeves, 1983]) —avec lesquels les simulations de fumées, vagues, nuages sont réalisables— qui sont la source d'inspiration algorithmique des Boids de Reynolds.

L'apport de l'algorithme de Reynolds à ce moment-là tient dans sa rupture avec l'approche de modélisation consistant à calculer individuellement les trajectoires des membres du groupe de façon globale. Reynolds propose en effet un calcul local d'interactions basées sur des perceptions locales. Ces mécanismes ont pour effet de voir émerger la formation des groupes de façon spontanée au niveau global du système. Reynolds parle alors d'acteurs, concept précurseur de nos agents actuels (réactifs en l'occurrence).

Règles de conception. Les trois règles fondamentales édictées par Reynolds à la base des décisions des "Boids" consistent en [Reynolds, 1987] :

- la séparation : éviter les collisions avec les plus proches voisins, ce qui revient à respecter une distance minimale avec les autres,
- l'alignement : adapter sa vitesse à celle de ses voisins, et rester dans la direction commune de déplacement,
- la cohésion : rester proche de ses voisins en se rapprochant du centre de la nuée.



(a) Illustration des trois règles de base dans l'algorithme de Reynolds.

(b) Principe de l'évitement d'un obstacle.

FIG. 1.1 – Règles de déplacement et d'évitement d'obstacles chez Reynolds

Quelques règles ou précisions complémentaires viennent ajuster l'application de ces trois règles de base :

- Chacune de ces règles induit une accélération de l'agent (c'est-à-dire la modification de son vecteur vitesse ; cf. figure 1.1(a)). Un facteur d'importance relative est associé à chacune de ces accélérations. En règle générale, on fait la moyenne des trois, mais si une action devient plus urgente que les autres, par exemple se séparer parce que deux agents sont trop proches, alors on renforce le poids de la règle qui corrige l'événement perçu. On peut également définir un ordre de priorité sur les règles et une limite à la norme des accélérations pour chaque agent.
- La règle d'alignement ne prend pas en compte la position des agents, tandis que les deux autres règles ont besoin de cette information.
- D'autre part, la règle de cohésion ne s'applique finalement qu'au centre du voisinage de perception et non au centre de la nuée globale. Il s'agit d'un calcul barycentrique sur les agents dans le voisinage de perception. Cette règle modifie peu le déplacement d'agents dans la nuée, mais ramène ceux du bord vers le centre.
- Cette même règle autorise la division du groupe en plusieurs parties, notamment pour gérer le passage d'un obstacle, contrairement aux modèles à forces centrales ou à agent "leader". En revanche la localité associée à l'application de cette règle rend la réunion de deux parties distinctes délicate puisque la perception limitée des agents ne leur permet pas forcément de réunir le groupe.

En termes d'implémentation, chaque règle utilise des modélisations ou paramètres spécifiques :

- La perception limitée est implémentée par une sphère de rayon limité et une sensibilité inversement proportionnelle à l'exponentielle de la distance. Cette perception est en général réduite à un cône pour favoriser la perception dans le sens du mouvement.
- La séparation et la cohésion utilisent des forces d'attraction et de répulsion inversement

proportionnelles au carré de la distance.

Une remarque pour conclure cette description des principes de base de la modélisation des boids : le modèle de perception limitée correspondrait davantage selon Reynolds à une situation biologique de bancs de poissons ou de troupeaux d'animaux que de nuées d'oiseaux, dans le sens où les oiseaux sont connus pour avoir une perception visuelle très étendue, ce qui n'est pas le cas dans l'implémentation informatique. Cette observation est cepedant contrebalancée par les résultats d'analyse expérimentales évoquées en section 1.2.1 dans les travaux de [Ballerini *et al.*, 2008] qui tendent à montrer que cette capacité perceptive importante dans le cas des oiseaux n'est pas exploitée en situation de flocking, où seuls les plus proches voisins sont des sources d'influence.

L'évitement d'obstacles. Deux approches principales sont développées pour l'évitement d'obstacles :

- Une approche à base de champs de forces : un champ de force répulsive est présent autour de l'obstacle. Mais cette technique recontre plusieurs défauts : certaines configurations d'incidence de l'agent sur l'obstacle ne produisent aucune modification (ex. : longer l'obstacle, ou l'aborder exactement de front), de plus la vision périphérique des agents peut produire des mouvements inattendus (ex. : se retourner sur l'obstacle).
- Une approche "guidage-pour-éviter"[15] donnant de meilleurs résultats de simulation, consiste pour l'agent à se laisser guider par l'obstacle. Plusieurs variantes existent, en voici deux des plus efficaces :
  - L'agent utilise un point test situé en avant dans sa direction de déplacement —c'est le point où il se situerait quelques pas de temps plus tard si aucun changement de direction ne se produisait (cf. figure 1.1(b))— à partir duquel il calcule un nouvel angle de déplacement si ce point entre en contact avec l'obstacle. Cette version simple est efficace pour un coût computationnel réduit.
  - L'agent suit le bord de l'obstacle en estimant la silhouette de l'arête la plus proche de l'obstacle. Cela implique pour l'agent d'avoir accès à une information sur la forme de l'obstacle qui lui permette de calculer sa silhouette par projection. Cette seconde version est plus lourde à mettre en oeuvre.

Une revue de ces différentes techniques accompagnées des graphiques explicatifs sont disponibles dans le document [Reynolds, 1988].

Remarques de conclusion. De nombreuses simulations et explications complémentaires sur le modèle de Reynolds sont visualisables sur le site associé à la référence [Reynolds, 1999].

Nous estimons que le modèle de Reynolds relève d'une vision très mécaniste et très physicienne du déplacement collectif, finalement assez proche d'un modèle particulaire : nous le montrons dans la suite par comparaison avec le modèle de Vicsek. La nécessité d'une détermination des voisins par calcul de distances et l'utilisation de champs de forces dérivant de champs de potentiels en découlent également.

[15] "steer-to-avoid" dans le texte

1.3.2 Le modèle de Vicsek et les systèmes particulaires

Un des modèles les plus connus chez les physiciens qui étudient les phénomènes complexes de la matière dite “active” —par opposition à la matière inerte—, est le modèle de système de particules autoguidées de Tamas Vicsek dans l'article [Vicsek *et al.*, 1995]. La notion de particule autoguidée attribue à la particule classique des physiciens une certaine autonomie vis-à-vis de l'origine de son mouvement : on ne s'intéresse plus à la source d'énergie des particules mais à l'influence des interactions interparticulaires sur la dynamique globale. Ce modèle est à la base de nombreux développements autour de la matière active et des systèmes biologiques : mouvements des matériaux granulaires, migration bactérienne, phénomènes de flocking, déplacement des fourmis,.... Ce modèle constitue, pour ce qui concerne le flocking et ses variantes, une version particulière du flocking façon Reynolds, écrit sous forme d'un système dynamique classique où la norme de la vitesse est constante, là où Reynolds propose un ensemble de règles de conception peu formalisées [Reynolds, 1987].

Les équations du mouvement.

Le modèle est composé de :
- N particules de position et vitesse notées respectivement $\mathbf{x_i}$ et $\mathbf{v_i}$ pour une particule i,
- un espace géométrique d'évolution qui est un tore à deux dimensions de taille L,
- un voisinage d'interaction entre particules de rayon r.

La vitesse de chaque particule est caractérisée par une norme constante v et un angle θ qui représentera la variation de la vitesse, on se place donc implicitement en coordonnées polaires. Au départ d'une simulation, les particules sont placées aléatoirement dans l'espace et leurs angles sont aléatoires également.

Les équations dynamiques du système s'écrivent en fonction de la position et de la vitesse par le système dynamique :

$$\begin{cases} \mathbf{x_i}^{t+1} & = \mathbf{x_i}^{t} + \mathbf{v_i}^{t} \\ \theta_i^{t+1} & = \langle \theta^t \rangle_{i,r} + \triangle\theta \\ \text{avec } \langle \theta^t \rangle_{i,r} & = \arctan\left(\dfrac{\langle \sin(\theta^t) \rangle_{i,r}}{\langle \cos(\theta^t) \rangle_{i,r}}\right) \end{cases} \tag{1.1}$$

où :
- $\langle \theta^t \rangle_{i,r}$ représente l'angle de la moyenne spatiale des directions des voisins sur le voisinage de rayon r centré sur la particule i,
- $\triangle\theta$ est un bruit aléatoire uniforme sur $[-\frac{\eta}{2}, \frac{\eta}{2}]$.

La vitesse est constante en norme, c'est pourquoi le système intègre plutôt l'angle que l'écriture complète de la vitesse [Vicsek *et al.*, 1995].

Résultats de simulation.

La variation aléatoire d'angle $\triangle\theta$ joue le rôle d'une température pour le système. L'autre paramètre important du modèle est la densité $\rho = \frac{N}{L^2}$ de particules dans l'espace. En fonction

de ces deux paramètres, les simulations produisent différents types de résultats (les simulations dans l'article original mettent en jeu $N = 300$ particules avec une vitesse $v = 0,03$) :

- pour une densité faible et un bruit faible, on obtient des groupes de déplacement cohérent mais qui ont des directions quelconques,
- si on augmente la densité et le niveau de bruit, on aboutit à des comportements individuels aléatoires légèrement corrélés,
- enfin si on a une densité de particule forte mais un bruit faible, alors tout le groupe adopte un déplacement d'ensemble dans une même direction qui apparaît spontanément.

Discussion.

Quelques remarques pour finir :

- Les particules utilisées ici sont bien autopropulsées puisque rien n'indique d'où elles tirent leur énergie cinétique.
- Les particules sont autoguidées car elles fixent spontanément leur direction de déplacement commune. On peut également parler d'un phénomène d'auto-organisation mu par une interaction de groupe. Cette interaction est radicale dans le sens où on remplace à chaque pas de temps l'angle courant par la moyenne des angles des plus proches voisins. Dans ce cadre, la particule n'a que très peu de libre arbitre et dépend entièrement de son entourage.
- Le modèle d'origine de vicsek n'implique aucune force de cohésion ou de répulsion, contrairement au modèle de Reynolds. La gestion des collisions n'est pas centrale ici, et la cohésion se développe spontanément par simple moyenne spatiale des angles de vitesses.

Ce modèle simple sert de base à l'étude de nombreux phénomènes d'agrégation et de déplacement d'entités mobiles dans un environnement. Cependant, les particules impliquées ici peuvent difficilement être considérées comme des agents, ou au mieux des agents très basiquement réactifs. Ils ne disposent en effet pas de véritable état interne si ce n'est leur position et leur vitesse, attributs traditionnels de la particule en physique. De plus, les grandeurs de position et de vitesse doivent être gérées de façon globale au système et entrent en contradiction avec le principe de proximité énoncé par Millonas d'un calcul uniquement local.

La formulation mathématique simplifiée autorise néanmoins des analyses fines comme le montre l'étude mathématique des propriétés de convergence établies par Cucker et Smale dans [Cucker et Smale, 2007]. Enfin, ce modèle est similaire dans ses équations dynamiques simplifiées aux algorithmes d'essaims particulaires exposés en section 2.1.2, coefficients stochastiques mis à part.

1.3.3 Le double pont de Deneubourg et l'algorithme fourmi de Dorigo

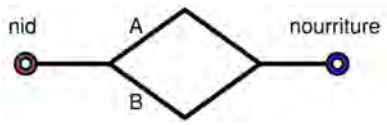
Nous retraçons ici en résumé la genèse des algorithmes dits "fourmis" actuels. L'expérience dite "du double pont" sur les fourmis est à l'origine de la conception des algorithmes fourmis. Le cadre d'étude et de modélisation est celui du fourragement, mais l'algorithme obtenu s'est depuis généralisé à bien d'autres situations comportementales.

Ces cas d'études et la modélisation qui l'accompagnent sont pour nous fondamentaux puisqu'ils nous permettent de confronter notre approche de modélisation à celle existante en dernière partie

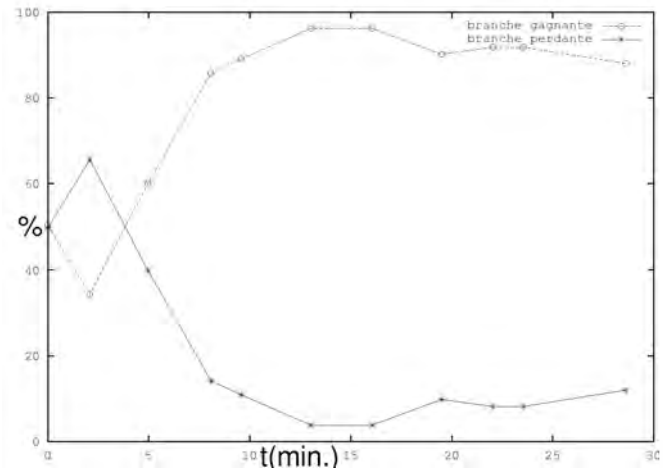
de ce document (cf. chapitre 12).

Brisure de symétrie du double pont

L'expérience du double pont a été mise au point par Jean-Louis Deneubourg et est présentée dans l'article [Deneubourg *et al.*, 1990]. Elle vise à expliquer le mécanisme de recherche collective de nourriture chez une espèce de fourmi argentine appelée *Linepithema humile*. Le protocole expérimental consiste dans un premier temps en un double pont reliant d'un côté le nid, et de l'autre une source de nourriture (cf. figure 1.2(a)). Deux chemins d'égales distances notés A et B séparent ces deux points extrêmes. Le résultat de l'expérience est la sélection collective d'un des



(a) Protocole expérimental.



(b) En abscisse le temps en minutes. En ordonnée le pourcentage de passage.

FIG. 1.2 – Expérience du double pont pour l'étude du fourragement chez les fourmis. Données tirées de [Deneubourg *et al.*, 1990].

deux chemins seulement par la colonie de fourmis. Il s'agit d'un processus d'auto-organisation de la colonie qui brise la symétrie du problème selon les auteurs. Il doit être noté cependant que le résultat de cette expérience dépend de l'espèce concernée, certaines espèces de fourmis explorant systématiquement les deux branches [Beckers *et al.*, 1990, Bonabeau et Theraulaz, 1994b].

Résultats expérimentaux. Le résultat de l'expérience donne une évolution globale tracée sur le graphique 1.2(b). Il montre clairement le choix collectif pour une seule et même branche au bout d'une dizaine de minutes.

Modèle décisionnel. Le mécanisme de dépôt de phéromones, substances chimiques utilisées entre autres par les fourmis pour échanger de l'information, est connu. Les hypothèses pour la modélisation de cette expérience ont été les suivantes :

- L'évaporation de phéromones n'est pas prise en compte du fait de la durée de l'expérience d'une heure environ.
- La quantité de phéromone déposée est constante : une unité pour chaque fourmi.

Le modèle explicatif proposé par Deneubourg est un processus stochastique dont les caractéristiques sont décrites ci-dessous [Deneubourg *et al.*, 1990, Bonabeau *et al.*, 1999b] :

- A_i (respectivement B_i) est le nombre de fourmis ayant choisi la branche A (respectivement B) après que la $i$ème fourmi a passé le pont : i agit comme un pas de temps pour l'algorithme. Chaque fourmi dépose une quantité 1 de phéromone sur la branche sur laquelle elle vient de passer (cf. schéma 1.2(b)).
- P_A (respectivement P_B) est la probabilité que la $(i+1)$ème fourmi choisisse la branche A (respectivement B) et s'écrit :

$$P_A = 1 - P_B = \frac{(K + A_i)^n}{(K + A_i)^n + (K + B_i)^n} \tag{1.2}$$

La formule 1.2 rend compte des données expérimentales le plus finement pour les valeurs de paramètres $K = 20$ et $n = 2$. Les simulations utilisant cette formule reproduisent la même forme d'évolution.

Discussion. Quelques points de remarque et de discussion peuvent être évoqués :

- Le problème n'est pas un problème d'optimisation de distance parcourue ici, puisque les deux branches sont d'égale distance, ce qui laisse supposer un comportement d'auto-organisation intrinsèque au fonctionnement de la colonie. Mais ce fonctionnement n'apparaît pas explicitement à travers la modélisation stochastique, qui selon nous explique la statistique obtenue (c'est donc une modélisation statistique) mais pas vraiment la dynamique interne de décision de chaque individu (modèle mécaniste du phénomène).
- Ce phénomène d'auto-organisation provient pour une part au moins de la forme de la fonction de décision stochastique 1.2. Or cette probabilité peut se réécrire sous la forme :

$$P(x) = \frac{1}{1 + x^n} \quad \text{avec } x = \frac{K + B_i}{K + A_i} \tag{1.3}$$

  On peut toujours faire en sorte de garder x entre 0 et 1 dans cette formule en changeant si besoin par la variable $y = \frac{1}{x}$. Pour $n = 2$, cette loi de probabilité est un cas particulier de la loi de Cauchy en théorie des probabilités. Cette loi a la particularité de n'avoir aucune moyenne ni aucune variance calculable, et d'ailleurs aucun moment de quelque ordre que ce soit [16]. Cette loi de probabilité restitue bien les données expérimentales mais ne dit rien de ses propriétés ; la théorie des probabilités est en diffculté pour nous apporter une analyse théorique *a priori* ou une prédiction ; seule la simulation expérimentale permet de connaître *a fortiori* le comportement du système.

Malgré ses quelques inconvénients d'ordre théorique, et compte tenu de ses résultats cohérents avec l'expérience, cette loi de décision du modèle fourmi de Deneubourg a donné naissance à l'algorithmique fourmi développée par Dorigo dont les principes de base sont présentées dans sa toute première version dans le paragraphe qui suit. En effet, la famille des algorithmes fourmis est née de l'observation du choix systématique du chemin le plus court dans l'expérience du double pont de Deneubourg lorsque les deux branches sont de longueurs différentes, ce qui a initié des applications vers le secteur de l'optimisation. La formulation complète de l'algorithme

[16] Elle ne vérifie pas non plus le théorème de la limite centrale qui dit que la moyenne de N variables indépendantes de même loi converge vers une loi normale pour $N \to \infty$.

de Dorigo fait l'objet de la section 2.1.3. Nous insistons tout particulièrement dans ce qui suit, sur le processus de bio-inspiration qui procède par traduction de phénomènes biologiques en mécanismes de calcul.

Principes de l'algorithme ACO.

Famille des algorithmes fourmis. La famille des algorithmes ACO ("Ant Colony Optimization") a été développée par Marco Dorigo puis par d'autres depuis le début des années 90 [Dorigo, 1992]. Comme mentionné précédemment, l'observation du choix du chemin le plus court dans l'expérience du double pont a motivé l'application de ces algorithmes comme méta-heuristiques pour des problèmes d'optimisation combinatoires comme le voyageur de commerce (TSP) symétrique ou asymétrique (cf. section 2.1.3 pour une spécification du problème). Ce problème classique en informatique est approprié pour poser les principes de fonctionnement de cet algorithme. L'extension du double pont, déjà représenté par un graphe minimaliste, à des graphes de taille quelconque, paraît naturelle.

Problème du voyageur de commerce. Rappelons l'énoncé de ce problème classique du voyageur de commerce. Le problème s'exprime très simplement en langage naturel : un voyageur de commerce doit visiter N villes une fois seulement en minimisant la distance totale parcourue. En termes mathématiques, il s'agit de trouver le(s) cycle(s) hamiltonien(s) le plus court d'un graphe pondéré, i.e. le cycle comportant tous les noeuds du graphe une fois seulement dont la somme des poids est minimale. Le graphe est en général un graphe complet, symétrique lorsque le graphe est non orienté, ou asymétrique dans le cas contraire. Voici les notations utilisées par la suite : le problème se compose d'un graphe $G = (N, A)$ où N est l'ensemble des noeuds et A celui des arcs dont les poids correspondent aux distances d_{ij} entre noeuds. Nous ferons mention implicite à ce problème pour traiter de l'algorithmique fourmi.

"Les" algorithmes fourmis. Rappelons simplement quelques jalons dans ce qui est devenu une "grande" famille des algorithmes ACO dans la littérature (cf. tableau 1.1 tiré de [Dorigo et Stützle, 2004a]), dans l'ordre chronologique (les noms des algorithmes sont cités en anglais) avec la mention de leur application au problème du voyageur de commerce et leur référence principale :

Principes de fonctionnement bio-inspirés. Des principes généraux inspirés des observations biologiques gouvernent ces algorithmes qui servent de base aux autres variantes dans le cadre des problèmes d'optimisation déjà cités :

- Construction d'une solution par progression (avancement) probabiliste :
  on distingue deux phases de fonctionnement pour l'agent fourmi artificiel : l'aller et le retour. La phase aller correspond à la découverte d'un chemin vers la source de nourriture (c'est-à-dire l'objectif/le noeud final), la phase retour permet le retour au nid (deuxième dépôt de phéromone) après avoir atteint l'objectif. Des variantes existent à ce niveau, notamment on peut éliminer le dépôt de phéromones à l'aller et ne maintenir que le dépôt du retour. Le choix d'un nouveau noeud de déplacement se fait selon une loi de probabilité

Algorithme ACO	TSP	Références principales
Ant System (AS)	oui	Dorigo [Dorigo, 1992, Dorigo *et al.*, 1996]
Elitist AS	oui	Dorigo [Dorigo, 1992]
Ant-Q	oui	Gambardella [Gambardella et Dorigo, 1995]
Ant Colony System	oui	Dorigo [Dorigo et Gambardella, 1997]
Max-Min AS	oui	Stützle [Stützle et Hoos, 1997]
Rank-based AS	oui	Bullnheimer [Bullnheimer *et al.*, 1999]
ANTS	non	Maniezzo [Maniezzo et Colorni, 1999]
Hyper-cube AS	non	Blum [Andrea *et al.*, 2001]

TAB. 1.1 – Nomenclature et évolution des algorithmes par colonies de fourmis, tiré de [Dorigo et Stützle, 2004a].

qui dépend du champ de phéromones local établi par la colonie jusque-là. Elle est de la forme :

$$p_{ij}^k = \begin{cases} \frac{\tau_{ij}^\alpha}{\sum_{l \in \mathcal{N}_i^k} \tau_{il}^\alpha} & \text{, si } j \in \mathcal{N}_i^k \\ 0 & \text{, si } j \notin \mathcal{N}_i^k \end{cases} \tag{1.4}$$

où $\mathcal{N}_i^k$ est l'ensemble des noeuds adjacents au noeud i, c'est-à-dire ceux avec lesquels il partage une arête. On retrouve dans cette formule la forme de la loi de Cauchy également présente dans la loi de Deneubourg pour $\alpha = 2$. Cette formule est expliquée en détail en section 2.1.3

– Chemin de retour mémorisé et mise à jour de la phéromone.
  La fourmi garde en mémoire la liste des noeuds visités pour atteindre son noeud de destination. Au retour, elle dépose de la phéromone sur le chemin mémorisé à l'aller, en ayant auparavant éliminé toutes les boucles.
– Dépôt de phéromone dépendant de la qualité de la solution trouvée.
  De façon cohérente avec certaines observations biologiques sur des espèces de fourmis, on fait varier la quantité de phéromone déposée selon la qualité de la solution, à savoir la distance totale dans le cas du TSP.
– Évaporation de la phéromone.
  Comme dans la nature, on fait décroître la quantité de phéromone par évaporation. Ceci dit, ce n'est pas toujours un facteur prépondérant dans la nature, comme on le constate avec l'expérience du double pont où l'évaporation ne joue pas un rôle majeur.

Une des premières versions de cette famille d'algorithmes, l'algorithme "Ant System", est complètement décrite en section 2.1.3.

1.3.4 Vers l'optimisation

Les algorithmes présentés dans ce chapitre sont emblématiques de l'intelligence en essaim. Ils sont issus d'une inspiration biologique, soit par des modèles biologiques ou statistiques dans l'exemple des fourmis, soit par la phénoménologie décrite par les biologistes dans l'exemple du

flocking. Ils produisent des simulations convaincantes mais n’ont pas d’application à ce stade en ingénierie informatique.

Comme la définition de Merkle et Blum dans [Blum et Merkle, 2008] le précise, l’intelligence en essaim a donné lieu à des métaheuristique pour des problèmes d’optimisation. Les deux axes de développement que nous avons choisi pour traiter de l’intelligence en essaim, ont chacun leur propre version de métaheuristique. Le modèle d’optimisation par essaim particulaire de Kennedy et Heberhart [Kennedy et Eberhart, 1995] tout d’abord fait partie de ces algorithmes de l’intelligence en essaim dédié à l’optimisation. Il dérive de modèles particulaires pour le flocking et est assez proche du modèle de Vicsek dans sa formalisation. Dans le cas des algorithmes fourmis ensuite, le lien avec la source d’inspiration biologique est plus évident encore et le mécanisme par dépôt de phéromone s’est popularisé en informatique des réseaux et de l’optimisation combinatoire. Ces applications à l’optimisation font l’objet du chapitre suivant.

2

Champs d'application de l'intelligence en essaim

2.1 Résolution de problèmes d'optimisation

2.1.1 Métaheuristiques et optimisation

Avant d'aborder deux métaheuristiques fondamentales de l'intelligence en essaim pour l'optimisation, à savoir les approches à base d'algorithmes fourmis et à base d'algorithmes par essaims particulaires, rappelons brièvement en quoi consiste l'optimisation et quels types de problèmes l'utilisation de métaheuristiques est judicieuse. Nous nous appuyons ici sur une vision du domaine décrite par Siarry et al. dans [Dréo *et al.*, 2003].

Domaine de l'optimisation. De nombreux problèmes en sciences de l'ingénieur peuvent se ramener à la recherche de l'extremum (minimum ou maximum) d'une fonction de coût ou d'une fonction objectif f : on peut évoquer par exemple les secteurs de la recherche opérationnelle, du traitement d'images, de l'électronique, ou de l'Intelligence Artificielle entre autres. L'optimisation consiste à déterminer l'ensemble des paramètres optimaux du problème qui correspondent à l'extremum de la fonction f. Autrement dit, il s'agit de trouver les valeurs des variables réelles de $f(x)$, de sorte qu'elle soit la plus petite (minimisation des coûts) ou la plus grande possible (maximisation des profits) selon les problèmes. Plus le nombre des variables est important, plus cette recherche est rendue difficile. Des contraintes supplémentaires sous forme d'égalités ou d'inégalités peuvent être ajoutées et restreindre la recherche à un certain sous-espace de l'espace total.

On distingue l'optimisation linéaire mettant en jeu des fonctions f variant proportionnellement à leurs variables, pour laquelle de nombreuses méthodes de résolution exactes existent depuis les années 50. En revanche, l'optimisation dite non linéaire pose des problèmes d'analyse pointus et reste actuellement l'objet de recherches poussées.

Parmi la gamme des problèmes d'optimisation, on distingue les problèmes d'optimisation discrets, des problèmes à variables continues comme les problèmes d'identification qui consistent

à minimiser une fonction erreur entre le modèle d'un système et des valeurs expérimentales. Dans ces deux ensembles, on trouve des problèmes dits d'"optimisation difficile" :

- pour les problèmes discrets, dans la classe des problèmes NP-complets, i.e. non tractables par algorithme déterministe en temps polynomial (comme le voyageur de commerce - TSP présenté en section 1.3.4)
- pour les problèmes continus, ceux dont on ne connaît pas d'algorithme permettant de trouver de façon sûre une solution optimale en nombre fini d'opérations (modèles météos par exemple).

Pour les problèmes discrets, de nombreuses heuristiques ont été développées pour réduire les temps de calcul et approcher au mieux l'optimalité : une heuristique procède d'une méthode algorithmique d'approximation qui permet de fournir au plus en temps polynomial une solution pas nécessairement optimale au problème posé. De plus les heuristiques sont spécifiques à un type de problème particulier. Pour les problèmes à variables continues, des méthodes d'"optimisation globale" existent mais elles deviennent inefficaces dès que certaines propriétés de la fonction objectif ne sont plus vérifiées (par exemple si la fonction objectif n'est pas convexe [Berthiau et Siarry, 2001]).

Recours à des métaheuristiques. Dans ces cas d'optimisation difficile, des approches nouvelles apparaissent avec les métaheuristiques à partir des années 80, qui peuvent s'adapter à l'un et l'autre type de problèmes. Elles partent de principes plus génériques que les heuristiques et sont susceptibles de s'appliquer à un cadre plus large de problèmes. On peut les caractériser par les points suivants [Berthiau et Siarry, 2001] :

- elles possèdent au moins une composante stochastique pour assurer l'exploration de l'espace de recherche,
- elles ne nécessitent que très peu d'information sur la fonction objectif et procèdent dans les problèmes continus par calcul direct des valeurs de la fonction en un point de l'espace de recherche, sans recours au calcul du gradient qui nécessite la connaissance des valeurs de la fonction au moins sur plusieurs points (nécessaire dans toute méthode de type "descente de gradient"),
- leur principe de fonctionnement s'inspire de divers domaines scientifiques : physique (ex : le recuit simulé), biologie (ex : les algorithmes évolutionnaires dont les algorithmes génétiques), éthologie (les algorithmes par colonie de fourmis, les essaims particulaires),
- elles ont l'inconvénient d'un temps de calcul élevé pour obtenir des solutions de qualité,
- bien que plus génériques que des approches classiques, elles demandent une étude approfondie du réglage de leurs paramètres.

Ces métaheuristiques permettent également d'aborder un cadre étendu des problèmes d'optimisation tel que :

- l'optimisation multi-objectif qui met en jeu plusieurs objectifs éventuellement contradictoires
- l'optimisation multimodale où le but est de considérer et détecter également les optimums locaux
- l'optimisation dynamique pour laquelle la fonction objectif dépend du temps
- la possibilité de paralléliser l'exécution lors du traitement d'un problème

Le principe important pour les métaheuristiques et qui fait échouer les approches classiques est d'éviter le piégeage dans des optimums locaux de la fonction objectif. Pour y parvenir, deux stratégies principales sont mises en oeuvre :

- autoriser des mouvements/directions de recherche contre-performants dans les limites d'un certain voisinage (ce qui permet de contrôler cette tendance). Ce principe est mis en oeuvre dans toutes les métaheuristiques "de voisinage", par exemple dans le recuit simulé ou la méthode tabou. Dans ce cas, une seule solution est traitée à la fois.
- distribuer la recherche en travaillant sur une population de solutions candidates simultanément. Ce principe est appliqué dans les algorithmes génétiques, les algorithmes de colonies de fourmis ou d'essaims particulaires.

Pour finir, la difficulté réside sur la question du choix et du réglage d'une métaheuristique pour un problème donné. Il semble qu'il n'y ait pas de consensus sur ce point et certaines approches consistent à hybrider les méthodes entre elles de façon à tirer parti des avantages de chaque technique. L'objectif reste dans tous les cas d'obtenir la meilleure solution possible à coût computationnel réduit.

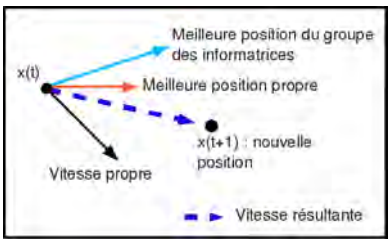
Les sections suivantes décrivent deux métaheuristiques importantes de l'intelligence en essaim que sont les algorithmes fourmis et les essaims particulaires. L'une et l'autre sont inspirées directement des phénomènes biologiques : sélection du plus court chemin vers une source de nourriture par les fourmis dans le premier cas, phénomènes de flocking et de recrutement chez les abeilles dans le second.

2.1.2 Optimisation par essaim particulaire

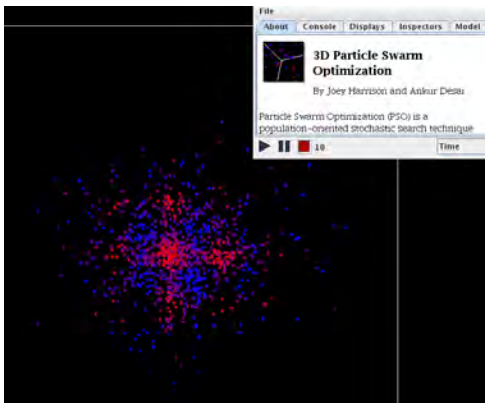
Origine. L'origine de la métaheuristique des essaims particulaires provient de l'inspiration des comportements de flocking qu'étudiaient Kennedy et Eberhardt (puis Yuhui Shi) aux Etats-Unis en 1995. De l'aveu même des auteurs [Kennedy *et al.*, 2001] c'est en programmant des simulations de flocking qu'ils ont découvert le potentiel que possédaient ces algorithmes pour l'optimisation. Cette "découverte" a mené à la conception d'une version spécifique à l'optimisation baptisée "Particle Swarm Optimization"/optimisation par essaim particulaire (PSO/OEP). Bien que l'algorithme soit issu de considérations sur le flocking à l'origine, la métaphore biologique la plus appropriée pour l'expliquer est celle des essaims d'abeilles : une abeille ayant découvert un site intéressant de nourriture va en communiquer l'emplacement à un certain nombre de ses congénères qui tenteront de retrouver cette source. L'information n'étant qu'approximative, une autre abeille cherchant à atteindre le site, n'aboutira peut-être qu'à une zone voisine de celle indiquée. Ce faisant, cette abeille aura pu découvrir en chemin ou sur la zone estimée, un meilleur site dont elle ramènera l'information à son essaim... On l'aura compris, cet algorithme bio-inspiré va tirer parti des meilleures performances globales sur l'essaim ainsi que des meilleures performances individuelles.

Application. Le problème type sur lequel les systèmes d'essaim particulaires sont particulièrement efficaces est la recherche du minimum d'une fonction f non linéaire continue dans un espace à D dimensions [Clerc, 2005]. Cette fonction est la fonction objectif du problème qu'il faut optimiser.

Principe. La performance d'une particule correspond à la valeur calculée de cette fonction à la position courante de cette particule. L'espace de recherche est le domaine de définition de la fonction objectif, la particule navigue en fait dans le champ scalaire associé à la fonction f sur son espace de définition. Voici le principe général de cet algorithme [Kennedy et Eberhart, 1995] :



(a) Principe de calcul de la nouvelle position d'une particule.



(b) Exemple d'exécution d'un algorithme PSO dans un espace de recherche en 3D [De Jong, 1975]. Le programme utilisé (écrit par Joey Harrison et Ankur Desai) provient de la plate-forme de simulation MASON [Luke *et al.*, 2003].

FIG. 2.1 – Principe et image de simulation 3D de l'algorithme par essaims particulaires.

- Chaque entité constituant l'essaim est une particule dont l'état est caractérisé par deux grandeurs classiques dans un espace de dimension D : un vecteur position $\mathbf{x}$ et un vecteur vitesse $\mathbf{v}$.
- Ces particules sont dotées d'une mémoire qui leur permet de stocker leur position courante et leur position passée ayant la meilleure performance (c'est-à-dire la plus proche de l'extremum considéré pour la fonction objectif).
- Au départ, l'essaim est distribué aléatoirement dans l'espace de recherche avec une vitesse finie aléatoire.
- Le principe mis en oeuvre pour l'optimisation repose sur le couplage entre les particules, exprimant ainsi mathématiquement l'interaction et l'aspect social du processus. La nouvelle vitesse d'une particule $\mathbf{v}(t+1)$ est calculée en prenant en compte les positions associées à sa meilleure performance passée $\mathbf{p}(t)$ (aspect temporel de l'optimisation), et la meilleure performance sur un sous-groupe de particules de l'essaim au même instant $\mathbf{g}(t)$ (aspect "spatial" de l'optimisation). De ces deux informations pondérées, on tire la nouvelle vitesse de la particule par couplage avec sa vitesse propre courante (voir la figure 2.1(a)) selon le système dynamique :

$$\begin{cases} \mathbf{v}_d(t+1) & = c_1\mathbf{v}_d(t) + c_2(\mathbf{p}_d(t) - \mathbf{x}_d(t)) + c_3(\mathbf{g}_d(t) - \mathbf{x}_d(t)) \\ \mathbf{x}_d(t+1) & = \mathbf{x}_d(t) + \mathbf{v}_d(t+1) \end{cases} \tag{2.1}$$

où d indique l'indice des coordonnées et (c_2, c_3) sont deux variables aléatoires de loi uniforme sur $[0, c_{max}]$. Le sous-groupe de particules qui seront "consultées" est appelé groupe

des informatrices ; il peut-être sélectionné par simple voisinage, ou par tirage au sort dans l'essaim.
- Lorsque la meilleure informatrice est détectée, la nouvelle vitesse d'une particule est donc la résultante des trois tendances suivantes [Clerc et Siarry, 2004] :
  - la tendance "aventureuse" correspond à continuer selon la vitesse actuelle
  - la tendance "conservatrice" ramène vers la position passée la meilleure
  - la tendance "panurgienne" comme son nom l'indique privilégie l'information venant des autres memebres de l'essaim et dirige vers la meilleure informatrice.

  Les deux dernières tendances sont modulées par un coefficient aléatoire pour maintenir la diversité dans l'essaim et un niveau d'exploration suffisant dans l'espace de recherche.

Afin de comparer les performances relatives des diverses métaheuristiques d'optimisation, un ensemble de fonctions sert comme batterie de tests. Cet ensemble correspond aux fonctions de "De Jong" qui en a composé la liste pour évaluer des algorithmes génétiques à l'origine [De Jong, 1975]. Pour cet ensemble de fonctions tests, on recherche les solutions de l'équation $f(\mathbf{x}^*) = 0$ où $\mathbf{x}^*$ est un vecteur en dimension 2 ou 3.

2.1.3 Optimisation par colonies de fourmis

Cette section décrit plus précisément la première version efficiente des algorithmes fourmis pour l'optimisation, à savoir l'algorithme du système fourmis[17]. Les autres variantes ne sont pas explicitées ici, mais nous renvoyons le lecteur à l'ouvrage très complet de Dorigo et Stützle [Dorigo et Stützle, 2004a]. Nous employons parfois le terme d'agent en lieu et place de "fourmi artificielle" même si cette dénomination ne fut pas systématiquement utilisée dans les travaux originaux.

Nous nous contentons ici de poser les bases de l'algorithme "Ant System" dans sa version originale qui servira par la suite de point de comparaison pour nos propres simulations avec le SMA logistique.

"Ant system" - Le système fourmis.

Cette version des algorithmes fourmis pour l'optimisation créée par Dorigo [Dorigo *et al.*, 1996] propose une version améliorée de son tout premier algorithme bio-inspiré. C'est la première version qui rivalise véritablement avec les autres métaheuristiques.

Description de l'algorithme appliqué au TSP. Rappelons la formulation du problème du voyageur de commerce déjà présenté en section 1.3.4 : le problème se compose d'un graphe $G = (N, A)$ où N est l'ensemble des noeuds et A celui des arcs dont les poids correspondent aux distances d_{ij} entre noeuds. On affecte également à chaque arc une quantité de phéromone τ_{ij}. L'objectif est de trouver un cycle hamiltonien de distance totale la plus courte (il peut éventuellement y en avoir plusieurs). Rappelons la difficulté de ce type de problèmes : il est NP-complet et les instances de graphes des librairies de tests n'ont pas toujours une solution optimale connue.

[17] "Ant system"

Principe. La version du système fourmis introduit une heuristique supplémentaire liée aux distances relatives entre villes η_{ij}, d'autre part elle élimine les boucles des chemins parcourus par la mémorisation d'une liste tabou $\mathcal{T}_i^k$ des villes déjà visitées par l'agent. Ainsi les éléments de base de l'algorithme deviennent les suivants [Dorigo *et al.*, 1996] :

- la fonction de décision de l'agent sur un noeud i pour choisir un noeud j adjacent est donnée par la probabilité :

$$p_{ij}^k = \begin{cases} \dfrac{\tau_{ij}^\alpha \ \eta_{ij}^\beta}{\sum_{l \in \mathcal{T}_i^k} \ \tau_{il}^\alpha \ \eta_{il}^\beta} & \text{, si } j \in \mathcal{T}_i^k \\ 0 & \text{, si } j \notin \mathcal{T}_i^k \end{cases} \tag{2.2}$$

  où $\mathcal{T}_i^k$ représente l'ensemble des noeuds restant à parcourir à partir du noeud i par la fourmi k (liste taboue) et $\eta_{ij} = \frac{1}{d_{ij}}$ l'heuristique employée pour minimiser la distance. α et β sont deux paramètres qui règlent l'influence relative du taux de phéromone par rapport à la distance parcourue. Une explication sur le réglage des jeux de paramètres est donnée en section suivante.
- La quantité de phéromone déposée à la fin d'un tour sur les arêtes du graphe suit une loi d'évaporation et un ajout de phéromone résumé dans la formule suivante :

$$\tau_{ij} \leftarrow (1-\rho)\tau_{ij} + \triangle\tau_{ij}^k \text{ , si } (i,j) \in A_k \tag{2.3}$$

où A_k est l'ensemble des arcs empruntés par la fourmi k. Si L^k est la distance totale parcourue par l'agent k sur un tour, la quantité ajoutée :

$$\triangle\tau^k = \begin{cases} Q, \text{ ou bien} \\ \frac{Q}{L^k} \end{cases} \tag{2.4}$$

Paramètres et variantes. Deux variantes existent également sur cette base qui diffèrent par la mise à jour immédiate des phéromones sur chacun des arcs visités :

- Une variante du modèle par "densité de fourmis" où les fourmis déposent une quantité constante de phéromone par pas de temps sur les arcs visités.
- Une variante par "quantité de fourmis" où les fourmis déposent cette même quantité divisée par le poids (la distance ici) de l'arc visité à chaque pas de temps.
- Ces deux variantes procèdent par dépôt asynchrone de phéromone lorsque la fourmi passe d'un noeud à l'autre. On peut résumer l'ensemble par les formules de calcul suivantes :

$$\triangle\tau_{ij}^k = \begin{cases} Q & \text{, dans la variante densité de fourmis} \\ \dfrac{Q}{d_{ij}} & \text{, dans la variante quantité de fourmis} \\ 0 & \text{, si } (i,j) \notin A_k \end{cases} \tag{2.5}$$

  où d_{ij} est la distance de l'arc (i,j).

On peut résumer le sens des paramètres utilisés :

- α donne l'importance relative de la piste de phéromone
- β donne l'importance relative de la visibilité sur le chemin
- ρ définit la persistance de la piste de phéromone
- Q donne une valeur unitaire de phéromone déposée.

	Meilleur tour	Moyenne	Déviation standard
Système fourmi	420	420,4	1,3
Recherche Tabou	420	420,6	1,5
Recuit simulé	422	459,8	25,1

TAB. 2.1 – Exemple de performance relative du système fourmi avec d'autres approches pour 10 simulations sur Oliver30 en nombres entiers

Performance des algorithmes fourmis pour l'optimisation.

Pour mesurer les performances de ces algorithmes sur un problème TSP, une librairie spécifique TSPLIB existe [TSP, 1995]. Elle comporte de nombreux problèmes tests dont on connaît les solutions optimales, permettant ainsi des comparaisons.

Une revue des performances relatives de cet algorithme par rapport aux autres algorithmes de la famille ACO est faite dans l'article de synthèse [Stützle et Dorigo, 1999]. Une comparaison entre l'algorithme présenté "Ant System" avec d'autres métaheuristiques dont le recuit simulé[18] et la recherche Tabou[19] est rapportée dans [Dorigo *et al.*, 1996]. Les résultats de simulation montrent une forte sensibilité vis-à-vis du nombre de fourmis engagées sur le problème. Ainsi, il existe pour chaque problème un nombre optimal de fourmis pour lequel l'algorithme a le plus de chances de converger vers la solution optimale. Cet aspect se généralise aux autres paramètres et il faut de nombreux essais pour trouver le bon jeu de paramètres. Ainsi le paramètre α utilisé dans la formule de décision 2.2 doit-il être soigneusement fixé : une valeur trop faible comme une valeur trop fortes conduisent à de mauvaises solutions et à de la stagnation. Le jeu de valeurs $\alpha = 1$ ou 2 et $\beta = 2$ donne généralement de bons résultats dans la plupart des situations. Aucune métaheuristique n'échappe hélas à cet inconvénient. Il s'agit de trouver le modèle comportant le moins de paramètres possibles.

Le tableau 2.1 repris de [Dorigo *et al.*, 1996] restitue les performances moyennes de ces trois algorithmes sur le problème Oliver30 (30 villes dont le cycle optimal a pour distance exacte $423,741$; pour le problème traité avec des nombres entiers, cette distance est ramenée à 420) de la TSPLIB. Ce problème est un petit problème TSP et ne permet pas vraiment une comparaison pertinente, mais cela suffit à montrer que l'algorithme fourmi rivalise avec les autres métaheuristiques standards. De façon générale, l'algorithme trouve rapidement de bonnes solutions et possède un jeu idéal de paramètres qui varie peu avec la taille du problème. On trouvera les performances comparées des versions du même type d'algorithmes fourmis sur de plus grands problèmes TSP dans [Dorigo et Stützle, 2004b].

Discussion sur la famille des algorithmes fourmis.

Cette famille d'algorithme a montré sa performance sur de nombreux problèmes d'optimisation : l'extension de l'algorithme à d'autres types de problèmes d'optimisation comme les

[18] "Simulated Annealing"
[19] "Tabu Search"

problèmes d'affectation quadratique [20] et ses variantes, ou encore les problèmes d'ordonnancement [21], est proposée dans [Dorigo *et al.*, 1996]. Cependant, cette efficacité questionne encore les scientifiques et de nombreux articles sont publiés pour tenter d'en comprendre les raisons essentielles. Deux questions se trouvent posées :

- premièrement, peut-on montrer que l'algorithme converge presque sûrement vers la solution optimale du problème d'optimisation,
- deuxièmement quelle théorie peut expliquer/fonder cette famille d'algorithme.

A la première question, une des premières réponses est apportée par Guthjahr dans [Gutjahr, 2000, Stützle et Dorigo, 2002] puis dans [Gutjahr, 2002] : ces articles démontrent que l'algorithme fourmi trouve la solution optimale du TSP, sans garantie d'un temps fini pour la trouver. A la seconde, une des voix d'analyse et de fondement théorique de l'algorithme trouve sa source dans le cadre plus général de la théorie de l'apprentissage par renforcement [Birattari *et al.*, 2002]. A ce propos, la version Ant-Q de l'algorithme fourmi exploite les principes de l'algorithme du "Q-learning" [Gambardella et Dorigo, 1995].

Mais cette référence à l'apprentissage par renforcement n'explique pas tout, notamment sur l'aspect dynamique de l'algorithme et des mécanismes intrinsèques mis en oeuvre. Malgré des analyses dynamiques menées pour montrer comment la phéromone influe sur le comportement dynamique des agents [Parunak, 1997], cette question reste ouverte...

2.2 Robotique en essaim

La robotique fait partie des applications visées par l'intelligence en essaim ; cette déclinaison de la robotique porte le nom de robotique en essaim. La robotique en essaim ajoute les contraintes du monde physique à celle de la théorisation des phénomènes en essaim. Sous un autre angle de vue, elle propose un terrain d'expérimentation et de validation stimulant pour les algorithmes de l'intelligence en essaim. C'est dans cet esprit que nous avons abordé la robotique en essaim dans cette thèse, comme secteur de validation de nos algorithmes. Nous ne faisons que synthétiser ici les enjeux et grands axes de développement de ce domaine.

Définition et caractéristiques de la robotique en essaim. La robotique en essaim cherche en effet à se démarquer de la robotique distribuée ou collective par sa bio-inspiration explicite. Une définition du domaine est donnée dans [Dorigo et Sahin, 2004, Sahin, 2005] :

Définition 2.2.1. *La robotique en essaim est l'étude de la façon dont un grand nombre d'agents relativement simples incarnés physiquement peuvent être conçus pour qu'un comportement voulu puisse émerger des interactions locales entre agents et entre les agents et l'environnement*[22].

La robotique en essaim considère des robots en général identiques, de conception simple qui ont des perceptions et interactions uniquement locales. Ces essaims de robots sont

[20] quadratic assignment problem (QAP)

[21] Job-shop scheduling problem (JSP)

[22] "Swarm robotics is the study of how a large number of relatively simple physically embodied agents can be designed such that a desired collective behavior emerges from the local interactions among the agents and between the agents and the environment."

susceptibles d'offrir les caractéristiques suivantes qui rejoignent celles de la robotique collective [Sahin *et al.*, 2008] :

- La robustesse : la redondance inhérente au système, la coordination décentralisée, la simplicité de conception de chaque entité et la distribution de la perception, font des essaims de robots des systèmes robustes aux pannes et aux variations de l'environnement.
- La flexibilité concerne la capacité des essaims de robots à traiter différents types de problèmes.
- Scalabilité : les performances ne sont globalement pas impactées par la taille de l'essaim.

Axes et problèmes de recherche. L'incarnation des agents dans un environnement physique augmente la difficulté par rapport à l'intelligence en essaim logicielle dont les algorithmes ne sont en général pas implémentés directement dans les robots.

Axes de recherches. Ainsi quelques axes principaux de recherche ressortent [Sahin *et al.*, 2008] par rapport à cette difficulté :

- L'axe conception : il doit répondre au besoin de construire les comportements des robots afin de pouvoir obtenir un comportement global émergeant des interactions locales qui corresponde à ce qui est attendu. Deux approches sont à l'oeuvre :
  - une approche empirique où chaque comportement visé pour l'essaim implique une reconception spécifique des agents-robots,
  - une approche fondée sur des principes génériques de construction des comportements, applicables quelle que soit la configuration posée. Il s'agit par exemple d'exploiter les principes de l'évolution artificielle pour l'apprentissage des comportements des robots dans un environnement artificiel de simulation, avant de plonger ces robots dans le monde physique avec les comportements appris dans le monde virtuel [Dorigo *et al.*, 2005].
- L'axe modélisation et analyse : il est rendu indispensable pour avoir le maximum de garanties et de prédictions quant au fonctionnement futur de l'essaim de robots dans le monde réel. La modélisation revient à concevoir des plate-formes virtuelles de simulation aussi proches de la réalité que possible, et d'y faire tous les tests et analyses avant l'expérimentation réelle.

Types de problèmes. Les types de problèmes à résoudre dans la robotique en essaim sont pour beaucoup issues des phénoménologies identifiées dans l'auto-organisation biologique :

- Agrégation et dispersion sont les deux faces d'une même pièce : l'essaim doit pouvoir explorer son environnement (dispersion) et également se regrouper pour exploiter (agrégation).
- Fourragement : inspiré des comportements fourmis, l'essaim doit être capable d'optimiser sa recherche de ressources.
- Auto-assemblage et mouvement coordonné : l'essaim doit pouvoir se souder pour constituer une structure répondant à une contrainte du terrain (ex : formation de pont chez les fourmis, tractage d'une proie, ...). Dans cette configuration où les robots sont reliés physiquement, l'objectif est de pouvoir continuer à se mouvoir collectivement.

- Transport coopératif et collectif : la métaphore ici est celle de fourmis transportant une proie à plusieurs individus.
- Génération de formes et auto-organisation : cela correspond à la capacité de l'essaim à produire des schémas de regroupement spécifiques ou singuliers par processus auto-organisé.

Cette courte revue peut être complétée par les articles [Sahin *et al.*, 2008, Beni, 2005, Dorigo *et al.*, 2005].

3

Quelques constats sur l'intelligence en essaim

Retour sur la conception des modèles existants. Quelques failles conceptuelles dans chacune des deux approches de l'intelligence en essaim considérées ici, ont été relevées en section 1.3 :

- Les modèles de flocking sont encore très proches des modèles de particules (modèles de Reynolds et de Vicsek), et n'ont pas vraiment besoin de la notion d'agent. Le recours à des champs de forces est dans la même lignée du modèle de particule et nous émettons de fortes réserves concernant la capacité de ce type de modèles à englober une large gamme de mécanismes impliqués dans l'intelligence en essaim. Ils présentent néanmoins une simplicité de conception qui en font des métaheuristiques puissantes.
- Les modèles fourmis sont fondées sur une loi de probabilité efficace et empirique mais difficile à appréhender mathématiquement (propriétés particulières de la loi de Cauchy). De fait un certain nombre d'analyses en termes de dynamique et de stabilité sur ce type d'algorithme sont impossibles dans leur formulation d'origine. Il est nécessaire d'avoir recours à une théorisation plus large comme l'apprentissage par renforcement pour en expliquer le fonctionnement.

Ces constats nous amènent à envisager d'autres pistes de modélisation afin de lever ces inconvénients.

Notre analyse des mécanismes. Notre analyse sur les modèles présentés dans ce chapitre nous conduit à distinguer deux types de mécanismes fondamentaux :

- dans le cas des modèles de flocking bio-inspirés ou théoriques, le mécanisme est selon nous principalement un mécanisme de couplage exprimant des interactions directes, couplage fort comme on l'a vu dans le modèle de Vicsek,
- dans les algorithmes fourmis en revanche, c'est principalement le champ de phéromone qui contrôle l'individu par le biais d'interactions indirectes.

Il y a donc selon notre analyse, deux origines principales aux mécanismes de ces algorithmes : un mécanisme de couplage et un mécanisme de contrôle décentralisé mais interne aux entités considérées qu'il faut tenter de mieux expliciter dans les modèles.

Limitations théoriques. La différence de modélisation des mécanismes identifiés ci-dessus ne contribue pas à la conception d'une vision unifiée du domaine de l'intelligence en essaim, autant qu'à la compréhension profonde des mécanismes d'auto-organisation qu'elle implique. Le cas des métaheuristiques pour l'optimisation est à ce titre révélateur : la multiplication des paramètres nécessaires au traitement de nouveaux problèmes réduit la visibilité du concepteur sur les choix de modélisation et les directions à privilégier pour améliorer les performances. La stratégie de l'essai-erreur n'est plus tenable passé un certain stade de développement. Les approches hybrides se multiplient, ce qui augmente la complexité de manipulation des modèles. Un renouvellement des approches de modélisation, ou le repositionnement des approches existantes dans des modèles de plus haut niveau d'abstraction, sont des voies pour débloquer ce type de situation.

Sur un plan de conception purement théorique maintenant, les deux mécanismes de couplage et de contrôle que nous identifions comme essentiels pour expliquer l'auto-organisation mise en jeu, ne sont pas réellement modélisés dans les approches présentées, exception faite de la dynamique des essaims particulaires qui procède par couplages de plusieurs variables d'états des particules. Par exemple, de nombreux auteurs s'accordent à considérer le flocking comme un phénomène de synchronisation. Reynolds le premier en affirmant par exemple dans [Reynolds, 1987] :

> Une nuée montre des aspects contrastés : elle est faite d'oiseaux individuels et pourtant le mouvement global est fluide ; son concept est simple et pourtant la nuée paraît visuellement si complexe, elle semble distribuée aléatoirement et pourtant elle est magnifiquement synchronisée[23].

Nous adhérons entièrement à cette description du flocking, mais la notion de synchronisation n'apparaît pas dans le modèle de Reynolds, et il n'est pas quantifié. Cet aspect n'est à notre connaissance pas utilisé non plus dans les essaims particulaires.

De la même façon, le contrôle subit par la fourmi via le champ de phéromones n'est pas explicite dans le modèle fourmi actuel, même si la grandeur phéromone apparaît dans le calcul de la fonction de décision associée à la fourmi. Pour autant, la formulation mathématique de cette décision n'exprime pas clairement qu'il s'agit d'un contrôle indirect, comme le serait précisément un paramètre de contrôle pour un système dynamique.

Retour aux sources. Nous avons fait le choix d'une remise à plat des approches de modélisation dans cette thèse, avec l'objectif de donner un sens à des aspects comme le contrôle interne décentralisé ou la notion de synchronisation collective par couplage, que nous avons identifiés dans les deux études de cas de ce chapitre sur l'intelligence en essaim. Le but de cette démarche est également de tendre vers des modèles plus génériques, comportant si possible un minimum de paramètres.

Pour atteindre cet objectif, nous estimons qu'il faut revenir "aux sources" des mécanismes et des phénomènes, à savoir ceux liés à l'auto-organisation en général, et à l'auto-organisation biologique en particulier. Mais revenir aux sources signifie aussi revenir aux théories et principes de modélisation développés spécifiquement autour des questions sur l'auto-organisation. C'est le

[23] A flock exhibits many contrasts. It is made up of discrete birds yet overall motion seems fluid ; it is simple in concept yet is so visually complex, it seems randomly arrayed and yet is magnificently synchronized.

coeur des deux chapitres suivants, le premier s'attachant à faire un retour sur le concept d'auto-organisation dans les systèmes, le deuxième à présenter la réflexion scientifique dans le champ des systèmes complexes.

4

Auto-organisation : concept fondateur de l'intelligence en essaim.

Après cette présentation de l'intelligence en essaim, réduite à deux types d'application algorithmique —algorithme de "flocking" et algorithmes "fourmis"—, et après avoir mis en lumière les problématiques associées et certaines limitations des approches existantes, nous cherchons à présent à expliciter la manière dont nous pouvons aborder l'intelligence en essaim dans le contexte plus large des systèmes complexes, qui nous a conduit à utiliser et exploiter un modèle de système complexe existant. Notre programme de recherche est en effet de reprendre d'une certaine façon la modélisation des phénoménologies liée à l'intelligence en essaim sur des bases plus génériques, il est nécessaire de remonter à l'origine des concepts et des démarches scientifiques pour engager une voie nouvelle de modélisation. C'est donc à la fois autour du concept d'auto-organisation que ce chapitre s'articule et des approches développées dans les systèmes complexes.

La figure 4.1 présente un panorama structuré sous forme de graphe des différents courants scientifiques qui ont rapport d'une manière ou d'une autre à la question de la complexité des systèmes et en particulier à la question de l'auto-organisation qui constitue le point de départ des sciences de la complexité.

Un commentaire sur ce diagramme avant d'aller plus loin : il est repris d'un document existant qui peut sembler réducteur notamment en ce qui concerne les différents protagonistes mentionnés. Nous ne validons pas l'ensemble du schémas, certains liens étant à notre avis abusifs : par exemple, placer les automates cellulaires dans les approches à base d'agents et non dans les sciences de la complexité nous semble être un contresens.

Ceci dit, nous retenons essentiellement du schéma la structuration des grandes branches de concepts relatives aux plus gros noeuds du graphe. Les divers domaines qui concernent nos travaux y sont représentés : les systèmes dynamiques non linéaires, les systèmes complexes, les systèmes multi-agents. L'intelligence en essaim dans sa version informatique, n'y figure pas en tant que tel, mais serait à intégrer, comme nous l'entendons dans notre définition du domaine, quelque part entre les systèmes multi-agents et les sciences de la complexité. Ainsi l'intelligence en essaim serait un noeud dans ce graphe reliant deux branches différentes. Nous montrons

dans ce chapitre qu'une troisième branche de développement nous semble essentielle à relier à ce noeud : la branche des systèmes dynamiques non linéaires, en particulier le "noeud" du chaos déterministe.

Historiquement tout démarre avec la cybernétique et la systémique. Nous expliquons dans ce chapitre d'où provient le concept d'auto-organisation au coeur de l'intelligence en essaim, quelle place il tient dans de nombreux domaines scientifiques aujourd'hui, et pourquoi il défie les scientifiques qui tentent d'en théoriser les mécanismes. Pour mieux comprendre ce phénomène dans les différents champs où il apparaît et pour en quantifier la manifestation dans les systèmes, la science dite "des systèmes complexes" propose des modèles et méthodologies d'analyse qui visent à pallier le manque de théorisation sur l'auto-organisation. C'est pourquoi, nous nous intéressons à un modèle typique des systèmes complexes présenté en fin de chapitre, les réseaux et gaz d'itérations couplées, qui nous semblent ouvrir la voie, par un modèle simple dans son écriture mais complexe dans sa dynamique, à de nouvelles approches pour l'intelligence en essaim.

4.1 Systémique et cybernétique

Petit historique.

Nous présentons ici une courte synthèse sur la cybernétique, car ce courant de pensée scientifique né à l'issue de la seconde guerre mondiale sur l'initiative du mathématicien Norbert Wiener, pose les bases de la réflexion sur le concept d'auto-organisation dans les systèmes. Wiener proposa une définition et un programme opératoire de la cybernétique[24] dans un ouvrage de 1948 "Cybernetics or Control and Communication in the Animal and the Machine" [Wiener, 1948]. Son ouvrage est emprunt de physique mathématique (mécanique statistique, séries temporelles), de théorie de l'information de Shannon [Shannon, 1948] et de théorie du contrôle naissantes :

> Nous avons décidé de désigner le domaine entier de la théorie de la communication et du contrôle, concernant la machine ou l'animal, par le terme de cybernétique.[25].

La cybernétique insiste sur l'importance de la communication et du contrôle de l'information pour comprendre les niveaux d'organisation les plus complexes. L'idée des cybernéticiens portés par les progrès technologiques de l'après-guerre a été dès le début de réfléchir aux liens et applications possibles de la technologie des machines digitales aux mécanismes du vivant.

La cybernétique rejoint d'ailleurs assez vite dans les années 50 la "théorie générale des systèmes" fondée par le biologiste Ludwig von Bertalanffy [von Bertalanffy, 1968] qui tente de rompre avec la méthodologie réductionniste classique, et a pour ambition d'aborder et de traiter les problématiques larges des systèmes ouverts évolutifs en interaction avec leur environnement. Là où la cybernétique vise plutôt des systèmes dirigés par un but à base de fonctions, la théorie des systèmes considère les relations entre les constituants d'un système plutôt que les propriétés des constituants eux-mêmes, amenant ainsi une vision holistique et émergente de l'ensemble. Elle a recours à des formalismes de type système dynamique pour modéliser les couplages en-

[24] Le terme grec "kubernetes" figure chez Platon pour désigner l'art de conduire un navire.

[25] "We have decided to call the entire field of control and communication theory, wether in the machine or in the animal by the name Cybernetics"

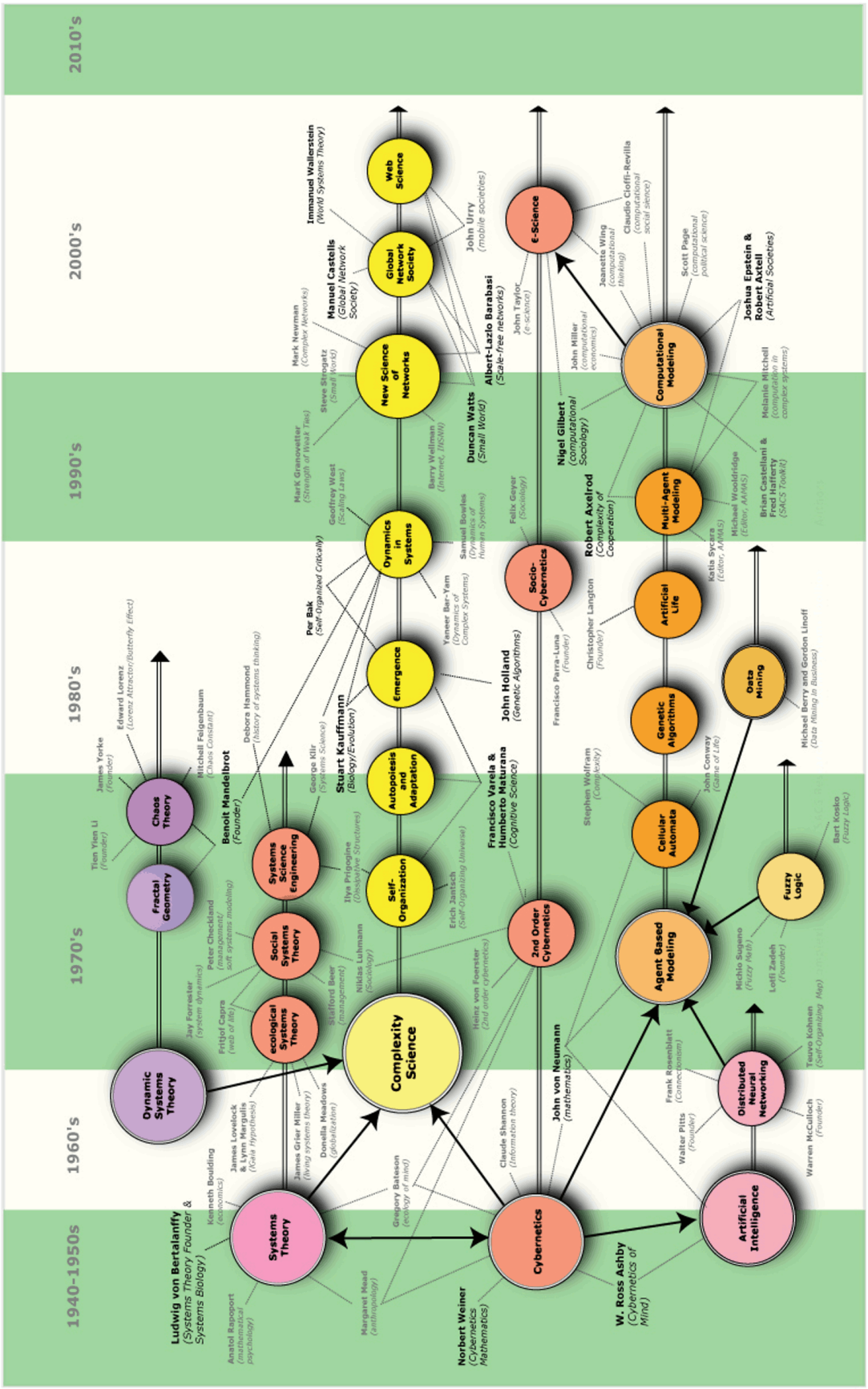


FIG. 4.1 – Carte globale sur le positionnement et l'évolution des sciences de la complexité. Carte réalisée par Brian Castellani dans l'ouvrage [Castellani et Hafferty, 2009]. Image sous licence Creative Commons Attribution-Share Alike 3.0, téléchargée sur le Web.

tre les différentes parties d'un système global. Ce que l'on nomme "science des systèmes" ou "systémique" recouvre en fait les deux champs théoriques des systèmes et de la cybernétique.

Enfin un second mouvement de la cybernétique, appelé "cybernétique du second ordre", lancé par Heinz von Foerster dans les années 70 [von Foerster, 1979], oriente et développe encore davantage les réflexions autour de l'autonomie, de l'auto-organisation des systèmes composites, de la cognition et du rôle de l'observateur, et laisse une part plus importante aux processus de rétroaction positive dans les systèmes [Heylighen et Joslyn, 2001].

Naissance du concept d'auto-organisation.

L'article [Heylighen, 1999] résume les principaux développements historiques autour de ce concept. Nous en relevons quelques éléments qui nous semblent essentiels à notre propos ci-dessous.

Un paradoxe thermodynamique. La réflexion de départ sur l'auto-organisation peut s'articuler autour d'un paradoxe thermodynamique : les observations des phénomènes d'auto-organisation dans la matière inerte, tels que la cristallisation, la convection dans les fuides, les oscillations dans certaines réactions chimiques,... montrent que la création d'un ordre spontané devrait s'accompagner d'une diminution de l'entropie thermodynamique du système (associée au désordre structurel), or cette conclusion est en contradiction avec le second principe de la thermodynamique qui affirme que l'entropie d'un système isolé ne peut que croître. Les travaux d'Ilya Prigogine sur les structures dissipatives dans les années 50 [Prigogine et Nicolis, 1977] établissent que l'entropie générée par ce type de système (réactions chimiques comme le "Brusselator", ou la réaction oscillante de Belousov-Zhabotinsky) qui sont des systèmes ouverts, se dissipe à l'extérieur du système (l'environnement). Ainsi, le second principe resterait globalement vérifié, pour peu que l'on redéfinisse les limites du système considéré. Ce constat reste valide dans le cas des organismes vivants : une plante ou un animal prélève dans l'environnement une matière à entropie basse (source de nourriture ordonnée) pour accroître son propre ordre interne, en compensation du rejet de déchets de matière à haute entropie (déstructuration de matière).

L'autre aspect troublant du phénomène est l'émergence spontanée du mécanisme sans aucune influence globale extérieure au système, par la simple mise en présence des constituants du système. L'auto-organisation vue sous cet angle apparaît comme une réorganisation des parties ordonnées et désordonnées du système. De plus cette phénoménologie de l'auto-organisation ne se produit, nous dit Prigogine, que pour des systèmes non linéaires d'une part, et loin de l'équilibre thermodynamique d'autre part. Nous notons une première référence à la notion de non linéarité des systèmes auto-organisants, caractéristique qui joue un rôle central dans l'approche développée dans cette thèse.

Premières conceptualisations de l'auto-organisation. La cybernétique amene une lecture plus abstraite et plus générique, en conceptualisant l'auto-organisation des systèmes. C'est Ross Ashby, psychiatre et ingénieur anglais, cybernéticien de la première heure qui définit cette notion dès 1947 [Ashby, 1947]. Sa lecture du phénomène l'amène à poser le "principe d'auto-organisation" qui stipule que tout système tend vers une forme d'équilibre (que l'on pourrait

traduire en termes modernes par attracteur) où les différentes parties du système s'adaptent mutuellement. En fait sa vision de l'auto-organisation s'articule autour de la sélection d'états particuliers du système, stables, parmi l'ensemble des états possibles, augmentant de fait la quantité d'information sur le système (accompagné d'une réduction d'entropie) [Ashby, 1962]. Selon Ashby, les systèmes qui s'auto-organisent sont ouverts en terme d'énergie mais fermés en terme d'information. Un autre cybernéticien Heinz von Foerster, physiscien et philosophe, s'est particulièrement attaché à définir ce concept, et a proposé une interprétation plus précise en terme de mécanisme à travers le principe d'"ordre issu du bruit"[26] : plus les perturbations aléatoires sur le système sont grandes plus vite il pourra atteindre un état d'équilibre. Le principe est d'explorer le plus largement possible les états accessibles pour pouvoir sélectionner les mieux adaptés (principe directement exploité par les algorithmes de type "recuit simulé" notamment). Dautre part, il prolonge la définition de l'auto-organisation de Ashby en précisant les relations du système avec son environnement : le système consomme l'énergie prélevée dans son environnement pour s'organiser.

Des concepts fondateurs.

Ce que nous retenons de la cybernétique est qu'elle a amené les cybernéticiens d'après-guerre à définir de nouveaux concepts extrêmement riches d'implications pour la science contemporaine (systèmes complexes notamment) tels que : complexité, auto-organisation, auto-reproduction, autonomie, réseau, connexionisme, adaptation. Ou encore des mécanismes fondamentaux [von Foerster, 1960, Ashby, 1957] comme : régulation, auto-référence, circularité, rétro-action positive et négative, contrôle, couplage de systèmes, génération d'"ordre à partir du bruit", principe de variété.

Parmi ces mécanismes, notre recherche se concentre à modéliser plus particulièrement le contrôle et le couplage au sein des systèmes, ainsi que la circularité des mécanismes. L'idée de générer de l'ordre à partir du bruit est également centrale dans notre approche, et nous la développons à partir de modèles de calcul empruntés aux systèmes dynamiques non linéaires déterministes.

4.2 Synergétique

La "synergétique" propose une tentative de théorisation de l'auto-organisation à base d'un principe théorique très générique, principe que nous avons repris dans nos travaux. Il constitue selon nous une première clé de compréhension des mécanismes. C'est pourquoi nous en présentons les grandes idées directrices dans cette section.

La théorie "synergétique" a été fondée par le physicien Hermann Haken à partir des années 70 [Haken, 1977]. La synergétique a pour objectif l'étude et la modélisation des phénomènes d'auto-organisation dans les systèmes complexes, un système complexe étant défini comme constitué de multiples entités / parties en interaction. Le mot synergétique se réfère au terme "synergie" qui étymologiquement signifie "travailler ensemble".

[26] "Order from noise"

L'auto-organisation synergétique

Hermann Haken définit l'auto-organisation par la formule suivante [Haken, 2008] :

Définition 4.2.1. *L'auto-organisation est la formation spontanée, souvent avec un objectif apparent, de structures spatio-temporelles, temporelles, spatiales ou de fonctions dans les systèmes constitués de peu ou de nombreux composants. En physique et biologie, l'auto-organisation apparaît dans les systèmes ouverts loin de l'équilibre thermodynamique*[27].

Les exemples suivants de phénomènes entrent dans ce cadre et sont des objets d'étude pour la synergétique : les mouvements de convection dans un fluide chauffé ou les propriétés non linéaires de la lumière laser en physique (dont Hermann Haken fut un des principaux acteurs scientifiques), les oscillations de réactifs chimiques (réaction de Belousov-Zhabotinski), les motifs observés sur la peau de certains animaux, la construction de termitières en biologie,.... La synergétique se donne d'autre part pour ambition d'étendre son analyse à d'autres champs d'investigation, notamment en ce qui concerne le fonctionnement humain comme la cognition [Haken, 2004] et la psychologie [Tschacher et Dauwalder, 1999].

Le principe d'asservissement par le champ.

Le principe fondamental de la synergétique qui nous intéresse plus particulièrement ici, repose sur les mécanismes observés dans les systèmes physiques. Il postule que l'évolution de nombreux systèmes est gouvernée par le milieu extérieur. Pour la synergétique, cela se traduit mathématiquement par la variation des paramètres d'ordre (ou de contrôle) du système due à l'environnement. Le système voit ainsi sa stabilité perturbée, ce qui peut conduire à d'autres motifs, d'autres régularités.

Pour modéliser ce principe de fonctionnement, la synergétique a recours à la théorie des bifurcations dans les systèmes dynamiques [Arnold *et al.*, 1999, Haken, 1979]. Cette démarche, de l'aveu même de l'auteur [Haken, 2008], partage ce principe avec la théorie des catastrophes de René Thom (1975), et se veut une théorie unificatrice des phénomènes d'auto-organisation. Sans vouloir nous prononcer sur cette ambition de grande envergure, ce qui nous semble important dans la synergétique est l'approche de l'auto-organisation en termes de systèmes dynamiques contrôlés par des paramètres qui vont modifier les attracteurs du système, à l'instar de la théorie des bifurcations.

Ainsi la synergétique repose sur un principe général dit principe d'asservissement[28] illustré sur la figure 4.2 à partir du cas de la photonique des lasers : les électrons subissent l'effet du champ électromagnétique auquel ils sont soumis qui agit sur leur dynamique comme un paramètre d'ordre (ou de contrôle), dans ce sens, le champ "asservit" les électrons. En retour, les électrons modifient par leur dynamique collective le champ dans lequel ils sont plongés en produisant un champ complémentaire : c'est le principe de "causalité circulaire"[29] illustré par

[27]"Self-organization is the spontaneous often seemingly purposeful formation of spatial, temporal, spatio-temporal structures or functions in systems composed of few or many components. In physics, chemistry and biology self-organization occurs in open systems driven away from thermal equilibrium."

[28]"Slaving principle"

[29]"Circular causality"

la figure 4.3. Nous retrouvons les éléments de la terminologie de la cybernétique. Les schémas d'illustration se fondent sur le fonctionnement de la mise en cohérence des photons d'un laser, érigé en principe.

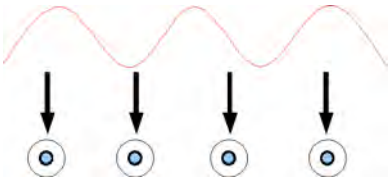


FIG. 4.2 – Illustration du principe d'asservissement.

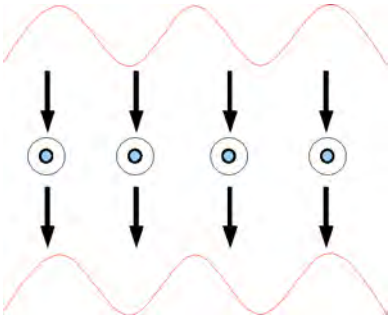


FIG. 4.3 – Illustration du principe de causalité circulaire.

Positionnement de notre approche.

Nous adhérons à cette vision de l'auto-organisation proposée par la synergétique. Nous en exploitons d'ailleurs le principe général, à savoir que les paramètres de contrôle/d'ordre pilotent le système auto-organisé via l'environnement, pour aboutir à la modification de la forme ou de la nature des attracteurs du système. Sans adopter le formalisme de la synergétique, nous développons dans cette thèse une sorte de paradigme minimal mettant en oeuvre le principe d'asservissement par le(s) champ(s) de l'environnement.

4.3 Mécanismes de l'auto-organisation biologique

L'intelligence en essaim est un domaine fortement bio-inspiré, on l'a vu au chapitre 1. Ce processus de l'inspiration biologique passe naturellement par la connaissance de l'analyse biologique des mécanismes impliqués dans un phénomène naturel. Force est de constater que c'est en biologie animale que les phénomènes d'auto-organisation sont les plus spectaculaires. Les phénomènes biologiques ont l'avantage d'être accessibles, nombreux et souvent à une échelle macroscopique facilement observable (colonnes de fourmis, nuées d'oiseaux, bancs de poissons,...). Cependant le recueil des données expérimentales s'avère délicat dans la plupart des cas, ce qui rend difficile la confrontation des modèles à leur réalité expérimentale.

Néanmoins, l'intérêt de la biologie est de révéler des mécanismes fondamentaux et spécifiques aux espèces animales concernant l'auto-organisation. Nous estimons que toute démarche de modélisation de l'intelligence en essaim doit prendre ces éléments en considération.

Concept d'auto-organisation en biologie.

Les biologistes s'accordent sur la définition de l'auto-organisation [Camazine *et al.*, 2001] suivante :

Définition 4.3.1. *L'auto-organisation est un processus au cours duquel les formes observées au niveau global (macroscopique) du système émergent uniquement des nombreuses interactions entre les parties constituantes du système au niveau inférieur (microscopique). De plus les règles régissant les interactions au niveau des constituants ne sont exécutées qu'en fonction d'informations locales, sans référence à la forme globale.* [30]

De cette définition ressort qu'il n'y a aucun contrôle centralisé qui dirigerait la construction des "formes" au niveau global. De même, l'émergence décrite ne se réalise que grâce à des interactions locales, c'est-à-dire sans information sur la totalité du système. Ces interactions peuvent se produire de façon directe (par contact par exemple) ou indirecte (ex. : dépôt de phéromones chez les fourmis, signaux visuels chez les poissons ou les oiseaux). Notons que la notion de forme est assez délicate à manipuler car elle n'a pas de définition mathématique précise. On se contente bien souvent d'évoquer une liste d'exemples remarquables mais peu caractérisables géométriquement. Voyons quels mécanismes ont été identifiés par les biologistes pour expliquer l'auto-organisation.

Les mécanismes d'auto-organisation biologiques.

Rétroaction positive et négative. Parmi les mécanismes responsables de l'auto-organisation biologique, on retrouve des mécanismes fondamentaux mis en exergue en cybernétique, à savoir les mécanismes de rétroactions positives et négatives considérées comme des modes d'interaction entre entités formant le système biologique [Camazine *et al.*, 2001]. La rétroaction négative est un processus connu de longue date dans les systèmes biologiques. Ce type de rétroaction a pour fonction essentielle de réguler, stabiliser certaines grandeurs caractéristiques d'un système autour d'un point d'équilibre et de contrebalancer toute perturbation à l'équilibre établi : par exemple dans le cas de grandeurs physiologiques chez l'homme, le taux de glucose dans le sang est régulée par la production d'insuline, la température corporelle est régulée par un comportement approprié (frissons, sudation,. . ....). Dans chaque cas des capteurs spécifiques mesurent ces grandeurs physiologiques particulières afin d'induire le comportement de réaction approprié qui rétablira l'équilibre. En revanche, la rétroaction positive tend à amplifier les perturbations et donc à quitter l'état d'équilibre. Amplification, auto-entrainement, auto-catalyse, sont des termes associés au processus de rétroaction positive.

[30] "Self-organization is a process in which pattern at the global level of a system emerges solely from numerous interactions among the lower-level components of the system. Moreover, the rules specifying interactions among the system's components are executed using only local information, without reference to the global pattern."

L'exemple typique pour expliquer les deux processus antagonistes de rétroaction se trouve en démographie : l'évolution d'une population dépend de trois facteurs essentiels, la mortalité, la natalité, et la capacité de charge (c'est-à-dire le nombre maximum d'individus d'une espèce qu'un territoire donné peut supporter). Cette évolution est exprimée mathématiquement dans l'équation démographique originale de Pierre-François Verhulst en 1838. Nous la présentons ici car elle est "l'ancêtre" à temps continu de l'application logistique utilisée dans cette thèse dans le cadre des sytèmes complexes :

$$\frac{dP}{dt} = r\ P\ (1 - \frac{P}{K}) = rP - r\frac{P^2}{K} \tag{4.1}$$

où P désigne la taille de la population, r le taux de croissance (calculé à partir des taux de natalité et de mortalité et du taux net de migration) et K la capacité de charge ou capacité limite. On voit que cette équation comporte deux termes : le premier terme est l'expression d'une rétroaction positive et le second d'une rétroaction négative sans laquelle la solution de l'équation est une exponentielle qui n'est pas viable dans le contexte biologique. Ce type d'équations est à la base de la classe des équations de populations, dont une version proies-prédateurs est donnée par l'équation de Lotka-Volterra (1925). Notons pour finir l'observation empirique suivante : les rétroactions positives sont souvent d'origine individuelle, tandis que les rétroactions négatives sont souvent dues aux limitations et contraintes environnementales [Camazine *et al.*, b]. C'est en partie par ces deux processus antagonistes qu'émergent les formes de l'auto-organisation biologique dans les grands groupes d'entités en interaction.

L'interaction entre individus. L'interaction entre les individus ou parties constituant un système biologique est centrale pour la problématique de l'auto-organisation. Car l'interaction procède par transfert d'information au sein du système. Elle est l'occasion pour les individus de capter et de transmettre l'information qui influe sur leur comportement. L'interaction peut donc être interprétée comme un échange, un transfert, un traitement de l'information locale au sens de la théorie de l'information, à l'image de la mise en contact de deux corps qui atteignent un équilibre thermique par transfert de chaleur en thermodynamique. Sur ces questions difficiles des liens entre organisation biologique et théorie de l'information, les travaux d'Henri Atlan [Atlan, 2006] figurent comme une référence.

Notre conception de l'information ici revêt un caractère moins statistique, et correspond simplement au transfert d'une quantité liée à une grandeur caractéristique du système (comme la phéromone chez les fourmis par exemple). On peut distinguer deux modalités principales pour ce transfert d'information : transfert (direct) d'individu à individu du groupe, ou transfert (indirect) par modification de l'environnement lui-même. La question de la nature des interactions est donc fondamentale et préalable à toute modélisation de l'auto-organisation.

Le phénomène de stigmergie. Les interactions indirectes en biologie sont principalement liées au phénomène de stigmergie. La stigmergie est un concept né des études de Grassé en 1959 sur la construction de termitières [Grassé, 1959] : il a conclu que la termitière en cours de construction a un effet amplificateur sur le travail des termites. L'objet qui émerge de la construction collective participe donc à son propre développement puisqu'il "transmet" de façon

passive l'information sur son état courant à toute la colonie. Au départ, le dépôt de morceaux de matériau de construction est désordonné, témoignant d'une phase de fluctuations, puis peu à peu ce dépôt se structure et une forme globale apparaît par le jeu des actions conjointes des termites sans autre forme de communication. Cette forme d'interaction est vue comme sociale et indirecte [Michener, 1974] : chaque élément de la colonie réalise ses actions propres sans prise en compte directe de celle de ses voisins.

On peut différencier en fait deux classes principales de stigmergie [Bonabeau *et al.*, 1999c] selon que le stimulus change de nature ou bien d'intensité au cours de la réalisation d'une tâche collective. La stigmergie qualitative rend compte de l'effet des stimulus qualitatifs, par exemple : la présence d'un trou dans une surface plane sera comblé pour retrouver la planéité de la surface, la planéité (nouveau stimulus) déclenche à son tour la construction de cellules pour l'élevage des larves de la colonie. La stigmergie quantitative quant à elle ne fait intervenir qu'un seul type de stimulus, mais dont l'intensité varie en fonction de la réalisation de la tâche. Ainsi la stigmergie à base de phéromone est considérée comme quantitative : elle ne change *a priori* pas de nature au cours de la stimulation et le comportement de la colonie est gouverné par la concentration de phéromone présente sur l'objet de construction. Ainsi la construction d'une piste dans le cas du fourragement par exemple, est-elle considéré comme un effet de stigmergie quantitative, même si l'objet de construction dans ce cas est la trace de phéromone elle-même qui indique le chemin à suivre.

Propriétés de l'auto-organisation biologique.

Ces mécanismes biologiques fondamentaux étant décrits, les principales propriétés des phénomènes biologiques d'auto-organisation sont les suivants [Camazine *et al.*, a] :

- Ce sont des processus dynamiques, dont les éléments constitutifs sont en interaction permanente pour maintenir l'auto-organisation du système. Ces processus dynamiques sont de nature non linéaire.
- Ces processus donnent lieu à des propriétés nouvelles émergentes, non présentes au sein des parties élémentaires et non simplement liées à leurs propriétés élémentaires (formation d'amas chez certaines larves d'insectes par exemple). De plus ces processus peuvent mettent en oeuvre des mécanismes localement simples mais qui produisent à grande échelle des résultats complexes.
- Ces processus sont paramétriques : certains paramètres liés au fonctionnement biologique (la composition chimique d'une phéromone par exemple) et d'autres de nature physique (par exemple la volatilité d'une phéromone, la température, les mouvements de l'air vont jouer sur son évaporation) peuvent avoir un fort impact sur les processus d'auto-organisation. Cette dépendance paramétrique renforce aussi le rôle et l'importance de l'environnement.

De fait, certaines de ces propriétés rejoignent celles déjà constatées dans d'autres domaines (en physique notamment), ce qui conforte l'idée d'une unité des mécanismes impliqués.

4.4 Auto-organisation en intelligence artificielle

Comme nous l'avons évoqué sur les intuitions pionnières des acteurs de la cybernétique, le concept central d'auto-organisation s'est développé de façon concomitante à la naissance de l'intelligence artificielle, de la théorie de l'information, parfois par les mêmes personnes. L'ordinateur est devenu un outil indispensable pour explorer ces domaines. De plus l'informatique en tant que science de l'information est directement concernée par l'auto-organisation dans les systèmes artificiels.

Les premiers champs de l'informatique à avoir exploité les phénomènes d'auto-organisation sont l'algorithmique évolutionnaire ainsi qu'un courant de recherche baptisé "computation émergente", si l'on met de côté l'intelligence en essaim plus récente dont nous avons dessiné les contours au chapitre 1.1. L'intérêt de l'approche informatique tient à son utilisation extensive de la notion d'agent qu'elle développe par ailleurs et qui est particulièrement adaptée à l'étude des phénomènes d'auto-organisation.

Algorithmique évolutionnaire La première véritable application à l'informatique de concepts biologiques de l'auto-organisation est l'algorithmique évolutionnaire née dans les années 70 avec entre autres Rechenberg et Schwefel [Rechenberg, 1973, Beyer et Schwefel, 2002] et J. Holland dès 1975 [Holland, 1992]. Elle s'inspire des mécanismes d'adaptation et d'évolution à l'oeuvre dans la nature pour créer des algorithmes capables de s'adapter à leur environnement pour atteindre un certain but. Plus précisément cette algorithmique se fonde sur un paradigme darwinien, mettant en oeuvre des principes d'évolution sur des espèces constituées d'individus/programmes informatiques :

- un principe d'objectif à atteindre (lié au problème posé) : cet objectif va se traduire en termes de survie et de reproduction de l'espèce. La spécification de l'objectif permet de définir une fonction dite de "fitness".
- un principe de sélection : l'avantage va au plus adapté (on sélectionne avec la fonction de "fitness")
- un principe de variabilité : on introduit des variations d'une génération à l'autre (croisement et mutation par exemple pour les algorithmes génétiques) pour explorer l'espace de recherche

Le résultat du processus d'adaptation donnera les meilleures solutions au problème. On trouve dans cette classe d'algorithmes les algorithmes génétiques, les algorithmes évolutionnaires et le recuit simulé pour les principaux. La communauté de l'algorithmique évolutionnaire emploie le terme d'auto-adaptation pour désigner la capacité des systèmes évolutionnaires à s'adapter dynamiquement et de façon autonome aux changements de leur environnement. Ce concept se répand dans la communauté évolutionnaire et au-delà, au même titre que l'auto-organisation. Holland utilise explicitement la notion d'agent, de système multi-agent comme architecture clé de l'informatique évolutionnaire bio-inspirée. Ce courant de l'informatique a également donné naissance à une branche spécifique des systèmes complexes appelée "systèmes complexes adaptatifs" (cf. section 5.2).

Computation émergente En ce qui concerne la "computation émergente", cette approche est née de la volonté de se démarquer de l'approche "à la physicienne" de l'auto-organisation et surtout d'intégrer la théorie du calcul comme contexte à part entière de ces problématiques. Il s'agissait donc de faire naître un courant spécifique dédié aux problématiques émergentistes de l'informatique et de montrer que ces phénoménologies existent également avec les objets manipulés par la théorie du calcul.

Stephanie Forrest utilise la notion d'agent pour définir la computation émergente [Forrest, 1991].

Définition 4.4.1. *La computation émergente est constituée des éléments suivants :*
- *Un ensemble d'agents programmés explicitement : un micro-programme local à l'agent définit son comportement localement (typiquement les automates cellulaires).*
- *Des interactions entre ces agents, réglés par leur propre programme, qui ont pour résultante la formation d'un schéma global ("pattern") au niveau macroscopique.*
- *L'interprétation naturelle de ces schémas comme du calcul.*

Dans cette définition entrent plusieurs domaines de développement de l'informatique : les modèles connexionnistes, les automates cellulaires, les modèles biologiques, les systèmes classifieurs, les modèles de vie artificielle, et les modèles de coopération dans les systèmes sociaux sans autorité centralisée. Forrest insiste sur le fait que la computation émergente concerne des processus de calcul non linéaires sans quoi il n'y aurait pas d'émergence, notion liée fortement à l'imprédictibilité intrinsèque des motifs générés. Enfin elle cible trois types d'applications possibles pour la computation émergente :

Le calcul parallèle. Le développement des machines parallèles laisse espérer une augmentation des performances plus que linéaire grâce aux techniques de computation émergente. Aujourd'hui encore cette perspective reste à démontrer expérimentalement.

Les langages de programmation. Dans la théorie des langages classique, les instructions s'opèrent indépendamment du contexte et suivent des lois de composition qui assurent autant que possible d'obtenir le résultat escompté. Ainsi par analogie avec la physique, un principe de superposition s'applique sur les composants logiciels, principe fondamentalement lié à la linéarité des opérations mises en oeuvre. Lorsque l'on aborde la computation émergente, ces opérations ne sont plus linéaires et certains théorèmes de la théorie des langages ou de la calculabilité ne s'appliquent plus. Cela ouvre des perspectives nouvelles qu'il faut explorer.

Les algorithmes de recherche. Trouver une solution à un problème dans un espace de recherche donné est un problème récurrent et générique de l'informatique. Les systèmes intelligents se caractérisent justement par le fait qu'ils peuvent trouver les solutions approchées acceptables dans de grands espaces de recherche associés à des problèmes intractables, par le recours à des heuristiques ou des algorithmes utilisant des aspects spécifiques du problème. La différence d'approche entre les algorithmes classiques de l'intelligence artificielle et les algorithmes émergentistes est particulièrement marquée dans ce cadre. L'approche émergentiste inclue les algorithmes évolutionnaires et l'intelligence en essaim.

4.5 Bilan intermédiaire sur l'auto-organisation

A l'issue de ce rapide survol autour des conceptions sur l'auto-organisation, voici les points qui nous semblent importants ici :

- L'auto-organisation concerne de nombreux domaines scientifiques. Le contexte biologique tient une place centrale concernant les réflexions sur l'auto-organisation, par la quantité de phénomènes typiques qu'il présente, par le fort aspect démonstratif de ces phénomènes, par la source importante d'inspiration et d'objets de modélisation qu'il constitue. L'auto-organisation biologique est au coeur de l'intelligence en essaim biologique et informatique.
- Une condition nécessaire à l'auto-organisation, qu'elle soit de nature biologique, informatique ou physico-chimique, réside dans l'aspect non linéaire du système. De plus, ces systèmes sont paramétrés, le principe d'asservissement par le champ.
- De même la notion d'interaction au sein du système est une condition nécessaire à l'auto-organisation et la biologie nous apprend qu'elle peut s'exercer de façon directe ou indirecte.
- Enfin la notion de boucle de rétro-action (biologie), de causalité circulaire (physique) ou de récurrence du calcul (informatique) est essentielle à l'émergence de formes nouvelles.

Chacun des points précédents appelle une réponse lorsque l'on cherche à modéliser un système auto-organisé, qui plus est dans le domaine de l'intelligence en essaim. C'est pourquoi on s'attache dans cette thèse à intégrer les mécanismes génériques identifiés sur l'auto-organisation biologique (stigmergie, rétro-action, interactions) à l'aide d'un modèle non linéaire paramétré. Les aspects récursifs et interactionnels des deux derniers points précédents sont amenés par une modélisation de type systèmes complexes au sens défini en section suivante.

5

Systèmes complexes pour l'intelligence en essaim.

Malgré une conceptualisation assez précoce, l'auto-organisation a souffert dès son apparition d'un manque de théorisation (dès la cybernétique), c'est-à-dire plus pragmatiquement de modèles mathématiques suffisamment génériques pour en décrire et prédire les différents types de manifestation. La théorie de l'information de Shannon ou la thermodynamique physique ont pu jouer ce rôle dans les premiers temps, mais elles se sont révélées insuffisantes. Nous montrons dans ce chapitre que le champ des systèmes complexes ouvre de nouvelles perspectives pour dépasser cet obstacle théorique.

Le recours aux méthodologies des systèmes complexes est motivé par un certain nombre de lacunes identifiés dans les divers champs scientifiques concernés par l'auto-organisation :

- la notion de forme ou de motif émergent est formulée de façon floue, sans description mathématique vraiment utilisable,
- l'auto-organisation est rarement mesurée car la mesure de l'auto-organisation dans les systèmes est un problème difficile à identifier mathématiquement comme nous le verrons en section 5.1,
- les modèles actuellement développés, notamment ceux visant l'auto-organisation biologique et en particulier l'intelligence en essaim, ne s'inspirent pas assez selon nous des théories abstraites et génériques des systèmes complexes qui permettraient d'unifier certains aspects phénoménologiques.

Le premier et le dernier point dépendent des outils mathématiques utilisés pour décrire les systèmes étudiés. Nous nous positionnons à ce niveau dans la théorie des systèmes dynamiques non linéaires, et ses théories connexes que sont la théorie du chaos déterministe et la théorie des bifurcations (dont tous les éléments utiles sont présentés en annexe de ce document A). Nous estimons que ces théories disposent de concepts suffisamment abstraits pour répondre notamment au premier point sur la forme donnée par la notion d'attracteur, et au dernier point par l'universalité des phénomènes qu'elles peuvent décrire et générer.

Le deuxième point dérive aussi de la théorie dans laquelle on raisonne. La notion de complexité vise à quantifier l'auto-organisation spontanée. Les systèmes dans lesquels cette complexité

existe sont appelés systèmes complexes. La détermination de la complexité des systèmes fait l'objet de la première section de ce chapitre.

5.1 Mesure de l'auto-organisation et notion de complexité

La mesure de l'auto-organisation est une question épineuse. Car il est difficile de conclure à un phénomène d'auto-organisation avec des arguments irréfutables. On peut définir un motif ou "pattern" comme vu précédemment en synergétique, mais difficilement le quantifier, au regard de la diversité des formes rencontrées. La recherche d'outils mathématiques capables de distinguer et de caractériser expérimentalement l'auto-organisation lorsqu'elle émerge, est un objectif crucial pour unifier ce domaine. La mesure de l'auto-organisation passe par la définition d'une grandeur, celle de la complexité du système. D'une certaine façon, on a transféré l'ambiguïté de définition de l'auto-organisation sur la notion de complexité qui possède un cadre de définition mathématique. Cette section présente différentes versions du concept de complexité pour leur valeur didactique et qualitative davantage qu'en vue d'un usage quantitatif, car nous ne les utilisons pas dans cette thèse. Ces notions nous permettront de mieux cerner le positionnement et la problématique des systèmes dits complexes.

La complexité statistique comme mesure de l'auto-organisation. Les auteurs de [Shalizi et Crutchfield, 2 proposent un nouveau citère pour mesurer l'auto-organisation lié à la variation de la complexité statistique du système. Ils partent du constat que les diverses tentatives pour mesurer l'auto-organisation ont été vouées à l'échec, notamment celles qui se fondent sur les mesures de l'entropie thermodynamique du système qui sont valables pour de nombreux systèmes physiques mais qui échouent à mesurer l'organisation des systèmes biologiques notamment. Fondée sur la théorie de l'information de Shannon, la complexité statistique repose sur la notion de complexité proposée dans [Grassberger, 1986] qui est définie comme :

> la quantité d'information minimale sur l'état du processus considéré qui maximise la précision de la prédiction de l'état futur.

La complexité statistique reprend cette notion et fournit une définition plus opératoire sur les concepts proposés. Cette mesure est testée sur des automates cellulaires cycliques les mieux étudiés et reconnus pour les formes d'auto-organisation qu'ils génèrent. Nous ne détaillons pas davantage cette complexité, dont une synthèse récente figure dans l'article [Shalizi *et al.*, 2004]. Comparé à d'autres approches, l'intérêt de cette complexité réside entre autres dans le fait qu'elle est expérimentalement calculable.

Complexité aléatoire et complexité organisée. Une notion plus simple à appréhender en informatique est la notion de complexité organisée. On peut retrouver l'idée de complexité organisée très tôt à l'époque de la cybernétique dans l'article [Weaver, 1948]. Ce texte expose déjà en des termes simples la confrontation entre les problèmes traités par la physique et ceux traités en biologie à travers le concept de complexité. Weaver y distingue deux types de complexité, la complexité désorganisée rencontrée en physique et la complexité organisée rencontrée principalement en biologie. L'auteur évoque par l'exemple simple d'un billard le paradoxe des outils

du physicien : la bille de billard peut être modélisée facilement par les lois de la mécanique classique car le nombre de variables est faible et la dynamique déterministe associée est simple ; dès que l'on augmente progressivement le nombre de billes dans le billard, le problème devient très vite impossible à résoudre sans l'aide d'un ordinateur ; en revanche dès que le nombre de billes devient extrêmement grand (cas du modèle du gaz parfait où les molécules sont de l'ordre du nombre d'Avogadro 10^{23}, il est de nouveau possible d'acquérir une information sur le système, grâce aux approches statistiques qui deviennent applicables. Il s'agit typiquement d'un problème de complexité désorganisée pour Weaver. Les problèmes rencontrés dans les phénomènes vivants et dans les sciences humaines sont des problèmes à nombre de variables intermédiaires, qui présentent des phénomènes d'organisation spontanée révélant une complexité organisée. Ce concept de complexité organisée est repris dans l'ouvrage [Delahaye, 1999] dans une formalisation mathématique aboutie. L'auteur met en opposition deux complexités (même s'il précise qu'elles peuvent se rejoindre parfois) : la complexité aléatoire et la complexité organisée.

Complexité aléatoire ou "de Kolmogorov". Tout d'abord, la complexité aléatoire est formalisée par la complexité dite "de Kolmogorov" :

Définition 5.1.1. *On définit la complexité de Kolmogorov $K(s)$ d'une suite finie binaire s par :*
$K(s) \equiv$ *longueur du plus court programme (programme minimal) qui engendre s*

Il faut distinguer cette complexité de Kolmogorov de la complexité des algorithmes, car elles ne sont pas équivalentes. Ainsi la première considère une suite s fixée, alors que la seconde considère l'augmentation asymptotique de la quantité de calcul ou d'espace lorsque la taille des données augmente. La complexité de Kolmogorov (complexité K) mesure la quantité d'aléatoire d'un système d'où son appellation de complexité aléatoire : la complexité K d'une suite de nombres aléatoires est grande, tandis que celle du calcul des k premières décimales de π ne l'est pas car on peut utiliser un programme court pour les calculer. La complexité K comme la complexité des algorithmes ne conviennent pas pour évaluer la complexité organisée : la complexité K augmente lorsque l'on passe d'un être vivant à un gaz (pour simplifier, on passe d'un système décrit par un code génétique à un système complètement aléatoire) ; pourtant l'être vivant est plus complexe qu'un gaz au sens de l'organisation. Ainsi la complexité K augmente avec l'incompressibilité, l'imprévisibilité et l'absence de structure de la suite s. L'idée est donc plutôt de pouvoir évaluer la "quantité en calcul" contenue dans un objet : intuitivement l'organisation apparaît en effet après un long processus de calcul.

Profondeur logique de Bennett. Dans cette idée, le concept de "profondeur logique de Bennett" [Bennett, 1986] rend mieux compte de la complexité organisée :

Définition 5.1.2. *La profondeur logique de Bennett de la suite binaire s est définie par :*
$P(s) \equiv$ *temps de calcul du programme minimal de s*

Comparons sur quelques exemples les deux complexités précédentes : un cristal est peu profond (car c'est un schéma répétitif) et a une faible complexité K, le nombre π a une faible complexité K mais une profondeur logique importante car il faut un temps de calcul important

pour calculer un grand nombre de décimales, un gaz est aléatoire et peu profond, un être vivant possède une grande complexité K mais surtout est profond au sens de Bennett (expression d'un code génétique). Ainsi deux résultats complémentaires de Bennett accompagnent la profondeur logique :

1. lors de processus déterministes, la profondeur logique ne peut s'accroître de façon brusque, la complexité organisée met du temps à se déployer dans ces processus ;
2. lors de processus probabilistes, la probabilité d'apparition d'objets profonds est très faible.

Enfin, mentionnons pour finir une limitation forte liée à la complexité K et à la profondeur logique de Bennett : ces deux grandeurs sont approchables mais difficilement utilisables en pratique à cause de propriétés fortes d'indécidabilité [Delahaye, 1999]. Ainsi en résumant succinctement, on ne peut pas savoir si l'on peut toujours trouver un programme minimal pour toute suite finie s et on ne sait pas si ce programme peut terminer en temps fini.

Pour conclure. Après ce court résumé des problématiques autour de l'aspect mesurable de la complexité, qui consitue une voie de quantification de l'auto-organisation, on conçoit la difficulté de l'entreprise : si les mesures qui viennent d'être décrites semblent plus appropriées ou plus opérationnelles aujourd'hui, d'autres mesures de complexité existent sans qu'aucune n'ait véritablement emporté l'adhésion de la communauté scientifique. D'autre part, ces mesures, à de rares exceptions, restent bien souvent difficiles à mettre en oeuvre en termes de calculabilité.

Cette transition de l'auto-organisation à la complexité nous conduit naturellement vers les systèmes dits complexes, dont nous traçons les contours du domaine en section suivante.

5.2 La science des systèmes complexes : une science fédératrice autour de l'auto-organisation

5.2.1 Définitions du domaine

On peut faire remonter la notion de système complexe à Aristote à qui l'on doit l'aphorisme célèbre suivant :

> Le tout est plus que la somme des parties.

Au-delà de cette courte affirmation, la science dite des systèmes complexes se donne pour objet d'étayer cette affirmation dans une démarche scientifique. Nous avons sélectionné dans cette section quelques-unes des conceptions sur les systèmes complexes qui correspondent le mieux à l'approche que nous développons dans cette thèse.

Positionnement et première définition

En termes de définition du domaine, la communauté scientifique s'accorde pour désigner les systèmes complexes comme des systèmes constitués de multiples parties, composantes, ou entités en interactions mutuelles. Les définitions restent cependant très générales, autorisant de nombreux cas d'études dans des contextes très divers. Le graphique de la figure 4.1 ne rend

pas complètement compte des liens entre les diverses branches représentées mais nous pourrions rajouter clairement des arcs dans ce graphe entre les approches orientées agents et les divers noeuds des sciences de la complexité. Nino Boccara dans la préface de son ouvrage dédié aux systèmes complexes [Boccara, 2004] propose la définition à base d'agents suivante :

Définition 5.2.1. *Bien qu'il n'existe pas de définition universellement acceptée pour définir un système complexe, la plupart des chercheurs décrirait comme complexe un système d'agents liés les uns aux autres qui exhibent un comportement global émergent non pas imposé par un contrôleur central, mais résultant des interactions entre les agents. Ces agents peuvent être des insectes, des oiseaux, des personnes, ou des entreprises, et leur nombre peut varier d'une centaine à un million*[31].

Boccara précise que ces systèmes échappent à de rares exceptions près à l'analyse mathématique et que leur étude doit avoir recours massivement à la simulation informatique. La plupart des progrès dans ce domaine ont été rendus possibles par simulation informatique. Boccara rappelle ensuite que l'objectif de la modélisation à base d'agents n'a pas vocation à étudier un système multi-agent en particulier mais les grands principes de conception que l'on pourra appliquer dans d'autres contextes. C'est donc le pouvoir explicatif et l'aspect archétypal de ces modèles qui doit être recherché, ainsi que des principes de conception les plus simples et les plus génériques possibles.

Importance du chaos dans les systèmes complexes

D'autres auteurs focalisent leur modélisation à des niveaux plus abstraits encore. Les travaux du physicien Kaneko sur les réseaux d'itérations couplées sont à ce titre exemplaires. Ainsi dans le premier chapitre de l'ouvrage [Kaneko et Tsuda, 2001b], les auteurs Kaneko et Tsuda affirment la forte relation entre chaos et systèmes complexes. Il est selon eux impossible d'aborder les systèmes complexes sans faire intervenir la théorie du chaos déterministe[32].

Pour soutenir cette hypothèse forte comparée aux recours à la non linéarité des systèmes au sens large vus précédemment, les auteurs reviennent tout d'abord sur l'approche réductionniste qui consiste à étudier un système en le réduisant à la somme de ses constituants élémentaires. Si cette approche a donné des résultats dans de nombreuses situations de systèmes "simples" ou linéaires, dans lesquels un principe de superposition peut s'appliquer, elle ne permet pas d'accéder à la compréhension des phénomènes macroscopiques dans le cas des systèmes complexes. Quand bien même l'étude des constituants élémentaires reste nécessaire, cela n'implique pas si simplement de pouvoir remonter au niveau global du phénomène. Les auteurs pointent justement le chaos déterministe comme le type de phénoménologie qui met en défaut cette approche réductionniste et quelques-unes de ses propriétés remarquables :

[31] Although there is no universally accepted definition of a complex system, most researchers would describe as complex a system of connected agents that exhibits an emergent global behavior not imposed by a central controler, but resulting from the interactions between the agents. These agents may be insects, birds, people, or companies, and their number may range from a hundred to a million.

[32] Le chaos déterministe exprime la sensibilité d'un système aux conditions initiales. Une définition plus précise du chaos est donnée en annexe A.

- Un système chaotique ne peut être ramené à la superposition de ses modes propres obtenus par transformée de Fourier puisqu'ils sont indénombrables (ce qui révèle d'ailleurs expérimentalement la présence du chaos).
- D'autre part, le chaos n'implique pas un grand nombre de degrés de liberté dans le système, puisque le chaos apparaît dès la dimension 3 en temps continu et à des dimensions inférieures à temps discret.
- Il dépasse ainsi le cadre réductionniste tout en conservant une complexité à faible échelle.
- Il faut comprendre le chaos, nous disent les auteurs, comme un "processeur d'information" et l'exploiter en tant que tel. En cela ils rejoignent la vision de Nicolis sur l'intérêt biologique des phénomènes chaotiques [Nicolis, 1991].
- Le chaos a la propriété unique d'intégrer à la fois l'ordre et l'aléatoire.

La présence de chaos n'est pas suffisante pour réaliser un système complexe. Encore faut-il en déterminer la localisation dans le système. Celui-ci est avant tout un grand réseau d'interactions entre entités élémentaires, s'influençant les unes les autres, où une petite fluctuation peut se voir amplifiée dans la chaîne de causes à effets du réseau (ex. : écosystèmes, réseaux de neurones, système immunitaire,...). Cet effet global du réseau ne peut être modélisé par une simple superposition de modes propres. C'est pour toutes ces raisons que Kaneko et Tsuda proposent une approche constructive d'étude des systèmes complexes, ce qui équivaut dans leur conception à des modèles de nature déterministe. Sur cette notion de constructivisme scientifique déterministe, on peut consulter le très intéressant ouvrage [Bourgine *et al.*, 2008] sous la direction de Paul Bourgine.

Cette approche est principalement fondée sur des modèles de calcul abstraits capables de reproduire les phénomènes chaotiques d'entités élémentaires et de les coupler en réseau. Cela donne lieu en particulier aux réseaux d'itérations couplées[33] [Kaneko, 1993], paradigme de système complexe, dont nous décrivons la formalisation en section 5.4. Comme le rappellent les auteurs, ce type de modèle n'a pas vocation à reproduire des données expérimentales même de façon approchée, il est plutôt une abstraction d'un réseau complexe générique, outil d'exploration et de synthèse des phénoménologies complexes.

Systèmes complexes adaptatifs.

Dans le cadre d'étude des systèmes complexes, un sous-domaine s'est détaché sous l'appellation de "systèmes complexes adaptatifs". Ce courant issu des développements de l'algorithmique évolutionnaire des années 70-80 (cf. 4.4) met l'accent sur les systèmes complexes en relation avec un environnement auquel ils s'adaptent dynamiquement. Ces systèmes sont nombreux dans la nature et notamment en biologie (réseaux de neurones, dynamique de population, reproduction des espèces et évolution, système immunitaire, ...). Les systèmes complexes adaptatifs sont donc principalement concernés par les mécanismes d'apprentissage et d'adaptation [Booker *et al.*, 2005]. Ces mécanismes permettent en effet de sélectionner un ou des schémas spécifiques générés par le système complexe dans son ensemble qui répondent à un problème de satisfaction de contraintes ou d'optimisation d'une fonction objectif.

[33] Coupled Map Lattice (CML)

Les algorithmes bio-inspirés de façon générales sont à classer dans cette catégorie, en particulier les algorithmes de l'intelligence en essaim. Le domaine de la "vie artificielle" [Farmer et d'A. Belin, 1990] se veut encore plus large dans son projet et vise à synthétiser des systèmes complexes adaptatifs qui simulent la vie.

La science des systèmes complexes.

Les conceptions précédentes ne représentent pas l'ensemble des systèmes complexes. Pour décrire ce domaine, Cosma Shalizi a tenté d'en dresser un panorama dans [Shalizi, 2006] et de renvoyer une vision structurée de la science des systèmes complexes. Cette structuration intègre les éléments précédemment décrits.

Shalizi commence par définir de façon très générale un système complexe comme un système composé de multiples parties dont les comportements sont à la fois hautement variables et très dépendants des comportements des autres parties du système. La science des systèmes complexes a donc pour objet l'étude des systèmes complexes et comporte quatre pôles méthodologiques : un pôle "Schémas" (traduction de "pattern" ici), un pôle "Outils" mathématiques, un pôle "Fondements" théoriques et un pôle "Thématiques" d'études.

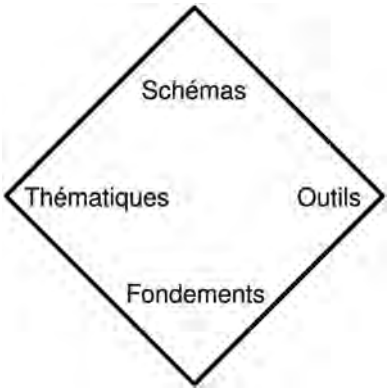


Fig. 5.1 – Pôles méthodologiques des sciences des systèmes complexes selon [Shalizi, 2006]

- Le pôle schémas consiste principalement en la réutilisation/l'inspiration de schémas ou de principes de fonctionnement de systèmes complexes d'un contexte scientifique dans un autre. L'informatique travaille régulièrement avec ce type de méthodologie en architecture logicielle avec notamment des "design patterns", des schémas de conception. Ainsi toute l'informatique bio-inspirée procède selon cette approche, dont les algorithmes fourmis sont une illustration évidente.
- Le pôle des fondements fait appel à la théorisation mathématique, dans les problématiques sur la définition et la mesure de la complexité et de l'auto-organisation (très lié au traitement et à l'analyse de données), des rapports entre l'information et le calcul entre autres. Ce pôle est capital pour assoir le domaine des systèmes complexes en tant que science.
- Le pôle relatif aux thématiques montre une grande diversité des contextes et des modèles rencontrés dans l'univers des systèmes complexes et qui chacun peuvent être sources

d'inspiration pour les autres. Il s'agit entre autres : des réseaux (sociaux, informatiques, neuronaux, de spins...), des algorithmes génétiques et évolutionnaires, de la turbulence, des systèmes physico-chimiques et de la morphogénèse, de la vie artificielle, des insectes sociaux et de leurs applications algorithmiques, des systèmes coopératifs, des dynamiques de groupes, du trafic automobile,... de l'économie évolutionnaire,... Il serait difficile de tracer les contours exacts du domaine en termes de thématiques abordées, mais on constate des liens très profonds entre les manifestations de ces divers champs d'étude.

- le pôle des outils est précisément celui qui caractérise sans doute le mieux la spécificité de la science des systèmes complexes. On peut y distinguer trois types d'outils méthodologiques :
  - Les outils d'analyse des données expérimentales : il s'agit des outils de statistique expérimentale, notamment l'étude des processus markoviens, et des outils d'analyse des systèmes non linéaires pour traiter les séries chronologiques.
  - Les outils de modélisation : on y compte principalement les systèmes dynamiques non linéaires, les automates cellulaires qui peuvent modéliser divers types de phénomènes spatialisés à grande échelle, les modèles à base d'agents qui sont parmi les modèles les plus employés aujourd'hui. A ce panel d'outils doit s'adjoindre une méthodologie d'évaluation des modèles qui passe entre autres par une analyse statistique ou analytique du modèle, la simulation, la comparaison des modèles entre eux.
  - Les outils de mesure de la complexité essentiellement fondés actuellement sur des extensions de la théorie de l'information.

Shalizi a pour sa part travaillé essentiellement sur les outils de mesure de la complexité (cf. [Shalizi *et al.*, 2004]).

5.2.2 Notre aproche : application logistique et RIC

Par rapport à ce qui vient d'être présenté, nous nous situons dans cette thèse dans la définition des systèmes complexes de Boccara 5.2.1. Nos outils de modélisation sont les systèmes multi-agents et les systèmes dynamiques non linéaires. Notre thématique et l'inspiration de nos schémas phénoménologiques se situent dans l'intelligence en essaim. Enfin, nous puisons les fondements de notre modélisation dans la théorie du chaos, à l'instar de Kaneko.

En accord avec ce que nous venons d'exposer sur les systèmes complexes, notre programme de conception d'un système multi-agent complexe pour l'intelligence en essaim implique donc d'avoir recours à des modèles simples capables :

- d'exhiber des propriétés non linéaires, et en particulier de générer des séries chaotiques,
- de proposer un schéma de calcul complexe, c'est-à-dire où de nombreuses unités de calcul sont interdépendantes.

Nous présentons dans les deux sections suivantes les modèles qui nous ont permis de répondre à ces deux objectifs, l'application logistique et les réseaux d'itérations couplées (RIC) respectivement, avec dans chaque cas les grandeurs clés qui permettent de les caractériser : exposants de Lyapunov et diagramme de bifurcation pour le premier, mesure de la synchronisation du système pour le second. L'application logistique possède une forme computationnelle minimale et des propriétés non linéaires universelles et est à la base des agents logistiques de la section 8.2. Les RIC donnent quant à eux le schéma de principe des couplages entre applications non linéaires et

figurent comme paradigme des systèmes complexes. Ils sont également à la base de la conception du SMAL.

5.3 L'application logistique : un modèle minimal non linéaire et chaotique

La théorie générales des systèmes dynamiques non linéaires[34] établit que de tels systèmes peuvent avoir des trajectoires très difficiles à décrire et à prédire à cause d'une sensibilité forte aux conditions initiales, et cela en dépit de règles de calcul déterministes. Ces trajectoires sont dites chaotiques [Ruette, 2003]. Il est établit que de tels phénomènes chaotiques ne peuvent apparaître qu'à partir de la dimension 3 pour les systèmes dynamiques à temps continus (cf. théorème de Poincaré-Bendixon A). En revanche, il apparaît à dimension plus faible dans les systèmes à temps discrets, qui sont par construction mieux adaptés à l'algorithmique des ordinateurs. Ainsi, la forme la plus simple des applications non linéaires à temps discret qui présente pourtant une large typologie de phénomènes non linéaires est une fonction quadratique appelée "application logistique". Dérivée de l'équation démographique de Pierre-François Verhulst en 1838 dont nous avons exposé les éléments en section 4.3, elle a été redécouverte et initialement analysée par Robert May en 1976 [May, 1976].

L'application logistique est définie par la formule récursive suivante (les notations utilisées ici sont celles habituellement en vigueur dans les ouvrages dédiés à la question, mais on utilisera également a comme paramètre pour cette application, ce qui correspond d'ailleurs à la notation originale de Robert May) :

$$x_{n+1} = f(x_n) = r\ x_n(1 - x_n) \tag{5.1}$$

où r désigne le paramètre de contrôle de l'application et x sa variable principale. Malgré la simplicité de cette formule de calcul, les itérées de f présentent tous les types de bifurcations des applications de dimension 1 en fonction de la valeur prise par r, ainsi qu'un comportement chaotique en particulier lorsque $r = 4.0$.
Les domaines de variation de ces grandeurs sont les suivants : $x \in [0, 1] \subset \mathbb{R}$ et $r \in [0, 4] \subset \mathbb{R}$ pour lesquels l'application f laisse $[0, 1]$ invariant. L'application est très peu utilisée en dehors de ces domaines de variation. Nous utilisons dans cette thèse le paramètre $a = \frac{r}{4}$ pour ramener les variations de r dans le segment $[0, 1]$ plus générique pour la modélisation des systèmes. Mais nous gardons dans cette section la présentation de cette application sous sa forme standard.

5.3.1 Analyse des séries temporelles de l'application logistique

Sur la figure 5.2 on peut voir le tracé de l'application f, qui est une parabole, en fonction de quelques valeurs du paramètre de contrôle. La bissectrice $x_{n+1} = x_n$ est également tracée car elle représente le lieu géométrique des points fixes de l'application. Son intersection avec la parabole de f donne donc immédiatement où se trouvent les points fixes. Ce type de graphique est également appelé "application de premier retour" puisque l'on construit le graphe de x_{n+1}

[34] Il est recommandé d'avoir consulté l'annexe A concernant cette théorie avant d'aborder la présente section.

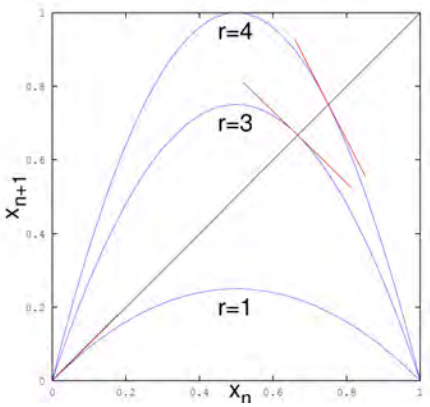


FIG. 5.2 – Tracé de $f(x)$ pour différentes valeurs du paramètre r, également appelée "application de premier retour". La tangente de la courbe pour $r = 3$ à l'intersection avec la première diagonale a une pente -1 : c'est le premier point de bifurcation de doublement de période.

en fonction de x_n.
La suite 5.1 a en effet deux points fixes vérifiant $\bar{x} = f(\bar{x})$ qui sont :

$$\bar{x_1} = 0 \tag{5.2}$$

$$\text{et } \bar{x_2} = 1 - \frac{1}{r} \tag{5.3}$$

La stabilité de ces points fixes est donnée par la pente de la tangente de f en ces points.

$$f'(x) = \frac{df}{dx}(x) = r(1 - 2\ x) \tag{5.4}$$

$$f'(x_1) = r \tag{5.5}$$

$$f'(x_2) = r(1 - 2\ (1 - \frac{1}{r})) = 2 - r \tag{5.6}$$

On peut remarquer par exemple que la pente de cette tangente prend la valeur 1 pour $r = 1$ avant de décroître pour $r > 1$. A cet endroit se produit une bifurcation transcritique. L'autre point fixe remarquable apparaît pour $r = 3$ où la pente de cette même tangente atteint la valeur critique -1 puis continue de décroître lorsque r augmente. Ce point est le point de la première bifurcation de doublement de période. Enfin la courbe pour $r = 4$ donne le comportement chaotique de l'application. Il se passe donc énormément de phénomènes dynamiques entre $r = 3$ et $r = 4$. Explicitons cela par un survol des différentes gammes de comportement numérique de l'application.

Domaine $0 < r < 1$

La figure 5.3 présente la construction de la série temporelle pour $r = 0,75$, sur la partie gauche 5.3(a) les itérations successives sont visualisées dans leur construction tandis qu'à droite 5.3(b), la série temporelle est tracée pour chaque pas de temps n. On constate ici que la suite atteint son son point fixe $\bar{x_1} = 0$. En effet, le calcul de la dérivée première de f aux deux

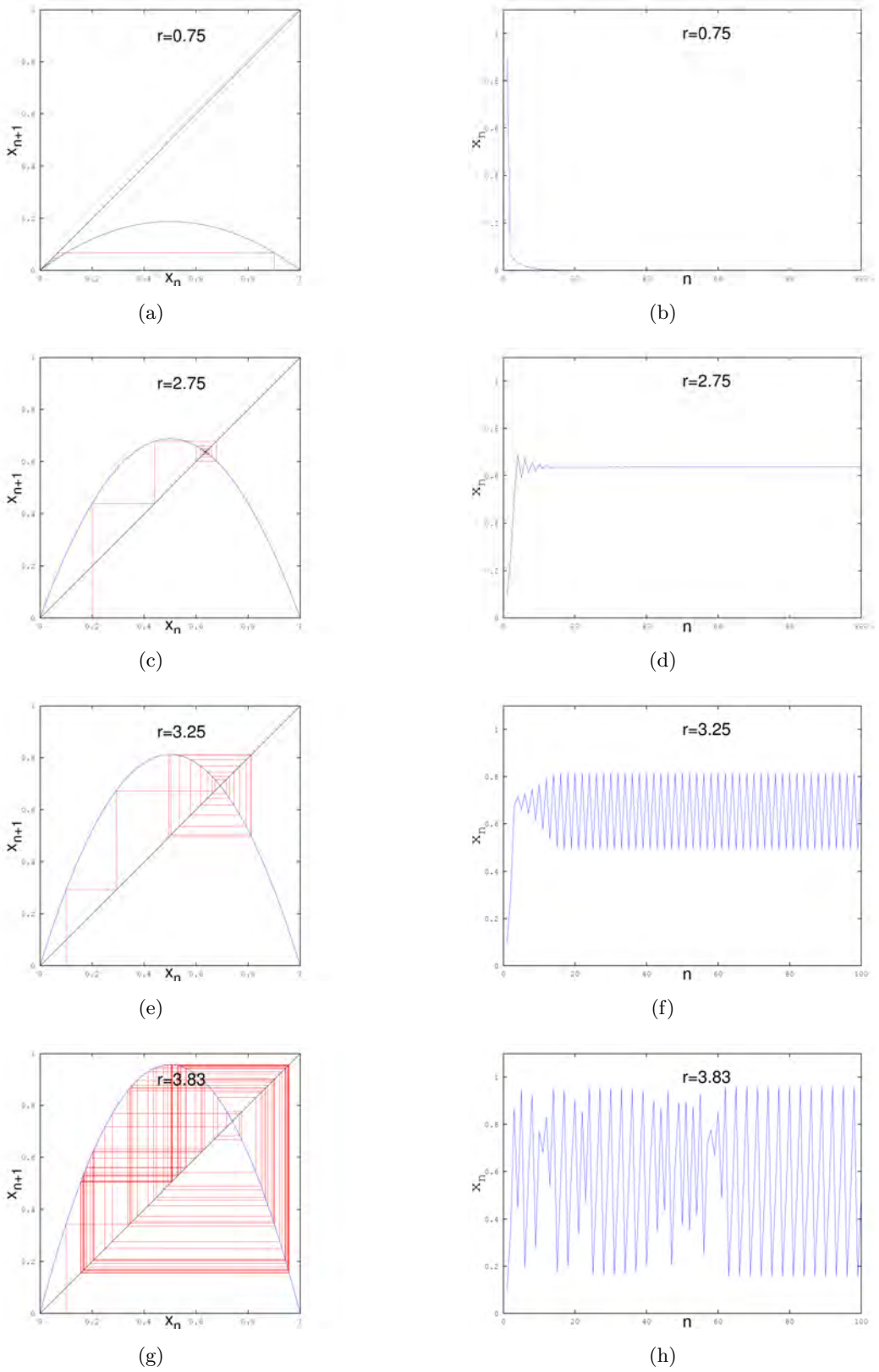


FIG. 5.3 – Applications de premier retour (à gauche) et séries temporelles associées (à droite) pour quelques valeurs du paramètre de contrôle de l'application logistique

points fixes de l'application donne sur ce domaine $|f'(\bar{x_1})| < 1$ et $|f'(\bar{x_2})| > 1$. En termes de stabilité cela signifie que $\bar{x_1}$ est un point fixe stable tandis que $\bar{x_2}$ est instable.
Lorsque $r = 1$, on a $|f'(\bar{x_1})| = |f'(\bar{x_2})| = 1$ et un changement de stabilité se produit.

Domaine $1 < r < 3$

Si l'on poursuit l'analyse sur f' précédente, on a maintenant la situation inverse suivante : $|f'(\bar{x_2})| < 1$ et $|f'(\bar{x_1})| > 1$. Les deux points fixes ont échangé leur stabilité. Nous sommes en présence d'un point de bifurcation transcritique en $r = 1$. La figure 5.3 confirme ce fait en montrant la série temporelle pour $r = 2,75$ qui se stabilise autour du point fixe $\bar{x} \simeq 0,63$. Un problème survient cependant lorsque $r = 3$ puisque le point fixe $\bar{x_2}$ devient indifférent : $|f'(\bar{x_2})| = 1$.

Domaine $3 < r < 3,56\cdots$ **: la cascade de doublements de périodes**

Désormais aucun des deux points fixes précédents n'est stable puisque $|f'(\bar{x_1})| > 1$ et $|f'(\bar{x_2})| > 1$ dans ce domaine de valeurs. Voyons numériquement ce qui se produit : on observe sur la figure 5.3(e) pour $r = 3,25$ la stabilisation de la suite de valeurs sur un carré ce qui signifie que la suite s'est stabilisée sur un cycle de période 2 comme le montre le graphe temporel 5.3(f). On est passé d'un mode avec un seul point fixe stable à un mode avec deux points fixes. Il s'agit en réalité pour le passage $r = 3$ d'un point de bifurcation par doublement de période deux nouveaux points fixes stables sont alors créés qui correspondent à un 2-cycle stable. Pour le comprendre, il faut analyser la courbe f^2 au voisinage de $r = 3$. La figure 5.4 montre

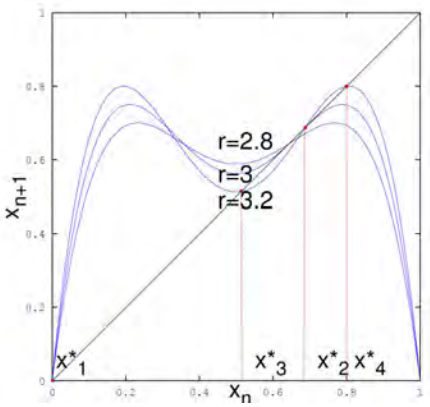


FIG. 5.4 – Tracé de $f(x)$ pour différentes valeurs du paramètre r, également appelée "application de premier retour". La tangente de la courbe pour $r = 3$ à l'intersection avec la première diagonale a une pente -1 : c'est le premier point de bifurcation de doublement de période.

trois courbes associées à $f^2(x) = f(f(x))$ pour trois valeurs autour de $r = 3$. Pour $r = 2.8$, seul le point fixe du domaine précédent est stable, à savoir le point intersection de la courbe f^2 avec la bissectrice $x_{n+1} = x_n$ possède une tangente inférieure à 1 en valeur absolue. L'égalité à 1 survient pour $r = 3$. C'est la situation $r = 3,2$ qui nous intéresse maintenant : les points

fixes de f^2 sont définis par l'équation $f^2(x) = x$. Il y en a quatre car on résout un polynôme de degré 4. Les quatre points fixes sont notés $x*_i, i = 1, \cdots, 4$ sur la figure 5.4, ils comportent les deux points fixes du domaine précédent qui sont instables et deux nouveaux points fixes qui sont stables pour f^2 puisque l'on a les relations suivantes :

$$|(f^2)'(x*_3)| < 1 \tag{5.7}$$

$$|(f^2)'(x*_4)| < 1 \tag{5.8}$$

L'application est sur un attracteur de type 2-cycle. On peut montrer que ce cycle est stable tant que $r < 1 + \sqrt{6} \simeq 3,449\cdots$ [Strogatz, 2001]. Au-delà il devient instable et on refait le même type de raisonnement sur f^4... Ainsi l'application logistique entre dans ce que l'on appelle une cascade de bifurcations par doublement de périodes. Appelons $r_i, i = 1, \cdots$ les valeurs pour lesquelles il y a bifurcation. Cette cascade se termine (à l'infini de dédoublement périodique) pour $r_\infty \simeq 3,56994\cdots$ obtenue numériquement notamment par Feigenbaum. Ce point est appelé point d'accumulation de la cascade de bifurcation. De plus on a la propriété suivante :

$$\delta = \lim_{n\to\infty} \frac{r_n - r_{n-1}}{r_{n+1} - r_n} = 4,669\cdots \tag{5.9}$$

δ est appelée constante de [Feigenbaum, 1978, Feigenbaum, 1980]. Cette constante semble être une constante "universelle" puisque l'on retrouve cette cascade de bifurcations et ce rapport constant dans de nombreux phénomènes réels : en mécanique des fluides, en électronique, en physique des lasers,en chimie et en biologie dans les réseaux de neurones (voir l'annexe à ce sujet).

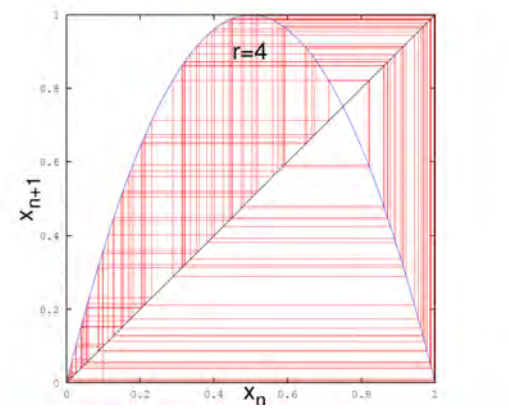
Le domaine $3,56\cdots < r < 4$ et la fenêtre de période 3

Au delà de la cascade précédente, l'application entre dans un domaine chaotique parsemé d'une infinité de fenêtres de périodicité. La plus importante de ces fenêtres est la fenêtre de période 3 car elle prouve l'existence du chaos pour cette application. Ce 3-cycle stable au milieu du chaos apparaît pour la valeur $r = 1 + \sqrt{8} \simeq 3,838\cdots$. Cette fenêtre de périodicité est importante puisqu'un théorème [Sarkovskii, 1964] montre que l'existence de cette périodicité dans les applications itérées de dimension 1 implique l'existence de cycles de toutes les périodes. Le théorème de Li et Yorke complète ce dernier en montrant l'existence de séries chaotiques. Ce théorème est résumé par la formule suivante : "période 3 implique chaos" [Li et York, 1975]. La série temporelle de la figure 5.3(h) montre le comportement de l'application dans cette zone et l'établissement de la périodicité 3. Le diagramme de bifurcation 5.6(f) montre clairement cette fenêtre de période 3 assez large.

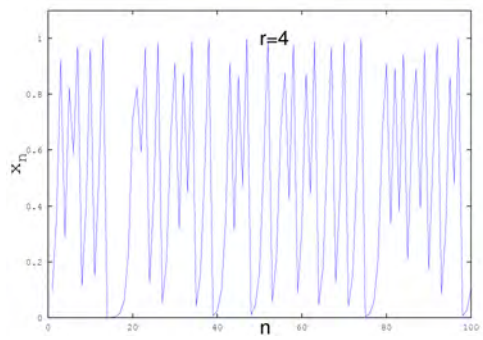
Le cas $r = 4$

Ce cas est le plus cité pour la génération du chaos avec l'application logistique. L'application avec cette valeur de paramètre a été étudiée par S. M. Ulam[35] sous la forme $x_{n+1} = 1 - 2x_n^2$ et proposée en tant que générateur de nombres aléatoires par Ulam et Von Neumann en 1947

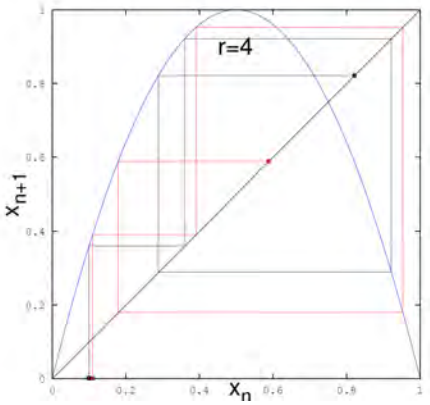
[35]Stanislaw Marcin Ulam (1909-1984) mathématicien polonais.



(a) Application itérée de premier retour.



(b) Série temporelle associée.



(c) Sensibilité aux conditions initiales après 5 itérations

FIG. 5.5 – Application logistique pour $r = 4$.

[Ulam et von Neumann, 1947]. Mais cet usage a été abandonné pour plusieurs raisons que nous évoquons en annexe B. La figure 5.5 révèle en effet une séquence apparemment aléatoire, autant sur le graphe temporel que sur l'application de premier retour. Les propriétés de l'application avec cette valeur de paramètre sont particulièrement remarquables et sont également présentées en annexe B. C'est pour la valeur $r = 4$ du paramètre de contrôle que l'application joue le rôle d'endomorphisme sur l'intervalle $[0, 1]$ puisque l'image de $[0, 1]$ est l'intervalle lui-même, ce qui n'est pas vrai pour les autres valeurs de r. En effet tout élément $x_n \in [0, 1], n > 1$ a deux antécédents :
$x_{n-1} = 0,5 \pm \sqrt{0,25 - \frac{x_n}{r}}$
ce qui n'est possible pour tout l'intervalle $[O, 1]$ seulement lorsque $r = 4$ (l'opérande de la racine carrée doit être positif). La section suivante confirme cette propriété grâce à la visualisation par diagramme de bifurcation.

5.3.2 Diagramme de bifurcation

L'ensemble de ces bifurcations est "résumé" dans un seul diagramme dit "diagramme de bifurcation". On reporte l'ensemble des points fixes $\bar{x}$ de l'application en fonction de la valeur du paramètre de contrôle r. Seuls les points fixes stables sont considérés. En pratique, il s'agit d'itérer un nombre de fois limité l'application logistique et d'afficher les points résultants. Pour illustrer cela graphiquement, la figure 5.6 présente l'évolution de ce diagramme en fonction du nombre d'itérations de l'application. En effet, la plupart des algorithmes présentés avec notre modèle d'agents logistiques procèdent par pas d'une seule itération. Le diagramme de bifurcation 5.6(f) est donc à considérer sur un plan qualitatif comme un diagramme limite lorsqu'aucune interaction d'un autre agent ne vient perturber le cours du calcul. D'autre part il est remarquable qu'après une seule itération de l'application, l'espace produit de l'espace des phases et de l'espace des paramètres, soit $[0, 1] \times [0, 4]$, soit réduit de moitié (cf. 5.6(a)). L'autre constat est que seule la valeur du paramètre $r = 4$ permet de conserver l'ensemble de l'intervalle $[0, 1]$ invariant.

5.3.3 Exposants de Lyapunov et chaos

Pour pouvoir affirmer qu'il existe des séquences chaotiques générées par l'application logistique, il faut pouvoir montrer la sensibilité aux conditions initiales caractéristique d'un comportement chaotique. Or cela signifie précisément que deux trajectoires initialement proches vont s'éloigner l'une de l'autre de façon exponentielle. Cela est montré sur la figure 5.5(c) où des trajectoires proches au départ se séparent rapidement après 5 itérations seulement. Pour quantifier cette séparation des trajectoires, on a recours à l'exposant de Lyapunov. Cet exposant mesure l'écart exponentiel de trajectoires initialement proches. En voici la définition (tirée de [Schuster et Just, 2005a, Dang-Vu et Delcarte, a]).

Définition 5.3.1. *Supposons deux points initiaux proches x_0 et $x_0 + \epsilon$. Après n itérations, l'exposant de Lyapunov est défini par l'équation :*

$$\epsilon e^{n\ \lambda(x_0)} = |f^n(x_0 + \epsilon) - f^n(x_0)| \tag{5.10}$$

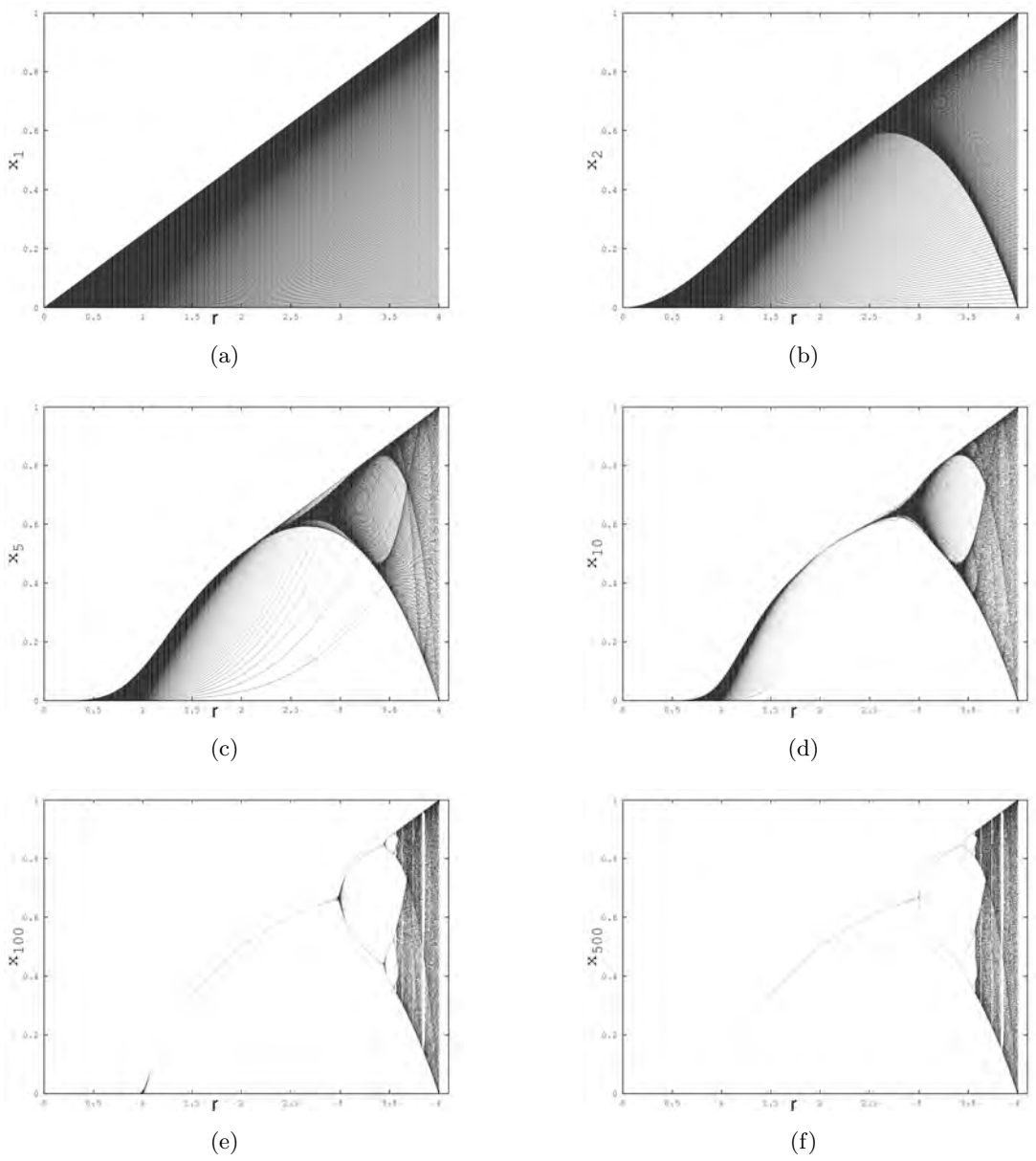


FIG. 5.6 – Diagramme de bifurcation de l'application logistique à différentes itérations

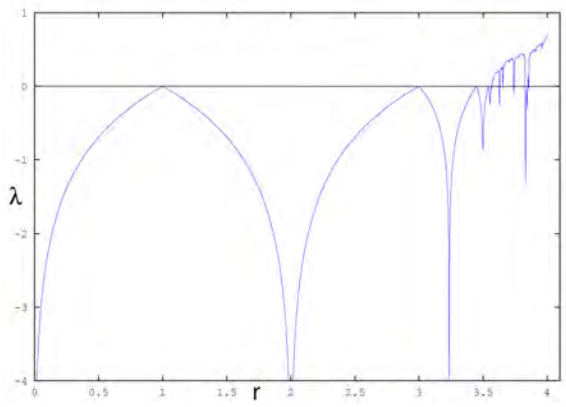


FIG. 5.7 – Estimation de l'exposant de Lyapunov pour l'application logistique en fonction du paramètre de contrôle.

A la limite $n \to \infty$ et $\epsilon \to 0$ on a l'expression finale du coefficient de Lyapunov :

$$\lambda(x_0) = \lim_{n\to\infty} \frac{1}{n} \log |\frac{df^n(x_0)}{dx}| \tag{5.11}$$

Dans le cas d'une application itérée le calcul de cet exposant se ramène en fait par dérivation en chaîne à la formule :

$$\lambda = \lim_{n\to\infty} \left(\frac{1}{n} \sum_{k=0}^{n-1} \ln |f'(x_k)| \right) \tag{5.12}$$

Selon la valeur calculée pour λ, on peut tirer l'une des conclusions suivantes :
- si $\lambda < 0$ alors l'attracteur de la dynamique (point fixe ou cycle) est stable
- si $\lambda = 0$ l'attracteur est indifférent
- si $\lambda > 0$ alors l'attracteur est chaotique.

La figure 5.7 montre le diagramme des exposants en fonction de r pour la fonction logistique. Le calcul effectué est approché : on a réalisé la moyenne sur 10000 itérations de l'application en enlevant du calcul les 500 premières pour éliminer les phases transitoires. La condition initiale pour x_0 est un tirage aléatoire et le calcul a été fait sur 1000 échantillons sur $r \in [0, 4]$. On reconnaît les principaux points de bifurcation où on observe un changement qualitatif de la dynamique par le fait que l'exposant s'annule en ces points, et les zones de stabilité, notamment la fenêtre de période 3 pour $r \simeq 3,82\cdots$ marquée par un pic vers les négatifs.
On calcule également la valeur de l'exposant en $r = 4$ numériquement évalué à $\lambda \simeq 0,69310$ alors que la valeur théorique est $\lambda = \ln 2 \simeq 0,69314$, ce qui constitue donc une bonne approximation. Cela confirme la chaoticité de l'application logistique pour ce paramètre.

5.3.4 Autres propriétés.

D'autres propriétés importantes de l'application logistique sont développées en annexe B :
- propriétés statistiques et densité de probabilité pour $r = 4$,
- constantes de Feigenbaum,

– universalité de la cascade de bifurcation par doublement de périodes.

L'application logistique et ses formes conjuguées nous procurent une algorithmique simple pour produire tous les phénomènes génériques présentés jusqu'ici, parfois hautement complexes. De plus, un seul paramètre suffit pour accéder à toutes ses propriétés dynamiques. C'est ce qui fait de ce type d'application une source riche de comportements pour les agents logistiques.

5.4 Réseaux d'itérations couplées : modèle archétypal de système complexe computationnel

5.4.1 Réseaux localement et globalement couplés

Les réseaux d'itérations couplées (RIC) élaborés par Kaneko [Kaneko, 1983, Kaneko, 1993] constituent un modèle de réseau pour l'étude des systèmes complexes spatialement étendus et spatio-temporellement chaotiques (les phénomènes de turbulence en mécanique des fluides par exemple). C'est un réseau de cellules chaotiques qui a en commun avec les automates cellulaires (AC) [von Neumann, 1966] une topologie de connexion en réseau et une dynamique entièrement déterministe, mais qui en diffère par la nature continue des espaces d'états cellulaires et les applications de transition d'états continues.

Types de Modèle	Espace	Temps	État
EDP (Équations aux Dérivées Partielles)	continu	continu	continu
RIC (Réseau d'Itérations Couplées)	discret	discret	continu
AC (Automate cellulaire)	discret	discret	discret

Un certain nombre de caractéristiques des RIC peuvent être énoncées mettant en avant les avantages de ce modèle par rapport aux deux autres apporches du tableau ci-dessus [Kaneko et Tsuda, 2001a] :

– Le modèle RIC est adapté au calcul sur machine, et notamment sur machines parallèles. Il permet donc d'explorer des phénoménologies dynamiques nouvelles.
– Le modèle est de conception simple et constructive (explicitement spécifié et déterministe), il peut s'appliquer a priori à l'étude de certains phénomènes expérimentaux en associant les grandeurs expérimentales aux variables et paramètres du modèle (surtout dans l'étude des systèmes physiques).
– Contrairement aux AC, le chaos est directement présent au niveau des cellules d'un RIC, non au niveau de la configuration globale émergente, car il est généré par les applications élémentaires chaotiques internes à ces cellules.
– De même contrairement aux AC, les RIC utilisent des variables, paramètres et applications continues qui entrent clairement dans le cadre d'analyse de la théorie des systèmes dynamiques non linéaires et autorisent donc l'utilisation des résultats existants de cette théorie.
– L'espace étant discrétisé pour ce modèle, on peut également utiliser les outils de la mécanique statistique.

Voici une formalisation du modèle RIC :

Définition 5.4.1. *On définit un réseau d'itérations couplées (RIC) à partir des données suivantes :*

- *Un graphe $G = (V, E)$ qui définit le réseau :*
  - *V est l'ensemble de noeuds du graphe, désignés par le terme de cellules ou sites. $V = \mathbb{Z}^n$ dans le cas d'un réseau n-dimensionnel infini et $V = \mathbb{Z}_L^n$ dans le cas d'un réseau n-dimensionnel torique, $\mathbb{Z}_L$ désignant l'ensemble des entiers modulo L.*
  - *E est l'ensemble des arêtes du graphe reliant les noeuds deux à deux (les deux noeuds ne sont pas forcément distincts : on prend ainsi en compte la cellule courante dans le voisinage).*
- *Un ensemble continu d'états Q qui est un (ou un produit) intervalle réel compact*[36] *sur lequel l'application est invariante (le plus couramment utilisé est : $Q = [0, 1]$). On note $x_i(t) \in Q$ la valeur de l'état de la cellule $i \in V$ à l'instant t.*
- *Un ensemble continu de paramètres P qui est également un intervalle réel compact donnant les valeurs possibles des paramètres.*
- *Une application de transition d'état locale f non linéaire paramétrée (exemple : l'application logistique présentée en section 5.3) :*
  $f : Q \times P \to Q$
  qui laisse Q invariant pour tout paramètre de P. On omet en général la présence du paramètre pour simplifier les écritures.
- *Un opérateur de couplage diffusif où U désigne comme pour les AC un voisinage identique pour toutes les cellules et ϵ désigne le coefficient de couplage diffusif du réseau :*

$$\begin{aligned} I_{\epsilon,U} : \quad & Q \times Q^{|U|} \to Q \\ & (u, v) \mapsto (1 - \epsilon)u + \frac{\varepsilon}{|U|} \sum_j v_j \end{aligned}$$

  $|U|$ est le cardinal fini ou infini de ce voisinage.

On aura affaire à deux grands types de RIC :
- *des RIC localement couplés où $|U| < \infty$*
- *des RIC globalement couplés où la valeur de $|U|$ dépendra alors de l'aspect fini ou infini du graphe G.*

Finalement un RIC se résume à l'équation de transition —ou de mise à jour— synchrone suivante lorsque $|U| < \infty$ (ce qui sera toujours le cas en pratique) :

$$x_i(t + 1) = (1 - \epsilon)f(x_i(t)) + \frac{\epsilon}{|U|} \sum_{j \in U} f(x_j(t)) \tag{5.13}$$

Dans le cas où $|U| \to \infty$, on parlera alors d'approximation de champ moyen et on remplacera le second membre de5.13 par la moyenne spatiale de f sur le réseau.

L'application f est en général une fonction simple : fonction polynomiale de degré au moins 2, fonction sinus, application tente (linéaire en deux morceaux), ... Les fonctions polynomiales de degré 2 sont les applications quadratiques ou applications logistiques utilisés dans cette thèse. Par exemple, on rencontrera utilisées la version classique de l'application logistique et une de ses conjuguées quadratiques :

$$x_i(t + 1) = 4\ a\ x_i(t)(1 - x_i(t)) \tag{5.14}$$

$$\text{ou la version conjuguée } x_i(t + 1) = 1 - a\ x_i(t)^2 \tag{5.15}$$

[36] fermé borné

cette dernière forme 5.19 est très employée par Kaneko, pour lesquelles a désigne le paramètre de contrôle de l'application. En fonction de la valeur de ce paramètre, l'application pourra exhiber toute une gamme de propriétés dynamiques résumées dans un diagramme de bifurcation. La caractéristique de cette application est de laisser invariant le domaine de définition de l'état cellulaire x quelque soit le paramètre de contrôle.

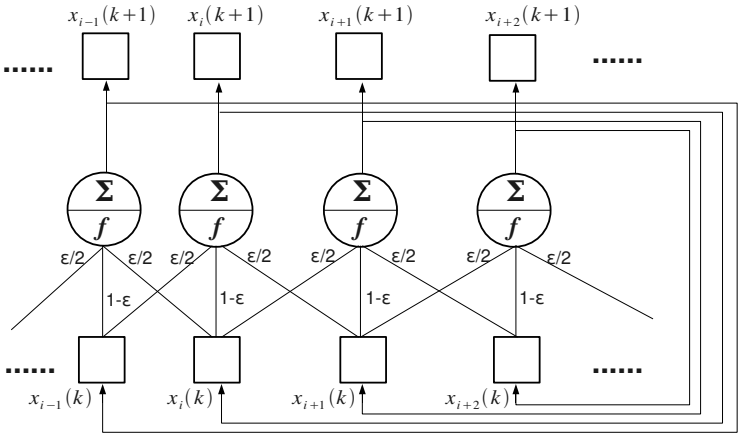


FIG. 5.8 – Schémas de principe d'un RIC localement couplé à une dimension.

Les deux cas d'un RIC localement couplé et globalement couplé[37] à une dimension lorsque $|U| < \infty$, dont le schéma de principe en version localement couplée est donnée en figure 5.8), évoluent selon les équations de transition respectives suivantes [Kaneko, 1990] :

$$x_i(t+1) = (1-\epsilon)f(x_i(t)) + \frac{\epsilon}{2}(f(x_{i-1}(t)) + f(x_{i+1}(t)) \tag{5.16}$$

$$x_i(t+1) = (1-\epsilon)f(x_i(t)) + \frac{\epsilon}{N}\sum_{j=1}^{N} f\left(x_j(t)\right) \tag{5.17}$$

où N est le nombre total de sites du réseau en couplage global. Les propriétés d'un réseau à couplage local du type 5.16 à une ou deux dimensions avec une application logistique ont été abondamment étudiées du point de vue dynamique ; cette version du réseau a des applications en mécanique des fluides, microélectronique, géophysique, écologie des populations, théorie de l'évolution entre autres ; une revue détaillée en est faite dans [Kaneko et Tsuda, 2001a].

La version globalement couplée du réseau 5.17 met en jeu des couplages à longue portée et correspond à une version "champ moyen" du modèle. Il est susceptible de fournir un modèle adapté au traitement de l'information biologique, où la capacité à mémoriser de l'information à travers les attracteurs du système est au coeur des réflexions. Cette hypothèse est sous-tendue par le type de phénoménologie des réseaux d'itérations globalement couplés centré sur la formation de groupes de cellules synchronisées que nous désignerons par le terme de "clusters" de

[37] On rencontrera ces deux types dans la littérature respectivement sous les sigles CML et GCM

synchronisation désormais dans cette thèse. Ce phénomène important est exposé dans ce qui suit.

5.4.2 Phénomènes de synchronisation

Synchronisation dans les réseaux globalement couplés.

La version réseau d'itérations globalement couplées (GCM) est décrite par l'équation maîtresse 5.17. Cette équation peut s'écrire sous une forme matricielle plus adaptée à l'étude analytique :

$$\mathbf{x}(t+1) = \left((1-\epsilon)\mathbf{I} + \frac{\epsilon}{N}\mathbf{G}\right) f(\mathbf{x}(t)) \tag{5.18}$$

où $\mathbf{x} = (x_1, x_2, \cdots, x_N)^T$ est le vecteur des états de toutes les cellules à l'instant t, $\mathbf{I}$ est la matrice identité $N \times N$, et $\mathbf{G}$ est la matrice de couplage $N \times N$ remplie de 1. La fonction utilisée pour l'étude de ce type de réseau est généralement l'application quadratique conjuguée de l'application logistique suivante :

$$x_{n+1} = f(x_n) = 1 - a\ x_n^2 \tag{5.19}$$

où $x \in [-1, 1]$ et $a \in [0, 2]$ (cf. figure 5.9).

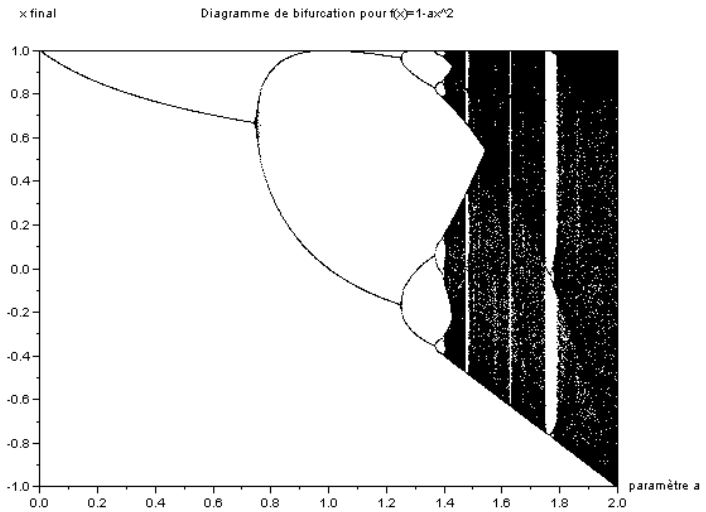


FIG. 5.9 – Diagramme de bifurcation de l'application itérée $x_{n+1} = 1 - a\ x_n^2$. Diagramme obtenu après 500 itérations de l'application.

Un tel réseau d'unités chaotiques globalement couplées exhibe une synchronisation totale lorsque les unités sont définies et paramétrées de façon identique sur tout le réseau (ϵ et a sont fixés de la même manière pour toutes les cellules), seulement dans une certaine plage de valeurs. C'est cette plage de valeurs que nous cherchons à déterminer théoriquement. On appelle seuil de synchronisation la valeur de ϵ va déclencher la synchronisation totale définie par :

Définition 5.4.2. *L'état de synchronisation totale est défini de la manière suivante :*
$\exists t_0$ *tq.* $\forall (i,j)\ \forall t > t_0\ x_i(t) = x_j(t) = x_*(t)$
On note $x_*(t)$ *la série temporelle synchrone.*

Cela ne signifie pas que les cellules adoptent une dynamique stationnaire mais qu'elles évoluent en phase à partir d'un pas de temps donné. Cet état de synchronisation totale n'apparaît pas pour toutes les valeurs de (a, ϵ).

Loi du seuil de synchronisation.

On peut montrer de manière analytique la loi d'évolution du seuil de synchronisation totale lorsque le couplage est symétrique (ce qui est le cas pour l'écriture standard 5.18). En fait, on cherche à déterminer pour quelles valeurs de (a, ϵ) l'état synchrone est stable. La stabilité de l'état synchrone dépend des valeurs propres de la matrice jacobienne relative à l'équation 5.18 en $\mathbf{x}_*$:

$$\mathbf{J} = f'(\mathbf{x}_*(t)) \left((1-\epsilon)\mathbf{I} + \frac{\epsilon}{N}\mathbf{G} \right) \tag{5.20}$$

La condition de stabilité s'écrit par le fait que les valeurs propres doivent se trouver dans le cercle unité du plan complexe. Les valeurs propres sont ici [Kaneko, 1990] :

- $\gamma = \exp^{\lambda_0}$ où λ_0 est l'exposant de Lyapunov (cf. section 5.3) de l'application f
- $(1-\epsilon)\gamma$ de dégénescence $N-1$

γ mesure la divergence dans la direction de l'état synchrone et les $N-1$ valeurs propres restantes mesurent la stabilité transversalement à l'état synchrone. C'est donc ces dernières qui indiquent la stabilité globale de l'état synchrone. En termes de coefficient de Lyapunov, l'équation de stabilité pour l'état totalement synchrone s'écrit donc à l'aide l'exposant de Lyapunov transverse :

$$\lambda_\perp = \lambda_0 + \ln(1-\epsilon) \tag{5.21}$$

$$\text{et } \lambda_\perp < 0 \tag{5.22}$$

Cela conduit à déteminer une valeur seuil pour ϵ, à savoir ϵ_* au-delà de laquelle la synchronisation totale est possible donnée par l'égalité :

$$\epsilon_* = 1 - \exp(-\lambda_0) \tag{5.23}$$

Analyse

Dans le cas de l'équation logistique 5.1 lorsque $r = 4$, ou pour la deuxième forme 5.19 lorsque $a = 2$, le coefficient de Lyapunov λ_0 est égal à $\ln 2$ (cf. par exemple [Schuster et Just, 2005b] pour une démonstration) et on a donc $\epsilon_* = \frac{1}{2}$. Cette courbe de seuil 5.23 est très irrégulière, à l'image du tracé de l'exposant simple de l'application logistique 5.7. Cependant on peut définir grossièrement une séparation en trois phases principales :

- une phase de synchonisation totale
- une phase de "clusters" : les "clusters sont des groupes de cellules totalement synchronisées connexes sur le réseau
- une phase turbulente où la synchronisation des cellules n'existe plus.

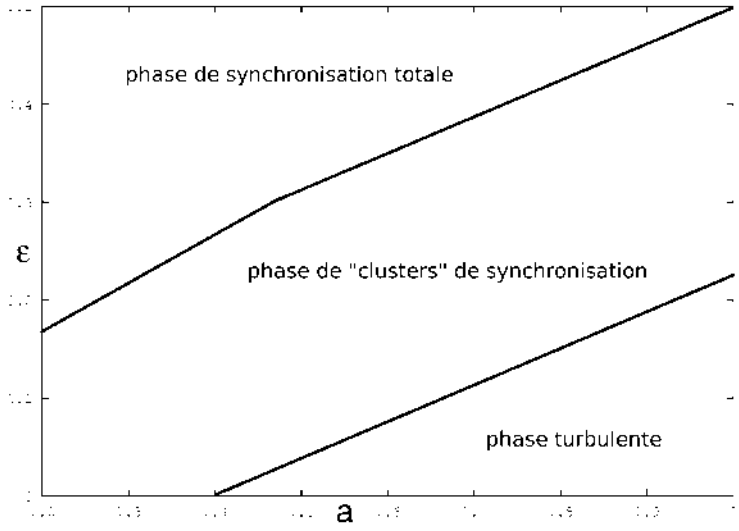


FIG. 5.10 – Diagramme de synchronisation de l'application itérée $x_{n+1} = 1 - a\ x_n^2$. D'après [Manrubia *et al.*, 2004].

Ces trois phases sont représentées sur la figure 5.10. La phase intermédiaire des "clusters" présente des groupes de cellules totalement synchronisées et parfois des celules passant alternativement d'un groupe à l'autre. Elle est particulièrement intéressante à cause de la complexité de structure très dépendante des conditions initiales, contrairement aux deux autres phases. Elle est considérée comme un phénomène émergent dans le réseau. Une version plus détaillée de ce diagramme avec une distinction fine des diverses phases présentes est disponible dans [Kaneko, 1990].

Enfin notons la spécificité de l'application logistique dans l'existence de ces clusters de synchronisation : en effet les mêmes simulations effectuées avec une application de type "tente" (deux morceaux linéaires en forme de tente) ne créent pas ce phénomène de groupements synchronisés [Kaneko, 1995]. Une conjecture établie par Cosenza et Gonzalez [Cosenza et Gonzalez, 1998] postule que l'application chaotique utilisée doit posséder certaines propriétés dans son diagramme de bifurcation pour générer des clusters synchronisés : ainsi, il serait nécessaire pour ce faire que l'application utilisée soit localement contractante et qu'il existe des fenêtres de périodicité au sein du domaine chaotique de cette application ; c'est précisément le cas pour l'application logistique contrairement à l'application tente.

Réseaux d'itérations aléatoirement couplés

Terminons cette revue susccincte des réseaux globalement couplés par la version randomisée du réseau globalement couplé. Cette version nous intéresse particulièrement car elle pose mathématiquement les bases pour aborder la dynamique des agents logistiques. Proposé par Manrubia et Mikhailov dans [Manrubia et Mikhailov, 1999], le réseau globalement couplé randomisé (RCM[38]) consiste à tirer aléatoirement des connexions de couplage dans le réseau de façon

[38]"Randomly Coupled Maps"

symétrique. Ce tirage est contrôlé par un paramètre κ qui est la connectivité moyenne du réseau : il définit la proportion de connexions à choisir sur le réseau correspondant au nombre est $\kappa\ (N-1)$. Plusieurs nouveaux phénomènes ou modifications apparaissent alors, toujours en utilisant l'application quadratique précédente, par rapport à la version d'origine du RIC global :

- La synchronisation entre cellules est maintenue jusqu'à près de 40% de connexions manquantes.
- La phase de synchronisation totale disparaît, laissant place à une phase de "clusters flous"[39] présentant le même type de comportements que les clusters précédents.
- La synchronisation est définie à une précision δ près :
  Définition 5.4.3. *les cellules i et j connexes sur le réseau sont synchronisées si $\exists t_0 \in \mathbb{N}$ et $\tau \in \mathbb{N}^*$ tq.*
  $\forall t,\ t_0 \leqslant t \leqslant t_0 + \tau, \quad \mid x_i(t) - x_j(t) \mid < \delta\ .$

  On peut ainsi déterminer des clusters à δ près.
- Les clusters ont une structure hiérarchique, c'est-à-dire que si δ augmente, les clusters fusionnent en des clusters plus grands.
- Les propriétés d'un RCM tendent vers celles du RIC global à la limite thermodynamique lorsque $N \gg 1$.

Intérêt du modèle

On constate donc ici l'importance des phénomènes de synchronisation cellulaire dès qu'un ensemble d'applications chaotiques sont couplées en réseau. Les phénomènes d'apparition de clusters notamment semblent une règle dès que la topologie des connexion devient aléatoire, ou que la population de cellules est hétérogène, ou encore lorsque les couplages ne sont plus symétriques, pour peu que l'on emploie les "bonnes" applications. Comme il a été mentionné plus haut, cette version randomisée du RIC global peut être utile à l'analyse des modèles à base d'essaim, l'essaim étant vu comme un ensemble de cellules libres de se déplacer sur le réseau. Précisément, une variante récente des RIC a été développée dans ce but : les gaz d'itérations couplées...

5.4.3 Le modèle de gaz d'itérations couplées

Le modèle de gaz d'itérations couplées (GIC)[40] est une adaptation du modèle de réseau d'itérations couplées (RIC) proposé par Shibata et Kaneko dans [Shibata et Kaneko, 2003], dans lequel les éléments chaotiques cellulaires dans le RIC deviennent capables de mobilité sur le réseau. Mais Kaneko et Shibata ne parlent pas pour autant d'agents, bien que l'état interne et l'autonomie du réseau le permette. Ceci dit, un environnement indépendant des entités chaotiques apparaît et porte un champ de forces influant de façon externe sur la dynamique des éléments chaotiques. Ainsi deux dynamiques interfèrent dans le système obtenu, une dynamique interne aux éléments chaotiques générée par des applications de type logistique et une dynamique

[39] "Fuzzy clusters"
[40] "Coupled Map Gas" (CMG) dans la littérature

externe causée par le champ de l'environnement. Le système est caractérisé par les grandeurs et définitions suivantes :

- Les éléments chaotiques possèdent un état interne scalaire x_k au temps k et un vecteur position $\vec{r}_k$.
- La loi d'évolution de l'état interne d'un élément est régie par l'application logistique associée au polynôme $f(x) = 1 - a\ x^2$.
- La dynamique d'un élément i est couplée à celle de ses voisins j dans un voisinage de rayon R :
  $\mathcal{N}(i) = j : |\vec{r}_n(i) - \vec{r}_n(j)| \leqslant R$
  De plus $N_n(i)$ représente le nombre de voisins dans le voisinage de l'élément i.
- L'interaction entre deux éléments s'exprime par une force qui dépend des états internes des éléments de la forme :
  $\mathcal{F}(x_n(i), x_n(j)) = F\ x_n(i)\ x_n(j)$

Tout ceci se résume dans les équations de transition suivantes [Shibata et Kaneko, 2003] pour une cellule mobile i :

$$x_{n+1}(i) = (1-\epsilon)f(x_n(i)) + \frac{\epsilon}{N_n(i)} \sum_{j \in \mathcal{N}(i)} f\left(x_j(t)\right) \tag{5.24}$$

$$\vec{r}_{n+1}(i) = \vec{r}_n(i) + \sum_{j \in \mathcal{N}(i)} \frac{\vec{r}_n(j) - \vec{r}_n(i)}{|\vec{r}_n(j) - \vec{r}_n(i)|} \mathcal{F}(x_n(i), x_n(j)) \tag{5.25}$$

L'interaction est ici clairement inspirée de forces centrales en $\frac{1}{d^2}$ présente dans de nombreux systèmes physiques. La phénoménologie résultante de ce modèle est mesurée au moyen du calcul d'un coefficient de diffusion et de la détermination de clusters d'élements, non totalement synchronisés. On peut ainsi distinguer plusieurs régimes globaux du système :

- Un régime de forte diffusion : des clusters d'éléments apparaissent et se divisent en permanence. Cela correspond à une phase dite "fluide".
- Un régime de faible diffusion : des clusters se forment et constituent une sorte de réseau de clusters ; les éléments transitent de l'un à l'autre de ces clusters. Cela correspond à une phase intermittente.
- Un régime sans diffusion : chaque élément appartient à un cluster particulier et y reste. Il peut correspondre à une phase dite "cohérente" résultant d'une synchronisation forte entre les éléments de chaque cluster. Mais il peut correspondre également à l'opposé à une phase désynchronisée où chaque cluster est séparé et constitué d'éléments non synchrones

Finalement l'article montre que les systèmes d'éléments chaotiques couplés, ici à couplage variable dans le temps, exhibent une diversité de structure gouvernée par quelques paramètres seulement et des propriétés émergentes difficilement prévisibles. Cet article de Shibata et Kaneko est le premier à modifier la topologie de connexion des cellules des modèles "rigides" des RIC. Il se trouve également que ces travaux n'ont pas connu de développements ultérieurs.

5.5 Entropie de Kolmogorov : mesure du degré de chaos d'un système

Le calcul de l'entropie dite de "Kolmogorov" ou d'une estimation de cette entropie, constitue un objectif important pour l'étude des phénomènes de synchronisation non linéaires (dans les RIC par exemple) et plus largement pour mesurer le degré de "chaoticité" d'un système complexe. Nous avons recours à cette mesure pour suivre la dynamique du système que nous manipulons dans cette thèse.

5.5.1 Définition de l'entropie K.

L'entropie de Kolmogorov (ou de Kolmogorov-Sinai) est la mesure qui caractérise le mieux le comportement chaotique d'un système quelle que soit sa dimension. De plus elle est reliée au spectre des coefficients de Lyapunov du système. Nous la désignerons sous le terme d'entropie K. Supposons un système dynamique dans un espace de dimension d dont la trajectoire est décrite par le vecteur $\vec{x} = (x_1(t), x_2(t), \cdots, x_d(t))^T$. Supposons de plus que l'espace des phases du système soit divisé en cellules de dimension l^d. Soit enfin $P_{i_0 i_1 \cdots i_n}$ la probabilité jointe que $\vec{x}$ soit dans i_0 à t, dans i_1 à $t+\tau$,..., dans i_n à $t+n\tau$, où τ désigne la période d'échantillonnage. Alors la grandeur :

$$K_n = -\sum_{i_0 i_1 \cdots i_n} P_{i_0 i_1 \cdots i_n} \log P_{i_0 i_1 \cdots i_n} \tag{5.26}$$

mesure l'information nécessaire au sens de Shannon pour localiser la trajectoire à la précision l près. La différence $K_{n+1} - K_n$ mesure notre perte d'information d'un pas de temps à l'autre.

Définition 5.5.1. *L'entropie de Kolmogorov K est définie par [Schuster et Just, 2005c] :*

$$K = \lim_{\tau \to 0} \lim_{l \to 0} \lim_{N \to \infty} \frac{1}{N\tau} \sum_{n=0}^{N-1} (K_{n+1} - K_n) \tag{5.27}$$

Dans le cas d'application où $\tau = 1$, la limite sur τ est omise.

5.5.2 Propriétés de l'entropie K.

L'entropie K a des propiétés essentielles [Schuster et Just, 2005c] :

- Elle mesure la perte moyenne d'information sur la dynamique du système.
- Elle est égale au coefficient de Lyapunov pour les applications monodimensionnelles. Pour les systèmes à plusieurs dimensions, l'entropie K mesure la déformation subie par une cellule de l'espace des phases par le système dynamique considéré. De plus pour des systèmes discrets, elle tend vers la somme des coefficients de Lyapunov positifs. Pour simplifier :

$$K \approx \sum_{j, \lambda_j > 0} \lambda_j \tag{5.28}$$

- Enfin elle mesure également le temps moyen de prédiction T_m sur le système de façon inversement proportionnelle. On a en effet la relation :

$$T_m \propto \frac{1}{K} \log(\frac{1}{l}) \tag{5.29}$$

Sur un calculateur, la grandeur l est relative à la précision de définition des nombres flottants utilisée. Ainsi en simple précision, $l \simeq 10^{-8}$ soit grossièrement $T_m \propto \dfrac{20}{K}$.

5.5.3 Calcul de l'entropie K.

Comme on l'a vu précédemment pour les systèmes complexes, les mesures de complexité proposées sont délicates à évaluer quantitativement. Nous rencontrons le même écueil avec l'entropie K, qui nécessite un calcul préalable du spectre de Lyapunov du système. Des techniques et algorithmes existent cependant pour pallier cette difficulté et déterminer le calcul approché de ces coefficients dans le cas de séries de données expérimentales (algorithme de Wolf [Wolf *et al.*, 1985] par exemple).

6

Synthèse

Les limitations identifiées au chapitre 3 dans le domaine de l'intelligence en essaim ont motivé à repenser les fondements des modèles de ce domaine en puisant successivement notre inspiration autour du concept d'auto-organisation, puis plus pragmatiquement dans la science des systèmes complexes qui offre une batterie de modèles computationnels spécifiques.

A travers une brève revue des conceptions sur l'auto-organisation, centrale pour l'intelligence en essaim, du point de vue de divers domaines scientifiques, nous avons montré la profondeur et la difficulté des questions posées. Depuis la cybernétique qui initia la réflexion sur les concepts clés, nous retenons des divers courants de pensée qui ont jalonné l'évolution des concepts, les axes de modélisation suivants :

- La pensée synergétique nous incite à considérer, à travers le principe d'asservissement et la causalité circulaire, l'importance des modèles à base de champs et des mécanismes à base de contrôle paramétrique par le champ.
- L'analyse biologique conduit à prendre en compte les mécanismes de boucles de rétroaction, de non linéarité, d'interactions directes et indirectes au sein des systèmes
- Enfin l'informatique vise de nouveaux algorithmes de calculs robustes, autonomes et capables de s'adapter en permanence de façon dynamique aux nouvelles configurations et données de l'environnement.

Nous nous tournons vers les systèmes complexes comme science d'étude de l'auto-organisation des systèmes, car la science des systèmes complexes dispose d'approches théoriques pour mesurer la quantité d'auto-organisation dans les systèmes, et de modèles de calcul comme les réseaux d'itérations couplés (RIC) qui permettent de répondre aux besoins de modélisation identifiés ci-dessus.

Les RIC constituent un cadre théorique fondé sur le chaos déterministe où la non linéarité est modélisée par une application de forme simple mais génère une large gamme de comportements non linéaires contrôlés par un paramètre. La mesure fondamentale ici est fondée sur le calcul des exposants de Lyapunov (qui donne également accès à l'entropie K). La notion de couplage y est également centrale car elle produit le phénomène émergent typique de ces réseaux qui porte le nom de "synchronisation". La synchronisation est quantifiable expérimentalement et prédictible théoriquement pour certaines configurations. La synchronisation est un concept majeur des systèmes dynamiques couplés, que l'on rencontre dans la modélisation de nombreux

systèmes complexes physiques ou biologiques. Un des objectifs de cette thèse est précisément d'utiliser cette synchronisation au niveau du modèle de calcul, et la traduire géométriquement dans l'environnement.

La démarche de modélisation développée dans cette thèse se situe de fait dans le cadre d'étude des systèmes complexes : nous nous intéressons aux outils de modélisation mathématiques typiques des systèmes complexes fondés à la fois sur des générateurs simples de phénomènes chaotiques paramétriques (réseaux d'itérations logistiques couplées) et à la fois sur un schéma de conception du type système multi-agent. A l'aide de ces outils, et par un principe d'adaptation/asservissement par le champ, nous focalisons notre étude sur deux aspects phénoménologiques distincts de l'intelligence en essaim. Si l'on reprend le graphe des domaines scientifiques 4.1, notre démarche de conception pourrait se résumer à un arc transverse reliant le noeud des systèmes multi-agent au noeud de la dynamique des systèmes complexes et au noeud de la théorie du chaos (à travers l'usage de l'application logistique).

Le gaz d'itérations couplées (GIC) décrit en section 5.4.3 est le point de départ pour notre proposition de modèle agent : le GIC nous semblent actuellement la modélisation de type "système complexe" à base d'applications non linéaires la plus aboutie pour aborder l'intelligence en essaim. Mais notre apport théorique réside également dans la conception d'un environnement à base de champs et d'interactions indirectes avec les agents logistiques. C'est de ces interactions par-et-avec le(s) champ(s) que résultent les mécanismes d'auto-organisation et d'adaptation dans notre approche. Notre proposition de système multi-agent logistique est conçue selon ces deux sources conceptuelles ; elle est entièrement formalisée dans la partie suivante.

Troisième partie

Modèle de SMA réactif pour l'intelligence en essaim : le SMA logistique

Cette partie de la thèse pose les bases de conception et la formalisation du SMA logistique. Elle propose de fusionner tous les éléments de conception théoriques retenus dans la partie précédente, en un seul système multi-agent.

Pour ce faire, nous précisons tout d'abord en section 7.1 ce qui se cache sous le concept agent et l'intérêt de travailler avec ce concept. Nous marquons à ce propos le distinguo entre la conception agent du physicien très proche du concept de particule évoqué déjà à maintes reprises dans ce document, et celle de l'informaticien issue de l'Intelligence Artificielle.

Pour ce qui concerne la modélisation du SMA logistique décrit dans cette partie, le modèle influence-réaction appliqué aux systèmes multi-agents (présenté en section 7.2) a servi d'architecture systémique, et de support de base à la formalisation mathématique. Nous en avons néanmoins simplifié la forme pour l'adapter à nos besoins de modélisation.

La formalisation propre au modèle du SMA logistique fait l'objet du chapitre 8.3. Nous proposons un schéma théorique global, original et cohérent au chapitre 8. Nous intégrons notamment le principe d'asservissement par le champ et le contrôle paramétrique de la synergétique comme mécanismes fondamentaux, moyennant certaines adaptations à notre formalisation : nous étendons en particulier la notion de champ à un cadre plus large que celui classiquement employé en physique, à savoir celui d'une structure de données générique qui compose la structure de l'environnement. Dès lors, les interactions s'exercent de façon systématiquement indirecte par couplage, et les phénomènes d'auto-organisation et d'adaptation sont rendus possibles par le flux d'informations via le(s) champ(s) de l'environnement.

Enfin nous terminons cette partie par un retour sur ce modèle sous forme de quelques justifications concernant l'emploi d'applications logistiques au coeur des mécanismes décisionnels des agents logistiques, au-delà de leur inspiration théorique issue des RIC. Nous montrons notamment que ce type d'application n'est pas seulement un générateur mathématique de phénomènes liés au chaos déterministe, mais aussi un modèle réduit possible du connexionisme neuronal biologique, dans certains cas spécifiques. Nous insinuons ainsi que cette application n'est pas qu'un "artifice" de calcul théorique, mais peut être reliée bel et bien à une phénoménologie biologique, ce qui constitue un argument expérimental supplémentaire en faveur de notre modèle.

7

Les approches orientées agent

7.1 Diverses conceptions de l'agent

7.1.1 De la particule à l'agent : l'agent du physicien

Évolution du concept de particule. La notion d'agent est de plus en plus utilisée par les physiciens, sans doute du fait de l'engouement croissant pour l'étude des systèmes complexes. Pourtant le modèle central en physique est traditionnellement celui de particule. Il nous semble intéressant ici de comprendre la transition entre ce modèle de particule et celui d'agent du point de vue de la physique, car la notion d'agent obtenue est bas niveau : on pourrait dire pour simplifier qu'il s'agit du niveau zéro de l'agent réactif. De plus, ce modèle sert de base aux systèmes particulaires employés en intelligence en essaim. Cette conception de l'agent-particule clarifie notamment le sens pris par le concept d'autonomie en physique.

On peut définir une particule en physique comme l'abstraction d'un objet matériel ponctuel associé au concept du point en mathématique. Ses variables d'état sont géométriques : la position $\mathbf{x}$ et la vitesse $\mathbf{v}$ de la particule sont repérés dans l'espace par un système de coordonnées. Les attributs intrinsèques d'une particule sont la masse ou la charge électrique qui permettent de quantifier les interactions subies par la particule dans son environnement baigné par un champ respectivement gravitationnel et électromagnétique. L'aspect particulaire réside dans sa loi d'évolution qui détermine sa position et sa vitesse future. Cette loi est exprimée par un système d'équations différentielles reliant directement les variables d'état à l'environnement. La particule modélise la matière inerte à son niveau microscopique.

L'évolution de cette notion de particule a été impulsée par les domaines de recherche à la frontière entre l'inerte et le vivant, les études de la matière condensée par exemple. Les objets d'étude sortent du domaine microscopique pour atteindre le domaine mésoscopique intermédiaire avec le macroscopique, tels les macromolécules ou les bactéries. On parle aujourd'hui du domaine d'étude de la "matière active" pour désigner ce secteur de recherche.

Sans quitter la formalisation mathématique attachée aux systèmes particulaires, la particule est devenue plus indépendante de son environnement en termes d'énergie. Ainsi apparaissent les concepts de particule "autopropulsée" [Czirók *et al.*, 1999] ou celui de particule "autoguidée"

comme dans le modèle de Vicsek 1.3.2. Ces deux nouvelles formes de particules sont mobiles mais l'origine énergétique de leur mouvement n'est pas la préoccupation principale. On s'intéresse davantage aux formes globales ou aux propriétés nouvelles qui peuvent émerger de ces ensembles de particules lorsque des interactions mutuelles sont définies entre la particule et son voisinage de particules.

Les agents browniens. Frank Schweitzer a développé le modèle des agents browniens, qui marque une avancée des physiciens dans le monde agent [Schweitzer, 2003a]. Ce modèle emprunte en effet clairement la terminologie agent et prend en compte la notion d'environnement. De plus le modèle vise à rendre compte des phénoménologies de l'intelligence en essaim biologique [Schweitzer *et al.*, 1997, Ebeling *et al.*, 1999, Ebeling et Schweitzer, 2003]. Le projet des agents browniens dérive des particules à mouvement brownien et se donne pour objet l'étude du mouvement d'agents biologiques simples.

La grande différence des agents browniens avec les particules du même nom réside dans leur capacité à extraire de l'énergie de leur environnement pour la stocker de façon interne, afin de la transformer ensuite en d'autres formes d'énergie (mécanique, chimique, ...). Ce qui caractérise l'agent brownien par rapport aux autres formes de modèles particulaires est qu'il possède un état propre, interne, régi par des lois d'évolutions internes [Ebeling *et al.*, 1999], ce qui lui donne en partie son caractère plus autonome comparativement aux particules. Son état interne comprend toujours la position et la vitesse de l'agent, plus d'autres attributs en fonction du contexte de modélisation, ainsi qu'une variable quantifiant son dépôt d'énergie interne. C'est en fonction de cette quantité que l'agent pourra ou non effectuer une action.

L'agent brownien peut également utiliser et transformer son environnement en y créant des champs chimiques par exemple. Les lois générales d'évolution de l'agent mêlent aspects déterministes et stochastiques, s'inspirant ainsi du modèle de Langevin du mouvement brownien. Le modèle d'interactions de l'agent brownien est également à mentionner car il est en fait très proche du nôtre. Les interactions sont de nature indirecte et transitent via un champ de communication scalaire multi-couches. Les différentes couches peuvent éventuellement échanger des données.

Nous renvoyons au chapitre 1 de l'ouvrage [Schweitzer, 2003b] pour la formalisation complète de ce modèle.

Pour finir, les agents browniens se sont révélés capables de simuler de nombreux phénomènes de déplacement biologiques (pistes de fourmis, flocking, déplacement de foules, agrégation,...).

Discussion. L'agent en physique est donc selon notre analyse lié au développement des études à la frontière entre l'inerte et le vivant. Son autonomie est relative à la fois à sa gestion interne de l'énergie exprimée par l'existence d'un état propre interne à l'agent, à la fois à sa capacité résultante d'interagir avec son voisinage par un champ de communication dans le cas des agents browniens ou par le champ moyen de vitesse dans le cas des particules autopropulsées.

7.1.2 L'agent de l'informaticien

Une autre vision de l'agent est celle adoptée en informatique qui part plutôt de réflexions autour de l'intelligence artificielle. Le premier à introduire l'agent en informatique semble être Allen Newell dans les années 80 [Newell, 1982], chercheur en intelligence artificielle et en psychologie cognitive. Cette première définition de l'agent intelligent a fait florès depuis lors, en initiant tout un champ de recherche informatique. Nous résumons dans cette section les grandes lignes de ce qu'on entend par agent en informatique, concept né au confluent de la robotique et de l'intelligence artificielle.

Une définition synthétique et générique de l'agent informatique.

Jacques Ferber donne dans son ouvrage sur les systèmes multi-agents [Ferber, 1995] la définition générique suivante de l'agent :

Définition 7.1.1. *On appelle agent une entité physique ou virtuelle :*
- *qui est capable d'agir dans un environnement,*
- *qui peut communiquer directement avec d'autres agents,*
- *qui est mu par un ensemble de tendances (sous la forme d'objectifs individuels ou d'une fonction de satisfaction, voire de survie, qu'elle cherche à optimiser),*
- *qui possède des ressources propres,*
- *qui est capable de percevoir (mais de manière limitée) son environnement,*
- *qui ne dispose que d'une représentation partielle de cet environnement (et éventuellement aucune),*
- *qui possède des compétences et offre des services,*
- *qui peut éventuellement se reproduire,*
- *dont le comportement tend à satisfaire ses objectifs, en tenant compte des ressources et des compétences dont elle dispose, et en fonction de sa perception, de ses représentations et des communications qu'elle reçoit.*

L'informatique distingue cependant deux grandes classes d'agents, les agents réactifs et les agents cognitifs.

Agents réactifs. Les agents dits réactifs sont caractérisés par :
- un mode de fonctionnement réactif basique de type stimulus-réponse,
- une absence de représentation interne de l'environnement,
- un mode d'interaction simple avec les autres agents et l'environnement,
- une organisation de groupe et une dynamique collective en modalité multi-agent qui en font un paradigme idéal pour l'étude des systèmes complexes.

Agents cognitifs. Un agent cognitif possède :
- une représentation élaborée de son environnement,
- des capacités de raisonnement sur les données dont il dispose,
- d'un but explicite à atteindre,

– d'une capacité à mémoriser ses actions passées, de planifier ses actions futures,
– de capacité de communication par un langage approprié[41].

On qualifie de rationnel un agent qui agit en fonction d'un objectif à atteindre.

Discussion. On pourrait dire grossièrement que les agents réactifs sont proches de l'agent du physicien ou du biologiste et que les agents cognitifs sont les héritiers des modèles d'intelligence artificielle façon systèmes experts. Cette segmentation a pour origine le type de modèle à la base du fonctionnement de l'agent, modèles logiques pour les agents cognitifs, modèles plus généralement dynamiques pour les agents réactifs.

Conception à base d'agent

Wooldridge et Jennings distinguent dans [Wooldridge et Jennings, 1995] :
– la théorie agent qui s'attache à la conceptualisation du domaine agent et aux formalismes mathématiques qui soutiennent les approches agents,
– l'architecture des agents qui regroupe les problématiques autour de la conception logicielle et matérielle des agents,
– les langages agents destinés à normaliser la communication entre agents.

Ces deux auteurs distinguent également la notion d'agent au sens faible et au sens fort. L'agent au sens faible reprend les caractéristiques mentionnées précédemment. L'agent au sens fort tend à rapprocher l'agent de plus en plus du fonctionnement humain. Ainsi on parlera de connaissance, de croyance, d'intention, d'obligation, d'émotions pour un agent alors que ce sont des notions habituellement attribuées à l'état mental de l'humain.

Enfin l'étude des phénomènes d'organisation collective et d'intelligence distribuée ont nécessité le recours à un autre type paradigme agent : les systèmes multi-agents.

7.2 Système multi-agent réactif

Les systèmes multi-agents.

Comme le rappellent Demazeau et Briot en introduction de [Briot et Demazeau, 2001], l'enjeu des systèmes multi-agent est l'organisation d'une part, l'autonomie d'autre part, et l'équilibre qui en résulte. L'auto-organisation est donc au coeur des problématiques multi-agents.

Ferber donne une définition des systèmes multi-agents dans [Ferber, 1995] et une spécification générique :

Définition 7.2.1. *On appelle système multi-agent (ou SMA), un système composé des éléments suivants :*
– *Un environnement E, c'est-à-dire un espace disposant généralement d'une métrique.*
– *Un ensemble d'objets O situés dans l'environnement (repérés par une position dans E). Ce sont des objets passifs c'est-à-dire qu'ils peuvent être perçus, créés, détruits et modifiés par les agents.*

[41] Agent Communication Langage

- *Un ensemble A d'agents, qui sont des objets particuliers représentant les entités actives du système.*
- *Un ensemble de relations R qui unissent les objets (et donc des agents) entre eux.*
- *Un ensemble d'opérations Op permettant aux agents de A de percevoir, produire, consommer, transformer et manipuler des objets de O.*
- *Des opérateurs chargés de représenter l'application de ces opérations et la réaction du monde à cette tentative de modification, que l'on appellera les lois de l'univers.*

Cette définition générique est précisée par le modèle Influence-Réaction qui propose une formalisation mathématique pour décrire les différentes parties du système et leurs relations. Nous présentons ce modèle dans ce qui suit car il constitue la base systémique du SMA logistique.

Le modèle influence-réaction : vision systémique du sma

Le modèle influence-réaction a été développé dans un premier temps par Jacques Ferber dans [Ferber, 1995] puis formalisé de façon générale avec J.-P. Müller [Ferber et Müller, 1996]. Nous interprétons ce modèle comme une formalisation des systèmes multi-agents situés sous forme d'un système dynamique très générique. Ce modèle décrit la relation entre un environnement et des agents situés dans cet environnement ; il rend compte de la boucle globale de rétro-action du système. Les concepts clés d'influence et de réaction sont spécifiques à l'approche multi-agent, mais héritent clairement de la pensée systémique qui vise à considérer les interactions entre les parties du système[42]. Dans le modèle influence-réaction, l'expression est entièrement basée sur des fonctions d'évolution des grandeurs d'état, excepté un opérateur qui combine les influences des agents. Le formalisme utilisé est celui des systèmes dynamiques discrets. Le modèle influence-réaction est l'écriture de la dynamique couplée des agents avec leur environnement. Plus précisément selon ce modèle [Ferber et Müller, 1996, Ferber, 1997], un SMA situé se définit comme le couplage de deux systèmes dynamiques, l'un relatif au fonctionnement des agents et l'autre relatif à l'environnement, le couplage étant assuré lors des phases de perception et d'action des agents. Le tout constitue un seul système qui décrit le bouclage dynamique de l'ensemble { environnement, agents }.

Formulation du modèle influence-réaction

Nous reprenons ici les notations de l'article de Ferber [Ferber, 1997] en posant que le système multi-agent est un triplet $\langle A, w, \Pi \rangle$ où :

- $A = \{a_1, \ldots, a_i, \ldots, a_N\}$ est l'ensemble des N agents,
- w le monde dans lequel évoluent les agents,
- Π un opérateur de combinaison d'influences des agents.

w est défini par la donnée de $\langle E, \Gamma, \Sigma, R \rangle$ où

- E est l'espace métrique dans lequel évoluent les agents. Il représente souvent le modèle de l'espace physique dans lequel se déplacent les agents.
- Γ est l'ensemble des influences (actions) possibles des agents sur le monde

[42] On pourra consulter sur ces questions l'essai de Robert Vallée [Vallée, 1996], cybernéticien de la première heure

- Σ est l'ensemble des états de E, l'état au pas de temps t est renvoyé par la fonction $\sigma(t)$.
- R décrit l'évolution du monde c'est-à-dire la réaction dont nous écrirons la formulation plus loin :
  $R : \Sigma \times \Gamma \rightarrow \Sigma$

Décrivons maintenant un agent $a \in A$ quelconque : il est caractérisé par le tuple suivant :
$\langle P_a, Percept_a, F_a, Infl_a, S_a \rangle$
où :

- P_a est la fonction de perception de l'agent et $Percept_a$ est l'ensemble des stimuli/percepts possibles observables

$$P_a : \Sigma \rightarrow Percept_a$$

- F_a et $Infl_a$ sont respectivement la fonction de comportement interne de l'agent a et sa fonction d'influence sur le monde (l'agent modifie à la fois son intérieur et son extérieur)
  $F_a : S_a \times Percept_a \rightarrow S_a$
  et $Infl_a : S_a \rightarrow \Gamma$
- S_a correspond à l'ensemble des états internes de l'agent dont $s_a(t)$ renvoie à tout instant l'état courant.

Considérons dans un premier temps un seul agent en interaction avec son environnment (cf. figure 7.1). Les lois d'évolution du système (agent + environnement) prennent la formulation suivante :

$$\begin{cases} s_a(t+1) & = F_a(s_a(t), P_a(\sigma(t))) \\ \sigma(t+1) & = R(\sigma(t), Infl_a(s_a(t))) \end{cases} \tag{7.1}$$

La première équation regroupe les phases de perception et décision de l'agent. La seconde regroupe les phases d'influence par l'action de l'agent et de réaction de l'environnement/monde. En fait dans ces deux étapes, il s'agit à chaque fois d'une composition des phases de calcul. Cette formulation correspond à l'écriture d'un système dynamique puisqu'il fait correspondre des grandeurs prises au temps t dans le membre de droite, aux mêmes grandeurs au temps $t+1$ à gauche. Ce qui permet de résumer le tout par un système global du type :

$$\begin{pmatrix} s_a(t+1) \\ \sigma(t+1) \end{pmatrix} = F \begin{pmatrix} s_a(t) \\ \sigma(t) \end{pmatrix} \tag{7.2}$$

En situation multi-agent, l'opérateur de combinaison des influences des agents joue pleinement son rôle. Il produit l'action conjointe des agents sur le monde. Nous obtenons maintenant le système dynamique suivant [Ferber, 1997] :

$$\begin{cases} s_i(t+1) & = F_i(s_i(t), P_i(\sigma(t))) \text{ pour } i \in A \\ \sigma(t+1) & = R(\sigma(t), \Pi_{i \in A} Infl_i(s_i(t))) \end{cases} \tag{7.3}$$

Notons ici que l'indice i des agents sur les fonctions de comportements et de perception laisse la possibilité d'agents hétérogènes, dont les capacités perceptives, motrices ou décisionnelles peuvent différer d'un agent à l'autre.

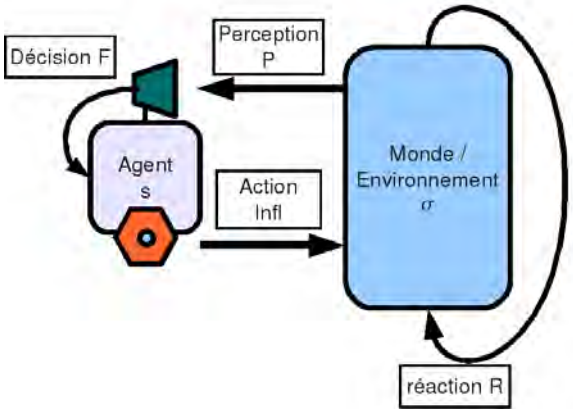


FIG. 7.1 – Schéma de la boucle globale de rétro-action agent-environnement "perception-décision-influence-réaction".

Commentaire.

L'originalité de l'approche est d'ajouter au triptyque sensorimoteur perception-décision-action centré sur l'agent, la réaction centrée sur l'environnement. Ainsi, il faudrait ici parler d'une boucle systémique globale **perception-décision-influence-réaction** qui intègre explicitement l'évolution de l'environnement. De plus la présence d'un "opérateur" d'influences fusionne une pluralité de fonctions d'influences en une seule. La formalisation de l'approche influence-réaction continue d'être développée et raffinée dans la communauté SMA : on pourra se reporter notamment aux articles [Weyns et Holvoet, 2004] et [Michel, 2007], ou encore à la thèse [Simonin, 2001].

Cette formalisation sert de base à la formalisation mathématique du SMA logistique.

8

Formalisation du SMA logistique

8.1 Principes de conception

Nous présentons dans cette section introductive les principes de conception de notre modèle de système multi-agent logistique (SMAL). Tous les éléments nécessaires à la compréhension de la genèse de ce modèle ont maintenant été abordés dans les chapitres et parties précédentes. Nous proposons à présent d'en fusionner les divers aspects pour en donner une image cohérente.

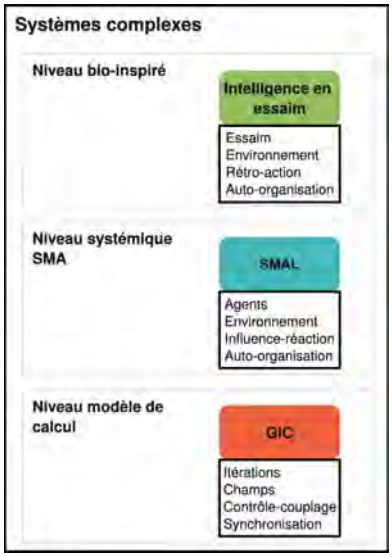


Fig. 8.1 – Représentation schématique des divers niveaux de conception : intelligence en essaim, système mulit-agent logistique (SMAL), gaz d'itérations couplées (GIC)

Conception.

La figure 8.1 présente les différents niveaux de conception que l'on peut distinguer dans ce travail :

- Au niveau de la source d'inspiration, qui correspond aux domaine phénoménologique que l'on souhaite prendre comme point de mire, la bio-inspiration s'avère principale pour l'intelligence en essaim, mais la phénoménologie des systèmes complexes de façon plus large est une source d'inspiration (cf. section 5.2). Ainsi comme on l'a décrit en première partie de thèse, les termes importants qui nous intéressent ici sont l'essaim, l'environnement et le bouclage de rétroaction qui mène à l'auto-organisation.
- Notre modèle d'architecture informatique est celui du système multi-agent. Il se place au niveau systémique (cf. section 4.1), dans le sens où il décrit comment le système étudié peut se segmenter en différentes parties. Ces parties du système sont d'une part les agents, d'autre part l'environnement et les relations entre ces parties sont décrites par le modèle générique influence-réaction qui réalise une boucle de rétroaction. A ce niveau d'observation, on pourra parler également d'auto-organisation du système par influence-réaction.
- Enfin au niveau du modèle mathématique de calcul employé, il s'agit d'utiliser les itérations chaotiques couplées dans une configuration de type gaz (cf. section 5.4.3). A ce niveau, on utilisera les itérations comme mécanismes de décision, des couches de champs (dérivés du sens mathématique) comme structures de données du système pour caractériser l'espace d'évolution, des paramètres de contrôle et de couplage qui produisent de la synchronisation comme phénomène émergeant du calcul.

Ces différents niveaux de modélisation appartiennent tous au domaine des systèmes complexes, cependant on les hiérarchise ici puisque les réseaux d'itération viennent s'intégrer dans un système multi-agent qui lui-même vise à modéliser des phénomènes bio-inspirés.

Importance de la notion de champ.

La structure de l'environnement en couches de champs est à l'origine d'une partie importante des mécanismes auto-organisationnels et adaptatifs du SMAL. Le circuit des informations dans le SMAL utilise en effet les champs de l'environnement du niveau modèle de calcul, comme sources de modification indirecte de l'état interne des agents : plus précisément le champ peut "asservir" l'agent par le biais indirect de son paramètre de contrôle ou de son paramètre de couplage. C'est ainsi que nous traduisons le principe d'asservissement par le champ de la synergétique (cf. section 4.2) dans notre conception. Cet asservissement est ici opéré au moyen des perceptions au niveau local des agents. La modification des champs par l'agent est réalisée par ses actions. L'environnement dans notre modèle est en fait stratifié en un ensemble de champs qui sont les vecteurs de toutes les interactions indirectes entre agents.

Ce survol des principes de conception et de leur lien avec les parties précédentes nous permet d'en donner à présent la formalisation mathématique complète.

8.2 Spécifications et formalisation du SMAL

Comme le stipule Demazeau dans son approche “voyelle“ [Demazeau, 1997], nous définissons le SMA Logistique selon la liste de spécifications suivantes :

- l'organisation (selon le modèle influence-réaction)
- l'environnement (à base de champs)
- les interactions (indirectes)
- les agents (logistiques)

Nous abordons la définition de toutes ces notions dans le cadre des systèmes dynamiques discrets et nous précisons autant que possible ces notions mathématiquement. Cette section se veut assez formalisée car elle pose les bases les plus génériques possibles de notre modèle de SMA logistique.

8.2.1 Définition du système multi-agent logistique

Tout d'abord définissons l'organisation globale et les parties distinctes du SMA logistique. Nous reprenons la définition générale de système multi-agent propre au modèle influence-réaction de Ferber (cf. section 7.2) tout en implémentant le modèle de nos spécifications. Rappelons que ce modèle est cohérent avec une approche de type ”bottom-up“ et émergentiste que nous souhaitons développer dans le cadre de cette thèse, approche rendue nécessaire pour étudier les phénomènes émergents des systèmes complexes.

Nous reprenons également la terminologie d'opérateurs employée pour les processus sensorimoteurs par Vallée dans [Vallée, 1951], en particulier pour le processus de perception et le processus d'action/influence des agents. La notion d'opérateur étant rattachée à une application a priori linéaire d'un espace vectoriel dans un autre, elle ne convient pas pour la partie décisionnelle de nos agents logistiques qui est intrinsèquement non linéaire.

Enfin notons que le terme de boucle dans le processus perception-décision-action n'est plus approprié si on ne prend pas en compte la dernière phase de ”réaction“ de l'environnement réalisée sous l'influence conjointe des agents. Nous emploierons simplement le terme de processus pour désigner ce triplet d'étapes computationnelle.

Définition du système.

Définition 8.2.1. *Le système multi-agent logistique est un système multi-agent réactif basiquement constitué du point de vue systémique (à temps discret) :*

- *d'un ensemble $\mathbb{A}$ de N agents logistiques $A_i, i = 1, \cdots, N$ caractérisés chacun par un état $s_i(t)$ à l'instant t discret,*
- *d'un environnement (ou monde dans la terminologie influence-réaction) $\mathbb{W}$ caractérisé par l'état $\sigma(t)$ à l'instant t*

De plus, il existe :

- *des lois d'évolution des états des agents désignés par des applications de transition de plusieurs variables F_i où l'indice i identifie l'agent considéré,*
- *d'une loi d'évolution de l'état de l'environnement appelée ”réaction“ désignée par l'application R,*

- *d'un opérateur de perception P_i associé à l'agent i,*
- *d'un opérateur d'action appelé influence I_i de l'agent i sur le monde,*
- *d'un opérateur d'influences combinant les influences de l'ensemble des agents sur le monde noté Π,*

tel que l'on puisse écrire la dynamique globale du système sous la forme d'un système dynamique à temps discret[43] *:*

$$\begin{cases} s_i^{t+1} = F_i\left(s_i^t, P_i(\sigma^t)\right) & \textit{pour tout agent } A_i \in \mathbb{A} \\ \sigma^{t+1} = R\left(\sigma^t, \Pi_{i\in A}\ I_i(s_i^t)\right) \end{cases} \tag{8.1}$$

Dans cette définition de l'évolution globale du système tirée du modèle influence-réaction, nous pouvons pointer certaines hypothèses de modélisation :

- Nous faisons implicitement l'hypothèse markovienne consistant à dire que l'état futur du système à $t+1$ ne dépend que de son état présent à t.
- Le système dynamique 8.1 exprime une causalité circulaire, fondamentale dans les approches systémiques et cybernétiques, causalité responsable des phénomènes d'adaptation agent-environnement du système ainsi que de l'auto-organisation du système.
- Enfin à l'image des automates cellulaires, le modèle ainsi posé est par essence synchrone, son écriture mathématique sous la forme d'un système dynamique à temps discret l'implique. Cependant ce caractère synchrone n'est pas incontournable puisqu'il va dépendre essentiellement de la façon dont la mise à jour de l'environnement et des agents se réalise. Nous n'abordons dans cette thèse que l'aspect synchrone du SMAL.

Cette écriture générique étant posée, précisons comment sont structurés ces différents éléments et relations mathématiques.

La modélisation des interactions

Modèle d'interactions indirectes. Nous abordons dès à présent le type d'interaction présent dans le modèle SMAL. Rappelons tout d'abord qu'une interaction indirecte est médiée par l'environnement tandis qu'une interaction directe est avant tout dirigée —donc nécessite de connaître l'identité du receveur— et transmise sans recours à l'environnement. Malgré cette différence de définition a priori, nous considérons que cette distinction tient davantage à un problème de modélisation ou d'interprétation qu'à une différence profonde liée à la nature de l'interaction. En d'autres termes, nous partons de l'hypothèse de travail qu'il est toutjours possible de traduire une interaction dite directe par une interaction indirecte médiée par l'environnement. Le phénomène stigmergique par exemple chez les fourmis est un type d'interaction indirecte qui procède par dépôt de phéromones dans l'environnement, et on sait par ailleurs que les fourmis procèdent également par interaction directe de contact. Ceci dit cette dernière interaction peut être facilement décrite par l'intermédiaire d'un champ qui stockerait "l'identité" de la fourmi à partir de laquelle une mesure de proximité est réalisable. Le recours à des mémoires ou des tables de stockage en informatique permet de réaliser des interactions apparemment directes —comme

[43] le temps est placé en indice haut ici pour alléger les écritures et nous adopterons cette notation pour le reste de ce document, sauf cas exceptionnel.

connecter une machine distante par exemple— qui s'avère impossible si un système d'adressage global des réseaux (et donc de référencement et de stockage) n'est pas établi.

Par conséquent toute interaction dans notre modèle se veut indirecte dans son mécanisme de réalisation, c'est-à-dire qu'elle peut être définie à travers des champs de l'environnement (définis en section 8.2.3) qui stockent de façon temporaire ou persistante l'information nécessaire à la réalisation de l'interaction. Ce choix est justifié par le fait que les agents logistiques sont construits selon un schéma invariant et non discernable d'un agent à l'autre, ce schéma est approprié à la notion d'essaim dont la robustesse tient précisément au caractère interchangeable des agents. En pratique, même si un identifiant est attribué à chaque agent logistique de façon implicite, cette information n'est pas exploitée dans l'interaction. Enfin les interactions directes n'ont pas été un concept nécessaire dans les simulations ou les problèmes que nous avons abordés.

Types d'interactions indirectes. Nous traiterons principalement dans nos cas d'études de deux types d'interactions indirectes :

- Des interactions "agent-environnement" de nature stigmergique dont le mécanisme est le dépôt et la perception d'une information dans un même champ de l'environnement. L'archétype biologique de ce type d'interaction est le recours à la phéromone comme support de l'information dans les sociétés d'insectes. La traduction de cette phéromone en intelligence en essaim informatique est la phéromone digitale, concept abstrait lié aux opérations de dépôt et de lecture d'une information numérique dans l'environnement informatique du problème. Ce type d'interaction manipule des champs persistants, ou bien dont les données "s'évaporent" (cette notion sera précisée mathématiquement en section 8.2.4).
- Des interactions interindividuelles "agent-agent" qui pourraient être considérées comme interactions directes, mais que nous traitons comme interaction indirecte en les faisant transiter également par l'environnement. Les champs impliqués dans ce type d'interaction sont non persistants au sens défini en section 8.2.4, mais possèdent le même mécanisme de lecture-écriture que dans l'interaction "agent-environnement". La non-persistance des données permet de marquer la position d'un agent à chaque pas de temps dans un champ scalaire de l'environnement sans mémoriser sa trajectoire.

Notons pour finir que les interactions interindividuelles entre agents prennent la forme mathématique d'un couplage de paramètre ϵ dans notre modèle.

8.2.2 Agents logistiques

Avant de préciser les mécanismes principaux de boucles sensorimotrices, définissons l'agent logistique et ses variables d'état.

Définition de l'agent logistique

L'agent logistique est une encapsulation "agent" du concept d'unité mobile des gaz d'itérations couplées (cf. 5.4.3). Ainsi la variable x devient-elle une variable de décision dans le cadre agent, et les deux paramètres (a, ϵ) prennent le statut de variables d'état de l'agent car ils sont

susceptibles d'être modifiés par les champs de l'environnement au cours d'un processus d'adaptation. En découle la définition de l'agent logistique dans le cas monodimensionnel :

Définition 8.2.2. *Un agent logistique est un agent à mémoire interne qui se caractérise par son état propre s stocké dans sa mémoire et par une application logistique f. Il se définit comme un agent réactif dont la variable de décision x est gouvernée par l'application logistique f (ou une de ses formes conjuguées) de paramètre de contrôle a, et qui dépend des autres agents par un couplage de coefficient ϵ. On caractérise l'état s d'un agent logistique par le tuple :*

$$s = \langle x, a, \epsilon \rangle \tag{8.2}$$

où :

- *x désigne la variable de décision de l'agent. Elle lui permet de déterminer quelles actions doivent être effectuées à l'instant suivant. Cette variable correspond au centre de commande, qui centralise toutes les informations provenant à l'agent. Le lien entre les réseaux de neurones et l'application logistique évoqué en annexe C.2 renforce cette analogie.*
- *a désigne la variable de contrôle de l'agent. Elle détermine le profil dynamique interne de l'agent à travers son rôle de paramètre de contrôle dans l'application logistique. Elle peut être vue également comme une variable "d'excitation" de l'agent qui gouverne la régularité des réponses décisionnelles de celui-ci.*
- *ϵ désigne la variable de couplage de l'agent avec le reste du système, environnement et autres agents. Elle quantifie de façon réductrice le caractère plus ou moins sociable de l'agent, c'est-à-dire sa propension plus ou moins forte à aligner son comportement sur les agents qui l'entourent.*

Ces trois variables prennent leurs valeurs dans un espace de définition $\mathbb{D} \subset \mathbb{R}$.

Dans toute cette thèse $\mathbb{D} = [0, 1]$.

Grandeur	Application logistique / RIC	Agent logistique
x	variable principale	variable décisionnelle / centre de commande
a	paramètre de contrôle	variable de contrôle / excitabilité de l'agent
ϵ	paramètre de couplage	variable de couplage / sociabilité de l'agent

Tab. 8.1 – Interprétation des variables d'état interne de l'agent logistique pour le modèle de calcul RIC et pour l'agent.

Extension à des espaces de plus grandes dimensions.

Nous venons de décrire le cas monodimensionnel, le plus "simple" qui puisse se présenter pour des agents logistiques. La plupart de nos expérimentations ne vont pas au-delà. Mais le nombre de degrés de libertés dans la décision de l'agent est relié à la dimension de l'environnement et du type d'action réalisable. Il est donc parfois nécessaire d'augmenter la dimension de l'espace de décision et par conséquent de l'état interne des agents en définissant de façon plus générale le domaine $\mathbb{D} = [0, 1]^k$ avec $k \in \mathbb{N}$. La section qui suit aborde plus précisément la structure de l'environnement dans notre modèle.

8.2.3 Modèle de l'environnement

L'outil conceptuel de champ est central dans le domaine de la physique et possède une définition générique en mathématique, mais nous estimons qu'il est central également dans le domaine des phénomènes biologiques et peut donc trouver sa place dans la bio-inspiration informatique comme structure de données. En particulier, il s'intègre selon nous parfaitement à la notion d'environnement. Nous prenons donc ce concept de champ dans son sens mathématique le plus large et pour une utilisation la plus générique possible de notre modèle.

Définition.

Commençons par définir ce qu'est l'environnement dans notre modèle, parce qu'il induit la définition des opérateurs de perception et d'action des agents. Son importance est mis en exergue dans l'article [Weyns *et al.*, 2004]. Nous nous situons d'ailleurs dans un courant de modélisation qui tente de concilier la notion de champ et celle d'environnement ; c'est le cas par exemple dans le modèle MMASS [Bandini *et al.*, 2002], ou dans celui des champs computationnels dans [Mamei *et al.*, 2002], ou encore dans les champs de phéromones numériques [Glad *et al.*, 2008]. L'idée sous-jacente est d'associer l'environnement à une mémoire du système, mémoire stockée et gérée dans la notion de champ, qui peut agir sur la dynamique du système.

Définition 8.2.3. *L'environnement $\mathbb{W}$ du SMA logistique est un environnement à mémoire constitué du tuple $\langle E, \mathbb{F} \rangle$ dont les éléments sont définis de la façon suivante :*

- *E est l'espace dans lequel sont situés les agents. Nous utilisons essentiellement dans cette thèse un espace métrique euclidien continu/discret ou un graphe selon le problème considéré.*
  Dans le cas d'un espace euclidien, un point peut être repéré par un ensemble de coordonnées notées par un vecteur $\mathbf{X} = (X, Y, \cdots)^T$. Nous noterons systématiquement en majuscules les coordonnées cartésiennes, les lettres minuscules x, y étant réservées aux variables de l'application logistique.
- *$\mathbb{F}$ est un tuple de champs de données $\langle \mathcal{F}_1, \mathcal{F}_2, \dots \rangle$ définis à partir de l'espace E :*
  un champ scalaire renvoie une valeur réelle, tandis qu'un champ vectoriel renvoie un vecteur (de l'espace vectoriel $\mathbb{R}^n$ ici) ce qui se traduit par :

$$\begin{array}{ll} \textit{Champ scalaire :} & E \to \mathbb{R} \\ \textit{Champ vectoriel :} & E \to \mathbb{R}^n \\ & \mathbf{X} \mapsto \mathcal{F}(\mathbf{X}) \end{array}$$

Discussion.

Cette définition de l'environnement amène quelques précisions. La distinction entre d'une part l'espace considéré et le(s) champ(s) d'autre part est très courante en géométrie. Les deux concepts n'ont pas la même utilité :

- L'espace donne le socle géométrique ou plus largement topologique, l'essentiel étant de pouvoir y définir un voisinage, dont nous avons besoin pour définir les perceptions lim-

itées dans notre approche. De plus, il peut être associé à une métrique et un système de coordonnées, ce qui est habituellement le cas dans nos simulations.
- Les champs peuvent être interprétés comme une sorte de stockage/mémoire sur l'espace. Un champ scalaire stocke une valeur réelle en tout point de l'espace, tandis qu'un champ vectoriel stocke un vecteur de la dimension de l'espace.
- Le concept de champ est pris ici au sens d'une structure de données informatique générique propre au système multi-agent logistique, qui autorise des opérations de lecture et d'écriture de données. L'implantation informatique peut être réalisée au moyen de tableaux, listes, matrices, ..., en vue de minimiser selon les cas la complexité algorithmique des opérations élémentaires associées.

8.2.4 Caractéristiques des champs de l'environnement

Propriétés associées au concept de champ

Un champ est une application dépendant du temps et qui peut suivre une loi d'évolution intrinsèque qui s'applique lors de la mise à jour des champs de l'environnement (phase de réaction).

Évaporation. Pour un champ scalaire de phéromone noté $\mathcal{F}$ par exemple, on définit la loi d'évaporation intrinsèque associée à un coefficient d'évaporation réel $\rho \in [0, 1]$ selon :

$$\mathcal{F}(\mathbf{X})^{t+1} = (1 - \rho)\ \mathcal{F}(\mathbf{X})^t \tag{8.3}$$

Tout champ peut être caractérisé par un coefficient d'évaporation qui détermine finalement la persistance des données dans le champ. Nous proposons donc de généraliser cette approche aux champs de l'environnement et de qualifier systématiquement tous les champs par sa valeur de coefficient d'évaporation.

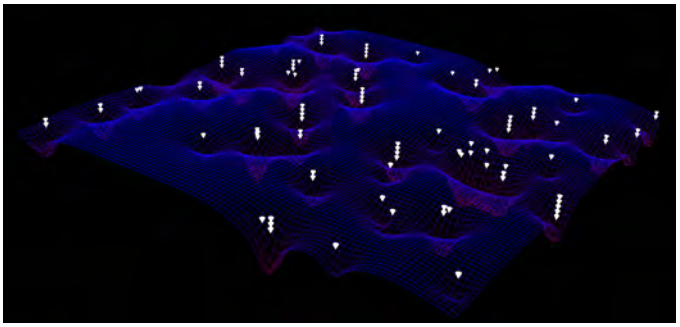
Définition 8.2.4. *Ainsi on distingue :*
- *les champs persistants pour lesquels $\rho = 0$,*
- *les champs volatils pour lesquels $0 < \rho < 1$ à décroissance exponentielle*
- *les champs non persistants pour lesquels $\rho = 1$.*

Diffusion. De même un champ peut être diffusif de coefficient D. Dans ce cas, le champ suit une autre loi du type :

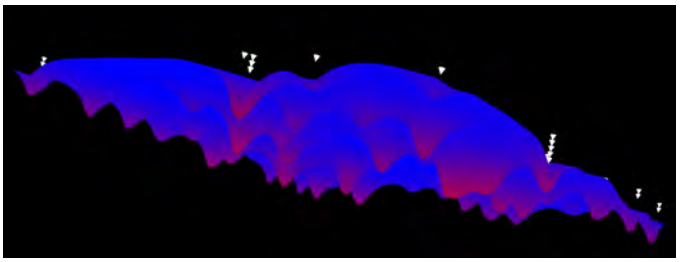
$$\mathcal{F}(\mathbf{X})^{t+1} = \mathcal{F}(\mathbf{X})^t + D\ \triangle \mathcal{F}(\mathbf{X})^t \tag{8.4}$$

où $\triangle$ définit l'opérateur laplacien discret $\triangle$ [44].
Ainsi on pourra parler de champ diffusif si $D > 0$ et de champ non diffusif si $D = 0$. .

[44] En continu l'opérateur laplacien est donné par $\triangle = \frac{\partial}{\partial X^2} + \frac{\partial}{\partial Y^2} + \cdots$



(a) Vue des agents “au-dessus” du champ avec un rendu fil de fer.



(b) Vue de dessous du champ scalaire, les vallées révèlent les positions des agents.

FIG. 8.2 – Illustration de la notion de champ dans un espace 3D tirée de “3D HeatBugs” par Gabriel Catalin Balan sur la plate-forme de simulation MASON [Luke *et al.*, 2003].

Illustration. La figure 8.2 illustre ce cas sur une représentation 3D de la simulation du problème dénommé ”Heatbug“ développée sur la plate-forme de simulation multi-agent MASON par Gabriel Catalin Balan. Le champ représenté en 3D est un champ de chaleur scalaire à la fois diffusif et évaporatif. Les agents représentés en blanc évoluent en réalité dans cette simulation dans un espace 2D. Les agents qui ont une même coordonnée 2D sont représentés l’un au dessus de l’autre dans cette visualisation.

Champs principaux utilisés avec le SMAL.

On peut distinguer quelques grands types de champs rencontrés avec le modèle SMAL.

Champs de décision, de contrôle et de couplage. En toute généralité, les trois composantes qui définissent l’état interne d’un agent logistique ont leur pendant en termes de champs de l’environnement. Nous les désignons respectivement par les termes de champ de décision, champ de contrôle, et champ de couplage, notés respectivement $\mathcal{X}$, $\mathcal{C}$ et $\mathcal{E}$. Ce sont des champs scalaires pour les problèmes que nous traitons. Nous les désignons par le même qualificatif que la composante à laquelle ils se rattachent parcequ’ils servent de sources de perception selon cette composante. Ils ne sont pas systématiquement présents dans le modèle mais interviennent essentiellement dans les mécanismes d’adaptation du SMAL. Ainsi les trois grandeurs caractéristique de l’agent logistique sont-elles déclinées sur les champs et les perceptions pour identifier les flux

de données dans le système.

- Le champ scalaire de décision $\mathcal{X}(\mathbf{X})$ stocke de façon non persistante la valeur de la partie décisionnelle de l'état interne d'un agent. Ce champ contribue à la décision collective des agents. Il est nécessaire dans tous les problèmes étudiés.
- Le champ de contrôle $\mathcal{C}$ est persistant ou volatil, il intervient via un opérateur de perception pour modifier la variable de contrôle interne de l'agent. Il représente un type de champ important pour les fourmis logistiques (au chapitre 12).
- Le champ de couplage $\mathcal{E}$ intervient de la même façon que le champ de contrôle sur la variable de couplage. Il est moins exploité dans nos instances pour la simulation que le champ de contrôle, mais le mécanisme d'adaptation qui en résulte est illustré par le flocking de la section 11.1.

Les champs de couplage et de contrôle n'interviennent pas simultanément dans les problèmes que nous avons traité.

Champs des positions individuelles. Les champs de position individuelle des agents sont essentiels pour la perception des autres champs (cf. section 8.2.6). Pour l'agent A_i situé en $\mathbf{X}_i$, le champ de position est noté δ_i et correspond à la fonction suivante :

$$\text{si E est continu, fonction décentrée de Dirac : } \delta_i(\mathbf{X}) = \delta(\mathbf{X} - \mathbf{X_i}) = \begin{cases} \infty & \text{, si } \mathbf{X} = \mathbf{X}_i \\ 0 & \text{sinon} \end{cases} \tag{8.5}$$

$$\text{si E est discret, symbole de Kronecker : } \delta_i(\mathbf{X}) = \begin{cases} 1 & \text{, si } \mathbf{X} = \mathbf{X}_i \\ 0 & \text{sinon} \end{cases} \tag{8.6}$$

Ce champ a la propriété de posséder une intégrale égale à 1 en continu, ou une somme égale à 1 en discret. Il est a priori non persistant puisque la position passée n'est pas conservée. Ce type de fonction est employé en physique pour modéliser une particule, une charge ou une masse ponctuelle.

Notons qu'en pratique, lors de l'implantation informatique, on ne fonctionne pas avec autant de champs qu'il y a d'agents, mais avec une structure de liste stockant tous les agents et leur position, plus économe en espace mémoire. De plus, on ne peut pas implémenter une fonction de Dirac sur ordinateur, si ce n'est en la modélisant comme une gaussienne très étroite. Nous opérons plutôt par discrétisation de l'espace, le symbole de Kronecker étant plus simple à manipuler que la fonction de Dirac.

Pour la cohérence du formalisme et pour l'écriture des opérateurs sur les champs, nous gardons des champs abstraits de de ce type. Cet aspect est discuté en section 8.2.9.

Champs des positions cumulées. Le champ de position cumulée $\mathcal{N}(\mathbf{X}) \in \mathbb{N}$ stocke le nombre des agents en $\mathbf{X}$ à l'instant t ; on passe du local au global par un opérateur de sommation si E est discret :

$$\mathcal{N}(\mathbf{X}) = \sum_{A_i \in \mathbb{A}} \delta_i(\mathbf{X}) \tag{8.7}$$

Ce champ permet simplement à l'agent d'évaluer le nombre d'agents dans son voisinage afin par exemple de calculer des moyennes sur d'autres grandeurs du champ.

Si E est continu, l'opération de dénombrement est plus compliquée avec les fonctions décentrées de Dirac. On a recours à une discrétisation au moyen d'une fonction créneau $h_R(\mathbf{X})$ avec R infiniment petit, définie par :

$$h_R(\mathbf{X}) = \begin{cases} 1 & \text{, si } \frac{-R}{2} \leqslant |\mathbf{X}| \leqslant \frac{R}{2}, \\ 0 & \text{, si } \mid \mathbf{X} \mid > \frac{R}{2} \end{cases} \tag{8.8}$$

Par produit de convolution noté $*$ entre h_R et δ_i on obtient un créneau centré sur $\mathbf{X}_i$:

$$\begin{aligned} (h_R * \delta_i)(\mathbf{X}) &= \int_E h_R(\mathbf{X} - \mathbf{Y})\delta_i(\mathbf{Y})d\mathbf{Y} \\ &= h_R(\mathbf{X} - \mathbf{X}_i) \end{aligned} \tag{8.9}$$

Ensuite, il ne reste qu'à sommer les fonctions créneaux obtenues sur l'ensemble des agents :

$$\mathcal{N}(\mathbf{X}) = \sum_{A_i \in \mathbb{A}} (\delta_i \ * \ h_R)(\mathbf{X}) \tag{8.10}$$

Ce faisant, nous avons construit un opérateur sur les champs δ_i.

Autres types de champs.

- Le champ scalaire de phéromones digitales volatiles ou persistantes peut être défini sur un espace E euclidien : $\mathcal{T}(\mathbf{X}) \in \mathbb{R}$. Ou bien sur un graphe : $\mathcal{T}(i, j)$ où (i, j) désigne un arc du graphe. Ce champ est utile pour les algorithmes fourmis.
- Le champ vectoriel des vitesses des agents dans l'environnement est noté $\mathcal{V}(\mathbf{X}) \in E$ et est utilisé notamment dans le flocking.

Champs d'influences des agents.

Intérêt des champs d'influences La phase de réaction du système 8.1 engage une mise à jour de tous les champs qui constituent l'état de l'environnement. Cette mise à jour résulte des influences combinées des agents. Nous modélisons ces influences par des champs temporairement stockés par l'environnement en vue d'un traitement global lors de la phase de réaction. Ces champs sont comme des champs tampons, systématiquement non persistants (i. e. dont les valeurs sont écrasées d'un pas de temps à l'autre), qui vont servir à maintenir l'aspect synchrone des mises à jour. Le modèle que nous proposons est en effet synchrone par défaut et l'existence des champs d'influence rend possible la simultanéité des mises à jour.

Notation Pour la notation de ces champs d'influences, nous proposons de poser un $\widetilde{tilde}$ en chapeau sur les champs de l'environnement concernés.
Exemples :

- $\widetilde{\mathcal{N}}$ désigne le champ influence de position cumulée,
- de façon générale $\widetilde{\mathcal{C}}$ désigne le champ influence du champ $\mathcal{C}$

Avantages liées à l'utilisation du concept de champ.

Nous concluons cette section consacrée à l'environnement par quelques remarques concernant l'intérêt d'utiliser ce concept de champ de façon systématique :

- De nombreux opérateurs peuvent être appliqués sur un champ, les opérateurs usuels gradient $\overrightarrow{\nabla}$ et laplacien $\triangle$ par exemple, ou l'opérateur de dénombrement vu précédemment. Quelques opérateurs de perception que nous employons couramment sont présentés en 8.2.6.
- Le concept de champ s'implante assez naturellement du point de vue informatique et permet d'abstraire les structures de données utilisées quelque soit l'espace E. Les plateformes de simulation orientées multi-agent comme MASON [Luke *et al.*, 2003], utilisent d'ailleurs ce type de structure de données pour stocker les grandeurs manipulées.
- L'intérêt des champs est également computationnel : on transfère ainsi une partie de la complexité temporelle des algorithmes en une complexité spatiale. Typiquement, on tire parti de l'accès direct en $O(1)$ des structures de type tableau de données n-dimensionnels. Les opérateurs de perception et d'action ont des complexités algorithmiques réduites du fait de l'usage de cette mémoire spatiale. Le voisinage de perception limitée réduit encore cette complexité. Une des sources de complexité algorithmique temporelle les plus coûteuses apparaît notamment lors de la recherche des plus proches voisins et du calcul des distances relatives entre agent, très utilisée dans les approches à base de champs de potentiels. Cette détection sert en effet au calcul des forces d'attraction / répulsion qui servent ensuite à déterminer la vitesse des agents (cas du flocking de Reynolds en particulier). Ce processus se réalise en $O(N^2)$, où N est le nombre d'agents, puisque pour chaque agent on recherche parmi tous les autres agents ceux dont la distance à l'agent courant ne dépasse pas un rayon de voisinage R. Dans notre approche, ce processus est différent et permet de diminuer la complexité via les champs de position cumulée des agents et l'opérateur de convolution sur ce champ associé à la perception sur un voisinage (cf. 8.2.6). La mise à jour des champs de position se fait en $O(1)$ pour chaque agent, puis le calcul sur le voisinage va uniquement dépendre de la taille de ce voisinage et non plus du nombre d'agents. Les distances jouent peu de rôles dans notre modèle qui n'utilise pas de forces dérivant de potentiels fonctions de la distance.

8.2.5 Équations maîtresses du SMAL

Transition d'état et pour l'agent logistique

La transition d'état de l'agent logistique A_i selon les trois variables s'écrit de façon générale d'après 8.1 en décomposant F selon les trois composantes respectives $\langle F_x, F_a, F_\epsilon \rangle$ ainsi que trois composantes pour la perception $\langle P_x, P_a, P_\epsilon \rangle$:

$$a_i^{t+1} = F_a\left(a_i^t, P_{a,i}(\sigma^t)\right) \tag{8.11}$$

$$\epsilon_i^{t+1} = F_\epsilon\left(\epsilon_i^t, P_{\epsilon,i}(\sigma^t)\right) \tag{8.12}$$

$$x_i^{t+1} = F_x\left(x_i^t, P_{x,i}(\sigma^t), a^{t+1}, \epsilon^{t+1}\right) \tag{8.13}$$

Remarques.

Écriture de la dynamique. A chaque variable est associée une perception qui opère à partir d'un ou plusieurs champs de l'environnement. Ces perceptions sont indicées également par i, indice de l'agent qui possède en général une perception limitée caractérisée par un voisinage centré sur sa position comme cela a été expliqué précédemment. Cette écriture peut sembler un peu lourde au premier abord mais elle exprime la composition de la dynamique de transition d'états avec les perceptions.

Séquencement des calculs. D'autre part, on remarque ici la distinction entre la variable x et les deux autres variables : x dépend de la mise à jour des deux autres variables, l'écriture exprime l'ordre séquentiel de l'algorithme de calcul. La variable x fusionne toutes les informations perçues par l'agent, d'où son rôle de variable de décision.

Invariance de F. Enfin, il est nécessaire de vérifier au moment de l'instanciation du modèle que le domaine de définition $\mathbb{D}^3$ dans le cas monodimensionnel sur chaque composante reste invariant par $F = \langle F_x, F_a, F_\epsilon \rangle$, c'est-à-dire que :
$F(\mathbb{D}^3) \subseteq \mathbb{D}^3$
afin de garantir le confinement des calculs dans un domaine borné.

Lien avec le gaz d'itérations couplées. La décision est finalisée par la dernière équation qui utilise les résultats des mises à jour des deux variables a et ϵ. Pour donner dès maintenant le résultat de ce formalisme, la prise de décision de l'agent se résume à l'équation générique suivante qui intègre les perceptions possibles de l'agent sur son environnement :

$$x_i^{t+1} = (1 - \epsilon_i^{t+1}) f(x_i^t, a_i^{t+1}) + \epsilon_i^{t+1} \; P_{x,i}(\sigma^t) \tag{8.14}$$

où f désigne une des formes conjuguées de l'application logistique. Il est délicat de préciser davantage à ce stade quelles formes prennent les perceptions selon a et selon ϵ, car elles dépendent du problème que l'on souhaite modéliser et simuler. Cela dit, dans les problèmes que nous traitons, notamment dans l'étude des phénomènes de groupe, on aura très souvent la formulation suivante pour $P_{x,i}(\sigma^t)$:

$$P_{x,i}(\sigma^t) = \frac{\sum_{j \in V_i} \; f(x_j^t, a_j^{t+1})}{N_i^t} \tag{8.15}$$

où N_i^t est le nombre d'agents perçus dans le voisinage V_i de l'agent A_i au temps t, y compris lui-même de façon à ne jamais avoir de perception nulle menant à une décroissance systématique de x_i vers 0. Cette inclusion de l'agent lui-même dans son voisinage de perception est rendue nécessaire par le fait qu'il n'y a pas nécessairement d'autres agents dans son voisinage, et afin que la formulation (8.14) reste valable quel que que soit le nombre d'agents voisins. Cela conduit à une formulation très proche de celle des gaz d'itérations couplées (GIC) en remplaçant l'expression (8.15) dans l'équation (8.14) :

$$x_i^{t+1} = (1 - \epsilon_i^{t+1}) f(x_i^t, a_i^{t+1}) + \frac{\epsilon_i^{t+1}}{N_i^t} \sum_{j \in V_i} \; f(x_j^t, a_j^{t+1}) \tag{8.16}$$

Elle est cependant plus générale que la formulation dans les GIC, puisqu'à la fois les voisinages, les couplages, et les paramètres de contrôles peuvent être sujets à variations.

En réalité cette formule implique deux processus successifs de perception-décision-action que nous détaillons en section 8.2.7.

8.2.6 Opérateurs de perception limitée.

Nous empruntons une approche orientée traitement du signal sur les aspects perceptifs de l'agent. Certaines propositions de formalisation des systèmes multi-agents réactifs ont également emprunté cette voie [Ferrand *et al.*, 1994], sans pouvoir véritablement exploiter ce cadre descriptif général et formel [Ferrand, 1997].

Définition du voisinage.

Pour expliquer cette approche, nous considérons le cas simple d'un environnement à une dimension d'espace pour présenter l'opérateur de perception limitée. Les données du problème sont celles-ci :

- un agent A_i situé en la coordonnée d'espace X_i à l'instant t,
- sa perception limitée est caractérisée par un voisinage de perception V centré en X_i, il ne peut lire et récupérer l'information qu'au travers de ce voisinage,
- ce voisinage intègre éventuellement l'agent lui-même afin de maintenir la validité des formulations du type (8.16) quel que soit le nombre d'agents voisins perçus.
- on suppose ici que le champ perçu est $\mathcal{C}(X)$ en X au temps t,
- le voisinage de perception V de rayon R est décrit par une fonction créneau $h_R(X)$ déjà définie dans la formule (8.8), où R peut être quelconque cette fois.

Opérateur de perception limitée dans un voisinage.

Le produit de convolution de $h_R(X)$ et de la fonction de Dirac $\delta_{X_i}(X)$[45] donne une fonction créneau "glissante" toujours centrée en X_i définie en 8.9.
L'information perçue via le voisinage V sur le champ $\mathcal{F}^t(X)$ est le produit de $\mathcal{F}^t(X)$ par cette convolution, soit au final le champ de perception :

$$\mathcal{F}_i^t(X) = \mathcal{F}^t(X) \cdot (h_R * \delta_{X_i})^t(X) \tag{8.17}$$

Pour finir, on définit l'opérateur de perception limitée P_{V_i} sur un voisinage V_i relatif à l'agent i comme :

$$P_{V_i} : \mathbb{F} \times \mathbb{A} \rightarrow \mathbb{F} \tag{8.18}$$

$$(\mathcal{C}, i) \rightarrow P_{V_i}(\mathcal{C}, i) = \mathcal{C}_i \tag{8.19}$$

[45] Elle vaut ∞ en X_i et 0 ailleurs, mais l'intégrale sur $\mathbb{R}$ donne 1

Opérateur de sommation.

Nous utilisons souvent la sommation Σ d'un champ $\mathcal{F}$ sur le voisinage de l'agent i, à savoir :

$$\Sigma_{V_i}\ \mathcal{F} = \begin{cases} \int_{V_i} P_{V_i}(\mathcal{F}, i) & \text{, si } E \text{ est continu,} \\ \sum_{V_i} P_{V_i}(\mathcal{F}, i) & \text{, si } E \text{ est discret} \end{cases} \tag{8.20}$$

Nous ferons référence à l'opérateur Σ_{V_i} désormais qui résume le processus de calcul sous-jacent sans avoir à le détailler.

Opérateur de perception ponctuelle.

Ce qui a été présenté précédemment concerne une perception sur un voisinage. Lorsque ce voisinage est réduit au point courant $\mathbf{X}_i$ où l'agent i est situé, la récupération de la valeur d'un champ $\mathcal{F}$ en ce point s'écrit simplement par les opérations suivantes :

$$\mathcal{F}(\mathbf{X}_i) = \begin{cases} \delta_i * \mathcal{F} & \text{, si } E \text{ est continu,} \\ \sum_k \delta_i(\mathbf{X}_k) \cdot \mathcal{F}(\mathbf{X}_k) & \text{, si } E \text{ est discret} \end{cases} \tag{8.21}$$

Opérateur de moyenne.

D'autre part, très souvent, nous utiliserons un opérateur de moyenne d'un champ sur le voisinage V_i, c'est-à-dire le rapport de la sommation du champ à la sommation d'un champ de dénombrement. On notera cet opérateur $\langle\ \rangle_{V_i}$ en précisant à quel objet cela fait référence.

Par exemple, la moyenne glissante sur les agents d'un champ quelconque $\mathcal{F}$ sera donc simplement la sommation de ce champ divisée par la sommation du champ $\mathcal{N}$ de position cumulée des agents. Il est défini par :

$$\langle \mathcal{F} \rangle^t_{V_i} = \frac{\Sigma_{V_i}\ \mathcal{F}^t}{\Sigma_{V_i}\ \mathcal{N}^t} \tag{8.22}$$

Opérateurs max et min.

Deux types d'opérateurs s'avèrent particulièrement utiles dans les problèmes d'optimisation qui permettent de récupérer le minimum ou maximum d'un champ sur un voisinage donné. La condition d'application de cet opérateur est que le domaine de valeurs du champ soit fini, c'est-à-dire dans les bornes des valeurs numériques admissibles par le calculateur utilisé, bien qu'il soit envisageable de prendre en compte ces valeurs hors limite en émettant une alerte et un processus de traitement adapté (gestion d'une exception dans certains langages de programmation). Nous notons ces opérateurs $\mathcal{M}ax$ et $\mathcal{M}in$. L'opérateur $\mathcal{M}in$ est par exemple défini relativement à un champ par :

$$\mathcal{M}in_{V_i}\ \mathcal{F} = \min\{\mathcal{F}(\mathbf{X}_k) | \mathbf{X}_k \in V_i\} \tag{8.23}$$

Application des opérateurs de perception.

Les opérateurs de perceptions sont composés des opérateurs déjà cités, des combinaisons de ces opérateurs, ou d'autres liés à un problème spécifique pour obtenir une fonction de perception finale Pi de l'agent i.

La perception agit *a priori* sur les trois composantes de l'état interne de l'agent, à savoir les trois composantes de perception $\langle P_x, P_a, P_\epsilon \rangle$.

8.2.7 Processus sensorimoteurs

Cette section détaille les différents processus algorithmiques mis en oeuvre pour obtenir la formulation (8.16) comprise sous l'angle des perceptions-décisions-actions des agents suivis de la réaction de l'environnement.

L'algorithme global 8.1 représente la boucle principale de mise en oeuvre du SMAL incluant les deux processus sensorimoteurs ainsi que la/les réactions de l'environnement qui font l'objet des explications suivantes.

Algorithm 8.1: AlgorithmeSMAL()

```
Initialisation :
repeat
    Processus sensorimoteurs individuels :
    for i = 1 to N do
        Process-Perception-Decision-Action1(A_i)
    Processus sensorimoteur collectif :
    for i = 1 to N do
        Process-Perception-Decision-Action2(A_i)
    Réaction de l'environnement :
    begin
        Mise à jour du champ de décision non persistant :
        X ← X̃
        Mise à jour de tous les champs modifiés par l'action des agents :
        begin C est un champ quelconque ici
            Champs persistants : C ← C + C̃
            Champs non persistants : C ← C̃
            Champs évaporatifs : C ← (1 − ρ)C + C̃
            Champs diffusifs : C ← C + D · △C + C̃
        end
    end
until stop
```

Processus sensorimoteur (1) agent-environnement.

Un premier processus sensorimoteur met en interaction l'agent et son environnement. Les étapes de ce processus sont les suivantes (cf. algorithme 8.2).

Algorithm 8.2: Process-Perception-Decision-Action1(A_i)

Variables d'états au temps t :
begin
  agent $(a_i^t, \epsilon_i^t, x_i^t)^T$
  environnement σ^t
end
Perceptions 1 :
begin
  Récupérer la position courante $\mathbf{X_i}$ de A_i dans le champ $\mathcal{P}_i$
  Calculer $P_{a,i}(\sigma^t)$
  Calculer $P_{\epsilon,i}(\sigma^t)$
end
Decision 1 :
begin
  $a_i^{t+1} \leftarrow F_a\left(a_i^t, P_{a,i}(\sigma^t)\right)$
  $\epsilon_i^{t+1} \leftarrow F_\epsilon\left(\epsilon_i^t, P_{\epsilon,i}(\sigma^t)\right)$
  $y_i \leftarrow f(x_i^t, a_i^{t+1})$
end
Action 1 :
begin
  $\widetilde{\mathcal{X}}(\mathbf{X_i}) \leftarrow \widetilde{\mathcal{X}}(\mathbf{X_i}) + y_i$
end

Perception 1. La perception réalise les calculs suivants :
- récupération de la position courante $\mathbf{X}_i$ de l'agent i pour calculer les perceptions dans le voisinage de l'agent,
- calcul de la perception $P_{a,i}(\sigma^t)$
- calcul de la perception $P_{\epsilon,i}(\sigma^t)$

Décision 1. La première partie de la décision est réalisée par :
- le calcul de (8.11),
- le calcul de (8.12),
- le calcul de l'application logistique f seule sur la variable de décision x de l'agent avec le paramètre de contrôle a_i^{t+1}, stocké dans une variable temporaire y_i soit :
  $y_i = f(x_i^t, a_i^{t+1})$

Actions 1. La seule action à ce niveau est l'écriture de y_i dans le champ d'influence $\widetilde{\mathcal{X}}^{t+1}$ —nul initialement— de la variable de décision au temps $t+1$:
$\widetilde{\mathcal{X}}(\mathbf{X}_i)^{t+1} = \widetilde{\mathcal{X}}(\mathbf{X}_i)^{t+1} + y_i$
Cette formule montre que ce champ est cumulatif sur l'ensemble des agents.

Processus sensorimoteur (2) agent-agent.

Le second processus concerne les interactions de l'agent avec son essaim. C'est au cours de ce processus qu'il couple son état interne avec celui des autres agents. Deux cas se présentent : ou bien l'agent perçoit d'autres agents dans son voisinage, ou bien il n'en perçoit pas (cf. algorithme 8.3).

Algorithm 8.3: Process-Perception-Decision-Action2(A_i)

Perception 2 :
begin
  Perception du nombre d'agents dans le voisinage V_i de A_i :
  $N_i \leftarrow \Sigma_{V_i} \mathcal{N}$
  if $N_i \neq 0$ *(il existe des voisins)* **then**
    $P_{x,i} \leftarrow \dfrac{\Sigma_{V_i} \widetilde{\mathcal{X}}}{N_i}$
  else
    $P_{x,i} \leftarrow f(x_i^t, a_i^{t+1})$
end
Décision 2 :
begin
  $x_i^{t+1} \leftarrow (1-\epsilon_i^{t+1}) f(x_i^t, a_i^{t+1}) + \epsilon_i^{t+1}\ P_{x,i}(\sigma^t)$
end
Actions 2 :
begin
  Calculer le déplacement sur E à partir de x_i^{t+1} :
  $\mathbf{X_i^{t+1}} \leftarrow M_i(\mathbf{X_i^t}, x_i^{t+1})$
  Mettre à jour le champ de position individuelle :
  $\widetilde{\delta_i}^{t+1} = \delta(\mathbf{X} - \mathbf{X}_i^{t+1})$
  Marquer les autres champs de l'environnement de façon similaire
end

Perception 2. L'agent perçoit le nombre d'agent N_i dans son voisinage V_i. Puis l'agent peut calculer la perception P_x :

$$P_{x,i}(\sigma^t) = \begin{cases} \langle\ \mathcal{X} \rangle_{V_i}^t & \text{, si } N_i \neq 0 \\ f(x_i^t, a_i^{t+1}) & \text{sinon} \end{cases} \tag{8.24}$$

Décision 2. La décision est donnée en remplaçant (8.24) dans (8.14) pour donner soit la formulation (8.16) s'il y a des voisins autour de l'agent, soit le résultat du calcul de l'application logistique seule $x_i^{t+1} = f(x_i^t, a_i^{t+1})$ s'il n'y a pas de voisins.

Actions 2. La spécification des actions finales dépend du problème posé. Ceci dit, les agents que nous considérons sont mobiles dans l'environnement. On a donc toujours une action de déplacement à envisager. La caractéristique des actions est de dépendre directement de la valeur mise à jour x_i^{t+1} de la variable de décision de l'agent. Nous exposons l'action de déplacement comme exemple type de construction des actions de façon générale.

Le déplacement de l'agent correspond à une action de translation des données du champ de position individuelle δ_i pour chaque agent, c'est-à-dire que l'on déplace le pic de Dirac sur la nouvelle position. Cela implique le calcul du champ d'influence associé $\widetilde{\delta_i}$ après avoir calculé la nouvelle position $\mathbf{X}_i^{t+1}$ de l'agent. On supposera qu'il existe une loi de déplacement M_i pour l'agent i de sorte que l'on puisse écrire l'action de déplacement complète :

$$\mathbf{X}_i^{t+1} = M_i(\mathbf{X}_i^t, x_i^{t+1}) \tag{8.25}$$

$$\widetilde{\delta_i}^{t+1} = \delta(\mathbf{X} - \mathbf{X}_i^{t+1}) \tag{8.26}$$

Les autres actions possibles de l'agent (dépôt d'une phéromone, d'une information) suivent le même schéma.

En particulier, le calcul de la vitesse de déplacement de l'agent est également une action dont la valeur est stockée dans le champ vecteur $\mathcal{V}$. Cette valeur intervient en général dans le calcul de la nouvelle position de l'agent.

8.2.8 Réactions de l'environnement

Il s'agit de la mise à jour des divers champs à partir des champs d'influence établis par l'ensemble des agents :

- le champ de décision $\mathcal{X}$ non persistant est mis à jour selon :
  $\mathcal{X}^{t+1} = \widetilde{\mathcal{X}}^{t+1}$
- les champs de position individuel sont également mis à jour selon :
  $\delta_i^{t+1} = \widetilde{\delta_i}^{t+1}$
  ainsi que le champ de positionnement cumulé $\mathcal{X}$ selon l'opération (8.9) en discret ou (8.10) en continu.
- De la même façon tous les champs sont mis à jour à partir des champs d'influences selon les caractéristiques du champ considéré $\mathcal{C}$:

$$\begin{aligned}
\text{Champs persistants :} \quad & \mathcal{C}^{t+1} &= \mathcal{C}^t + \widetilde{\mathcal{C}}^{t+1} \\
\text{Champs non persistants :} \quad & \mathcal{C}^{t+1} &= \widetilde{\mathcal{C}}^{t+1} \\
\text{Champs volatils :} \quad & \mathcal{C}^{t+1} &= (1-\rho)\mathcal{C} + \widetilde{\mathcal{C}}^{t+1} \\
\text{Champs diffusifs :} \quad & \mathcal{C}^{t+1} &= \mathcal{C}^t + D \cdot \triangle\mathcal{C}^t + \widetilde{\mathcal{C}}^{t+1}
\end{aligned}$$

- Enfin tous les champs d'influence non persistants sont remis à 0 pour le pas de temps suivant.

8.2.9 Discussion

Cette formalisation très détaillée à ce niveau présente les deux aspects principaux des mécanismes mis en oeuvre dans le SMAL. Dans les applications concrètes que nous étudions, il est rare que tous ces aspects apparaissent en même temps. Quoiqu'il en soit, nous avons posé la structure et définit les concepts qui seront utilisés dans toute la suite de cette thèse. Suivent quelques sujets de discussion qui complètent la présentation du modèle.

Adaptation et auto-organisation.

Nous avons distingué ce qui relève des processus d'adaptation concernant notamment les équations de transition (8.11) et (8.12), des interactions entre agents à la base des processus auto-organisés exprimés dans l'équation principale (8.13). La partie de ce document consacrée à l'exploitation des processus d'auto-organisation correspond au chapitre 10, quant aux mécanismes typique d'adaptation aux données des champs de l'environnement, ils font l'objet des chapitres 11 et 12 de ce document.

Le problème des coordonnées d'espace et la centralisation des informations.

Le problème du positionnement est traité de façon très particulière dans notre modèle puisqu'il n'apparaît pas dans l'état de l'agent, contrairement aux approches de type particulaire. Nous avons fait le choix de transférer cette information dans l'environnement puisque connaître sa position dans un espace est nécessairement une information centralisée extérieure à l'agent qui entre en contradiction avec la connaissance partielle de l'environnement. Cette information, l'agent doit la lire ou la recevoir éventuellement d'une autre structure, en l'occurrence l'environnement par l'intermédiaire d'un champ propre à l'agent, s'il doit utiliser cette information explicitement. On l'a vu, la position intervient mathématiquement comme un champ δ_i dans les opérateurs et est à la base du calcul des voisinages et de la perception limitée de l'environnement.

En pratique et dans les instanciations du SMAL qui seront présentées par la suite, nous ne rappellerons pas systématiquement le détail des opérations à réaliser sur les champs position et vitesse. Nous manipulerons la position et la vitesse propres de l'agent comme si ces grandeurs faisaient partie de l'état propre de l'agent considéré. De plus, on discrétise l'espace pour pouvoir faire des calculs, étant entendu qu'une fonction de Dirac ne peut qu'être approchée sur ordinateur, et on utilise plutôt le symbole de Kronecker. Mais le modèle théorique reste formalisé avec des champs éventuellement continus.

Remarque sur l'optimisation.

L'optimisation d'une fonction objectif ou la réalisation d'un but déterminé n'ont pas été évoquées dans cette description des mécanismes et concepts de base qui font le SMAL : seule l'algorithmique générale des processus décisionnels réactifs individuels et collectifs a été développée. L'optimisation implique des principes de résolution qui viennent se greffer à la structure algorithmique déjà établie. Ces aspects sont traités en section 2.1.

8.3 Analyse du SMA logistique

Examinons à présent le modèle que nous venons de présenter. Il s'agit de comprendre dans un premier temps comment peuvent être exploitées les propriétés non linéaires de l'application logistique dans le cadre de l'intelligence en essaim et ainsi anticiper sur la mise au point des mécanismes auto-organisationnels et adaptatifs.

D'autre part, nous justifions de son utilisation sur un plan biologique. Il nous semble essentiel de rattacher la phénoménologie chaotique à une origine dans le monde physique ou biologique, sans la considérer seulement comme une étrangeté mathématique. C'est l'étude des réseaux de neurones chaotiques qui nous fournit ce lien avec le monde biologique. Il est pour le moins troublant de trouver dans ce secteur de recherche des éléments issus d'études expérimentales qui corroborent l'existence de comportements chaotiques aux caractéristiques très proches de celles observées avec l'application logistique. Ainsi nous postulons que l'application logistique peut figurer comme modèle réduit d'un certain type de réseau de neurones et apporter ainsi un argument supplémentaire qui soutient notre modélisation, nouvelle dans le domaine des SMA pour l'intelligence en essaim [Charrier *et al.*, 2007d].

8.3.1 Intérêt algorithmique : génération des comportements d'exploration et d'exploitation

Étant entendu que l'intelligence en essaim est essentiellement appliquée en tant que méta-heuristique pour résoudre des problèmes d'optimisation en informatique, on peut ramener ce type de problèmes à un scénario du type "exploration-exploitation". Nous justifions ici en quoi l'application logistique peut servir a priori de fonction de base pour ce type de problème. L'ensemble des propriétés de l'application logistique itérée exposées en section 5.3 permettent de concevoir l'utilisation de cette application comme fonction centrale de décision de l'agent dans le cadre de l'intelligence en essaim. Plusieurs aspects de l'application et de son diagramme de bifurcation justifient cette hypothèse. Ils sont énoncés dans ce qui suit. Rappelons l'expression standard associée à cette application :

$$x_{n+1} = f(x_n) = r\ x_n(1 - x_n) \tag{8.27}$$

Chaos

L'application logistique est intrinsèquement un générateur aléatoire lorsque son paramètre de contrôle est fixé à la valeur qui génère un mode chaotique : le plus connu apparaît pour $r = 4$ dans la formule (8.27). Ce générateur a des défauts causés par des corrélations fortes dans les séries temporelles générées et de ce fait ne peut être utilisé comme générateur de nombres aléatoires. Ces défauts statistiques peuvent être corrigés (consulter l'article [Phatak et Rao, 1995] pour le détail de ces corrections) mais ce n'est pas notre objectif. Ce générateur aléatoire en nombres réels est de toute façon moins rapide et performant que les générateurs existants à base d'entiers. En revanche l'application logistique pour $r = 4$ génère des orbites denses dans l'intervalle $[0, 1]$ et cela suffit pour produire des phases exploratoires de l'environnement, phases nécessaires à toute heuristique de recherche d'une solution dans l'espace d'état d'un problème.

Paramètre de contrôle

L'application logistique est gouvernée par un paramètre de contrôle qui la fait évoluer de modes oscillatoires chaotiques à des modes périodiques et enfin à des modes points fixes. Ce paramètre de contrôle est susceptible d'être lui-même gouverné par des données extérieures via les perceptions de l'agent. Nous suivons en cela l'idée de la synergétique qui fait dépendre les paramètres de contrôle du système par les différents champs scalaires ou vectoriels présents dans l'environnement. Nous exploitons cette variabilité grâce au diagramme de bifurcation pour concevoir des mécanismes d'adaptation.

Diagramme de bifurcation

Dans le prolongement de la remarque précédente, le diagramme de bifurcation de l'application logistique décrit de manière statique et déterministe l'évolution au niveau microscopique de la résolution d'une recherche, en redéfinissant en termes dynamiques tout processus de résolution d'une recherche par le passage contrôlé d'un mode chaotique à un point fixe.

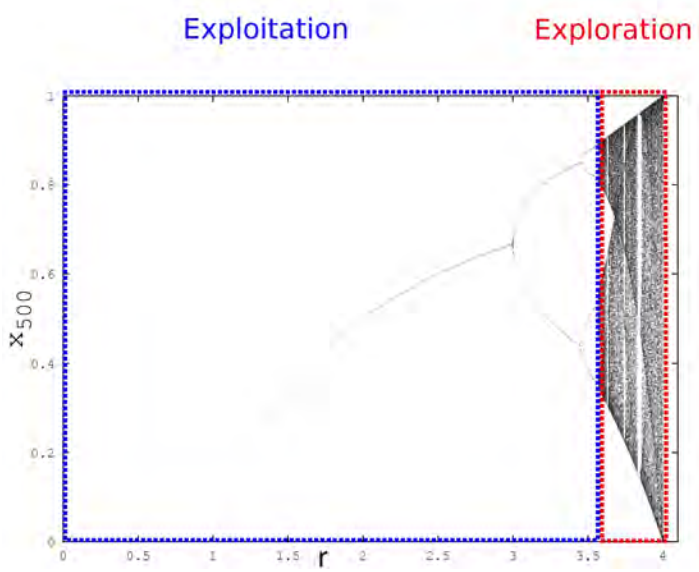


FIG. 8.3 – Application du principe exploration-exploitation aux zones du diagramme de bifurcation de l'application logistique $x_{n+1} = r\ x_n(1 - x_n)$

La figure 8.3 montre comment associer le principe d'exploration-exploitation aux zones du diagramme de bifurcation dans le cas de l'application logistique [46]. On peut séparer grossièrement ce diagramme en deux zones, une zone de chaos reliée au processus d'exploration, et une zone de périodicité ou de points fixes reliée à l'exploitation. Cette distinction est certes assez grossière, mais nous montrons dans les applications du SMA logistique pour l'optimisation qu'elle

[46] Cela est en fait valable pour toute application conjuguée de cette application, i.e. ayant le même type qualitatif de diagramme de bifurcation

s'avère pertinente. En outre, le diagramme de bifurcation nous permet de visualiser le "chemin dynamique" effectué par l'agent.

Synchronisation

Enfin l'application logistique est couplée avec d'autres applications du même type pour générer des phénomènes émergents de synchronisation que nous révélons être dans cette thèse d'une grande importance pour l'intelligence en essaim. Cet aspect n'est pas exploité dans toute sa complexité, étant donné les potentialités supposées de stockage et de classification de l'information de ces configurations dynamiques [Kaneko, 1994].

Bilan

Pour résumer, les caractéristiques essentielles de l'application logistique, à savoir sa capacité à générer des séquences pseudo-aléatoires par ses propriétés chaotiques, sa contrôlabilité via son diagramme de bifurcation, et ses possibilités de couplage enfin, en font un modèle de système complexe adapté à l'intelligence en essaim. La partie IV est consacrée à montrer cette assertion.

8.3.2 Interprétation biologique de la modélisation chaotique

L'application logistique itérée n'est pas seulement intéressante du point de vue algorithmique, elle porte une universalité de comportement liée à ses propriétés dynamiques que l'on rencontre également dans les systèmes biologiques, plus particulièrement dans les réseaux de neurones. L'annexe C fait une courte synthèse sur les avancées expérimentales et théoriques concernant les réseaux de neurones dits chaotiques, car ils montrent toute une gamme non linéaire de comportements. Résumons les grandes lignes de ces résultats :

- Les réponses chaotiques de neurones ont été observées dans des réseaux neuronaux spécialisés : dans le bulbe olfactif de chat, rat, lapin ; dans le système visuel de la salamandre et de l'homme dans les cas d'épilepsie.
- Dans chaque cas, on a pu montrer une route vers le chaos pilotée par la variation d'un paramètre de contrôle, lui-même gouverné par la perception. En particulier une route par doublement de période s'est révélée dans le cas de la vision.
- Cette route vers le chaos a été analysée mathématiquement sur la base de deux neurones, l'un excitateur, l'autre inhibiteur, et une preuve analytique a été établie de la conjugaison de ce réseau simple avec les applications quadratiques itérées [Wang, 1991]. Cela renforce d'autant plus l'aspect universel de la route vers le chaos par doublement de période.

Ces observations biologiques récentes et les résultats théoriques établis de modélisation d'une certaine façon spectaculaires amènent à reconsidérer les phénomènes non linéaires comme inhérents aux mécanismes de perception/décision pour certains réseaux de neurones spécialisés. Bien que restreints à des cas de réseaux bien particuliers, ces constats nous confortent dans l'idée que cette voie de modélisation dynamique est fructueuse et très proche de la réalité biologique. L'universalité des phénomènes observés (route vers le chaos par doublement de périodes) autorise selon nous des modèles extrêmement réducteurs des mécanismes internes d'entités biologiques

simples (comme les fourmis) ou dans des situations comportementales spécifiques (la perception visuelle dans le vol groupé d'oiseaux) en faisant l'hypothèse de comportements totalement réactifs.

Par conséquent le recours aux applications quadratiques itérées comme l'application logistique au coeur des processus de décision n'est pas une hypothèse de modélisation irréaliste. Les auteurs de l'article [Korn et Faure, 2003] mentionnent d'ailleurs en fin d'article les réseaux d'itérations couplées de Kaneko comme modèle de système complexe d'un grand intérêt pour l'étude des réseaux de groupes de neurones. L'application logistique est censée modéliser à un niveau mésoscopique un groupe de neurones pris comme une boîte noire. Son contrôle est assuré par le biais du paramètre de contrôle et son couplage avec les autres groupes de neurones par un paramètre de couplage global sur le réseau ou spécifique à la connexion entre deux groupes.

Nous nous situons entièrement dans cette démarche constructive de modélisation. Nous modélisons la boucle perception-décision-action de nos agents autour d'une application logistique ; en faisant cela, nous réduisons en quelques sortes un système neuronal sous-jacent en une forme plus simple et plus générique à manipuler : elle ne comporte que trois grandeurs, une variable de décision et deux paramètres (contrôle et couplage).

9

Synthèse

Cette partie consacrée à la notion d'agent et à la formalisation de notre modèle SMAL a permis de décrire en détails les principes et définitions fondateurs du SMA logistique. Celui-ci résulte de la fusion du modèle influence-réaction des systèmes multi-agents et des gaz d'itérations couplées. Là où les modèles stochastiques ont recours aux probabilités pour générer des phénomènes aléatoires présents dans les processus naturels, nous avons substitué des applications simples qui génèrent des séries chaotiques. Certes l'aléatoire qui en résulte n'est pas de même nature ; nous estimons cependant qu'il reste valide et pertinent, d'autant que le mécanisme chaotique générateur peut être relié à des observations biologiques sur des réseaux de neurones.

La notion de champ est à la base de notre modèle d'environnement. Tout est donc champ ici, y compris les positions des agents stockées par l'environnement. Les champs jouent le rôle de structures de données actives, qui peuvent évoluer indépendamment des agents (évaporation, diffusion), et qui peuvent être lus et écrits par les agents. Cette architecture à base de champs permet de concevoir des opérateurs de perception et d'action combinant éventuellement plusieurs champs à la fois. Enfin, les champs ont un rôle essentiel pour les mécanismes d'auto-organisation et d'adaptation par le fait qu'ils peuvent influer par le biais des perceptions des agents sur la variable de couplage ou la variable de contrôle au sein des agents.

Ainsi, notre modèle est simplificateur au bon sens du terme : il résume la mécanique de l'agent logistique à trois grandeurs internes principales : une grandeur de décision qui fixe les actions, une grandeur de contrôle qui module la décision et une grandeur de couplage qui prend en compte les autres agents du système, ainsi que des champs de données qui peuvent modifier ces grandeurs et être également modifiés en retour.

L'ensemble constitue une structuration cohérente et originale dans le domaine des SMA réactifs. Ce schéma théorique permet de rendre compte des divers comportements d'auto-organisation et d'adaptation dans le système, en fonction de l'environnement, des mécanismes de perception et d'action des agents. Les mécanismes associés doivent cependant être précisés, éprouvés et analysés. C'est tout l'objet de la partie suivante dédiée aux applications de ce modèle.

Quatrième partie

Mise en oeuvre du SMA logistique pour l'intelligence en essaim

Avant de commencer cette partie sur les applications de notre modèle SMAL à l'intelligence en essaim, présentons-en le plan et les objectifs.

Le SMAL hérite du modèle de calcul RIC issu des systèmes complexes. A ce titre, il est capable de générer des schémas de coordination intrinsèques au groupe d'agents, c'est-à-dire sans données externes dans l'environnement (donc pas d'adaptation non plus), que l'on associe aux phénomènes d'auto-organisation du modèle. Le SMAL est alors un système complexe au même titre que les réseaux d'itérations couplées. Une autre catégorie de situations implique un processus d'adaptation à l'environnement. Ces situations mettent en jeu des données externes déjà présentes dans un champ de l'environnement, qui agit sur les valeurs des variables internes de l'agent via les perceptions. Le SMAL devient alors un système complexe adaptatif et permet d'aborder des problèmes d'optimisation traités par l'intelligence en essaim. Nous procédons dans cette présentation en progressant des mécanismes et instances les plus simples aux plus compliquées, en ajoutant au fur et à mesure les briques élémentaires de construction [Charrier *et al.*, 2007b].

Le chapitre 10 traite donc en premier lieu des phenomènes auto-organisés liés au modèle SMAL dans sa version la plus simple. Nous y aborderons principalement deux cas de figure :

- Le cas où tous les agents logistiques sont dans un mode de comportement chaotique. Nous vérifions alors expériementalement la valeur d'un seuil théorique du couplage $\epsilon^* = 0,5$ pour l'émergence de la synchronisation dans le système.
- Le cas où les agents logistiques voient leur paramètre de contrôle tiré aléatoirement selon une loi uniforme sur $[0, 1]$. Dans ce cas l'apparition de clusters de synchronisation révèle une structure dynamique très instable qui s'apparente visuellement à des phénomènes de flocking.

Ces simulations sont analysées de façon poussée, par le calcul du nombre de clusters formés, mais aussi par une évaluation de l'entropie K vue en section 5.5 grâce à l'intégration d'agents jumeaux dans le système. Nous initions ainsi une nouvelle voie de simulation du flocking en situant la cause du phénomène dans la synchronisation des états internes des agents.

Sur cette base, le chapitre 11 focalise ensuite sur les mécanismes d'adaptation possibles avec le SMAL. Nous repartons simplement de l'agorithme basique de flocking vu au chapitre 10 pour intégrer des mécanismes simples d'adaptation à l'environnement. Un premier effet visible est la formation de files d'agents dans l'environnement, cas étudié mathématiquement. Puis en modifiant légèrement l'instance précédente, nous présentons une version d'algorithme proie-prédateurs où une ou plusieurs proies sont entourées et stoppées par un essaim de prédateurs. Nous montrons que ce problème est assimilable à un problème de recherche de zéros dans un champ dynamique non-linéaire et s'apparente à certains problèmes typiques traités avec les algorithmes PSO. Enfin, une version plus sophistiquée d'algorithme de flocking inspirée à la fois des modèles de Reynolds et de Vicsek (cf. section 1.3) est proposée en fin de chapitre qui sert de base de modélisation pour réaliser un flocking de robots en fin de cette partie.

Le chapitre 12 expose notre modèle de "fourmi logistique" et considère l'application du SMAL aux problèmes d'optimisation tels ceux que l'intelligence en essaim traite déjà, notamment le problème du voyageur de commerce (TSP) abondamment étudié, mais qui constitue un bon problème test pour notre méta-heuristique SMAL. Il s'agira dans un premier temps de reprendre

l'expérience du double pont des fourmis pour introduire sur cette situation archétypale le principe de notre agorithme adaptatif. Puis la situation sera généralisée à la prise de décision sur un noeud d'un graphe quelconque pour préparer la situation d'un problème de plus court chemin dans un graphe. La résolution approchée du problème TSP est alors envisagée : nous y comparons les performances de l'algorithme par colonies de fourmis "Ant System" (AS) qui est un des premiers, à celles de notre algorithme SMAL. Notre objectif n'est pas à ce niveau de rivaliser avec l'algorithme AS, mais de situer le SMAL en termes de performances et de jauger de son potentiel et de ses avantages.

Nous terminons les applications par un quatrième chapitre court qui a valeur de perspective car ce travail est encore en cours de finalisation, consacré à la robotique en essaim, montrant comment nous avons pu réaliser un flocking de robots en implémentant l'algorithme de flocking à base d'agents logistiques vu au chapitre 11 Cette partie s'achève sur un chapitre de discussion.

10

Phénomènes d'auto-organisation

10.1 Synchronisation du SMAL chaotique

Nous étudions dans ce chapitre les propriétés auto-organisationnelles su SMAL [Charrier *et al.*, 2007a]. Un premier cas nous intéresse, celui où les agents logistiques sont tous chaotiques. Ce cas est important puisqu'il correspond à la situation initiale d'exploration pour la plupart des problèmes que nous abordons ; cet aspect exploratoire est notamment crucial pour les problèmes d'optimisation. Ce cas consiste concrètement à plonger la dynamique interne des agents dans l'espace géométrique de l'environnement.

Instanciation du SMAL.

Voici les spécifications du modèle dans ce cas :

L'environnement. L'environnement d'état σ est composé :

- de l'espace E qui est le tore [47] continu de dimension 2 de taille $L \in \mathbb{R}$ avec $L > 1$ que nous noterons $\mathbb{T}^2_L$,
- des champs définis en section 8.2.4 dont les champs de position individuelles δi, de position cumulée $\mathcal{N}$ et de décision $\mathcal{X}$, qui sont non persistants et donc mis à 0 d'un pas de temps à l'autre avant mis à jour avec les nouvelles valeurs.

On associe à l'espace torique E un système de coordonnées polaires $(\mathbf{r}, \theta)$ plus adaptées à la formulation du cas traité ici [48]. Cependant, θ n'est pas vraiment l'angle qui nous intéresse directement dans notre cas, mais un angle α qui donne la nouvelle direction de déplacement à chaque pas de temps (voir la figure 10.2).

[47] Il est défini à partir du tore unité $\mathbb{T}^2 = \frac{\mathbb{R}^2}{\mathbb{Z}^2}$ que l'on transforme par homothétie de rapport L.

[48] Ce systèmes est relié aux coordonnées cartésiennes (X, Y) par les formules suivantes :

$$\begin{cases} X = r\ \cos(\theta) \\ Y = r\ sin(\theta) \end{cases}$$

L'agent logistique. L'agent logistique A_i est tel que :

– son état à t s'écrit :

$$s_i(t) = \langle x_i(t), a^0, \epsilon^0 \rangle \tag{10.1}$$

ce qui précise l'écriture générique de définition de l'état en fonction du problème. En effet le cas que nous étudions ici ne fait intervenir aucune adaptation à l'environnement, ce qui a pour corollaire que les variables de contrôle et de couplage sont initialisées de façon globale (indépendamment des agents) et restent stationnaires au cours du temps.

– L'application logistique f utilisée prend ici sa forme la plus classique dont le diagramme de bifurcation est donné en figure 10.1 :

$$\begin{aligned} f : \quad \mathbb{D} \times \mathbb{D} \quad &\to \mathbb{D} \\ (x, a) \quad &\mapsto f(x, a) = 4\ a\ x\ (1 - x) \end{aligned} \tag{10.2}$$

Nous rappelons que $\mathbb{D} = [0, 1]$.

Nous posons $a^0 = 1$ pour que les agents soient tous chaotiques.

– Ainsi, le changement d'état de l'agent s'écrit :

$$a_i^{t+1} = a^0 = 1 \tag{10.3}$$

$$\epsilon_i^{t+1} = \epsilon^0 \tag{10.4}$$

$$x_i^{t+1} = (1 - \epsilon^0) f(x_i^t, a^0) + \epsilon^0\ P_{x,i}(\sigma^t) \tag{10.5}$$

Le processus de perception-décision-action se réalise en deux temps comme expliqué dans 8.2.7, mais nous synthétisons dans la suite ce processus aux points essentiels.

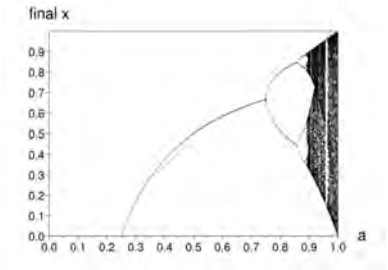


FIG. 10.1 – Diagramme de bifurcation de l'application logistique
$x_{t+1} = f(x_t, a) = 4\ a\ x_t\ (1 - x_t)$ après 500 itérations sur 500×500 échantillons des couples $(a, x^0) \in ([0, 1] \times [0, 1])$

La perception. La perception des agents est limitée sur un voisinage de rayon R. $P_{x,i}(\sigma^t)$ prend la forme déjà explicitée en 8.24 où l'agent A_i perçoit N_i voisins dans son voisinage V_i :

$$P_{x,i}(\sigma^t) = \begin{cases} \langle \mathcal{X} \rangle_{V_i} = \dfrac{\sum_{j \in V_i}\ f(x_j^t, a^0)}{N_i^t} & \text{, si } N_i \neq 0 \\ f(x_i^t, a_i^{t+1}) & \text{sinon} \end{cases} \tag{10.6}$$

Équation maîtresse de la décision. En intégrant l'écriture de la perception 10.6 dans 10.5, on obtient l'équation maîtresse pour la décision de l'agent si $N_i \neq 0$:

$$x_i^{t+1} = (1-\epsilon^0) f(x_i^t, a^0) + \epsilon^0 \, \frac{\sum_{j \in V_i} f(x_j^t, a^0)}{N_i^t} \tag{10.7}$$

qui est à peu de choses près l'équation maîtresse des gaz d'itérations couplées 5.24. Il est clair ici que cette équation laisse le domaine de définition $\mathbb{D}$ de x invariant comme somme convexe de deux éléments de $\mathbb{D}$: une valeur dans $\mathbb{D}$ d'une part et une moyenne sur des éléments de $\mathbb{D}$ d'autre part.

Les actions de l'agent. Après la mise à jour la variable de décision x_i, les actions sont effectuées en relation directe avec sa nouvelle valeur x_i^{t+1}. Les actions effectuées par l'agent sont de deux ordres : se déplacer et marquer le champ de décision non persistant d'influence $\widetilde{\mathcal{X}}$.

– calculer les nouvelles coordonnées sur E selon la loi de déplacement :

$$\begin{cases} \alpha_i^{t+1} & = 2\pi \; x_i^{t+1} \\ \mathbf{r}_i^{t+1} & = \mathbf{r}_i^t + v_0 \; \mathbf{u}_{\alpha_i^{t+1}} \\ \theta_i^{t+1} & = \arg(\mathbf{r}_i^{t+1}) \end{cases} \tag{10.8}$$

où v_0 correspond à la norme de la vitesse de l'agent pris comme une constante identique pour tous les agents, $\mathbf{u}_{\alpha_i^{t+1}}$ est le vecteur directeur de la vitesse d'angle α_i^{t+1} et arg est la fonction qui renvoie l'angle polaire θ à partir du rayon vecteur $\mathbf{r}$.
La figure10.2 montre la construction de ces grandeurs d'un pas de temps à l'autre.

– modifier les champs d'influence des agents correspondants.

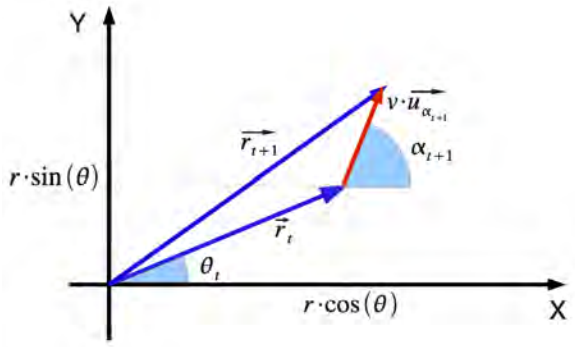


FIG. 10.2 – Schéma récapitulatif du changement de position de l'agent chaotic.

Réaction de l'environnement. L'environnement procède dans la phase de réacion à la mise à jour synchrone de tous les champs à partir des champs d'influence modifiés par les agents, au moyen d'opérateurs de sommation dans le cas des positions cumulées (cf. 8.2.8).

Simulations et analyse.

Objectif de la simulation. A partir des lois d'évolution précédentes, un algorithme est établi (cf. algorithme 10.1).

L'initialisation du système est semblable à celle du modèle de Vicsek : on place les agents

Algorithm 10.1: Algorithme du SMAL chaotique.

Initialisation :
begin
  for $i = 1$ *to* N / *pour chaque* A_i **do**
    $\left(X_i^0, Y_i^0\right) \leftarrow (random() * L, random() * L)$
    $\widetilde{\delta}_i\left(X_i^0, Y_i^0\right) \leftarrow 1$
    $\theta_i^0 \leftarrow 2\pi * random()$
end
repeat
  Processus sensorimoteur individuel :
  for $i = 1$ *to* N / *pour chaque* A_i **do**
    $y_i \leftarrow f(x_i^t, a_0)$
  Processus sensorimoteur collectif :
  for $i = 1$ *to* N **do**
    $P_{x,i} \leftarrow \dfrac{\sum_{j \in V_i} f(x_j^t, a_0)}{N_i^t}$
    $x_i^{t+1} \leftarrow (1 - \epsilon_0) f(x_i^t, a_0) + \epsilon_0 \; P_{x,i}(\sigma^t)$
    $(\alpha_i^{t+1}, \mathbf{r}_i^{t+1}) \leftarrow (2\pi \; x_i^{t+1}, \mathbf{r}_i^t + v \; \mathbf{u}_{\alpha_i^{t+1}})$
    $\left(X_i^{t+1}, Y_i^{t+1}\right) \leftarrow \left(X_i(\theta_i^{t+1}, \mathbf{r}_i^{t+1}), Y_i(\theta_i^{t+1}, \mathbf{r}_i^{t+1})\right)$
    $\widetilde{\delta}_i\left(X_i^{t+1}, Y_i^{t+1}\right) \leftarrow 1$
  Réactions de l'environnement :
  Mise à jour des champs non persistant de position, de positon cumulée, de décision.
until *stop*

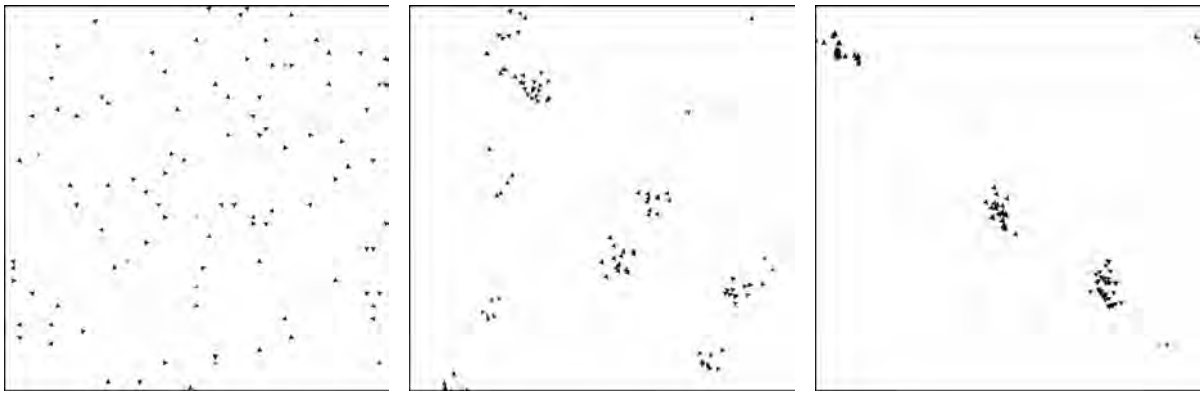
au hasard dans l'espace avec un angle de déplacement initial également tiré aléatoirement dans $[0, 2\pi]$.

Des simulations ont été réalisées à partir de cet algorithme dans l'objectif de vérifier la valeur théorique du seuil de synchronisation partielle. Rappelons que le mécanisme de synchronisation présenté en section 5.4.2 prévoit un seuil de couplage théorique ϵ^* au-delà duquel la synchronisation apparaît, y compris dans des topologies de connexion non stationnaires comme c'est toujours le cas avec le SMAL. Une conséquence est que le nombre d'agents N_i dans le voisinage d'un agent A_i est une variable dépendant du temps. Rappelons la loi théorique de calcul de ce seuil (cf équation (5.23)) :

$\epsilon_* = 1 - \exp(-\lambda_0)$

Comme tous les agents logistiques ont une variable de contrôle $a^0 = 1$, et sachant que l'application logistique a pour exposant de Lyapunov $\lambda_0 = \ln 2$, on déduit que le seuil doit être égal à $0,5$. On sait que la détection de ce seuil sur ordinateur peut poser problème à cause de la limitation en précision des calculs [Pikovsky *et al.*, 2001].

Résultats de simulation. Un exemple de simulation est donné sur la figure 10.3 dans le cas où la taille de l'environnement est $L = 30.0$, le rayon de perception est $R = 4.0$, le nombre total d'agents est $N = 100$, le couplage $\epsilon^0 = 0.9$, la vitesse intiale $v_0 = 1$.



(a) $t = 1$ (b) $t = 500$ (c) $t = 2100$

FIG. 10.3 – Images de simulations à différents pas de temps où $a_0 = 1$ et $\epsilon^0 = 0.9$ pour chacun des $N = 100$ agents. De plus le rayon de voisinage de perception est $R = 4$.

Signalons que le mouvement des agents chaotiques avec l'application logistique n'est pas isotrope, i.e. invariant par rapport à la direction. En effet, il suffit de consulter la distribution de l'application pour $a = 1$ pour constater qu'elle est non uniforme sur $[0, 1]$ (cf. annexe B). Elle présente même deux pics en 0 et en 1, les limites de l'intervalle. Transformé en angles de déplacement, cela implique que les agents vont en moyenne dans une direction fixée, celle correspondant à l'angle 0 de l'environnement, tout en maintenant des déplacements transverses aléatoires moins fréquents.

On constate néanmoins qu'au cours du temps, des regroupements d'agents s'opèrent dans l'espace et finissent par se concentrer en des clusters de synchronisation. Mais comme dans le cas des réseaux d'itérations aléatoirement couplées (GCM), la synchronisation n'est jamais totale, elle est partielle et fragmentée en plusieurs clusters "flous" de synchronisation [Manrubia *et al.*, 2004]. On le constate également sur le graphique temporel 10.4(a) où sont portées en ordonnée les valeurs x internes de tous les agents. Au pas de temps $t = 2700$ le nuage de points s'éclaircit tout en conservant une densité de distribution à peu près stable. On constate également que ces synchronisations spatiales sont fragiles, puisqu'entre les pas de temps $t = 2100$ et $t = 2700$ existe une zone d'instabilité liée à des bouffées chaotiques.

Pour vérifier et mesurer cet effet de regroupement d'agents synchrones, nous calculons le nombre de clusters formés à chaque pas de temps pour en obtenir l'évolution moyenne.

Mesure du nombre de clusters de synchronisation.

Calcul du nombre de clusters. Tout d'abord, il convient de définir un cluster de synchronisation partielle.

Définition 10.1.1. *Un cluster de synchronisation d'agents logistiques à la précision γ est défini à l'instant t par la propriété d'appartenance suivante : deux agents A_i et A_j appartiennent au même cluster si l'un se trouve dans le voisinage de l'autre et s'ils vérifient de plus la relation : $\left|x_i^t - x_j^t\right| < \gamma$.*

L'algorithme de calcul correspondant procède par récursion :

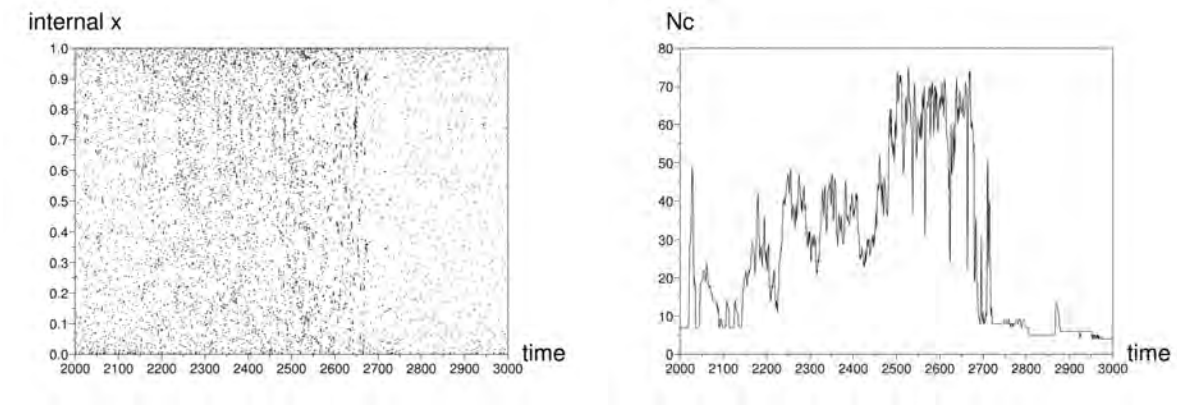
1. au départ un agent n'appartient à aucun cluster,
2. on répète une boucle de tirage aléatoire des agents tant qu'il reste des agents sans cluster (ces agents jouent le rôle de germes pour de nouveaux clusters)
3. à partir d'un agent tiré au sort, on calcule le cluster auquel il est associé en appliquant la propriété d'appartenance ci-dessus récursivement dans le voisinage des agents de façon semblable au parcours d'un graphe en profondeur d'abord.
4. les agents sélectionnés sont ajoutés au cluster.

Algorithm 10.2: Calcul des clusters de synchronisation

```
Initialisation
for i = 1 to N do
    estDansCluster(A_i) ← false
ListeClusters ← null
Processus récursif de calcul de clusters :
while ∃i tel que estDansCluster(A_i) = false do
    estDansCluster(A_i) ← true
    Cluster_i ← nouveauCluster
    ajouter Cluster_i à ListeClusters
    ajouter A_i à Cluster_i
    rechercheCluster(A_i,Cluster_i)
```

```
Fonction rechercheCluster(A_i,Cluster_i) :
foreach A_j ∈ V_{A_i} do
    if estDansCluster(A_j) = true then
        break
    else
        if |x_i^t − x_j^t| < γ then
            estDansCluster(A_j) ← true
            ajouter A_j à Cluster_i
            rechercheCluster(A_j,Cluster_i)
```

Soit $N_c(t)$ le nombre de clusters à l'instant t. La figure 10.4(b) montre l'évolution au cours du temps du nombre de clusters lors de la synchronisation. Lorsque le système est non synchronisé, ce nombre est égal au nombre d'agents, lorsque le système se synchronise par clusters, il décroît vers 1 sans l'atteindre puisque l'on n'a pas de synchronisation totale.



(a) Visualisation de la synchronisation temporelle : x est affiché en fonction du temps.

(b) Nombre de clusters formés au cours du temps.

FIG. 10.4 – Simulation avec $\epsilon^0 = 0.95$, et un rayon de perception $R = 4$ pour tous les agents. La transition de synchronisation se produit au pas de temps $t = 2700$.

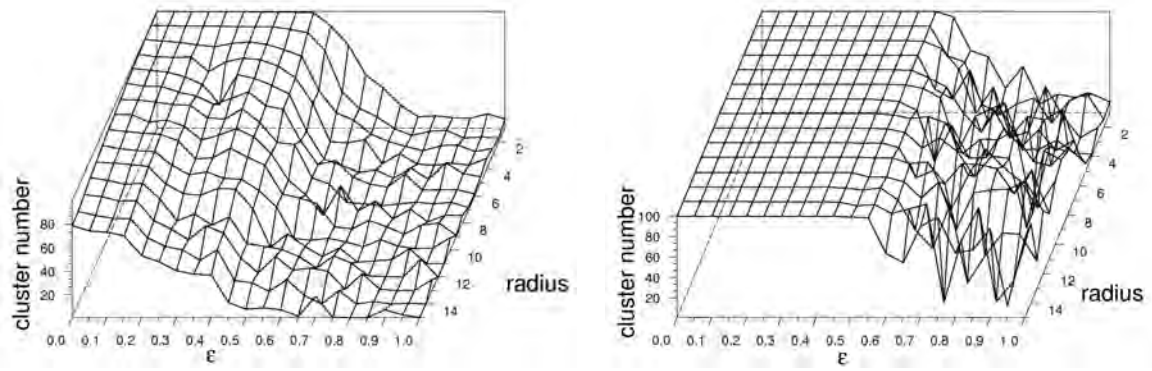
Mesure de la moyenne du nombre de clusters. La mesure que nous utilisons consiste à calculer la moyenne de $N_c(t)$ sur 500 pas de temps après une phase transitoire de 2500 pas de temps, soit :

$$\bar{N_c} = \frac{1}{500} \sum_{t=2501}^{t=3000} N_c(t) \tag{10.9}$$

Une série de simulations a été effectuée pour déterminer comment cette moyenne évolue en fonction des paramètres du système, à savoir la taille du voisinage de perception caractérisée par son rayon $R \in [1, 15]$ et la valeur du couplage $\epsilon^0 \in [0, 1]$. La valeur maximale du rayon de voisinage, moitié de la taille de l'environnement, correspond quasiment à une situation de perception totale des autres agents. Deux séries ont été lancées, une pour laquelle la précision de segmentation en clusters $\gamma = 10^{-2}$ est grossière, l'autre plus fine à $\gamma = 10^{-10}$. Les résultats sont donnés respectivement sur les figures 10.5(a) et 10.5(b). Les figures 10.6(a) et 10.6(b) donnent la déviation standard dans les deux cas.

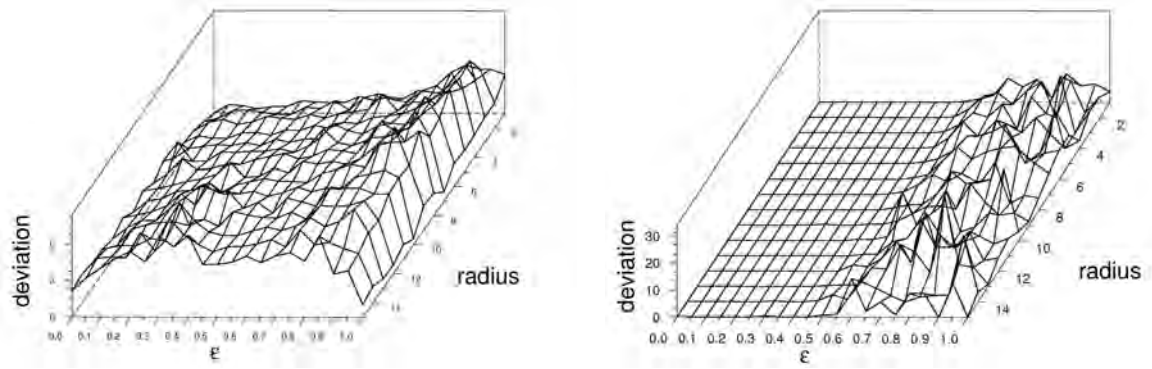
Analyse des résultats. La précision γ employée pour la détection des clusters joue le rôle d'une loupe d'observation du phénomène de synchronisation. Le premier constat est qu'à faible précision, la détection des clusters est très floue et permet de ne tirer que des conclusions générales sur ce qui ressort le plus nettement. Ainsi peut-on remarquer clairement que l'augmentation du couplage renforce la synchronisation quel que soit le voisinage de perception. Même avec des voisinages faibles une synchronisation partielle a lieu. D'autre part, lorsque ϵ^0 reste faible, un rayon de perception plus grand tend à rendre plus difficile la synchronisation. Et l'-analyse de la déviation standard semble renforcer cette observation : elle augmente à mesure de l'augmentation du rayon de perception et présente des variations très brutales, ce qui tendrait à montrer que les clusters sont globalement moins stables, se scindant et se recomposant plus rapidement. Ce fait est également confirmé en haute précision.

A haute précision de détection des clusters, le graphique présente une image plus fidèle de



(a) $\bar{N}_c$ à la précision 10^{-2} (b) $\bar{N}_c$ à la précision 10^{-10}

FIG. 10.5 – Cas des agents chaotiques : nombre moyen de clusters formés en z, en fonction du couplage ϵ^0 en y et du rayon de voisinage de perception R en x pour $N = 100$ agents.



(a) Précision $\gamma = 10^{-2}$ (b) Précision $\gamma = 10^{-10}$

FIG. 10.6 – Déviation standard sur le nombre moyen de cluster $\bar{N}_c$ en z en finction du couplage en y et du rayon de voisinage en x pour $N = 100$ agents.

la situation. Ainsi voit-on clairement apparaître la valeur du seuil de synchronisation sur la figure 10.5(b) : lorsque $\epsilon^0 < 0,5$ le nombre moyen de clusters $\bar{N}_c$ reste égal au nombre d'agents, ce qui signifie qu'il n'y a pas de synchronisation partielle en deçà de cette valeur. Ce n'est qu'à partir de $\epsilon^0 = 0,5$ que le nombre de clusters commence à diminuer, révélant la présence d'une synchronisation partielle. Ce constat est confirmé par les valeurs de déviation standard sur la figure 10.6(b) qui est quasi nulle pour $\epsilon^0 < 0,5$ ce qui montre la stabilité de ce seuil de synchronisation. Ces valeurs de déviation sont en revanche parfois très élevées dès que $\epsilon^0 > 0,5$, ce qui suppose que l'existence des clusters est fragile à ce niveau de précision d'observation. Cette analyse se renforce dès que $R \gtrsim 10$ où les plus hautes valeurs de déviation apparaissent. On peut émettre l'hypothèse qu'il est en moyenne plus difficile de constituer des clusters lorsque les agents ont une perception plus grandes, ce qui implique des interactions avec les autres agents plus nombreuses et par conséquent la formation de clusters de plus grandes tailles.

Discussion.

Seuil de synchronisation. Dans le cas particulier des agents logistiques tous chaotiques, nous avons pu vérifier la valeur théorique du seuil de synchronisation $\epsilon^* = 0,5$ donnée par la loi du seuil de synchronisation pour les réseaux d'itérations globalement couplées (cf équation (5.23)). Contrairement à ces réseaux, la synchronisation n'est pas totale mais uniquement partielle et donne lieu à des clusters de synchronisation à une précision γ près, comme le modèle des réseaux d'itérations aléatoirement couplées (RCM) le prévoit.

Traduction spatiale de la synchronisation interne. D'autre part, l'action de déplacement des agents a traduit une synchronisation d'origine interne en un regroupement spatial visible dans l'environnement. Cette traduction d'un phénomène interne aux agents en une manifestation externe géométrique dans l'environnement —via l'action—, est un principe véritablement central dans le SMAL. Ce mécanisme place le phénomène de synchronisation au coeur de nombreux phénomènes de groupes. Cette conjecture s'étend même selon nous à d'autres phénomènes de l'intelligence en essaim. Cela constitue une des conclusions majeures de cette thèse.

Lien avec les réseaux aléatoires. Notons pour finir que les agents logistiques fonctionnent ici avec une variable de décision de dimension 1 pour effectuer un déplacement dans un espace de dimension 2. Cette situation est très similaire à celle des réseaux aléatoirement couplés. Ce parallèle s'établit assez bien sur les premiers pas de temps d'une simulation où les agents sont placés aléatoirement dans l'espace et les clusters ne sont pas encore formés. En effet en reprenant l'équation maîtresse des RCM (cf.section 5.4.2), on peut évaluer l'ordre de grandeur du taux moyen de connexions κ dans le SMAL. Comme ce taux correspond à la probabilité d'un agent de se trouver dans le voisinage d'un autre, on peut ramener l'espace physique à un espace probabilisé à n cellules de taille $\frac{L^2}{S_V}$ où S_V est la surface du voisinage d'un agent. La probabilité cherchée revient donc à la probabilité de choisir une cellule parmi les n d'où :

$$\kappa_{SMAL}^{t_0} = \frac{S_V}{L^2} \tag{10.10}$$

Comme le voisinage de perception ne peut être supérieur à l'espace lui-même, on a bien : $\kappa_{SMAL}^{t_0} < 1$. Cette valeur n'est plus vraie dès que $t > 0$ car les interactions entre agents modifient les lois de distributions spatiales de l'ensemble du groupe.

Bilan.

Cette première expérimentation du SMAL montre un des phénomènes complexes typiques de ce modèle, la synchronisation chaotique étudiée par ailleurs sur les versions réseaux des RIC. Ce phénomène de synchronisation est central dans les interprétations associées au SMAL. La synchronisation émane d'un effet de couplage interne aux agents et se traduit dans l'espace physique ou géométrique de l'environnement.

L'analyse théorique en section 5.4.2 basée sur le calcul du coefficient de Lyapunov transverse à l'espace stable est appliquée sur une topologie de connexions non stationnaires, et les seuils de synchronisation restent inchangés, mais la nature de la synchronisation observée change. Une mesure de la stabilité des clusters de synchronisation qui se forment, est nécessaire pour en savoir plus sur le temps de vie d'un cluster. La section suivante propose une mesure approchée de la stabilité des clusters formés dans le SMAL.

10.2 Mesure de la stabilité

L'entropie K de Kolmogorov, introduite en section 5.5, nécessite le calcul ou l'estimation du spectre de Lyapunov du système. L'estimation de cette entropie constitue un critère central pour l'étude de la synchronisation en clusters vu en section 10.1. La question de la stabilité des clusters formés est naturelle puisque l'on a des clusters de synchronisation partielle. On souhaite pouvoir prévoir au moins sur un certain horizon temporel la stabilité d'un cluster chaotique donné et donc d'une certaine façon sa durée de vie. Autrement dit, on veut savoir à quel moment un cluster donné est susceptible de se déstructurer et de se dissocier en plusieurs parties.

L'estimation de l'entropie K passe par l'estimation des exposants de Lyapunov du système. Nous présentons ici une technique fondée sur le calcul individuel du coefficient de Lyapunov maximal pour chaque agent, considérant que les agents constituent les degrés de libertés du système, pour approcher la valeur de cette entropie.

10.2.1 Des agents jumeaux pour étudier la stabilité des clusters.

Malgré la dimension 1 de l'application logistique de nos agents, les couplages nombreux et variables compliquent le calcul analytique du spectre de Lyapunov du système. C'est pourquoi, nous utilisons une méthode de calcul spécifique aux données expérimentales d'une part, et d'autre part nous considérons une estimation individu-centrée des coefficients de Lyapunov du système.

Principe de calcul. Le principe de l'algorithme que nous avons mis en oeuvre pour obtenir les exposants de Lyapunov est tiré de l'algorithme de Wolf [Wolf *et al.*, 1985]. Cet algorithme vise à approcher le calcul de l'exposant de Lyapunov maximal λ_m d'un système à partir de la série

temporelle qu'il génère en évitant les dépassements de précision des calculs. Il s'agit d'un calcul approché prenant en compte l'évolution à intervalle de temps τ d'une variation infinitésimale $\vec{\gamma}(t)$ en un point $\vec{x}(t)$ de la trajectoire (cf. figure 10.7).

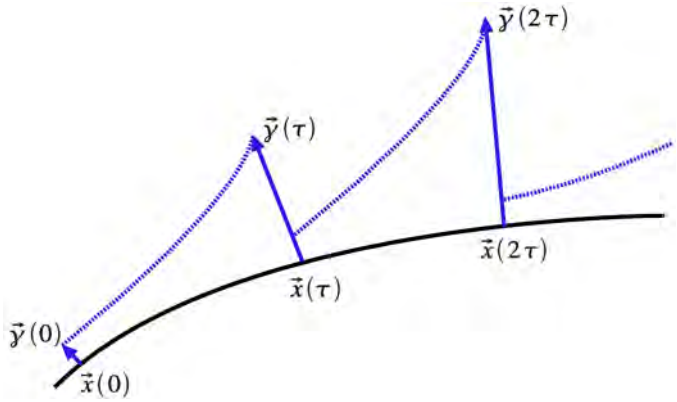


FIG. 10.7 – Méthode de calcul approché du plus grand coefficient de Lyapunov qui évite les dépassements des limites de calcul.

$$\begin{aligned}
\vec{\gamma}(\tau) &= \vec{\gamma}(0)\exp(\lambda_1\ \tau) \\
\vec{\gamma}(2\tau) &= \frac{\vec{\gamma}(\tau)}{\|\vec{\gamma}(\tau)\|}\exp(\lambda_2\ \tau) \\
&\cdots \qquad (10.11)
\end{aligned}$$

$$\lambda_m = \lim_{n\to\infty}\frac{1}{n}\sum_{k=1}^{n}\lambda_k \qquad (10.12)$$

On ramène la variation à un vecteur normalisé à chaque réinitialisation de l'échantillon pour éviter une dérive progressive des vecteurs de variation.

Les agents jumeaux. Revenons à présent à notre étude des clusters de synchronisation partielle. L'agent logistique décide selon une variable scalaire, donc à une dimension, ce qui simplifie l'algorithme de calcul précédent. Nous considérons l'effet subi du point de vue de chaque agent indépendamment des autres lorsqu'une petite variation vient modifier cette variable de décision scalaire. Ce faisant, nous opérons donc une réduction de la dimension du problème au calcul du coefficient de Lyapunov pour chaque agent, ce qui fournit au final une sorte de spectre de Lyapunov individu-centré de l'essaim d'agents. Ce n'est en toute rigueur pas le spectre de Lyapunov du système. Il faudrait pour l'obtenir considérer le système dans son écriture dynamique la plus globale. Cela dit, cette alternative offre une mesure intéressante et plus facile à calculer.

Reste à calculer ce coefficient selon le principe exposé précédemment. Pour cela, nous introduisons la notion d'agent jumeau :

Définition 10.2.1. *L'agent jumeau noté A_{i^*} d'un agent A_i de variable de décision x_i est un agent de variable de décision x_{i^*}, initialisé au temps t_0 avec des caractéristiques en tous points*

identiques à l'agent jumelé, excepté sa variable interne de décision initialisée à $x_{i^*}^{t_0} = x_i^{t_0} \pm \gamma_0$ *de façon que l'on ait toujours l'inégalité* $(0 < x_{i^*}^{t_0} < 1)$ *vérifiée, avec la variation infinitésimale* $\gamma_0 \ll 1$.

Pour retrouver le calcul du paragraphe précédent pour un agent A_i, on opère l'algorithme suivant au temps $t_0 + k\tau$:

1. on initialise l'agent jumeau A_{i^*} à $t_0 + k\tau$, sa variable interne ne diffère que de γ_0 de son frère A_i,
2. on relève la valeur $x_{i^*}^{t_0+(k+1)\tau}$ après avoir laissé l'agent i^* évoluer dans le même contexte que i pendant τ pas de temps,
3. on calcule le taux d'évolution pendant ces τ pas de temps représenté par la variable :

$$\delta_{k+1} x_i = \frac{|x_{i^*}^{t_0+(k+1)\tau} - x_i^{t_0+(k+1)\tau}|}{\gamma_0} \tag{10.13}$$

4. on calcule le coefficient de Lyapunov partiel :

$$\lambda_{k+1} = \log\left(\delta_{k+1} x_i\right) \tag{10.14}$$

5. on incrémente k et on recommence au point 1.

On récupère ainsi, un ensemble de coefficients partiels qui nous renseignent sur la stabilité partielle de chaque agent. Le calcul de la moyenne par la formule (10.12) donne l'évaluation à long terme du coefficient maximal de l'agent. En fait, étant donné le caractère très fluctuant de la dynamique de nos simulations, nous nous contentons d'une moyenne approchée sur quelques dizaines de séquences de τ pas de temps pour calculer le spectre, avec τ de l'ordre d'une dizaine de pas de temps. On note $\widehat{\lambda_{m,i}}(p)$ ce coefficient de Lyapunov partiel sur p séquences de calcul pour l'agent i :

$$\widehat{\lambda_{m,i}}(p) = \frac{1}{p} \sum_{k=1}^{p} \lambda_k \tag{10.15}$$

Calcul de l'entropie K partielle. A partir de là, et en considérant que l'entropie K est la somme des coefficients de Lyapunov positifs du système, on peut calculer l'entropie K partielle pour p séquences de τ pas de temps par :

Définition 10.2.2. *L'entropie K partielle notée $\widehat{K}$ sur p séquences de τ pas de temps est définie en tenant compte des coefficients de Lyapunov partiels calculés par la formule* (10.15) *sur un groupe G d'agents :*

$$\widehat{K}_p = \sum_{i \in G} \begin{cases} \widehat{\lambda_{m,i}}(p) \ \textit{si} \ > 0 \\ 0 \ \textit{sinon.} \end{cases} \tag{10.16}$$

Discussion. Ce calcul de coefficients de Lyapunov individus-centrés ne correspond pas en toute rigueur aux coefficients du système dynamique global. Cette technique ne donne que des valeurs approchées d'une part, mais est fondé sur l'*a priori* que chaque agent constitue un degré de liberté du système et peut être traité indépendamment des autres. Ceci dit, d'une certaine

façon, nous remontons à la source de définition des coefficients de Lyapunov issue de techniques variationnelles : nous nous plaçons du point de vue d'un agent particulier, et nous mesurons l'écart résultant sur son état interne après un certain nombre de pas de temps d'une petite variation de départ.

Le recours à un agent jumeau permet de prendre en compte dans une certaine mesure seulement l'influence de l'interaction avec les autres agents. En effet l'agent jumeau n'est pas perceptible par les autres agents et sa présence n'est que fantôme ; en toute rigueur le calcul de la variation devrait également prendre en compte les interactions de l'agent jumeau, ce qui n'est pas le cas dans notre approche pour des raisons de coût computationnel. La prise en compte complète serait lourde en coût computationnel puisque pour chaque agent jumeau il faudrait générer une simulation spécifique sur τ pas de temps. La complexité temporelle de l'algorithme se verrait donc multipliée par le nombre d'agents, puisqu'il faudrait au total $(N+1)$ simulations (N simulations des jumeaux plus la simulation principale).

Cette vision individu-centrée du calcul apporte malgré tout une information utilisable pour caractériser les clusters, en identifiant les agents intrinsèquement instables. Ces coefficients de Lyapunov partiels sont finalement une grandeur assez souple d'utilisation dans notre cadre d'étude.

10.2.2 Application au suivi des clusters de synchronisation partielle.

Dans toutes les simulations qui suivent [49], les agents sont tous chaotiques et les paramètres précédents sont fixés comme suit :

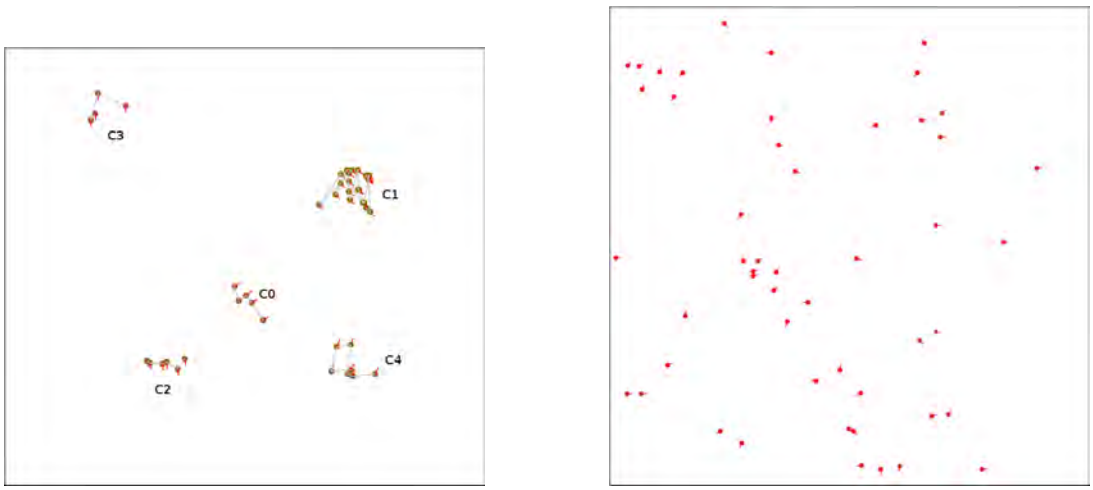
- $\tau = 5$, $p = 10$ et $\gamma_0 = 10^{-10}$,
- les calculs sont effectués en quadruple précision (128 bits) pour les flottants.
- il y a $N = 50$ agents dont le rayon de perception est $R = 5$ dans un environnement torique de taille 100×100.

On calcule donc une estimation de l'entropie partielle sur 50 pas de temps, ce qui permet de suivre les évolutions de cette grandeur. Nous utiliserons également la moyenne des coefficients de Lyapunov partiels sur un cluster C_j occupé par $|C_j|$ agents, notée et calculée par :

$$\overline{\lambda_{m,j}}(p) = \frac{\sum_{i \in C_j} \widehat{\lambda_{m,i}}(p)}{|C_j|} \tag{10.17}$$

Spectre de Lyapunov individus-centré approché. Une première visualisation est possible avec le spectre de coefficients de Lyapunov partiels calculés avec l'algorithme précédent. La figure 10.2.2 montre deux simulations au bout de 18000 pas de temps, de gauche à droite : une avec un couplage $\epsilon = 0,7$ supérieur au seuil de synchronisation (qui est de $0,5$), l'autre avec $\epsilon = 0,4$ inférieur. On constate par la seule couleur des agents sur ces graphiques —le rouge indiquant les valeurs positives du coefficient de Lyapunov de l'agent, le vert indiquant les valeurs négatives— qu'une valeur de couplage sous le seuil ne synchronise pas les agents (on a exactement 50 clusters, chaque agent constituant son propre cluster), tandis qu'une valeur au dessus les synchronise. Le tableau 10.1 est relatif à la situation de la figure 10.8(a) présentant 5

[49] Simulations et calculs réalisés sur le simulateur multi-agent MASON [Luke *et al.*, 2003]



(a) Cas où $\epsilon = 0,7$.

(b) Cas où $\epsilon = 0,4$.

FIG. 10.8 – Clusters de synchronisation après environ 18000 pas de temps. La couleur rouge des agents est associée aux valeurs positives de leur coefficient de Lyapunov, la couleur verte indique les valeurs négatives. Les arcs entre agents relient les agents d'un même cluster.

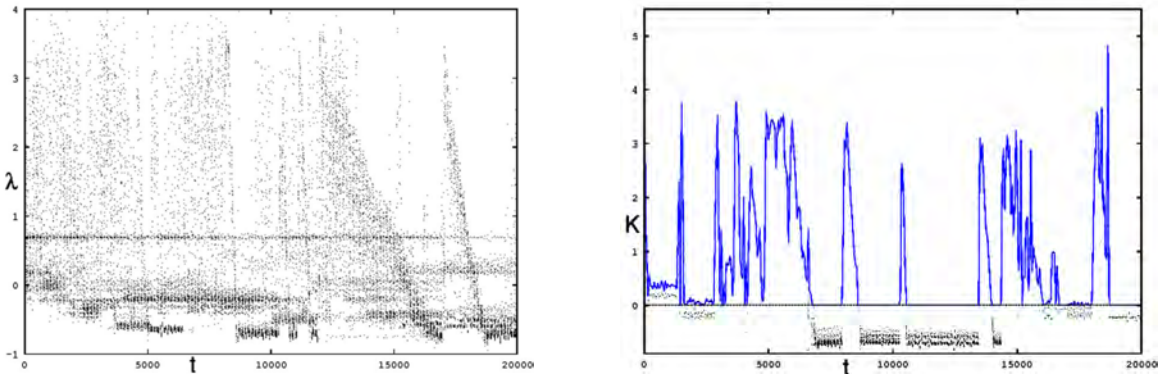
Cluster	nombre d'agents	$\widehat{K}_p$	$\overline{\lambda_{m,j}}(p)$
C_0	5	0	$-0,133$
C_1	24	$0,850$	$-0,101$
C_2	6	0	$-0,132$
C_3	4	$0,308$	$0,077$
C_4	7	0	$-0,206$

TAB. 10.1 – Valeurs du calcul de l'entropie K partielle et de la moyenne des coefficients partiels pour les 5 clusters visualisés. Il manque 4 agents qui sont isolés dans l'environnement.

clusters. Le résultat su calcul de l'entropie K partielle et de la moyenne des coefficients sur chaque cluster sont donnés arrondis à la troisième décimale. Les clusters très stables C_0, C_2, et C_4 ont une entropie K nulle et une moyenne négative de coefficients de Lyapunov partiels. En revanche, même si la moyenne de C_1 est négative, son entropie reste positive et sa stabilité est comptée. Ce cluster risque de se scinder. Quant à C_3 il est également instable, mais a paradoxalement une durée de vie plus longue que C_1.

Suivi d'un cluster de synchronisation. Au cours d'une simulation impliquant la valeur $\epsilon = 0,8$, nous avons suivi la valeur du cluster C_0, dont l'agent de départ est toujours l'agent A_0 dans l'algorithme de formation des clusters. La figure 10.2.2 présente deux visualisations pour le suivi de la dynamique de notre système : d'une part le spectre complet (ici 50 coefficients) porté sur la même abscisse temporelle tous les $p\ \tau = 50$ pas de temps, d'autre part l'évolution de l'entropie K partielle calculée au même échantillonnage pour le cluster C_0 uniquement.

Dans le graphique 10.9(a) on suit l'évolution du nuage de coefficients de Lyapunov, celui-ci



(a) Évolution du spectre de Lyapunov (50 coefficients) tous les 50 pas de temps.

(b) Évolution de l'entropie du cluster C_0 en bleu ainsi que les coefficients de Lyapunov partiels de ses agents (points).

FIG. 10.9 – Spectre de Lyapunov partiel et suivi de l'entropie K du cluster C_0 en fonction du temps avec $\epsilon = 0,8$. L'intervalle de temps est de 50 pas de temps à chaque évaluation.

oscillant au gré des formations et dissolution de clusters. Cependant ce nuage tend à se déplacer vers les valeurs négatives à mesure de l'augmentation du temps de simulation. On peut également observer des groupes de valeurs persistantes pendant plusieurs séquences de calcul : ils révèlent la formation de clusters de synchronisation.

Le graphique 10.9(b) montre de façon restreinte au cluster C_0 les valeurs du spectre propre au cluster et l'entropie K estimée sur le cluster (en courbe bleue). Cette valeur tombe à zéro dès que le cluster entre dans une phase de synchronisation forte, mais il peut être rapidement détruit par une collision avec un autre cluster.

Discussion. Cette dernière constatation donne les limites de la grandeur d'entropie K partielle que nous avons calculée. Sa valeur de prédiction ne tient que si le cluster est seul à évoluer, mais dès que d'autres clusters existent simultanément, des événements de collisions difficilement prévisible surviennent. Pour prévoir dans une certaine mesure ces événements collisionnels entre clusters, il faudrait encore réduire le temps d'échantillonnage pour le calcul de l'entropie K, au risque d'une perte complète de validité de la mesure.

10.3 Auto-organisation de type flocking

Cette section présente un modèle très peu différent du précédent dans ses mécanismes. Tout est même identique à la différence près que les valeurs des variables de contrôle a sont maintenant propres à chaque agent, tout en restant des variables stationnaires au cours du temps. Nous n'avons donc plus affaire à une population d'agents uniquement chaotiques mais à un melting-pot d'agents de tout profil dynamique, certains plutôt chaotiques, d'autres plutôt réguliers, i.e. de variable x_i^t à tendance stationnaire (point fixe). Cette version d'instanciation du SMAL produit des simulations de type "flocking" semblables aux simulations de Reynolds [Reynolds, 1987] pour certains jeux de paramètres. Nous discutons de cette interprétation en fin de section.

Le groupe d'agents est caractérisé par sa distribution sur les variables de contrôle a. Or on voit directement sur le diagramme de bifurcation 10.1 de l'application logistique f quel est l'effet d'une distribution sur la répartition des profils. Ainsi, pour une distribution uniforme de a sur $[0, 1]$, la zone des profils dynamiques chaotiques ne correspond qu'à 10% de l'ensemble des profils.

10.3.1 Cas d'étude et instanciation.

Nous proposons ici d'étudier un cas simple généré par une distribution uniforme des variables de contrôle des agents logistiques. La variable de contrôle a est donc une variable aléatoire de distribution uniforme sur $[0, 1]$. On notera a_i sa réalisation pour un agent A_i. L'agent, par ailleurs identique en toute point à la section précédente, est résumé par les changements suivants :

– l'état de l'agent s'écrit maintenant comme le tuple :

$$s_i(t) = \left\langle x_i(t), a_i^0, \epsilon^0 \right\rangle \tag{10.18}$$

– les équations de changement d'état interne de l'agent sont identiques pour les équations 10.4 et 10.5 à l'exception de :

$$a_i^{t+1} = a_i^0 \equiv \text{ tiré aléatoirement de façon uniforme sur } [0, 1] \text{ à } t = 0 \tag{10.19}$$

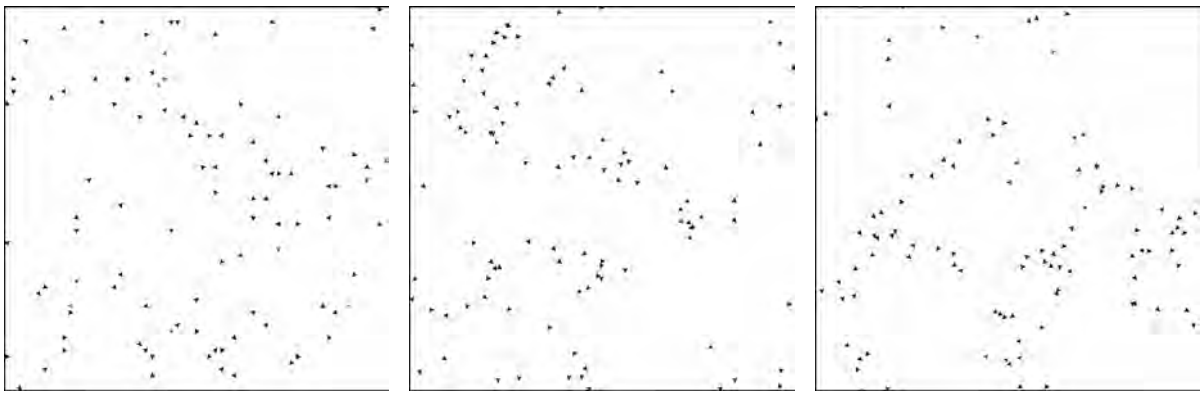
Les perceptions et actions de l'agent suivent les mêmes règles qu'en 10.1, ainsi que la réaction de l'environnement. L'algorithme de simulation reste inchangé.

10.3.2 Résultats de simulations et analyse.

Les simulations effectuées engagent également la même configuration qu'en 10.1.

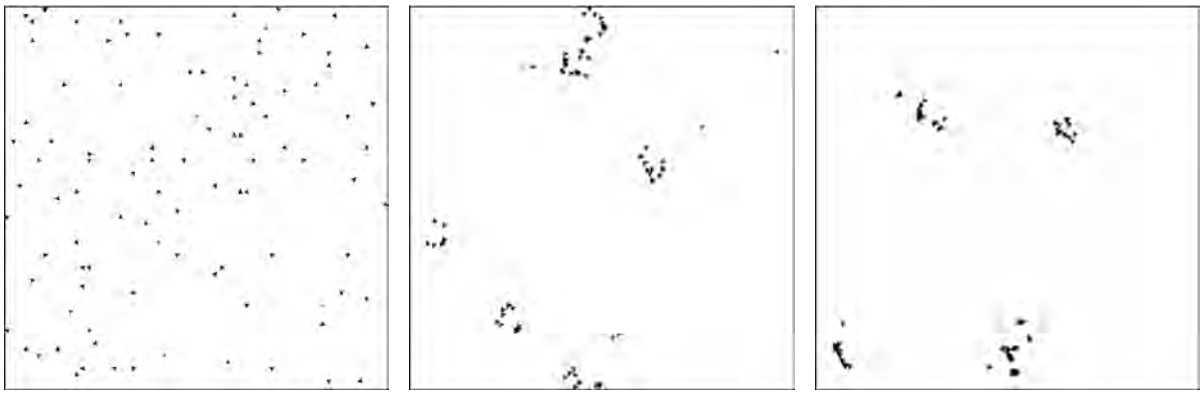
Analyse qualitative. Les figures 10.10 et 10.11 montrent deux exemples de simulation qui permettent déjà de tirer quelques conclusions. La différence au bout de 2000 pas de temps est nette : une synchronisation s'est opérée sous forme de clusters partiels lorsque $\epsilon^0 = 1$, valeur extrême du couplage pour laquelle l'agent prend simplement le champ moyen sur son voisinage comme valeur de variable de décision. Cette synchronisation commence d'ailleurs très rapidement pour cette valeur de couplage total, comme le montre le graphique 10.12(a) sur la phase transitoire. En revanche la valeur $\epsilon^0 = 0.8$ pour laquelle une synchronisation dans le cas chaotique pouvait avoir lieu, ne donne ici aucune synchronisation suffisamment stable dans le temps. La conséquence est que les agents ne se regroupent pas non plus spatialement, malgré un alignement manifestement non isotrope.

On constate également, lorsque le couplage est suffisamment fort $\epsilon^0 > 0.9$, un déplacement globalement plus fluide des agents, contrairement au cas des agents totalement chaotiques. En fait le graphique 10.12(b) trace la phase de synchronisation pour $\epsilon^0 = 1$ et exhibe plusieurs segments de synchronisation correspondant à des clusters séparés compris entre $0,54$ et $0,58$. Les segments les plus stables sont les plus bas en valeur, les fils les plus haut vivent quelques événements de désynchronisation chronique. Ces segments comprennent une proportion plus importante d'agents chaotiques. Le graphique 10.12(c) pour $\epsilon^0 = 0,95$ révèle une synchronisation



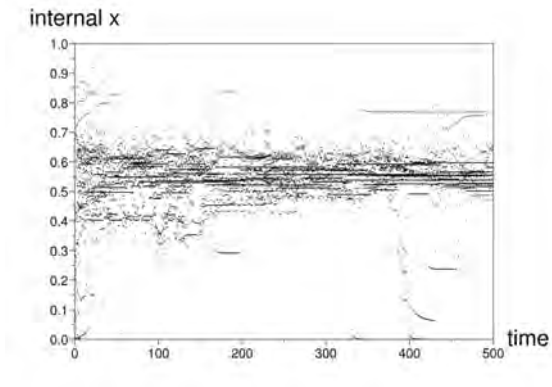
(a) $t = 1$ (b) $t = 500$ (c) $t = 2300$

FIG. 10.10 – Copies d'écran d'une simulation avec $N = 100$ agents de rayon de voisinage $R = 4$, et de couplage $\epsilon^0 = 0.8$

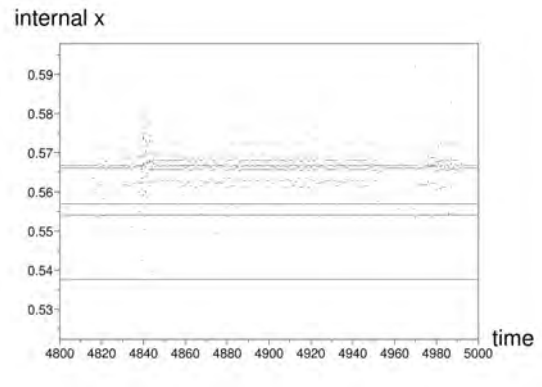


(a) $t = 1$ (b) $t = 500$ (c) $t = 3000$

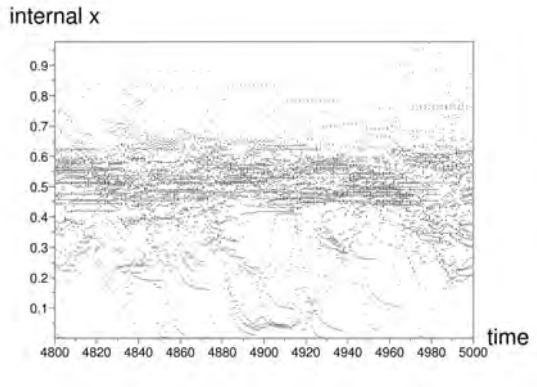
FIG. 10.11 – Copies d'écran d'une simulation avec $N = 100$ agents de rayon de voisinage $R = 4$, et de couplage $\epsilon^0 = 1$



(a) Phase transitoire pour $\epsilon^0 = 1$



(b) Phase synchronisée pour $\epsilon^0 = 1$

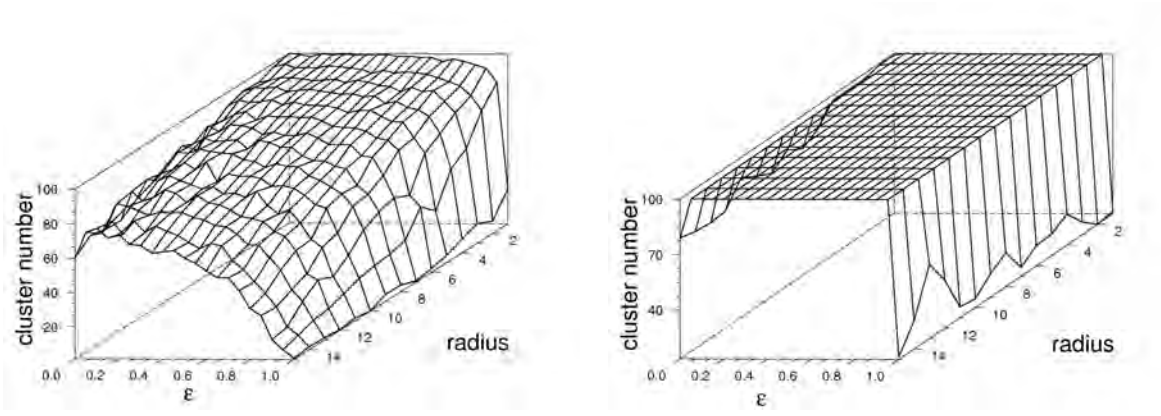


(c) Phase synchronisée pour $\epsilon^0 = 0.95$

FIG. 10.12 – Evolutions temporelles de l'ensemble des variables décisionnelles internes x correspondant à tous les agents en fonction du temps. Le rayon de perception est $R = 4$.

plus floue comprise entre $0,4$ et $0,65$, malgré la forte valeur de couplage. Il présente beaucoup de segments de synchronisation donc plus de clusters et des événements perturbatifs plus fréquents. On y voit notamment les agents à faible variable de contrôle $a < 0,25$ tendre vers leur point fixe 0 dès qu'ils sortent d'un cluster.

Analyse quantitative. Le graphique 10.13 résulte de multiples simulations visant à évaluer le degré de synchronisation en fonction des paramètres du système, comme en 10.1. La distinction entre deux précisions d'observation pour la sélection des clusters rend l'analyse plus évidente. En effet, à haute précision donc faible valeur $\gamma = 10^{-10}$, peu de valeurs de couplage permettent une synchronisation, excepté pour les deux valeurs extrêmes $\epsilon^0 = 0$ et $\epsilon^0 = 1$. Nous venons d'expliquer la raison du comportement dynamique pour la valeur forte du couplage $\epsilon^0 = 1$ dans le paragraphe précédent, la synchronisation pour $\epsilon^0 = 0$ est plus surprenante. La raison en est cependant assez simple : la distribution uniforme de a sur $[0;1]$ donne sans couplage les proportions suivantes de profils dynamiques à partir du diagramme de bifurcation 10.1 : 25% des agents avec $a \in [0;0,25]$ convergent vers le point fixe 0, 50% avec $a \in [0,25;0,75]$ converge vers un point fixe strictement positif mais deux agents n'ont pas le même point fixe de convergence, enfin 25% avec $a \in [0,75;1]$ont des profils de type périodique ou chaotique. Lorsque le rayon de perception des agents avec $a \in [0;0,25]$ augmente, il englobe à partir d'une certaine taille d'autres agents du même type dont la dynamique interne converge vers 0, ces agents sont donc naturellement synchronisés. Lorsque le rayon atteint sa valeur maximale où la perception est totale, le taux de synchronisation est de 25% ce qui correspond à la proportion des agents qui convergent vers 0. Ce fait est corroboré par la valeur du nombre de clusters lorsque $R = 15$ sur la figure10.13(b). Dès qu'un couplage existe quel que soit sa valeur, cette synchronisation intrinsèque disparaît du fait des interactions multiples. A plus faible précision d'analyse de la



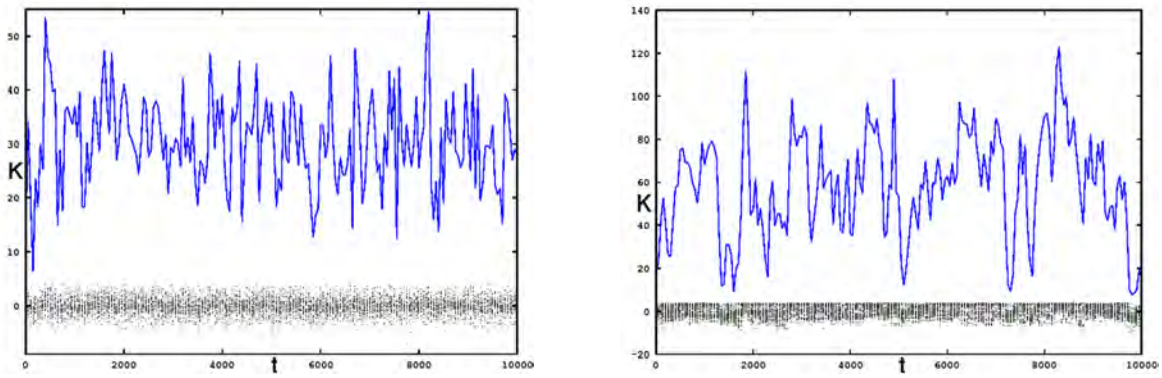
(a) $\bar{N}_c$ à la précision 10^{-2}

(b) $\bar{N}_c$ à la précision 10^{-10}

FIG. 10.13 – Cas d'agents de tout profil dynamique : nombre moyen de clusters formés en z, en fonction du couplage ϵ^0 en y et du rayon de voisinage de perception R en x pour $N = 100$ agents.

formation des clusters, outre le fait précédent toujours visible, on peut conclure que la formation des clusters est influencée positivement par l'augmentation du couplage et l'augmentation du

rayon de perception.



(a) Cas où $\epsilon = 0,8$.

(b) Cas où $\epsilon = 0,99$.

FIG. 10.14 – Spectre de Lyapunov partiel et suivi de l'entropie K du système en fonction du temps avec deux valeurs ϵ. L'intervalle de temps est de 50 pas de temps à chaque évaluation.

Analyse entropique. La figure 10.3.2 montre comme précédemment le calcul des valeurs de l'entropie K partielle et le spectre de Lyapunov partiel [50] sur tout le groupe d'agents. La conclusion paradoxale est de constater qu'à couplage fort (cf. figure 10.14(b)), l'entropie subit des variations d'amplitude plus grande qu'à couplage modéré (cf. figure 10.14(a)). En revanche la fréquence de ces variations est moins grande. Des clusters plus stables semblent se former mais ils se dissocient également d'autant plus fortement. Le couplage fort contraint le système composé d'éléments très divers du point de vue dynamique à synchroniser et donc à former un ensemble géométrique cohésif. Mais dès qu'une collision survient, les clusters de plus basse entropie volent en éclats.

10.3.3 Discussion.

Les simulations effectuées dans ce cas précis d'une distribution uniforme de la variable de contrôle sur l'ensemble des agents que nous avons exposées, génèrent des schémas/"patterns" qui ressemblent à des simulations produites par l'algorithme de Reynolds dédié au flocking [Reynolds, 1987]. Cependant, là où les règles de l'algorithme de Reynolds sont finalement assez figées, notamment la convergence vers le centre de masse du groupe, exprimé par l'alignement en moyenne des vitesse dans le modèle de Vicsek, l'intérêt du modèle SMAL est de pouvoir moduler et contrôler dans une certaine mesure la forme et la structure des groupes obtenus via la distribution de la variable de contrôle et de la variable de couplage sur le groupe d'agents, ce qui permet d'imaginer de nombreux scénarios. Nous n'avons fait ici qu'effleurer les possibilités qu'offre le SMAL dans la façon de moduler ses paramètres, peu nombreux au final relativement à la génération des comportements qu'ils permettent.

Deux difficultés apparaissent malgré tout dans nos simulations pour qu'elles puissent devenir de véritables simulations de flocking :

[50] Simulations et calculs réalisés sur le simulateur multi-agent MASON [Luke *et al.*, 2003]

- La référence des angles dans notre SMAL instancié ici est fixe par rapport à l'espace de l'environnement. L'effet de groupe assimilé au flocking apparaît grâce à cette référence commune. Les états internes des agents convergeant vers une valeur entre $0,5$ et $0,6$, l'angle de déplacement du groupe est directement proportionnel et donc de l'ordre de π de façon déterminée pour chaque simulation. C'est l'effet de la loi d'action choisie et de la dimension 1 de l'espace de décision pour un déplacement en dimension 2.
- Dans le même ordre d'idée, l'espace E de environnement est ici un tore de dimension 2, donc un espace sans bord, qui facilite les simulations et la conception algorithmique. Mais d'un point de vue dynamique, cette réduction n'est pas neutre, elle opère un repliement de l'espace grâce à l'opérateur modulo qui évite d'avoir à gérer les conditions aux limites de l'espace. En particulier, appliquer notre algorithme sur des robots dans des espaces physiques n'est donc pas possible en l'état.

Ces deux limitations sont levées dans le modèle de flocking plus élaboré de la section 11.1.

10.4 Bilan

Cette première partie applicative du SMAL s'est concentrée à développer et exploiter les effets liés à certains mécanismes d'auto-organisation du système. Ces mécanismes sont essentiellement dus aux couplages sur les états internes des agents, traduits ensuite géométriquement dans l'espace. L'expression d'un effet interne aux agents en une manifestation externe dans l'environnement est l'essence même de notre approche à travers le modèle SMAL. Plus précisément, ce sont les mécanismes de synchronisation inhérents au modèle de calcul par itérations chaotiques couplées qui sont à la base des effets géométriques observés. Le concept de synchronisation prend donc une place centrale dans la conception des mécanismes internes au SMAL et apparaît comme la source principale de l'auto-organisation dans nos cas d'études avec le SMAL.

Un autre aspect s'est révélé également à travers cette instance du SMAL pour le flocking : dans les deux cas présentés, l'environnement joue le même rôle de mémoire temporaire nécessaire au couplage, puisque toutes les interactions sont indirectes dans notre modélisation. Le champ de décision utilisé stocke des données endogènes au système, c'est-à-dire générées par les agents eux-mêmes. Pour cette raison nous qualifions le phénomène d'auto-organisé, car les données échangées sont endogènes. La partie suivante focalise sur des mécanismes adaptatifs en plus du mécanisme d'auto-organisation toujours présent. Ces mécanismes ont la particularité de prendre en compte des champs de données exogènes au système, c'est-à-dire de données générées de façon externe et antérieures aux agents.

11

Mécanismes adaptatifs

Le principe des mécanismes d'adaptation que nous mettons en place dans les instances du SMAL présentées dans ce deuxième chapitre applicatif consiste à modifier dynamiquement la valeur de la variable de contrôle ou de couplage de l'agent logistique avec des données de nature exogène perçues dans un champ de l'environnement. Ce caractère exogène des données de l'environnement, i.e. non générées par les agents mais considérées comme provenant des entrées extérieures au système, marque la différence essentielle entre mécanismes d'auto-organisation et d'adaptation du SMAL.

Cependant, les deux mécanismes coexistent en général et les agents modifient en retour l'état de l'environnement par influence combinée de leurs actions, elles-mêmes basées sur les variables de décision des agents. Ces mécanismes d'adaptation donnent lieu à des phénomènes stigmergiques comme ceux impliquant les phéromones numériques, d'autres permettent de prendre en compte les obstacles dans un environnement comme nous le voyons dans une nouvelle version de flocking avec obstacles dans la section qui suit.

11.1 Flocking dans un espace avec obstacles

Position du problème

Jusqu'ici l'environnement de nos simulations est un tore à deux dimensions et les conditions aux limites n'existent pas. Cet espace particulier sans bords facilite donc dans un premier temps l'expérimentation des mécanismes de déplacement collectif et l'étude mathématique. Il devient vite évident que l'on ne peut pas en rester à des espaces aussi abstraits dès que l'on envisage des applications dans le monde réel, telle que la robotique en essaim. Il est donc nécessaire de passer à des mondes bornés, comportant éventuellement des obstacles, qui correspondent davantage aux conditions réelles pour les phénomènes de déplacements de groupe.

Carences du modèle de flocking précédent. Notre premier modèle de flocking a permis de relier un phénomène de synchronisation interne à des formes géométriques externes. Cependant plusieurs limitations sont apparues (cf. section 10.3) qui doivent être levées selon deux objectifs :

- faire en sorte que l'angle de déplacement global du groupe n'ait plus de référence fixe,
- mettre en oeuvre un mécanisme adaptatif qui permette aux agents d'évoluer dans un environnement borné, c'est-à-dire prenant en compte des zones de déplacement interdite (limites de l'environnement et obstacles).

Principe général de l'algorithme.

Approche de Reynolds. L'algorithme de Reynolds pour le flocking combine trois composantes d'accélération pour calculer la vitesse d'un agent i, où N_i est le nombre de voisins présents dans le voisinage V_i lorsque $N_i \neq 0$:

$$\begin{aligned}
\mathbf{v_i}^{t+1} - \mathbf{v_i}^{t} &= \alpha\ \mathbf{a_i}_c^t + \beta\ \mathbf{a_i}_a^t + \gamma\ \mathbf{a_i}_s^t \\
\mathbf{a_i}_c^t &= \frac{1}{N_i} \sum_{j \in V_i} (\mathbf{r_j} - \mathbf{r_i}) \\
\mathbf{a_i}_a &= \frac{1}{N_i} \sum_{j \in V_i} \mathbf{v_j} \\
\mathbf{a_i}_s &= -\frac{1}{N_i} \sum_{j \in V_i} \frac{\mathbf{r_j} - \mathbf{r_i}}{\|\mathbf{r_j} - \mathbf{r_i}\|^2}
\end{aligned}$$

La première composante d'accélération correspond à la cohésion vers le centre de masse du voisinage, la deuxième à l'alignement sur la moyenne des vitesses, la troisième à la séparation, c'est-à-dire la répulsion des voisins trop proches. En général, la combinaison donne le même poids à chaque composante, mais en fonction des événements perçus par l'agent, cette proportion change et donne la priorité à l'une ou l'autre des composantes. Ce mécanisme d'ajustement est très peu formalisé dans les articles de Reynolds et semble résulter d'une analyse empirique.

Lorsque des obstacles sont présents, une accélération supplémentaire s'ajoute aux trois précédentes qui tient compte de l'obstacle : comme on l'a vu en section 1.3.1, elle peut correspondre à une force répulsive dérivée d'un champ de potentiel ou d'un calcul purement géométrique qui nécessite des informations sur la géométrie des obstacles.

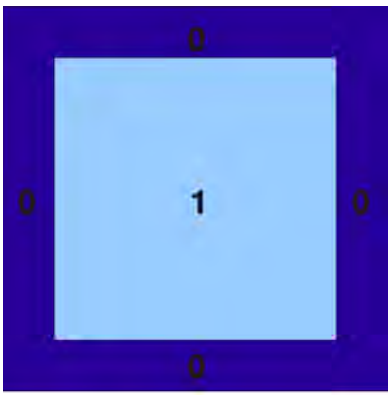
Notre approche. Contrairement à l'approche de Reynolds, nous n'avons avec le modèle SMAL pas de notion de cohésion, ni de séparation. Cela est remplacé par un niveau plus ou moins important de synchronisation entre variables de décision de deux agents voisins. Le principe est simple : il n'y a pas de collision si les mouvements d'agents proches sont exactement semblables. L'alignement s'opère selon la direction moyenne du groupe voisin. Notre approche de l'évitement d'obstacle mêle une approche par le champ de l'environnement et une approche géométrique. L'ensemble du déplacement résulte d'un couplage piloté par le champ, ce qui caractérise les mécanismes impliqués. Pour résumer et sans préciser dès à présent le détail de l'obtention de cette formule, la forme prise pour la mise à jour de la direction de la vitesse d'un agent A_i est donnée par le calcul de l'angle d'un vecteur composé de 3 vecteurs directeurs pondérés selon (cf. figure 11.1) :

$$\mathbf{U}_i = (1 - \epsilon_{i,1})\mathbf{T}_{O \to i} + \epsilon_{i,1}(\mathbf{T}_{i,1} + \mathbf{T}_{i,2}) \tag{11.1}$$

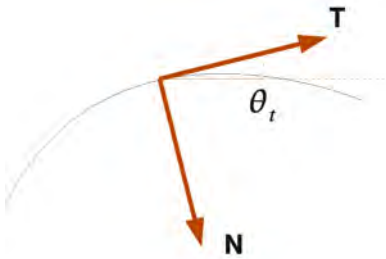
où $\mathbf{T}_{O\to i}$ est le vecteur directeur pour l'évitement d'obstacle dont l'importance relative dépend de la valeur prise par $\epsilon_i(\sigma^t)$ qui dépend lui-même de l'état de l'environnement au temps t. En particulier deux cas extrêmes se présentent :

- en dehors de la perception de tout obstacle, $\epsilon_i(\sigma^t) = 1$, et le premier terme de (11.1) disparaît,
- sur l'obstacle, $\epsilon_i(\sigma^t) = 0$ et seul le premier terme de (11.1) intervient.

Les composantes directrices $\mathbf{T}_{i,1}$ et $\mathbf{T}_{i,2}$ correspondent respectivement à l'alignement au groupe et à la synchronisation interne. Nous éliminons de fait les notions d'attraction et de répulsion pour les remplacer par des phénomènes de synchronisation entre états internes des agents et un ajustement des vitesses. L'obtention de ces diverses composantes est exposée dans l'instanciation détaillée qui suit.



(a) Champ de couplage.



(b) Repère de Frenet.

FIG. 11.1 – Champ de l'environnement et repérage géométrique.

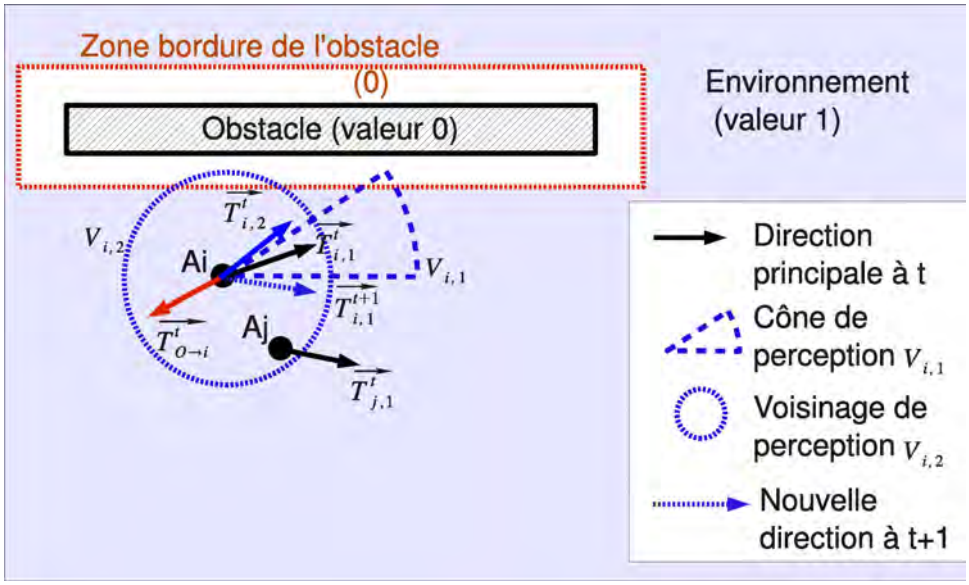


FIG. 11.2 – Schéma de principe de l'évitement d'obstacles pour l'agent logistique et du calcul de la direction globale résultante.

Structure de l'environnement.

Environnement. L'environnement est un espace géométrique $2D$ continu E fermé borné de bordure carrée $L \times L$, où $L \gg 1$. Il comprend, outre les champs de position habituels, un champ de couplage $\mathcal{E}$ et un champ de direction des vitesses noté $\mathcal{T}$. Avant de décrire plus précisément ces deux champs, rappelons les grandeurs et fonctions qui nous sont utiles ici.

Notations et fonctions utiles. Les grandeurs manipulées sont des vecteurs de dimension 2 dans l'espace géométrique. Le repère naturel ici est le repère de Frenet comportant deux vecteurs unitaires orthogonaux en $2D$: $\mathbf{T}$ est le vecteur unitaire tangent à la trajectoire et $\mathbf{N}$ le vecteur unitaire normal dirigé vers la concavité de la trajectoire. L'angle que fait $\mathbf{T}$ avec la référence des angles est noté comme précédemment θ (cf. figure 11.1(b)). Dans ce repère la vitesse instantanée d'un agent A_i s'écrit :

$$\mathbf{v}_i = v_i \mathbf{T}_i \tag{11.2}$$

Nous considérons ici l'évolution de la norme de la vitesse v_i indépendamment de sa direction indiquée par $\mathbf{T}_i$. Nous utilisons systématiquement la fonction "arg" dont l'argument est un vecteur, pour renvoyer l'angle de ce vecteur. Ainsi :

$$\begin{aligned} \theta^t &= \arg(\mathbf{v}^t) \\ &= \arctan(\frac{v_y^t}{v_x^t}) \end{aligned} \tag{11.3}$$

où v_y et v_x désignent respectivement l'ordonnée et l'abscisse cartésienne du vecteur.

Définition des champs de l'environnement.

Champ de couplage aux obstacles. Le champ de couplage aux obstacles $\mathcal{E}$ est défini par les données suivantes en tenant compte du bord qui constitue un obstacle infranchissable (nous sommes ici en coordonnées cartésiennes (X, Y)) :

$$\mathcal{E}(X, Y) = \begin{cases} 1 \text{ si } l \leq X, Y \leq L - l \\ \xi \text{ sinon} \end{cases} \tag{11.4}$$

où $\xi \geqslant 0$ est la valeur minimum du champ sur une bande de largeur l longeant les bords et la valeur 1 partout ailleurs (cf. figure 11.1(a)). Ce champ est persistant. Il en serait de même avec tout obstacle disposé dans l'environnement, en imposant une valeur ξ près des bords. La valeur minimale du champ est une constante qui permet de moduler l'influence du reste du groupe d'agents sur les agents près des bords.

Champ de directions des vitesses. Le champ $\mathcal{T}$ est un champ continu non persistant qui permet de connaître l'orientation courante de la vitesse d'un agent. En fait, ce champ donne accès au vecteur tangent unitaire de déplacement $\mathbf{T}_i^t$ de l'agent A_i à chaque pas de temps t, car nous traitons séparément la direction et la norme de la vitesse. Ce champ est mis à jour à

chaque pas de temps par l'agent lui-même après modification par les perceptions. En réalité, ce n'est pas un vecteur qui doit être stocké mais simplement l'angle $\arg(\mathbf{T}_i^t)$ puisque ce vecteur est toujours de norme unitaire. Nous allons voir que cet angle est au final relié de façon simple à la variable de décison $x_{i,1}$ de l'agent. Ce champ de directions des vitesses se ramène donc au champ de décision $\mathcal{X}_1$.

États internes des agents.

C'est le seul exemple traité dans cette thèse comportant deux dimensions pour la variable décisionnelle interne de l'agent logistique. On a ainsi l'état d'un agent i décrit par le tuple :

$$s(t) = \left\langle x_{i,1}^t, x_{i,2}^t, a_i^0, \epsilon_{i,1}^t, \epsilon_2^0 \right\rangle \tag{11.5}$$

avec les définitions associées :

- $x_{i,1}^t$ correspond à la décision de suivre la direction commune de déplacement ; cette variable est traduite en une valeur d'angle dans l'environnement par la relation $\theta_{i,1}^t = 2\pi\ x_{i,1}^t$ qui oriente le vecteur directeur $\mathbf{T}_{i,1}$.
- Ce première variable de décision est associée à la variable de couplage $\epsilon_{i,1}^t$ qui varie en fonction de l'état de l'environnement. C'est par ce couplage que s'opère l'adaptation aux obstacles en pondérant l'importance de la perception relative aux obstacles.
- $x_{i,2}^t$ correspond à la décision de l'agent de dévier par rapport à cette direction commune de façon autonome ; elle se traduit également dans l'environnement en une valeur d'angle selon la même formule que précédemment et oriente le vecteur directeur $\mathbf{T}_{i,2}$.
- Le couplage ϵ_2^0 qui reste constant et identique pour tous les agents. C'est pourquoi il ne comporte pas d'indice i. Ce couplage régit la synchronisation interne entre l'agent et ses proches voisins.
- La variable $x_{i,2}^t$ suit l'évolution définie pour le flocking décrit en section 10.3.
- a_i^0 est la variable interne de contrôle des agents qui régit l'évolution de $x_{i,2}^t$.

Principe général de l'algorithme de déplacement des agents comparativement au modèle de Vicsek.

Nous suivons ici le principe de déplacement du modèle de Vicsek (cf. section 1.3.2) auquel on ajoute une contrainte liée aux obstacles. Ainsi nous considérons l'évolution de l'angle de direction de déplacement θ_i^t comme composée de deux éléments principaux, un angle de base de la direction liée à la décision interne $x_{i,1}^t$ et une composante perturbatrice liée à la variable interne $x_{i,2}^t$ de l'agent. Cette perturbation est ici le fruit de l'application logistique couplée et non le bruit blanc du modèle de Vicsek. Ce n'est donc pas réellement un bruit, mais bien le résultat de la décision autonome de l'agent. On peut donc en moduler la variation en profitant des effets de la synchronisation associée des états internes. On résume simplement le calcul final par la formule similaire à celle de Vicsek mais qui prend l'expression :

$$\theta_i^t = \theta_{i,1}^t + \triangle\theta_{i,2}^t \quad mod(2\pi) \tag{11.6}$$

$$= \arg\left((1-\epsilon_{i,1}^t)\mathbf{T}_{O\to i}^t + \epsilon_{i,1}^t \mathbf{T}_{i,1}^t\right) + \epsilon_{i,1}^t (x_{i,2}^t - \frac{1}{2})\ \pi \quad mod(2\pi) \tag{11.7}$$

Voyons à présent comment on obtient cette formulation via les perceptions dans notre modèle.

Perceptions de l'agent.

Voisinages. Deux voisinages sont utilisés (cf. figure 11.1) :

- un voisinage $V_{i,2}$ de type boule de rayon R_2 centrée sur l'agent et associé aux perceptions sur $x_{i,2}$ notamment, c'est-à-dire à la synchronisation interne entre individus du groupe,
- un voisinage $V_{i,1}$ de type cône de perception de rayon R_1 et d'angle d'ouverture ψ centré sur l'agent pour la perception des obstacles et la régulation de la vitesse. Il s'avère qu'une perception angulaire restreinte de type cônique semble nécessaire pour générer un comportement dynamique global rapidement adapté à la configuration des obstacles et cohérent par rapport aux attentes phénoménologiques. Ce point est traité dans l'analyse des simulations.

Calcul du vecteur directeur de l'obstacle. En environnement clos, la perception des obstacles est essentielle. Elle est traitée dans notre approche par la détermination d'un vecteur directeur $\mathbf{T}_{O\to i}$ qui donne l'orientation pour éviter l'obstacle. Le principe de la détermination de ce vecteur est simple : dès qu'une valeur minimale est perçue par l'agent dans le champ $\mathcal{E}$, valeur localisée en un point $\mathbf{O}$ dans l'environnement situé en général sur l'obstacle, l'agent récupère la direction du vecteur directeur allant de l'agent vers l'obstacle perçu $\mathbf{T}_{i\to O}$. Le point $\mathbf{O}$ est défini comme le barycentre de l'ensemble des points $\mathbf{X}$ réalisant le minimum du champ $\mathcal{E}$ sur le voisinage cônique $V_{i,1}$ exprimé par la formulation suivante :

$$\mathbf{O} = \langle \arg\min_{\mathbf{X}} \ \mathcal{E}^t(\mathbf{X}) \rangle_{V_{i,1}} \tag{11.8}$$

On obtient simplement le vecteur d'évitement de l'obstacle par le vecteur directeur :

$$\mathbf{T}_{O\to i} = \mathbf{T}_{i\to O} + \pi \quad mod(2\pi) \tag{11.9}$$

Ce vecteur $\mathbf{T}_{O\to i}$ tend à repousser l'agent de l'obstacle.

Perceptions. Rappelons tout d'abord que le rôle des opérateurs de perception est de capter et transformer les informations des champs de l'environnement pour les exploiter dans l'espace d'état interne de l'agent logistique. Ainsi la direction de la vitesse de l'agent donnée par $\mathbf{T}_i$ est calculée à l'issue de plusieurs perceptions. Ces perceptions servent également à mettre à jour les variables d'états internes.

Les perception selon les composantes $\epsilon_{i,1}$ et $x_{i,1}$ sont respectivement :

$$P_{i,\epsilon}^{t+1} = \mathcal{M}in_{V_{i,1}}\mathcal{E}^t \tag{11.10}$$

$$P_{i,1}^{t+1} = \arg\left((1 - P_{i,\epsilon}^{t+1})\mathbf{T}_{O\to i} + P_{i,\epsilon}^{t+1}\mathbf{T}_i^{t+1}\right) \tag{11.11}$$

$$\text{où } \mathbf{T}_i^{t+1} = (1-\epsilon_2^0)\mathbf{T}_{i,1}^t + \epsilon_2^0 \frac{\sum_{j\in V_{i,1}} \mathbf{T}_{j,1}^t}{\Sigma_{V_{i,1}}\mathcal{N}^t} \tag{11.12}$$

L'équation 11.11 est illustrée sur la figure 11.1. En développant la fonction arg, on obtient le calcul suivant pour l'équation 11.11 :

$$P_{i,1}^{t+1} = \arctan\left(\frac{(1-P_{i,\epsilon}^{t+1})\sin(\theta_{O\to i}^t) + P_{i,\epsilon}^{t+1}\left((1-\epsilon_2^0)\sin(\theta_{i,1}^t) + \epsilon_2^0\ \langle\sin(\mathcal{T})\rangle_{V_{i,1}}^t\right)}{(1-P_{i,\epsilon}^{t+1})\cos(\theta_{O\to i}^t) + P_{i,\epsilon}^{t+1}\left((1-\epsilon_2^0)\cos(\theta_{i,1}^t) + \epsilon_2^0\ \langle\cos(\mathcal{T})\rangle_{V_{i,1}}^t\right)}\right) \tag{11.13}$$

où $\theta^t_{O\to i}$ et $\theta^t_{i,1}$ sont les angles (arguments) respectifs des vecteurs directeurs $\mathbf{T}_{i,O}$ et $\mathbf{T}^t_{i,1}$. Quant à l'expression plus globale $\langle \sin(\mathcal{T})\rangle^t_{V_{i,1}}$ elle correspond au calcul de la moyenne des fonctions sinus sur tous les angles contenus dans le voisinage $V_{i,1}$ du champ $\mathcal{T}$.

Cette formule apparemment compliquée se simplifie grandement avec les deux valeurs principales prises par la perception $P^{t+1}_{i,1}$ à savoir ξ ou 1. Ainsi dès qu'une valeur faible est perçue dans l'environnement, la direction de l'agent est immédiatement pilotée par l'obstacle, tandis que lorsque cette perception cesse, un équilibre est établi entre la direction propre et celle du groupe d'agents.

Évolution de l'état interne de l'agent.

Les perceptions étant maintenant posées, on peut écrire les équations d'évolution des variables internes de l'agent. Nous n'explicitons pas ce qui concerne la mise à jour de la variable de décision $x^t_{i,2}$, car elle est identique à ce qui a été vu dans la première version de flocking en section 10.3. L'application logistique utilisée ici est la forme quadratique classique $f(x) = 4\,x(1-x)$. Cette partie déjà connue étant établie, les évolutions des autres variables internes sont régies par les équations suivantes :

$$\begin{cases} \epsilon^{t+1}_{i,1} & = P^{t+1}_{i,\epsilon} \\ x^{t+1}_{i,1} & = \frac{1}{2\pi}(P^{t+1}_{i,1}) \end{cases} \tag{11.14}$$

Actions de l'agent.

Les actions de l'agent sont essentiellement des actions de déplacement qui se décompose en deux éléments : la mise à jour de la norme de la vitesse et la mise à jour de la direction de déplacement.

Mise à jour de la vitesse. La norme de la vitesse de l'agent i suit une loi d'évolution qui utilise la perception du nombre d'agents dans le cône frontal de voisinage $\Sigma_{V_{i,1}}\mathcal{N}^t$, y compris l'agent lui-même ; cette grandeur a donc toujours la valeur minimale 1. Cette modulation de la vitesse par le nombre d'agents perçus devant soi s'est révélé très efficace pour éviter les collisions des agents et la régulation du mouvement du groupe. Cette loi s'écrit en fonction du paramètre v_0 identique pour tous les agents et qui limite la norme de la vitesse :

$$v^{t+1}_i = \frac{(1 + x^{t+1}_{i,2})}{\Sigma_{V_{i,1}}\mathcal{N}^t}\; v_0 \tag{11.15}$$

Il est immédiat que la vitesse de l'agent ne peut dépasser $2v_0$ et varie dans l'intervalle $]0, 2v_0]$. Cette vitesse n'est jamais nulle du fait du coefficient $(1 + x^{t+1}_{i,2})$ toujours supérieur à 1.

Mise à jour de la position. La mise à jour de la position courante résume l'action de déplacement de l'agent A_i. La formule reprend le rayon vecteur de l'agent en coordonnées polaires

additionné à un petit déplacement selon la nouvelle direction $\mathbf{T}_{i,1}^{t+1}$ orientée selon :

$$\begin{cases} \arg(\mathbf{T}_{i,1}^{t+1}) = 2\pi\ x_{i,1}^{t+1} + \epsilon_{i,1}^{t+1}(x_{i,2}^{t+1} - \frac{1}{2})\ \pi \quad mod(2\pi)) \\ \mathbf{r}_i^{t+1} = \mathbf{r}_i^t + v_i^{t+1}\ \mathbf{T}_{i,1}^{t+1} \end{cases} \tag{11.16}$$

Mise à jour des autres champs. Le champ de couplage n'est pas modifié puisqu'il est fixé *a priori* par la configuration des obstacles. En revanche, le champ des directions des vitesses doit être mis à jour. Ce champ est non persistant et stocke en chaque nouvelle position de l'agent son nouvel état interne :

$$\mathcal{T}(\mathbf{r}_i^{t+1}) = 2\pi\ x_{i,1}^{t+1} \tag{11.17}$$

Simulations réalisées.

Paramètres généraux. L'environnement des simulations effectuées ici[51] est un carré de côté $L = 100$. Le champ $\mathcal{E}$ est discrétisé pour les simulations présentées, en cellules 1×1 pour définir les bordures de façon simple. Un environnement comprenant trois obstacles de tailles différentes a été testé : le mécanisme d'évitement d'obstacles s'applique de la même façon aux obstacles intérieurs que sur les bords de l'environnement. Ce mécanisme ne dépend pas de la géométrie de ces obstacles.

Les conditions intiales sont à chaque fois une répartition aléatoire des agents dans l'environnement avec des directions et des variables internes tirées aléatoirement dans leur domaine de définition respectif. Dans toutes les simulations qui vont suivre, et sauf mention contraire, la valeur par défaut de l'angle d'ouverture du cône perceptif des agents est $\psi = \frac{\pi}{2}$. Les rayons de perception sont $R_1 = R_2 = 4$. De même la valeur minimale par défaut du champ de couplage aux obstacles $\mathcal{E}$ est $\xi = 0,2$. Le couplage de synchronisation est donné par $\epsilon_2^0 = 0,9$ sauf mention contraire.

Différents types de populations d'agents. L'autre aspect dans ces simulations tient à la population d'agents et à la distribution de profil dynamique relativement à l'application logistique, c'est-à-dire à la façon dont est initialisée la variable de contrôle a_i^0 stationnaire pour chaque agent. Rappelons que cette grandeur interne gouverne le type de comportement dynamique asymptotique de l'agent qui peut être selon la valeur de a un point fixe, un cycle ou un attracteur chaotique. Trois types de distributions ont donc été testées pour le choix de a_i^0 :

- une distribution ponctuelle (ou distribution de Dirac) sur une valeur particulière $a_0 \in [0, 1]$. Tous les agents ont donc la même valeur de contrôle interne. Les deux valeurs qui nous intéressent ici sont $a_0 = 0$ pour laquelle l'agent n'a pas d'autonomie de déplacement car $x_2 = 0$ en permanence, seule la direction commune de déplacement va jouer un rôle, et $a_0 = 1$ pour laquelle tous les agents sont chaotiques et possèdent un comportement de déviation aléatoire.
- une distribution uniforme sur $[0, 1]$

[51] Toutes les simulations présentées ici ont été effectuées avec le simulateur NetLogo [Wilensky, 1999].

– une distribution normale $\mathcal{N}(\mu, \sigma^2)$ où $\mu \in [0, 1]$ est la moyenne de la loi et σ^2 sa variance. Ainsi on se sert d'une distribution en cloche pour privilégier un certain type de dynamique. Une propriété utile concernant cette loi est alors que 68% de la population générée possède une valeur a dans l'intervalle $[\mu - \sigma, \mu + \sigma]$ et que 95% de la population possède une valeur a dans $[\mu - 2\sigma, \mu + 2\sigma]$.

Ces trois dernières situations font l'objet d'un paragraphe spécifique ci-après.

Résultats des simulations.

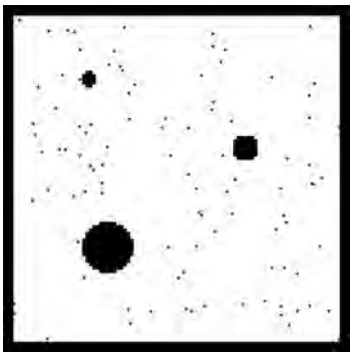
Constats généraux. Un point important déjà signalé lors de la présentation des deux voisinages de perception utilisés jusqu'ici est la nécessité d'avoir un voisinage cônique pour la détection des obstacles et la régulation des vitesses. On constate par simulation (cf. figures 11.3(b) et 11.3(c), ou 11.3(d) et 11.3(e)) qu'un voisinage de type disque pour percevoir les obstacles et gérer la norme de la vitesse ne permet pas de produire le phénomène de flocking. En effet, le mécanisme utilisé ici pour l'évitement d'obstacle conduit à piéger l'agent près des bords s'il subit l'influence de l'obstacle quelle que soit la direction de son vecteur vitesse. Pour que l'agent puisse "s'échapper" de l'obstacle, il faut qu'il arrête de percevoir celui-ci dès qu'il lui tourne le dos. Sinon, l'agent ne fait localement que des aller-retours interminables du fait de l'ajout de l'angle π à chaque perception de l'obstacle. Il est donc nécessaire que l'agent possède un cône perceptif centré sur la direction de sa vitesse. C'est le premier résultat lié à cette modélisation.

Population d'agents à profil dynamique identique. Les deux types de population de même profil dynamique, c'est-à-dire comportant la même valeur pour leur variable de contrôle interne a, sont de deux types :

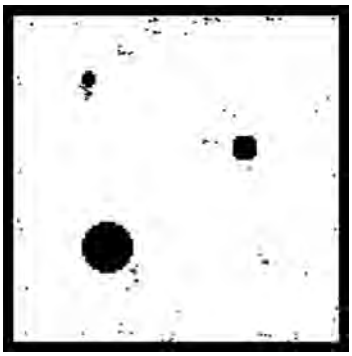
1. Tous les agents possèdent une variable $a_0 = 0$, ils ne varient quasiment pas de la direction moyenne du groupe, sauf lorsque les obstacles les perturbent. Nous les désignons par le terme d'"agents réguliers" du fait de leur comportement assez prévisible.
2. Tous les agents possèdent une variable $a_0 = 1$. Il ont donc une composante chaotique forte et nous les désignons un peu abusivement sous le terme d'agents "chaotiques".

Nous pouvons comparer qualitativement sur les figures respectives 11.3(c) et 11.3(e) réalisées dans les mêmes conditions, que les agents réguliers adoptent après 500 pas de temps un comportement de groupe très ordonné et stable malgré un allongement du groupe d'agents le long des bords de l'environnement, tandis que les agents chaotiques ont un mouvement groupé plus compact par clusters. Ce dernier type de configuration à base d'agents chaotiques a tendance à explorer plus largement l'espace de l'environnement en changeant souvent de direction, les clusters se séparant et fusionnant régulièrement tout en conservant le même type de dynamique.

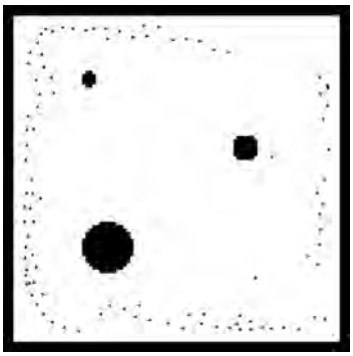
Population d'agents à profil dynamique uniformément distribué. Ici les agents ont une variable de contrôle tirée initialement selon une loi uniforme sur l'intervalle $[0, 1]$. On reprend finalement les conditions initiales du flocking exposé en section 10.3. De nombreuses situations peuvent apparaître dans cette configuration contrairement au cas précédent. Nous fixons dès le départ un couplage fort $\epsilon_2^0 = 0,9$ car nous savons par les analyses faites en section 10.3 qu'il



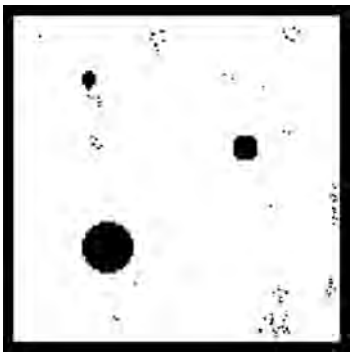
(a) $t = 0$. Phase d'initialisation



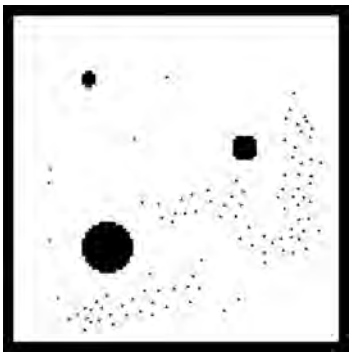
(b) $t = 500$. Agents réguliers avec un cône de perception d'angle $\psi = 2\pi$



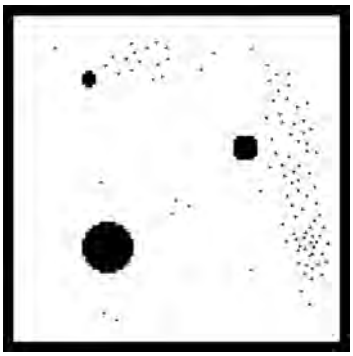
(c) $t = 500$. Agents réguliers avec un cône de perception d'angle $\psi = \frac{\pi}{2}$



(d) $t = 500$. Agents chaotiques avec un cône de perception d'angle $\psi = 2\pi$

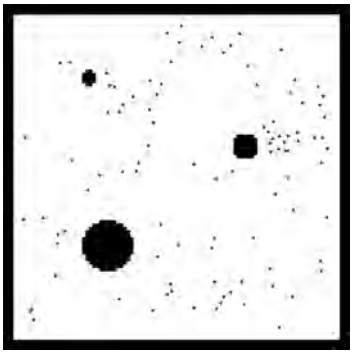


(e) $t = 500$. Agents chaotiques avec un cône de perception d'angle $\psi = \frac{\pi}{2}$

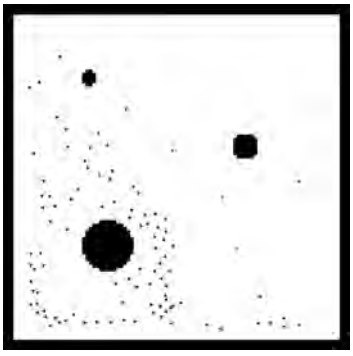


(f) $t = 1000$. Agents chaotiques avec un cône de perception d'angle $\psi = \frac{\pi}{2}$

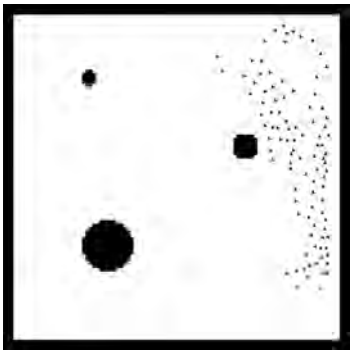
FIG. 11.3 – Simulations dans le cas où tous les agents ont même variable interne de contrôle a. La première ligne concerne le cas $a = 0$ des agents de Vicsek et le deuxième le cas $a = 1$ des agents chaotiques. Une analyse comparative est faite sur l'angle d'ouverture du cône de perception des obstacles.



(a) $t = 1000$, $R_2 = 2$, $R_1 = 4$, $\epsilon_2^0 = 0,9$.

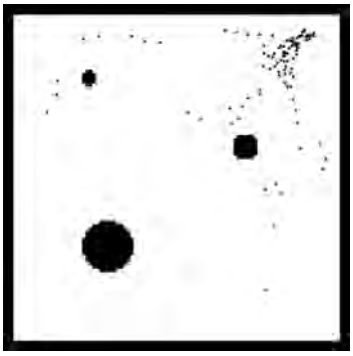


(b) $t = 1000$, $R_2 = 4$, $R_1 = 4$, $\epsilon_2^0 = 0,9$.

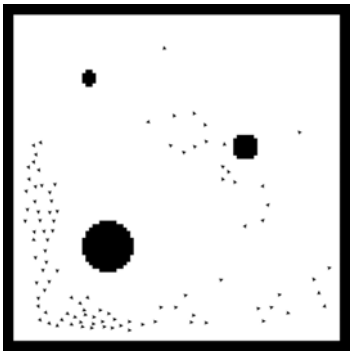


(c) $t = 1000$, $R_2 = 6$, $R_1 = 4$, $\epsilon_2^0 = 0,9$.

(d) $t = 660$, $R_2 = 4$, $R_1 = 4$, $\epsilon_2^0 = 0,9$.



(e) $t = 730$, $R_2 = 4$, $R_1 = 4$, $\epsilon_2^0 = 0,9$.



(f) $t = 1000$, $R_2 = 4$, $R_1 = 4$, $\epsilon_2^0 = 0,9$, loi normale $\mathcal{N}(0,6;(0,1)^2)$

FIG. 11.4 – Simulations dans le cas où les agents ont une variable interne de contrôle a tirée aléatoirement sur $[0,1]$. La première ligne montre l'effet du rayon de perception de synchronisation interne aux agents. La deuxième ligne montre un cas d'engorgement du groupe d'agents dans un angle de l'environnement et sa résolution. Enfin un cas de simulation avec une distribution normale des variables internes de contrôle.

est nécessaire pour qu'une synchronisation apparaisse. La figure 11.4 présente deux types de résultats : la première série de simulations (cf. figures 11.4(a), 11.4(b), 11.4(c)) tend à comparer l'effet de la taille du voisinage de perception $V_{i,2}$ de rayon R_2 sur l'aspect global du groupe. On constate une meilleure cohésion avec l'augmentation du rayon R_2 qui détermine l'étendue instantannée du couplage de synchronisation dans le groupe, et une meilleure robustesse du groupe vis-à-vis des obstacles : cette robustesse s'exprime par le fait qu'un obstacle casse en plusieurs morceaux un grand groupe d'agents pour un faible rayon R_2 tandis que ce groupe se scinde ne deux parties seulement pour des valeurs R_2 plus importantes.

La seconde série (cf. figures 11.4(d), 11.4(e)) rend compte d'une situation particulière où le groupe vient s'accumuler dans un angle de l'environnement. Dans ce cas critique, les agents se rapprochent dangereusement les uns des autres et la synchronisation additionnée à la loi de vitesse (11.15) ne suffit plus à maintenir une distance suffisante entre agents. Bien que le groupe finisse par sortir de l'ornière, cette situation s'avère risquée avec des robots matériels. Une voie

de résolution consiste à rendre loi de vitesse (11.15) dépendante du nombre d'agents présents dans le voisinage, élevé à une puissance entière (au carré par exemple, pour ralentir de façon drastique la vitesse des agents dès qu'ils se croisent).

Population d'agents à profil dynamique distribué selon une loi normale. Cette situation est intermédiaire aux deux précédentes puisqu'elle permet de moduler par une loi normale la distribution de la variable a des agents. On peut retrouver une distribution quasiment ponctuelle avec une loi très étroitement centrée sur sa moyenne avec un faible écart-type, ou une loi plate quasiment uniforme avec un fort écart-type. Entre ces deux paramétrages, on peut cibler des fenêtres de valeurs pour la variable de contrôle interne aux agents, ce qui permet de ne sélectionner qu'un certain type de profil dynamique. Là encore de nombreuses situations peuvent émerger et nous n'avons fait qu'effleurer cette phénoménologie.

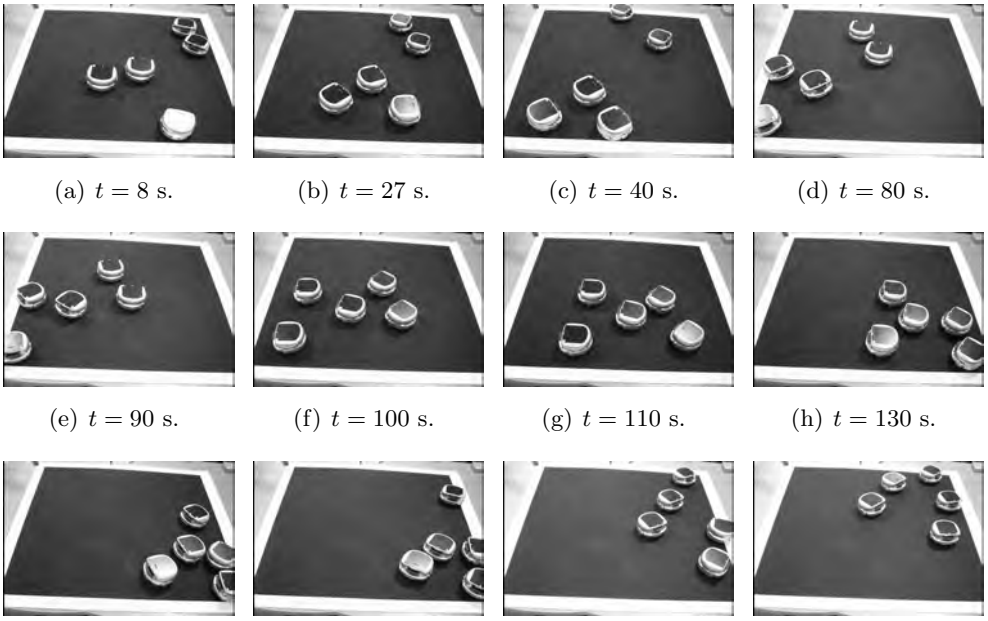
La figure 11.4(f) représente une simulation avec des agents sélectionnés selon la loi normale de moyenne $\mu = 0,6$ et d'écart-type $\sigma = 0,2$.

Discussions

Notre premier constat porte sur la comparaison de notre approche avec les deux autres approches présentées, Reynolds et Vicsek. Nous nous écartons d'une part de la vision particulaire du modèle de Vicsek par le fait que toute action de l'agent est motivée par des décisions internes ; il n'y a pas non plus d'utilisation d'une variation aléatoire probabiliste ajoutée à la direction principale de déplacement du groupe, mais une variation générée par le processus de l'application logistique interne à l'agent, qui caractérise l'autonomie de l'agent. Comparativement au modèle de Reynolds d'autre part, bien que nous utilisions également une notion de champ avec le SMA logistique, il ne s'agit en aucune façon de champs de potentiel semblables à ceux employés dans les modèles physiques, comme les champs de forces d'attraction ou de répulsion. Le calcul des forces dérivées de ces champs est en général coûteux du fait des calculs de distances entre agents, de l'odre de $O(N^2)$. Ici rien de tel, tout est basé sur le degré de synchronisation des variables internes de l'agent. Cette synchronisation ne prend en compte à chaque pas de temps qu'une partie des champs de l'environnement modifié par les agents en $O(1)$, ce qui réduit la complexité totale à $O(m\ N)$ où m désigne la complexité de l'opération de perception sur le voisinage considéré (convolution ou minimum) qui ne dépend que de la taille de celui-ci, ce qui revient à dire qu'il a un coût quasi-constant. Cela ramène le total à une complexité linéaire en temps.

L'analyse faite au niveau de nos simulations reste qualitative et les mesures liées à la détection de clusters comme l'entropie K subit trop de variations dues aux obstacles pour être un indicateur véritablement pertinent de la dynamique. En effet le nombre de clusters formés est très sensible à la présence permanente d'obstacles : la perception d'un obstacle par un agent rend l'agent asservi à l'obstacle et donc coupé momentanément de son cluster de synchronisation. Comme c'est le cas pour tous les agents en perception active avec l'obstacle, le nombre de clusters calculé fluctue en permanence.

Enfin, ces simulations préfigurent l'implantation de l'algorithme dans un essaim de robots



(a) $t = 8$ s. (b) $t = 27$ s. (c) $t = 40$ s. (d) $t = 80$ s.

(e) $t = 90$ s. (f) $t = 100$ s. (g) $t = 110$ s. (h) $t = 130$ s.

(i) $t = 170$ s. (j) $t = 200$ s. (k) $t = 240$ s. (l) $t = 257$ s.

FIG. 11.5 – Photos successives d'une séquence d'un flocking de 5 robots Khepera.

matériels. Les blocages théoriques du modélisation sont en partie levées et permettent d'envisager le portage de l'algorithme sur des robots autonomes. La section suivante présente de façon très succincte ce travail en cours.

11.2 Application au flocking de robots

Nous ne relaterons ici que le résultat de l'implantation de l'algorithme de flocking précédent sur une flottille de cinq robots khepera. Ces robots disposent de 9 capteurs infra-rouges en ceinture ainsi que deux autres sous le robot sur la partie avant. Dans notre implémentation, les capteurs en ceinture servent à la perception des autres robots et ceux à l'avant sont réservés à la détection des obstacles délimités par des bandes blanches : c'est la situation exactement inverse en termes de couleurs que celle de l'environnement logiciel précédent.

De plus les robots sont équipés d'un système d'odométrie précis qui leur permettent de calculer leurs positions successives et leur orientation courante (angle de direction nécessaire au calcul du déplacement). Nous ne disposions pas dans cette expérimentation de module embarqué de type compas électronique qui aurait donné à tout moment l'angle directionnel par rapport à une référence fixe. Cette information a donc été calculée par odométrie.

D'autre part, les robots n'étant pas dotés de système performant de communication, la valeur de l'état interne a transité via un réseau Wifi pour être redistribué sur les différents robots. Nous avons donc du utiliser un contrôle global pour gérer les perceptions locales et recréer virtuellement les champs de données de l'environnement. Cet artifice nécessaire n'est pas vraiment conforme à la modalité distribuée de l'approche. Cependant, il nous a permis de vérifier la pertinence de l'algorithme global et l'efficacité de la synchronisation sur la navigation

groupée des robots. Les collisions ont été très rares, alors qu'aucune gestion spécifique de cet événement n'existe dans notre modélisation.

La figure 11.5 relate en quelques clichés une des séquences d'un flocking de 5 robots sur un environnement de 1m×1m. Le rayon de perception des robots est fixé à 25cm. Un robot perçoit donc au maximum un demi-environnement. Au départ, les robots sont répartis dans l'environnement de façon aléatoire après avoir été étalonnés relativement à un angle de référence commun. On observe un regroupement des 5 robots après une centaine de secondes, puis une situation de blocage apparaît dans un des angles à 130 secondes, similaire à celle évoquée en figure 11.4(d). Au bout d'une cinquantaine de secondes la situation se débloque spontanément sans collision. Le groupe de robots est ensuite synchronisé et navigue en flocking.

Nous considérons comme concluante la vérification d'une implantation effective et efficace de notre algorithme de flocking dans des robots matériels autonomes. Ce travail est encore à l'étude et nous pensons pouvoir augmenter le nombre de robots de façon significative sans que cela ne soit problématique. Nous pourrons ainsi vérifier la propriété de scalabilité recherchée pour la robotique en essaim.

11.3 Apparition de files d'agents par adaptation

Dans les premières simulations de flocking que nous avons de réalisées (cf. section 10.3), seule l'auto-organisation du groupe par synchronisation a été mise en évidence. Dans le flocking avec obstacles précédent (cf. 11.1), un mécanisme d'adaptation de l'environnement a impacté la variable de couplage des agents via un champ de couplage exogène. On s'intéresse maintenant à déterminer ce qu'un mécanisme d'adaptation sur le paramètre de contrôle des agents peut produire dans un environnement contenant un champ de contrôle aléatoire. Nous montrons ainsi qu'une file d'agents logistiques apparaît spontanément. Cet exemple simple permet d'introduire le principe des mécanismes adaptatifs de contrôle dans notre modèle.

Instanciation du SMAL.

Le modèle SMAL hérite du modèle exposé en section 10.3 avec quelques modifications et ajouts. Le modèle se décompose maintenant comme suit.

Environnement. L'espace E de l'environnement est le même tore continu de dimension 2 qu'en 10.3. On lui adjoint un espace discrétisé E' qui correspond à une grille de sites de taille $L \times L$ en fixant $L > 1$ entier. On indicera la localisation d'un site sur E' par un couple d'entiers (k, l).

L'espace est doté d'un champ de contrôle noté $\mathcal{C}$ défini et initialisé par :

$$\begin{aligned} \mathcal{C} : E' &\to [0,1] \\ (k,l) &\mapsto \mathcal{C}(k,l) = p_{k+l} \end{aligned}$$

où p_{k+l} est la $(k+l)$ème réalisation d'une variable aléatoire uniforme sur $[0,1]$. Chaque site de ce champ contient un réel quelconque entre 0 et 1. Ce champ a la propriété d'être persistant et stationnaire (pas d'évolution dans le temps).

L'agent logistique.

État interne de l'agent. L'agent possède une fonction de perception sur sa composante de contrôle interne a. Son état interne s'écrit donc

$$s(t) = \left\langle x_i^t, a_i^t, \epsilon^0 \right\rangle \tag{11.18}$$

car a dépend désormais de la perception sur $\mathcal{C}$ au cours du temps. Autrement dit l'agent va donc être sous le contrôle du champ $\mathcal{C}$. C'est dans cette perception que réside l'essence même du mécanisme d'adaptation traité dans cette section.

Perception. Nous ne donnons ici que la perception selon a, le reste des mécanismes étant identiques au cas traité en section 10.3. Cette perception s'écrit au moyen de l'opérateur de moyenne (8.22) de façon semblable à la perception $P_{x,i}$ déjà vue pour le flocking pour un agent i à la différence que la moyenne se faite sur le nombre de cellules de E' comprises dans le voisinage V_i :

$$P_{a,i}(\sigma^t) = \langle \mathcal{C} \rangle^t_{V_i,E'} \tag{11.19}$$

Calcul de la décision de l'agent. Les équations d'évolution d'état interne pour l'agent i sont données par les équations (la perception est ici remplacée par son expression) :

$$\epsilon_i^{t+1} = \epsilon^0$$

$$a_i^{t+1} = (1-\epsilon^0)a_i^t + \epsilon^0 \ \langle \mathcal{C} \rangle^t_{V_i,E'} \tag{11.20}$$

$$x_i^{t+1} = (1-\epsilon^0)f(x_i^t, a_i^{t+1}) + \epsilon^0 \ \langle \mathcal{X} \rangle^t_{V_i,\mathcal{N}} \tag{11.21}$$

où f prend la forme (10.2). Son diagramme de bifurcation est tracé en figure 10.1.

Actions de l'agent. Avant le calcul de la nouvelle position de l'agent dans E, l'agent met à jour la norme de sa vitesse selon la formule : $v_i^{t+1} = x_i^{t+1} * v_0$
L'agent calcule ensuite son déplacement et sa position en coordonnées polaires par l'équation similaire à l'équation (10.8). On a donc finalement le système de mise à jour pour les positions et vitesses suivantes :

$$\begin{cases} \alpha_i^{t+1} = 2\pi \ x_i^{t+1} \\ v_i^{t+1} = x_i^{t+1} * v_0 \\ \mathbf{r}_i^{t+1} = \mathbf{r}_i^t + v_i^{t+1} \ \mathbf{u}_{\alpha_i^{t+1}} \end{cases} \tag{11.22}$$

Réaction de l'environnement. Pas de changement majeur concernant la réaction de l'environnement par rapport à la section 10.3.

Résultats de simulations et interprétation.

Résultats. Le résultat d'une simulation est visualisé sur la figure 11.6 dont les paramètres sont précisés en légende. Les conditions initiales consistent à placer les agents aléatoirement dans l'environnement avec des variables internes tirées aléatoirement de façon uniforme sur $[0, 1]$, sauf pour la variable de couplage identique pour tous les agents. On voit clairement apparaître une configuration assez différente d'un flocking comme celui de la section 10.3. Le groupe d'agents

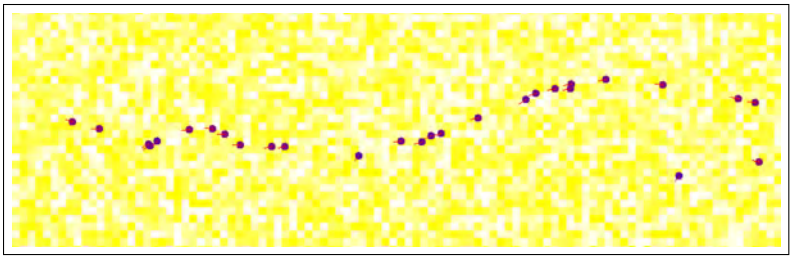


FIG. 11.6 – Simulation obtenue au bout de 2160 pas de temps avec : environnement 100×100, $N = 50$ agents, $\epsilon^0 = 0,20$, voisinage de perception de rayon $R = 20$

s'est étiré dans l'espace en une longue file semblable à ce que l'on pourrait observer pour une piste de fourmis. Là s'arrête l'analogie, mais le principe est important puisque cet effet apparaît pour les raisons conjuguées suivantes :

- L'adaptation à l'environnement crée une variation de a autour de la valeur $0,5$ puisque la perception agit sur un voisinage qui moyenne les valeurs perçues dans l'environnement. Celles-ci étant uniformément distribuées sur $[0, 1]$, la moyenne de la variable aléatoire p étant $0,5$, a oscille dans un voisinage autour de $0,5$.
- Dans cette zone de valeurs, $\bar{x} = 0,5$ est un point fixe superstable car $f(0,5) = 0,5$ et $f'(0,5) = 0$ pour $a = 0,5$.
- La vitesse de l'agent dépend directement de la valeur de sa variable de décision x et voit donc sa valeur varier également autour de $\frac{v_0}{2}$. Cette variation permet une régulation du groupe en fonction du terrain.
- L'aspect fondamental ici est lié au couplage $\epsilon^0 = 0,2$ dans la simulation présentée. Il y a un faible couplage sur les états internes x, en revanche cette valeur est importante du point de vue de la variable a qui est très rapidement asservie par le champ. Ce deuxième effet d'asservissement par le champ $\mathcal{C}$ l'emporte ici sur le couplage de type flocking.

Ce dernier aspect contribue à expliquer la forme de longues chaînes d'agents, car c'est le champ $\mathcal{C}$ qui contrôle dès lors la trajectoire des agents. La direction est donnée par x donc varie autour de π, reste à déterminer où passent les agents : chaque agent emprunte la même trajectoire ce qui laisse supposer une forme d'optimisation locale.

Résultats analytiques. Cette dernière affirmation est corroborée par le fait que le point $(\bar{a}, \bar{x}) = (\frac{1}{2}, \frac{1}{2})$ est un point fixe superstable du diagramme de bifurcation 10.1 que nous noterons par la suite $\overline{ax}$. Les agents adoptent le chemin qui leur permet de rester sur ce point fixe. En effet, on peut montrer qu'autour de ce point fixe toute perturbation tend à revenir à ce point.

Pour le démontrer remarquons que l'équation (11.20) est indépendante de l'équation (11.21) ce qui n'est pas réciproque. On pourra donc traiter leurs propriétés l'une après l'autre.

Proposition 11.3.1. *Pour un voisinage de perception suffisamment grand et un couplage $\epsilon^0 < 1$, la variable de contrôle a_i de tout agent $i \in \{1, \cdots, N\}$ converge en loi vers un processus gaussien de moyenne $\frac{1}{2}$ et d'écart-type $\sigma^* = \frac{1}{2\ R\ \sqrt{3\pi}}$, où R est le rayon de voisinage de perception de l'agent.*

Preuve. Réécrivons l'équation (11.20) sous la forme du système dynamique :

$$\begin{cases} a_i^{t+1} = (1-\epsilon^0)a_i^t + \epsilon^0 z_i^t \\ z_i^{t+1} = \langle \mathcal{C} \rangle_{V_i,E'}^{t+1} \end{cases} \tag{11.23}$$

Rappelons tout d'abord que $\langle \mathcal{C} \rangle_{V_i,E'}^{t+1}$ correspond à une moyenne glissante sur le champ $\mathcal{C}$. Ce champ est stationnaire mais l'agent se déplaçant dans l'environnement, à chaque pas de temps la perception du champ change. Évaluons la proportion de ce changement : chaque cellule du champ $\mathcal{C}$ est une réalisation d'un processus uniforme p sur $[0,1]$ qui a pour moyenne $\mu = \frac{1}{2}$ et pour écart-type $\sigma = \frac{1}{\sqrt{12}}$. Par application du théorème de la limite centrale, on peut assimiler la moyenne glissante sur un voisinage V contenant $N_V = \pi\ R^2$ cellules en deux dimensions —donc N_V processus uniformes—, à un processus aléatoire que l'on écrira $p^* = \frac{1}{2} + \eta$ où η est une loi normale de moyenne nulle et d'écart-type $\sigma^* = \frac{\sigma}{\sqrt{N_V}}$, soit $\sigma^* = \frac{1}{2\ R\ \sqrt{3\pi}} \approx \frac{1}{6\ R}$ dans notre problème. Pour $R = 10$, l'écart-type est égal à $\sigma^* \approx 0,017$ donc suffisamment faible pour considérer en première approximation la perception de la moyenne glissante comme stationnaire. En acceptant l'approximation de stationnarité de la moyenne glissante $\langle \mathcal{C} \rangle_{V_i,E'} = \frac{1}{2} + \eta$, cette valeur est constante et on peut calculer le point fixe de (11.23) qui correspond au vecteur $(\bar{a}, \bar{z})^T$ qui vérifie le système d'équations :

$$\begin{cases} \bar{a} = (1-\epsilon^0)\ \bar{a} + \epsilon^0\ \bar{z} \\ \bar{z} = \frac{1}{2} + \eta \end{cases} \tag{11.24}$$

On en tire facilement le point fixe $(\bar{a}, \bar{z}) = (\frac{1}{2} + \eta, \frac{1}{2} + \eta)$.

La matrice jacobienne correspondante au système (11.23) est indépendante du point fixe sur lequel on raisonne :

$$J(z,a) = \begin{bmatrix} (1-\epsilon^0) & \epsilon^0 \\ 0 & 0 \end{bmatrix} \tag{11.25}$$

dont les valeurs propres sont clairement $\mu_1 = 0$ et $\mu_2 = (1-\epsilon^0)$, dont les valeurs vérifient $|\mu_{1,2}| < 1$. Le système est stable vis-à-vis du point fixe $(\bar{a}, \bar{z})$. Ce résultat reste valable quel que soit la valeur du point fixe $(\bar{a}, \bar{z})$ et notamment quel que soit η. Ce qui permet de conclure. □

La suite consiste à vérifier la proposition suivante :

Proposition 11.3.2. *Pour un voisinage de perception suffisamment grand et un couplage $\epsilon^0 < 1$, après la phase de régime transitoire du système, le point $\overline{ax} = (\frac{1}{2}, \frac{1}{2})$ est point fixe stable du système* (11.21).

Preuve. Remarquons d'abord que la proposition précédente induit que pour tout agent i, la variable interne x_i se trouve dans un voisinage du point $\overline{ax}$ (cf. le rectangle bleu de la figure 11.7). Cela est justifié par la valeur de couplage assez faible que nous considérons ainsi que la stabilité et l'attraction forte du point $\bar{x} = \frac{1}{2}$ révélée dans le diagramme des coefficients de Lyapunov 5.7 de l'application logistique : $\lambda_0 \to -\infty$ lorsque $\bar{x} \to \frac{1}{2}$.

Dans un premier temps, nous faisons de nouveau une approximation, une approximation de champ moyen tendant à coupler l'agent avec une constante H proche de la valeur $\frac{1}{2}$. Nous émettons l'hypothèse d'un régime stationnaire où les autres agents influent globalement de façon invariante. En fait, H suit le type de loi statistique vue dans la proposition précédente. Ainsi l'équation (11.21) se ramène à un système d'équations pour l'agent i :

$$\begin{cases} a_i^{t+1} = (1-\epsilon^0)a_i^t + \epsilon^0 \ \langle \mathcal{C} \rangle_{V_i,E'} \\ x_i^{t+1} = (1-\epsilon^0)f(x_i^t, a_i^{t+1}) + \epsilon^0 H \end{cases} \tag{11.26}$$

En simplifiant encore davantage, à savoir si on pose $\langle \mathcal{C} \rangle_{V_i,E'} = \frac{1}{2}$ et $H = \frac{1}{2}$ en élimant leur variation statistique au premier ordre, alors le vecteur $(\bar{a}, \bar{x})^T = (\frac{1}{2}, \frac{1}{2})^T$ est point fixe du système dynamique (11.26). La matrice jacobienne au point fixe s'écrit :

$$J(\bar{a}, \bar{x}) = (1-\epsilon^0) \begin{bmatrix} 1 & 0 \\ \frac{\partial f}{\partial a}(\bar{a}, \bar{x}) & \frac{\partial f}{\partial x}(\bar{a}, \bar{x}) \end{bmatrix} \tag{11.27}$$

or on vérifie qu'au point fixe :

$$\begin{cases} \frac{\partial f}{\partial a}(\bar{a}, \bar{x}) = 4 \ \bar{x} \ (1-\bar{x}) = 1 \\ \frac{\partial f}{\partial x}(\bar{a}, \bar{x}) = 4\bar{a}(1-2\bar{x}) = 0 \end{cases} \tag{11.28}$$

Les deux valeur propres sont donc ici $(1-\epsilon^0) < 1$ et 0, ce qui confirme la stabilité du point fixe $(\frac{1}{2}, \frac{1}{2})$.

On suppose maintenant que de petites variations sur $\langle \mathcal{C} \rangle_{V_i,E'} = \frac{1}{2} + \eta$ et $H = \frac{1}{2} + \delta H$ se répercutent sur des variations autour de $\bar{a} = \frac{1}{2} + \delta\bar{a}$ et de $\bar{x} = \frac{1}{2} + \delta\bar{x}$. Il suffit de vérifier comment ces variations influent sur les valeurs propres du jacobien (11.27). On a autour du point fixe à considérer uniquement un développement limité autour de la dérivée partielle selon x de (11.28). On pose :
$G(\bar{a}, \bar{x}) = \frac{\partial f}{\partial x}(\bar{a}, \bar{x}) = 4\bar{a}(1-2\bar{x})$
On a donc le développement au premier ordre de G suivant :

$$\begin{aligned} dG(\bar{a}, \bar{x}) &= \frac{\partial G}{\partial \bar{a}}(\bar{a}, \bar{x})d\bar{a} + \frac{\partial G}{\partial \bar{x}}(\bar{a}, \bar{x})d\bar{x} \\ &= -8 \ \bar{a} \ d\bar{x} + 4 \ (1 - 2 \ \bar{x}) \ d\bar{a} \end{aligned}$$

Au point fixe $(\frac{1}{2}, \frac{1}{2})$ considéré, l'écriture précédente se résume à

$$dG(\frac{1}{2}, \frac{1}{2}) = -4 \ d\bar{x} \tag{11.29}$$

Ce développement montre que la variation de G s'oppose à la variation de $\bar{x}$, la variation de $\bar{a}$ n'ayant pas d'influence au point fixe à l'ordre 1, et elle tend donc à ramener le système vers le point fixe stable $(\frac{1}{2}, \frac{1}{2})$. Cette dernière constatation conclue la proposition. □

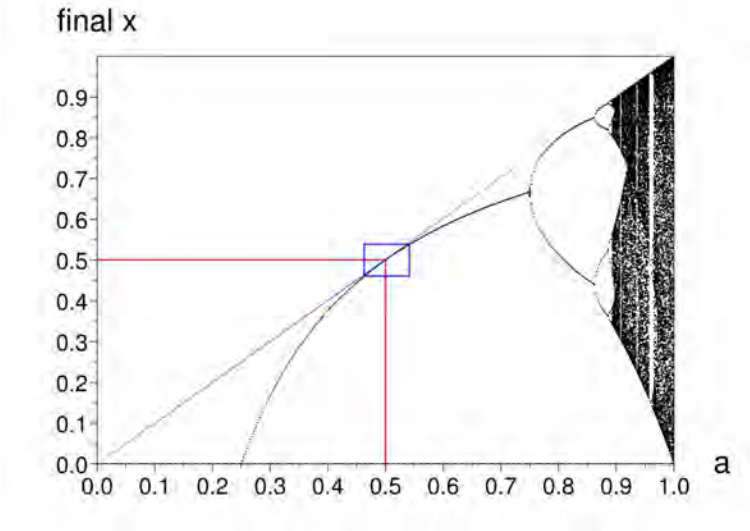


FIG. 11.7 – Point $(\frac{1}{2}, \frac{1}{2})$ superstable sur le diagramme de bifurcation de l'application logistique $x_{t+1} = f(x_t, a) = 4\ a\ x_t\ (1 - x_t)$

Conclusion.

Cette asservissement par le champ est très efficace et impose à l'ensemble des agents une configuration autour du point superstable $(\bar{a} = \frac{1}{2}, \bar{x} = \frac{1}{2})$ et rend les agents ordonnés par le champ. La forme émergente de file indienne prise par les agents résulte de cette attraction vers ce point fixe cumulée au couplage avec les autres agents. La file se déforme dans l'environnement pour converger au mieux vers le point fixe attracteur du système malgré les variations autour de celui-ci.

11.4 Application à un système proie-prédateurs vu comme problème d'optimisation

Dans cette section, nous explorons un mécanisme d'adaptation assez semblable au précédent avec des lois d'évolution différentes. En résulte une simulation d'un système proie-prédateurs [Charrier *et al.*, 2009a, Charrier *et al.*, 2009b], qui s'avère être dans son fonctionnement très proche des problèmes d'optimisation de fonctions non linéaires en optimisation particulaire.

Description du problème proie-prédateurs.

On se propose de traiter d'un problème proie-prédateurs comme autre illustration des mécanismes d'adaptation vers des problèmes d'optimisation. Les problèmes proies-prédateurs dans le cadre de l'informatique ont été initiés par l'article [Benda *et al.*, 1986]. Nous proposons ici un problème un peu différent, sans toutefois être si éloigné d'une situation biologique :

Définition 11.4.1. *Le problème consiste pour un essaim de N prédateurs (N est de l'ordre*

de quelques dizaines ou plus) à entourer une ou quelques proies. La proie module sa vitesse de façon inversement proportionnelle au nombre de prédateurs entrant dans son voisinage. La proie est déclarée capturée lorsque 10 *prédateurs se trouvent simultanément dans son voisinage de Moore (voisinage* 8*). Cela définit l'état de terminaison du problème. Lorsque plusieurs proies sont présentes, l'état de terminaison survient lorsque toutes les proies sont bloquées.*

La proie ne fait ici que diminuer sa vitesse, elle ne change pas sa direction. Ce cas basique est certes peu réaliste avec des situations biologiques, cependant nous cherchons à montrer la capacité d'agrégation de l'essaim de prédateurs en situation dynamique. La métaphore biologique correspond à certaines situations de capture chez les insectes sociaux où une grosse proie pourra être immobilisée par un nombre important d'individus qui viennent s'agréger au processus de capture.

Instanciation du modèle.

Objectif. Toujours sur la base du SMAL, et dans la lignée du modèle précédent, nous présentons les éléments qui vont constituer les fondements du système étudié. L'idée générale est de générer deux types de comportements pour les agents :

- un comportement aléatoire pour explorer l'espace,
- un comportement d'agrégation pour capturer la proie.

Le premier comportement est assuré par une variable interne de contrôle maintenue à la valeur 1 lorsqu'aucune proie n'est perçue grâce au champ de contrôle $\mathcal{C}$. Le second est généré par un mécanisme d'adaptation à la proie par perception de la proie.

Environnement. L'espace E de l'environnement est le même qu'en section précédente 11.3 en $2D$. Il comprend notamment un espace discrétisé E' de taille $L \times L$.

L'espace est doté du champ de contrôle $\mathcal{C}$ défini et initialisé à 1 partout :

$$\begin{aligned} &\mathcal{C} : E' \to [0,1] \\ &(k,l) \mapsto \mathcal{C}(k,l) \text{ avec } \mathcal{C}(k,l)^0 = 1,\ \forall (k,l) \in E' \end{aligned}$$

Nous expliquons plus loin la raison de cette initialisation au regard des mécanismes d'adaptation.

Les agents logistiques prédateurs.

État interne de l'agent. L'état interne d'un agent i s'écrit de nouveau :

$$s(t) = \left\langle x_i^t, a_i^t, \epsilon^0 \right\rangle \tag{11.30}$$

a dépend de la perception sur $\mathcal{C}$ au cours du temps. Initialement, on a $a_i^0 = 1$.

Perceptions. Nous ne traitons de nouveau que la perception selon pour la composante a, le reste des mécanismes étant identique aux cas précédents, notamment la perception $P_{x,i}(\sigma^t)$. Cependant nous distinguons le voisinage de perception selon la composante x que nous noterons $V^{\mathcal{X}}$ et celui selon la composante a noté $V^{\mathcal{C}}$. Nous rappelons que le voisinage $V^{\mathcal{X}}$ intervient dans l'interaction de couplage entre agents-prédateurs. La perception selon a utilise un opérateur min avec la formule :

$$P_{a,i}(\sigma^t) = \min_{(k,l)\in V_i^{\mathcal{C}}} \{\mathcal{C}^t(k,l)\} \tag{11.31}$$

Cette perception par opérateur minimum va permettre aux prédateurs de se diriger plus directement sur la proie et détecter sa présence de façon plus précise.

Calcul de la décision de l'agent. Les équations d'évolution d'état interne pour l'agent i sont données par les équations :

$$\begin{aligned} \epsilon_i^{t+1} &= \epsilon^0 \\ a^{t+1} &= (1-\epsilon^0)a^t + \epsilon^0 \min_{k\in V_i^{\mathcal{C}}} \{\mathcal{C}_k(t)\} && (11.32) \\ x_i^{t+1} &= (1-\epsilon^0) f(x_i^t, a_i^{t+1}) + \epsilon^0 \ \langle \mathcal{X} \rangle^t_{V_i^{\mathcal{X}},\mathcal{N}} && (11.33) \end{aligned}$$

La mise à jour de a ne modifie sa valeur intiale seulement lorsqu'il existe une valeur plus faible dans l'environnement. L'application logistique f impliquée ici est différente mais possède les mêmes caractéristiques qualitatives que la forme classique utilisée jusqu'ici. Elle est définie par la récurrence :

$$x^{t+1} = f(x^t, a^t) = a^t \ (1 - 2 \ x^t)^2 \tag{11.34}$$

Son diagramme de bifurcation est donné en 11.8.

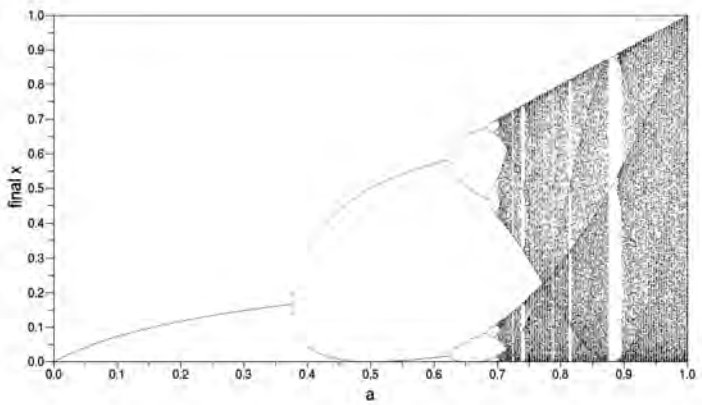


Fig. 11.8 – Diagramme de bifurcation de l'application quadratique $x^{t+1} = a \ (2 \ x^t - 1)^2$, calculée avec 500 iterations sur 500 échantillons dans l'intervalle $[0, 1]$

Actions de l'agent. Par rapport au cas du système 11.22, la loi d'évolution de α_i est modifiée. Elle déstructure la synchronisation géométrique mais permet une exploration plus efficace de l'espace sans direction privilégiée. L'agent calcule ensuite sa vitesse et sa position en coordonnées polaires par le système d'équations de mise à jour :

$$\begin{cases} \alpha_i^{t+1} = \alpha_i^t + (x_i^t - \frac{1}{2})\ \varphi(mod 2\pi) \\ v_i^{t+1} = x_i^{t+1}\ v_0 \\ \mathbf{r}_i^{t+1} = \mathbf{r}_i^t + v_i^{t+1}\ \mathbf{u}_{\alpha_i^{t+1}} \\ \theta_i^{t+1} = \arg(\mathbf{r}_i^{t+1}) \end{cases} \tag{11.35}$$

où $v_0 = 1$ unité de distance par pas de temps et φ est un paramètre incrémental de l'angle. En coordonnées cartésiennes, cela revient au système :

$$\begin{cases} v_{xi}^{t+1} = x_i^{t+1}\ v_0\ \cos(\theta_i^{t+1}) \\ v_{yi}^{t+1} = x_i^{t+1}\ v_0\ \sin(\theta_i^{t+1}) \\ X_i^{t+1} = X_i^t + v_{xi}^{t+1}\ (mod\ L) \\ Y_i^{t+1} = Y_i^t + v_{yi}^{t+1}\ (mod\ L) \end{cases} \tag{11.36}$$

Réactions de l'environnement. La réaction de l'environnement, outre les mises à jour des positions de chaque agent, et des autres champs de données $\mathcal{X}$ et $\mathcal{N}$, réalise la mise à jour du champ $\mathcal{C}$ à partir du champ d'influence de la proie $\widetilde{\mathcal{C}}$.

$$\mathcal{C}^{t+1}(k,l) = 1 - \widetilde{\mathcal{C}}^{t+1}(k,l) \tag{11.37}$$

Ainsi on obtient un champ rempli de valeur 1 et une décroissance progressive des valeurs autour de la proie jusqu'à 0 sur la proie. Cette constitution du champ $\mathcal{C}$ est importante pour comprendre la dynamique obtenue sur les agents. Nous allons voir à présent comment ce champ d'influence $\widetilde{\mathcal{C}}$ est produit par la proie.

Les agents logistiques proies. Les agents proies sont moins complexes dynamiquement, mais ont un effet de marquage avec diffusion dans l'environnement. Ce mécanisme est essentiel pour la perception des agents prédateurs.

État interne des proies. L'état interne des proies est donné par le triplet $s_p(t) = \langle x_p^t, a_p^0, \epsilon_p^0 \rangle$. La proie est en fait un agent identique à ceux rencontrés pour le flocking. On a les transitions d'état internes suivantes :

$$\epsilon_p^{t+1} = \epsilon_p^0 = 0$$

$$a_p^{t+1} = a_p^0 \equiv \text{ tiré aléatoirement de façon uniforme sur } [0, \frac{1}{4}] \text{ à } t = 0 \tag{11.38}$$

$$x_p^{t+1} = f(x_p^t, a_p^0) \tag{11.39}$$

L'évolution de la variable de décision interne à la proie est donc des plus simples et converge très vite vers un point fixe (cf. 11.8).

Perception des proies. On définit un voisinage spécifique à l'agent proie noté V_p. Ici ce voisinage se restreint au voisinage de Moore dans E' l'espace de l'environnement discrétisé comprenant 8 cellules périphériques. Dans ce voisinage, la proie capte le nombre de prédateurs selon l'opérateur de sommation du champ de position cumulée des prédateurs (cf. section 8.2.6) selon la formulation :

$$P_p^t = \Sigma_{V_p} \mathcal{N}^t \tag{11.40}$$

Actions de la proie. Les actions sont de deux ordres :

– L'action de la proie est d'abord une action de déplacement, les variables selon :

$$\begin{cases} \theta_p^{t+1} = 4\ x_p^t\ 2\pi \text{ (pour donner un angle dans } [0, 2\pi]) \\ v_p^{t+1} = \frac{v^0}{(1+P_p^{t+1})} \\ \mathbf{r}_p^{t+1} = \mathbf{r}_p^t + v_p^{t+1}\ \mathbf{u}_{\theta_p^{t+1}} \end{cases} \tag{11.41}$$

Comme on le voit, la vitesse de la proie décroît à mesure que le nombre de prédateurs dans son voisinage de Moore augmente. De plus l'angle de déplacement est constant puisque la valeur x_p atteint très vite un point fixe (cf. équation 11.38).

– L'autre action importante de la proie est de marquer sa position dans le champ d'influence $\widetilde{\delta}_p$ discret, où le couple (k_p, l_p) représente la coordonnée discrète dans E' de la proie :

$$\widetilde{\delta}_p^{t+1}(k,l) = \begin{cases} 1 \text{ si } (k,l) = (k_p^{t+1}, l_p^{t+1}) \\ 0 \text{ sinon} \end{cases} \tag{11.42}$$

Le champ $\widetilde{\delta}_p$ est donc rempli de 0 sauf à la position de la proie.

Le reste de l'action de la proie consiste en un opérateur d'influence qui agit à la transformation du champ de contrôle $\mathcal{C}$.

Influence de la proie sur l'environnement. Nous considérons en fait que le procédé de perception de la proie par les prédateurs revient à définir une zone de marquage de la proie dans le champ de contrôle de l'environnement. Cette zone correspond à un "voisinage de présence" de la proie qui définit sa limite d'observabilité par les prédateurs. Cette zone dépend de l'acuité des prédateurs à percevoir la proie : elle est d'autant plus étendue que les prédateurs ont de grandes capacités de perception. L'opérateur d'influence de la proie crée dans un premier temps un champ d'influence $\widetilde{\mathcal{C}}$ non persistant et discret, défini par la convolution d'un filtre sur le champ de position de la proie :

$$\widetilde{\mathcal{C}}^{t+1}(k,l) = \alpha\ \widetilde{\delta}_p^{t+1}(k,l) * K(k,l) \tag{11.43}$$

$$= \alpha \sum_u \sum_v \widetilde{\delta}_p^{t+1}(k-u, l-v)\ K(u,v) \tag{11.44}$$

où K est le noyau ou masque du filtre de convolution et α un paramètre d'ajustement. Cette technique est rencontrée fréquemment dans le domaine du traitement d'images. Le noyau peut être représenté comme une matrice ici de taille (m, m), avec $m < L$. Nous donnons par exemple

ci-dessous une matrice de taille 5 utilisée pour réaliser cette convolution. K est un filtre de type conique, mais un filtre gaussien aurait pu convenir également :

$$\begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 2 & 2 & 2 & 0 \\ 1 & 2 & 5 & 2 & 1 \\ 0 & 2 & 2 & 2 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix} \tag{11.45}$$

$\alpha = \frac{1}{5}$ ici pour reproduire la valeur 1 là où se trouve la proie et une décroissance autour de la proie. Le reste du champ est nul. L'équation (11.37) termine la mise à jour du champ de contrôle dans l'environnement.

Résultats de simulations et interprétation.

Exemple de simulation. Les trois clichés de la figure 11.9 montre une simulation sans terminaison à trois pas de temps successifs en gros plan centré sur la proie. Les paramètres pris pour cette simulation sont résumés dans le tableau 11.1.

Au début les agents prédateurs sont distribués aléatoirement dans l'environnement, leur couleur est relative à leur variable interne x : le blanc correspond à la valeur 0, le noir à la valeur 1. Entre les deux on affecte un degré de tonalité rouge. La proie est désignée par un disque blanc et son marquage de présence dans l'environnement par des disques concentriques de couleur brune. Le bleu de l'environnement correspond à la valeur 1 répartie dans tout le champ $\mathcal{C}$, excepté aux abords de la proie. Au pas de temps $t = 500$, la proie est entourée par

(a) visualisation de la simulation à $t = 0$

(b) La proie a été capturée $t = 500$

(c) L'ensemble du groupe de prédateurs s'est agrégé autour de la proie à $t = 10000$

FIG. 11.9 – Simulation du problème proie-prédateurs avec $N = 50$ predateurs. La proie est repérésentée par un petit disque et son voisinage de "présence" est marqué dans l'environnement. Les parameters sont $\epsilon^0 = 0.2$, le masque de marquage de la proie est de taille $(11, 11)$, le rayon de perception des prédateurs est $R = 1.0$

dix prédateurs dans son voisinage de Moore et la simulation devrait s'arrêter. Mais nous avons poursuivi la simulation jusqu'à une phase où tous les agents se sont agrégés sur la proie.

Objet	type	valeur
Environnement	tore	100×100
Agent	Prédateur	50
	ϵ^0	0.2
	$V^{\mathcal{C}}$	$R = 1$
	$V^{\mathcal{X}}$	$R = 2$
	v^0	1 unités/pas de temps
	Incrément d'angle φ	30°
Agent	Proie	1
	Masque diffusion	$(11, 11)$
	v_p	$0,05$ unités/pas de temps

TAB. 11.1 – Paramètres de simulation proie-prédateurs.

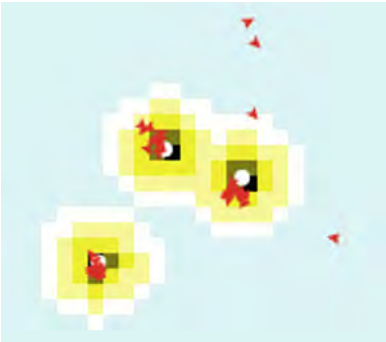


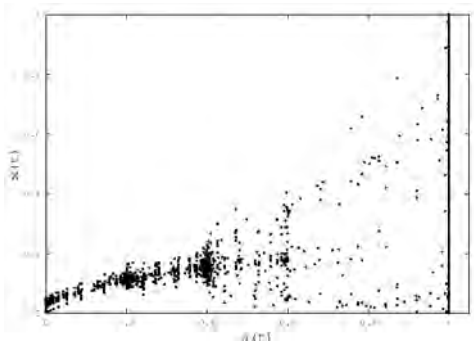
FIG. 11.10 – Image d'une simulation avec capture de trois proies

La figure 11.10 montre une simulation comportant 3 proies qui sont progressivement entourées de prédateurs. On observe pas ici de focalisation sur une proie en particulier puis sur une autre... Chaque proie a la même influence. Une extension possible serait de moduler la taille des voisinages de présence en fonction de l'importance relative de la proie pour vérifier si la dynamique tient compte de cette dissymétrie. Mais cette expérience n'a pas été effectuée ici. Notre étude s'est concentrée sur la conception du mécanisme d'adaptation pour réaliser un comportement de capture chez les prédateurs.

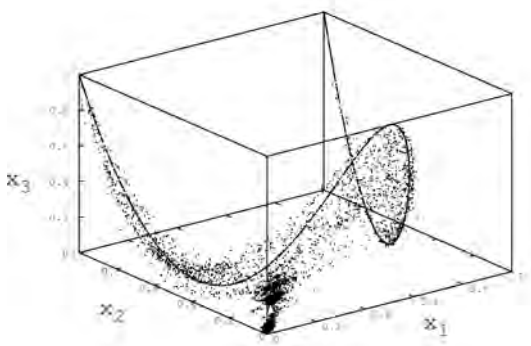
Quelques éléments d'analyse.

Interprétation du mécanisme. Le coeur de la conception du mécanisme impliqué dans ce problème s'éclaire à la lecture du diagramme de bifurcation de l'application logistique 11.8 et sachant la forme de l'asservissement des agents-prédateurs au champ $\mathcal{C}$ via la variable de contrôle interne. En effet, tant qu'une proie n'a pas été détectée l'agent explore son espace grâce à une valeur de sa variable de contrôle stationnaire et égale à 1, donc dans un mode chaotique (cf. diagramme 11.8). Le chaos conjugué à la loi de déplacement des agents (cf. 11.35) permet ici une exploration complète de l'espace par sa propriété d'ergodicité (cf. annexe5.3).

Dès qu'une variation du champ $\mathcal{C}$ est détectée par l'agent-prédateur dans son voisinage de perception selon la fonction 11.31, la proie est proche et la perception a l'effet de faire décroître la valeur de la variable a_i de l'agent. Plus l'agent se rapproche du centre de la proie, plus sa variable a_i diminue jusqu'à la limite nulle exactement sur la proie. La variable de décision prend des valeurs successives qui diminuent également progressivement (cf. diagramme de bifurcation 11.11(a)). La loi de déplacement 11.35 enferme alors la trajectoire des agents autour de sa



(a) Suivi de la dynamique d'un agent-prédateur par son diagramme de bifurcation. L'évolution se lit de droite à gauche dans ce diagramme.



(b) Reconstruction 3D de l'attracteur pour un agent-prédateur : $X_1 = x^t$, $X_2 = x^{t+1}$, $X_3 = x^{t+2}$

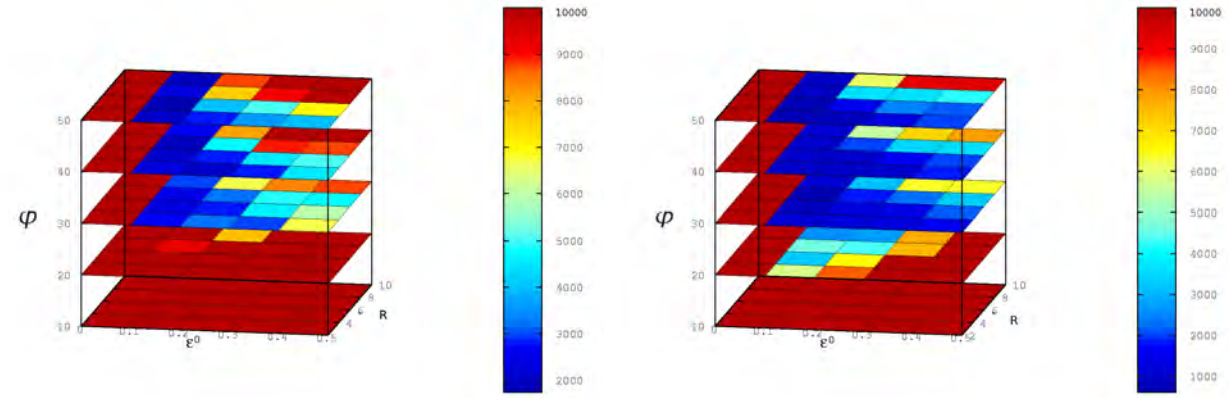
FIG. 11.11 – Évolution de la dynamique interne d'un prédateur capturant la proie — 10000 pas de temps sont portés sur le graphique.

proie qui devient l'attracteur de la trajectoire de l'agent. Mais cet attracteur est dynamique et

se déplace dans l'environnement tant qu'il n'y a pas suffisamment d'agents autour de la proie. Cela explique qu'une capture puisse échouer.

Attracteurs du système. Une vision de la dynamique indépendamment du temps est donnée par la reconstruction 3D de la dynamique des prédateurs sur le graphique 11.11(b). Ce graphique est construit selon la méthode des retards [Takens, 1981] où on affiche les valeurs retardées de la série temporelle ($X_1 = x^t, X_2 = x^{t+1}, X_3 = x^{t+2}$) sur les trois dimensions. Selon cette méthode et le théorème qui en est à l'origine, l'attracteur d'une série temporelle à 1 dimension peut être reconstruit dans un espace de dimension 3. On y voit la dynamique sans détection de proie qui forme un nuage autour d'une parabole déformée dans l'espace 3D, et d'un nuage plus dense de points qui relie cette parabole au point $(0, 0, 0)$. Pour finir la proie et sa zone de marquage de présence constituent le bassin d'attraction pour la dynamique interne des prédateurs. La proie représente un attracteur de type point fixe pour les prédateurs dans cette dynamique.

Influence des paramètres. Pour mesurer l'impact des divers paramètres du modèle ainsi réalisé, une série de simulations a été effectué dont les résultats sont compilés sur le graphique 11.12 qui représente le temps de capture moyen sur 5 simulations de même ensemble de paramètres en fonction de l'angle incrémental φ de 10° à 50° degrés, du rayon de voisinage $V^{\mathcal{X}}$ pour la perception des autres prédateurs de 2 à 10, et du couplage ϵ^0 de 0 à $0,5$ (au-delà de cette valeur de couplage, la synchronisation chaotique nuit à l'exploration). Si la capture n'a pas eu lieu après 10000 pas de temps, la simulation est arrêtée et cette valeur limite est affectée à la simulation.



(a) Simulations comprenant 20 prédateurs et une proie. (b) Simulations comprenant 50 prédateurs et une proie.

Fig. 11.12 – Résultats des temps de capture pour deux séries de simulations pour deux groupes d'agents. Le temps de capture est mesuré par couleur et résulte d'une moyenne des résultats sur 5 simualtions de même paramètres.

Ces résultats montrent que :

– Les performances augmentent avec le nombre de prédateurs, mais nous n'avons pu établir de loi générale liant cette performance au nombre de prédateurs.

- Les meilleures configurations donnent des temps de capture inférieurs à 1000 pas de temps en moyenne, la couverture de l'espace par les agents est efficace sans que l'on puisse réellement distinguer quel paramètre influe le plus sur ce résultat.
- Le couplage entre prédateurs ne doit pas être ni trop important ni complètement absent (la valeur nulle ne donne aucune capture), ici $\epsilon^0 = 0,1$ semble une bonne valeur dans toutes les configurations. En effet, paradoxalement plus le couplage est fort moins les agents pourront stopper sur la proie, étant en permanence perturbés par les autres agents. En revanche, si un point d'agrégation suffisamment stable (nombreux prédateurs) se forme, un couplage fort permet d'attirer les agents à proximité. Une idée d'amélioration du mécanisme serait de commencer avec un couplage assez faible lors de la phase d'exploration puis d'augmenter ce couplage à mesure qu'un point d'agrégation se forme sur une proie.
- La remarque précédente vaut également pour le voisinage d'interaction avec les autres agents prédateurs.
- Enfin, l'angle incrémental φ joue un rôle important puisque des valeurs inférieures à un seuil (ici 30°) rend difficile la capture. Ce seuil dépend également implicitement de la valeur d'autres paramètres (vitesse maximum des agents par exemple).

Discussion et conclusion.

A travers ce problème proie-prédateurs, on a pu illustrer comment des changements assez simples dans les lois de déplacement et de perception des agents peut modifier le mécanisme d'adaptation pour réaliser des situations qui peuvent évoquer des phénomènes biologiques. Ce qui nous semble important également ici est le recours aux outils de visualisation de la dynamique à travers le diagramme de bifurcation et la reconstruction d'un attracteur 3D, rendu possible parce que la décision interne de l'agent est ici mono-dimensionnelle. Ces outils permettent une meilleure compréhension du lien entre la dynamique intrinsèque aux agents d'une part et la dynamique externe géométrique.

Au-delà de ces aspects, ce problème est en fait assez semblable à des problèmes d'optimisation non linéaires utilisés pour tester les métaheuristiques comme l'optimisation par essaims particulaires présentés en section 2.1.2. Certains de ces problèmes consistent en effet à trouver le zéro d'une fonction non linéaire dans un espace continu. Les particules de l'essaim explorent l'espace jusqu'à se rapprocher au plus près de la solution optimale. Ces problèmes sont difficiles car les fonctions considérées comportent soit de nombreux points sous-optimaux, soit un seul point optimal qui est un pic très fin. Une batterie de ces fonctions tests constitue les fonctions de De Jong [De Jong, 1975]. Elle comporte entre autres un cas très proche de notre problème proie-prédateurs, la fonction d'Easom (cf. figure 11.13) ou encore de la fonction du chemin d'Ackley [Pohlheim, 2006]. Nous abordons donc avec notre problème une premier cas d'optimisation puisque le problème de nos agents prédateurs revient à chercher finalement un 0 dans le champ de contrôle $\mathcal{C}$ de l'environnement. Le mécanisme d'adaptation utilise le couplage mais surtout la modification dynamique de la variable de contrôle a asservie par le champ. Cette technique est utilisée plus profondément dans notre vision des fourmis logistiques, objet du prochain chapitre.

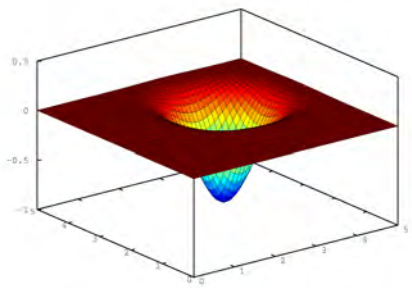


FIG. 11.13 – Fonction d'Easom : $f_{Easom}(x_1, x_2) = -\cos(x_1)\ \cos(x_2)\ e^{-((x_1-\pi)^2+(x_2-\pi)^2)}$

11.5 Synthèse

Nous venons de montrer quelqu'unes des pistes d'utilisation du SMAL en mode adaptatif, c'est-à-dire lorsque les agents sont sensibles aux données exogènes présentes dans l'environnement indépendamment des données endogènes produites par les autres agents. La perception de ces informations joue sur la valeur des variables internes de contrôle ou de couplage des agents et indirectement sur leur variable interne de décision. De nouveaux schémas apparaissent alors spontanément : la formation de files d'agents contrôlées par le champ de l'environnement et l'agrégation d'agents en un point de l'espace géométrique (proie-prédateurs) se produisent par adaptation de la variable de contrôle à un champ de contrôle, tandis que la formation de groupes d'agents en flocking se réalise par adaptation de la variable de couplage à un champ de couplage. Dans ce dernier cas, le portage de l'algorithme dans un essaim de robots matériels s'est avéré prometteur sur la capacité d'étendre le dispositif à des essaims de taille plus importante.

Les deux aspects de couplage et de contrôle sont les clés des mécanismes d'adaptation avec le SMAL mais ils ne sont pas utilisés conjointement dans les simulations présentées. Il s'agit pour nous de saisir d'abord séparément la complexité des deux processus d'adaptation avant de les réunir dans une seule et même instance du SMAL.

Enfin, l'aspect métaheuristique pour l'optimisation a été simplement suggéré à travers la simulation d'un problème proie-prédateurs que l'on a pu relier à un problème test typique de l'optimisation par essaim particulaire. Cependant ce cas na pas été traité à partir d'un véritable problème d'optimisation et n'indique pas vraiment si le SMAL peut servir de métaheuristique pour l'intelligence en essaim. Pour en avoir une idée plus pragmatique, le chapitre suivant expose le modèle que nous avons appelé "fourmi logistique" et qui vise à proposer une métaheuristique pour l'optimisation combinatoire, à l'instar des algorithmes fourmis décrits en section 2.1.3.

12

Mécanismes adaptatifs des fourmis logistiques

L'étude originale de l'expérience fondatrice du double pont a été rappelée en section 1.3.3. Cette expérience montre qu'une colonie de fourmis dans le cadre du fourragement brise spontanément la symétrie du double pont d'égale distance reliant le nid à une source de nourriture. Le modèle probabiliste élaboré par Deneubourg [Deneubourg *et al.*, 1990] a donné la base pour la conception des algorithmes fourmis, mais a également inspiré beaucoup d'autres modélisations et simulations de phénomènes biologiques, où l'on retrouve la même de loi de probabilité pour la décision des agents (spécialisation des rôles, émergence de structures hiérarchiques, tri de couvains, ...). Ces phénomènes sont emblématiques de l'intelligence en essaim et révèlent des mécanismes génériques. Si le SMAL est approprié pour traiter de l'intelligence en essaim comme nous le soutenons, il doit être en mesure de répondre également à la problématique fourmi, c'est-à-dire de pouvoir intégrer un mécanisme de type stigmergique utilisant des phéromones numériques.

Ce chapitre a pour objectif de montrer comment le SMAL peut traiter de certains problèmes fourmis. L'autre versant de cet objectif est de montrer comment une fonction objectif à optimiser, caractéristique du problème considéré, est insérée dans le modèle et quels mécanismes d'adaption sont mis en oeuvre.

Une première section est consacrée à présenter de façon générale le modèle des fourmis logistiques qui servira de base aux sections suivantes. Nous reconsidérons ensuite le problème du double pont avec des branches d'égale distance pour simuler le phénomène au moyen de notre modèle, car c'est le phénomène de base de l'auto-organisation fourmi. Puis une extension à la prise de décision sur un noeud du graphe comportant plusieurs arêtes est envisagé comme première étape vers l'optimisation dans les graphes. Enfin nous proposons une métaheuristique capable d'approcher l'optimisation d'un problème tel que le problème du voyageur de commerce, vu comme problème test de notre algorithme. Nous désignons cette métaheuristique sous le terme de métaheuristique ou algorithme des fourmis logistiques.

12.1 Modélisation des fourmis logistiques

Définition générale de l'environnement.

Environnement. La base de l'espace du problème que l'on va considérer dans les deux sections suivantes est un graphe $G = (S, A)$ où S désigne l'ensemble des sommets ou nœuds du graphe et A désigne les arcs du graphe. On repère les nœuds par un indice $k = 1, \cdots, n$ où n est le nombre total de nœuds. Un arc est représenté par un doublet d'indices (k, l) relatifs aux deux nœuds qui constituent l'arc. Le fait que le graphe soit orienté ou non, détermine l'importance de l'ordre des indices.

L'espace géométrique E de l'environnement sur lequel on va raisonner est en fait un espace discret de dimension 2 correspondant au produit cartésien $\mathbb{N}_n^* \times \mathbb{N}_n^*$, où $\mathbb{N}_n^* = \{1, \cdots, n\}$.

Champs de l'environnement.

Champ d'adjacence. Dans ces conditions, on peut définir le champ d'adjacence équivalent à la matrice d'adjacence du graphe, selon la définition suivante :

Définition 12.1.1. *Le champ d'adjacence est défini sur E à valeurs dans $\{0, 1\}$:*

$$\begin{aligned} \mathcal{A} : \quad & E \to \{0, 1\} \\ & (k, l) \mapsto \mathcal{A}(k, l) \end{aligned} \tag{12.1}$$

Deux nœuds k et l sont adjacents si les conditions équivalentes suivantes sont vérifiées :
- *il existe un arc (k, l) reliant les deux nœuds,*
- *le champ d'adjacence $\mathcal{A}$ vérifie : $\mathcal{A}(k, l) = 1$*

Si les nœuds k et l ne sont pas connectés, alors les conditions suivantes sont vérifiées :
- *il n'existe aucun arc reliant les deux nœuds,*
- *le champ d'adjacence $\mathcal{A}$ vérifie : $\mathcal{A}(k, l) = 0$*

Ce champ est persistant et *a priori* stationnaire.

Champ des poids de connexion. Si les arcs du graphe sont affectées de poids de connexion $w(k, l)$, alors :

Définition 12.1.2. *Le champ des poids de connexion noté $\mathcal{W}$ est défini sur E à valeurs dans les réels de façon générale :*

$$\begin{aligned} \mathcal{W} : \quad & E \to \mathbb{R} \\ & (k, l) \mapsto \mathcal{W}(k, l) = w(k, l) \end{aligned} \tag{12.2}$$

Ce champ stocke par exemple la distance d'une ville à l'autre dans le problème du voyageur de commerce. Ce champ est également persistant, *a priori* stationnaire et exogène par définition.

Champ de phéromone. On définit également le champ de phéromones ainsi :

Définition 12.1.3. *Le champ de phéromone $\mathcal{T}$ est défini sur E à valeurs dans les réels positifs :*

$$\begin{aligned} \mathcal{T} : \quad & E \to \mathbb{R}+ \\ & (k,l) \mapsto \tau(k,l) \end{aligned} \tag{12.3}$$

Les champs que nous définissons ici ont davantage un sens mathématique qu'un sens physique, même si certains auteurs ont pu assimiler le champ de phéromones à un champ physique de potentiel (cf. par exemple [Parunak, 1997]), hypothèse que nous ne partageons pas avec le SMAL.

D'autre part, le champ de phéromone étant modifié par les agents, on définit également un champ d'influence associé $\widetilde{\mathcal{T}}$ pour une mise à jour synchrone du champ. Contrairement au champ de phéromone, le champ d'influence est non persistant d'un pas de temps à l'autre ; il n'est utile que lors de la mise à jour. Ces deux champs sont initialisés à 0. Le champ de phéromone est volatil avec un coefficient d'évaporation ρ.

Champ de décision. Le champ de décision $\mathcal{X}$ sensiblement identique aux versions précédentes permet de stocker les valeurs calculées $f(x_i, a_i)$ éventuellement cumulées de la variable interne de décision des agents, dans le cas où une perception selon x est définie. Ce champ stocke ces valeurs uniquement sur les nœuds du graphe, c'est-a-dire aux points (k,k) de E.

Définition 12.1.4. *Le champ de décision $\mathcal{X}$ est défini par :*

$$\begin{aligned} \mathcal{X} : \quad & E \to [0,1] \\ & (k,k) \mapsto \mathcal{X}(k) \end{aligned}$$

De même, les agents ayant à déposer de l'information dans ce champ, un champ d'influence non persistant lui sera associé avec initialisation nulle.

Champ de position de l'agent. Le champ de position de l'agent est dans sa définition identique au champ d'adjacence du graphe, si ce n'est que les positions d'un agent sont restreintes aux nœuds du graphe :

Définition 12.1.5. *Le champ de position δ_i de l'agent A_i est défini par :*

$$\begin{aligned} \delta_i : \quad & E \to \{0,1\} \\ & (k,k) \mapsto \delta_i(k) \end{aligned}$$

où 0 indique l'absence d'agent et 1 sa présence en un nœud du graphe (cette restriction au nœud est arbitraire car le champ pourrait considérer des positions sur les arcs du graphe).

La fourmi logistique.

État interne. La fourmi logistique est un agent logistique adaptatif qui possède un état interne $s(t) = \langle x_i^t, a_i^t, \epsilon^0 \rangle$. Ici la variable de contrôle a_i^t dépend au cours du temps du champ de phéromone $\mathcal{T}^t$, qui joue donc le rôle d'un champ de contrôle.

Perceptions. Les perceptions de la fourmi logistique s'opèrent selon les composantes x et a de l'état interne. Ces perceptions vont dépendre des champs $\mathcal{X}$ et $\mathcal{T}$ pour l'agent i :

$$\begin{cases} P_{i,x}(\sigma^t) = P_{i,x}(\mathcal{X}^t) \\ P_{i,a}(\sigma^t) = max_{V_i}\{\mathcal{T}^t\} \end{cases} \tag{12.4}$$

La formulation précise de ces lois dépend du contexte d'application. La seconde équation de modification récupère le maximum de la quantité de phéromone perçue sur un voisinage V_i contenant les arcs liées au nœud où se trouve l'agent. En général ce voisinage est à horizon 1 et ne considère que les arcs des nœuds directement adjacents au nœud courant.

Évolution de l'état interne et calcul de la décision. L'évolution suit les lois suivantes pour la fourmi logistique i :

$$\begin{cases} \epsilon_i^{t+1} = \epsilon^0 \\ a_i^{t+1} = F_a(P_{i,a}(\mathcal{T}^t)) \\ x_i^{t+1} = (1-\epsilon^0)f(x_i^t, a_i^{t+1}) + \epsilon^0\ P_{i,x}(\mathcal{X}^t) \end{cases}$$

Comme on le constate, la mise à jour de a se fait à travers une fonction F_a qui est à préciser en fonction du problème. L'application logistique utilisée dans tout ce chapitre est de nouveau une forme conjuguée de l'application standard dont le diagramme de bifurcation est donnée sur la figure 12.1 :

$$f(x,a) = 1 - a(2\ x - 1)^2 \tag{12.5}$$

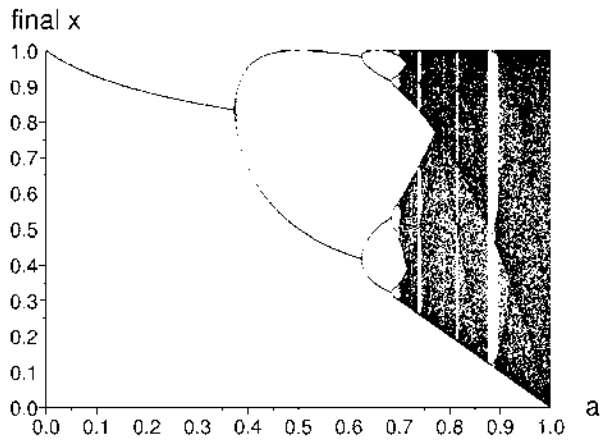


FIG. 12.1 – Diagramme de bifurcation de l'application $x^{t+1} = 1 - a(2\ x^t - 1)^2$

Actions de la fourmi logistique. Les actions de la fourmi logistique sont de deux ordres :

- Effectuer un déplacement vers un nœud adjacent au nœud courant du graphe en fonction de la valeur de la variable de décision mise à jour. La variable $2\pi\ x^{t+1}$ est alors considérée comme un angle sur le cercle qui pointe sur un secteur correspondant à la k-ième quantité de phéromone sur les arcs triés par ordre décroissant de phéromone. Cette action résulte d'un tirage de type "roue de la fortune". Ce mécanisme sera précisé notamment dans l'application au TSP en section 12.3.
- Effectuer un dépôt de phéromone sur l'arc choisi (k^*, l^*) à l'étape précédente en phase de déplacement, ainsi que sur le nœud courant. Cet ajout de phéromone est égal à la valeur de la variable de décision x et se fait de façon cumulative sur les agents ayant emprunté la même voie :

$$\begin{cases} \widetilde{\mathcal{T}}(k^*, l^*)^{t+1} = \widetilde{\mathcal{T}}(k^*, l^*)^{t+1} + x_i^{t+1} \\ \widetilde{\mathcal{X}}(k^*)^{t+1} = \widetilde{\mathcal{X}}(k^*)^{t+1} + x_i^{t+1} \end{cases} \tag{12.6}$$

Réactions de l'environnement.

L'environnement met à jour les champs selon les équations :

$$\begin{cases} \mathcal{T}(k, l)^{t+1} = (1 - \rho)\ \mathcal{T}(k, l)^t + \widetilde{\mathcal{T}}(k, l)^{t+1} \\ \mathcal{X}(k, l)^{t+1} = \mathcal{X}(k, l)^{t+1} + \widetilde{\mathcal{X}}(k)^{t+1} \end{cases} \tag{12.7}$$

Le champ de phéromone subit une évaporation exprimée par le facteur d'évaporation $(1 - \rho)$.

Discussion.

Principe du mécanisme d'exploration-exploitation pour l'optimisation. Le choix de la fonction F_a du système 12.5 est crucial car il constitue le rouage principal du mécanisme de réalisation d'une optimisation. En effet, l'optimisation va consister à conduire l'ensemble de la colonie de fourmis logistiques vers un point de fonctionnement qui est un point fixe de leur dynamique interne, en l'occurrence le point fixe $(0, 1)$ du diagramme 12.1. On a déjà abordé ce type de mécanisme d'adaptation dans le problème proie-prédateur par exemple en section 11.4.C'est pourquoi la fonction F_a est choisie comme une fonction monotone décroissante de la perception $P_{i,a}$. Ainsi, les fourmis logistiques partent d'un état où $a = 1$ en phase d'exploration vers un état où $a = 0$ qui doit coïncider avec la solution proposée par le système multi-agent au problème d'optimisation. L'application concrète de cette procédure est détaillée lors du traitement du problème du voyageur de commerce en section 12.3.

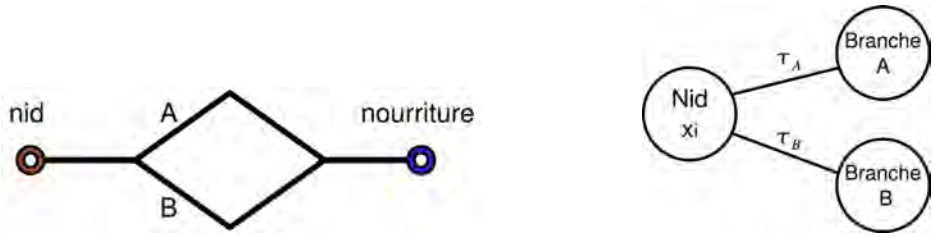
Interprétation de la péromone pour la fourmi logistique. La mise à jour du champ de phéromone par la fourmi logistique met en jeu sa variable décisionnelle interne. Cela implique dans notre conception une interprétation explicite de la phéromone numérique comme le cumul des variables de décision interne des divers agents, et d'une certaine façon comme la décision collective de l'essaim. De ce fait, l'aspect décisionnel de la phéromone apparaît clairement dans cette mise en jeu des variables décisionnelles individuelles. Le champ de phéromone est donc composé de données endogènes car les données manipulées par les fourmis sont issues de leur propre variable de décision interne. Les mécanismes impliqués alors sont de fait auto-organisationnels.

Bilan. Cette description générale de l'algorithme des fourmis logistiques comporte quelques éléments à spécifier davantage dans les applications que nous abordons dans les sections suivantes, d'abord en reprenant le cas d'étude du double pont en configuration d'égale distance pour marquer l'aspect auto-organisationnel du phénomène, puis dans le cadre de l'optimisation du TSP où l'adaptation joue un rôle prépondérant.

12.2 L'expérience du double pont revisitée

Fourmis logistiques pour le double pont.

Les conditions du protocole expérimental d'origine est détaillé en section 1.3.3. La figure 12.2(b) montre la modélisation simplifiée du problème à l'aide d'un graphe à trois noeuds reliés par deux arêtes qui représentent les deux branches accessibles. On indicera ici par Nid, A et B les trois noeuds considérés. Le champ de phéromones τ est défini sur chaque arête du graphe. Les hypothèses de modélisation pour simuler le double pont sont celles-ci :



(a) Protocole expérimental. (b) Un graphe à trois noeuds

FIG. 12.2 – Expérience du double pont et sa modélisation sous forme de graphe.

- les fourmis traversent une à une le pont (le processus est séquentiel) ; chaque passage d'une fourmi correspond à un pas de temps de l'algorithme,
- la quantité de phéromone déposée par chaque fourmi est égale à sa variable de décision interne x,
- le champ de phéromone ne s'évapore pas.

Nous avons donc un environnement à $n = 3$ noeuds. La perception des fourmis s'écrit simplement dans le cas du double pont par :

$$\begin{cases} P_x^t = \mathcal{X}^t = x_{i-1}^t \text{ décision de la fourmi précédente} \\ P_a^t = max\left\{\tau_a, \tau_b\right\} \end{cases} \tag{12.8}$$

Ce qui conduit au système de transition de l'état interne :

$$\begin{cases} a_i^{t+1} = \frac{\alpha}{\alpha + P_a^t} \\ x_i^{t+1} = (1-\epsilon^0) f_a^m(x_i^t) + \epsilon^0 x_{i-1}^t \end{cases} \tag{12.9}$$

où $\alpha > 0$ est un paramètre de modulation de la loi d'évolution de a.

Ici, une différence est apportée à l'algorithme général dans le calcul de l'application logistique puisque l'on calcule m itérées de l'application plutôt qu'une seule. La situation séquentielle et non

parallèle de l'algorithme rend cet ajout nécessaire afin de garantir la propriété pseudo-aléatoire de l'application au départ de l'algorithme [Charrier *et al.*, 2007c].

Résultats de simulation et interprétation.

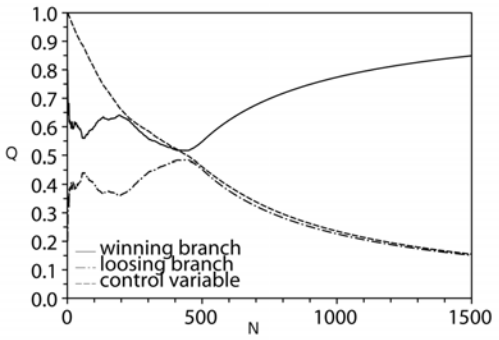
Algorithme. L'algorithme 12.1 résume l'ensemble du processus. Les résultats de passage

Algorithm 12.1: Algorithme de la fourmi logistique pour l'expérience du double pont

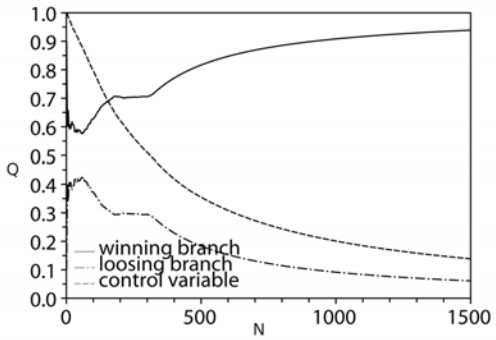
Initialisation
$\tau_A, \tau_B \leftarrow 0$
$N_A, N_B \leftarrow 0$
for *ant* $i = 1$ *to* N **do**
  init : $x^0 \leftarrow random(0,1)$
  Perception
  read $\tau_{max} \leftarrow max\{\tau_A, \tau_B\}$
  $index_{max} \leftarrow argmax_{\{A,B\}}\{\tau_A, \tau_B\}$
  $index_{min} \leftarrow argmin_{\{A,B\}}\{\tau_A, \tau_B\}$
  $P_x \leftarrow x_{i-1}$ si $i > 1$ **Décision**
  $threshold \leftarrow \frac{1}{2}$
  $a \leftarrow \alpha / (\alpha + \tau_{max})$
  $x_i \leftarrow (1 - \epsilon_0)\ f^m(x^0, a) + \epsilon_0 P_x$
  Action
  if $x_i < threshold$ **then**
    Action 1
    $\mathcal{N}(Nid, index_{max}) \leftarrow \mathcal{N}(Nid, index_{max}) + 1$
    $\tau_{index_{max}} \leftarrow \tau_{index_{max}} + x_i$
  else
    Action 2
    $\mathcal{N}(Nid, index_{min}) \leftarrow \mathcal{N}(Nid, index_{min}) + 1$
    $\tau_{index_{min}} \leftarrow \tau_{index_{min}} + x_i$
  $\mathcal{X}(nid) \leftarrow x_i$

des fourmis sont compilés dans un champ de positions cumulées $\mathcal{N}$ persistant. La valeur du paramètre α est $\alpha = 100$.

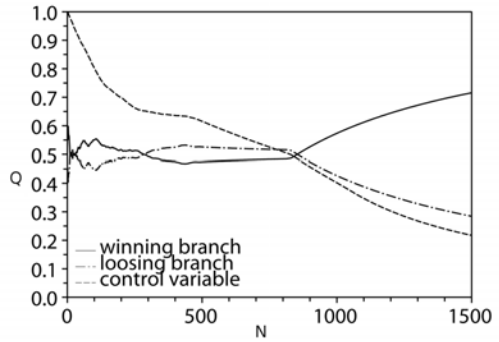
Résultats. Les simulations effectuées ont mis en jeu 1500 fourmis logistiques, ce qui correspond à autant de pas de temps de simulation. On affiche sur les graphiques 12.3(a), 12.3(b) et 12.3(c) le taux $Q = \frac{N_A}{N}$ du nombre de fourmis logistiques passant par la branche A en fonction du nombre total de fourmis déjà passées N (en abscisse), de même pour la branche B. Différentes configurations sont établies, avec couplage moitié ou sans couplage avec la fourmi précédente $\epsilon^0 = 0,5$ ou $\epsilon^0 = 0$. De plus, les simulations révélent une sensibilité au nombre d'itérations m effectuées pour caluler la valeur de décision résultant de l'application logistique. Pour un couplage nul cette sensibilité n'est pas vraiment visible. C'est seulement pour un couplage $\epsilon^0 = 0,5$ que



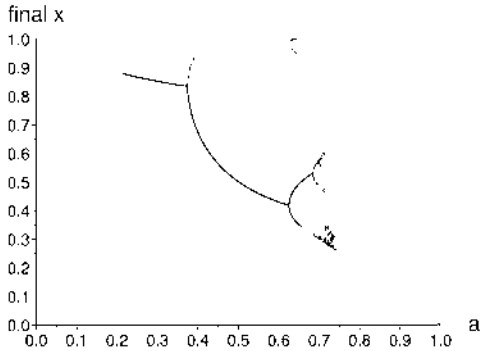
(a) Cas où $\epsilon^0 = 0$ et $m = 500$



(b) Cas où $\epsilon^0 = 0.5$ et $m = 499$



(c) Cas où $\epsilon^0 = 0.5$ et $m = 500$



(d) Diagramme de bifurcations reconstruit sur les données de simulation avec $N = 1500$ fourmis logistiques , $\epsilon^0 = 0.5$ and $m = 500$

FIG. 12.3 – Résultats de simulation du double pont comportant 1500 fourmis logistiques. Le résultat affiché est une moyenne sur 10 simualtions. Q désigne le rapport du nombre de fourmis ayant emprunté une branche sur le total N de fourmis ayant traversé au temps t.

l'on voit apparaître cette sensibilité. On remarque que la parité du nombre d'itérations joue un rôle qu'il n'est pas évident d'interpréter *a priori*.

Interprétation.

Un premier résultat. Les diagrammes des données de simulation précédents affichent également l'évolution des valeurs de la variable de contrôle a^t au cours du temps. Comme la propriété monotone décroissante de la loi d'évolution (12.9) le laisse prévoir, la valeur a^t décroît avec le temps de simulation. On peut établir la proposition suivante :

Proposition 12.2.1. *Sachant que le seuil de décision est ici $x^* = \frac{1}{2}$, la branche gagnante est déterminée par la branche possédant le maximum de phéromone lorsque $a^t \leqslant \frac{1}{2}$.*

Preuve. Le diagramme de bifurcation 12.1 montre que le point correspondant à $a = \frac{1}{2}$ est le point 2-périodique $\bar{x} = (\frac{1}{2}, 1)$. Pour s'en convaincre il suffit de vérifier que $f(\frac{1}{2}, \frac{1}{2}) = (1, \frac{1}{2})$ et que $f(1, \frac{1}{2}) = (\frac{1}{2}, \frac{1}{2})$. En deçà du seuil, c'est-à-dire pour $a < \frac{1}{2}$ les valeurs des points fixes doubles ou simples sont tous supérieurs au seuil de décision $x^* = \frac{1}{2}$. A partir de ce moment, toutes les fourmis logistiques fixent leur choix sur la même branche du pont, à savoir celle qui possède le maximum de phéromone.

Reste à vérifier que la fonction a^t est strictement décroissante en fonction du temps. La perception P_a^t est une fonction croissante puisqu'elle récupère le maximum de la phéromone perçue sur le champ $\mathcal{T}$, qui est un champ cumulatif de valeurs positives sans évaporation. La forme de la loi de transition F_a du système(12.9) concernant a est une fonction inversement proportionnelle à P_a^t vérifiant :

$$F_a^{t+1} - F_a^t = -\alpha \, \frac{P_a^{t+1} - P_a^t}{(\alpha + P_a^{t+1}) \, (\alpha + P_a^t)} \tag{12.10}$$

Comme $\alpha > 0$, on en déduit que F_a^t et P_a^t évoluent en sens contraire. On a effectivement $P_a^{t+1} - P_a^t \geqslant 0$ mais on doit vérifier que $P_a^{t+1} - P_a^t > 0$ à partir d'un certain temps.

Pour cela observons que $P_a^{t+1} = P_a^t$ si et seulement si $x_i^{t+1} = 0$. Cherchons les antécédents de 0 par f (cf. (12.5)) :

$$f^{-1}(0) = \begin{cases} \dfrac{1 + \frac{1}{\sqrt{a}}}{2} \\ \text{ou } \dfrac{1 - \frac{1}{\sqrt{a}}}{2} \end{cases} \tag{12.11}$$

Or $a \in [0, 1]$ donc $\frac{1}{\sqrt{a}} \geqslant 1$. Des deux antécédents possibles, seule la première forme peut donner un point lorsque $a = 1$, le point $x = 1$. De même la pré-image de 1 par f est $\frac{1}{2}$. Le point $x = 0$ est un point fixe instable lorsque $a = 1$ or l'ensemble des pré-images de 0 par f dans $[0, 1]$ est un ensemble de mesure nulle sinon la fonction f ne serait pas ergodique pour $a = 1$ (cf. section 5.3) ; autrement dit la probabilité d'aboutir sur 0 en partant d'un point tiré aléatoirement dans $[0, 1]$ est nulle. On en conclue donc que P_a est presque surement strictement décroissante.

On en déduit également que l'évolution de a est strictement décroissante. Comme a^t est également borné dans $[0, 1]$, cela implique :

$\lim_{t\to\infty} a^t = 0$.
Pour finir, il existe donc un temps $0 < t^* < \infty$ à partir duquel $a^{t^*} < \frac{1}{2}$, ce qui conclue la proposition. □

Une autre proposition découle directement de la précédente :

Proposition 12.2.2. *Les variables internes (a, x) des fourmis logistiques convergent vers le point $(0, 1)$ du diagramme de bifurcation de f lorsque le nombre de fourmis $N \to \infty$.*

Preuve. Le fait que $\lim_{t\to\infty} a^t = 0$ implique que $\lim_{i\to\infty} x_i \to 1$ par continuité de f :
soit $\beta > 0$, il existe t^* tel que $t > t^* \Rightarrow |a^t| < \beta$, cela implique dans l'expression de f (cf. formule (12.5)) l'inégalité suivante, sachant que $0 \leqslant x^t \leqslant 1$:

$$\begin{aligned} &1 - \beta < f(a^t, x^t) \leqslant 1 \\ &\text{soit } 0 \leqslant |f(a^t, x^t) - 1| < \beta \end{aligned}$$

ce qui démontre la limite de f vers 1 lorsque $t \to \infty$, ce qui revient à dire que $f \to 1$ lorsque le nombre de fourmis $N \to \infty$. Ceci conclue la proposition énoncée. □

Cette proposition est importante pour la suite des applications des fourmis logistiques, car le principe qu'elle expose est au coeur de la résolution des problèmes d'optimisation dans notre approche.

Effet de la parité du nombre d'itérées. Un deuxième constat concerne la parité du nombre d'itérées m sur la rapidité de décroissance de a^t. On remarque qu'un nombre d'itérées pair retarde la convergence sur une des branches pour un couplage fort, sans toutefois l'empêcher. Cela signifie que le système explore les deux branches plus longtemps. Ce point est moins clair cependant que le précédent et nous n'évoquerons que des hypothèses à ce sujet soutenues par le graphique 12.3(d). En effet ce diagramme représente les points suivis par le système des fourmis logistiques au cours de l'algorithme lorsque le nombre d'itérées est pair. On observe une seule des branches de bifurcation empruntée lorsque $a < 0,7$. Raisonnons sur la partie où l'application logistique comporte une zone de points 2-périodiques stables sur deux branches de bifurcation, à savoir pour $a < 0,6$. Si le système commence à emprunter la branche la plus basse, étant donné le nombre pair d'itérations effectuées, le point fait des allers-retours entre les deux branches en revenant systématiquement sur la branche de départ, la branche basse donc. Cela n'est évidemment plus vrai lorsque le nombre d'itérées est impair. Dans ce cas l'alternance entre les deux branches de bifurcation a lieu. Cette stabilité relative de la branche basse a pour conséquence de prolonger l'indécision quant à la branche de pont à choisir pour les fourmis logistiques, puisque le seuil de décision est $\frac{1}{2}$. Le fait que ce phénomène se produise pour un couplage fort est évident, puisque le couplage renforce la mémoire sur l'état de la fourmi logistique précédente. L'explication est qualitative mais montre que le choix d'un nombre pair d'itérées à des conséquences non négligeables sur l'exploration du graphe.

Cas d'un noeud connecté à plus de deux autres noeuds.

Ce cas nous intéresse puisqu'il trace la voie vers les graphes quelconques. La figure 12.4 présente une simulation réalisée selon le même principe algorithmique avec 6 branches de pont d'égale distance. Ici encore, la symétrie est brisée et seule une branche est finalement sélectionnée.

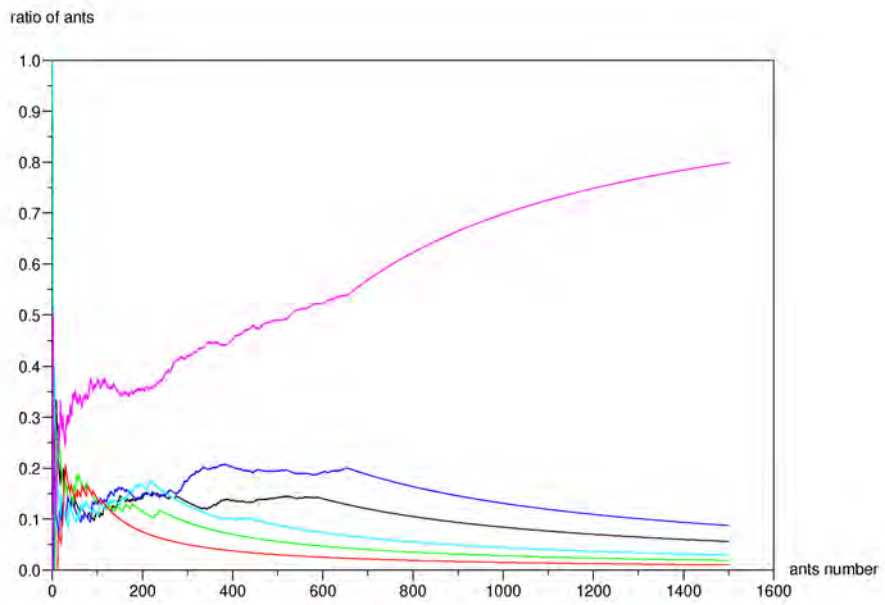


FIG. 12.4 – Rapport de nombres d'agents passés par un arc donné du graphe sur le nombre total qui a traversé ; avec $N = 1500$ fourmis logistiques, 6 branches, $\epsilon = 0$ et $m = 499$

Bilan.

Nous venons de proposer un algorithme des fourmis logistiques permettant de simuler la brisure de symétrie du problème du double pont et de reproduire la forme des courbes produites expérimentalement. Notre modèle s'appuie sur une sémantique claire et des mécanismes explicites visualisables par des diagrammes qui en facilitent la compréhension et le suivi. C'est la force de notre approche de pouvoir disposer d'outils d'analyse et éventuellement de preuve des mécanismes impliqués. La reproduction des courbes expérimentales du double pont n'est certes pas suffisante pour affirmer que l'on peut simuler une intelligence en essaim de type fourmi avec le SMAL. C'est pourquoi notre évaluation du modèle se tourne vers sa capacité à traiter de problèmes d'optimisation tels que le voyageur de commerce au même titre que l'algorithme fourmis vu en section 2.1.3.

Les prémisses nécessaires à la conception d'un algorithme des fourmis logistiques pour le TSP ont été posées à travers cette simulation du double pont. Ce cas a montré la part auto-

organisationnelle des mécanismes des fourmis logistiques en explicitant l'effet stigmergique de la phéromone comme donnée de nature endogène au système. Lorsque des données exogènes sont impliquées dans le problème, comme le poids des noeuds dans le graphe, on introduit alors un mécanisme adaptatif qui répond à une certaine fonction objectif à optimiser, comme nous allons le voir à présent sur un problème précis de TSP.

12.3 Application des fourmis logistiques au problème du TSP

12.3.1 Algorithme des fourmis logistiques pour le TSP

Introduction.

Notre intention concernant l'optimisation des problèmes TSP n'est pas de rivaliser avec toute la gamme des algorithmes déjà existants, mais de montrer les performances de notre proposition d'algorithme à base de fourmis logistiques et d'en situer le potentiel comparativement à l'algorithme le plus proche en termes de conception, à savoir la version "Ant System" (AS) des algorithmes fourmis de Dorigo décrite en section 2.1.3. Nous n'en sommes qu'au début de l'étude sur ce nouvel algorithme et notre ambition à ce niveau est mesurée : elle se limite à vérifier notre hypothèse de départ dans cette thèse concernant la capacité du SMAL à traiter des problèmes classiquement abordés par les algorithmes de l'intelligence en essaim aujourd'hui.

Dans cette optique, nous utilisons un des problèmes de la base de référence TSPLIB [TSP, 1995] pour le TSP : le problème "att48" à 48 villes. C'est un problème de type TSP symétrique, c'est-à-dire sur un graphe non orienté (la distance entre deux villes est la même dans un sens ou dans l'autre).

Description de l'algorithme.

L'algorithme est semblable à l'algorithme AS dans son séquencement global, à savoir que la mise à jour des phéromones s'opère de façon synchrone au bout de n pas de temps, laps de temps nécessaire pour terminer un cycle hamiltonien du graphe, qui comporte n nœuds. Ce que l'on va appeler un pas de temps dans la suite de ce descriptif correspond en réalité à N pas de temps d'unités algorithmiques[52], pendant lesquels les N fourmis se sont déplacées d'un nœud sur le graphe.

Principe. L'algorithme procède globalement de la même façon que dans le cas du double pont. La différence tient au fait que deux processus fusionnent ici : d'un côté la propension de la colonie à choisir collectivement un seul arc pour chaque nœud, de l'autre l'orientation vers la minimisation d'une fonction qui calcule la distance sur le chemin parcouru.

Plus précisément, le mécanisme d'adaptation mis en œuvre ici consiste à faire progressivement décroître la variable de contrôle a des agents via une fonction sigmoïde prenant en compte la

[52] L'unité algorithmique est le temps nécessaire pour réaliser le processus sensorimoteur perception-décision-action d'une fourmi.

perception de l'agent sur l'environnement. La fonction sigmoïde gère l'équilibre entre phases d'exploration et d'exploitation. Les opérateurs de perception et d'action intègrent le principe d'optimisation : la perception sélectionne l'information utile à l'optimisation, ici le maximum de la quantité de phéromone sur un arc, tandis que l'action, à travers la mise à jour de la phéromone, stocke et structure l'historique des décisions collectives pour les diffuser à tout le groupe d'agents de façon indirecte.

Initialisation des données du problème. Nous introduisons tout d'abord quelques modifications dans les données de base du problème.

Génération du graphe et champ des distances. Le graphe généré est un graphe complet symétrique pour lequel on calcule les distances de tous les arcs avant même de lancer les fourmis. Le nombre de nœuds est n, on a donc $\frac{n(n-1)}{2}$ distances à calculer. On définit pour stocker ces informations un champ des distances $\mathcal{D}$. Soit $d(k,l)$ la distance de l'arc liant le nœud k au nœud l, on définit le champ $\mathcal{D}$ par :

$$\begin{aligned} \mathcal{D} : \quad & E \to \mathbb{R}+ \\ & (k,l) \mapsto d(k,l) \end{aligned} \tag{12.12}$$

Initialisation du champ des poids du graphe. Les poids dans les problèmes TSP sont des nombres entiers en général correspondant aux distances entre villes. Cette notion de distance n'est pas normalisée et constitue un handicap pour notre approche. Là où l'algorithme AS emploie les valeurs de distances entières qui dépendent du problème considéré pour le calcul de la mise à jour des phéromones, nous transformons ces poids en des valeurs normalisées indépendantes du problème. Le champ $\mathcal{W}$ est initialisé de la manière suivante : nous recalculons les poids associés $w(k,l)$ relativement à la distance minimale sur l'ensemble global A des arcs du graphe $G = (S, A)$:

$$w(k,l) = \frac{\min_{(i,j)\in A} d(i,j)}{d(k,l)} \tag{12.13}$$

de cette façon, tous les poids du graphe sont ramenés à des valeurs dans l'intervalle $[0,1]$.

Normalisation des distances des cycles hamiltoniens. De même on calcule la quantité notée G_{min} qui correspond à la somme des distances minimales partant de chaque nœud du graphe :

$$G_{min} = \sum_{k\in S} \min_{l\in S\setminus\{k\}} \{d(k,l)\} \tag{12.14}$$

Cette distance ne correspond pas, sauf exception, à un cycle hamiltonien, mais constitue une borne inférieure des distances minimales de cycles hamiltoniens. Elle va permettre de ramener la distance totale parcourue L_i par la fourmi logistique A_i sur un cycle hamiltonien à une valeur $x_{L,i}$ dans l'intervalle $[0,1]$ par la formule :

$$x_{L,i} = \frac{L_{min}}{L_i} \tag{12.15}$$

Initialisation du champ de phéromones. Les champs de phéromone est initialisé à zéro partout, ce qui garantit dans notre algorithme une exploration maximale au départ.

Initialisation des fourmis logistiques. A chaque nouveau cycle de recherche d'un cycle hamiltonien dans le graphe, les fourmis sont placées aléatoirement sur les nœuds du graphe, leur variable de décision interne est tirée aléatoirement sur $[0, 1]$ et leur variable de contrôle est initialisée à 1. L'état interne de la fourmi logistique ne comporte pas de variable de couplage comme précédemment dans notre étude.

Autres champs de l'environnement. Nous avons besoin de champs supplémentaires à ceux précédemment définis :

- Un champ des nœuds/arcs visités relatif à chacune des fourmis logistiques : il stocke le parcours de chaque fourmi sur un cycle, est persistant sur un cycle. Il est défini pour la fourmi A_i par :

$$\begin{aligned} \mathcal{V}_i : \quad & E \to \{0, 1\} \\ & (k, l) \mapsto \mathcal{V}_i(k, l) = \begin{cases} 1 \text{ si } (k, l) \text{ a été visité par } A_i, \\ 0 \text{ sinon} \end{cases} \end{aligned} \tag{12.16}$$

  Ce champ marque à la fois les nœuds et les arcs. Le calcul de la longueur totale L_i parcourue par une fourmi logistique A_i sur un cycle hamiltonien se calcule simplement au moyen de l'opérateur produit terme à terme $\otimes$ et de l'opérateur de sommation $\sum$ sur l'espace total par :

$$\begin{aligned} L_i &= \sum_E \mathcal{V}_i \otimes \mathcal{D} \\ &= \sum_{(k,l) \in A} \mathcal{V}_i(k, l)\ \mathcal{D}(k, l) \end{aligned} \tag{12.17}$$

- Un champ des arcs triés selon leur quantité de phéromone noté $\mathcal{O}$, persistant sur un cycle, tel que (n étant le nombre total de nœuds) :

$$\begin{aligned} \mathcal{O} : \quad & E \to \mathbb{N}^*_{n-1} = \{1, \cdots, n-1\} \\ & (k, l) \mapsto \mathcal{O}(k, l) = \text{ indice de tri de l'arc } (k, l) \text{ parmi les arcs partant du nœud } k \end{aligned} \tag{12.18}$$

  Ce champ est non symétrique contrairement au champ de phéromone par exemple, car un arc comprend deux nœuds, il est donc trié différemment selon que l'on considère un nœud ou l'autre. Initialement, ce champ est généré par les indices de l'ordre de construction des arcs (processus séquentiel).
- Enfin un champ d'influence $\widetilde{\mathcal{I}}_i$ non persistant et initialement nul à chaque début de cycle de recherche associé à chaque agent, permet le stockage des valeurs de modification pour le champ de phéromone.

Le processus sensorimoteur de la fourmi logistique. Nous détaillons ci-dessous les trois phases perception-décision-action qu'effectue la fourmi logistique A_i pendant un pas de temps. Supposons que le nœud courant soit le nœud k, la liste des nœuds déjà visités, appelée liste taboue, est stockée dans le champ $\mathcal{V}_i$.

Perception de la fourmi A_i. Ici, la perception selon la variable interne x n'est pas utilisée, seule la perception selon la variable de contrôle a est définie (cf. formules (12.4)) :

$$P_{i,a}(\sigma^t) = \mathcal{T}(argmax_{l \in S \setminus \{k \bigcup \mathcal{V}_i\}}\{\mathcal{O}(k,l)\}) \tag{12.19}$$

Elle renvoie donc la quantité de phéromone sur l'arc partant du nœud k non encore visité, d'indice maximal dans le champ de tri $\mathcal{O}$, ce qui revient à sélectionner l'arc non visité avec le maximum de phéromone comme base de décision.

Décision de la fourmi A_i. La mise à jour des variables a_i et x_i de la fourmi logistique se font selon la formule :

$$\begin{cases} a_i^{t+1} = \dfrac{1}{1 + e^{\alpha\ (P_{i,a}(\sigma^t) - \tau_0)}} \\ x_i^{t+1} = f(x_i^t, a_i^{t+1}) \end{cases} \tag{12.20}$$

La transition selon a est une sigmoïde de pente α et centrée en τ_0. Cette fonction permet de

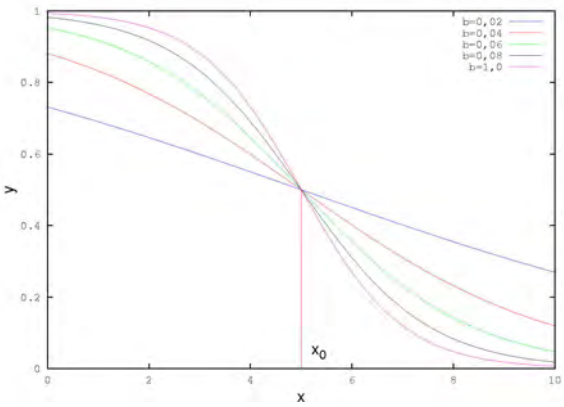


FIG. 12.5 – Tracé de quelques sigmoides d'équation $y = \dfrac{1}{1 + e^{b(x - x_0)}}$ avec $x_0 = 5$.

moduler le caractère plus ou moins accentué de la décroissance des valeurs de a au cours de l'algorithme (cf. figure 12.5). Au départ de l'algorithme, les agents ont des valeurs a proches de 1 (en haut à gauche de la sigmoïde) pour finir vers des valeurs proches de zéro (en bas à droite de la sigmoïde sur la figure).

Action de la fourmi A_i. A partir de sa variable de décision mise à jour x_i^{t+1}, la fourmi procède à plusieurs actions :

– Choisir l'arc des nœuds restant à visiter —supposons que la liste taboue (nœuds déjà visités) comporte n^* nœuds— d'indice $\lceil x_i^{t+1}\ (n - n^*) \rceil$ où l'opérateur $\lceil\rceil$ renvoie la partie

entière arrondie à l'entier supérieur du nombre calculé.
Soit l^* le nœud choisi correspondant. Il s'agit d'un tirage du type roue de la fortune.

- Se déplacer vers le nœud l^*. Le champ δ_i de position est donc modifié et donne 1 sur le nœud l^* et zéro ailleurs.
- Mettre à jour le champ des nœuds et arcs visités en ajoutant le nœud et l'arc choisis au champ :
  $\mathcal{V}_i^{t+1}(k, l^*) = 1$ et $\mathcal{V}_i^{t+1}(k, k) = 1$.
- Mettre à jour le champ d'influence de phéromone non persistant partout à zéro sauf sur l'arc (k, l^*) selon le calcul :

$$\widetilde{\mathcal{T}}_i^{t+1}(k, l^*) = x_i^{t+1}\ \mathcal{W}(k, l^*) \tag{12.21}$$

Réaction de l'environnement. La réaction de l'environnement se déroule selon deux fréquences temporelles : une réaction à chaque pas de temps (i.e. lorsque tous les agents se sont déplacés d'un nœud) et une réaction globale à la fin d'un cycle hamiltonien pour tous les agents où on met en œuvre une politique élitiste (cf. la version "Elitist Ant" des algorithmes fourmis) pour renforcer le meilleur chemin.

Mise à jour du champ de phéromone à chaque pas de temps. A chaque pas de temps la réaction de l'environnement est essentiellement une modification du champ de phéromone et du champ de tri des arcs du graphe :

- Le champ de phéromone est mis à jour selon la formule :

$$\mathcal{T}^{t+1} = \mathcal{T}^t + \sum_{i=1}^{N} \widetilde{\mathcal{T}}_i \tag{12.22}$$

  Une fois cette opération effectuée, les champs $\widetilde{\mathcal{T}}_i$ sont remis à zéro pour l'itération suivante.
- Le champ des arcs triés $\mathcal{O}$ est mis à jour suite aux nouvelles valeurs de phéromones de l'environnement. Ce tri n'est pas coûteux puisqu'il ne concerne que les arcs modifiés en quantité de phéromone. Or au plus N arcs voient leur phéromone modifiée par pas de temps, puisqu'il y a N fourmis. Ce qui au plus produit $2N$ opérations de tri (une opération par nœud d'arc). Chaque opération de tri engage au plus $n-1$ comparaisons, ce qui donne une complexité en $O(nN)$ par pas de temps. Ce n'est pas négligeable mais pas handicapant sachant que plus l'algorithme progresse vers une solution, plus l'ajout de phéromone est ciblé sur quelques arcs seulement.

Mise à jour du champ de phéromone à chaque fin de cycle hamiltonien. Lorsque tous les agents ont terminé de parcourir un cycle hamiltonien (donc après n pas de temps), l'environnement met à jour de nouveau le champ de phéromones en utilisant une heuristique

élitiste supplémentaire (elle fait intervenir le calcul (12.17)) :

$$\triangle\tau = \frac{G_{min}}{\min_{i\in\mathbb{A}}\{L_i\}} \tag{12.23}$$

$$\mathcal{T}^{t+1}(k,l) = \begin{cases} (1-\rho)\mathcal{T}^t(k,l) + \triangle\tau \text{ si } (k,l) \text{ est un arc du chemin minimal,} \\ (1-\rho)\mathcal{T}^t(k,l) \text{ sinon.} \end{cases} \tag{12.24}$$

Ré-initialisation à chaque fin de cycle hamiltonien. Enfin à chaque fin de séquence de recherche d'un cycle hamiltonien, la population d'agents est réinitialisée, avec une variable interne de décision tirée aléatoirement dans $[0,1]$, une position tirée aléatoirement sur les nœuds du graphe. Le champ des nœuds visités, le champ de phéromone et le champ des arcs triés sont réinitialisés à zéros pour les deux premiers et dans l'ordre d'origine pour le dernier. Et un nouveau cycle de recherche de solution peut démarrer.

Algorithme global

L'algorithme 12.2 résume l'ensemble de ce qui vient d'être expliqué dans une boucle finie de calculs qui termine pour un temps fixé t_{stop}.

12.3.2 Simulations et résultats

Les simulations effectuées et les résultats exposés ici ont été réalisés systématiquement en fixant le nombre maximal à 20000 pas de temps de l'algorithme pendant lequel tous les agents effectuent un déplacement de nœud. Le problème est donc "att48" dont la représentation et la solution optimale sont donnés en figure 12.6(a). La distance du circuit hamiltonien minimal est 10628. Le tableau 12.1 donne les meilleures performances de l'algorithme "Ant System" (AS) de Dorigo et de notre algorithme des fourmis logistiques (noté LA ici pour "Logistic Ants") en variant le nombre de fourmis jusqu'au nombre de villes. Ces meilleures performances ont été sélectionnées pour un jeu de paramètres impliquant le coefficient d'évaporation dans les deux algorithmes, qui varie ici de $0,0$ à $0,1$ par pas de $0,01$ unité. Cinq simulations ont été lancées sur chaque configuration. Deux autres paramètres ont été fixés également pour l'algorithme LA : la pente et le centre de la sigmoïde fixés respectivement après étude comparée des performances à $\alpha = 0,04$ et $\tau_0 = 30,0$.

Aucun des deux algorithmes n'est parvenu à trouver la solution optimale sur ce temps limité de fonctionnement. Les deux algorithmes ont des résultats en performance pure assez proches. Cependant, l'évolution dans le temps des deux algorithmes est assez différente. Les figures 12.6(b) et 12.6(d) donnent les résultats dans les mêmes échelles de l'algorithme AS et de notre algorithme des fourmis logistiques (noté LA ici pour "Logistic Ants") sous forme de courbes. On constate une descente très rapide de l'algorithme AS vers des sous-optimums tandis que l'algorithme LA prend plus de temps à converger vers des solutions intéressantes. Le nombre d'agents impliqués accélère nettement cette tendance dans le cas LA tandis que cela est plus diffus dans le cas AS.

En termes de temps de calcul pur, le tableau 12.2 donne une idée pour comparer les deux algorithmes exécutés sur une même machine, implémentés dans le même langage (java en l'oc-

Algorithm 12.2: Logistic ant algorithm for the TSP.

```
Initialization
{G(S, A, n)} ← Graph("att48")
W ← computeWeights("att48")
p ← 0
T ← 0
repeat
    O ← edgeOrderingFrom("att48")
    V ← 0
    for t = pn to ((p + 1)n − 1) do
        for i,  i = 1 to N do
            Perception
            P_i ← T(argmax_{l∈S\{k∪V_i}}{O(k, l)})
            Decision
            a_i ← 1 / (1 + e^{α (P_i − τ_0)})
            x_i ← f(x_i, a_i)
            Action
            l* ∈ S ← chooseEdge(x_i)
            moveTo(l*)
            V_i(k, l*) ← 1
            V_i(k, k) ← 1
            T̃_i(k, l*) ← x_i W(k, l*)
        Local reaction
        T ← T + Σ_{i=1}^{N} T̃_i
        ∀i,  T̃_i ← 0
        O ← edgeOrdering(G(S, A, n))
    Global reaction
    L_min ← min_i{L_i}
    △τ ← G_min / L_min
    for (k, l) ∈ A do
        if (k, l) ∈ L_min then
            T(k, l) ← (1 − ρ)T(k, l) + △τ
        else
            T(k, l) ← (1 − ρ)T(k, l)
    p ← p + 1
until t < t_stop
```

Algorithme	Nombre d'agents	coefficient d'évaporation	meilleure performance
AS	10	$0,0$	11028
AS	20	$0,06$	10845
AS	30	$0,06$	10847
AS	40	$0,08$	10777
LA	10	$0,01$	11074
LA	20	$0,0$	11049
LA	30	$0,02$	10894
LA	40	$0,01$	11026

TAB. 12.1 – Tableau des résultats associées aux meilleures performances pour les deux algorithmes "Ants System" (AS) et fourmis logistiques (LA).

Calcul effectué	Temps AS	Temps LA
20000 pas de temps 10 fois, $N = 10$	$13s$	$17,6s$
20000 pas de temps 10 fois, $N = 20$	$17,6s$	$31,9s$
20000 pas de temps 10 fois, $N = 30$	$22,6s$	$44,1s$
20000 pas de temps 10 fois, $N = 40$	$28,0s$	$55,8s$

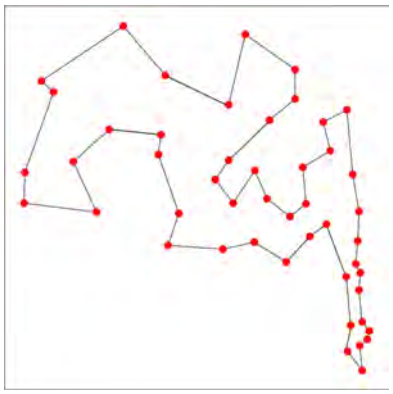
TAB. 12.2 – Tableau des temps de calcul comparés des deux algorithmes "Ant System" (AS) et fourmis logistiques (LA).

currence), et sur une même plate-forme de simulation qui garantit l'usage de mêmes classes de base. On constate en utilisant des calculs en double précision dans les deux cas, un facteur deux dans les temps de calcul. Ce facteur est décuplé si l'on emploie des précisions plus élevées dans le cas de l'algorithme LA (quadruple précision par exemple).

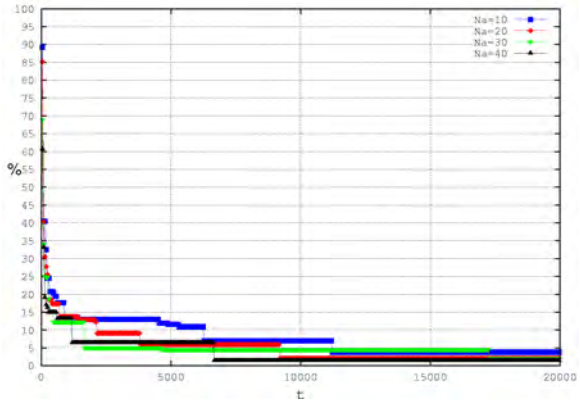
Enfin la figure 12.7 montre les variations de performance dans les 5000 premiers pas de temps sous l'effet du coefficient d'évaporation. Ces graphiques confirment tout d'abord la remarque précédente concernant la vitesse de convergence qualitative des deux algorithmes, l'algorithme des fourmis logistiques étant plus long à trouver de bonnes solutions. On constate ensuite un effet quasiment inverse de l'évaporation dans les deux cas. En effet, des coefficients d'évaporation élevés tendent à augmenter le temps de convergence vers de bonnes solutions pour les fourmis logistiques, tandis qu'ils semblent nécessaires dans le cas AS pour raccourcir ce temps de convergence.

12.3.3 Bilan

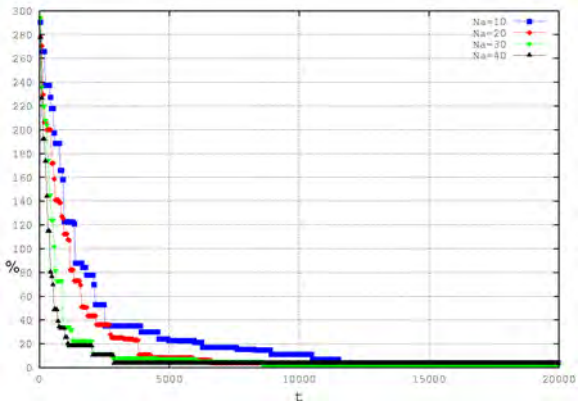
Les résultats obtenus par simulation nous semblent suffisamment probants pour affirmer que le modèle SMAL est en mesure de constituer une métaheuristique susceptible de répondre à des problèmes d'optimisation au même titre que les algorithmes déjà existants de l'intelligence en essaim. Notre algorithme est encore en phase de développement et nous pensons pouvoir en améliorer les performances ainsi qu'en automatiser les réglages. En effet, certains



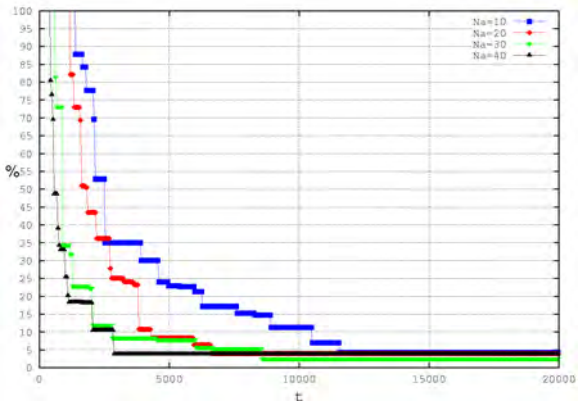
(a) Graphe att48 et cycle optimal.



(b) Meilleures performances en % par rapport à l'optimal pour l'algorithme "Ant System" en fonction du temps et du nombre de fourmis.



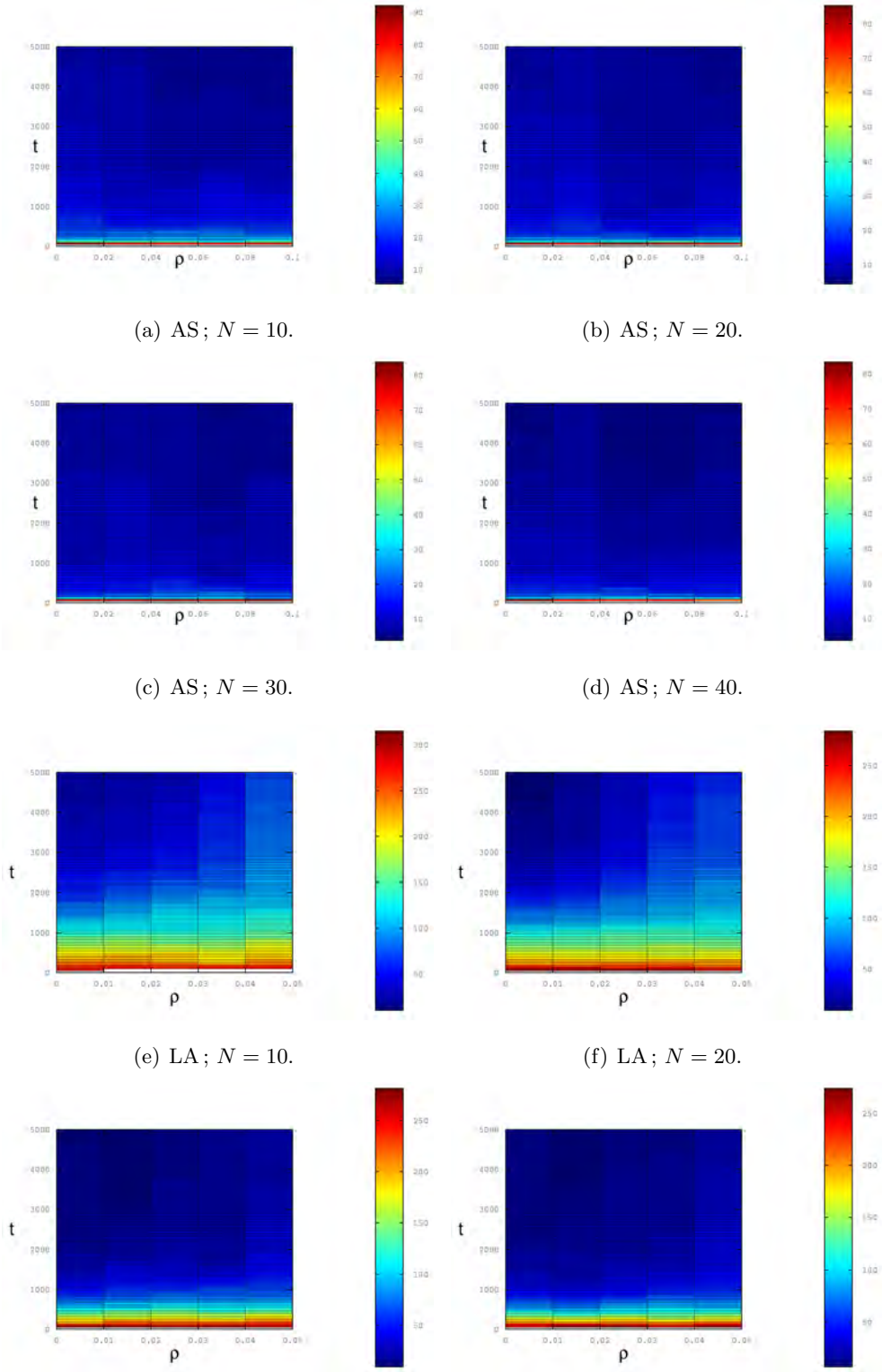
(c) Meilleures performances en % par rapport à l'optimal pour les fourmis logistiques en fonction du temps et du nombre de fourmis.



(d) Fenêtre de 0 à 100%.

FIG. 12.6 – Problème TSP att48 (48 villes) de cycle hamiltonien optimal 10628. Meilleures performances de l'algorithme AS et des fourmis logistiques en fonction du nombre de fourmis utilisées.

paramètres, concernant la sigmoïde notamment, nécessitent encore des simulations pour connaître les meilleures plages de valeurs pour l'optimisation. Nous visons à trouver des relations plus systématiques entre ces paramètres et les grandeurs caractéristiques du problème considéré.



(a) AS ; $N = 10$.

(b) AS ; $N = 20$.

(c) AS ; $N = 30$.

(d) AS ; $N = 40$.

(e) LA ; $N = 10$.

(f) LA ; $N = 20$.

(g) LA ; $N = 30$.

(h) LA ; $N = 40$.

FIG. 12.7 – Performances comparées sur 5000 pas de temps et en fonction du coefficient d'évaporation pour les deux algorithmes Ant System (AS) et fourmis logistic (LA).

13

Discussion

Analyse et interprétation de l'algorithme

Nous estimons que l'objectif de modélisation du comportement de fourragement des fourmis est atteint et peut-être mieux encore. Car notre modèle apporte une véritable explication constructive aux phénomènes mis en jeu, en plus de fournir une base algorithmique générique. Nous avons en effet pu exhiber clairement les mécanismes d'auto-organisation et d'adaptation par contrôle décentralisé qui sont à l'oeuvre dans les fourmis logistiques, via les processus sensorimoteurs des agents. L'auto-organisation dans ce cas est vue comme une modification de la variable de contrôle interne et locale à l'agent qui passe d'une valeur maximale impliquant un comportement chaotique exploratoire des agents, à une valeur faible qui génère un comportement stable et prédictible des agents. L'adaptation a pour effet d'associer cette convergence à une solution du problème, pas nécessairement optimale.

Ainsi l'algorithme montre que le champ de phéromone apparaît à la fois comme un champ de contrôle pour les agents et comme un champ de décision, puisqu'il est construit à partir des valeurs décisionnelles cumulées des agents de façon persistante (ou plus souvent volatile). Le champ de phéromone produit un mécanisme d'auto-organisation puisque la phéromone est une donnée endogène aux agents. Lorsqu'une fonction objectif est injectée dans le système, en l'occurrence pour minimiser une distance dans le TSP, le mécanisme devient également adaptatif puisque ces données d'objectif sont exogènes. De ce fait, on peut affirmer que l'algorithmique des fourmis logistiques met en jeu à la fois des mécanismes d'auto-organisation et d'adaptation qui fusionnent dans la phéromone numérique.

Perspectives

Les voies d'amélioration de notre modèle de fourmis logistiques suivent deux axes :
- utiliser le couplage qui est inexploité pour le moment dans la version actuelle,
- exploiter des principes d'optimalité plus génériques.

Le premier point vise à tirer parti des phénomènes de flocking vus dans les chapitres précédents, pour explorer les solutions voisines d'une solution en construction. La métaheuristique dite de "recherche locale" est déjà fondée sur ce principe : l'idée est modifier localement une

solution pour trouver l'optimal, autrement dit on visite les solutions dans un voisinage de modification de la solution courante. Cette technique est également utilisée dans une certaine mesure par les essaims particulaires pour l'optimisation, puisque les meilleures solutions découvertes par l'essaim sont diffusées et influent sur la dynamique des autres particules.

Le second point focalise sur l'aspect théorique des méthodes d'optimisation. Nous avons de fait suivi la démarche de modélisation existante des algorithmes fourmis de Dorigo, mais il est nécessaire de rapprocher ces métaheuristiques de théories plus générales pour une meilleure compréhension de leur fonctionnement et pour en augmenter les performances. Un certain nombre de travaux tentent déjà d'interpréter les algorithmes fourmis dans le cadre de l'apprentissage par renforcement (cf. [Birattari *et al.*, 2002, Boumaza et Scherrer, 2007]) ; une version de l'algorithme adaptant l'approche "Q-learning" est proposée dans [Gambardella et Dorigo, 1995].

Cette section termine notre exploration de l'algorithmique fourmi à l'aide de notre modèle de fourmis logistiques issu du SMAL. Après avoir repris la modélisation de l'expérience originelle du double pont, nous avons montré la pertinence de notre approche sur des problèmes du type TSP. L'application de notre algorithme à ce type de problème ouvre l'accès à d'autres types de problèmes d'optimisation comme les algorithmes fourmis de Dorigo ont pu le faire. Ce que nous voulions avant tout démontrer à ce stade était l'adéquation du modèle SMAL avec les aspects métaheuristiques de l'intelligence en essaim aujourd'hui.

Cinquième partie

Conclusion

Originalité et contributions principales.

Généricité. Le système multi-agent logistique présenté dans cette thèse est à notre connaissance une des rares tentatives de modélisation de deux grands types de phénoménologies rencontrées dans le domaine de l'intelligence en essaim, à savoir les dynamiques de formation de groupes d'agents (désignées ici sous le terme générique de flocking) et les dynamiques stigmergiques à la base des comportements individuels et collectifs présentes chez de nombreuses espèces animales, comme les insectes sociaux pour lesquels la phéromone chimique est un vecteur majeur de l'interaction. Ce modèle ne prétend pas en l'état actuel modéliser ces phénomènes dans une vision biologique, c'est-à-dire visant à reproduire le plus fidèlement possible les données expérimentales. En revanche il constitue une représentation abstraite et simplifiée permettant d'approcher, par une description qualitative et conceptuelle, la compréhension de certains mécanismes de l'intelligence en essaim. Ce modèle est un modèle théorique et notre objectif a été d'en formaliser les éléments constitutifs de façon à rendre l'éclairage qu'il projette sur l'intelligence en essaim, le plus cohérent possible.

Dans cet esprit, seules trois grandeurs propres à l'agent logistique sont identifiées : la décision, le couplage et le contrôle internes. D'autre part, le concept de champ est systématiquement employé pour décrire l'environnement ainsi que sa fonction de mémoire intégrative et computationnelle du système. Ces concepts simples et génériques garantissent des possibilités d'instanciation larges pour aborder divers types de problèmes, dont certains ont été traités dans cette thèse. C'est bien cet aspect générique et unificateur de notre modèle que nous souhaitons faire ressortir en premier lieu.

Utilisation des concepts et outils du chaos déterministe. L'utilisation de la théorie du chaos déterministe est également une des particularités de ce modèle typiquement issu de la modélisation des systèmes complexes. Bien que de nombreuses précautions soient nécessaires pour manipuler ces concepts parfois d'une haute abstraction mathématique, nous disposons d'une abondante littérature et d'outils puissants d'analyse et de visualisation à ce sujet. Nous sommes convaincus que cette voie est prometteuse : de nombreux phénomènes naturels apparemment très différents présentent des comportements régis par des lois sous-jacentes non linéaires universelles. Ainsi l'application logistique résume-t-elle à elle seule une grande partie de ces phénomènes dynamiques spécifiques que l'on retrouve aussi bien dans la physique des lasers que dans le comportement de certains réseaux neuronaux. Dans ce domaine l'aspect qualitatif de la dynamique prime sur le quantitatif ce qui justifie l'emploi de cette application comme fonction génératrice des comportements dynamiques des agents. De ce fait, l'application logistique dans l'agent logistique peut être considérée comme la réduction d'un certain type de réseau de neurones chaotiques. Cette abstraction est de fait plus proche de la réalité biologique qu'on ne pourrait l'imaginer au premier abord. Certains scientifiques considèrent ainsi le chaos comme une façon de générer de l'aléatoire naturel, même si les deux concepts sont mathématiquement différents. C'est également l'hypothèse implicite qui est faite dans cette thèse.

D'autre part, l'approche déterministe des mécanismes permet une analyse au moyen de grandeurs caractéristiques telles que les exposants de Lyapunov. Le calcul de ceux-ci peut être approché dans le cas des agents logistiques par une technique que l'on a désignée sous le terme

d'"agents jumeaux". Cette technique est en fait applicable dans de nombreuses configurations de systèmes, en dehors de notre modèle SMAL. A partir de ces estimations, des mesures plus globales ont été considérées, comme l'entropie de Kolmogorov dans une version approchée, qui donne là encore un élément précieux d'analyse pour les grands systèmes. Cette définition de l'entropie résume l'instabilité d'un système à une seule grandeur. Elle peut également caractériser les systèmes fragmentés comportant différents clusters d'agents, en quantifiant des grandeurs entropiques partielles.

Enfin, les possibilités de visualisation et de suivi de la dynamique des agents, via leur diagramme de bifurcation ou les techniques de reconstruction des attracteurs à partir des données expérimentales, complètent la gamme des outils que procure la théorie des systèmes dynamiques non linéaires.

Tous ces outils d'analyse et de visualisation sont associés à notre démarche de modélisation et lui confèrent un aspect qualitatif et quantitatif, et dans certains cas analytique, qui manque parfois aux approches agents. Bien que l'emploi de ces outils ne soit pas encore généralisé dans notre cas de figure, il semble indispensable d'en développer l'usage de façon systématique car ces outils justifient l'intérêt porté à ce type de modélisation déterministe.

Construction et interprétation des mécanismes d'auto-organisation et d'adaptation. D'autre part, notre système multi-agent logistique montre explicitement comment peuvent se construire et se combiner les mécanismes d'auto-organisation et d'adaptation. Toute l'information du système passant par les champs de l'environnement, ces mécanismes résultent donc du traitement des données perçues par les agents dans le(s) champ(s) qui agissent ensuite sur une de ses trois grandeurs internes pour générer une(des) action(s). Ces actions ont globalement une influence sur l'environnement en modifiant éventuellement en retour la valeur des champs de l'environnement.

La différence entre phénomènes d'auto-organisation et d'adaptation provient du caractère endogène ou exogène des données manipulées par les agents dans l'environnement. En fait le modèle SMAL révèle une auto-organisation liée à des champs endogènes, comme le champ de phéromone (cas des fourmis) ou le champ de décision des agents (cas du flocking), qui sont créés par les agents eux-mêmes, tandis que l'adaptation implique des champs persistants et exogènes de données, les "entrées" du système multi-agent, auxquels les agents répondent par une configuration ou un schéma spécifique, par exemple dans le cas d'un objectif d'optimisation explicite.

Nous avons pu mettre en évidence deux résultats conceptuels importants qui découlent de notre modélisation :

- Le phénomène d'auto-organisation de type flocking est l'expression directe dans l'espace géométrique d'une synchronisation des états internes des agents par l'effet du couplage. Cette synchronisation souvent décrite comme intrinsèque au phénomène de flocking prend ici un sens mathématique concret et computationnel. Autrement dit, nous avons géométrisé la dynamique interne des agents dans l'environnement.
- Le phénomène de stigmergie quant à lui résulte d'un processus d'auto-organisation et d'adaptation à l'environnement par le biais d'un champ de phéromone jouant le rôle d'un

champ de contrôle, accessible aux agents en lecture et écriture. Ce champ agit directement sur la variable de contrôle interne des agents et indirectement sur leur réponse comportementale dynamique (via leur variable de décision).

On a montré ainsi que les deux grandeurs complémentaires à la décision de l'agent que sont le couplage et le contrôle sont respectivement associées aux deux types de phénomènes de l'intelligence en essaim étudiés dans cette thèse.

Enfin la fonction objectif des problèmes d'optimisation est vue comme une couche supplémentaire aux mécanismes de couplage et de contrôle internes au SMAL, qui contient des données spécifiques stockées dans un champ de données exogènes de l'environnement et auquel le système doit répondre de façon optimale. Toute la question a été de distribuer localement cet objectif global aux agents du SMAL pour leur calcul de décision, et donc de structurer le flux de données, afin qu'ils fournissent une réponse conjointe satisfaisante.

Résultats obtenus. Les résultats d'implémentation du système multi-agent logistique a permis de confirmer les capacités du modèle à générer des schémas typiques de l'intelligence en essaim. Parmi ces schémas, la formation de clusters d'agents partiellement synchronisés a conduit à de variations autour des phénomènes de flocking. On a notamment pu vérifier expérimentalement une loi de seuil théorique dans le cas d'agents tous chaotiques et caractériser la dynamique du flocking par le calcul de coefficients de Lyapunov et l'entropie K.

De cette classe de phénomènes, on est passé à une classe de phénomènes stigmergiques par ajout de mécanismes d'adaptation simples, ce qui nous a conduit à proposer au final un algorithme de fourmis logistiques. Les résultats obtenus sont suffisamment convaincants pour confirmer le potentiel de notre algorithme comme métaheuristique naissante pour l'optimisation combinatoire. L'optimisation par essaim particulaire, sans avoir été réellement comparée, est mentionnée dans une déclinaison d'un problème "proie-prédateurs". Elle figure ainsi comme une extension future de l'application de notre modèle à d'autres types d'optimisation.

Perspectives et travaux futurs

Nos travaux futurs sont nombreux sur le SMAL car l'étendue des possibilités est beaucoup plus grande que ce que nous avons pu montrer dans cette thèse. Nous avons tenté de baliser quelques pistes importantes qui illustrent selon nous les mécanismes essentiels de ce modèle, utiles pour traiter de l'intelligence en essaim. Les axes à développer s'orientent selon plusieurs objectifs complémentaires.

Espaces n-dimensionnels. Notre modèle à été instancié essentiellement dans des cas présentant 1 dimension décisionnelle. Concevoir des instances à décision multidimensionnelle est un enjeu central pour pouvoir aborder l'optimisation dans des espaces de dimensions quelconques. L'extension du repère de Frenet, employé pour établir des prises de décision de changement de directions de façon locale, est envisageable à des espaces de grandes dimensions, en multipliant les angles caractérisant l'orientation de l'agent.

Mécanismes d'adaptation. Les mécanismes d'adaptation ont été explorés séparément selon les composantes de couplage et de contrôle des agents ; il est nécessaire de comprendre plus finement comment articuler conjointement ces deux types d'adaptation à l'environnement, notamment en ce qui concerne l'application au problème proie-prédateurs et à notre algorithme des fourmis logistiques.

Outils d'analyse. L'aspect mathématique déterministe du modèle SMAL permet *a priori* des analyses fines sur les dynamiques produites dans le modèle, qui seraient hors d'atteinte avec d'autres approches. Nous avons montré quelques exemples de ces outils qui permettent d'analyser le degré d'avancement d'un mécanisme (diagramme de bifurcation) ou le degré de stabilisation des schémas formés (entropie K). Cette démarche doit être approfondie et systématisée car elle donne tout son intérêt au modèle et permet d'avancer autant que possible sur des bases établies mathématiquement : cela implique cependant certaines limitations dues aux systèmes complexes non linéaires que l'on manipule.

Optimisation. Enfin, pour ce qui relève plus spécifiquement des problèmes d'optimisation, une réflexion doit être menée sur la faisabilité d'appliquer notre modèle comme métaheuristique de résolution pour des théories plus généralistes dans ce cadre, comme la programmation dynamique. Une discussion à ce propos est développée en section 13.

Note finale de l'auteur

Cette thèse relève d'une certaine façon d'une "pensée complexe" au sens que lui donne le philosophe Edgar Morin [Morin et Weinmann, 1998], c'est-à-dire au sens étymologique de "ce qui est tissé ensemble" : elle puise ses sources conceptuelles dans divers champs scientifiques, dont les systèmes complexes et les systèmes multi-agents, et propose de fusionner les deux types de modélisation. Cette liaison des deux approches est selon nous riche de conséquences, car elle bénéficie des caractéristiques propres aux deux mondes. Elle conduit également à une nouvelle lecture, interprétation, compréhension des phénomènes résultant de ce rapprochement. Plus globalement, notre intention est enfin que les liens "tissés" ainsi puissent amener une nouvelle vision de l'intelligence en essaim.

Sixième partie

Annexes

A

Les systèmes dynamiques

A.1 Définitions générales sur les systèmes dynamiques

Introduction

Cette annexe a pour but de faire un survol synthétique des notions qui nous sont utiles dans cette thèse sur le domaine mathématique des systèmes dynamiques. Nous commencerons par définir le formalisme habituellement utilisé, les concepts principaux utilisés dans ce texte et l'énoncé des théorèmes majeurs du domaine restreints au cadre de notre utilisation de cette théorie très vaste. Nous nous focaliserons ici plus particulièrement sur les aspects non linéaires de la théorie des systèmes dynamiques à temps discrets, même si certaines notions seront posées d'abord dans le domaine du continu. Le fait de positionner une démarche de modélisation mathématique dans ce champ théorique n'est en effet pas sans conséquences à la fois sur le plan purement formel et sur le plan conceptuel et sémantique. Le courant des approches dynamiques dans la modélisation des réseaux de neurones en est un exemple convaincant, notamment en ce qui concerne les réseaux de neurones chaotiques, puisqu'il permet des analyses dynamiques en termes d'attracteurs et de bifurcations. Il est donc nécessaire d'en rappeler les bases conceptuelles. La plupart des énoncés, définitions, notations et la base formelle présentés ici sont largement inspirés de l'ouvrage [Dang-Vu et Delcarte, 2000].

Rappel historique et premières définitions

L'origine de la théorie des systèmes dynamiques est liée à l'étude des équations différentielles connues depuis Newton et qui s'est engagée dans des approches plus qualitatives à partir des travaux de Henri Poincaré échafaudant une "théorie qualitative des équations différentielles" aujourd'hui appelée théorie des systèmes dynamiques. La théorie des perturbations, la théorie des bifurcations et la théorie des catastrophes en sont par exemple des branches plus contemporaines. La théorie des systèmes dynamiques vise à étudier des processus déterministes, finis et différentiables [Arnold, 1988] :

- Un processus est déterministe si toute son évolution et son passé sont définis de façon unique par son état présent. L'ensemble de tous les états de ce processus

s'appelle espace des phases.
- Un processus est de dimension finie s'il en est de même de son espace des phases, c'est-à-dire si le nombre des paramètres indispensable à la description de son état est fini.
- Un processus est différentiable si son espace des phases est muni d'une structure de variété différentiable, et ses changements d'états dans le temps définis par des fonctions différentiables.

D'une manière plus formelle :

Définition A.1.1. *On appelle système dynamique à temps continu, une équation différentielle de la forme :*

$$\frac{dx}{dt} = \dot{x} = f(x, t, \nu) \tag{A.1}$$

où $x \in U \subseteq \mathbb{R}^n$ désigne un vecteur de l'espace des phases $\mathbb{R}^n$ du système et $\nu \in V \subseteq \mathbb{R}^p$ est un vecteur de l'espace des paramètres $\mathbb{R}^p$.
De même à temps discret on écrira une transition d'état du système dynamique sous la forme suivante :

$$x_{k+1} = f(x_k, \nu) \tag{A.2}$$

où $k = 1, 2, \ldots$ désigne le temps discret.

Pour des raisons de commodité d'écriture et comme c'est généralement le cas avec la manipulation des systèmes dynamiques, les paramètres du système seront omis par la suite, sauf lorsqu'ils jouent un rôle prépondérant. Nous introduisons dans un premier temps les principaux concepts en temps continu pour ensuite les transcrire en temps discret.

Un premier élément sur les systèmes dynamiques qui explique la forme différentielle générique A.1, est que toute équation différentielle d'ordre $n > 1$ peut être transformée en un système de n équations différentielles du premier ordre. Ainsi une équation différentielle du type :

$$\frac{d^n x}{dt^n} = f(x, \frac{dx}{dt}, \cdots, \frac{d^{n-1}x}{dt^{n-1}}, t) \tag{A.3}$$

peut s'écrire sous la forme d'un système du type A.1 par changement de variable $x_1 = x$ puis en écrivant le système d'équations :

$$\begin{aligned}
\frac{dx_1}{dt} &= x_2 \\
\frac{dx_2}{dt} &= x_3 \\
&\cdots \\
\frac{dx_{n-1}}{dt} &= x_{n-2} \\
\frac{dx_n}{dt} &= f(x_1, \ldots, x_{n-1}, t)
\end{aligned}$$

Ainsi l'étude des systèmes dynamiques peut se ramener systématiquement à l'ordre différentiel 1 en augmentant la taille de l'espace des phases. Tout système dynamique fait donc intervenir

un vecteur $x = (x_1, x_2, \ldots, x_n)^T$ de l'espace des phases $\mathbb{R}^n$, des paramètres et un champ de vecteurs $f = (f_1, f_2, \cdots, f_n)^T$, chaque composante f_i étant de classe C^r en x et t avec $r \geqslant 1$. La solution du système A.1 est noté $x(x_0, t)$, où $x_0 = x(t_0)$ est la condition initiale, et correspond à une trajectoire ou orbite du système. Si f dépend explicitement de t le système est dit non autonome et la solution précédente dépend explicitement de t_0, sinon il est dit autonome. Bien que nous n'utiliserons pas cette notion explicitement dans cette thèse, le flot associé à un système dynamique est défini de la manière suivante :

Définition A.1.2. *Soit $x(x_0, t)$ une solution de A.1 avec la condition intitale $x(0) = x_0 \in D \subset \mathbb{R}^n$, que l'on appelle trajectoire ou orbite du système. On appelle flot du système A.1 l'application $\phi_t : D \to \mathbb{R}^n$ définie par :*

$$\phi_t(x_0) = x(x_0, t) \tag{A.4}$$

et possède les propriétés suivantes :
(i) $\phi_t(x_0)$ est de classe C^r
(ii) $\phi_0(x_0) = x_0$
(iii) $\phi_{t+s}(x_0) = \phi_t(\phi_s(x_0))$
Ajoutons qu'un ensemble $V \subset \mathbb{R}^n$ est un ensemble invariant [53] *du système A.1 si $\phi_t(V) \subset V$*

Ainsi définie, la famille d'applications ϕ_t a une structure de semi-groupe.

Point fixe et notion de stabilité

Point fixe. Abordons la notion importante de point fixe :

Définition A.1.3. *Un point fixe du système A.1 est un point $\bar{x}$ de l'espace des phases vérifiant l'équation :*

$$f(\bar{x}) = 0 \tag{A.5}$$

Remarques :
- par changement de variable, on peut toujours ramener un point fixe en 0
- un point fixe est également appelé point critique, point d'équilibre ou point stationnaire
- de plus un point fixe peut être stable, asymptotiquement stable ou instable : par exemple le pendule à tige rigide permet de visualiser ces trois situations. Il y a deux points d'équilibre à la verticale l'un de l'autre, un point d'équilibre bas et un point haut. Le point d'équilibre haut est instable puisque la moindre perturbation fera basculer le pendule d'un côté ou de l'autre. Le point d'équilibre bas est simplement stable s'il n'y a aucun effet d'amortissement (du aux forces de frottement notamment) car le pendule aura un mouvement de balancier perpétuel sans jamais stopper. En revanche si un amortissement existe, le pendule tend vers le point d'équilibre bas à temps infini, il est donc asymptotiquement stable en ce point.

[53] Un concept plus large est celui de variété invariante, une variété est un espace topologique abstrait défini par recollement d'espaces simples.

Stabilité du point fixe. Ces notions de stabilité sont précisées mathématiquement par les définitions suivantes :

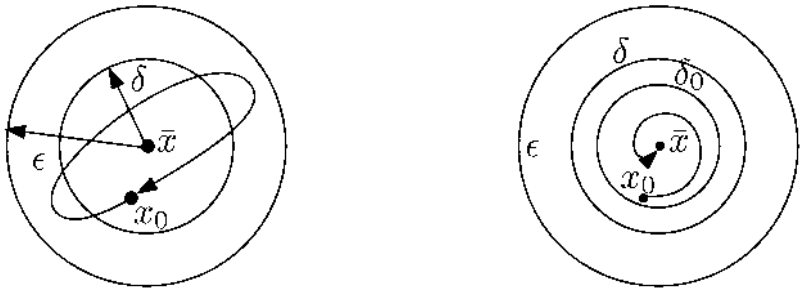
Définition A.1.4. *Un point fixe $\bar{x} \in \mathbb{R}^n$ est stable si $\forall\ \varepsilon > 0, \exists\ \delta > 0$ tel que :*

$$\|x(0) - \bar{x}\| < \delta \Rightarrow \|x(t) - \bar{x}\| < \varepsilon$$

Le point $\bar{x}$ est asymptotiquement stable si de plus $\exists\ 0 < \delta_0 < \delta$ tel que :

$$\|x(0) - \bar{x}\| < \delta_0 \Rightarrow \lim_{t \to \infty} x(t) = \bar{x}$$

Sinon le point fixe est instable.



(a) Point fixe stable

(b) Point fixe asymptotiquement stable

FIG. A.1 – Schémas illustant les types de stabilité d'un point fixe.

Cette définition de stabilité est très générale et il est possible d'accéder à une analyse plus fine du type de point fixe par linéarisation locale du système dynamique au voisinage d'un point fixe $\bar{x}$ par calcul des valeurs propres de la matrice jacobienne du système linéarisé. Ainsi, en supposant le point fixe ramené en 0 par changement de variable, on a par développement de Taylor au premier ordre au voisinage de $\bar{x} = 0$ l'approximation suivante :

$$\dot{x} = f(x) \approx Df(0)\ x \tag{A.6}$$

avec $f = (f_1, f_2, \cdots, f_n)^T$ et $x = (x_1, x_2, \ldots, x_n)^T$. On définit la matrice jacobienne de $f(x)$ de cette façon :

$$Df(x) \equiv \left(\frac{\partial f_i(x)}{\partial x_j} \right)$$

Le déterminant de $Df(x)$ est le jacobien de $f(x)$ et permet de calculer les valeurs propres de la matrice jacobienne. Supposons que cette matrice possède n valeurs propres distinctes $(\lambda_i)_{i=1,2,\ldots,n}$, la solution de A.6 s'écrit :

$$x(t) = \sum_{i=1}^{n} c_i\ a_i \exp^{\lambda_i t} \tag{A.7}$$

où c_i $(i = 1, \ldots, n)$ est une constante et a_i $(i = 1, \ldots, n)$ est le vecteur propre associé à la valeur propre λ_i. De l'analyse de ces valeurs propres découle un ensemble de conclusions possibles sur le point fixe :

- Si une des valeurs propres est strictement positive, le point fixe est instable.
- Si toutes les valeurs propres ont une partie réelle strictement négative, le point fixe est asymptotiquement stable.
- Si une ou plusieurs de ces valeurs imaginaire pure (partie réelle nulle), les autres étant strictement négatives, le point fixe est stable : c'est un centre ou un point elliptique.
- Si toutes les valeurs propres sont réelles et de même signe, le point fixe est un noeud (stable ou instable).
- S'il existe deux valeurs propres distinctes à parties réelles non nulles mais de signes opposés, le point fixe est un point selle.

Le théorème de Poincaré-Bendixon. Après cette typologie des points fixes à temps continu, citons un des théorèmes importants du domaine qui permet de déterminer la forme d'une solution sans forcément la connaître analytiquement, le théorème de Poincaré-Bendixon :

Théorème A.1.1. *On considère le système dynamique en deux dimensions suivant :*

$$\frac{dx}{dt} = f(x, t, \nu), \qquad x = (x_1, x_2)^T, \qquad f = (f_1, f_2)^T$$

On suppose qu'une orbite $x(x_0, t)$ reste à tout instant dans une partie compacte $D \subset \mathbb{R}^2$. On peut alors conclure à l'une des trois situations suivantes :

(i) l'orbite est une solution périodique du système
(ii) l'orbite tend vers une solution périodique
(iii) l'orbite tend vers un point fixe

Ce théorème implique également le fait suivant important suivant : il interdit l'existence de phénomènes chaotiques dans les systèmes dynamiques à temps continu de dimension inférieure à 2. Nous verrons dans la section suivante qu'il en est tout autrement pour les systèmes à temps discret.

Attracteurs et chaos déterministe

Attracteurs d'un système. Enfin pour clore cette introduction aux concepts de base des systèmes dynamiques, explicitons ce qu'on appelle un attracteur [Strogatz, 2001]. De façon intuitive, un attracteur est un ensemble de l'espace des phases vers lequel toute trajectoire de son voisinage converge.

Définition A.1.5. *Soit A un ensemble de l'espace des phases compact, fermé. A est un attracteur s'il vérifie les propriétés suivantes :*

- *A est invariant par f, i.e. il vérifie la relation $\phi_t(A) \subset A$, $\forall t$. Autrement dit, toute trajectoire commençant dans A reste dans A au cours du temps*
- *A est attracteur d'un ensemble ouvert de conditions initiales, c'est-à-dire :*
  il existe un ensemble ouvert $U \supset A$ tel que si $x(0) \in U$ alors la distance de $x(t)$ à A tend vers 0 lorsque $t \to \infty$. Ce qui signifie que A attire toute trajectoire suffisamment proche. Le plus grand des ensembles U est le bassin d'attraction de A.

– *A est minimal : il n'existe pas de sous-ensemble de A vérifiant les deux conditions précédentes.*

L'attracteur le plus simple est le point fixe, mais un attracteur peut aussi être un cycle limite. Un attracteur étrange est un attracteur qui possède essentiellement une propriété de sensibilité aux conditions initiale supplémentaire : deux trajectoires sur l'attracteur initialement voisines se séparent de façon exponentielle (la notion d'exposant de Liapunov permet de quantifier cette divergence). On peut distinguer deux types d'attracteurs étranges, les attracteurs fractals et les attracteurs chaotiques.

Chaos déterministe. Le terme de "chaos" désormais établi dans la communauté des systèmes dynamiques n'a pas pour autant de définition qui fasse consensus. On associe cependant au chaos trois caractéristiques essentielles qui sont réunies sous cette définition [Strogatz, 2001] :

Définition A.1.6. *Le chaos désigne le comportement apériodique à long terme d'un système dynamique déterministe qui exhibe une sensibilité aux conditions initiales.*

Détaillons ces caractéristiques :
- Le comportement apériodique à long terme implique qu'il existe des orbites du système qui ne sont ni des points fixes, ni des orbites périodiques ou pseudo-périodiques lorsque $t \to \infty$. Cependant ces orbites apériodiques ne doivent pas être non plus trop rares, c'est-à-dire que leur probabilité d'apparition à condition initiale aléatoire est non nulle.
- Déterministe signifie qu'on ne fait intervenir aucune entrée/cause bruitée ou aléatoire dans le système. L'imprédictibilité du système provient uniquement de son caractère non linéaire.
- La sensibilité aux conditions initiales a déjà été définie pour les attracteurs étranges. On peut ajouter que cette caractéristique est quantifiée par l'existence d'un coefficient de Lyapunov positif pour le système.

Une revue des divers apects mathématiques du chaos déterministe est proposée dans [Ruette, 2003].

Coefficient de Lyapunov. Le coefficient de Lyapunov est précisément le moyen de quantifier la sensibilité aux conditions initiales d'un système dynamique, et de mesurer le degré de chaoticité de son évolution. Son calcul repose sur la mesure logarithmique de la divergence résultant d'une perturbation des conditions initiales. Il est abordé en section 5.3.3 de ce document dans le cas de l'application logistique.

A.2 Les systèmes dynamiques non linéaires discrets

Intérêt des systèmes à temps discrets. Jusqu'ici, nous avons considéré des systèmes dynamiques à temps continu, or ce sont les systèmes à temps discrets qui nous concernent dans cette thèse. Les concepts utilisés sont quasiment les mêmes, à l'exception de quelques spécificités des systèmes à temps discret. Ces systèmes sont également appelés équations aux différences,

relation de récursion, applications itérées ou même itérations tout court. L'intérêt de ces applications à temps discret apparaît pour différentes raisons et dans différents contextes :

- elles permettent d'analyser plus facilement certaines équations différentielles en ramenant par discrétisation du temps l'analyse de stabilité de solutions périodiques à un problème de point fixe [Strogatz, 2001],
- la modélisation en temps discret est appropriée dans l'étude de phénomènes naturels, par exemple pour l'étude des populations en biologie, l'électronique numérique et les algorithmes informatiques, l'économie, ...
- ce sont des paradigmes pour l'étude des phénomènes non linéaires, simple à calculer sur machine, et qui révèlent de façon étonnante des propriétés universelles des systèmes non linéaires : le chaos apparaît à temps discret dès la dimension 1 alors qu'il en faut au minimum 3 à temps continu (cf. théorème A.1.1).

Définitions principales. La plupart des définitions qui suivent sont tirées de [Dang-Vu et Delcarte, 2000]. Soit $D \subset \mathbb{R}^n$ et $f: \ D \to D$ une application continue (ou transformation). On note f^k la $k$ième itérée de f soit :
$f^0(x) = x, \ f^1(x) = f(x), \ldots, \quad f^{k+1}(x) = f(f^k(x)), \ldots$.
On associe immédiatement à k le rôle de temps discret dans l'équation A.2, que l'on peut écrire :

$$x_{k+1} = f(x_k) = f^{k+1}(x_0) \tag{A.8}$$

Pour tout $x \in D$, on appelle orbite ou trajectoire de x l'ensemble $\{f^k(x), k = 1, 2, \ldots, +\infty\}$. Un ensemble A est invariant par f si $f(A) = A$. De la même façon qu'en continu, un attracteur est un ensemble compact fermé $A \subset D$ invariant par f qui possède un voisinage V tel que pour tout $x_0 \in V$ l'orbite de x_0 est une séquence qui converge vers A. V est appelé bassin d'attraction de A et on a :

$$A = \bigcap_{k=1}^{\infty} f^k(V)$$

De même, un ensemble A est répulsif pour l'application s'il possède un voisinage V dans lequel l'orbite de tout point x_0 s'écarte de A. A est un attracteur étrange si tout point x de A a une orbite dense dans A (ce qui signifie que l'on peut approcher tout point de A aussi près que l'on veut par un point de l'orbite de x) et s'il est sensible aux conditions initiales (les orbites de deux points distincts sur l'attracteur s'éloignent l'une de l'autre).

Point fixe et stabilité. Un point fixe $\bar{x} \in D$ du système dynamique A.8, attracteur le plus simple de la dynamique, est défini par l'équation :

$$f(\bar{x}) = \bar{x} \tag{A.9}$$

Théorème A.2.1. *En dimension* 1 *pour* $f : \mathbb{R} \to \mathbb{R}$, *le type de point fixe est déterminé par le multiplicateur de* f *au point* $\bar{x}$ *:* $m = f'(\bar{x})$. *En effet :*

- *si* $|m| < 1$ *le point fixe est stable*
- *si* $|m| > 1$ *le point fixe est instable*
- *si* $m = 0$ *le point fixe est superstable*

– *si* $|m| = 1$ *le point fixe est indifférent*

Ce théorème s'étend en dimension quelconque en examinant les valeurs propres de la matrice jacobienne de f calculées en $\bar{x}$. Ainsi :

Théorème A.2.2. *Si toutes les valeurs propres de* $J(\bar{x}) = Df(\bar{x})$ *sont dans le disque unité,* $\bar{x}$ *est stable, mais si une des valeurs propres est extérieure au disque unité,* $\bar{x}$ *est instable.*

Notion de p-cycle. Une particularité des applications itérées est la notion de p-cycles, qui est le pendant des orbites périodiques en continu :

Définition A.2.1. *Un cycle d'ordre* p *ou* $p-$*cycle est une séquence de points* $x_0, x_1, \cdots, x_{p-1}$ *vérifiant :*
$f(x_i) = x_{i+1}$ *pour* $i = 0, 1, \cdots, p-2$ *et* $f(x_{p-1}) = x_0$.

Chaque point d'un cycle d'ordre p est un point périodique et est en fait un point fixe de f^p, ce qui ramène leur étude à celui des points fixes, mais n'est pas un point fixe de $f^k, k < p$.
De la même façon que pour les points fixes, le théorème précédent A.2.2 s'applique avec un des points périodiques d'un $p-$cycle. En effet, on définit également un multiplicateur du cycle par :

$$m_p \equiv (f^p)'(x_0) = (f^p)'(x_1) = (f^p)'(x_2) = \cdots = (f^p)'(x_{p-1}) = f'(x_{p-1}) \cdots f'(x_1)\, f'(x_0) \quad \text{(A.10)}$$

Ces dérivations sont égales pour chaque point du cycle : l'expression du membre de droite permet d'établir ces égalités par permutation des produits de dérivées, sachant que $x_0 = f(x_p)$. Et on peut également appliquer le théorème A.2.1 à l'un quelconque des points périodiques du cycle : la conclusion sur l'un de ces points suffit pour conclure sur l'ensemble du cycle.

Conclusion. Nous avons survolé très succinctement les concepts de la théorie des systèmes dynamiques utiles à la compréhension de l'application logistique largement utilisée dans cette thèse. L'application de ces différents concepts est donnée sur l'application logistique en annexe. Cependant nous ne pouvons traiter du comportement de l'application logistique sans aborder la notion de bifurcation que nous n'aborderons ici que dans le cadre d'un seul paramètre de bifurcation (systèmes dits de "codimension" 1).

A.3 Quelques éléments de théorie des bifurcations

Nous ne ferons ici que donner une description très partielle et succincte des bifurcations restreintes à un paramètre sur un système à temps discret, qui suffiront à nos besoins dans ce document (voir [Strogatz, 2001] pour une introduction didactique, [Dang-Vu et Delcarte, 2000] pour une présentation plus formelle et des développements détaillés, et [Sprott, 2003] pour des approches numériques).

Une bifurcation dans un système dynamique $\dot{x} = f(x, \nu)$ désigne un changement qualitatif de son comportement dynamique du fait des variations d'un ou plusieurs de ses paramètres. Techniquement parlant, une bifurcation apparaît lorsque le type topologique du portrait de phase[54]

[54] i.e. la représentation des trajectoires suivies par le système dans l'espace des phases

du système change lorsque un ou plusieurs paramètres passent par une valeur critique : les portraits de phase avant et après ne sont alors plus topologiquement équivalents[55]. Ce point critique est appelé point de bifurcation du système. L'ensemble de ces points constitue l'ensemble de bifurcation. Ce phénomène de bifurcation se rencontre dans de nombreux champs scientifiques où des systèmes qui ont des dynamiques propres, vont être modifiés par l'influence d'un paramètre extérieur, dit pour cette raison paramètre de contrôle du système. La physique des lasers en est un exemple d'application où l'on amplifie l'émission photonique jusqu'à la synchronisation complète des atomes émetteurs qui donnent une lumière cohérente. Plus généralement, les transitions de phases dans les systèmes physiques peuvent être assimilés à des phénomènes de bifurcation. De même en biologie, les bifurcations jouent un rôle important notamment dans les études de populations. On peut classer les bifurcations selon différents critères [Sprott, 2003] :

- Applications ou flots : les systèmes à temps continus et à temps discrets peuvent avoir des points de bifurcation, certains types de points de bifurcation sont spécifiques à l'un ou l'autre des systèmes
- La dimension : une bifurcation dans un système peut n'apparaître qu'à partir d'une certaine dimension ; il suffit dès lors d'étudier les bifurcations dans la plus petite dimension possible.
- La codimension : elle indique combien de paramètres vont varier en même temps. Nous ne parlerons ici que de bifurcations de codimension 1 qui sont les plus courantes et les plus simples à analyser.
- Le caractère local ou global : une bifurcation locale concerne des points particuliers dont la nature varie (points d'équilibre qui apparaissent ou disparaissent ou qui modifient leur stabilité), une bifurcation globale concerne toute l'orbite.
- Le caractère continu ou discontinu : une bifurcation continue est appelée sur-critique lorsque des valeurs propres de la matrice jacobienne deviennent stables ou instables, tandis qu'une bifurcation discontinue appelée sous-critique survient lorsque des valeurs propres apparaissent ou disparaissent.

A chaque type de bifurcation est associée sa forme normale, qui est le système type minimal générant cette bifurcation. Dans ce qui suit nous passons en revue trois types de bifurcation de codimension 1 pour les applications itérées à une dimension. Nous étudions des systèmes de la forme :

$$x_{k+1} = f(x_k, \nu) \tag{A.11}$$

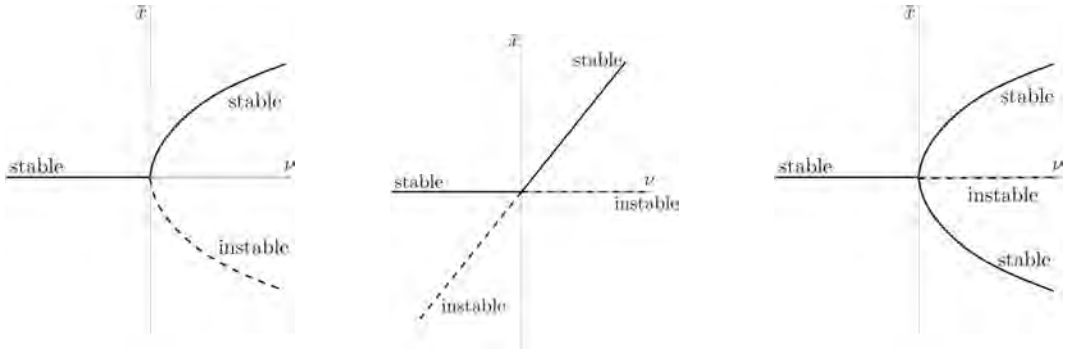
où ν est le paramètre de bifurcation (ou paramètre de contrôle) du système. Les formes normales présentées sont construites de façon à ce que la bifurcation se produise pour $\nu = 0$.

A.3.1 Bifurcation noeud-col

La bifurcation noeud-col occasionne la suppression ou la création de points fixes pour l'application. La forme normale pour cette bifurcation est la suivante :

$$f(x) = \nu + x - x^2 \tag{A.12}$$

[55] L'équivalence topologique signifie qu'il existe un homémorphisme (application continue et d'inverse continue) permettant de passer de façon continue de l'un à l'autre et inversement



(a) Bifurcation noeud-col

(b) Bifurcation trans-critique

(c) Bifurcation "flip" (doublement de période)

FIG. A.2 – Les trois principaux types de bifurcation pour des applications en dimension 1.

Dans ce cas, un point fixe vérifie $f(x) = x$ ce qui se produit lorsque $x = \pm\sqrt{\nu}$ (les deux points fixes) et $\nu \geq 0$. Il y a donc apparition ici de deux points fixes $\bar{x} = \pm\sqrt{\nu}$ dès que $\nu \geq 0$ qui n'existent pas lorsque $\nu < 0$, avec un point fixe stable et l'autre instable puisque $f'(\pm\sqrt{\nu}) = 1 \pm \nu$ (voir la figure A.2(a)).
Une bifurcation de ce type se produit avec l'application logistique lors de l'apparition de la fenêtre de période 3 pour $a = 1 + \sqrt{8} \approx 3,83$.

A.3.2 Bifurcation trans-critique

Cette bifurcation se produit avec la forme normale suivante :

$$f(x) = (1 + \nu)x - x^2 \tag{A.13}$$

Ici les deux points fixes sont $\bar{x} = 0$ et $\bar{x} = \nu$ qui s'échangent leur caractère de stabilité quand $\nu = 0$ (voir la figure A.2(b)). Ce type de bifurcation apparaît avec l'application logistique lorsque son paramètre de contrôle $a = 1$.

A.3.3 Bifurcation par doublement de période

La bifurcation par doublement période est très présente dans le comportement de la fonction logistique. Ce type de bifurcation est aussi appelée bifurcation "flip" ou bifurcation sous-harmonique ou bifurcation fourche. La forme normale associée :

$$f(x) = -(1 + \nu)x + x^3 \tag{A.14}$$

Il n'y a qu'un seul point fixe $\bar{x} = 0$ qui est stable pour $\nu < 0$ et instable pour $\nu > 0$. Cependant la deuxième itération $f^2(x)$ fait apparaître deux nouveaux points fixes stables $\bar{x} \simeq \pm\sqrt{\nu}$ qui constituent un $2-$cycle stable qui disparaît lorsque $\nu \to 0$ (voir la figure A.2(c)). Cette version de la bifurcation par doublement de période est sur-critique, la version sous-critique est la suivante : $f(x) = -(1 + \nu)x - x^3$
La seule différence est que le 2-cycle n'existe que pour $\nu < 0$ et est instable.

B

Quelques propriétés complémentaires sur l'application logistique

B.0.4 Propriétés statistiques de l'application logistique pour $r = 4$

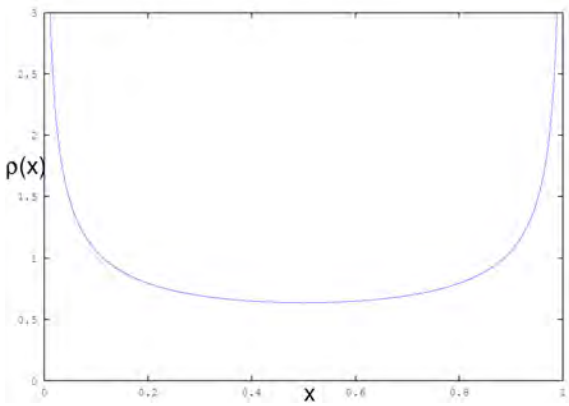


FIG. B.1 – Densité de probabilité pour l'application logistique en $r = 4$.

Nous étudions ici les propriétés statistiques de l'application logistique $x_{n+1} = f(x_n) = r\ x_n(1 - x_n)$ dans le cas chaotique $r = 4$.

Ce cas est en effet d'une grande importance car l'application y possède la propriété d'ergodicité, ou transitivité topologique. Sans entrer dans le détail technique, car la théorie ergodique dépasse le cadre de cette thèse, nous retiendrons les deux propriétés principales suivantes (voir par exemple [Walters, 1982]) :

- presque toute orbite est dense dans $[0, 1]$, c'est-à-dire que l'on peut approcher aussi près que l'on veut tout point de $[0, 1]$ par presque toute orbite de l'application quelque soit sa condition initiale. Autrement dit, une orbite quelqonque de f pour $r = 4$ visite tout l'intervalle $[0, 1]$.
- la moyenne temporelle pour un système ergodique de toute fonction intégrable est égale à

sa moyenne spatiale sur l'espace des phases (dit théorème de Birkhoff[56]).
Le premier point précise que "presque toute" orbite est dense, car il existe en réalité une infinité d'orbites périodiques instables (qui correspondent entre autres aux nombreux points et cycles rencontrés lorsque $r \neq 4$) qui représentent un ensemble de mesure nulle, i.e. la probabilité de démarrer aléatoirement sur une orbite périodique est nulle.
Le second point est plus délicat à expliciter : soit φ une fonction intégrable sur $[0, 1]$, si f est une application ergodique sur $[0, 1]$, alors on a l'égalité :

$$\lim_{n\to\infty} \frac{1}{n} \sum_{k=1}^{n} \varphi(f^k(x)) = \int_0^1 \varphi(x)dx \tag{B.1}$$

La caractéristique également connue est la densité de probabilité de l'application logistique pour $r = 4$. Elle peut être obtenue analytiquement en opérant le changement de variables suivant :

$$x_n = \frac{1}{2}(1 - \cos(2\pi y_n)) \tag{B.2}$$

qui conduit à l'application de décalage de Bernouilli[57] (voir les détails de la démonstration dans [Schuster et Just, 2005b]) :

$$y_{n+1} = 2y_n \mod 1 \tag{B.3}$$

Enfin on trouve la densité de probabilité suivante :

$$\rho(x) = \frac{1}{\pi} \frac{1}{\sqrt{x(1-x)}} \tag{B.4}$$

$$\text{et} \int_0^1 \rho(x)dx = 1 \tag{B.5}$$

Cette densité n'est pas uniforme (voir sa représentation graphique sur la figure B.1) d'une part et les séries chaotiques générées sont fortement corrélées d'autre part, ce qui interdit l'usage de l'application logistique $r = 4$ comme générateur de nombres aléatoires. Cependant, des voies existent pour faire de cette application une source "naturelle" de nombres aléatoires moyennant quelques transformations [Phatak et Rao, 1995]. Pour d'autres valeurs de r, aucune formulation systématique n'existe pour la dednsité de probabilité.

Pour finir cette revue des propriétés de l'application logistique pour $r = 4$, notons les précautions à prendre pour en réaliser le calcul des itérées. Nous avons vu en effet que une infinité d'orbites périodiques instables existent. Mais on peut également parler des "points fixes éventuels" [Sprott,] pour lesquels il n'y a plus d'évolution après un certain nombre d'itérations. Notamment si l'on atteint $x_k = 0,5$ pour une itération k donnée, alors $x_{k+1} = 1$ et $x_{k+1+p} = 0, \forall p > 0$. Tous les antécédents de $0,5$ mènent au même point final. Cela a des répercussions sur le calcul des itérées de l'application sur ordinateur : la précision finie du calcul des nombres décimaux sur ordinateur n'est pas sans conséquences. Supposons que les décimaux soient codés en simple précision (soit 32 bits de codage), cela correspond à une précision numérique de $\varepsilon \simeq 5 \cdot 10^{-8}$. Donc tout nombre x vérifiant $0,5 - \varepsilon < x < 0,5 + \varepsilon$ est potentiellement arrondi à $0,5$ et conduit à 0 deux itérations plus loin ! Concrètement, la probabilité pour qu'une itération d'une orbite

[56] du mathématicien américain du même nom (1884-1944)
[57] ou également application de décalage binaire

quelconque soit dans cet intervalle est de l'ordre de $\sqrt{\varepsilon}$. Ce qui donne un nombre d'itérations moyen de $\frac{1}{\sqrt{\varepsilon}} \simeq 4000$ pour que n'importe quelle condition initiale conduise à ... 0 ! Il est donc nécessaire d'augmenter la précision de codage des décimaux : en double précision, le nombre moyen d'itérations avant de rencontrer le problème monte à 40 millions d'itérations ce qui est déjà mieux mais pas forcément suffisant. Nous procédons dans cette thèse avec des décimaux codés sur 128 bits, donc en quadruple précision, ce qui garantit environ $4 \cdot 10^{31}$ itérations avant de converger sur 0.

B.0.5 Universalité de l'application logistique

Comme on a pu le constater dans les sections précédentes, la complexité des séquences numériques générées par les itérations d'un simple polynôme de degré 2 est surprenante au vu de la simplicité de l'opération effectuée. Il s'avère que ce type de comportement dynamique est très générique.

Famille des applications unimodales

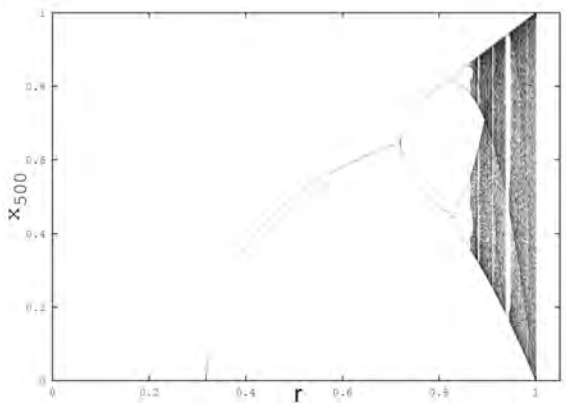


FIG. B.2 – Diagramme de bifurcation de l'application itérée $x_{n+1} = r \sin(\pi\ x_n)$.

En effet, un ensemble de résultats théoriques résumés dans [Collet et Eckmann, 1980] montre que toute application S-unimodale possède qualitativement le même type de dynamique (même type de diagramme de bifurcation et notamment de route vers le chaos par doublement de période). L'application logistique est le prototype de la famille des applications S-unimodales, ainsi que les applications quadratiques de la forme $x_{n+1} = 1 - \mu x_n^2$ où $\mu \in [0, 2]$.
Une application S-unimodale f de dimension 1 de $[0, 1]$ sur $[0, 1]$ est une application continue telle que ([Collet et Eckmann, 1980]) :

- $f(0) = f(1) = 0$
- f a un unique extremum c dans $]0, 1[$ (donc f est croissante sur $[0, c)$ et décroissante sur $(c, 1])$
- f est dérivable sur $[0, 1]$ excepté éventuellement en c
- f est C^3 : sa dérivée triple f''' existe et est continue

– la dérivée de Schwartz de f est négative $Sf < 0$, elle est donnée par l'opérateur suivant :

$$Sf = \frac{f'''}{f'} - \frac{3}{2}(\frac{f''}{f'})^2$$

Comme illustration de ce résultat, l'application sinusoïdale définie par la récurrence :

$$x_{n+1} = r\sin(\pi\ x_n)$$

montre un diagramme de bifurcation étonnamment proche de celui de l'application logistique (voir la figure B.2). Les deux dynamiques coïncident sur l'aspect qualitatif de façon remarquable : même diagramme, même fenêtres de périodicité qui surviennent dans le même ordre, même route vers le chaos par doublement de période,... Quantitativement cependant les deux applications diffèrent sur tous ces aspects avec des valeurs différentes du paramètre pour lesquels ils surviennent. En résumé, la forme algébrique des applications unimodales a peu d'inluence sur la dynamique qualitative que ces applications engendrent.

Formes conjuguées de l'application logistique

La notion de conjugaison est la suivante [Dang-Vu et Delcarte, b] :

Définition B.0.1. *On dira que les applications f et g sont conjuguées s'il existe une fonction ϕ bijective continue d'inverse continue (ou homéorphisme) telle que :*

$$\phi \circ f = g \circ \phi \ ouf = \phi^{-1} \circ g \circ \phi \tag{B.6}$$

Par exemple, tout polynôme quadratique de la forme $f(x) = ax^2 + bx + d, a \neq 0$ est conjugué avec $g(y) = y^2 + c$ par l'application $\phi(x) = \alpha x + \beta$ avec $\alpha = a$, $\beta = \frac{b}{2}$ et $c = \beta - \beta^2$, ce qui s'applique à l'application logistique. En fait, une conjugaison opère un changement de variables pour passer d'une application à une autre, tout en conservant les propriétés qualitatives de sa dynamique. Ainsi, un point fixe ou périodique pour f donne un point fixe ou périodique pour g par la transformation ϕ. Si ϕ est de plus différentiable, la nature attractive ou répulsive de l'attracteur est également conservée. Les applications quadratiques de la forme $g(y) = b - y^2$ sont aussi conjuguées avec l'application logistique. Nous utilisons dans cette thèse l'application logistique ou des versions conjuguées de cette application.

Constantes de Feigenbaum et renormalisation

Nous avons mis à jour une des deux constantes de Feigenbaum —δ— associées à la cascade de doublement de période liée à la régularité d'apparition des points de bifurcation, basée sur une mesure des différences de ces points en abscisse selon le paramètre r. Il en existe une autre liée aux distances des cycles 2-périodiques en ordonnée relativement à la valeur $0,5$: on trace sur le diagramme de bifurcation 5.6(f) la droite horizontale $x = 0,5$ qui coupe les branches de doublement de période en des points d'intersection successifs d'abscisses $R_k, k = 1, 2, \cdots$. En chacune de ces abscisses, on mesure d_k correspondant à la distance entre le point d'intersection à

l'abscisse R_k précédent et le point de même abscisse sur la branche de bifurcation la plus proche. On établit ainsi la seconde constante de Feigenbaum (ou le "rapport d'échelles") :

$$\alpha = -\lim_{k\to\infty} \frac{d_k}{d_{k+1}} = 2,502907\cdots \tag{B.7}$$

Toutes les applications S-unimodales définies précédemment présentent les mêmes valeurs pour ces constantes, ce qui renforcent leur aspect universel. De plus, leur mise en évidence dans des phénomènes physiques de transition de phase du second ordre a été obtenue en magnétisme et mécanique des fluides notamment.

Enfin, nous ne ferons qu'évoquer la théorie de renormalisation pour le doublement de période établi par [Feigenbaum, 1980]. Puisqu'il existe des constantes universelles pour la cascade de doublement de période, l'idée de renormaliser revient à rechercher la fonction universelle $g(x)$ obtenue en ramenant l'échelle α à 1 pour chaque doublement de période. La fonction $g(x)$ obéit à l'équation fonctionnelle suivante [Feigenbaum, 1980, Collet et Eckmann, 1980] (on pourra également trouver une présentation dans [Dang-Vu et Delcarte, b]) :

$$g(x) = Tg(x) = -\alpha g\left[g(-\frac{x}{\alpha})\right] \tag{B.8}$$

$$g(0) = 1 \tag{B.9}$$

C'est une fonction de x^2 dont on peut calculer un développement limité en 0. Cette théorie de renormalisation confirme et conclue l'universalité des phénomènes dynamiques liés à l'application logistique.

C

Réseaux de neurones chaotiques

C.1 Les prémisses biologiques

Cette annexe vise à survoler les faits biologiques observés dans des réseaux de neurones biologiques, faits qui ont permis la mise en évidence de phénomènes chaotiques dans ces réseaux. Ces phénomènes sont désormais une réalité biologique qu'il faut prendre en compte et dont l'existence même et les mécanismes interrogent les biologistes : c'est le constat fait dans l'article [Korn et Faure, 2003] qui se propose de dresser un panorama des connaissances et des données disponibles sur cette question en 2003. Il ne s'agit pas de faire ici une revue exhaustive de ces phénomènes qui dépassent largement le cadre de cette thèse mais de montrer que la modélisation de la dynamique des états internes des agents logistiques que nous utilisons dans nos simulations peut aussi se justifier d'un point de vue biologique. Nous présentons ici quelques travaux importants qui ont jalonné l'historique de la mise en évidence et la modélisation des phénomènes chaotiques dans les réseaux de neurones biologiques. Pour un historique et une présentation plus complets, l'article [Korn et Faure, 2003] fait une revue de ce domaine d'étude en 2003.

Les travaux pionniers. La première mise en évidence du chaos biologique est due à Bessler et Freemann en 1980 lorsqu'ils ont observés des électroencéphalogrammes de chats, rats et lapins stimulés par des odeurs [Bressler et Freeman, 1980]. Il s'agissait donc d'étudier la réponse à des stimulus olfactifs de certaines zones cérébrales spécialisées dans la détection des odeurs, notamment le bulbe olfactif très étudié par ailleurs. Dans [Bressler et Freeman, 1980], le recueil des données est analysée d'un point de vue biologique et la conclusion fut que les processus oscillatoires résultent de l'interaction synaptique entre groupes de neurones excitateurs et inhibiteurs. Ces courbes de l'activité cérébrale ont révélé par analyse spectrale une structure périodique et apériodique particulière et spécifique à cette fonction olfactive. Les conclusions au niveau de cet article n'évoquent pas encore la piste chaotique, c'est dans [Freeman, 1987] que l'analyse sera reconduite à l'aide d'une modélisation à base d'équations différentielles reproduisant le phénomène chaotique. Le phénomène est interprété comme provenant de boucles de rétroaction entre groupes de neurones excitateurs et inhibiteurs. La modélisation réalisée considère donc les interactions entre groupes de neurones tous inhibiteurs ou tous excitateurs ; ce changement d'échelle implique de considérer le groupe de neurones comme une boîte noire dont on modélise

le système dynamique d'entrée-sortie. La fonction sigmoïde joue toujours un rôle dans cette modélisation. La conclusion de ce travail de modélisation est la confirmation de la présence de phénomènes chaotiques révélés par la résolution numérique du système dynamique modélisé.

Freeman interprète le phénomène olfactif par l'association d'un attracteur chaotique à chaque odeur dont la reconnaissance est déjà apprise par le réseau. Si une nouvelle odeur se présente, le comportement du réseau est destructuré tant que l'apprentissage de cette nouvelle odeur n'est pas effectué, apprentissage qui a pour effet de "créer la route" vers un nouvel attracteur associé. Cet apprentissage s'opère par modification des poids synaptiques par un renforcement de type hebbien (on renforce les poids entre deux neurones s'ils sont activés en même temps) [Freeman, 1987] jusqu'à "stabiliser" la dynamique sur un attracteur. Un dernier aspect de l'interprétation de Freeman est le processus de "route vers le chaos" pour laquelle il invoque une route de type "Ruelle-Takens-Newhouse". Il s'agit pour résumer d'une route vers le chaos gouvernée par l'évolution d'un paramètre de contrôle k : lorsque $k = 0$, le réseau de neurones adopte un comportement de type point fixe, puis lorsque k augmente sous l'effet du stimulus perçu, le système converge vers un cycle limite par une "bifurcation de Hopf" puis vers un tore, enfin lorsque k continue d'augmenter, le système converge vers un attracteur chaotique. Cette interprétation a de fortes implications en termes de modélisation. Elle est corroborée par les travaux que nous évoquons maintenant et qui présentent un autre type de "route vers le chaos" dans les réseaux de neurones.

Fig. C.1 – Etude d'un réseau de neurones biologiques de la rétine de salamandre d'après [Crevier et Meister, 1998]. En haut les données biologiques et en bas la réponse du système modélisé par circuit électrique.

Doublement de période dans le cortex visuel. Crevier et Meister relatent dans [Crevier et Meister, 1998] leur étude du système visuel chez la salamandre par l'effet d'impulsions lumineuses sur les cellules de la rétine. Ils étendent ensuite leurs conclusions au cas du système visuel humain. L'expérience menée consiste à stimuler les cellules rétinales par des flashs lumineux de fréquence croissante et à mesurer la réponse grâce à des électrodes implantées à la base du nerf optique. A faible fréquence, le signal de sortie du réseau de cellules neuronales suit fidèlement la fréquence imposée, mais à mesure que la fréquence des flashs augmente, la réponse en fréquence n'est pas linéaire : les cellules rétinales ne suivent plus la fréquence des flashs et ne transmettent plus qu'une portion des flashs produits. Cette dernière observation a également été effectuée sur le système visuel humain et permet d'aborder certains aspects du déclenchement des crises d'épilepsie chez les personnes atteintes de ce type de troubles. Plus précisément les auteurs de l'expérimentation montrent que cette réponse suit une loi non linéaire, comportant des sauts qualitatifs relatifs à des bifurcations dans le comportement. Les résultats sont présentés sur la figure C.1 où l'on peut voir le tracé point par point des résultats expérimentaux. Deux séries de mesures ont été effectuées : d'une part, la mesure du potentiel de réponse en fonction de la fréquence des flashs à contraste lumineux constant ($C = 1.0$ et d'autre part la mesure du

potentiel de réponse en fonction de la valeur du contraste à fréquence constante (en l'occurrence $f = 16Hz$). Le premier constat sur ces résultats expérimentaux est tout d'abord que les cellules répondent de façon synchronisées aux flashs lumineux jusqu'à certaines fréquences ou contrastes, ce qui permet d'obtenir un signal facilement identifiable. Le second constat est que la réponse en fréquence bifurque par doublement de période jusqu'à passer dans un mode chaotique. Ainsi à fréquence lumineuse faible, le réseau de cellules répond en suivant la fréquence stimulatrice, puis à un certain seuil il ne répond plus qu'à un flash sur deux puis un flash sur quatre. Ensuite le réseau entre dans un mode chaotique. En fait le diagramme de bifurcation expérimental C.1 montre une route vers le chaos par doublements de périodes ce qui est un processus rencontré dans de nombreux autres phénomènes physiques (en optique non linéaire des lasers par exemple) ou mathématiques (l'application logistique présente ce même type de route vers le chaos). Dans ces diagrammes, c'est la fréquence des flashs ou le niveau de contraste qui joue le rôle de paramètre de contrôle du système et l'on mesure l'amplitude du potentiel électrique . Cette mise en évidence expérimentale est confortée par une modélisation par circuit électrique qui reproduit assez fidèlement les diagrammes de bifurcation C.1. Ce résultat est remarquable par sa précision et montre l'universalité des phénomènes de route vers le chaos par doublement de périodes.

Intérêt biologique du chaos. Une question essentielle réside dans l'utilité du chaos dans les réseaux de neurones. L'évidence du phénomène est maintenant établie, reste à savoir quelle est la fonction biologique de ce type de comportement ? Korn et Faure [Korn et Faure, 2003] résument les conceptions sur ce point. Tout d'abord, la vision "chaotique" du fonctionnement cérébral va à l'encontre des conceptions computationnalistes/symboliques de ce même fonctionnement, bien que la question reste ouverte à ce sujet. On l'a vu à propos de la reconnaissance des odeurs, un attracteur chaotique cumule deux caractéristiques : il est à la fois ordonné et instable, ordonné parcequ'il maintient le système dans une certaine région de l'espace des phases, et instable car sensible aux conditions initiales (en l'occurrence les entrées du réseau). Cette sensibilité est un atout dans le cas de la détection olfactive puisqu'elle permet de passer d'un attracteur à l'autre par transition de phase, chaque attracteur étant associé à une odeur. Une nouvelle odeur correspond à la construction d'un nouvel attracteur. Cette forme de codage de l'information —ici sur les odeurs— est de fait souple et réactive, mais constitue un problème observé mais non résolu.

Pour résumer et généraliser ce point de vue, la variabilité dans tout système biologique est nécessaire, jusqu'au bruit qui permet au système de se reconfigurer. Cette variabilité peut être générée par des phénomènes chaotiques. Ceux-ci sont modélisables par des modèles déterministes qui ont l'intérêt de proposer une interprétation biologique à chaque paramètre du système. De même le jeu du contrôle et des boucles de rétroaction devient plus lisible avec ce type de modèles.

C.2 La modélisation et l'analyse numérique

A la suite des résultats expérimentaux précédents qui désormais établissent clairement la présence de phénomènes chaotiques au sein de certains réseaux de neurones, notamment ceux spécialisés dans l'olfaction ou la vision, la question est de savoir si l'on retrouve de tels effets

avec les modèles formels de neurones et surtout si l'on peut prouver analytiquement l'existence de phénomènes chaotiques. En effet, de nombreuses expériences de simulations numériques ont montré l'apparition de tels phénomènes mais aucune véritable preuve mathématique ne fut établie pour comprendre l'origine de leur apparition.

Route vers le chaos dans un réseau de deux neurones. Les premières véritables avancées dans ce sens sont établies dans l'article [Sompolinsky *et al.*, 1988] qui montre qu'un réseau de neurones à temps continu et à matrice de connexion asymétrique aléatoire adopte un comportement chaotique lorsque le nombre de neurones tend vers l'infini. Ces conditions d'émergence étant assez drastiques, la question est de savoir si un réseau de dimension finie peut générer spontanément et sans influence extérieure d'aucune sorte (forçage d'une entrée, délais) un comportement chaotique. L'article [Wang, 1991] répond à cette question. Son auteur Xin Wang propose une preuve mathématique analytique de l'équivalence d'un réseau récurrent simple à deux neurones avec les applications non linéaires unimodales itérées. Plus précisément il montre dans cet article que les deux systèmes dynamiques correspondants sont topologiquement conjugués, ce qui revient à dire que l'on peut trouver un homéomorphisme (bijectif et continu) reliant les deux systèmes, ce qui implique que la dynamique qualitative des deux systèmes est semblable. Des simulations numériques sur le réseau à deux neurones montrent des diagrammes de bifurcation très similaires à celui d'une application logistique par exemple et surtout une route vers le chaos par doublement de période, typique de ce genre de système. La figure C.2 montre la topolo-

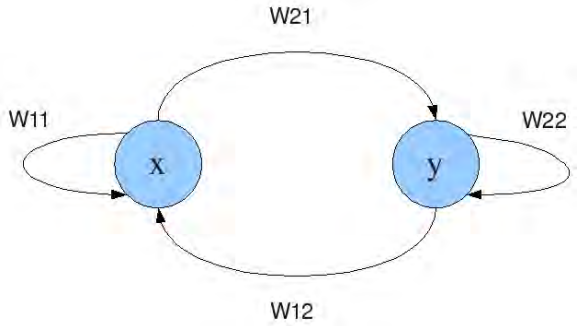


Fig. C.2 – Réseau à deux neurones de Xin Wang.

gie du réseau récurrent très simple considéré par Xin Wang. Le vecteur (x, y) représente l'état des neurones qui évolue dans le domaine $I^2 = [0, 1] \times [0, 1]$. La matrice de connexion est notée $W = [w_{ij}]$ et la fonction sigmoïde est définie par :

$$\sigma_\mu(z) = \frac{1}{1 + \exp^{-\mu\ z}}$$

Le système dynamique discret étudié est donc le suivant :

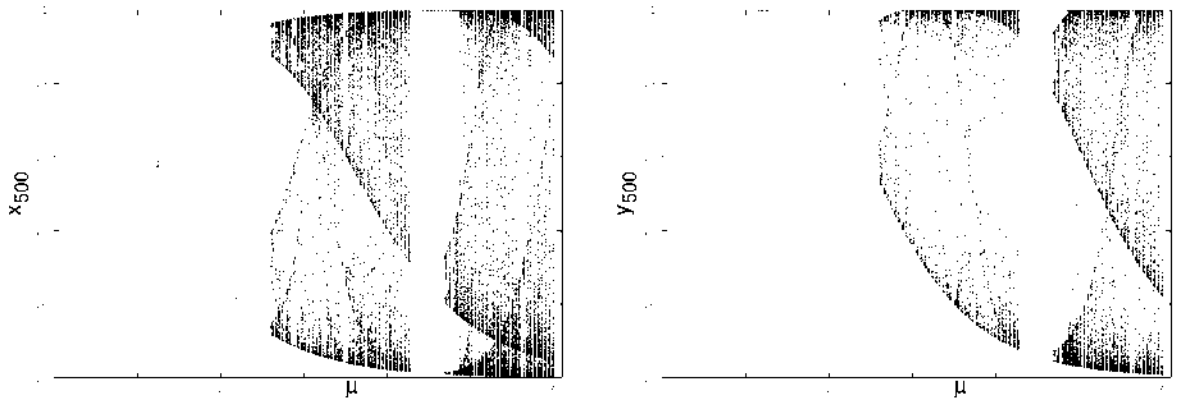
$$\begin{pmatrix} x(t+1) \\ y(t+1) \end{pmatrix} = \begin{pmatrix} \sigma_\mu(w_{11}x(t) + w_{12}y(t)) \\ \sigma_\mu(w_{21}x(t) + w_{22}y(t)) \end{pmatrix} \equiv F_\mu \begin{pmatrix} x(t) \\ y(t) \end{pmatrix} \tag{C.1}$$

La matrice de connexion considérée a la forme particulière asymétrique suivante :

$$W = \begin{bmatrix} a & ka \\ b & kb \end{bmatrix} \tag{C.2}$$

Wang démontre que ce réseau neuronal écrit comme un système dynamique discret à 2 dimensions est topologiquement conjugué à une application non linéaire itérative de dimension 1 S-unimodale. L'application logistique par exemple appartient à cette famille d'applications S-unimodale. Les définitions relatives aux concepts d'application "S-unimodale" et de conjugaison topologique ont été données en annexe B.0.5. Cette propriété de conjugaison nous assure de la similarité des dynamiques des deux types de systèmes. Autrement dit, ce résultat signifie qu'une application comme l'application logistique est très proche dynamiquement du réseau à deux neurones de Wang.

Fort de ce résultat, X. Wang vérifie par simulation numérique la pertinence de ce résultat. Nous reproduisons ici les diagrammes de bifurcation pour les valeurs suivantes de la matrice $W = \begin{bmatrix} -5 & 5 \\ -25 & 25 \end{bmatrix}$. Le réseau utilisé est donc constitué de deux neurones dont l'un joue un rôle inhibiteur et l'autre excitateur. D'autre part c'est le paramètre μ qui joue le rôle de paramètre de contrôle du système, à savoir le gain de la sigmoïde. Il est à noter ici que la dynamique proposée



(a) Bifurcations sur x (b) Bifurcations sur y

Fig. C.3 – Diagrammes de bifurcation des deux états (x, y) des neurones en fonction du paramètre de contrôle μ.

est à temps discret, ce qui théoriquement autorise l'apparition de phénomènes chaotiques à très faible dimension d'espace d'état (à partir de la dimension 1), tandis qu'il ne peut exister à temps continu qu'à partir de la dimension 3. Cela permet de simplifier également le modèle dynamique comme dans le système C.1.

Les approches dynamiques de la cognition. D'autres travaux sont menés dans la lignée de l'article [Sompolinsky *et al.*, 1988] avec des réseaux de neurones récurrents de taille arbitraire mais dont les poids de connexions sont générés aléatoirement selon des lois gaussiennes pour rendre la matrice de connexions non symétrique, condition nécessaire pour l'apparition

de phénomènes non périodiques. Les travaux de [Doyon *et al.*, 1992, Daucé *et al.*, 2001] notamment étudient l'effet d'un apprentissage hebbien sur le diagramme de bifurcation du réseau où là encore le paramètre de contrôle est le gain de la sigmoïde.

Le courant de recherche consistant en l'approche dynamique de la cognition se donne comme point de mire la compréhension sous l'angle dynamique des réseaux de neurones aussi complexes soient-ils [Guillot et Daucé, 2002]. L'idée fondatrice sous-jacente est que l'état d'un réseau de neurone se trouve entre ordre et désordre et qu'il peut être caractérisé par un attracteur éventuellement chaotique très sensible aux variations des paramètres du réseau. L'apprentissage d'un tel réseau peut également être interprété comme la trajectoire vers un tel attracteur dans un espace de grande dimension.

Les modèles de masses de neurones. L'idée de considérer l'activité de groupes entiers de neurones a donné lieu aux modèles dits de masses neuronales[58] (cf. [Lopes da Silva *et al.*, 1976, Jansen et Rit, 1995]). Ces masses de neurones interagissent à un niveau global de description vis-à-vis du neurone lui-même, et peuvent avoir des effets entièrement inhibiteurs ou excitateurs. Ces modèles sont décrits par des systèmes d'équations différentielles non linéaires dont les diagrammes de bifurcation calculés numériquement renseignent sur la dynamique de ces systèmes. Nous ne poursuivrons pas davantage sur ce domaine de modélisation qui sort du cadre de cette thèse.

[58]"Neural mass"

Bibliographie

[App,] Applied soft computing.

[TSP, 1995] (1995). Tsplib.

[Andrea *et al.*, 2001] Andrea, C. B., Roli, A. et Dorigo, M. (2001). Hc-aco : The hypercube framework for ant colony optimization. *In IEEE Transactions on Systems, Man, and Cybernetics—Part B*, pages 399–403.

[Arnold, 1988] Arnold, V. I. (1988). *Equations différentielles ordinaires.* Editions Mir, 4 édition.

[Arnold *et al.*, 1999] Arnold, V. I., Afrajmovich, V. S., Il'yashenko, Y. S. et Shil'nikov, L. P. (1999). *Bifurcation Theory and Catastrophe Theory.* Springer, 2 édition.

[Ashby, 1947] Ashby, W. R. (1947). Principles of the self-organizing dynamic system. *Journal of General Psychology*, 37:125–128.

[Ashby, 1957] Ashby, W. R. (1957). *An Introduction to Cybernetics.* Chapman & Hall, London, 2 édition.

[Ashby, 1962] Ashby, W. R. (1962). *Principles of the Self-Organizing Systems*, pages 255–278. Principles of Self-Organization. Pergamon Press, New York.

[Atlan, 2006] Atlan, H. (2006). *L'organisation biologique et la théorie de l'information.* La librairie du XXIème siècle. Seuil, 3 édition.

[Ballerini *et al.*, 2008] Ballerini, M., Cabibbo, N., Candelier, R., Cavagna, A., Cisbani, E., Giardina, I., Lecomte, V., Orlandi, A., Parisi, G., Procaccini, A., Viale, M. et Zdravkovic, V. (2008). Interaction ruling animal collective behavior depends on topological rather than metric distance : Evidence from a field study. *Proceedings of the National Academy of Sciences*, 105(4):1232–1237.

[Bandini *et al.*, 2002] Bandini, S., Manzoni, S. et Simone, C. (2002). Dealing with space in multi–agent systems : a model for situated mas. *In AAMAS '02 : Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, pages 1183–1190, New York, NY, USA. ACM Press.

[Beckers *et al.*, 1990] Beckers, R., Deneubourg, J., Goss, S. et Pasteels, J. (1990). Collective decision making through food recruitment. *Insectes sociaux*, 37(3):258–267.

[Beekman *et al.*, 2008] Beekman, M., Sword, G. A. et Simpson, S. J. (2008). Biological foundations of swarm intelligence. *In* Blum, C. et Merkle, D., éditeurs : *Swarm Intelligence*, Natural Computing Series, pages 2–41. Springer-Verlag.

[Benda *et al.*, 1986] Benda, M., Jagannathan, V. et Dodhiawala, R. (1986). On optimal cooperation of knowledge sources - an empirical investigation. Rapport technique, Boeing Advanced Technology Center.

[Beni, 2005] Beni, G. (2005). From swarm intelligence to swarm robotics. *In* Beni, G. et Payton, D., éditeurs : *Swarm Robotics*, volume 3342 de *Lecture Notes in Computer Science*, pages 1–9. Springer Berlin Heidelberg.

[Beni et Wang, 1989] Beni, G. et Wang, J. (1989). Swarm intelligence in cellular robotics systems. *In Proceeding of NATO Advanced Workshop on Robots and Biological System*, pages 26–30, Tuscany, Italy.

[Bennett, 1986] Bennett, C. (1986). On the nature and origin of complexity in discrete, homogenous, locally-interacting systems. *Foundations of Physics*, 16:585–592.

[Berthiau et Siarry, 2001] Berthiau, G. et Siarry, P. (2001). État de l'art des méthodes d'"<optimisation globale">. *RAIRO - Operations Research - Recherche Opérationnelle*, 35(3):329–365.

[Berthold et Querner, 1981] Berthold, P. et Querner, U. (1981). Genetic basis of migratory behaviour in european warblers. *Science*, 212:77–79.

[Beyer et Schwefel, 2002] Beyer, H.-G. et Schwefel, H.-P. (2002). Evolution strategies – a comprehensive introduction. *Natural Computing*, 1(1).

[Birattari *et al.*, 2002] Birattari, M., Caro, G. D. et Dorigo, M. (2002). Toward the formal foundation of ant programming. *Lecture Notes in Computer Science*, 2463:188.

[Blum et Merkle, 2008] Blum, C. et Merkle, D., éditeurs (2008). *Swarm Intelligence*. Natural Computing Series. Springer-Verlag.

[Boccara, 2004] Boccara, N. (2004). *Modeling Complex Systems*. Springer.

[Bonabeau *et al.*, 1999a] Bonabeau, E., Dorigo, M. et Theraulaz, G. (1999a). *Swarm Intelligence. From Natural to Artificial Systems*, chapitre 1, pages 7–8. Studies in the Sciences of Complexity. Oxford University Press, New York Oxford.

[Bonabeau *et al.*, 1999b] Bonabeau, E., Dorigo, M. et Theraulaz, G. (1999b). *Swarm Intelligence. From Natural to Artificial Systems*, chapitre 1, pages 26–30. Studies in the Sciences of Complexity. Oxford University Press, New York Oxford.

[Bonabeau *et al.*, 1999c] Bonabeau, E., Dorigo, M. et Theraulaz, G. (1999c). *Swarm Intelligence. From natural to artificial systems*. Springer-Verlag, Berlin Heidelberg New York.

[Bonabeau et Theraulaz, 1994a] Bonabeau, E. et Theraulaz, G. (1994a). *Intelligence collective*. Systèmes Complexes. Hermès, Paris.

[Bonabeau et Theraulaz, 1994b] Bonabeau, E. et Theraulaz, G. (1994b). *Le recrutement et l'exploitation collective de sources de nourriture*, pages 62–86. Systèmes Complexes. Hermès, Paris.

[Booker *et al.*, 2005] Booker, L., Forrest, S., Mitchell, M. et Riolo, R. (2005). Introduction. *In* Booker, L., Forrest, S., Mitchell, M. et Riolo, R., éditeurs : *Perspectives on Adaptation in Natural and Artificial Systems*, Santa Fe Institute Studies in the Sciences of Complexity, pages 1–10. Oxford University Press.

[Boumaza et Scherrer, 2007] Boumaza, A. et Scherrer, B. (2007). Convergence and rate of convergence of a simple ant model. *In International Joint Conference on Autonomous Agent and Multiagent Systems - AAMAS' 2007*.

[Bourgine *et al.*, 2008] Bourgine, P., Chavalarias, D. et Cohen-Boulakia, C., éditeurs (2008). *Déterminismes et Complexités : du Physique à l'Ethique -Autour d'Henri Atlan*. Colloque De Cerisy. La Découverte.

[Bourjot *et al.*, 2003] Bourjot, C., Chevrier, V. et Thomas, V. (2003). A new swarm mechanism based on social spiders colonies : from web weaving to region detection. *Web Intelligence and Agent Systems*, 1:47–64.

[Bressler et Freeman, 1980] Bressler, S. L. et Freeman, W. J. (1980). Frequency analysis of olfactory system eeg in cat, rabbit and rat. *Electroenceph. Clin. Neurophysiol.*, 50:19–24.

[Bricmont, 1995] Bricmont, J. (1995). Science of chaos or chaos in science ? *Physicalia Magazine*, 17(3-4):159–208.

[Briot et Demazeau, 2001] Briot, J.-P. et Demazeau, Y., éditeurs (2001). *Principes et architecture des systèmes multi-agents*. Lavoisier.

[Bullnheimer *et al.*, 1999] Bullnheimer, B., Hartl, R. et Strauss, C. (1999). A new rank based version of the ant system — a computational study. *Central European Journal for Operations Research and Economics.*, 7(1):25–38.

[Camazine *et al.*, a] Camazine, S., Deneubourg, J., Franks, N. R., Sneyd, J., Theraulaz, G. et Bonabeau, E. *Characteristics of Self-Organizing Systems*, chapitre 3. Princeton University Press.

[Camazine *et al.*, b] Camazine, S., Deneubourg, J., Franks, N. R., Sneyd, J., Theraulaz, G. et Bonabeau, E. *How Self-Organization Works*, chapitre 2. Princeton University Press.

[Camazine *et al.*, 2001] Camazine, S., Deneubourg, J.-L., Franks, N. R., Sneyd, J., Theraulaz, G. et Bonabeau, E. (2001). *Self-Organization in Biological Systems*. Princeton studies in complexity. Princeton University Press.

[Castellani et Hafferty, 2009] Castellani, B. et Hafferty, F. W. (2009). *Sociology and Complexity Science*. Springer.

[Charrier *et al.*, 2007a] Charrier, R., Bourjot, C. et Charpillet, F. (2007a). Flocking as a synchronization phenomenon with logistic agents. *In ECCS'07, European Conference on Complex Systems, Dresden*, Dresden.

[Charrier *et al.*, 2007b] Charrier, R., Bourjot, C. et Charpillet, F. (2007b). A nonlinear multi-agent system designed for swarm intelligence : the logistic MAS. *In SASO'07, International Conference on Self-Adaptive and Self-Organizing Systems, Boston*.

[Charrier *et al.*, 2009a] Charrier, R., Bourjot, C. et Charpillet, F. (2009a). Modeling a swarm phenomenon using logistic agents : Application to a predators-prey pursuit. *In AAMAS'09 Eight International Conference on Autonomous Agents and Multi-agent Systems, Budapest*.

[Charrier *et al.*, 2009b] Charrier, R., Bourjot, C. et Charpillet, F. (2009b). Study of self-adaptation mechanisms in a swarm of logistic agents. *In SASO 2009, Third IEEE International Conference on Self-Adaptive and Self-Organizing Systems, San Francisco*.

[Charrier *et al.*, 2007c] Charrier, R., Charpillet, F. et Bourjot, C. (2007c). Deterministic nonlinear modeling of ant algorithm with a logistic multi-agent system. *In AAMAS '07, International Conference on Autonomous Agents and Multiagent Systems,Hawaii, 14-18 mai.*

[Charrier *et al.*, 2007d] Charrier, R., Charpillet, F. et Bourjot, C. (2007d). Un modèle connexionniste pour l'intelligence en essaim : le système multi-agent logistique. *In ARCO'07, Association pour la Recherche Cognitive, Nancy.*

[Clerc, 2005] Clerc, M. (2005). *L'optimisation par essaims particulaires.* Hermes Sciences, lavoisier édition.

[Clerc et Siarry, 2004] Clerc, M. et Siarry, P. (2004). Une nouvelle métaheuristique pour l'optimisation difficile : la méthode des essaims particulaires. Séminaire OEP'2003.

[Collet et Eckmann, 1980] Collet, P. et Eckmann, J.-P. (1980). *Iterated Maps on the Interval as Dynamical System.* Birkhäuser Boston.

[Cosenza et Gonzalez, 1998] Cosenza, M. G. et Gonzalez, J. (1998). Synchronization and collective behavior in globally coupled logarithmic maps. *Progress of Theoretical Physics*, 100(1):21–38.

[Crevier et Meister, 1998] Crevier, D. W. et Meister, M. (1998). Synchronous period-doubling in flicker vision of salamander and man. *Journal of Neurophysiology*, 79:1869–1878.

[Cucker et Smale, 2007] Cucker, F. et Smale, S. (2007). Emergent behavior in flocks. *IEEE transactions on automatic control*, 52(5):852–862.

[Czirók *et al.*, 1999] Czirók, A., Barabási, A.-L. et Vicsek, T. (1999). Collective motion of self-propelled particles : Kinetic phase transition in one dimension. *Physical Review Letters*, 82(1).

[Dang-Vu et Delcarte, a] Dang-Vu, H. et Delcarte, C. *Mesures et structures du chaos*, chapitre 4. Ellipses.

[Dang-Vu et Delcarte, b] Dang-Vu, H. et Delcarte, C. *Mesures et structures du chaos*, chapitre 3. Ellipses.

[Dang-Vu et Delcarte, 2000] Dang-Vu, H. et Delcarte, C. (2000). *Bifurcations et chaos.* Universités. Ellipses. Introduction à la dynamique contemporaine avec des programmes en Pascal, Fortan et Mathématica.

[Daucé *et al.*, 2001] Daucé, E., Moynot, O., Pinaud, O. et Samuelides, M. (2001). Mean-field theory and synchronization in random recurrent neural networks. *Neural Processing Letters*, 14:115–126.

[De Jong, 1975] De Jong, K. (1975). *An analysis of the behaviour of a class of genetic adaptive systems.* Thèse de doctorat, University of Michigan.

[Delahaye, 1999] Delahaye, J.-P. (1999). *Information complexité et hasard*, chapitre 4. Hermes Sciences Publications.

[Demazeau, 1997] Demazeau, Y. (1997). Steps towards multi-agent oriented programming. *In 1st International Workshop on Multi-Agent Systems (IWMAS'97).*

[Deneubourg *et al.*, 1990] Deneubourg, J., Aron, S., Goss, S. et Pasteels, J. (1990). The self-organizing exploratory pattern of the argentine ant. *Insect Behavior*, 3:159–168.

[Dorigo, 1992] Dorigo, M. (1992). *Optimization, Learning, and Natural Algorithms (in Italian)*. Thèse de doctorat, Politecnico di Milano.

[Dorigo et Gambardella, 1997] Dorigo, M. et Gambardella, L. M. (1997). Ant colony system : A cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, 1:53–66.

[Dorigo *et al.*, 1996] Dorigo, M., Maniezzo, V. et Colorni, A. (1996). The ant system : optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics*, 26(1):1–13.

[Dorigo et Sahin, 2004] Dorigo, M. et Sahin, E. (2004). Swarm robotics. *Autonomous Robots*, 17:111–113.

[Dorigo et Stützle, 2004a] Dorigo, M. et Stützle, T. (2004a). *Ant Colony Optimization*. The MIT Press.

[Dorigo et Stützle, 2004b] Dorigo, M. et Stützle, T. (2004b). *Ant Colony Optimization Algorithms for the Travelling Salesman Problem*, pages 65–117. The MIT Press.

[Dorigo *et al.*, 2005] Dorigo, M., Tuci, E., Gross, R., Trianni, V., Labella, T. H., Halva, T., Nouyan, S., Ampatzis, C., louis Deneubourg, J., Baldassarre, G., Francesco, Nolfi, S., Mondada, F., Floreano, D. et Gambardella, L. M. (2005). The swarm-bot project. *In Kunstliche Intelligenz*, pages 31–44. Springer Verlag.

[Doyon *et al.*, 1992] Doyon, B., Cessac, B., Quoy, M. et Samuelides, M. (1992). Control of the transition to chaos in neural networks with random connectivity. *International Journal of Bifurcation and Chaos*, 3:279–291.

[Dréo *et al.*, 2003] Dréo, J., Pétrowski, A., Siarry, P. et Taillard, E. (2003). *Métaheuristiques pour l'optimisation difficile*. Eyrolles.

[Dyer *et al.*, 2008] Dyer, J. R., Ioannou, C. C., Morrell, L. J., Croft, D. P., Couzin, I. D., Waters, D. A. et Krause, J. (2008). Consensus decision making in human crowds. *Animal Behaviour*, 75:461–470.

[Ebeling et Schweitzer, 2003] Ebeling, W. et Schweitzer, F. (2003). Self-organization, active brownian dynamics, and biological applications. *NOVA ACTA LEOPOLDINA NF*, 88:169.

[Ebeling *et al.*, 1999] Ebeling, W., Schweitzer, F. et Tilch, B. (1999). Active brownian particles with energy depots modelling animal mobility. *BioSystems*, 49:17–29.

[Farmer et d'A. Belin, 1990] Farmer, J. D. et d'A. Belin, A. (1990). Artificial life : The coming evolution. Santa Fe Institute.

[Feigenbaum, 1978] Feigenbaum, M. J. (1978). Quantitative universality for a class of nonlinear transformations. *Journal of Statistical Physics*, 19:24–52.

[Feigenbaum, 1980] Feigenbaum, M. J. (1980). Universal behavior in nonlinear systems. *Los Alamos Science*, 1:4–27.

[Ferber, 1995] Ferber, J. (1995). *Les systèmes multi-agents*. InterEditions.

[Ferber, 1997] Ferber, J. (1997). Les systèmes multi-agents : un aperçu général. *Technique et science informatiques*, 16(8):979–1012.

[Ferber et Müller, 1996] Ferber, J. et Müller, J.-P. (1996). Influences and reactions : A model of situated multiagent systems. *In* M., T., éditeur : *Proceedings of 2nd International Conference on Multiagent Systems (ICMAS-96)*, pages 72–79. The AAAI Press.

[Ferrand, 1997] Ferrand, N. (1997). *Modèles pour un outil d'aide à la décision et la négociation en aménagement du territoire - approche multi-agents.* Thèse de doctorat, Université Joseph Fourier, Grenoble.

[Ferrand *et al.*, 1994] Ferrand, N., Labbi, A., Giacometti, A., Amy, B. et Demazeau, Y. (1994). Entre systèmes multi-agents réactifs et réseaux d'automates. *In 3ème Journée Nationale du PRC-IA sur les Systèmes Multi-Agents*, PARIS.

[Fogel, 1995] Fogel, D. B. (1995). *Evolutionary Computation : Toward a New Philosophy of Machine Intelligence.* Piscataway, NJ : IEEE Press.

[Forrest, 1991] Forrest, S., éditeur (1991). *Emergent Computation : Self-organizing, Collective and Cooperative Phenomena in Natural and Artificial Computing Networks.* Special Issues of Physica D. MIT Press.

[Freeman, 1987] Freeman, W. J. (1987). Simulation of chaotic eeg patterns with a dynamic model of the olfactory system. *Biological Cybernetics*, 56(2-3):139–150.

[Gambardella et Dorigo, 1995] Gambardella, L. M. et Dorigo, M. (1995). Ant-q : A reinforcement learning approach to the traveling salesman problem. *In Proceedings of the Eleventh International Conference on Machine Learning*, pages 252–260. Morgan Kaufmann Publishers.

[Glad *et al.*, 2008] Glad, A., Simonin, O., Buffet, O. et Charpillet, F. (2008). Theoretical study of ant-based algorithms for multi-agent patrolling. *In 18th European Conference on Artificial Intelligence ECAI'08*, pages 626–630.

[Grassberger, 1986] Grassberger, P. (1986). Towards a quantative theory of self-generated complexity. *International Journal of Theoretical Physics*, 25(9):907–938.

[Grassé, 1959] Grassé, P.-P. (1959). La reconstruction du nid et les coordinations interindividuelles chez bellicosi- termes natalensis et cubitermes sp. la theorie de la stigmergie : essai d'interpretation des termites constructeurs. *Insectes sociaux*, IV(1):41–83.

[Guillot et Daucé, 2002] Guillot, A. et Daucé, E. (2002). *Approche dynamique de la cognition artificielle.* Traité des Sciences Cognitives. Hermes Sciences, Paris.

[Gutjahr, 2000] Gutjahr, W. J. (2000). A graph-based ant system and its convergence. *Future Gener. Comput. Syst.*, 16(9):873–888.

[Gutjahr, 2002] Gutjahr, W. J. (2002). Aco algorithms with guaranteed convergence to the optimal solution. *Information Processing Letters*, 82(3):145–154.

[Haken, 1977] Haken, H. (1977). *Synergetics : An Introduction. Nonequilibrium Phase-Transitions and Self-Organization in Physics, Chemistry and Biology.* Springer-Verlag, New York.

[Haken, 1979] Haken, H. (1979). Synergetics and bifurcation theory. *Annals New York Academy of Sciences.*

[Haken, 2004] Haken, H. (2004). *Synergetic Computers and Cognition*, volume 50 de *Springer Series in Synergetics.* Springer, 2 édition.

[Haken, 2008] Haken, H. (2008). Self-organization. *Scholarpedia*, 3(8):1401.

[Helbing *et al.*, 2000] Helbing, D., Farkas, I. et Vicsek, T. (2000). Simulating dynamical features of escape panic. *Nature*, 407:487–490.

[Heylighen, 1999] Heylighen, F. (1999). The science of self-organization and adaptivity. *In Knowledge Management, Organizational Intelligence and Learning, and Complexity*, The Encyclopedia of Life Support Systems, EOLSS, pages 253–280. Publishers Co. Ltd.

[Heylighen et Joslyn, 2001] Heylighen, F. et Joslyn, C. (2001). *Cybernetics and Second-Order Cybernetics*.

[Holland, 1992] Holland, J. H. (1992). *Adaptation in Natural and Artificial Systems : An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. The MIT Press, 2 édition.

[Jansen et Rit, 1995] Jansen, B. H. et Rit, V. G. (1995). Electroencephalogram and visual evoked potential generation in a mathematical model of coupled cortical columns. *Biological Cybernetics*, 73:357–366.

[Kaneko, 1983] Kaneko, K. (1983). Transition from torus to chaos accompanied by the frequency locking with symmetry breaking – in connection with the coupled logistic map. *Progres of theorical physics*, 69(5):1427.

[Kaneko, 1990] Kaneko, K. (1990). Clustering, coding, switching, hierarchical ordering, and control in network of chaotic elements. *Physica D*, 41:137–172.

[Kaneko, 1993] Kaneko, K. (1993). *Theory and applications of coupled map lattices*. John Wiley & Sons.

[Kaneko, 1994] Kaneko, K. (1994). Relevance of clustering to biological networks. *Physica D*, 75:55–73.

[Kaneko, 1995] Kaneko, K. (1995). Remarks on the mean field dynamics of network of chaotic elements. *Physica D*, 86.

[Kaneko et Tsuda, 2001a] Kaneko, K. et Tsuda, I. (2001a). Cmls : Constructive approach to spatiotemporal chaos. *In* Asakura Publishing Co., L. T., éditeur : *Complex systems : chaos and beyond*, pages 57–106. Springer-Verlag, Berlin Heidelberg New York.

[Kaneko et Tsuda, 2001b] Kaneko, K. et Tsuda, I. (2001b). *Complex systems : chaos and beyond*. Springer-Verlag, Berlin Heidelberg New York.

[Kennedy et Eberhart, 1995] Kennedy, J. et Eberhart, R. C. (1995). Particle swarm optimization. *In Proc. of the IEEE Int. Conf. on Neural Networks*, pages 1942–1948, Piscataway, NJ. IEEE Service Center.

[Kennedy *et al.*, 2001] Kennedy, J., Eberhart, R. C. et Shi, Y. (2001). *Swarm Intelligence*. The Morgan Kaufmann Series in Artificial Intelligence. Morgan Kaufmann Publishers.

[Korn et Faure, 2003] Korn, H. et Faure, P. (2003). Is there chaos in the brain ? ii. experimental evidence and related models. *C. R. Biol*, 326:787–840.

[Li et York, 1975] Li, T.-Y. et York, J. A. (1975). Period three implies chaos. *The American Mathematical Monthly*.

[Lopes da Silva *et al.*, 1976] Lopes da Silva, F., van Rotterdam, A., Barts, P., van Heusden, E. et Burr, W. (1976). Models of neuronal populations : the basic mechanisms of rhythmicity. *Progress in brain research*, 45:281–308.

[Luke *et al.*, 2003] Luke, S., Balan, G. C., Sullivan, K. et Panait, L. (2003). Mason : Multi-agent simulator of neighborhoods.

[Mamei *et al.*, 2002] Mamei, M., Zambonelli, F. et Leonardi, L. (2002). Co-fields : Towards a unifying approach to the engineering of swarm intelligent systems. *In ESAW*, pages 68–81.

[Maniezzo et Colorni, 1999] Maniezzo, V. et Colorni, A. (1999). The ant system applied to the quadratic assignment problem. *IEEE Transactions on Knowledge and Data Engineering.*, 11(5):769–778.

[Manrubia et Mikhailov, 1999] Manrubia, S. C. et Mikhailov, A. S. (1999). Mutual synchronization and clustering in randomly coupled chaotic dynamical networks. *Phys. Rev. E*, 60(2):1579–1589.

[Manrubia *et al.*, 2004] Manrubia, S. C., Mikhailov, A. S. et Zanette, D. H. (2004). Clustering. *In* Mikhailov, A. S., éditeur : *Emergence of Dynamical Order*, volume 2 de *World Scientific Lecture Notes in Complex Systems*, chapitre 8, pages 171–202. World Scientific.

[May, 1976] May, R. M. (1976). Simple mathematical models with very complicated dynamics. *Nature*, 261.

[Michel, 2007] Michel, F. (2007). Le modèle irm4s de l'utilisation des notions d'influence et de réaction pour la simulation de systèmes multi-agents. *Revue Revue d'Intelligence Artificielle*, 21:5–6.

[Michener, 1974] Michener, C. D. (1974). *The Social Behavior of Bees : a Comparative Study*, page 33. Harvard University Press.

[Millonas, 1993] Millonas, M. M. (1993). Swarms, phase transitions, and collective intelligence. *In eprint arXiv :adap-org/9306002*, page 6002.

[Morin et Weinmann, 1998] Morin, E. et Weinmann, H. (1998). *La complexité humaine.* Champs. Flammarion.

[Newell, 1982] Newell, A. (1982). The knowledge level. *Artificial Intelligence Magazine*, 18(1): 87–127.

[Nicolis, 1991] Nicolis, J. S. (1991). *Chaos and Information Processing.* World Scientific Publishing.

[Parunak, 1997] Parunak, H. (1997). Go to the ant : Engineering principles from natural multi-agent systems. *Annals of Operations Research*, 75:69–101.

[Phatak et Rao, 1995] Phatak, S. C. et Rao, S. S. (1995). Logistic map : a possible random number generator. *Physical Review E.*

[Pikovsky *et al.*, 2001] Pikovsky, A., Popovych, O. et Maistrenko, Y. (2001). Resolving clusters in chaotic ensembles of globally coupled identical oscillators. *Physical Review Letters*, 87(4).

[Pohlheim, 2006] Pohlheim, H. (2006). Geatbx : Genetic and evolutionary algorithm toolbox for use with matlab.

[Prigogine et Nicolis, 1977] Prigogine, I. et Nicolis, G. (1977). *Self-Organization in nonequilibrium systems*. Wiley.

[Rechenberg, 1973] Rechenberg, I. (1973). *Evolutionsstrategie : Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Frommann-Holzboog Verlag.

[Reeves, 1983] Reeves, W. T. (1983). Particle systems-a technique for modeling a class of fuzzy objects. *In acm Transactions on Graphics*, volume 2 de *acm SIGGRAPH '83 Proceedings*, pages 359–376.

[Reynolds, 1987] Reynolds, C. W. (1987). Flocks, herds, and schools : A distributed behavioral model. *Computer Graphics*, 21(4):25–34.

[Reynolds, 1988] Reynolds, C. W. (1988). Not bumping into things.

[Reynolds, 1999] Reynolds, C. W. (1999). Steering behaviors for autonomous characters. *In Proceedings of Game Developers Conference 1999*.

[Ruette, 2003] Ruette, S. (2003). *Chaos for continuous interval maps : a survey of relationship between the various sorts of chaos*. Université Paris-Sud.

[Sahin, 2005] Sahin, E. (2005). Swarm robotics : From sources of inspiration to domains of application. *In Proceedings of the First International Workshop on Swarm Robotics (at SAB 2004)*, volume 3342 de *Lecture Notes in Computer Science*, pages 10–20. Springer.

[Sahin *et al.*, 2008] Sahin, E., Girgin, S., Bayindir, L. et Turgut, A. E. (2008). Swarm robotics. *In* Blum, C. et Merkle, D., éditeurs : *Swarm Intelligence*, Natural Computing Series, pages 87–100. Springer-Verlag.

[Sarkovskii, 1964] Sarkovskii, A. N. (1964). Co-existence of cycles of a continous mapping of a line onto itself. *Ukrainian Mathematical Journal*, 16:61–71.

[Schuster et Just, 2005a] Schuster, H. G. et Just, W. (2005a). *Deterministic Chaos : An Introduction*, chapitre 4, pages 21–25. WILEY-VCH Verlag GmbH and Co. KGaA, Weinheim.

[Schuster et Just, 2005b] Schuster, H. G. et Just, W. (2005b). *Deterministic Chaos : An Introduction*, chapitre 4, pages 34–67. WILEY-VCH Verlag GmbH and Co. KGaA, Weinheim.

[Schuster et Just, 2005c] Schuster, H. G. et Just, W. (2005c). The kolmogorov entropy. *In Deterministic Chaos : An Introduction*, pages 96–121. WILEY-VCH Verlag GmbH and Co. KGaA, Weinheim.

[Schweitzer, 2003a] Schweitzer, F. (2003a). *Brownian Agents and Active Particles*. Springer-Verlag, Berlin Heidelberg New York.

[Schweitzer, 2003b] Schweitzer, F. (2003b). Complex systems and agent models. *In Brownian Agents and Active Particles*, pages 1–46. Springer-Verlag, Berlin Heidelberg New York.

[Schweitzer *et al.*, 1997] Schweitzer, F., Lao, K. et Family, F. (1997). Active random walkers simulate trunk trail formation by ants. *BioSystems*, 41:153–166.

[Shalizi, 2006] Shalizi, C. R. (2006). *Methods and Techniques of Complex Systems Science : An Overview*, chapitre 1, pages 33–114. Complex Systems Science in Biomedicine. Springer.

[Shalizi et Crutchfield, 2001] Shalizi, C. R. et Crutchfield, J. P. (2001). Computational mechanics : Pattern and prediction, structure and simplicity. *Journal of Statistical Physics*, 104:817–879.

[Shalizi *et al.*, 2004] Shalizi, C. R., Shalizi, K. L. et Haslinger, R. (2004). Quantifying self-organization with optimal predictors. *Physical Review Letters*, 93.

[Shannon, 1948] Shannon, C. E. (1948). A mathematical theory of communication. *The Bell System Technical Journal*, 27:379–423.

[Shibata et Kaneko, 2003] Shibata, T. et Kaneko, K. (2003). Coupled map gas : structure formation and dynamics of interacting motile elements with internal dynamics. *Physica D*, 181:197–214.

[Simonin, 2001] Simonin, O. (2001). *Le modèle satisfaction-altruisme : coopération et résolution de conflits entre agents situés réactifs, application à la robotique.* Thèse de doctorat, Université Montpellier 2.

[Sompolinsky *et al.*, 1988] Sompolinsky, H., Crisanti, A. et Sommers, H. J. (1988). Chaos in random neural networks. *Physical Review Letters*, 61(3):259–262.

[Sprott,] Sprott, J. C. S. *One-dimensional maps*, chapitre 2. Oxford University Press.

[Sprott, 2003] Sprott, J. C. S. (2003). *Chaos and Time-Series Analysis.* Oxford University Press.

[Strogatz, 2001] Strogatz, S. H. (2001). *Nonlinear Dynamics and Chaos.* Westview Press, 1 édition.

[Stützle et Dorigo, 1999] Stützle, T. et Dorigo, M. (1999). Aco algorithms for the traveling salesman problem. *In Evolutionary Algorithms in Engineering and Computer Science*, pages 163–183. John Wiley & Sons.

[Stützle et Dorigo, 2002] Stützle, T. et Dorigo, M. (2002). A short convergence proof for a class of ant colony optimization algorithms. *IEEE Trans. Evolutionary Computation*, 6(4):358–365.

[Stützle et Hoos, 1997] Stützle, T. et Hoos, H. H. (1997). The max –min ant system and local search for the traveling salesman problem. *In Proceedings of the IEEE International Conference on Evolutionary Computation (ICEC'97)*, pages 309–314.

[Takens, 1981] Takens, F. (1981). Detecting strange attractors in turbulence. *In* Rand, D. et Young, L., éditeurs : *Proceedings of the Symposion on Dynamical Systems and Turbulence,University of Warwick, 1979-1980.* Springer.

[Tschacher et Dauwalder, 1999] Tschacher, W. et Dauwalder, J.-P., éditeurs (1999). *Dynamics Synergetics Autonomous Agents*, volume 8 de *Studies of Nonlinear Phenomena in Life Science.* World Scientific.

[Ulam et von Neumann, 1947] Ulam, S. M. et von Neumann, J. (1947). On combination of stochastic and deterministic processes. *Bulletin of the American Mathematical Society*, 53.

[Vallée, 1951] Vallée, R. (1951). Sur deux classes d'"< opérateurs d'observation ">. *Comptes Rendus de l'Académie des Sciences*, (233):1350–1951.

[Vallée, 1996] Vallée, R. (1996). *Cognition et système – Essai d'épistémo-praxéologie.* Système(s). L'Interdisciplinaire, Paris.

[Vicsek *et al.*, 1995] Vicsek, T., Czirok, A., Jacob, E. B., Cohen, I. et Schochet, O. (1995). Novel type of phase transitions in a system of self-driven particles. *Physical Review Letters*, 75:1226–1229.

[von Bertalanffy, 1968] von Bertalanffy, L. (1968). *General System Theory : Foundations, Development, Applications*. George Braziller, New York, revised edition édition.

[von Foerster, 1960] von Foerster, H. (1960). On self-organizing systems and their environments. *In Self-Organizing Systems*, pages 31–50. Pergamon Press.

[von Foerster, 1979] von Foerster, H. (1979). Cybernetics of cybernetics. *Communication and Control*, pages 5–8.

[von Neumann, 1966] von Neumann, J. (1966). *Theory of Self-Reproducing Automata*. University of Illionois Press, Champain, IL, USA.

[Walters, 1982] Walters, P. (1982). *An Introduction to Ergodic Theory*. Springer-Verlag, 2 édition.

[Wang, 1991] Wang, X. (1991). Period-doublings to chaos in a simple neural network : An analytic proof. *Complex Systems*, 5:425–441.

[Weaver, 1948] Weaver, W. (1948). Science and complexity. *American Scientist*, 36(536).

[Weyns et Holvoet, 2004] Weyns, D. et Holvoet, T. (2004). A formal model for situated multi-agent systems. *Fundam. Inform.*, 63(2-3):125–158.

[Weyns *et al.*, 2004] Weyns, D., Parunak, H. V. D., Michel, F., Holvoet, T. et Ferber, J. (2004). Environments for multiagent systems state-of-the-art and research challenges. *In* Weyns, D., Parunak, H. V. D. et Michel, F., éditeurs : *Environments for Multi-Agent Systems, First International Workshop, E4MAS 2004, New York, NY, USA, July 19, 2004, Revised Selected Papers*, pages 1–47. Springer.

[Wiener, 1948] Wiener, N. (1948). *Cybernetics, or Control and Communication in the Animal and the Machine*. Librairie Hermann & Cie,MIT Press.

[Wilensky, 1999] Wilensky, U. (1999). Netlogo.

[Wolf *et al.*, 1985] Wolf, A., Swift, J. B., Swinney, H. L. et Vastano, J. A. (1985). Determining Lyapunov exponents from a time series. *Physica D*, 16:285–317.

[Wooldridge et Jennings, 1995] Wooldridge, M. et Jennings, N. R. (1995). Intelligent agents : Theory and practice. *Knowledge Engineering Review*, 10:115–152.

Résumé

L'intelligence en essaim constitue désormais un domaine à part entière de l'intelligence artificielle distribuée. Les problématiques qu'elle soulève touchent cependant à de nombreux autres domaines ou questions scientifiques. En particulier le concept d'essaim trouve pleinement sa place au sein de la science dites des "systèmes complexes". Cette thèse présente ainsi la conception, les caractéristiques et les applications d'un modèle original, le système multi-agent logistique (SMAL), pour le domaine de l'intelligence en essaim. Le SMAL trouve son origine en modélisation des systèmes complexes : il est en effet issu des réseaux d'itérations logistiques couplées dont nous avons adapté le modèle de calcul au schéma "influence-réaction" des systèmes multi-agents. Ce modèle est fondé sur des principes communs à d'autres disciplines, comme la synchronisation et le contrôle paramétrique, que nous plaçons au coeur des mécanismes d'auto-organisation et d'adaptation du système. L'environnement à base de champs est l'autre aspect fondamental du SMAL, en permettant la réalisation des interactions indirectes des agents et en jouant le rôle d'une structure de données pour le système. Les travaux décrits dans cette thèse donnent lieu à des applications principalement en simulation et en optimisation combinatoire. L'intérêt et l'originalité du SMAL pour l'intelligence en essaim résident dans l'aspect générique de son schéma théorique qui permet de traiter avec un même modèle des phénomènes considérés *a priori* comme distincts dans la littérature : phénomènes de "flocking" et phénomènes stigmergiques "fourmis" à base de phéromones. Ce modèle répond ainsi à un besoin d'explication des mécanismes mis en jeu autant qu'au besoin d'en synthétiser les algorithmes générateurs.

Mots-clés: Intelligence en essaim, Intelligence artificielle distribuée, Systèmes complexes, Auto-organisation, Adaptation, Contrôle distribué, Chaos déterministe, Phénomènes de synchronisation

Abstract

Swarm Intelligence is from now on a full part of Distributed Artificial Intelligence. Its associated problematics meet many other fields and scientific questions. The concept of swarm in particular belongs to the science called the science of complex systems. This phd thesis shows the design and the characteristics and the applications of a novel type of model called the logistic multi-agent system (LMAS) dedicated to the Swarm Intelligence field. The LMAS has its foundations in complex system modeling : it is inspired from the coupled logistic map lattice model which has been adapted to the "Influence-Reaction" modeling of multi-agent systems. This model is based on universal principles such as synchronization and parametric control which are considered as the main mechanisms of self-organization and adaptation in the heart of the system. The field-layered based environment is the other important feature of the LMAS, since it enables indirect interactions and plays the part of a data structure for the whole system. The work of this thesis is put into practice for simulation and optimization. The novelty of the LMAS lies in its generic theoretical framework, which enables to tackle problems considered as distinct in the literature, in particular flocking and ant-like stigmergic behavior. This model meets the need of explaining basic mechanisms and the need of synthesizing generative algorithms for the Swarm Intelligence.

Keywords: Swarm Intelligence, Distributed Artificial Intelligence, Complex Systems, Self-Organization, Adaptation, Distributed Control, Deterministic Chaos, Synchronization Phenomena

---

Laboratoire Lorrain de Recherche en Informatique et Applications

Equipe MAIA

Campus Scientifique, B.P.239 - 54506 Vandœuvre-lès-Nancy Cedex