



HAL
open science

Méthodes approchées pour la résolution de problèmes inverses : identifcaiton paramétrique et génération de formes

Fazia Aiboud

► **To cite this version:**

Fazia Aiboud. Méthodes approchées pour la résolution de problèmes inverses : identifcaiton paramétrique et génération de formes. Modélisation et simulation. Université Blaise Pascal - Clermont-Ferrand II, 2013. Français. NNT : 2013CLF22399 . tel-03917482

HAL Id: tel-03917482

<https://theses.hal.science/tel-03917482>

Submitted on 2 Jan 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

UNIVERSITE BLAISE PASCAL - CLERMONT-FERRAND II

ECOLE DOCTORALE

SCIENCES POUR L'INGENIEUR DE CLERMONT-FERRAND

PROJET DE THESE

en Informatique

présenté par

Fazia AIBOUD

Méthodes approchées pour la résolution de
problèmes inverses : identification paramétrique
et génération de formes

Thèse dirigée par Sylvie NORRE et Nathalie GRANGEON

Soutenue le 29 Novembre 2013

Directeur de Thèse

Sylvie NORRE Professeur des Universités Université Blaise Pascal Clermont-Ferrand

Co-Directeur de Thèse

Nathalie GRANGEON Maître de Conférences Université Blaise Pascal Clermont-Ferrand

Rapporteurs

Bernard GRABOT Professeur des Universités Ecole Nationale d'Ingénieurs de Tarbes
Patrick SIARRY Professeur des Universités Université Paris-Est Créteil

Examineurs

Michel GOURGAND Professeur des Universités Université Blaise Pascal Clermont-Ferrand
Sébastien VEREL Maître de Conférences Université du Littoral Côte d'Opale Calais

Résumé

Cette thèse est réalisée dans le cadre d'un projet de collaboration entre le LIMOS et l'entreprise Integrative Bio-Computing (IBC). Cette entreprise développe une plateforme générique de simulation et de modélisation, qui génère des modèles multi-échelles des systèmes biologiques en particulier les organes humains. Cette plateforme combine des modèles à deux niveaux de description : macroscopique pour modéliser le fonctionnement de l'organe par des équations différentielles et microscopique pour modéliser les interactions au niveau cellulaire par automates cellulaires.

L'objectif de cette thèse est de proposer des méthodes approchées pour résoudre, d'une part le problème d'identification des paramètres des équations différentielles, d'autre part, la détermination des règles (ou fonction de transition) de l'automate cellulaire permettant d'obtenir la bonne structure de l'organe.

Les méthodes proposées pour la modélisation macroscopique sont à base de métaheuristiques (recuit simulé et algorithme génétique). Elles consistent à déterminer les paramètres des équations différentielles à partir d'un ensemble d'observations temporelles.

Dans le cas de la modélisation microscopique, des heuristiques (algorithme glouton, recherches locales déterministes) et métaheuristiques (algorithme génétique, recherche locale itérée et recuit simulé) ont été proposées. Elles ont pour rôle de déterminer la fonction de transition de l'automate cellulaire et son nombre de générations permettant d'obtenir une forme voulue. Différentes approches, codages, critères, systèmes de voisinage et systèmes d'interaction ont été proposés. Différentes expérimentations ont été menées pour la génération de formes (pleines, symétriques et quelconques).

Mots clés

Problèmes inverses, identification paramétrique, automates cellulaires et équations différentielles, métaheuristiques.

Abstract

This work is realized in collaboration between LIMOS laboratory and IBC Company. This company develops a generic simulation and modeling framework, which generates multi-scales biological systems models, particularly, synthetic human organs. This framework combines different models with two description scales: *macroscopic* with differential equations to model the organ operation and *microscopic* to model cellular interactions by cellular automata.

The aim of this work is to propose methods to solve two problems: parameter identification of differential equations and rules determination of cellular automata which modeled the organ structure.

Proposed methods to microscopic modeling are based on metaheuristics (simulated annealing and genetic algorithm). These methods determine the parameters of differential equations from a set of temporal observations.

Heuristics (greedy algorithm and deterministic local search) and metaheuristics (genetic algorithm, iterated local search and simulated annealing) are proposed in the case of macroscopic modeling. These methods determine the transition function and the number of generations of the cellular automaton generating a shape as near as a targeted shape. Different approaches, encoding, criteria, neighborhood systems and interaction system are proposed. Experimentations are realized to generate some type of shapes (full, symmetric and any shape).

Keywords

Inverse problem, parameter identification, cellular automata, differential equations, metaheuristics

Remerciements

Je tiens à remercier en premier lieu le professeur **Sylvie Norre** ma directrice de thèse, et **Nathalie Grangeon** ma co-encadrante, qui ont su par leurs qualités scientifiques et humaines, leurs précieux conseils et leurs expériences, diriger mon travail et m'orienter dans mes recherches de façon à mener à bien ce projet. Grâce à vous, j'ai appris plein d'autres choses (vocabulaire, chasse aux escargots, loup garou,...etc).

Que le professeur des universités **Michel Gourmand**, notre exceptionnel chef trouve dans ce rapport mes remerciements les plus profonds. Ça a été une grande fierté et un grand plaisir pour moi de travailler avec lui.

Je tiens également à remercier les membres du jury, le professeur **Patrick Siarry** et le professeur **Bernard Grabot** d'avoir accepté d'examiner ce rapport. Leurs conseils et leurs remarques m'ont été très bénéfiques. Un grand merci à **Sébastien Verel**, qui a accepté de faire partie du jury, et pour tous nos échanges scientifiques pendant nos rencontres en conférences.

Mes remerciements vont aussi vers **David Sarramia** du Laboratoire de Physique Corpusculaire (LPC) et **Laurent Derroussi** du LIMOS, qui ont toujours été présents pour m'aider et me conseiller.

Ces années de thèse ne seraient pas aussi agréables sans la présence de mes collègues de bureau. **Yves-Jean Daniel** enseignant au département GLT et **Benjamin Louvet** doctorant du LIMOS. Je sais que ça n'a pas été toujours facile pour vous de me supporter. Grâce à vous j'ai pu surmonter toutes les difficultés rencontrées.

Je tiens à remercier le **conseil général d'Allier** qui a financé cette thèse.

Je remercie **Cécile Charasse** directrice de l'IUT d'Allier qui m'a accueillie dans son établissement et toute l'**équipe GLT** (Anne-Céline, Valérie, Fabrice, Christine, Catherine, Sandra, Emilie, Nadine, Didier, Christophe, Paul, Marc et Alexy) pour la belle ambiance qui ne pouvait que m'aider à l'aboutissement de ce travail. Merci pour les repas partagés ensemble. Vous avez été ma seconde famille.

Je voudrais remercier **Jean-Louis Robert** responsable du département GSP, **Claudy Aubert Dassy** chef du département GTE et **Jean-Paul Brouackert** chef du département TC qui m'ont donné l'opportunité d'enseigner dans leurs départements.

Je remercie également le professeur **Abdelhamid ADANE** et **Amir Nakib** qui m'ont donné le goût de la recherche ainsi que **Virginie André** qui m'a beaucoup aidé et qui m'a consacré du temps pour relire ce rapport qui malheureusement reste plein de coquilles.

Ces remerciements ne seraient pas complets sans les membres de ma famille, qui m'ont toujours encouragée et soutenue en particulier mes parents qui ont cru en moi, mon frère **Hamid** qui m'a beaucoup encouragée et mon très cher **Toufik**, grâce à qui j'ai réussi à surmonter tous les moments difficiles et qui a su me reconforter surtout en fin de thèse. Un grand merci à ma belle-famille.

Je tiens à remercier toutes les personnes qui ont contribué au bon déroulement de cette thèse d'une quelconque manière.

SOMMAIRE

INTRODUCTION GENERALE	1
CHAPITRE I. CONTEXTE DE LA MODELISATION DES SYSTEMES BIOLOGIQUES, PROBLEMATIQUES ET METHODES	5
Introduction	5
I.1 Modélisation des systèmes biologiques	6
I.1.1 Représentation graphique.....	7
I.1.2 Niveaux de description du modèle.....	8
I.1.3 Type du modèle.....	9
I.1.4 Orientation du modèle.....	9
I.1.5 Mode de communication.....	11
I.2 Problèmes à résoudre	11
I.3 Méthodes d'optimisation	12
I.3.1 Heuristiques	13
I.3.2 Métaheuristiques	14
Conclusion.....	23
CHAPITRE II. MODELISATION MACROSCOPIQUE : PROPOSITION DE METHODES DE RESOLUTION POUR L'IDENTIFICATION PARAMETRIQUE.....	25
Introduction	25
II.1 Présentation du problème	26
II.1.1 Problème général	26
II.1.2 Problème d'un système physique	27
II.2 Proposition de méthodes de résolution	28
II.2.1 Codage d'une solution	28
II.2.2 Fonction objectif.....	29
II.2.3 Recuit simulé	30
II.2.4 Algorithme génétique	31
II.3 Résultats dans le cas des paramètres entiers.....	33
II.3.1 Enumération.....	34
II.3.2 Recuit simulé	35
II.3.3 Algorithme génétique	36
II.4 Résultats dans le cas de paramètres réels	36
II.4.1 Résultat du recuit simulé	37
II.4.2 Résultat de l'algorithme génétique	38
Conclusion.....	38
CHAPITRE III. AUTOMATES CELLULAIRES ET PROBLEMES INVERSES : ETAT DE L'ART	41
Introduction	41
III.1 Automate cellulaire	41
III.1.1 Définition.....	42
III.1.2 Notations	43
III.1.3 Dynamique de l'automate cellulaire.....	51
III.1.4 Cas particuliers d'automates cellulaires	51
III.2 Problèmes inverses d'automates cellulaires	52
III.2.1 Définitions	52
III.2.2 Quelques problèmes inverses d'automates cellulaires	53
Conclusion.....	74

CHAPITRE IV. MODELISATION MICROSCOPIQUE : PROPOSITION DE METHODES DE RESOLUTION DU PROBLEME INVERSE.....	75
Introduction.....	75
IV.1 Définition du problème.....	76
IV.2 Critères de similarité.....	77
IV.2.1 Nombre de cellules identiques.....	77
IV.2.2 Ecart quadratique.....	77
IV.2.3 Adaptation du critère de De Garis aux formes creuses.....	78
IV.2.4 Information mutuelle.....	78
IV.2.5 Moments de Legendre.....	79
IV.3 Formalisation mathématique.....	80
IV.3.1 Automate cellulaire 1D.....	81
IV.3.2 Automate cellulaire 2D.....	83
IV.3.3 Automate cellulaire 3D à plusieurs états.....	86
IV.4 Codages d'une solution et approches de résolution.....	87
IV.4.1 Détermination simultanée de la fonction transition (F) et du nombre de générations (Ng).....	87
IV.4.2 Détermination séparée de F et Ng	90
IV.5 Heuristiques.....	93
IV.5.1 Algorithme glouton.....	93
IV.5.2 Recherche locale déterministe avec le meilleur voisin.....	95
IV.5.3 Recherche locale déterministe avec le premier voisin.....	96
IV.6 Métaheuristiques.....	96
IV.6.1 Algorithme génétique.....	96
IV.6.2 Systèmes de voisinage.....	98
IV.6.3 Recherche locale itérée.....	99
IV.6.4 Recuit simulé.....	100
IV.7 Hybridation des méthodes.....	102
IV.8 Cas particuliers des formes symétriques.....	103
IV.8.1 Symétrie horizontale.....	104
IV.8.2 Espace de recherche.....	107
Conclusion.....	108
CHAPITRE V. MODELISATION MICROSCOPIQUE : MISE EN ŒUVRE ET RESULTATS.....	111
Introduction.....	111
V.1 Bibliothèque d'instances.....	112
V.1.1 Formes 2D binaires quelconques.....	112
V.1.2 Formes 2D binaires symétriques.....	113
V.1.3 Formes 3D binaires symétriques.....	113
V.1.4 Formes 2D à trois états symétriques.....	114
V.2 Comparaison et choix des critères de similarité.....	115
V.2.1 Comparaison des critères : nombre de cellules identiques, information mutuelle et moments de Legendre.....	116
V.2.2 Comparaison du nombre de cellules identiques et de l'information mutuelle.....	118
V.3 Justification de l'utilisation des métaheuristiques.....	120
V.4 Comparaison avec les formes de la littérature.....	121
V.4.1 Comparaison avec les formes de Chavoya et Duthen.....	121
V.4.2 Résultats sur les formes de Breukelaar et Back.....	122
V.5 Résultats des formes 2D binaires quelconques.....	122
V.5.1 Heuristiques.....	122
V.5.2 Métaheuristiques.....	125
V.6 Résultats des formes 2D binaires symétriques.....	130
V.6.1 Algorithme génétique.....	130
V.6.2 Recherche locale itérée.....	132

V.6.3 Recuit simulé	135
V.6.4 Résultats groupés par approches	138
V.6.5 Résultats groupés par système de voisinage.....	139
V.6.6 Temps de calcul	139
V.7 Résultats des formes binaires 3D	141
V.8 Résultats des formes symétriques 2D à trois états	142
V.9 Autres applications	143
Conclusion.....	145
CONCLUSION ET PERSPECTIVES.....	147
Identification du type de symétrie	153
La symétrie d'une forme	153
Les règles équivalentes par symétrie.....	153
Symétrie verticale.....	153
Symétrie horizontale et verticale.....	154
Rotation à 90°	155
Rotation à 180°.....	156
REFERENCES BIBLIOGRAPHIQUES.....	159

INTRODUCTION GENERALE

Depuis très longtemps, l'homme cherche à comprendre les processus naturels qui sont à l'origine du développement des formes des espèces connu sous le nom de morphogénèse. Il s'est intéressé en particulier au processus d'embryogénèse qui est à l'origine des formes créées par croissance cellulaire.

Une grande variété de modèles continus ou discrets a été proposée dans la littérature (Deutch & Dormann, 2005). Les modèles peuvent aller d'une simple molécule à différents composants du système du vivant aussi complexes qu'une cellule (tissus, organe,...). Par conséquent, différents niveaux de description peuvent être définis: *macroscopique*, lorsque l'on s'intéresse uniquement à la population de cellules ou de molécules et à son fonctionnement (tissus, organe, ...) et *microscopique*, lorsque la modélisation se focalise sur les individus (cellules, molécules,...). D'autres modèles multi-échelles combinent ces deux niveaux de description et intègrent plusieurs échelles spatio-temporelles.

Un projet de collaboration entre le Laboratoire d'Informatique de Modélisation et d'Optimisation des Systèmes (LIMOS UMR 6158), le Laboratoire de Physique Corpusculaire (LPC UMR 6533) et l'entreprise Integrative BioComputing (IBC) est en cours. Cette entreprise développe une *Plateforme Générique de Modélisation et de Simulation* (PGMS) qui correspond à un générateur de modèles multi-échelle des systèmes biologiques.

Ce travail se place dans le cadre de la modélisation et la simulation d'un être humain synthétique de façon à pouvoir comprendre les interactions intervenant dans le corps humain. En particulier, le projet européen VPH (Virtual Physiological Human) du FP6 européen est à rapprocher de ce travail de thèse. Un des grands challenges est de produire des modèles de santé personnalisés patient. La différence majeure entre le projet VPH et le sujet proposé est que VPH consiste en le développement de modèles d'organes par des équipes de recherche différentes de façon indépendante pour ensuite les assembler tandis que l'approche de ce projet consiste à générer n'importe quel organe pour tout patient à partir du même modèle à base d'automates cellulaires.

Les objectifs de ce projet sont multiples :

- *Apprentissage et formation* : produire et simuler des modèles informatiques de croissance cellulaire permettant d'étudier le processus de morphogénèse afin de faciliter la compréhension des différents processus biologiques, biomécaniques du système du vivant. Cette modélisation tient compte des phénomènes d'auto-organisation qui permettent, à partir d'un équilibre local (les interactions entre les cellules ou molécules) au niveau microscopique de créer un équilibre global au niveau macroscopique (la fonction des cellules, des tissus, des organes et du système global du vivant).
- *Application en radiobiologie* : étudier l'impact d'un défaut au niveau génétique sur la croissance cellulaire de l'organisme. A ce titre, *Benjamin Louvet* un doctorant du LIMOS et du LPC, travaille sur un sujet dans ce sens (automates cellulaires pour la modélisation multi-échelle de systèmes biologiques, application en radiobiologie). Ses travaux consistent à développer un modèle de croissance bactérienne dans un environnement normal, dans un

environnement stressant, ou dans un environnement normal avec exposition temporaire à un stress radiatif.

- *Paralléliser sur GPU (Graphical Processing Unit) et optimiser l'implémentation de la plateforme PGMS* : réaliser les étapes de la modélisation et la simulation des systèmes biologiques de manière parallèle pour accélérer les traitements et réduire le temps de simulation. Ce travail a été réalisé dans le cadre de la thèse de *Jonathan Caux* au LIMOS soutenue en novembre 2012 (parallélisation et optimisation d'un simulateur de morphogenèse d'organes : application aux éléments du rein).
- *Aide à la décision médicale, recherche biomédicale et biopharmaceutique* : développer des modèles permettant de simuler et de prédire l'évolution de certaines maladies, réaliser des tests de médicaments sur ces modèles et développer des modèles fonctionnels à partir d'images médicales.

Notre contribution dans le projet concerne la proposition d'un outil d'aide à la décision permettant de choisir les nombreux paramètres des modèles, afin de reproduire la structure et le fonctionnement du système biologique modélisé. Actuellement, ces paramètres sont choisis par le concepteur.

La plateforme PGMS combine des modèles continus (équations différentielles) pour la modélisation macroscopique et des modèles discrets (automates cellulaires) pour la modélisation microscopique. La génération d'organes par automate cellulaire et équations différentielles est un problème de double complexité car, d'une part, il va falloir trouver les bons paramètres des équations différentielles permettant de reproduire le bon fonctionnement de l'organe (ex : cœur avec une bonne fonction cardiaque reproduisant un électrocardiogramme correct) ; d'autre part, il faudra choisir les bonnes règles d'évolution de l'automate cellulaire permettant de comprendre comment des formes complexes peuvent émerger à partir d'un ensemble de cellules initialement identiques.

L'objectif principal de cette thèse est de proposer des méthodes permettant de traiter ces deux problèmes inverses qui sont des problèmes d'optimisation difficile. Pour les résoudre, nous proposons d'utiliser des méthodes d'optimisation et en particulier des métaheuristiques. Ces méthodes ont l'avantage de s'adapter aux problèmes continus (modélisation macroscopique de la fonction de l'organe par équations différentielles) et aux problèmes discrets (modélisation microscopique de la génération de la forme de l'organe par automates cellulaires).

Cette thèse est composée de cinq chapitres :

- Le *premier chapitre* concerne la modélisation des systèmes biologiques. Nous présentons les modèles des systèmes biologiques pour la morphogénèse, le contexte et la problématique de la thèse qui concerne l'identification paramétrique et le problème inverse des automates cellulaires. Les méthodes d'optimisation à base d'heuristiques et métaheuristiques seront présentées dans ce chapitre.
- Le *deuxième chapitre* concerne la modélisation macroscopique et l'identification des paramètres d'un système décrit par des équations différentielles. Ces méthodes ont pour but de déterminer les paramètres des équations différentielles à partir des observations du système. Nous proposons des méthodes à base de métaheuristiques. Les méthodes proposées seront appliquées pour identifier les paramètres d'un système physique oscillant.
- Dans le *troisième chapitre*, les automates cellulaires et leurs notations sont présentés dans une première partie. Dans une seconde partie, une classification des problèmes inverses recensés dans la littérature est présentée. Pour résoudre ces problèmes inverses, la plupart des travaux de la littérature proposent des algorithmes évolutionnaires.

-
- Dans *le quatrième chapitre*, des méthodes de résolution du problème inverse des automates cellulaires sont proposées. Elles ont pour rôle de déterminer la fonction de transition F de l'automate cellulaire ainsi que son nombre de générations N_g , qui permettraient à l'automate cellulaire d'obtenir à partir d'une configuration initiale, une configuration proche d'une configuration souhaitée. La première partie du chapitre est consacrée aux critères de similarité permettant de comparer deux configurations de l'automate cellulaire. Dans une deuxième partie, une formalisation mathématique du problème est proposée. Deux codages de solutions donnant trois approches de résolution sont présentés dans une troisième partie. Dans une première approche (F, N_g) , nous proposons de chercher simultanément la fonction de transition et le nombre de générations de l'automate cellulaire. Nous proposons ensuite de dissocier le problème, ce qui a donné deux approches (N_g/F) et (F/N_g) . Dans l'approche (N_g/F) , le nombre de générations est recherché pour chaque fonction de transition. Dans l'approche (F/N_g) , pour chaque nombre de génération, la meilleure fonction de transition est calculée. Des méthodes à base d'heuristiques (algorithme glouton, recherche locale déterministe avec le meilleur voisin et avec le premier voisin) et de métaheuristiques (algorithme génétique, recherche locale itérée et recuit simulé) sont proposées. Afin de combiner la capacité d'exploration et d'intensification de ces méthodes, une hybridation des heuristiques est proposée. Pour accélérer la convergence des méthodes d'optimisation, nous proposons un codage qui prend en compte les informations *a priori* sur la configuration de l'automate cellulaire recherchée (cas des configurations symétriques).
 - Dans *le cinquième et dernier chapitre*, la mise en œuvre des méthodes proposées est présentée. Une bibliothèque de formes est proposée. Elle est composée de différentes formes binaires et à 3 états, quelconques et symétriques, 2D et 3D). Les résultats des méthodes proposées (heuristiques et métaheuristiques) sont présentés. Une application des méthodes proposées pour la segmentation d'images est donnée.
 - Ces cinq chapitres sont suivis par deux annexes et les références bibliographiques.

Chapitre I. CONTEXTE DE LA MODELISATION DES SYSTEMES BIOLOGIQUES, PROBLEMATIQUES ET METHODES

Introduction

Dans ce travail, nous nous intéressons à la modélisation des systèmes biologiques afin de comprendre les processus naturels de la *morphogénèse*. La *morphogénèse* représente les processus responsables de la création de la structure et de la forme des organismes vivants. L'*embryogénèse* est un cas particulier de morphogénèse. Elle regroupe tous les processus de développement d'un organisme. Ces processus dépendent de l'activité des cellules et évoluent dynamiquement.

Un organe est composé de plusieurs cellules qui interagissent entre elles et avec leur environnement pour remplir son rôle dans un organisme vivant. Ainsi, chaque organe remplit une fonction bien précise, dans un être vivant complexe, qui est intimement liée à l'activité des cellules qui le composent. L'équilibre des échanges entre les cellules au niveau local forme un équilibre global de l'organe. Ces deux équilibres ont donné lieu à deux niveaux de description : *microscopique* au niveau local et *macroscopique* au niveau global.

Plusieurs modèles microscopiques et macroscopiques ont été proposés dans la littérature. Ces modèles sont soit discrets, soit continus dans l'espace et/ou le temps. Il existe d'autres modèles multi-échelles qui combinent ces deux niveaux de description et qui intègrent différentes échelles spatio-temporelles pertinentes allant de la cellule au système biologique.

Pour modéliser un organe aux différents niveaux de description, plusieurs paramètres doivent être définis pour reproduire la structure globale de l'organe et son bon fonctionnement. Parmi ces paramètres, nous trouvons le nombre de types cellulaires, les différents types d'interaction, les phénomènes de différenciation et la diffusion des nutriments. Il est évident qu'il existe plusieurs paramètres pour caractériser un organe ou les interactions entre les cellules. La principale difficulté sera de sélectionner ceux qui sembleront les plus pertinents.

Un de nos deux problèmes est de déterminer les paramètres qui conduisent à un état final du système modélisé à partir d'un état initial connu. Ce qui correspond à la modélisation des interactions cellulaires. Le second consiste à identifier les paramètres d'un système, connaissant son fonctionnement. Par exemple : quels sont les paramètres d'un modèle mathématique afin de reproduire le fonctionnement d'un organe.

Ces problèmes, connus sous le nom de problèmes inverses, peuvent être formulés en termes de problèmes d'optimisation difficile. Pour les résoudre, nous proposons d'utiliser des méthodes d'optimisation, en particulier des métaheuristiques.

Ce chapitre est composé de trois sections :

- La première section est un résumé des modèles de modélisation de systèmes biologiques pour la morphogénèse. Plusieurs modèles ont été proposés dans la littérature. Ces modèles

dépendent du niveau de description macroscopique à l'échelle de l'organe, ou microscopique à l'échelle de la cellule et du type du modèle (discret ou continu). Une classification, en lien avec la représentation graphique, le type et le niveau de description sera donnée.

- Dans la deuxième section, nous présenterons le contexte général de cette thèse, les modèles considérés et les problèmes à résoudre. Ces problèmes concernent, d'une part, l'identification paramétrique pour la modélisation macroscopique, qui est un problème d'optimisation continu. D'autre part, la résolution d'un problème inverse pour la génération de forme dans le cas de la modélisation microscopique, qui est un problème d'optimisation discret.
- Pour la dernière section, les méthodes d'optimisation (heuristiques et métaheuristiques) qui seront utilisées pour résoudre ces deux problèmes sont présentées. Ces méthodes peuvent s'appliquer aux problèmes continus (cas de la modélisation macroscopique) et aux problèmes discrets (modélisation microscopique).

I.1 Modélisation des systèmes biologiques

Dans la littérature, plusieurs modèles de morphogénèse ont été proposés. Initialement ces modèles ont été classifiés selon deux approches :

- *Approche basée sur la croissance cellulaire* : principale approche introduite en 1917 par d'Arcy. Dans ces travaux, (D'Arcy, 1942) décrit la forme d'un organisme comme le résultat d'une croissance cellulaire. Cette croissance correspond à une augmentation de taille de la structure qui entraîne une modification progressive de la forme. Il a défini trois paramètres pour décrire une forme : la taille, la direction et le temps. La forme est déterminée par son taux de croissance dans les différentes directions.
- *Approche basée sur les flux de substances chimiques à travers les tissus* : (Turing, 1952) propose une autre approche, qui se focalise sur les flux des substances chimiques à travers les tissus qui contribuent à la génération d'une forme. Il considère qu'un organisme est une masse de tissus qui ne sont pas en croissance, mais dans lesquels des substances chimiques réagissent et diffusent. Ces substances chimiques sont appelées « *morphogènes* » pour désigner des producteurs de forme. Ces processus de diffusion-réaction sont modélisés par des équations différentielles.

Ces deux approches sont équivalentes dans le cas de l'embryogénèse, car la forme de l'embryon a un grand impact sur la diffusion des morphogènes et donc sur l'état des cellules. Réciproquement, l'état de chaque cellule détermine l'évolution de la forme de l'embryon. Ces deux approches sont la première classification des modèles de morphogénèse.

Suivant les modèles utilisés pour décrire le processus de morphogénèse, nous rencontrons la thermodynamique et sa formulation des états d'équilibre, la géométrie et les fractales, les automates cellulaires ou les L-Systèmes. Prusinkiewicz a proposé une autre classification des modèles de morphogénèse (Prusinkiewicz, 1993) basée sur :

- la représentation graphique (topologie, dimension,...) ;
- le niveau de description (microscopique ou macroscopique) ;
- le type du modèle, selon le caractère continu ou discret de l'espace, du temps ou des variables d'état locales ;
- le mode de communication et l'orientation du modèle (espace ou structure).

En plus de ces différents points de classifications, (Bourgine & Lesne, 2006) ont ajouté un autre point lié au régime d'évolution du modèle. Cette évolution peut être soit déterministe soit stochastique.

I.1.1 Représentation graphique

La première étape pour la génération d'un modèle de morphogénèse est le choix de la représentation graphique des cellules. Cette représentation dépend du processus biologique que l'on veut simuler, qui peut être modélisé soit en une, deux ou trois dimension(s).

Dans la représentation la plus simple des cellules, chaque cellule est représentée par un point dans un réseau régulier. Dans cette représentation, toutes les cellules sont de même taille et ont la même forme (carrée, hexagonale, triangulaire,...etc). Cette représentation est souvent utilisée pour modéliser la croissance cellulaire, qui tient compte de la division, de la différenciation et des mouvements cellulaires.

Pour modéliser les échanges mécaniques cellulaires dus aux diffusions chimiques complexes (Champman & Jones, 2012), la représentation peut être réalisée sur un réseau irrégulier, où chaque cellule est identifiée par son centre de gravité. Les deux représentations sont :

- *Représentation sous forme de polygones* (Figure I.1) dans le domaine de Dirichlet (*diagramme de Voronoï*) : une cellule représente un sous-ensemble de l'espace donné par les plus proches points du réseau de son centre de gravité (Honda, 1978; Graner & Sawada, 1993). Selon les auteurs, ce type de représentation est plus proche de la réalité, mais il est très coûteux en termes de temps.

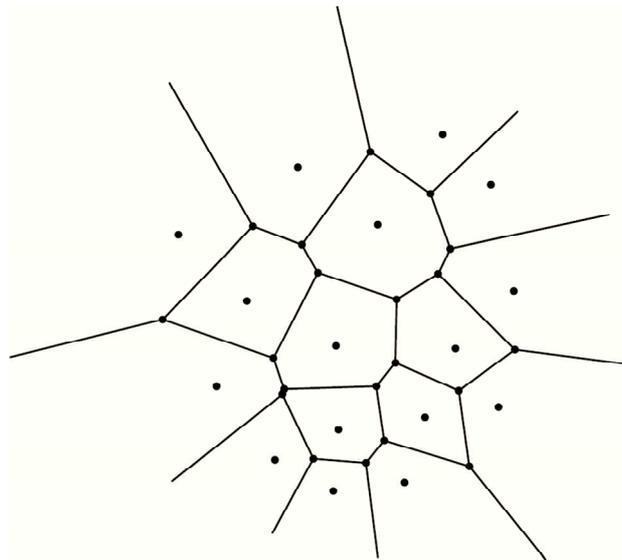


Figure I.1 : Représentation des cellules sous forme de polygones.

- *Représentation sous forme d'un graphe* : les sommets du graphe représentent les cellules et chaque arête reliant deux sommets exprime la relation d'adjacence entre les cellules de ces deux sommets. Cette représentation ne fournit pas d'informations réelles sur la géométrie des cellules.

L'espace de représentation peut être fixé lors de sa création, ou bien variable au cours du temps. Dans ce dernier cas, l'expansion peut être limitée aux contours de la structure, ou même à l'intérieur de la structure.

Ce mode de représentation est principalement utilisé dans le cas de la description au niveau cellulaire, mais ne permet pas de prendre en compte le fonctionnement de l'organe, tant d'un point de vue global que d'un point de vue cellulaire. Ces deux points de vue ont donné deux niveaux de description au modèle.

I.1.2 Niveaux de description du modèle

Il existe différents niveaux de description : microscopique et macroscopique. Les modèles multi-échelles combinent ces deux niveaux pour intégrer toutes les échelles spatio-temporelles pertinentes allant, de l'ADN au système biologique, en passant par l'ARN, les protéines, les cellules, les tissus et les organes.

I.1.2.1 Niveau de description macroscopique

Nous rencontrons ce type de niveau de description dans le cas de la diffusion de fluides, le transport de nutriments entre les cellules et les phénomènes de réarrangements spatiaux des cellules (ex : déplacement des amibes vers une source de nourriture).

Dans l'approche macroscopique, on s'intéresse généralement à la dynamique, au fonctionnement et en particulier aux déformations du système. La dynamique du modèle est modélisée soit par un système d'équations aux dérivées partielles (temps continu) soit par des équations différentielles (temps discret). Dans ce genre de modèle, les cellules peuvent se déplacer dans un espace continu ou discret.

Les propriétés macroscopiques des tissus et organes sont liées aux interactions complexes de leurs composantes microscopiques.

I.1.2.2 Niveau de description microscopique

Le système de particules en interaction et les automates cellulaires sont deux modèles basés sur les phénomènes d'auto-organisation. Ils permettent de modéliser des systèmes biologiques au niveau microscopique.

a. Système de particules en interaction

Le système de particules en interaction est une population de particules (cellules, molécules,...) se déplaçant dans un espace discret en un temps continu. Les particules sont caractérisées par leur position et leur état interne. Les interactions des particules peuvent être locales ou globales, directes ou indirectes (Deutch & Dormann, 2005). Le système de particules en interaction est un modèle stochastique. Chaque particule possède un état interne à un instant t , l'évolution au cours du temps de cet état interne est déterminée avec une certaine probabilité. Cette probabilité dépend des états d'un ensemble de particules voisines. En l'absence d'interaction, l'état de chaque particule évolue comme une chaîne de Markov.

b. Automate cellulaire

Un automate cellulaire est un système dynamique discret dans le temps et l'espace avec des états discrets. Son champ d'application est vaste : simulation du corps humain, d'une avalanche, de propagation de feux de forêt, de trafic urbain, ...

L'automate cellulaire de base consiste en une grille régulière de cellules contenant chacune un état choisi parmi un ensemble fini d'états, et qui évolue au cours du temps. L'état d'une cellule à une date donnée est fonction de l'état à la date précédente d'un nombre déterminé de cellules dites voisines, définies par un système d'interaction. A chaque unité de temps, les mêmes règles locales sont appliquées à toutes les cellules de la grille, produisant ainsi une nouvelle génération de cellules dépendant entièrement de la génération précédente. L'ensemble de toutes les règles d'évolution locales forment la fonction de transition globale de l'automate cellulaire.

Le nombre de règles de l'automate cellulaire augmente de façon exponentielle en fonction du nombre d'états et de la taille du système d'interaction (nombre de cellules voisines considérées).

I.1.3 Type du modèle

Du fait des différentes visions des biologistes, des biochimistes, des informaticiens, des physiciens et des cliniciens, deux types de modèles sont proposés : discrets et continus. Les informaticiens et les biologistes sont plus enclins à voir le monde comme une construction d'entités discrètes. D'autre part, les physiciens, les mécaniciens et les électriciens ont tendance à modéliser le monde comme des quantités continues.

(Berec, 2002) a proposé une classification des modèles de morphogénèse (voir Tableau I.1) en prenant en compte l'aspect discret ou continu de l'espace, du temps et des états cellulaires. Cette classification a été complétée par (Deutchsh & Dormann, 2005).

Approches du modèle	Espace	Temps	Etat des cellules
Equations aux dérivées partielles	continu	continu	continu
Ensemble de règles	continu	continu	discret
Equations intégrales-différentielles	continu	discret	continu
Ensemble de règles	continu	discret	discret
Equations différentielles ordinaires	discret	continu	continu
Système de particules en interaction	discret	continu	discret
Equations différentielles, modèles de Boltzmann	discret	discret	continu
Automate cellulaire, lattice-gaz	discret	discret	discret

Tableau I.1 : Classification des modèles de génération de formes (Deutchsh & Dormann, 2005).

Parmi ces modèles,

- Les modèles développés par (Turing, 1952) ont été très exploités. Meinhardt a proposé une extension de ce modèle en intégrant la notion de différenciation des cellules (Meinhardt, 1976). L'état d'une cellule est caractérisé par la concentration de quatre morphogènes (substances chimiques) qui déterminent si la cellule est en phase de différenciation ou pas, si elle appartient à la structure ou pas. Ce modèle combine des composantes continues (les morphogènes diffusent dans un espace continu) et discrètes (la différenciation cellulaire est décrite par l'état discret de la cellule).
- D'autres modèles de diffusion-réaction sont proposés pour modéliser la croissance des coraux (Kaandorp, 1992), la diffusion dans ces modèles est limitée par la quantité des nutriments qui entourent chaque cellule et qui sont consommés pendant la croissance. L'expansion se fait dans la direction où la concentration des nutriments est la plus importante.
- (Witten & Sander, 1983) ont proposé un modèle d'agrégation avec une diffusion limitée. Le principe de ces modèles est le même que celui des précédents modèles. La seule différence réside dans le fait que les cellules peuvent se déplacer vers la direction des nutriments (réarrangement des cellules).

Au delà de l'aspect continu ou discret des modélisations présentées, les travaux peuvent présenter des niveaux de description différents.

I.1.4 Orientation du modèle

Il existe deux types de modèles : les modèles orientés structure et les modèles orientés espace.

Dans les modèles orientés structure, on cherche généralement à décrire où se situe chaque composante de la structure dans l'espace. Les modèles orientés structure se focalisent sur les composantes qui contribuent à former une structure. (Kenneth, et al., 2003) qualifient cette approche d'une approche grammaticale car elle est guidée par des règles de réécriture

grammaticales qui gèrent la croissance cellulaire. Ces règles grammaticales ont été introduites pour la première fois par (Lindenmayer, 1968), qui créa les L-Systèmes pour décrire des formes complexes telles que la structure des arbres. Un L-système est un groupe de règles et de symboles utilisé pour la modélisation du développement des plantes. Ces règles sont appliquées de manière itérative, en partant d'un symbole initial. Elles permettent de construire une chaîne de symboles qui sera modifiée à chacune des étapes de la modélisation.

Un L-système est défini par :

- un ensemble de symboles ;
- un ensemble de constantes ;
- un axiome (ou initiateur) ;
- un ensemble de règles de production.

Prenons l'exemple de la modélisation d'une algue. Le modèle est le suivant :

- Symboles : A B
- Constantes : aucune
- Axiome : A
- Règles : $A \rightarrow AB$ et $B \rightarrow A$

Le L-système évolue de la manière suivante :

- $t = 0$: A
- $t = 1$: AB
- $t = 2$: AB A
- $t = 3$: AB A AB
- $t = 4$: AB A AB AB A
- $t = 5$: AB A AB AB A AB A AB
- $t = 6$: AB A AB AB A AB A AB AB A AB AB A

L'interprétation biologique de ce système consiste à dire qu'une tige (A) crée une tige (A) et un nœud (B). Un nœud (B) crée une tige (A). La Figure I.2 montre l'interprétation graphique de ce système.

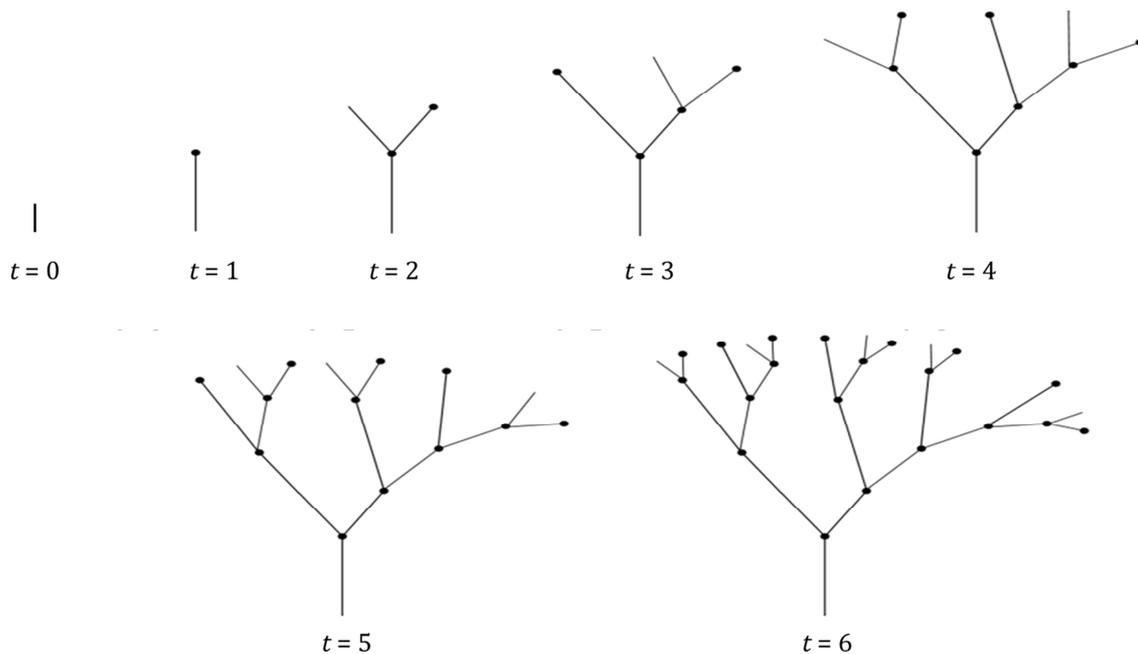


Figure I.2: Evolution du modèle de l'algue.

Dans les modèles orientés espace, on décrit ce qui se trouve dans chaque point de l'espace (état de chaque cellule). Par exemple, les automates cellulaires sont des modèles orientés espace.

Certains modèles orientés espace et certains orientés structure sont équivalents. (Stauffer & Sipper, 1998) ont montré l'équivalence entre les automates cellulaires et les L-systèmes dans le cas de l'étude des phénomènes auto-réplicatifs.

I.1.5 Mode de communication

Il existe deux modes de communication :

- *Par interaction* : dans ce mode de communication, les cellules interagissent localement entre elles (communication endogène) ou avec leur environnement (exogène). Ce mode de communication est fréquent dans le cas des modèles microscopiques (automates cellulaires, systèmes de particules en interaction,...).
- *Par transfert d'information* : l'information est transférée d'une composante parent à sa descendance.

I.2 Problèmes à résoudre

Actuellement, un projet de modélisation et de simulation d'un être humain synthétique se déroule en collaboration entre le LIMOS, le LPC et l'entreprise IBC. Ce projet consiste à produire des modèles biologiques.

Pour proposer des modèles multi-échelles des tissus et des organes couvrant de multiples fonctions, il est important de considérer les phénomènes d'auto-organisation. Ces modèles combinent différents niveaux de description : microscopique et macroscopique. La fonction du système biologique (tissu, organe, ...) est modélisée au niveau macroscopique. La structure de l'organe (forme, volume,...) est modélisée au niveau microscopique, où l'organe est généré à partir d'un ensemble de cellules initialement identiques qui sont en croissance cellulaire, et qui interagissent entre-elles et leur environnement au cours du temps.

La *plateforme Générique de Modélisation et de Simulation* (PGMS) utilise les automates cellulaires pour générer dynamiquement la structure des organes et des tissus et les équations différentielles pour modéliser leurs fonctions. Une des difficultés principales de cette modélisation réside dans le choix des nombreux paramètres de ces modèles. Actuellement, le choix de ces paramètres est réalisé par le concepteur de cette plateforme.

Afin de permettre une meilleure maîtrise pour la génération d'organes humains, le principal objectif de cette thèse est de proposer des méthodes permettant, d'une part, d'identifier les paramètres des équations différentielles modélisant la fonction des organes. D'autre part, ces méthodes doivent permettre à l'automate cellulaire de sélectionner la fonction de transition nécessaire pour son évolution lors de la modélisation de la structure de l'organe.

Ces problèmes sont des problèmes d'optimisation difficile du fait du grand nombre de paramètres à identifier. Ces nombreux paramètres sont dus à la complexité des éléments biologiques modélisés et à la complexité de l'équilibre local et global mis en place entre ces différents éléments. Dans ces travaux, nous nous focalisons sur les niveaux de description macroscopique et microscopique, indépendamment l'un de l'autre.

Une façon pertinente de résoudre ces problèmes difficiles est d'utiliser des méthodes d'optimisation à base de métaheuristique. Ces méthodes ont l'avantage de s'adapter aux différents problèmes d'optimisation continus et discrets.

Dans le cas de la modélisation microscopique par automate cellulaire, ces méthodes seront appliquées pour la génération de formes 2D et 3D. Ces formes peuvent être stéréotypées (triangle, carré, losange, cercle, pyramide, sphère, ...) ou quelconques.

Selon le problème traité (discret ou continu), différents codages et différents systèmes de voisinage des métaheuristiques seront proposés, ce qui permet de mieux exploiter l'espace de recherche et de converger vers la solution souhaitée. Des chaînages de ces méthodes entre elles pour mieux exploiter leurs capacités d'exploration et de diversification seront proposées.

Les résultats théoriques qui seront obtenus pourraient avoir des applications non seulement dans le domaine de la génération automatique d'organes, mais aussi dans les nombreux domaines faisant intervenir des modèles à base d'automates cellulaires (sciences du vivant, physique des matériaux, trafic urbain, ...).

Nous résumons la problématique de cette thèse dans la Figure I.3.

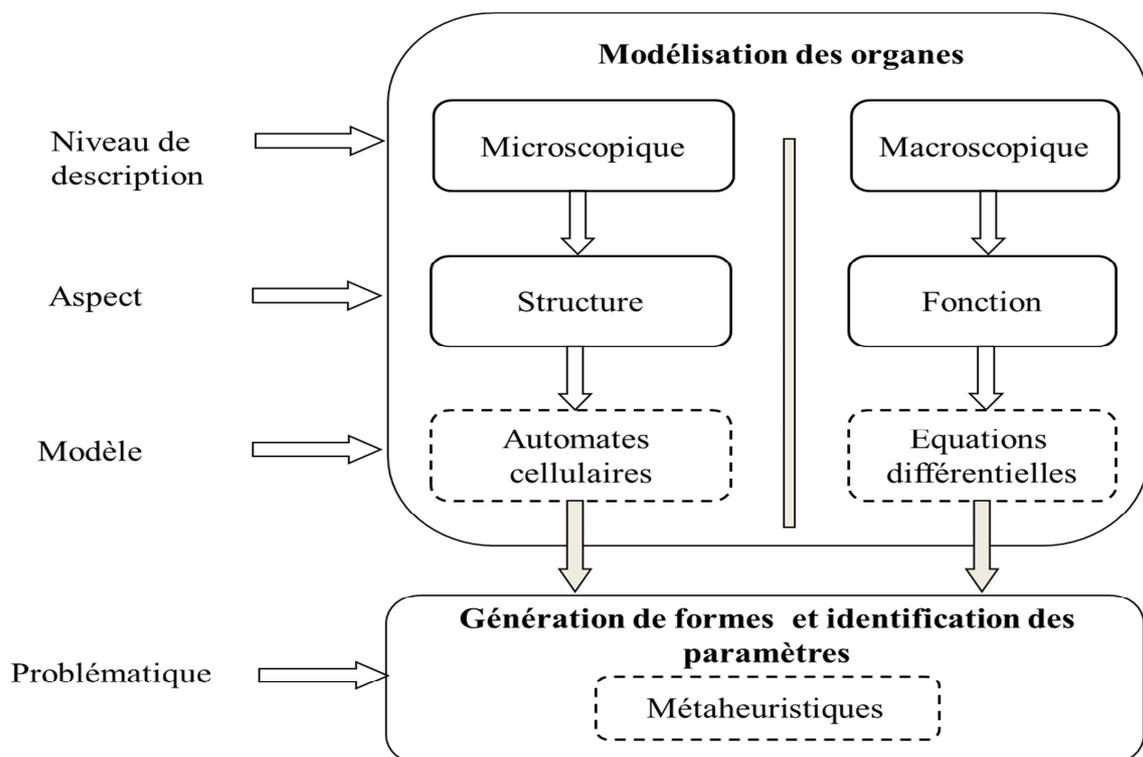


Figure I.3 : Problématique.

Dans la section suivante, nous présentons les méthodes d'optimisation qui seront utilisées à la fois aux niveaux macroscopique et microscopique.

I.3 Méthodes d'optimisation

Un problème d'optimisation combinatoire est généralement caractérisé par un ensemble fini de solutions admissibles Ω et une fonction objectif $f: \Omega \rightarrow \mathbb{R}$ associant une valeur à chaque solution admissible. La résolution du problème consiste à déterminer une solution de Ω minimisant ou maximisant la fonction objectif f .

Les problèmes d'optimisation sont divers (Dréo, et al., 2003) :

- *Problèmes d'optimisation mono-objectif* : ils ont pour but de trouver l'optimum d'une seule fonction objectif. Cet optimum correspond à la meilleure solution au sens d'un critère de comparaison. En général, on le définit comme la valeur minimale d'une fonction.
- *Problèmes d'optimisation multi-objectif* : plusieurs objectifs contradictoires doivent être considérés en même temps. L'objectif de la méthode d'optimisation est de trouver un ensemble d'optima qui forment une surface de compromis pour les différents objectifs du problème, appelée *front de Pareto*.
- *Problèmes d'optimisation multi-modale* : il faudra déterminer l'ensemble des optima globaux et/ou locaux.
- *Problèmes d'optimisation dynamique* : la fonction objectif varie dans le temps, il est donc nécessaire d'approcher l'optimum à chaque pas de temps.

Les problèmes d'optimisation peuvent être *discrets* lorsque les variables sont discrètes ou *continus* si les variables sont continues. En pratique, on rencontre également des *problèmes mixtes*, qui comportent à la fois des variables discrètes et continues (Dréo, et al., 2003).

De nombreuses méthodes de résolution ont été développées en recherche opérationnelle et en intelligence artificielle. Ces méthodes peuvent être classées en deux grandes catégories : les *méthodes exactes* et les *méthodes approchées*.

Le principe essentiel d'une *méthode exacte* consiste généralement à énumérer l'ensemble des solutions de l'espace de recherche. Une technique pour détecter le plus tôt possible les échecs (calculs de bornes) est utilisée pour améliorer l'énumération des solutions (Hao, et al., 1999). Parmi les méthodes exactes, nous pouvons citer : *la programmation linéaire*, *la programmation dynamique* (Bellman, 1957), *la méthode de séparation et évaluation* (Nemhauser & Wolsey, 1988), *les méthodes polyédrales* (Augerat, 1995)...

La résolution des problèmes d'optimisation difficile avec des méthodes exactes est très coûteuse en temps, parfois leur utilisation est impossible. Face à ces problèmes d'optimisation difficile, des *méthodes approchées* peuvent être utilisées pour déterminer une solution approchée et réalisable du problème, qui n'est pas nécessairement optimale.

Nous distinguons deux types de méthodes approchées. Des méthodes ciblées sur un problème particulier (cas des *heuristiques*) et des méthodes conçues pour résoudre n'importe quel problème d'optimisation difficile, discret ou continu (cas des *métaheuristiques*).

I.3.1 Heuristiques

Une heuristique est définie comme une règle d'estimation, une stratégie, une astuce, une simplification, ou toute autre sorte de système qui limite la recherche des solutions dans l'espace des solutions possibles. (Newell, et al., 1957) ont montré qu'une heuristique peut résoudre un problème donné, mais elle n'offre pas la garantie de le faire, selon (Romanycia, 1985).

De nombreuses heuristiques ont été développées au fil du temps. Bien qu'elles soient généralement conçues pour résoudre un problème en particulier, la plupart d'entre-elles peuvent être classées en deux groupes distincts (Soriano & Gendreau, 1997) :

- Le premier regroupe les approches constructives, telles que les algorithmes gloutons, qui s'appliquent généralement à des problèmes d'optimisation dans lesquels nous prenons un ensemble de choix dans le but d'arriver à une solution optimale (Chvatal, 1979; Clarkson, 1983). L'idée d'un algorithme glouton est de faire chaque choix de manière optimale au niveau local (Cormen, et al., 1990). La solution est construite sans jamais remettre en

question les choix précédents. Les heuristiques constructives sont généralement très rapides, mais la solution qu'elles procurent est généralement d'une qualité pas très satisfaisante.

- Le deuxième groupe d'heuristiques regroupe des méthodes d'exploration locale (ou amélioration itérative). Ces méthodes partent généralement d'une solution initiale réalisable et ensuite explorent son voisinage par une séquence de modifications qui améliorent cette solution. Les méthodes d'exploration locale sont généralement plus puissantes que les heuristiques constructives, mais elles sont plus coûteuses, en termes de ressources informatiques. Toutefois ces familles d'heuristiques restent incapables de progresser au-delà du premier optimum local rencontré.

1.3.2 Métaheuristiques

Les métaheuristiques sont des méthodes d'optimisation qui sont utilisées pour résoudre des problèmes d'optimisation difficile lorsque les méthodes exactes échouent. Elles sont généralement inspirées par des analogies avec la physique (recuit simulé, ...), avec la biologie (algorithmes évolutionnaires, recherche tabou,...) ou avec l'éthologie (essais particuliers, colonies de fourmis,...). Les métaheuristiques s'appliquent généralement aux problèmes de type discret, mais elles s'adaptent très facilement aux problèmes continus. La plupart des métaheuristiques sont stochastiques. Le principal avantage des métaheuristiques réside dans la possibilité de contrôler le temps de calcul (Grangeon, 2001; Dréo, et al., 2003).

Nous distinguons deux types de métaheuristiques :

- *les méthodes à voisinage* ou de *recherche locale* ou à *individu* : ces méthodes consistent à faire évoluer une seule solution à la fois.
- *les méthodes à population* : ces méthodes consistent à utiliser un ensemble de solutions simultanément pour améliorer l'exploration de l'espace de recherche.

D'autres méthodes appelées « *hybrides* » combinent les deux types de métaheuristiques.

1.3.2.1 Méthodes à individu

Elles réalisent un processus itératif, qui consiste à remplacer la solution courante par une de ses solutions voisines (choisie selon un système de voisinage) en tenant compte de la fonction objectif à optimiser. Les méthodes à voisinage se distinguent les unes des autres par le système de voisinage utilisé et la stratégie de parcours de ce système de voisinage.

Parmi ces méthodes, nous citons: la recherche locale simple (déterministe ou stochastique), la recherche locale itérée, le recuit simulé et la recherche tabou.

a. Recherche locale

Les étapes d'une simple recherche locale (cas de minimisation de la fonction objectif) sont données par l'Algorithme I.1. Le principe de la méthode de recherche locale consiste, à partir d'une solution initiale s , à choisir une solution s' dans le voisinage V de s qui améliore la fonction objectif H . On peut décider soit d'examiner toutes les solutions du voisinage et prendre la meilleure de toutes, soit d'examiner un sous-ensemble du voisinage.

Le choix du système de voisinage est important, ainsi que la procédure utilisée pour explorer les solutions voisines, appelée *mécanisme de visite* du système de voisinage V . Lorsqu'une solution représente la meilleure solution dans le système de voisinage, le mécanisme de visite est *déterministe*. Dans le cas où la solution voisine est choisie uniformément et aléatoirement, le mécanisme de recherche est *stochastique*.

Cette méthode a l'avantage d'être simple à implémenter, mais elle converge très rapidement vers le premier optimum local rencontré. Pour pallier ce problème, des méthodes, à savoir la recherche locale itérée, le recuit simulé et la recherche tabou, autorisent de temps à autre une certaine dégradation de la solution trouvée, afin de mieux explorer tout l'espace de recherche.

Algorithme I.1 : Algorithme de principe de la recherche locale stochastique

1. Générer une solution initiale s
 2. **Répéter**
 3. Choisir uniformément et aléatoirement $s' \in V(s)$
 4. **Si** $H(s') \leq H(s)$ **alors**
 5. $s \leftarrow s'$
 6. **Fin Si**
 7. **Jusqu'à ce que** $H(s') \geq H(s) \forall s' \in V(s)$
-

b. Recherche locale itérée

La recherche locale est une méthode d'optimisation qui permet d'explorer le voisinage d'une solution initiale. Elle se fige généralement sur le premier optimum local rencontré. Une façon simple de sortir de cet optimum local est de recommencer la recherche stochastique à partir d'une nouvelle solution initiale et de réaliser des recherches stochastiques successives (Fleury, 1993; Lourenço, et al., 2001). La recherche locale itérée ou *algorithme du Kangourou* (Fleury, 1993) suit ce principe. Le principe de cette méthode est donné par l'Algorithme I.2. Après une recherche locale avec un système de voisinage V , si la valeur de la fonction objectif n'a pas évolué depuis A itérations, n'importe quelle dégradation de la solution courante est acceptée en utilisant le système de voisinage W . Le choix d'une solution voisine dans le système W est appelée « *Saut* ».

Il existe plusieurs possibilités pour choisir le système de voisinage W . Le système de voisinage W peut être le même que le système de voisinage V . Il peut être également une application successive du système de voisinage V . Il peut être défini de façon complètement indépendante du système de voisinage V .

La recherche locale itérée converge avec une probabilité de 1 vers les optima globaux si le système de voisinage W respecte la condition d'accessibilité. La preuve de cette convergence repose sur le fait que la recherche locale construit une chaîne de Markov où chaque état conduit à un état absorbant. L'ensemble des états absorbants constituent l'ensemble des optima globaux (Fleury, 1993).

Pour calculer la longueur du palier A , (Fleury, 1993) a proposé la formule suivante :

$$A \approx \frac{\ln(1 - \rho)}{\ln\left(1 - \frac{1}{\text{Card}(V)}\right)} \quad (1)$$

ρ désigne la probabilité qu'il n'existe pas de solutions meilleures que la solution courante au moment du saut et $\text{Card}(V)$ la taille du système de voisinage V .

La formule de calcul de A précédente peut être simplifiée, lorsque le $\text{Card}(V)$ est très grand, comme suit :

$$A \approx \text{Card}(V) \ln(1 - \rho) \quad (2)$$

Algorithme I.2 : Algorithme de principe de la recherche locale itérée

-
1. Soient : s solution initiale choisie aléatoirement et $A > 0$ la longueur du palier
 2. $k \leftarrow 0$
 3. $s^* \leftarrow s$
 4. **Répéter**
 5. **Si** $k < A$ **alors** (Recherche locale)
 6. Choisir uniformément et aléatoirement $s' \in V(s)$
 7. **Si** $H(s') \leq H(s)$ **alors**
 8. **Si** $H(s') < H(s)$ **alors**
 9. $k \leftarrow 0$
 10. **Si** $H(s') < H(s^*)$ **alors**
 11. $s^* \leftarrow s'$
 12. **Fin Si**
 13. **Fin Si**
 14. $s \leftarrow s'$
 15. **Fin Si**
 16. $k \leftarrow k+1$
 17. **Sinon** (Pas d'améliorations depuis A itérations)
 18. Choisir uniformément et aléatoirement $s' \in W(s)$
 19. **Si** $H(s') \neq H(s)$ **alors**
 20. $k \leftarrow 0$
 21. **Fin Si**
 22. **Si** $H(s') < H(s^*)$ **alors**
 23. $s^* \leftarrow s'$
 24. **Fin Si**
 25. $s \leftarrow s'$
 26. **Fin Si**
 27. **Jusqu'à** critère d'arrêt
-

c. Recuit simulé

L'algorithme du recuit simulé est un algorithme stochastique proposé par (Kirkpatrick, et al., 1983). Le recuit simulé est une analogie au recuit physique des solides utilisé en métallurgie. Pour améliorer la qualité d'un solide, on part d'un état de la matière à haute température, puis on diminue progressivement en marquant des à chaque baisse. Ceci conduit la matière à retrouver sa forme solide tout en cherchant à atteindre un état d'énergie libre minimale qui correspond à une structure stable du métal. Dans le cas de la métaheuristique, l'énergie correspond à la fonction objectif.

Le recuit simulé est une procédure évolutive qui permet d'éviter de rester dans des optima locaux de la fonction objectif, en autorisant occasionnellement une diminution de cette fonction. Il s'appuie sur l'algorithme de Metropolis utilisé pour simuler l'évolution d'un système physique vers son équilibre thermodynamique à une température T donnée (Metropolis, et al., 1953) : on part d'une solution initiale que l'on perturbe, si cette perturbation entraîne une diminution $\Delta H = H(s') - H(s)$ de la fonction objectif (énergie dans le système physique) alors cette solution est acceptée, dans le cas où la perturbation entraîne une augmentation ΔH de la fonction objectif, elle est acceptée ; avec une probabilité $\exp(-\Delta H/T)$. En appliquant successivement cette règle de Metropolis, une séquence de solutions est engendrée qui constitue une chaîne de Markov où chaque solution dépend de la solution qui la précédait. Lorsque la longueur de cette chaîne est infinie (de longueur suffisante en pratique), le système atteint son équilibre thermodynamique (Aarts & Van Laarhoven, 1985; Romeo & Sangiovanni-Vincentelli, 1985).

L'évolution de l'algorithme est contrôlée par un facteur « Température » par analogie à la température du recuit réel. Lorsque la température T est élevée, $\exp(-\Delta H/T)$ tend vers 1, ce qui signifie qu'il y a de fortes chances d'accepter les solutions dégradantes. L'algorithme dans ce cas est équivalent à une marche aléatoire dans l'espace de recherche. La probabilité d'accepter les solutions dégradantes décroît avec la température. A basse température, $\exp(-\Delta H/T)$ tend vers 0, la probabilité d'accepter des dégradations est donc faible. La température est contrôlée par une fonction décroissante, qui définit un schéma de refroidissement. Ce schéma de décroissance peut être classé en trois classes :

Réduction par paliers ou recuit simulé homogène : la décroissance de la température se fait par paliers. Un palier correspond à un certain nombre d'itérations où la température est maintenue. Le processus est stoppé lorsque le système est figé. Un système est figé lorsque sa température est nulle, ou bien qu'aucune amélioration de la fonction objectif n'a été réalisée pendant le palier.

Réduction continue ou recuit simulé inhomogène : la température décroît à chaque itération du recuit simulé. Le recuit simulé inhomogène est un recuit simulé où les paliers sont de longueur égale à 1.

Réduction non-monotone : la température décroît à chaque itération, avec des augmentations occasionnelles.

Le principe d'un recuit simulé est présenté dans l'Algorithme I.3.

Algorithme I.3 : Algorithme de principe du recuit simulé.

1. Choisir une solution initiale s
 2. Choisir une température initiale T
 4. **Répéter**
 5. **Tant que** l'équilibre thermodynamique n'est pas atteint à la température T **faire**
 6. Choisir aléatoirement et uniformément une solution voisine $s' \in V(s)$
 7. **Si** $H(s') \leq H(s)$ **alors**
 8. $s \leftarrow s'$
 9. **Sinon**
 10. Choisir aléatoirement $\alpha \in [0,1]$
 11. **Si** $\alpha \leq \exp[-\Delta H / T]$ **alors**
 12. $s \leftarrow s'$
 13. **Fin si**
 14. **Fin si**
 18. **Fin Tant que**
 19. Décroître la température T
 20. **Jusqu'à** critère d'arrêt
 21. Retourner s
-

L'efficacité du recuit simulé dépend de paramètres tels que sa température initiale, le facteur de décroissance de la température et la taille du palier pour chaque température.

Plusieurs réglages ont été proposés pour le choix de ces paramètres (Romeo & Sangiovanni-Vincentelli, 1985; Siarry, 1995). Le choix de la température initiale est généralement basé sur des hypothèses de départ où l'algorithme accepte toutes les solutions. Pour la décroissance de la température, plusieurs approches ont été proposées.

Il existe plusieurs variantes du recuit simulé :

- *le recuit microcanonique*. L'énergie totale d'un système physique isolé est constante, quel que soit son évolution dynamique. En se basant sur cette propriété, (Creutz, 1983) a proposé une variante du recuit simulé qui conserve son énergie totale E au cours de son évolution.

Cette énergie est la résultante d'une énergie cinétique E_c et d'une énergie potentielle E_p . L'énergie cinétique joue le même rôle que la température du recuit et doit être positive. L'objectif est de minimiser l'énergie potentielle. L'algorithme accepte les mouvements vers les solutions avec une basse énergie potentielle. Les solutions avec une haute énergie potentielle sont acceptées lorsque $\Delta E < E_p$.

- *la méthode du seuil* (Dueck & Scheuer, 1989). La principale différence entre le recuit simulé et la méthode du seuil est le critère d'acceptation des solutions dégradantes. Dans le recuit, une solution dégradante est acceptée avec une certaine probabilité. Dans le cas de la méthode du seuil, cette solution est acceptée si la dégradation de la fonction objectif qu'elle engendre ne dépasse pas un certain seuil.

d.Recherche tabou

La recherche tabou a été développée par (Glover, 1986) et indépendamment par (Hansen, 1986) sous l'appellation de « Steepest ascent midest descent ». Le principe de cette méthode est simple. Partant d'une solution initiale quelconque, une série de solutions voisines est générée en perturbant la solution initiale. La meilleure solution voisine est ensuite retenue pour l'itération suivante.

L'originalité de cette méthode réside dans le fait que la meilleure solution voisine est retenue même si elle est plus mauvaise que la solution courante, ce qui permet à l'algorithme d'accepter des dégradations de la fonction objectif et évite ainsi le blocage dans un optimum local. Néanmoins, dès qu'une solution voisine plus mauvaise que la solution courante est acceptée, un risque de cycle existe dans la procédure. Pour éviter ce phénomène, les perturbations qui renvoient sur une solution déjà visitée sont provisoirement interdites. Une liste Tabou T des solutions déjà visitées est ainsi créée. A chaque itération, après avoir généré une solution voisine de la solution courante, l'algorithme vérifie que celle-ci n'appartient pas à la liste Tabou T . Dans le cas où cette solution voisine est dans la liste tabou, cette solution voisine est rejetée. Le principe de cette méthode est décrit par l'Algorithme I.4.

Algorithme I.4 : Algorithme de principe de la recherche Tabou

1. Choisir une solution initiale s
 2. $s^* \leftarrow s$
 3. **Répéter**
 4. Créer une liste de solutions candidates dans le voisinage $V(s)$
 5. Choisir la meilleure solution s' non tabou parmi les solutions candidates
 6. **Si** $H(s') < H(s^*)$ **alors**
 7. $s^* \leftarrow s'$
 8. **Fin si**
 9. Mettre à jour la liste tabou
 10. $s \leftarrow s'$
 11. **Jusqu'à** critère d'arrêt
 12. Retourner s^*
-

I.3.2.2 Méthodes à population

Cette partie est consacrée aux méthodes à population. Nous présentons :

- les algorithmes génétiques ;
- les essais particuliers ;
- les colonies de fourmis.

a. Algorithmes génétiques

Les algorithmes génétiques sont des méthodes d'optimisation fondées sur la sélection naturelle, la reproduction et l'adaptation. Ils sont introduits pour la première fois par John Holland en 1975 (Holland, 1975). La popularité des algorithmes génétiques est due aux travaux de (Goldberg, 1989).

Un algorithme génétique commence par la génération d'une population de solutions appelées individus ou chromosomes. Cette population évolue durant un certain nombre de générations. A chaque génération, des opérateurs de reproduction (sélection, croisement et mutation) sont appliqués aux individus pour en générer de nouveaux, appelés enfants, à partir d'un ou plusieurs individus appelés parents. Un opérateur de remplacement est ensuite utilisé pour réinsérer les nouveaux individus dans la population de la génération suivante.

L'Algorithme I.5 présente l'algorithme de principe d'un algorithme génétique.

Algorithme I.5 : Algorithme de principe de l'algorithme génétique

1. Soit P_0 une population initiale de p chromosomes
 2. $i \leftarrow 0$
 3. **Répéter**
 4. $i \leftarrow i + 1$
 5. Soit P_i une population vide de chromosomes enfants
 6. **Répéter**
 7. Utiliser l'opérateur de sélection pour choisir deux chromosomes I_1 et I_2 de la population P_{i-1}
 8. Appliquer l'opérateur de croisement sur I_1 et I_2 pour obtenir les enfants O_1 et O_2
 9. Ajouter O_1 et O_2 à P_i
 10. **Jusqu'à P_i contient p chromosomes enfants**
 11. Appliquer l'opérateur de mutation sur certains chromosomes de P_i
 12. $P_i = P_i \cup P_{i-1}$
 13. Sélectionner p chromosomes de P_i et supprimer les p autres
 14. **Jusqu'à critère d'arrêt**
-

Les opérateurs de reproduction sont :

- *La sélection* : c'est un opérateur très important dans la reproduction génétique. Elle permet d'identifier statistiquement les meilleurs individus d'une population en fonction de leur efficacité (au sens de la fonction objectif). En règle générale, la probabilité de survie d'un individu est directement liée à son efficacité au sein de la population.
- *Le croisement* : c'est un opérateur qui permet l'échange d'informations entre les chromosomes par le biais de leur combinaison. La zone de croisement est généralement choisie de façon aléatoire dans les chromosomes. Toutefois, un chromosome sélectionné lors de la reproduction ne subit pas forcément l'action d'un croisement. Ce dernier ne s'effectue qu'avec une certaine probabilité p_c . Plus cette probabilité est élevée, plus la population subira des changements.
- *La mutation* : c'est un opérateur qui permet d'éviter la dégénérescence de la population. Cette dégénérescence peut se traduire par une convergence des individus vers un même optimum local. Cet opérateur a pour but de diversifier la population afin d'examiner d'autres zones de l'espace de recherche. Le rôle de cet opérateur est de modifier aléatoirement, avec une certaine probabilité P_m , la valeur d'un composant du chromosome.

Lors de la génération d'une nouvelle population, il faut établir un compromis entre les chromosomes enfants produits et les chromosomes parents en utilisant un mécanisme de remplacement (insertion) tout en gardant la taille de la population fixe à chaque génération.

b.Essaims particulaires

Les essais particuliers ont été introduits par (Eberhart & Kennedy, 1995) pour résoudre des problèmes d'optimisation continue. Ils sont inspirés du comportement social des animaux évoluant en essaim, tels que les vols groupés d'oiseaux. L'optimisation par essaim particulière est un algorithme évolutionnaire qui utilise une population de particules. Malgré l'intelligence limitée de chaque particule qui ne dispose que d'une connaissance locale de sa situation dans l'essaim, le comportement de l'essaim est relativement complexe. L'information locale et la mémoire de chaque particule sont utilisées pour décider de son déplacement. Des règles simples, telles que « rester proche des autres individus », « aller dans une même direction » ou « aller à la même vitesse », suffisent pour maintenir la cohésion de l'essaim, et permettre la mise en œuvre de comportements collectifs complexes et adaptatifs.

A chaque itération t , chaque particule i possède dans la dimension d une position $x_{id}(t)$ (solution) et une vitesse $v_{id}(t)$ permettant de déterminer son changement de position. La particule a la capacité de se souvenir de sa meilleure position et de la meilleure position de ses particules voisines, appelées informatrices. Pour se déplacer, la particule combine trois tendances : suivre la direction courante de son déplacement $x_{id}(t)$, se diriger vers la meilleure position par laquelle elle est déjà passée l_{id} ou bien se diriger vers la meilleure position de ces informatrices P_{gd} (Kennedy & Eberhart, 1997) selon la formule (3).

$$\begin{cases} x_{id}(t+1) = x_{id}(t) + v_{id}(t) \\ v_{id}(t+1) = v_{id}(t) + \varphi(p_{id} - x_{id}(t)) + \varphi(p_{gd} - x_{id}(t)) \end{cases} \quad (3)$$

où φ désigne une valeur aléatoire positive choisie pour chaque particule i dans chaque dimension d . L'algorithme de principe des essais particuliers est présenté dans l'Algorithme I.6.

Algorithme I.6 : Algorithme de principe de l'optimisation par essais particuliers

1. Initialiser la position $v_{id}(0)$ de chaque particule i
 2. Initialiser la vitesse $x_{id}(0)$ de chaque particule i
 3. Initialiser la meilleure position l_{id} de chaque particule i
 4. Déterminer la meilleure position de l'essaim P_{gd}
 5. **Répéter**
 6. **Pour** chaque particule i **faire**
 7. Mettre à jour sa vitesse $v_{id}(t)$
 8. Mettre à jour sa position $x_{id}(t)$
 9. Mettre à jour la meilleure position de la particule l_{id}
 10. Mettre à jour la meilleure position de l'essaim P_{gd}
 11. **Fin Pour**
 12. **Jusqu'à critère d'arrêt**
-

c.Colonies de fourmis

En 1992, Marco Dorigo a proposé une méthode d'optimisation combinatoire qui utilise le principe des colonies de fourmis (Dorigo, 1992). Les fourmis possèdent la capacité de sélectionner le chemin le plus court pour aller de leur nid vers une source de nourriture grâce au dépôt et au suivi des pistes de phéromone, qui est une substance s'évaporant au cours du temps. En effet, les fourmis déposent la phéromone à l'aller et au retour vers la source de nourriture. Au départ les fourmis choisissent les chemins de façon aléatoire. Au fil du temps, les fourmis ont tendance à emprunter le chemin où il y a plus de phéromone, donc le plus court.

L'Algorithme I.7 décrit le principe de cette méthode. L'espace des solutions est donné sous forme d'un graphe de m sommets et arêtes. Le voyageur de commerce est un bon exemple pour

expliquer le principe de cette méthode : les sommets sont les villes et les arêtes désignent les chemins entre les villes. Soit $\tau_{ij}(k)$ la concentration des phéromones sur l'arête (i, j) à l'itération k . A chaque itération de l'algorithme $m-1$ déplacements sont réalisés. Chaque fourmi doit choisir à l'instant t de l'itération k le sommet où elle se déplacera à l'instant $t+1$. Ce déplacement est effectué en fonction d'une probabilité qui dépend de la quantité de phéromone. Cette probabilité est forte lorsque la quantité de phéromone est élevée.

A chaque itération, la quantité de phéromone sur chaque arête est mise à jour selon la formule suivante :

$$\tau_{ij}(k+1) = \rho \tau_{ij}(k) + \Delta \tau_{ij}(k) \quad (4)$$

Avec ρ le coefficient d'évaporation, $\Delta \tau_{ij}(k)$ la quantité de phéromone déposée par toutes les fourmis qui sont passées par l'arête (i, j) à l'itération k . A l'état initial, toutes les arêtes possèdent la même quantité de phéromone.

Algorithme I.7 : Algorithme de principe des colonies de fourmis

1. Initialiser les phéromones et choisir un nombre f de fourmis
 2. $k \leftarrow 0$
 3. **Répéter**
 4. $k \leftarrow k+1$
 5. Construire f solutions en utilisant les quantités $\tau_{ij}(k)$ de phéromones
 6. Mettre à jour les phéromones selon $\tau_{ij}(k+1) = \rho \tau_{ij}(k) + \Delta \tau_{ij}(k)$
 7. **Jusqu'à critère d'arrêt**
-

I.3.2.3 Approches hybrides

Les approches hybrides consistent à combiner différentes méthodes d'optimisation entre elles. Ces combinaisons ont pour objectif d'accélérer la convergence des méthodes. L'origine des travaux hybrides remonte aux travaux de (Glover, 1977; Grefenstette, 1987). Chacun d'entre eux a fait intervenir une méthode de recherche simple (recherche locale).

Une première approche consiste à combiner des méthodes exactes et des métaheuristiques. Ces différentes combinaisons sont bien détaillées dans (Puchinger & Raidl, 2005). Il existe également des couplages métaheuristiques-métaheuristiques (Talbi, 2002). Une classification des méthodes d'hybridation de métaheuristiques est proposée dans (Talbi, 2002; Jourdan, et al., 2009; Jourdan, 2010). Cette classification est basée sur le niveau et mode de coopération. La coopération entre les méthodes d'optimisation peut être de *bas niveau* ou de *haut niveau*. Dans les méthodes de coopération de *bas niveau*, une méthode devient un opérateur d'une autre méthode qui l'englobe. Dans le cas de coopération de *haut niveau*, les méthodes communiquent entre elles tout en gardant leur intégrité (Figure I.4).

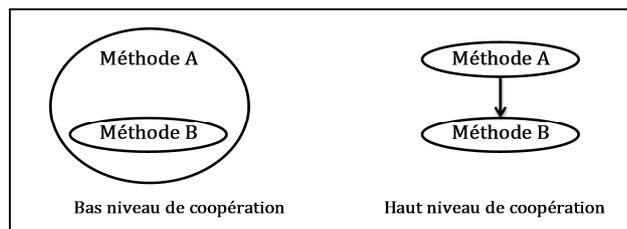


Figure I.4 : Niveaux de coopération entre méthodes d'optimisation (Jourdan, 2010).

(Jourdan, et al., 2009) ont défini deux modes de coopération pour la classification des méthodes hybrides (Figure I.5). Dans le premier mode, appelé « *mode relais* », l'exécution des méthodes d'optimisation se fait de manière séquentielle. Les résultats de la précédente servent de point de départ pour la méthode courante. Dans le deuxième mode, dit « *mode co-évolution* », les méthodes d'optimisation fonctionnent en parallèle.

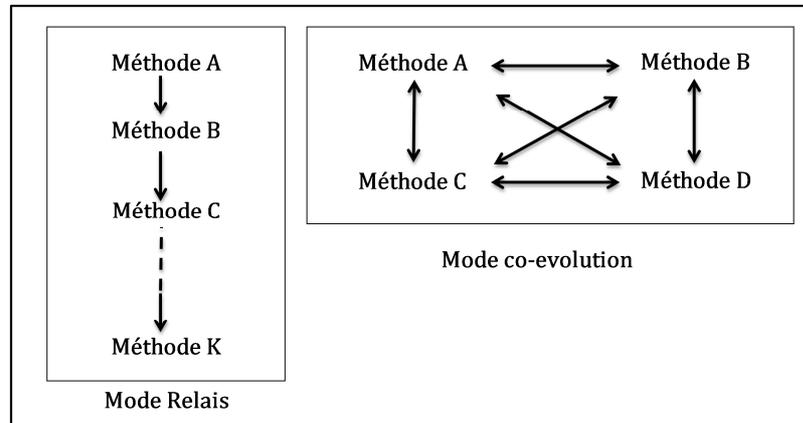


Figure I.5 : Modes de coopération entre méthodes d'optimisation (Jourdan, 2010).

Selon le mode et le niveau de coopération, quatre classes hiérarchiques peuvent être définies : relais de bas niveau, relais de haut niveau, co-évolution de bas niveau et co-évolution de haut niveau (Jourdan, et al., 2009).

a. Relais de bas niveau

Les méthodes relais de bas niveau sont généralement des méthodes à solution unique qui englobent une autre méthode d'optimisation. Le résultat de la méthode globale dépend du résultat de la méthode englobée. Ce type de coopération peut être illustré par l'utilisation d'une méthode approchée pour améliorer une méthode exacte.

b. Co-évolution de bas niveau

Cette classe concerne généralement les méthodes à population dans lesquelles un opérateur (ou plusieurs opérateurs) est remplacé par une autre méthode généralement à solution unique. Les algorithmes mimétiques, qui sont des algorithmes génétiques combinés avec une recherche locale remplaçant l'opérateur de mutation, sont des algorithmes de co-évolution de bas niveau. L'avantage de ce type de coopération est la combinaison de la diversification des méthodes à population et de l'intensification de la méthode à solution unique.

c. Relais de haut niveau

Dans cette classe, les méthodes sont exécutées de façon séquentielle. Chacune des méthodes utilisées garde son intégrité, contrairement aux méthodes de bas niveau. Le résultat de la méthode courante sert à initialiser la méthode suivante. Cette coopération peut permettre de combiner les avantages de chaque méthode utilisée en termes d'exploration de l'espace de recherche et d'intensification en utilisant un algorithme génétique (pour la diversification) suivi d'une recherche locale permettant d'améliorer les solutions obtenues. L'utilisation d'un algorithme glouton pour générer des solutions initiales pour une autre méthode d'optimisation est aussi très répandue.

d. Co-évolution de haut niveau

Cette classe correspond à l'exécution de plusieurs méthodes en parallèle. Les méthodes communiquent entre elles au cours de leur exécution par échange de solutions. Un exemple de

cette classe concerne un algorithme génétique exécuté en parallèle (Tanese, 1987). Il consiste à générer une population de solutions. Cette population est décomposée en plusieurs sous-populations qui sont considérées comme des îles. Un algorithme génétique est appliqué à chaque sous-population. Pour améliorer les solutions de chaque île, des échanges de solutions se font entre elles.

Conclusion

Dans cette thèse, nous nous intéressons à la modélisation des organes humains à deux niveaux :

- *niveau macroscopique* : pour modéliser le fonctionnement d'un organe. Le fonctionnement des systèmes biologiques est généralement décrit par des équations différentielles. Ces équations différentielles sont caractérisées par un ensemble de paramètres.
- *niveau microscopique* : pour modéliser la structure de l'organe (sa forme, son volume,...). La structure et la forme d'un organe résultent des processus dynamiques basés essentiellement sur l'auto-organisation. Parmi les modèles qui tiennent compte de cette caractéristique, nous utilisons les automates cellulaires. Ce sont des modèles discrets qui font émerger des comportements globaux complexes à partir de règles locales simples. Le nombre de règles augmente exponentiellement avec le nombre de types de cellules pris en compte et des interactions cellulaires considérées.

Pour obtenir un organe avec une bonne structure et un bon fonctionnement, il est important de bien paramétrer les modèles. Il va falloir, d'une part, déterminer les règles d'évolution locales de l'automate cellulaire permettant d'obtenir la forme la plus proche de celle de l'organe, d'autre part, identifier les paramètres du système des équations différentielles. Le principal objectif de cette thèse est donc de proposer des méthodes permettant :

- d'identifier et déterminer les paramètres des équations différentielles modélisant la fonction des organes à partir des observations réelles ;
- de sélectionner une fonction de transition de l'automate cellulaire. Ce problème est très complexe (Ganguly, et al., 2003) et la taille de l'espace de recherche augmente exponentiellement avec le nombre d'états des cellules pris en compte et la taille du système d'interaction dans lequel les cellules interagissent.

Pour résoudre ces problèmes d'optimisation continus (cas des équations différentielles) et discrets (cas des automates cellulaires), nous proposons des méthodes à base de métaheuristiques.

Les méthodes proposées seront appliquées, dans un premier temps, pour identifier les paramètres d'un système d'équations différentielles et, dans un second temps, pour rechercher des formes simples et complexes (2D et 3D à 2 et 3 états). Ces travaux serviront par la suite pour modéliser des systèmes biologiques réels.

Chapitre II. MODELISATION MACROSCOPIQUE : PROPOSITION DE METHODES DE RESOLUTION POUR L'IDENTIFICATION PARAMETRIQUE

Introduction

La dynamique et la fonction de certains systèmes biologiques et physiques sont modélisées par des équations différentielles ou des équations aux dérivées partielles. La difficulté principale rencontrée dans ces systèmes réside non seulement dans le choix du modèle mathématique adéquat mais surtout dans le choix des bons paramètres de ce modèle.

L'identification paramétrique d'un système décrit par des équations différentielles est un problème inverse. Ce problème consiste à déterminer les paramètres d'un système à partir d'observations. Une classification de ces problèmes appliqués aux équations de la cinétique d'une réaction catalysée par une enzyme est présentée dans (Gennemark & Wedelin, 2009). Cette classification est basée sur le type d'équations différentielles, le nombre de paramètres du modèle et les conditions initiales.

Le problème d'identification paramétrique peut être formulé comme un problème d'optimisation continu. Il consiste à déterminer les paramètres du modèle permettant de reproduire au mieux le fonctionnement du système étudié. Le but principal de ce travail consiste à proposer des méthodes permettant de résoudre le problème d'identification paramétrique d'un système décrit par des équations différentielles d'une façon générale (système biologique, système physique,...).

Dans ce chapitre, nous proposons des méthodes à base de métaheuristiques (recuit simulé et algorithme génétique) pour résoudre ce problème. Nous proposons un codage où une solution est un vecteur des paramètres à identifier. La fonction objectif proposée est la somme des écarts quadratiques entre les observations réelles et celles du système. Comme les paramètres de certains systèmes respectent certaines contraintes et sont liés entre eux, nous proposons un système de voisinage et un opérateur de mutation qui tiennent compte de ces contraintes.

Ce chapitre est composé de quatre parties :

- Dans la première partie, une présentation du problème d'identification paramétrique des systèmes décrits par équations différentielles est présentée. Pour illustrer ce problème, un système physique oscillant est considéré.
- Le codage de solution, la fonction objectif ainsi que les méthodes proposées pour résoudre ce problème sont présentés dans la deuxième partie.
- Pour valider les méthodes proposées, nous avons appliqué ces méthodes dans un premier temps à des systèmes physiques avec des paramètres entiers. Une énumération de l'espace des solutions a permis d'étudier l'espace de la fonction objectif. Les résultats sont donnés dans la troisième partie.

- Dans la dernière partie, les résultats des méthodes proposées pour identifier les paramètres d'un système physique avec des paramètres continus sont donnés.

II.1 Présentation du problème

Dans cette partie, nous présentons le problème d'identification paramétrique d'un système décrit par des équations différentielles dans le cas général. Ensuite, nous illustrons ce problème dans le cas d'un système physique oscillant.

II.1.1 Problème général

La modélisation de systèmes biologiques par équations différentielles est très répandue. Elle permet de décrire l'évolution et le comportement d'une population de cellules ou de nombreux types d'interactions et échanges entre plusieurs populations de cellules.

Dans le domaine des systèmes d'équations différentielles, il est fréquent de faire des analogies entre l'électricité, la mécanique et les sciences du vivant. Prenons pour exemple la pression artérielle sanguine. Otto Frank en 1899 a modélisé mathématiquement le modèle analogique artériel décrit par William Harvey et par Stephen Hales aux 17^{ème} et 18^{ème} siècles. Il décrit l'hémodynamique (circulation sanguine) des systèmes artériels comme des systèmes électriques composés de résistances, de condensateurs et d'inductances (Figure II.1). Il présente trois modèles appelés de *windkessel* qui veut dire *chambre à pression*. Le premier modèle à deux éléments « 2-élément WK » est composé d'une résistance et d'un condensateur en parallèle. Le deuxième modèle « 3-élément WK » comprend en plus du premier modèle une autre résistance en série. Dans le dernier modèle à quatre éléments « 4-élément WK », la résistance du deuxième modèle est en parallèle avec une inductance. Ces modèles sont décrits par des équations différentielles.

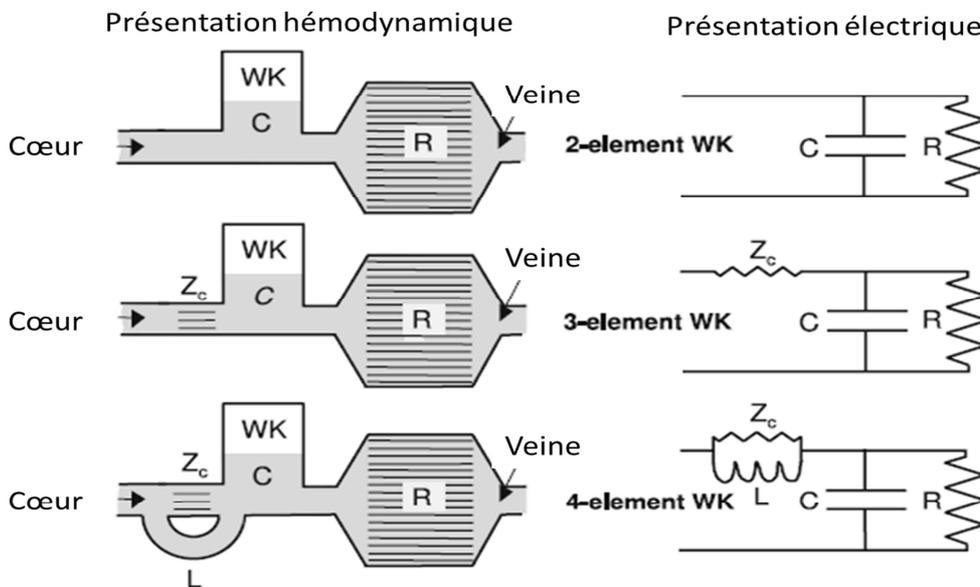


Figure II.1 : Modèles d'hémodynamique de windkessel.

Il existe d'autres modèles faisant appel aux circuits électriques comme les modèles des membranes séparant le milieu intracellulaire du milieu extracellulaire (Hodgkin & Huxley, 1952). Des analogies entre les systèmes électriques et physiques sont fréquentes, comme dans le cas des systèmes oscillants en physique et en électricité. Si l'on prend le système de circulation

sanguine « 3-élément WK », le signal de sortie peut être défini par une équation différentielle du premier ordre, comme suit :

$$S(t) = K(1 - e^{-t/\tau}) = \frac{R}{Z_c + R}(1 - e^{-t/\tau}) \quad (5)$$

Une réponse indicielle (réponse à un échelon unitaire) du système permet de déterminer les valeurs du gain statique K et la constante de temps τ avec $\tau = (R/Z_c)*C$.

Pour reproduire le bon fonctionnement de tels systèmes, il est très important de trouver leurs bons paramètres. L'estimation des paramètres d'un système décrit par un modèle mathématique tel que les équations différentielles à partir des observations ou mesures est un problème inverse.

Ce problème inverse peut être décrit comme suit :

Soient un système décrit par une équation différentielle de I paramètres et $Y(t)$ les observations temporelles du système à différents instants t . L'objectif consiste à déterminer le vecteur des paramètres \hat{p} dans l'espace de recherche Ω qui minimise l'écart entre les observations réelles $Y(t)$ et celles obtenues par le modèle $y(t)$.

II.1.2 Problème d'un système physique

Pour illustrer l'approche proposée, nous considérons un système physique. L'objectif consiste à déterminer les paramètres de ce système qui lui permettent d'osciller et de passer par les points les plus proches des observations temporelles du système réel (Figure II.2).

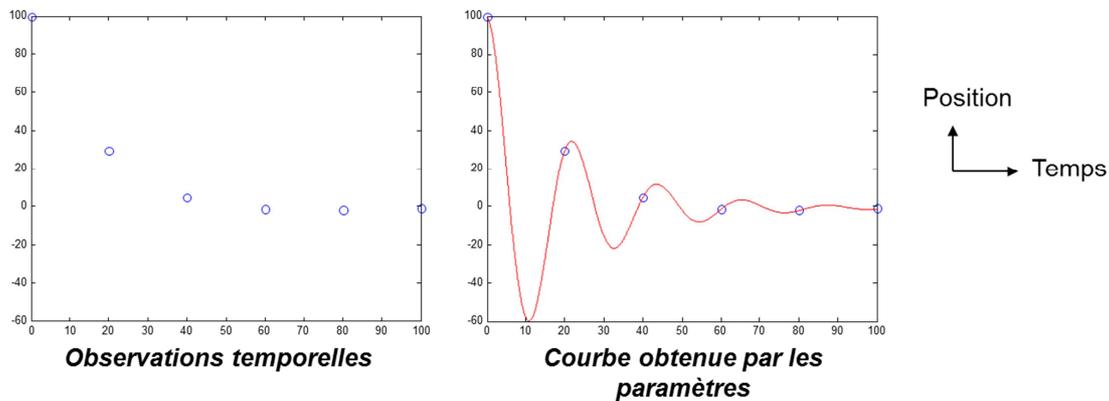


Figure II.2 : Présentation du problème.

Le système physique considéré est un oscillateur composé d'une masse m et d'un ressort avec une constante de raideur k . Ce système est amorti par les frottements de coefficient c . Dans le cas d'un système physique, la dynamique de ce système est décrite par l'équation différentielle du second ordre suivante :

$$\frac{d^2 y(t)}{dt^2} + \frac{c}{m} \frac{dy(t)}{dt} + \frac{k}{m} y(t) = 0 \quad (6)$$

$y(t)$ représente les positions de la masse en fonction du temps.

L'équation caractéristique de cette équation différentielle est $r^2 + 2\lambda r + \omega_0^2 = 0$ avec $\omega_0^2 = \frac{k}{m}$ et

$$\lambda = \frac{c}{2m}.$$

Le système physique peut avoir trois comportements, selon les valeurs du discriminant Δ de cette équation ($\Delta = \lambda^2 - \omega_0^2$) :

- Si $\Delta < 0$: régime oscillatoire (périodique)
- Si $\Delta = 0$: régime critique
- Si $\Delta > 0$: régime apériodique

Le système est au régime périodique si ses paramètres respectent la condition d'oscillation suivante :

$$0 \leq c < 2\sqrt{km} \quad (7)$$

Dans le cas d'un système physique oscillant, les solutions de l'équation différentielle dépendent des conditions initiales et des valeurs des paramètres m , c et k . Les conditions initiales du système concernent la position initiale du système y_0 et sa vitesse initiale y'_0 . Dans notre cas, ces conditions sont connues.

II.2 Proposition de méthodes de résolution

Pour résoudre ce problème inverse, nous proposons des métaheuristiques à base de recuit simulé et algorithme génétique. Nous présentons dans cette partie le codage d'une solution, la fonction objectif et les méthodes d'optimisation (recuit simulé et algorithme génétique).

II.2.1 Codage d'une solution

Dans cette partie, le codage d'une solution dans le cas général est donné. Ensuite, le codage des solutions dans le cas d'un système oscillant est présenté.

II.2.1.1 Problème général

Une solution $P = (p_1, \dots, p_i, \dots, p_I)$ est un vecteur de I paramètres du modèle d'équations différentielles. Chaque paramètre p_i est choisi dans un intervalle $[p_{imin}, p_{imax}]$. La valeur minimale p_{imin} et la valeur maximale p_{imax} de chaque paramètre p_i sont choisies en se basant sur des informations *a priori* du système réel étudié.

Lorsque les conditions initiales du système sont inconnues, elles peuvent être codées dans la solution. Une solution $X = (P, CI)$ est composée de l'ensemble I des paramètres à identifier = $(p_1, \dots, p_i, \dots, p_k)$ et de l'ensemble des h conditions initiales $CI = (CI_1, \dots, CI_i, \dots, CI_h)$.

II.2.1.2 Problème d'un système physique oscillant

Dans le cas d'un système physique, nous considérons trois cas :

- Dans un premier temps, nous supposons que deux paramètres du système sont connus et qu'il faudra juste déterminer un seul paramètre (cas 1) ;
- Dans le deuxième cas (cas 2), nous cherchons à déterminer deux paramètres du système, connaissant un seul paramètre ;
- Dans le dernier cas (cas 3), on cherche à déterminer tous les paramètres du système.

Une solution du système (vecteur des paramètres) est admissible si et seulement si les paramètres de cette solution respectent la condition d'oscillation (7).

a.Cas 1

Dans ce cas, la solution est composée d'un seul paramètre (c). Les paramètres k et m sont connus. Ce paramètre doit respecter la condition d'oscillation précédente.

$$P = (c) \quad (8)$$

b.Cas 2

Dans ce cas, la solution est composée de deux paramètres à identifier, m et c . Le paramètre k est connu. Le paramètre m est choisi en ayant des informations *a priori* sur le système. Le deuxième paramètre est choisi en fonction de la valeur de m et de k .

$$P = (m, c) \quad (9)$$

c.Cas 3

Dans ce dernier cas, la solution regroupe tous les paramètres du système : m , c et k . Les deux premiers paramètres sont choisis de telle sorte qu'ils soient représentatifs. Le dernier paramètre, c , respecte la condition d'oscillation.

$$P = (m, c, k) \quad (10)$$

En résumé, les solutions doivent respecter les contraintes suivantes :

	Cas 1	Cas 2	Cas 3
k	–	–	$k \in [k_{\min}, k_{\max}]$
m	–	$m \in [m_{\min}, m_{\max}]$	$m \in [m_{\min}, m_{\max}]$
c	$c \in [0, 2\sqrt{km}[$	$c \in [0, 2\sqrt{km}[$	$c \in [0, 2\sqrt{km}[$

Tableau II.1 : Contraintes sur les solutions.

II.2.2 Fonction objectif

II.2.2.1 Problème général

La fonction objectif F représente l'écart entre les observations temporelles $Y(t) = (Y_0, Y_1, \dots, Y_n)$ aux instants (t_0, t_1, \dots, t_n) du système et celles obtenues par le modèle avec les paramètres P $y_p(t) = (y_0, y_1, \dots, y_n)$. La plupart des travaux traitant le problème d'identification paramétrique des équations différentielles utilisent le maximum de vraisemblance entre les observations temporelles du système et celles du modèle.

Dans ce travail, nous considérons la somme des carrés des distances euclidiennes entre les deux courbes $Y(t)$ et $y(t)$. Elle est donnée par cette équation :

$$F(P) = \sum_{t=t_0}^{t_n} (y_p(t) - Y(t))^2 \quad (11)$$

Pour calculer cette fonction objectif, il va falloir, connaissant les paramètres P , résoudre les équations différentielles, afin de déterminer la solution $y_p(t)$.

II.2.2.2 Problème d'un système oscillant

Le Tableau II.2 résume les solutions de l'équation différentielle en fonction des paramètres du système physique dans le cas du régime périodique.

Régime périodique $\Delta < 0$				
$\lambda \neq 0$		$\lambda = 0$		
$\omega_1^2 = \omega_0^2 - \lambda^2$		$y_0 = 0$	$y_0 > 0$	$y_0 < 0$
Amplitude A	$\frac{y_0}{\cos \phi}$	$-\frac{\omega_0}{y'_0} \cdot \sin \phi$	$-\frac{y_0}{\cos \phi}$	$-\frac{y_0}{\cos \phi}$
Phase ϕ	$\arctg\left(\frac{-\lambda}{\omega_1}\right)$	$\frac{-\pi}{2} \cdot \text{signe}(y'_0)$	$\arctg\left(\frac{-y'_0}{y_0 \omega_0}\right)$	$\pi + \arctg\left(\frac{-y'_0}{y_0 \omega_0}\right)$
Solution $y_p(t)$	$y_p(t) = Ae^{-\lambda t} \cos(\omega_1 t + \phi)$		$y_p(t) = A \cos(\omega_0 t + \phi)$	

Tableau II.2 : Solutions des équations différentielles.

II.2.3 Recuit simulé

Un recuit simulé inhomogène est considéré. Le recuit simulé génère uniformément et aléatoirement une solution initiale admissible p du problème. A chaque itération du recuit simulé, une nouvelle solution p' est choisie selon un système de voisinage V . La solution voisine est acceptée lorsqu'elle est meilleure que la solution courante ou avec une certaine probabilité.

La température du recuit simulé décroît à chaque itération, en la multipliant par un facteur de décroissance de température T_d ($T_d < 1$). Ce facteur est calculé en fonction du nombre d'itérations du recuit simulé N_{Iter} , de la température initiale T_0 et de la température d'arrêt T_f . Il est donné par : $T_d = \sqrt[N_{Iter}]{T_0 / T_f}$.

Le système de voisinage V consiste à modifier un paramètre p_i ou plusieurs paramètres de la solution choisis aléatoirement. Il est caractérisé par son rayon d . Le paramètre p_i est choisi dans l'intervalle $[p_i - d, p_i + d]$. Au départ, lorsque la température est élevée, le rayon initial d_0 du système de voisinage est important et la solution courante peut être fortement modifiée. Ce rayon décroît à chaque amélioration de la solution courante.

Les contraintes de dépendance entre les paramètres sont prises en compte dans le système de voisinage. La solution voisine doit respecter ces contraintes pour qu'elle soit admissible.

Dans le cas d'un système physique, nous proposons un système de voisinage qui génère une solution voisine où tous les paramètres de cette solution sont modifiés. Les paramètres de la solution voisine doivent respecter la condition d'oscillation et les contraintes réelles liées à leurs valeurs maximales et minimales :

a. Cas 1

La solution voisine P' de la solution P est obtenue en respectant la condition d'oscillation (7).

Algorithme II.1 : Algorithme de génération d'une solution voisine P'

1. Soit une solution $P = (c)$
 2. Choisir aléatoirement et uniformément $c' \in [\max(c - d, 0), \min(c + d, 2\sqrt{km})]$
 3. $P' \leftarrow (c')$
-

b.Cas 2

L'Algorithme II.2 donne les étapes de génération d'une solution voisine $P = (m', c')$ d'une solution courante $P = (m, c)$.

Algorithme II.2 : Algorithme de génération d'une solution voisine P'

1. Soit une solution $P = (m, c)$
2. Choisir aléatoirement et uniformément $m' \in [\max(m - d, m_{\min}), \min(m + d, m_{\max})]$
3. Choisir aléatoirement et uniformément $c' \in [\max(c - d, 0); \min(c + d, 2\sqrt{km'})]$
4. $P' \leftarrow (m', c')$

c.Cas 2

L'Algorithme II.3 donne les étapes de génération d'une solution voisine $P = (k', m', c')$ d'une solution courante $P = (k, m, c)$.

Algorithme II.3 : Algorithme de génération d'une solution voisine P'

1. Soit une solution $P = (k, m, c)$
2. Choisir aléatoirement et uniformément $k' \in [\max(k - d, k_{\min}); \min(k + d, k_{\max})]$
3. Choisir aléatoirement et uniformément $m' \in [\max(m - d, m_{\min}), \min(m + d, m_{\max})]$
4. Choisir aléatoirement et uniformément $c' \in [\max(c - d, 0); \min(c + d, 2\sqrt{k'm'})]$
5. $P' \leftarrow (m', c')$

Le principe du recuit simulé proposé est présenté dans l'Algorithme II.4.

Algorithme II.4 : Algorithme de principe du recuit simulé inhomogène

1. Choisir uniformément et aléatoirement une solution initiale p
2. $\hat{p} \leftarrow p$
3. $T \leftarrow T_0$
4. $d \leftarrow 10$
5. **Tant que** $T > T_a$ **faire**
6. Choisir uniformément et aléatoirement une solution voisine $p' \in V(p)$ selon le rayon d
7. **Si** $F(p') - F(p) \leq 0$ **alors**
8. $p \leftarrow p'$
9. $d \leftarrow \lfloor 10(d-1)/d \rfloor$
10. **Sinon**
11. Choisir uniformément et aléatoirement $\alpha \in [0,1]$
12. **Si** $\alpha \leq \exp [-(F(p') - F(p))/T]$ **alors**
13. $p \leftarrow p'$
14. **Fin si**
15. **Fin si**
16. $T \leftarrow T * T_d$
17. **Si** $F(\hat{p}) < F(p)$ **alors**
18. $\hat{p} \leftarrow p$
19. **Fin si**
20. **Fin tant que**
21. Retourner \hat{p}

II.2.4 Algorithme génétique

Nous proposons un algorithme génétique pour résoudre le problème d'identification paramétrique des équations différentielles dans le cas où les trois paramètres sont inconnus. Chaque chromosome est une solution admissible du problème (vecteur des paramètres à identifier). De nouveaux chromosomes sont produits grâce à l'opérateur de sélection et

l'opérateur de croisement. Lorsque les paramètres du système étudié sont soumis à certaines contraintes, l'opérateur de croisement peut produire des solutions non admissibles. L'opérateur de mutation a pour but de rendre ces solutions admissibles.

Le principe de cet algorithme génétique est décrit dans l'Algorithme II.5.

Algorithme II.5 : Algorithme de principe de l'algorithme génétique

1. **Pour** i de 1 à $NIter$ **faire**
2. Choisir via un opérateur de sélection par roulette k chromosomes
3. Appliquer l'opérateur de croisement aux chromosomes sélectionnés pour obtenir des chromosomes enfants
4. Appliquer l'opérateur de mutation aux chromosomes enfants non admissibles
5. Réinsérer les chromosomes enfants dans la population
6. **Fin pour**

II.2.4.1 Opérateur de croisement

L'opérateur de croisement est appliqué aux chromosomes parents sélectionnés via un opérateur de sélection par roulette. Le point de croisement est choisi aléatoirement.

Nous présentons un exemple de croisement dans le cas d'un système physique (Figure II.3). Les deux chromosomes sélectionnés respectent la condition d'oscillation ($c_1 \in [0, 2\sqrt{m_1 \cdot k_1}]$ et $c_2 \in [0, 2\sqrt{m_2 \cdot k_2}]$). Les solutions obtenues à l'issue de cet opérateur peuvent ne pas respecter la condition d'oscillation.

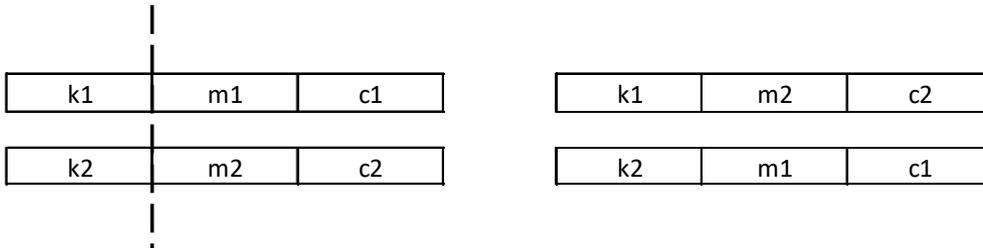


Figure II.3 : Exemple de croisement.

II.2.4.2 Opérateur de réparation

Dans le cas général des algorithmes génétiques, la mutation est appliquée avec une certaine probabilité de mutation. Dans ce travail, nous proposons d'appliquer l'opérateur de réparation à tous les chromosomes pour lesquels la condition d'oscillation n'est pas respectée.

Le paramètre c d'une solution non admissible est modifié de telle sorte que la condition d'oscillation soit respectée.

Le principe de la réparation est présenté par l'Algorithme II.6, avec P un vecteur de trois paramètres $(p_1, p_2, p_3) = (k, m, c)$

Algorithme II.6 : Algorithme de réparation des solutions

1. **Pour** P de 1 à nombre de chromosomes **faire**
2. **Si** $p_3 \notin [0; 2\sqrt{p_1 p_2}]$ **alors**
3. Choisir aléatoirement et uniformément $c \in [0; 2\sqrt{p_1 p_2}]$
 $p_3 \leftarrow c$
4. **Fin si**
5. **Fin Pour**

II.3 Résultats dans le cas des paramètres entiers

Dans cette partie, afin de valider les méthodes de résolution proposées, nous allons étudier le cas où les paramètres du système sont des entiers. Dans un premier temps, une énumération des solutions de l'espace de recherche est donnée. Cette énumération permet, d'une part, de voir l'espace de la fonction objectif, d'autre part, de valider les résultats des approches proposées. Les résultats du recuit simulé et de l'algorithme génétique seront donnés.

Les valeurs des paramètres sont supposées entières. Les valeurs maximales et minimales de la masse m et du coefficient de raideur k sont les mêmes pour tous les systèmes. $m_{\min} = 0$, $m_{\max} = 10000$, $k_{\min} = 0$ et $k_{\max} = 1000$. Pour que les solutions soient admissibles, la valeur de c maximale est déterminée en fonction de la valeur de la masse m et de la constante de raideur k .

Les méthodes proposées ont été testées pour identifier les paramètres de cinq systèmes. Les paramètres de chacun de ces systèmes ont été générés aléatoirement, de telle sorte que le système soit en régime périodique. Le Tableau II.3 donne les paramètres des systèmes étudiés.

Système	k (N/m)	m (kg)	c (kg/s)
1	55	650	62
2	123	3250	258
3	12	150	5
4	357	6125	27
5	128	257	307

Tableau II.3 : Systèmes en régime périodique.

Pour chaque système (Tableau II.4), deux instances sont générées avec les mêmes conditions initiales : la position initiale $y(0) = 100$ et la vitesse initiale $\dot{y}(0) = 0$. Ces instances dépendent du nombre d'observations temporelles (six pour la première instance et onze pour la deuxième instance), des instants $t = (t_0, t_1, \dots, t_n)$ auxquels les observations ont été effectuées (assez espacés (système 3 instance 1) ou rapprochés (système 5)) et de la précision des observations (10^{-2} pour les quatre premiers systèmes et de 10^{-6} pour le dernier système).

système	Instance	Positions $x(t)$ aux instants t											
		t											
1	1	t	0		20		40		60		80		100
		$y(t)$	100,00		29,64		4,7		-1,3		-1,56		-0,83
	2	t	0	10	20	30	40	50	60	70	80	90	100
		$y(t)$	100,0	-57,01	29,64	-13,48	4,7	-0,43	-1,3	1,72	-1,56	1,2	-0,83
2	1	t	0		20		40		60		80		100
		$y(t)$	100,0		-41,34		8,92		2,11		-3,32		1,93
	2	t	0	10	20	30	40	50	60	70	80	90	100
		$y(t)$	100,0	-8,78	-41,34	22,18	8,92	-13,96	2,11	5,38	-3,32	-0,97	1,93
3	1	t	0		40		80		120		160		200
		$y(t)$	100,0		12,2		-22,68		-10,06		2,94		3,54
	2	t	0	5	10	15	20	25	30	35	40	45	50
		$y(t)$	100	19,93	-78,84	-39,85	55,12	49,79	-32,15	-51,52	12,2	47,17	3,42
4	1	t	0		20		40		60		80		100
		$y(t)$	100,0		10,2		-89,3		-29,10		75,33		43,31
	2	t	0	10	20	30	40	50	60	70	80	90	100
		$y(t)$	100,0	-72,47	10,20	54,44	-89,3	78,4	-29,10	-32,5	75,33	-78,97	43,31
5	1	t	0		5		10		15		20		25
		$y(t)$	100,000000		6,105671		-0,442025		-0,001992		0,001187		-0,000031
	2	t	0	2	4	6	8	10	12	14	16	18	20
		$y(t)$	100,000000	54,983089	15,153809	1,661579	-0,654685	-0,442025	-0,135513	-0,019413	0,003840	0,003479	0,001187

Tableau II.4 : Instances avec paramètres entiers.

II.3.1 Enumération

Comme les paramètres du système sont des entiers, nous avons énuméré pour chaque système les paramètres recherchés.

Nous présentons l'évolution de la fonction objectif en fonction des valeurs du paramètre c connaissant les paramètres m et k (Figure II.4) du système 1. Cette courbe n'est pas convexe et présente deux minima : un minimum global entre 0 et 100 et un minimum local entre 300 et 400. La valeur du paramètre c permettant la transition de la fonction objectif vers des grandes valeurs correspond à la limite des solutions admissibles c_{max} . Les valeurs supérieures à cette limite représentent les solutions non admissibles permettant un système apériodique. Le temps de l'énumération est de l'ordre de l'heure.

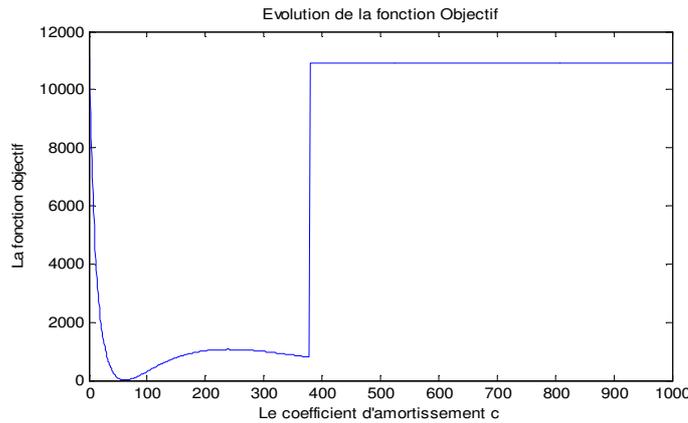


Figure II.4 : Variation de la fonction objectif en fonction d'un seul paramètre c

Lorsque k est connu, m et c sont à déterminer. Dans ce cas, la solution initiale est un vecteur des deux paramètres à retrouver, telle que la masse m est choisie aléatoirement dans un intervalle donné, et c est choisi d'une façon à satisfaire la condition d'oscillation donnée précédemment. $m \in [0; m_{max}]$ et $c \in [0; 2\sqrt{km}]$. La masse maximale m_{max} ne doit pas dépasser la force maximale que peut supporter le ressort. Les solutions non admissibles (du système 1) sont présentées dans la zone à gauche du graphique de la Figure II.5. Les solutions admissibles sont présentées dans la zone à droite dans laquelle plusieurs minimums locaux existent. Le temps de calcul est de l'ordre de 3h par système et instance.

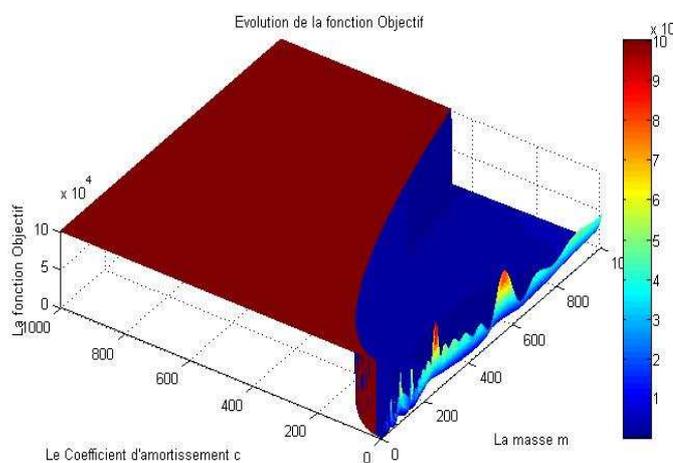


Figure II.5 : Variation de la fonction objectif en fonction de deux paramètres m et c .

Dans le dernier cas, l'énumération permet également de déterminer les trois paramètres du système recherché. Il faut plusieurs heures pour obtenir le résultat.

II.3.2 Recuit simulé

Les paramètres du recuit simulé utilisés sont la température initiale $T_0 = 10$, une température d'arrêt $T_f = 0.01$ et un nombre d'itérations $N_{iter} = 10^6$. Comme le recuit simulé est stochastique, pour chaque système et chaque instance, nous avons appliqué 50 fois notre algorithme pour déterminer les paramètres de chaque système.

Le recuit simulé a été utilisé pour traiter les trois cas d'identification :

- **Cas 1** : identification d'un seul paramètre c ;
- **Cas 2** : identification des deux paramètres c et m ;
- **Cas 3** : identification des trois paramètres k , c et m .

Le recuit simulé permet de retrouver les paramètres recherchés de tous les systèmes dans le cas 1 et le cas 2.

Dans le troisième cas (identification des trois paramètres), les résultats obtenus sont présentés dans le Tableau II.5. Dans la plupart des cas, en moyenne, les résultats obtenus avec les instances 2 (avec 11 observations temporelles) sont meilleurs que ceux obtenus avec les instances 1 (avec six observations temporelles).

Système	1		2		3		4		5	
Instances	1	2	1	2	1	2	1	2	1	2
Min(F)	0	1,00E-04	0	1,00E-04	1,08E-03	0	7,12E-04	0	2,42E-06	4,74E-05
Moy(F)	1,52E-01	1,11E-03	6,98E-04	1,44E-03	1,60E+00	3,39E-04	1,23E-02	7,23E-03	1,37E-04	8,40E-05
Max(F)	6,53E+00	3,40E-03	6,20E-03	6,90E-03	9,98E+00	4,09E-03	6,04E-02	1,78E-02	6,55E-03	3,13E-04
ET(F)	9,23E-01	5,02E-04	1,17E-03	1,67E-03	2,00E+00	8,32E-04	1,25E-02	5,21E-03	9,26E-04	4,60E-05

Tableau II.5 : Résultats du recuit simulé pour l'identification de trois paramètres (k , m et c).

Nous présentons dans la Figure II.6, les observations temporelles du système 1, instance 1, et les mesures obtenues par la solution donnant une erreur maximale de l'instance 1.

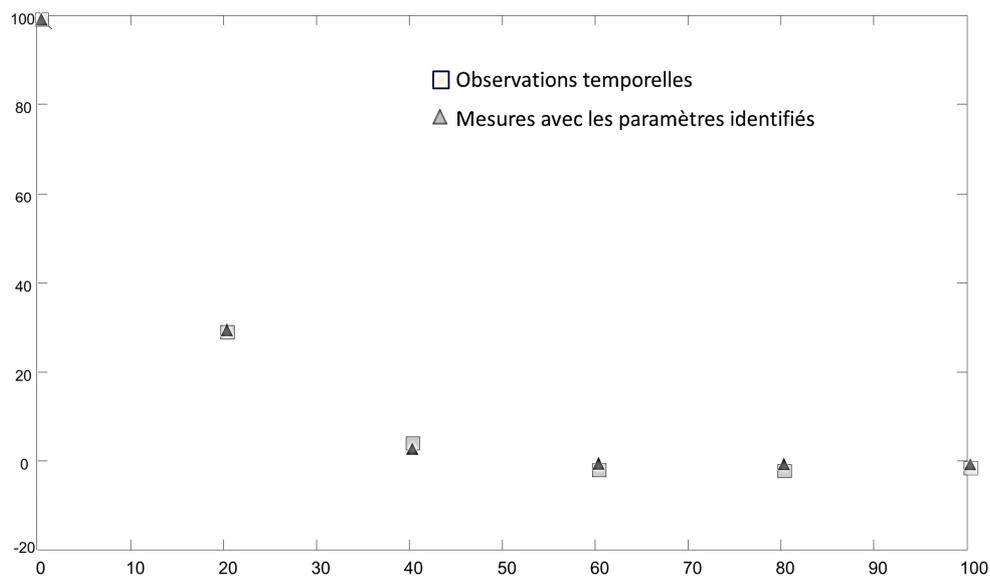


Figure II.6 : Observations temporelles et celles obtenues par les paramètres donnant une erreur maximale du système 1.

Nous avons constaté qu'il existe plusieurs solutions équivalentes. Ces solutions sont équivalentes à la solution recherchée, car elles donnent le même comportement du système. De même pour les autres systèmes, il existe plusieurs solutions équivalentes à la solution recherchée.

Le Tableau II.6 un exemple de deux solutions équivalentes, en termes de valeurs de la fonction objectif obtenues. Nous présentons un exemple pour le système 1.

	k (N/m)	m (kg)	c (kg/s)	α	$\lambda = c/2m$	$\omega_0^2 = k/m$
Solution recherchée	55	650	62		0,5636	0,0846
Solution équivalente 1	330	3900	372	6	0,5636	0,0846
Solution équivalente 2	401	4739	452	7,29	0,5636	0,0846

Tableau II.6 : Exemple de solutions équivalentes.

II.3.3 Algorithme génétique

L'algorithme génétique est utilisé pour déterminer les trois paramètres du système k , m et c . Le codage d'un chromosome est le même que pour le cas 3 du recuit simulé.

L'algorithme génétique est lancé 50 fois pour identifier les trois paramètres de chaque système et pour chaque instance. La taille de la population est de 1000 et le nombre de générations est de 1000. La probabilité de croisement est de 0,7.

Les résultats obtenus sont présentés dans le Tableau II.7. Les résultats de l'algorithme génétique sont moins bons que ceux du recuit simulé.

Système	1		2		3		4		5	
Instance	1	2	1	2	1	2	1	2	1	2
Min(F)	0	0	0	0	1,12E-06	3,44E-06	2,69E-04	1,43E-05	2,39E-08	1,07E-06
Moy(F)	1,88E-01	6,93E-02	9,60E-04	6,08E-04	2,14E-01	2,29E-03	1,09E+00	3,59E-02	1,70E-02	1,04E-04
Max(F)	2,64E+00	8,16E-01	1,20E-02	6,40E-03	7,90E-01	6,14E-02	9,99E+00	6,24E-01	2,40E-01	2,37E-03
ET(F)	1,16E-01	2,20E-01	1,85E-03	1,01E-03	2,35E-01	8,72E-03	2,82E+00	9,20E-02	5,49E-02	3,29E-04

Tableau II.7 : Résultats de l'algorithme génétique pour l'identification de trois paramètres entiers (k , m et c).

II.4 Résultats dans le cas de paramètres réels

Les valeurs des paramètres sont supposées réelles. Les valeurs maximales et minimales de la masse m et du coefficient de raideur k sont les mêmes pour tous les systèmes : $m_{\min} = 0$, $m_{\max} = 10000$, $k_{\min} = 0$ et $k_{\max} = 1000$.

Pour que les solutions soient admissibles, la valeur de c maximale est déterminée par la condition d'oscillation en fonction de la valeur de la masse m et de la constante de raideur k .

Les méthodes proposées ont été testées pour identifier les paramètres de cinq systèmes. Les paramètres de chacun de ces systèmes ont été générés aléatoirement, de telle sorte que le système soit en régime périodique. Le Tableau II.8 donne les paramètres des systèmes étudiés.

Système	k (N/m)	m (kg)	c (kg/s)
6	55.5	650.8	62.6
7	123.9	3250.2	258.8
8	12.53	150.87	5.61
9	357.86	6125.65	27.42
10	128.64	257.37	310.43

Tableau II.8 : Systèmes en régime périodique.

Pour chaque système (Tableau II.9), deux instances sont générées avec les mêmes conditions initiales : la position initiale $y(0)=100$ et la vitesse initiale $\dot{y}(0)=0$. Ces instances dépendent du nombre d'observations temporelles (six pour la première instance et onze pour la deuxième instance), des instants $t = (t_0, t_1, \dots, t_n)$ auxquels les observations ont été effectuées (assez espacés (système 3, instance 1) ou rapprochés (système 5)) et de la précision des observations (10^{-2} pour les deux premiers systèmes, 10^{-4} pour les systèmes (3 et 4) et de 10^{-6} pour le dernier système).

système	Instance	Positions $y(t)$ aux instants t											
		t	0		20		40		60		80		100
1	1	t	0		20		40		60		80		100
		$y(t)$	100		20,89		1,75		-0,89		-0,58		-0,2
	2	t	0	10	20	30	40	50	60	70	80	90	100
		$y(t)$	100	-48,3	20,89	-7,56	1,75	0,37	-0,89	0,81	-0,58	0,36	-0,2
2	1	t	0		20		40		60		80		100
		$y(t)$	100		-40,9586		8,3521		2,4831		-3,45		1,9
	2	t	0	10	20	30	40	50	60	70	80	90	100
		$y(t)$	100	-9,27	-40,96	22,56	8,35	-13,93	2,48	5,16	-3,45	-0,79	1,9
3	1	t	0		40		80		120		160		200
		$y(t)$	100		2,0974		-1,4155		-0,1543		0,0094		0,0032
	2	t	0	5	10	15	20	25	30	35	40	45	50
		$y(t)$	100	25,7109	-53,5496	-27,9507	25,332	22,6335	-9,7649	-15,7941	2,0974	9,9094	1,0857
4	1	t	0		20		40		60		80		100
		$y(t)$	100		10,7055		-88,9562		-30,399		74,2952		45,0092
	2	t	0	10	20	30	40	50	60	70	80	90	100
		$y(t)$	100	-72,6142	10,7055	53,7594	-88,9562	78,8513	-30,399	-30,8868	74,2952	-79,2551	45,0092
5	1	t	0		5		10		15		20		25
		$y(t)$	100		6,388969		-0,409811		-0,004487		0,001103		-0,000019
	2	t	0	2	4	6	8	10	12	14	16	18	20
		$y(t)$	100	55,065605	15,431472	1,901355	-0,54053	-0,409811	-0,133097	-0,022236	0,002076	0,002912	0,001103

Tableau II.9 : Instances avec paramètres réels.

II.4.1 Résultat du recuit simulé

Les paramètres du recuit simulé utilisés sont la température initiale $T_0 = 10$, une température d'arrêt $T_f = 0,01$ et un nombre d'itérations $N_{iter} = 10^6$. Comme le recuit simulé est stochastique, pour chaque système et chaque instance, nous avons appliqué 50 fois notre algorithme pour déterminer les paramètres de ce système.

a. Identification d'un seul paramètre c

Dans ce cas, on cherche à déterminer le paramètre c , connaissant les paramètres m et k de chaque système. Les résultats obtenus sont présentés dans le Tableau II.10. Pour chaque système et instance, le minimum $Min(F)$, le maximum $Max(F)$, la moyenne $Moy(F)$ et l'écart-type $ET(F)$ de la fonction objectif des 50 réplifications du recuit simulé sont donnés.

Ces résultats montrent que le recuit simulé retrouve le paramètre recherché. Le recuit simulé retrouve une erreur nulle pour les quatre premiers systèmes. Pour le dernier système, la fonction objectif varie de 0 et 10^{-12} .

Système	1		2		3		4		5	
Instance	1	2	1	2	1	2	1	2	1	2
Min(F)	0	0	0	0	0	0	0	0	0	0
Moy(F)	0	0	0	0	0	0	0	0	2,00E-12	3,76E-12
Max(F)	0	0	0	0	0	0	0	0	1,70E-11	2,10E-11
ET(F)	0	0	0	0	0	0	0	0	3,73E-12	4,81E-12

Tableau II.10 : Résultats du recuit simulé pour l'identification d'un seul paramètre réel (c).

b. Identification de deux paramètres c et m

Les résultats obtenus sont présentés dans le Tableau II.11. En moyenne, les résultats du recuit simulé sont satisfaisants. Dans tous les cas, les résultats des instances avec 11 observations temporelles sont meilleurs que ceux obtenus avec les instances avec 6 observations temporelles.

Système	1		2		3		4		5	
Instances	1	2	1	2	1	2	1	2	1	2
Min(F)	1,00E-04	0	0	0	9,76E-05	4,78E-06	4,17E-06	1,45E-06	1,90E-06	2,14E-07
Moy(F)	3,05E-01	1,64E-02	6,03E-02	5,00E-05	9,29E-02	6,04E-04	1,51E-01	6,26E-05	1,18E-03	1,26E-05
Max(F)	1,31E+00	8,15E-01	9,93E-01	1,00E-04	3,34E-01	2,47E-03	1,25E+00	3,31E-04	6,72E-03	1,00E-04
ET(F)	5,09E-01	1,14E-01	2,08E-01	5,00E-05	4,27E-02	5,68E-04	3,16E-01	6,85E-05	2,09E-03	1,66E-05

Tableau II.11 : Résultats du recuit simulé pour l'identification de deux paramètres réels (m et c).

c. Identification des trois paramètres k , c et m

Dans cette partie, on cherche à déterminer tous les paramètres du système k , m et c . Les résultats obtenus sont présentés dans le Tableau II.12.

Comme les paramètres sont dépendants les uns des autres. Il existe plusieurs solutions qui sont équivalentes et proches de la solution optimale. Les paramètres de ces solutions sont des multiples d'un facteur réel α de ceux recherchés.

Système	1		2		3		4		5	
Instance	1	2	1	2	1	2	1	2	1	2
Min(F)	0	0	0	0	0	0	1,00E-08	1,00E-08	2,00E-12	5,10E-11
Moy(F)	8,44E-02	3,40E-03	0	0	3,05E-02	4,17E-07	3,25E-01	2,69E-06	5,99E-04	3,49E-08
Max(F)	1,31E+00	8,61E-02	0	0	9,55E-02	1,06E-05	8,48E+00	1,53E-05	3,00E-02	1,24E-06
ET(F)	1,92E-01	1,67E-02	0	0	3,36E-02	1,54E-06	1,23E+00	3,82E-06	4,20E-03	1,73E-07

Tableau II.12 : Résultats du recuit simulé pour l'identification de trois paramètres réels (k , m et c).

II.4.2 Résultat de l'algorithme génétique

L'algorithme génétique est lancé 50 fois pour identifier les trois paramètres de chaque système et pour chaque instance. La taille de la population est de 1000 et le nombre de générations est de 1000. La probabilité de croisement est de 0,7.

Les résultats obtenus sont présentés dans le Tableau II.13. Les résultats de l'algorithme génétique sont moins bons que ceux du recuit simulé.

Système	1		2		3		4		5	
Instance	1	2	1	2	1	2	1	2	1	2
Min(F)	4,94E-02	1,83E-01	1,46E+00	5,75E-03	5,12E+00	3,05E-01	3,67E+00	5,98E+00	3,90E-02	8,10E-01
Moy(F)	3,31E-01	1,39E-01	3,97E+00	6,52E-02	4,82E+01	7,60E+01	1,74E+02	6,91E+02	1,36E-02	1,64E-01
Max(F)	1,87E+00	5,33E+00	1,06E+01	2,31E-01	9,51E+01	1,67E+02	7,74E+02	5,10E+03	2,39E-04	8,00E-03
ET(F)	3,12E-01	1,26E+00	2,16E+00	5,47E-02	2,65E+01	4,82E+01	1,80E+02	9,15E+02	8,40E-03	1,66E-01

Tableau II.13 : Résultats de l'algorithme génétique pour l'identification de trois paramètres réels (k , m et c).

Conclusion

Dans ce chapitre, nous avons proposé des métaheuristiques pour identifier les paramètres d'un système décrit par des équations différentielles. L'identification de ces paramètres est réalisée à

partir des observations temporelles du système. Nous supposons que les équations décrivant le système sont connues.

Les méthodes proposées sont basées sur un recuit simulé et un algorithme génétique. Chaque solution du problème est un vecteur des paramètres à identifier. La fonction objectif consiste à minimiser l'écart quadratique entre les observations temporelles et celles du modèle.

Dans certains systèmes, les paramètres sont soumis à certaines contraintes et peuvent être liés entre eux. Nous avons proposé pour le recuit simulé un système de voisinage qui génère une solution admissible qui respecte ces contraintes pour la solution courante. Pour l'algorithme génétique, des solutions non admissibles peuvent être obtenues par l'application de l'opérateur de croisement à deux solutions admissibles. Nous avons proposé un opérateur de réparation appliqué à toutes ces solutions non admissibles pour les rendre admissibles.

Les méthodes proposées ont été utilisées pour identifier les paramètres d'un système physique décrit par une équation différentielle du second ordre. Les résultats obtenus sont satisfaisants et ces méthodes permettent d'identifier les bons paramètres des différents systèmes étudiés. Lorsque les paramètres sont dépendants les uns des autres, il existe une multitude de solutions équivalentes à la solution optimale. Le recuit simulé a permis d'obtenir de meilleurs résultats que ceux de l'algorithme génétique.

Il serait intéressant de traiter ce problème sur des systèmes dont les conditions initiales sont inconnues et de tester d'autres méthodes d'optimisation continue de type algorithme de gradient et essais particuliers.

Dans le cadre du projet de collaboration (LIMOS, IBC), les méthodes que nous avons proposées serviront de premiers travaux pour la modélisation macroscopique de systèmes biologiques. Elles peuvent être adaptées à d'autres systèmes décrits par des équations différentielles, tels que la modélisation de la pression artérielle et les déformations d'une forme biologique. Ces méthodes peuvent également être adaptées à l'identification des paramètres du modèle de croissance bactérienne permettant d'obtenir des courbes de croissance des bactéries dans différents milieux (naturel, exposé au stress radioactif).

Chapitre III. AUTOMATES CELLULAIRES ET PROBLEMES INVERSES : ETAT DE L'ART

Introduction

Un automate cellulaire se définit comme un système de cellules interagissant localement de manière simple et évoluant vers un comportement global complexe (Wolfram, 1984). Les interactions entre les cellules sont définies par des règles locales. L'ensemble de toutes ces règles forment la fonction de transition de l'automate cellulaire.

Les automates cellulaires sont utilisés pour modéliser une variété de systèmes complexes. En biologie, ils sont utilisés pour la modélisation des systèmes biologiques au niveau microscopique. Pour obtenir un comportement précis du système étudié, il est important de déterminer la fonction de transition de l'automate cellulaire à appliquer au niveau microscopique. Ce problème est le problème inverse des automates cellulaires appelé également « problème d'identification paramétrique » des automates cellulaires.

Dans ce chapitre, les systèmes complexes, ainsi que leurs problèmes inverses, sont présentés en deux parties. Dans la première partie, les automates cellulaires sont décrits de façon détaillée en donnant les différentes notations utilisées dans les prochains chapitres. La deuxième partie de ce chapitre présente un état de l'art des problèmes inverses des automates cellulaires. Ces problèmes inverses consistent à déterminer la fonction de transition permettant d'obtenir une configuration donnée de l'automate cellulaire.

III.1 Automate cellulaire

Les automates cellulaires ont été introduits pour la première fois dans les années 40 par Stanislas Ulam et John Von Neumann et ils ont été publiés dans (Von Neumann & Burks, 1966). La motivation d'Ulam était de générer dynamiquement des objets graphiques qui pouvaient se répliquer dans certains cas en utilisant un jeu de règles d'évolution simples. La base était un univers cellulaire. Chacune des cellules pouvait avoir deux états : allumé ou éteint. Partant d'une configuration donnée, la génération suivante était déterminée en fonction de règles de voisinage. Von Neumann, pour sa part, s'est intéressé aux phénomènes d'autoreproduction et voulait construire une machine pouvant créer des copies exactes d'elle-même, « le kinématon ». Ulam suggéra à Von Neumann d'utiliser un espace cellulaire pour pallier aux difficultés pratiques de la réalisation de son automate autoreproducteur. Sur cette base, il conçut un automate cellulaire de 200.000 cellules à 29 états contenant un copieur universel, une description de lui-même et une machine de Turing pour la supervision (Von Neumann & Burks, 1966; Beyer, et al., 1985; Rennard, 2000; Fatès, 2001).

En 1970, l'automate cellulaire baptisé "Le jeu de la vie" est créé par John Horton Conway et popularisé par (Gardner, 1970). Cet automate 2D permet, avec des règles très simples, de faire évoluer des configurations finies qui se reproduisent elles-mêmes ou se déplacent. Cet automate va faire l'objet de beaucoup d'études qui continuent actuellement (Bayes, 1987; Sapin, et al., 2003; Adachi, 2004). Les automates unidimensionnels ont, pour leur part, été popularisés par

(Wolfram, 1984; Wolfram, 1986). Wolfram a proposé une classification d'automates cellulaires basée sur leur comportement en quatre classes. Dans la première classe, presque toutes les configurations initiales convergent, en un temps fini, vers un même état uniforme et stable. Dans la deuxième classe, presque toutes les configurations initiales convergent, en un temps fini, vers des états éventuellement distincts, mais tous périodiques dans le temps. La troisième classe regroupe les automates cellulaires dont l'évolution est aléatoire et chaotique. Quant à la quatrième et dernière classe, elle regroupe les automates cellulaires produisant un comportement complexe. D'autres classifications (toutes basées sur celle de Wolfram) ont été proposées par la suite (Culik, et al., 1990a; Culik, et al., 1990b; Fatès, 2001; Ollinger, 2002).

Depuis, les automates cellulaires sont devenus un champ d'étude pluridisciplinaire, allant de la physique à la biologie, en passant par l'informatique. Ils ont été utilisés pour étudier certains systèmes complexes tels que : la simulation du trafic urbain (Whale, et al., 2001), la propagation des feux de forêt (Karalylidis, 1999), l'étude des ondes d'excitation cardiaque (Feldman, et al., 1999), l'évolution des tumeurs (Moreira & Deutsch, 2002), les phénomènes de diffusion (Margolus, 1984) et la génération de formes (De Garis & Brain Builder Group, 1999; Chavoya, et al., 2010).

III.1.1 Définition

Un automate cellulaire se présente sous la forme d'une grille à d dimensions divisée en cellules. Une cellule peut prendre, à chaque instant t donné, un état, choisi parmi un nombre fini d'états. À chaque nouvelle unité de temps, une nouvelle génération de cellules est produite. L'évolution de chaque cellule est assurée par une règle d'évolution locale, qui détermine son état futur, en fonction de l'état de chacune des cellules voisines définies par un système d'interaction. L'ensemble de ces règles d'évolution locales forme la fonction de transition de l'automate cellulaire.

Exemple 1

Soit un automate cellulaire 2D à deux états {noir, blanc}, l'état d'une cellule dépend de son état et de ceux des huit cellules qui l'entourent. A l'instant initial, la grille de l'automate cellulaire est composée d'une seule cellule noire au centre de la grille. La fonction de transition de l'automate F se résume comme suit :

- une cellule blanche entourée par des cellules blanches reste blanche ;
- une cellule blanche entourée de une à quatre cellules noires devient noire ;
- une cellule noire entourée de quatre cellules noires ou plus devient blanche, sinon elle conserve son état.

La Figure III.1 montre l'évolution de cet automate au cours du temps.

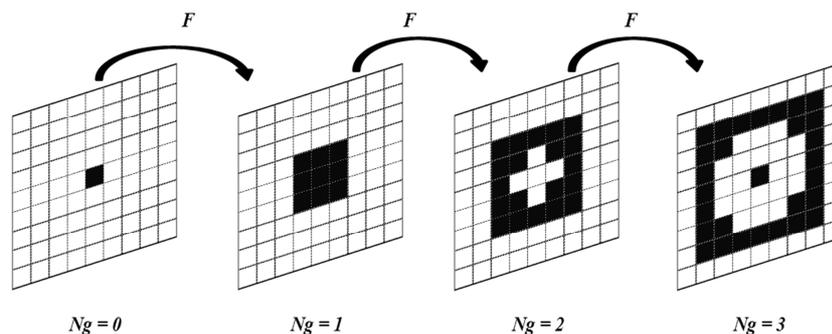


Figure III.1: Evolution de l'automate cellulaire au cours du temps.

Exemple 2 : « Jeu de la vie »

Le jeu de la vie est un automate cellulaire 2D à deux états : {vivante, morte}. Les règles d'évolution qui régissent cet automate sont :

- une cellule vivante entourée de 2 ou 3 cellules vivantes reste vivante, sinon elle meurt ;
- une cellule morte entourée exactement de 3 cellules vivantes devient vivante, sinon elle conserve son état.

A partir de ces règles simples, des comportements complexes apparaissent : des formes stables qui n'évoluent pas au cours du temps, des formes périodiques et des planeurs. Une forme est dite périodique si elle réapparaît au même endroit aux mêmes périodes (ex : à chaque génération paire). Les « planeurs » ou « vaisseaux » sont des structures périodiques qui se déplacent dans la grille de l'automate cellulaire. Nous donnons un exemple de chaque structure dans la Figure III.2.

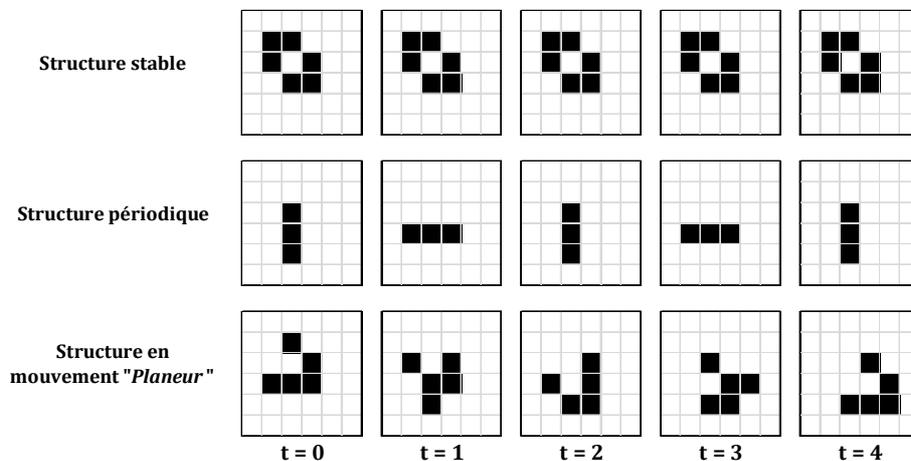


Figure III.2 : Structures observées dans le jeu de la vie.

III.1.2 Notations

La modélisation d'un processus dynamique par automate cellulaire commence par la définition des règles structurelles et fonctionnelles qui le régissent. Les règles structurelles définissent la structure spatiale de l'automate (la forme des cellules, la dimension de l'automate, le système d'interaction et les conditions aux limites). Les règles fonctionnelles permettent le fonctionnement de l'automate cellulaire (le nombre d'états, la fonction de transition et le système de mise à jour).

Un automate cellulaire peut être défini par un quadruplet $A = (G, \sigma, N, F)$, avec :

- G : la grille de l'automate cellulaire,
- σ : l'ensemble d'états,
- N : le système d'interaction,
- F : la fonction de transition.

III.1.2.1 Grille de l'automate cellulaire G

G est une grille de cellules correspondant à un pavage régulier dans le domaine Ω de \mathfrak{R}^d . d est la dimension de la grille G ($d = 1, 2$ ou 3).

Les cellules de l'automate cellulaire peuvent être carrées, triangulaires ou hexagonales (Figure III.3). La forme des cellules définit la topologie de la grille de l'automate cellulaire.

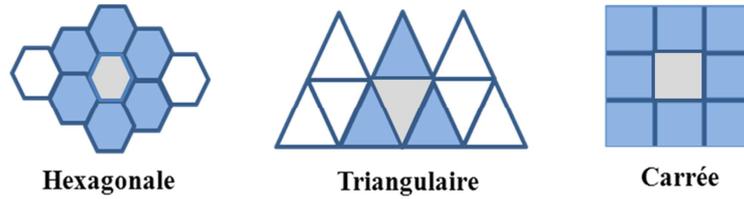


Figure III.3 : Topologies de la grille de l'automate cellulaire.

Lorsque les cellules sont carrées ou cubiques, chaque cellule c est identifiée par ses coordonnées dans la grille dans chaque dimension. Dans les autres cas, chaque cellule est repérée par les coordonnées de son centre de gravité.

III.1.2.2 Ensemble d'états σ

$\sigma = \{\sigma_1, \dots, \sigma_{|\sigma|}\}$ est l'ensemble des états que peut prendre une cellule. Le nombre d'états que peut prendre une cellule $|\sigma|$ est supérieur ou égal à 2. On notera $E_t(c) \in \sigma$ l'état d'une cellule c à un instant t donné.

Soit $list = (a, \dots, g)$ une liste de cellules. Par abus de langage, nous notons $E_t(list)$ la liste des états des cellules de la liste $list$, c'est-à-dire : $E_t(list) = (E_t(a), \dots, E_t(g))$.

Les états d'une cellule peuvent être désignés par des couleurs, des lettres, des valeurs logiques ou entières, des conditions, ...

III.1.2.3 Système d'interaction N

Le système d'interaction $N(c)$ de taille n est l'ensemble des cellules voisines, notées c' , de la cellule observée c par une relation de proximité. Lorsque les cellules sont carrées ou cubiques, la relation de proximité est donnée par la distance D , qui doit être inférieure au rayon de voisinage R . Cette distance représente la longueur du chemin parcouru de cette cellule vers la cellule observée.

$$N : G \rightarrow \mathcal{P}(G) \quad (12)$$

$$c \rightarrow N(c) = \{c' \in G ; D(c, c') \leq R\}$$

$\mathcal{P}(G)$ est l'ensemble des sous-ensembles de la grille G .

Lorsque les cellules sont hexagonales ou triangulaires, les distances D entre les cellules sont calculées en fonction des coordonnées de leurs centres de gravité.

Différents systèmes d'interaction et topologies des automates cellulaires sont présentés dans (Doner, 1988). Nous avons choisi de présenter les deux systèmes d'interaction les plus utilisés, décrits par Von Neumann et Moore.

a. Système d'interaction de Von Neumann

Dans le cas général, pour un automate cellulaire de dimension d , la distance entre deux cellules $c(c_1 \dots c_d)$ et $c'(c'_1 \dots c'_d)$, selon le système de Von Neumann, est donnée par la relation suivante :

$$D_{VN}(c, c') = \sum_{r=1}^d |c_r - c'_r| \quad (13)$$

Dans le cas d'un automate cellulaire bidimensionnel et en utilisant un rayon $R = 1$, les cellules voisines de la cellule observée sont les cinq cellules voisines (nord, est, ouest et sud) (Figure III.4). Les cellules voisines de chaque cellule observée $c_{i,j}$ (i et j les coordonnées de la cellule observée dans la grille G) sont $N(c_{i,j}) = \{c_{i,j}, c_{i-1,j}, c_{i,j-1}, c_{i+1,j}, c_{i,j+1}\}$.

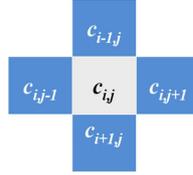


Figure III.4: Système d'interaction de Von Neumann de rayon $R = 1$.

Lorsque le rayon $R = 2$ (Figure III.5), le système de Von Neumann est composé des cellules $N(c_{i,j}) = \{c_{i,j}, c_{i-1,j-1}, c_{i,j-1}, c_{i+1,j-1}, c_{i-1,j}, c_{i+1,j}, c_{i-1,j+1}, c_{i,j+1}, c_{i+1,j+1}, c_{i-2,j}, c_{i,j-2}, c_{i+2,j}, c_{i,j+2}\}$.

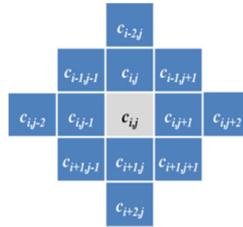


Figure III.5 : Système de Von Neumann de rayon $R = 2$.

b. Système d'interaction de Moore

La distance de Moore entre deux cellules $c(c_1...c_d)$ et $c'(c'_1...c'_d)$, dans le cas d'un automate cellulaire de dimension d , est donnée par la relation suivante :

$$D_{Moore}(c, c') = \max_{r=1,d} |c_r - c'_r| \quad (14)$$

Dans le cas d'un automate cellulaire bidimensionnel de rayon $R = 1$, le système d'interaction de Moore est donné par les neuf cellules qui entourent la cellule observée $c_{i,j}$. Ce système d'interaction représente le système d'interaction de Von Neumann, auquel on ajoute les quatre cellules qui sont sur ses diagonales (Figure III.6). Le système d'interaction de Moore est donné par : $N(c_{i,j}) = \{c_{i-1,j-1}, c_{i,j-1}, c_{i+1,j-1}, c_{i-1,j}, c_{i+1,j}, c_{i-1,j+1}, c_{i,j+1}, c_{i+1,j+1}\}$.

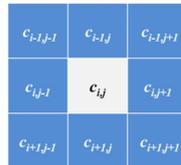


Figure III.6 : Système d'interaction de Moore

III.1.2.4 Fonction de transition F

L'évolution et la dynamique de l'automate cellulaire sont assurées par une fonction de transition F . Cette fonction détermine, pour chaque combinaison d'états $E_t(N(c))$ du système d'interaction à la génération t , l'état de la cellule observée c à la génération $t+1$, par une règle d'évolution locale f .

a.Représentation

a.1 Représentation par vecteur

La représentation de la fonction de transition la plus utilisée est la représentation sous forme d'un vecteur codé en base σ (nombre d'états). Cette représentation est également appelée représentation en table de règles (Mitchell, et al., 1996; Zhao & Billings, 2006). Sa taille N_c dépend du nombre d'états $|\sigma|$ et de la taille du système d'interaction n , elle est donnée par $|\sigma|^n$. Le nombre de fonctions de transition possibles est donné par $|\sigma|^{N_c}$.

Chaque élément de ce vecteur est une règle locale correspondant à une combinaison d'états des cellules du système d'interaction. Ces combinaisons sont données dans un ordre lexicographique. Dans cette représentation, l'ordre des cellules dans le système d'interaction est important. Il permet de déterminer, pour une configuration (k) donnée du système d'interaction, la règle $f(k)$ qui lui correspond dans le vecteur de la fonction de transition.

Pour déterminer la règle $f(k)$ qui correspond à une configuration d'état du système d'interaction $E_i(N(c))$, des coefficients de pondération sont affectés aux cellules du système d'interaction. Le coefficient de pondération de chaque cellule est donné en fonction de la position x de la cellule dans le système d'interaction par $|\sigma|^{n-x}$. Ces coefficients de pondération dépendent du nombre d'états et de la taille du système d'interaction (le nombre de cellules qui sont considérées). Ils permettent de passer d'une base d'état σ à une base décimale.

Exemple 1 : cas 1D

Dans cet exemple, nous considérons un automate cellulaire 1D binaire et un système d'interaction de rayon $R = 1$. Le système d'interaction d'une cellule observée est composé de la cellule à sa gauche, de cette cellule observée et de celle à sa droite. Si l'on décide d'affecter le poids le plus fort à la première cellule du système d'interaction et le plus faible à la dernière, la matrice des coefficients de pondération peut être donnée par :

$$[p_1 \ p_2 \ p_3] = [4 \ 2 \ 1] \quad (15)$$

Le Tableau III.1 donne le numéro (traduction en base 10) de la règle d'évolution locale k dans la fonction de transition qui correspond aux combinaisons d'états du système d'interaction.

Combinaison du système d'interaction			Coefficients de pondération			Numéro de la règle locale
$E_i(C_{i-1})$	$E_i(C_i)$	$E_i(C_{i+1})$	p_1	p_2	p_3	k
0	0	0	4	2	1	0
0	0	1	4	2	1	1
0	1	0	4	2	1	2
0	1	1	4	2	1	3
1	0	0	4	2	1	4
1	0	1	4	2	1	5
1	1	0	4	2	1	6
1	1	1	4	2	1	7

Tableau III.1 : Correspondance entre l'état du système d'interaction et la fonction de transition.

Exemple 2 : cas 2D

Nous considérons un automate cellulaire binaire. Nous considérons le système d'interaction Von Neumann de rayon $R = 1$ (à cinq cellules). Le nombre de combinaisons N_c des états des cellules du système d'interaction est de 2^5 (Figure III.7). Nous avons représenté l'ensemble des combinaisons du système d'interaction, qui sont numérotées de 0 à 31 en base 10, correspondant

à l'ordre lexicographique de $(c_{i,j}, c_{i-1,j}, c_{i,j-1}, c_{i+1,j}, c_{i,j+1})$ en base 2, avec les coefficients de pondération [16 8 4 2 1]. La cellule centrale est celle qui possède le coefficient de pondération le plus fort. La combinaison (0) représente une cellule vide entourée de 4 cellules vides et la combinaison (31) est une cellule occupée entourée uniquement par des cellules occupées.

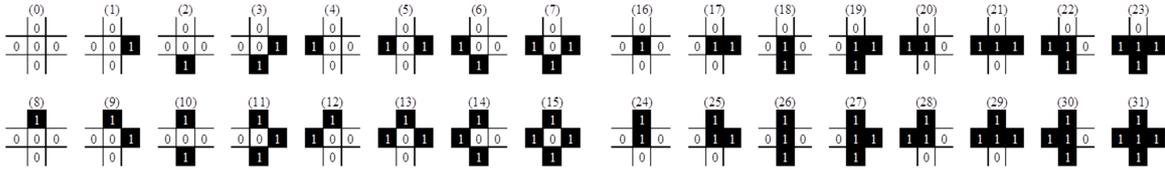


Figure III.7 : Combinaisons des états des cellules du système d'interaction

Nous présentons, dans la Figure III.8, un exemple de fonction de transition dans lequel nous avons défini un nouvel état par une règle d'évolution locale pour chaque combinaison. Par exemple, l'état de la cellule observée $c_{i,j}$ de la combinaison (0) est « 0 » et celui de la combinaison (31) est « 1 ». Les seize premières combinaisons représentent les règles d'évolution locales lorsque la cellule observée est vide et les seize dernières combinaisons pour une cellule observée pleine. Etant donné que nous considérons uniquement deux états, nous avons 2^{32} fonctions de transition possibles.

Combinaison	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)	(13)	(14)	(15)	(16)	(17)	(18)	(19)	(20)	(21)	(22)	(23)	(24)	(25)	(26)	(27)	(28)	(29)	(30)	(31)	(32)
Etat futur	0	1	1	0	1	0	0	1	0	1	0	1	0	1	0	1	1	1	0	0	0	1	0	1	0	1	1	0	0	1	0	1	1

Figure III.8 : Exemple d'une fonction de transition

Le nombre de fonctions de transition augmente avec le nombre d'états que peut prendre une cellule et la taille du système d'interaction, il est donné par : $TF = |\sigma|^{Nc}$, où $Nc = |\sigma|^n$ désigne le nombre de combinaisons des états des cellules du système d'interaction et $|\sigma|$ représente le nombre d'états. Lorsque nous utilisons un système de Moore de rayon 1, la taille de la fonction de transition est de 512 et le nombre de fonctions de transition possibles est de 2^{512} .

a.2 Représentation booléenne

Une fonction de transition de l'automate cellulaire binaire peut être représentée par des fonctions booléennes basiques (Non, Ou, Et, XOR).

Exemple

Soit la fonction de transition d'un automate cellulaire 1D binaire présenté par la Figure III.8.

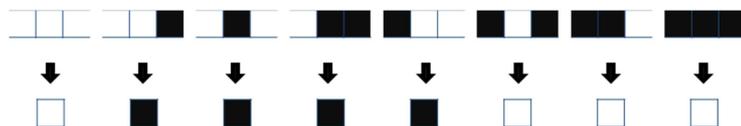


Figure III.9 : exemple de fonction de transition.

L'état d'une cellule c_i peut être exprimé en fonction de son état et de celui de ses cellules voisines (c_{i-1} et c_{i+1}), comme suit :

$$E_{t+1}(c) = (c_{i-1} \wedge \overline{c_i} \wedge \overline{c_{i+1}}) \vee (\overline{c_{i-1}} \wedge c_i) \vee (\overline{c_{i-1}} \wedge c_{i+1}) \tag{16}$$

– $E_t(c)$: l'état de la cellule c à l'instant t ;

- \wedge : l'opérateur « ET » ;
- \vee : l'opérateur « OU » ;
- La barre pour l'opérateur « Non ».

(Yang & Billings, 2003) ont proposé une représentation booléenne de la fonction de transition de l'automate cellulaire qui utilise uniquement les deux opérateurs logiques (ET et XOR). La règle locale d'une cellule c entourée de n cellules voisines c_i peut être écrite comme suit :

$$E_{t+1}(c) = a_0 \oplus a_1 E_t(c_1) \oplus a_2 E_t(c_2) \oplus \dots \oplus a_{N_c} (E_t(c_1) \wedge \dots \wedge E_t(c_n)) \quad (17)$$

- c_1, \dots, c_n : l'ensemble des cellules voisines de la cellule c ;
- \oplus : l'opérateur logique Ou-Exclusif « XOR » ;
- a_0, \dots, a_{N_c} : sont les termes de la fonction booléenne, lorsque $a_i = 1$, le terme correspondant de la fonction booléenne est pris en compte, avec $N_c = 2^n - 1$.

Chaque fonction de transition d'un automate cellulaire binaire peut être exprimée en fonction de fonctions booléennes (OR et AND). Ces fonctions booléennes sont non linéaires.

a.3 Représentation polynomiale

(Adamatzky, 1994) a proposé une représentation polynomiale de la fonction de transition de l'automate cellulaire. L'avantage principal de cette représentation est de transformer la représentation booléenne qui est non linéaire en une représentation linéaire de la fonction de transition.

La fonction de transition d'un automate cellulaire peut être donnée par une expression polynomiale exacte. L'état d'une cellule c en fonction du système d'interaction $N(c)$ à n cellules voisines est donné par :

$$E_{t+1}(c) = \theta_0 + \theta_1 E_t(c_1) + \dots + \theta_n E_t(c_n) + \dots + \theta_{N_c} E_t(c_1) \times \dots \times E_t(c_n) \quad (18)$$

avec : $N_c = 2^n - 1$.

$\theta_0, \dots, \theta_{N_c}$ sont les coefficients polynomiaux.

Exemple

Considérons un automate cellulaire 1D avec un système d'interaction de rayon $R = 1$. L'état d'une cellule c_i est donné en fonction de son état et celui de sa voisine de gauche c_{i-1} et celui de sa voisine de droite c_{i+1} . L'état de cette cellule à l'instant $t+1$ est donné par :

$$\begin{aligned} E_{t+1}(c_i) = & \theta_0 + \theta_1 E_t(c_{i-1}) + \theta_2 E_t(c_i) + \theta_3 E_t(c_{i+1}) \\ & + \theta_4 E_t(c_{i-1}) \cdot E_t(c_i) + \theta_5 E_t(c_{i-1}) \cdot E_t(c_{i+1}) + \theta_6 E_t(c_i) \cdot E_t(c_{i+1}) \\ & + \theta_7 E_t(c_{i-1}) \cdot E_t(c_i) \cdot E_t(c_{i+1}) \end{aligned} \quad (19)$$

b. Système de mise à jour

Selon la façon dont la fonction de transition est appliquée aux cellules, deux mises à jour peuvent être définies : *synchrone* et *asynchrone*.

Lorsque la fonction de transition est appliquée à toutes les cellules en même temps, le système de mise à jour est dit synchrone, et la règle d'évolution locale est définie comme suit :

$$\begin{aligned} f: \sigma^n &\rightarrow \sigma \\ N(c) &\rightarrow E_{t+1}(c) = f(E_t(N(c))) \end{aligned} \quad (20)$$

L'état d'une cellule à l'instant $t+1$ ne dépend que de l'état des cellules de son système d'interaction à l'instant t .

Dans le cas où la fonction de transition n'est pas appliquée à toutes les cellules en même temps, le système de mise à jour est dit asynchrone, et la règle de transition locale f peut être donnée par :

$$\begin{aligned} f: \sigma^n &\rightarrow \sigma \\ N(c) &\rightarrow E_{t+1}(c) = f(E_t(N_1(c)), E_{t+1}(N_2(c))) \end{aligned} \quad (21)$$

où $N_1(c) \subset N(c)$ et $N_2(c) \subset N(c)$ sont deux sous-ensembles complémentaires du système d'interaction ($N(c)$) : $N_1(c) \cup N_2(c) = N(c)$ et $N_1(c) \cap N_2(c) = \emptyset$.

L'état d'une cellule à l'instant $t+1$ dépend de l'état d'un sous-ensemble de son système d'interaction à l'instant t et de l'état du sous-ensemble complémentaire de son système d'interaction à l'instant $t+1$. Ces deux sous-ensembles du système d'interaction ne sont pas les mêmes pour toutes les cellules. Ils dépendent de la position de la cellule à mettre à jour dans la grille de l'automate cellulaire.

Le balayage des cellules dans le système de mise à jour est important et ne donne pas forcément les mêmes résultats en utilisant la même fonction de transition.

Différents systèmes de mise à jour asynchrone sont proposés par (Schönfisch & De Roos, 1999) :

- **Direction fixée ligne par ligne** : le balayage des cellules se fait ligne par ligne, en partant de la cellule à gauche de la première ligne de la grille.
- **Balayage aléatoire fixe** : il consiste à choisir un ordre aléatoire des cellules selon une loi uniforme et à mettre à jour les cellules de la grille suivant cet ordre. L'ordre des cellules à mettre à jour reste invariant à chaque génération de l'automate cellulaire.
- **Balayage aléatoire** : dans ce balayage, l'ordre des cellules est choisi uniformément et aléatoirement. A chaque génération de l'automate cellulaire, un nouvel ordre aléatoire est utilisé pour mettre les cellules à jour.
- **Choix uniforme** : les cellules à mettre à jour sont choisies selon une loi uniforme avec remise. La première cellule est choisie parmi n cellules, la seconde est aussi choisie parmi n cellules et ainsi de suite. A un instant t , une cellule peut être choisie plusieurs fois ou jamais.

Exemple

Nous considérons un automate cellulaire binaire de taille 41×41 et un système d'interaction de Von Neumann à quatre cellules voisines. La Figure III.10 montre les configurations après 8 générations de l'automate cellulaire obtenues avec différents systèmes de mise à jour en partant d'une configuration initiale de l'automate cellulaire, où une cellule à l'état « 1 » est placée au centre de la grille de l'automate cellulaire. Les règles d'évolution locale se résument comme suit : une cellule à l'état « 1 » reste à « 1 ». Si une cellule possède au moins une cellule voisine à « 1 », elle prend l'état « 1 ». Dans les autres cas, la cellule reste à l'état « 0 ». Avec le balayage

aléatoire, deux exécutions de l'automate cellulaire avec la même fonction de transition ne donnent pas les mêmes configurations de l'automate cellulaire.

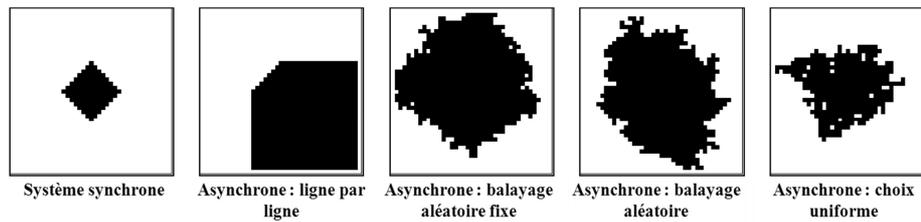


Figure III.10: Systèmes de mise à jour.

Les systèmes de mise à jour décrits précédemment permettent de mettre à jour toutes les cellules de l'automate cellulaire. Cependant, une question se pose lorsque nous cherchons à mettre à jour les cellules qui sont au bord de notre automate cellulaire. C'est-à-dire, comment choisir les systèmes d'interaction des cellules au bord, appelés « conditions aux limites » ou « conditions aux bords ».

III.1.2.5 Conditions aux limites

Les conditions aux limites de l'automate cellulaire doivent être définies. Elles peuvent être périodiques, fixes ou réflexives (El Yacoubi, et al., 2003; Abdellaoui, 2003).

- Conditions aux limites périodiques (Figure III.11) : les cellules les plus à gauche sont les cellules voisines des cellules les plus à droite et les cellules de la première ligne sont les voisines de celles de la dernière ligne.

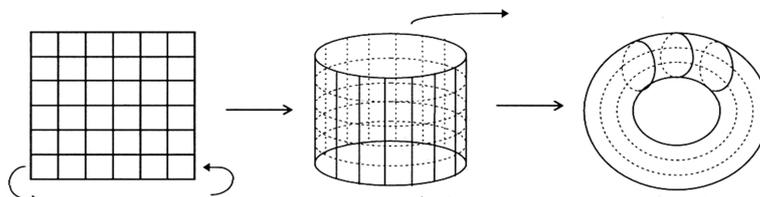


Figure III.11 : Conditions aux limites périodiques.

- Conditions aux limites réflexives : les cellules du bord de l'automate cellulaire ont comme voisines les images d'elles-mêmes.
- Conditions aux limites fixes : les cellules des bords de la grille ont un état imposé qui reste constant pendant l'évolution de l'automate cellulaire. Par exemple, les cellules des bords restent noires.

La Figure III.12 montre les deux cas des conditions aux limites : réflexives et fixes.

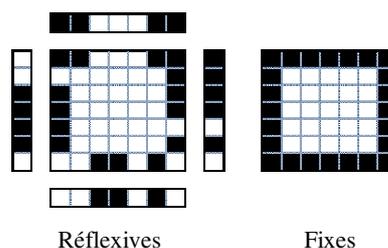


Figure III.12: Conditions aux limites.

III.1.3 Dynamique de l'automate cellulaire

Pour passer d'une configuration initiale à une configuration finale, l'automate cellulaire passe par différentes configurations intermédiaires. L'ensemble de ces configurations est appelé dynamique de l'automate cellulaire. Cette dynamique est généralement présentée par un diagramme spatio-temporel. La dimension du diagramme spatio-temporel est égale à la dimension de l'automate cellulaire à laquelle nous ajoutons une dimension pour le temps.

La Figure III.13 montre un exemple de diagramme spatio-temporel d'un automate cellulaire 1D.

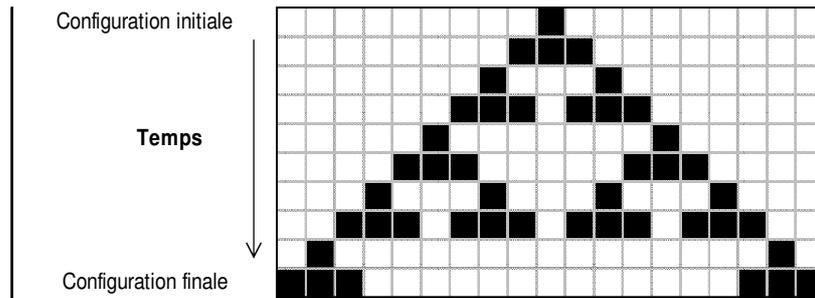


Figure III.13 : Exemple de diagramme spatio-temporel d'un automate cellulaire 1D.

III.1.4 Cas particuliers d'automates cellulaires

Automate cellulaire additif : il existe des automates cellulaires dits sommatifs ou additifs. Dans ce type d'automate, l'état d'une cellule observée est déterminé en fonction de la somme des états des cellules voisines. La fonction de transition peut être simplifiée comme suit :

$$f : \sigma^n \rightarrow \sigma$$

$$N(c) \rightarrow E_{t+1}(c) = f\left(E_t(c), \sum E_t(N(c))\right) \tag{22}$$

Nous pouvons retrouver ce type d'automate dans la plupart des automates cellulaires classiques, comme « *le jeu de la vie* » qui utilise des règles locales additives. La fonction de transition peut se présenter comme un vecteur. Chaque élément de ce vecteur représente l'état futur de la cellule observée c obtenue par une règle d'évolution additive. Nous présentons dans la Figure III.17 la fonction de transition du « jeu de la vie » avec un système d'interaction de Moore (neuf cellules).

$E_t(c)$	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
$\sum E_t(N(c))$	0	1	2	3	4	5	6	7	8	0	1	2	3	4	5	6	7	8
↓																		
$E_{t+1}(c)$	0	0	0	1	0	0	0	0	0	0	0	1	1	0	0	0	0	0

Figure II.14: Fonction de transition du jeu de la vie.

Automate cellulaire stochastique : un automate cellulaire est dit stochastique (probabiliste), lorsque des variables aléatoires sont introduites dans les règles d'évolution locales de l'automate cellulaire (Grinstein, et al., 1985; Lebowitz, et al., 1990). L'état futur d'une cellule est défini en fonction de l'état de son système d'interaction et d'une certaine probabilité. Ces automates sont très utilisés pour la modélisation de la propagation des feux de forêts et l'évolution des tumeurs (Moreira & Deutsch, 2002).

Automate cellulaire non uniforme : lorsqu'une seule fonction de transition est appliquée à toutes les cellules de la grille de l'automate cellulaire, il est dit « *uniforme* ou *homogène* ». Dans

le cas où chaque cellule possède sa propre règle d'évolution (Sipper & Tomassini, 1999), l'automate cellulaire est dit « *non uniforme* ».

Automate cellulaire continu : lorsque l'état d'une cellule est une valeur réelle, l'automate cellulaire est dit continu, ou à valeur réelle (Flocchini & Cezar, 2001). Ce type d'automate est très utilisé pour simuler certains phénomènes physiques, tels que la diffusion : dans ce type de modèle, l'état d'une cellule dépend de la moyenne des états de ses cellules voisines. Il est également utilisé pour simuler des surfaces de propagation (Gosalvez, et al., 2009).

Automate cellulaire réversible : les automates cellulaires réversibles sont caractérisés par la propriété de conservation de l'information. Connaissant la fonction de transition de l'automate cellulaire et en partant d'une configuration de l'automate cellulaire à n'importe quelle génération, on peut retracer la dynamique de l'automate cellulaire, jusqu'à sa configuration initiale. Dans ce cas, une configuration possède une seule configuration « successeur » et une seule configuration « prédécesseur ».

Automate de Margolus « lattice gaz » : Margolus (Margolus 1984) a proposé un système d'interaction pour étudier le phénomène de diffusion des gaz. Ce système est composé de deux blocs de taille 2×2 . La règle locale de l'automate cellulaire est appliquée à un bloc de quatre cellules au lieu d'une cellule. Ces blocs de quatre cellules dépendent de la parité du nombre de générations de l'automate cellulaire (Figure III.15) :

- Si la génération est paire, le premier bloc est considéré comme le système d'interaction de la cellule observée $N(c_{i,j}) = \{c_{i,j}, c_{i-1,j}, c_{i-1,j-1}, c_{i,j-1}\}$.
- Si la génération est impaire, le second bloc qui est considéré est :

$$N(c_{i,j}) = \{c_{i,j}, c_{i,j+1}, c_{i+1,j+1}, c_{i+1,j}\}.$$

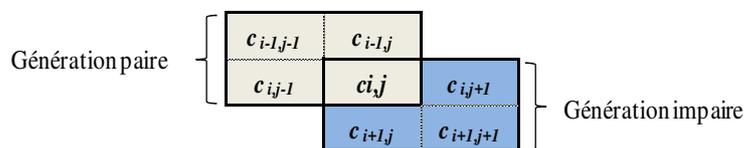


Figure III.15 : Système d'interaction de Margolus.

III.2 Problèmes inverses d'automates cellulaires

Dans cette partie, nous donnons dans un premier temps une définition du problème inverse de façon générale, puis du problème inverse des automates cellulaires. Ensuite, nous présentons quelques problèmes inverses des automates cellulaires traités dans la littérature.

III.2.1 Définitions

De façon générale, dans un problème direct, on déduit les effets connaissant les causes (Keller, 1976). A l'inverse du problème « direct », un problème est « inverse » si les effets sont connus et on cherche à connaître les causes.

III.2.1.1 Définition du problème inverse

D'un point de vue mathématique, un problème inverse consiste à inverser un ou plusieurs opérateurs (Cavalier, 2011). Un problème inverse classique est défini comme suit : soit A un opérateur de l'espace de Hilbert H dans K .

Pour $k \in K$ donné, déterminer $f \in H$ telle que $Af = k$.

Le problème est donc de gérer cette inversion pour obtenir une reconstruction précise. Il existe des cas où l'on ne peut pas inverser cet opérateur. C'est le cas des problèmes *mal posés*.

III.2.1.2 Définition du problème bien posé

Un problème est *bien posé* (Hadamard, 1932), s'il respecte les conditions suivantes:

- il existe une solution au problème ;
- la solution, si elle existe, est unique ;
- la solution dépend constamment des données.

Un problème inverse peut aussi se définir lorsqu'une des trois conditions précédentes n'est pas satisfaite.

En règle générale, la simulation des systèmes par automates cellulaires consiste à créer l'environnement le plus proche de celui du phénomène étudié (conditions initiales, paramètres, règles d'évolution,...etc.). Ce type de problèmes est dit « direct », car on veut connaître l'état futur de l'automate cellulaire, connaissant sa configuration initiale et les règles qui régissent son évolution.

Dans certains cas, afin d'obtenir une structure à une génération donnée ou une dynamique particulière de l'automate cellulaire, il est important de déterminer et choisir une fonction de transition qui permettrait à l'automate d'évoluer et de donner cette structure ou cette dynamique. Ce problème est connu sous le nom du problème inverse ou problème d'identification des automates cellulaires.

On peut distinguer différents problèmes inverses :

1. déterminer la fonction de transition de l'automate cellulaire permettant de retrouver des structures dynamiques particulières (comportement) de l'automate cellulaire ;
2. déterminer la fonction de transition de l'automate cellulaire en connaissant ses configurations initiale et finale ;
3. retrouver la configuration initiale de l'automate cellulaire en connaissant sa fonction de transition et sa configuration finale.

Dans ce travail, nous nous intéressons uniquement au deuxième cas des problèmes inverses des automates cellulaires. (Ganguly, et al., 2003) ont qualifié ce problème d'extrêmement difficile.

III.2.2 Quelques problèmes inverses d'automates cellulaires

Le problème inverse des automates cellulaires consiste à déterminer les règles d'évolution locales des cellules (fonction de transition) permettant d'obtenir une configuration de l'automate cellulaire donnée.

Cette partie présente quelques problèmes inverses des automates cellulaires traités dans la littérature (Figure III.16) dans différentes spécialités, allant des études théoriques, à des applications en biologie, en passant par la physique. Tous ces problèmes ont pour objectif de déterminer la fonction de transition de l'automate cellulaire.

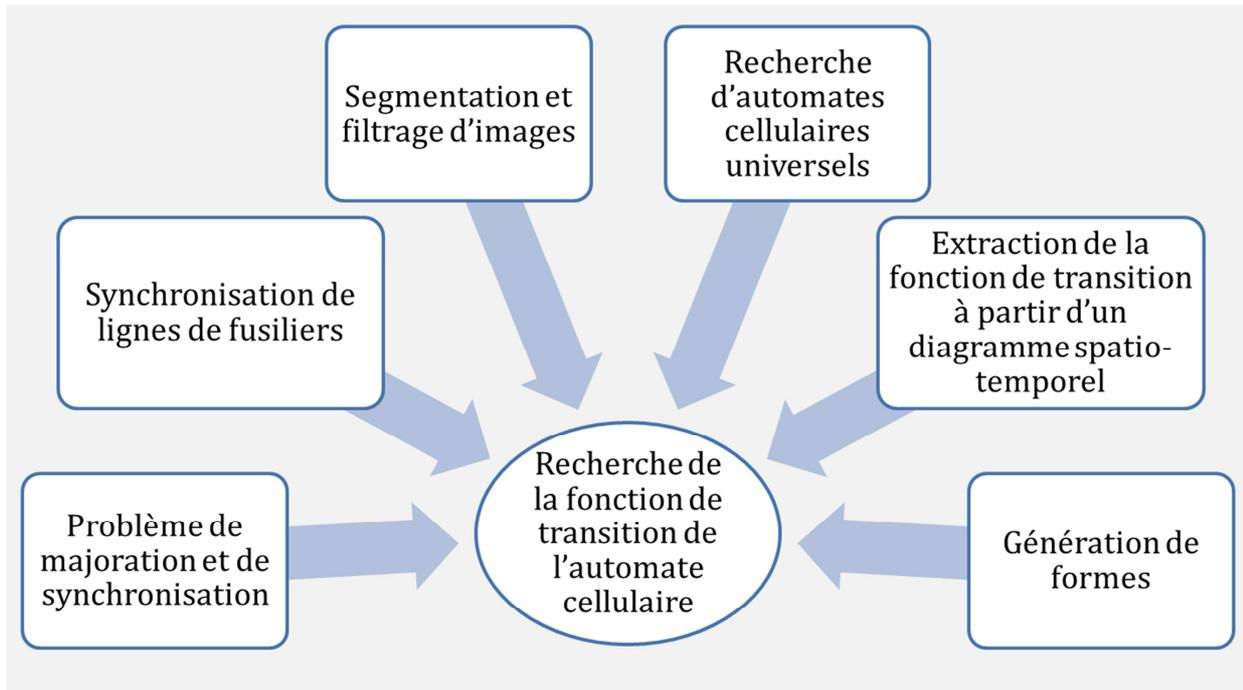


Figure III.16 : Quelques problèmes inverses des automates cellulaires.

III.2.2.1 Problème de majorité et de synchronisation

a. Problème de majorité

a.1 Définition

Le problème de majorité, appelé également problème de classification par densité, est un problème très étudié dans la littérature (Mitchell, 1993; Mitchell, et al., 1996; Breukelaar & Back, 2005; Gacs, et al., 1978). Il consiste à chercher une fonction de transition d'un automate cellulaire binaire permettant de classifier le maximum de configurations initiales.

La classification des configurations initiales est réalisée en fonction de la densité des cellules à « 1 » dans la configuration initiale, appelée densité ρ_0 . L'objectif consiste à déterminer la fonction de transition qui permet de maximiser le nombre de configurations initiales « bien classifiées ». Une configuration initiale est dite « bien classifiée » si elle vérifie l'une de ces deux conditions après N_g générations :

- si densité de la configuration initiale $\rho_0 > 0,5$ et l'automate cellulaire converge vers une configuration finale composée seulement de cellules à « 1 ».
- si densité de la configuration initiale $\rho_0 < 0,5$ et l'automate cellulaire converge vers une configuration finale composée seulement de cellules à « 0 ».

Par conséquent, la configuration finale dépend uniquement de la densité ρ_0 des cellules à « 1 » dans la configuration initiale. Le nombre de générations N_g est généralement choisi de l'ordre de la taille de l'automate cellulaire.

La Figure III.17 donne le principe du problème de majorité sur un diagramme spatio-temporel. La première ligne de ce diagramme représente la configuration initiale de l'automate cellulaire à l'instant $t = 1$.

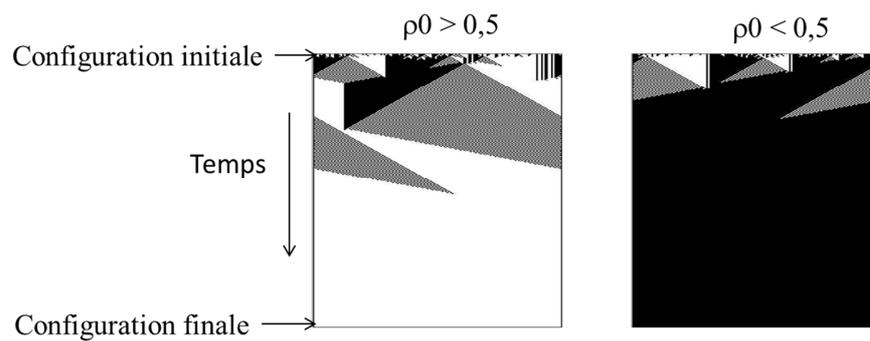


Figure III.17 : Principe du problème de majorité.

De nombreux travaux (Mitchell, et al., 1996; Andre, et al., 1996; Gacs, et al., 1978) ont traité ce problème sur des automates cellulaires 1D binaire de taille $L = 149$. La taille de la fonction de transition est donnée par 2^{2R+1} , avec R le rayon du système d'interaction de l'automate cellulaire. Le nombre de configurations initiales est de 2^{149} et le nombre de fonctions de transition possibles est donné par $2^{2^{2R+1}}$. Dans tous ces travaux, le rayon $R = 3$ est considéré.

a.2 Méthode de résolution de (Mitchell, 1993)

Nous présentons dans cette partie la méthode proposée par (Mitchell, 1993). Ils ont utilisé un algorithme génétique pour la recherche d'une fonction de transition qui permet de classifier le maximum de configurations initiales. Les étapes de cet algorithme sont les suivantes:

- L'algorithme génétique génère une population de chromosomes, chaque chromosome représente une fonction de transition de l'automate cellulaire.
- La fonction *fitness* de chaque chromosome est évaluée en fonction du nombre de configurations initiales bien classifiées. L'évaluation est réalisée en choisissant un échantillon E contenant une moitié des configurations initiales ayant une densité $\rho_0 > 0,5$ et l'autre moitié est composée de configurations initiales ayant une densité $\rho_0 < 0,5$. Le chromosome est ensuite utilisé pour faire évoluer l'automate cellulaire à partir de chaque configuration initiale de l'échantillon E pendant N_g générations de l'automate. La fonction *fitness* d'un chromosome correspond au pourcentage de configurations initiales bien classifiées dans l'échantillon E .
- Des opérateurs de croisement et de mutation sont utilisés pour générer de nouveaux chromosomes qui sont insérés dans la population par un mécanisme élitiste.
- A chaque génération de l'automate cellulaire, un nouvel ensemble E de configurations initiales est généré pour évaluer la population.

a.3 Résultats et conclusion

Plusieurs travaux ont traité ce problème de majorité. Le meilleur taux de classification obtenu est de 82,326 obtenu par (David, et al., 1996) utilisant la programmation génétique.

(Breukelaar & Back, 2005) ont traité le problème de majorité d'un automate cellulaire 2D de taille 13×13 et 3D de taille $7 \times 7 \times 7$. Ils ont considéré un système d'interaction de Von Neumann en 2D et 3D et un nombre de générations de l'automate cellulaire de 320. Ce nombre de générations a été fixé en se basant sur de précédents travaux. La fonction de transition de l'automate cellulaire est déterminée par un algorithme génétique dont les caractéristiques sont : une population de taille 100, un opérateur de sélection par tournoi sur des sous-populations de 20

chromosomes, un croisement avec une probabilité de 0,6 et une mutation avec une probabilité de 2 divisée par la taille d'un chromosome.

a.4 Problème de synchronisation

Le problème de synchronisation (Mitchell, et al., 1996) est illustré par la Figure III.18. Il consiste à trouver la fonction de transition de l'automate cellulaire qui permet, après un nombre de générations important, d'obtenir une configuration finale oscillant entre une configuration ne comportant que des cellules à l'état « 0 » et une configuration où toutes les cellules sont à l'état « 1 ». Un algorithme génétique a été également proposé pour résoudre ce problème.

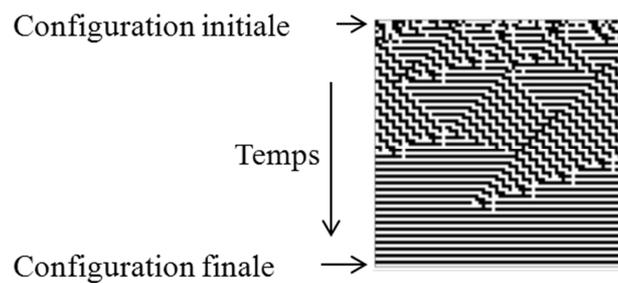


Figure III.18 : Principe du problème de synchronisation.

(Mitchell, 1993) ont utilisé les mêmes méthodes de résolution du problème de majorité pour résoudre le problème de synchronisation.

Nous avons présenté les problèmes de majorité et de synchronisation. Nous allons maintenant décrire un autre problème inverse d'automate cellulaire, le problème de la synchronisation des lignes de fusiliers.

III.2.2.2 Synchronisation des lignes de fusiliers

a.1 Définition

Le problème de lignes de fusiliers a été inventé par Myhill en 1957 mais formalisé par (Moore, 1964) comme suit :

« Soit un automate cellulaire 1D où toutes les cellules sont à un même état sauf les deux qui sont aux bords. L'état d'une cellule dépend de son état et de celui de ses deux voisines (une de chaque côté). Au départ, toutes les cellules sont dans un état appelé « latent », sauf l'une des deux cellules situées aux bords. Le problème consiste à déterminer le **nombre d'états** et à trouver **une fonction de transition** de l'automate, de telle façon que, au bout d'un certain nombre de générations, toutes les cellules se mettent en même temps à un état jamais apparu avant, appelé état de « feu » ».

La définition suivante (Yunès, 1993) n'est qu'une définition imagée du problème, mais qui lui a donné son nom : « comment synchroniser une ligne de fusiliers pour qu'ils se mettent à tirer tous en même temps, alors que l'ordre donné par un général situé à l'un des deux bords de l'escadron met un temps pour se propager ? ».

En d'autres termes, ce problème consiste à déterminer la fonction de transition avec le minimum de nombre d'états en un minimum de temps.

Plusieurs travaux se sont intéressés à la résolution de ce problème en un nombre de générations minimal Ng_{min} appelé temps optimal (Mazoyer, 1987). Ce nombre de générations dépend de la taille L de la ligne de fusiliers (taille de la grille de l'automate cellulaire) avec $Ng_{min} = 2L - 2$. Un

résumé des solutions optimales trouvées dans la littérature est présenté dans (Umeo, et al., 2004). Les algorithmes de résolution de ce problème sont des heuristiques. Par exemple, dans les travaux de (Goto, 1962), la méthode « *diviser pour régner* » est utilisée.

a.2 Résultats et conclusion

Le Tableau III.2 résume l'ensemble des solutions trouvées dans la littérature avec le nombre d'états et si la solution est pour un temps optimal ou pas.

Référence	Nombre d'états	Temps optimal	Temps non optimal
(Goto, 1962)	Plusieurs milliers	✓	
(Gerken, 1987)	32		✓
(Waksman, 1966)	16	✓	
(Balzer, 1967)	8	✓	
(Yunès, 1993)	8		✓
(Gerken, 1987)	7	✓	
(Yunès, 1993)	7		✓
(Mazoyer, 1987)	6	✓	
(Balzer, 1967)	4	Pas de solution	
(Balzer, 1967)	3	Pas de solution	
(Yunès, 1993)	3		Pas de solution

Tableau III.2 : Solutions du problème de synchronisation de fusiliers

(Balzer, 1967) a démontré que le problème ne possède pas de solutions à 3 états et à 4 états en un temps optimal. (Yunès, 1993) a prouvé que le problème n'a pas de solution à 3 états en un temps non optimal. Ce problème reste ouvert pour le cas à 5 états en un temps optimal et pour les cas d'un automate cellulaire à 4, 5 ou 6 états en un temps non optimal. La meilleure solution trouvée jusqu'à présent est celle de (Mazoyer, 1987) à six états en un temps optimal. Un exemple de la solution obtenue par (Mazoyer, 1987) est donné dans la Figure 17.

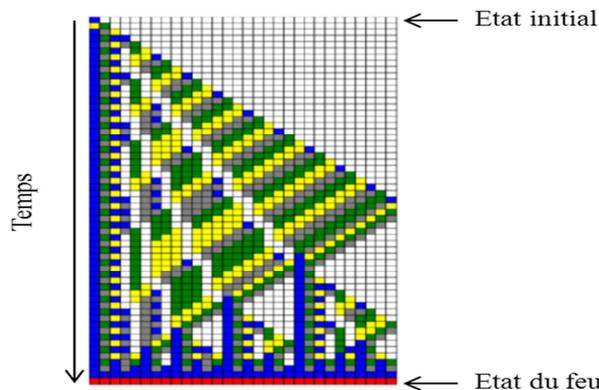


Figure III.19 : Solution de Mazoyer à 6 états (Mazoyer, 1987).

Un état de l'art du problème est donné dans (Umeo, et al., 2004).

III.2.2.3 Filtrage et segmentation d'images

L'analyse d'images fait essentiellement appel à la segmentation d'images, qui consiste à partitionner une image en régions homogènes dignes d'intérêt, appelées « *objets* ». Compte tenu de l'existence d'un grand nombre de techniques de segmentation, trois grandes catégories de segmentation peuvent être distinguées, selon que les approches se font par régions, par contours ou mixtes. Les approches par régions consistent à regrouper les pixels de l'image selon un critère

d'homogénéité prédéfini. Les approches par contours permettent de détecter les transitions entre les régions de l'image. Elles sont fondées sur la prise en compte de différentes représentations de la même image, soit en faisant appel à des méthodes différentes, soit en considérant une même méthode, mais avec différents paramètres.

Plusieurs travaux dans la littérature (Batouche, et al., 2006; Rosin, 2006; Rosin, 2010) proposent d'utiliser les automates cellulaires pour segmenter des images en utilisant l'approche par contours. La segmentation par automate cellulaire consiste à déterminer la fonction de transition permettant à l'automate cellulaire d'évoluer d'une configuration initiale, qui est l'image à segmenter, vers une configuration finale, représentant les contours de l'image à segmenter.

(Rosin, 2006) a proposé d'utiliser une recherche déterministe basée sur la méthode de Hill Climbing pour chercher la fonction de transition de l'automate cellulaire permettant de filtrer (éliminer le bruit), segmenter et déterminer les objets présents dans des images binaires. Il a considéré un système d'interaction à huit cellules voisines. La taille de la fonction de transition est donc de 2^8 . En prenant compte la symétrie axiale et rotationnelle, la taille fonction de transition a été réduite à 51. (Rosin, 2006) ont choisi de prendre l'erreur quadratique entre l'image obtenue et celle segmentée ou filtrée comme critère de comparaison.

(Batouche, et al., 2006) ont proposé un algorithme génétique pour chercher la fonction de transition de l'automate cellulaire permettant de déterminer les contours des images de départ. Le principe consiste à chercher la fonction de transition de l'automate cellulaire qui permet d'obtenir à partir d'une image non segmentée, une image segmentée. Ils ont considéré le système d'interaction de Moore à 9 cellules et ont pris en compte la symétrie axiale et rotationnelle pour réduire la taille de leur espace de recherche. La fonction objectif (*Fit*) utilisée est donnée par la formule suivante :

$$Fit = \frac{1}{v+1} \quad (23)$$

v est le nombre de pixels mal classifiés. Un pixel est mal classifié s'il n'appartient pas au contour de l'objet présent dans l'image.

III.2.2.4 Recherche d'automates cellulaires universels

(Turing, 1948) a introduit la notion d'une caractéristique fondamentale appelée « universalité ». Un automate cellulaire est universel s'il est capable de calculer toutes les fonctions calculables, c'est ce que l'on appelle l'*universalité de Turing*. Une fonction f est calculable s'il existe un algorithme permettant de calculer $f(x)$ pour chaque argument x en un nombre d'états fini.

(Berlekamp, et al., 1982) ont montré que le jeu de la vie est un automate cellulaire universel car il peut réaliser n'importe quelle fonction calculable et que toutes les portes (opérateurs) logiques peuvent être réalisées à l'aide d'interactions entre figures connues, stables ou périodiques (Figure III.2). Le *planeur* (les planeurs sont des formes qui réapparaissent dans des positions différentes après un certain nombre de générations de l'automate cellulaire) est équivalent à l'unité de base *bit*, qui permet à l'information de circuler. Chaque nombre est une *chaîne de bits* obtenue par un faisceau de planeurs générés par des *canons à planeurs*.

(Sapin, 2003) s'est basé sur cette notion pour rechercher des automates cellulaires universels. Un automate cellulaire est universel s'il accepte :

- des planeurs, qui se déplacent dans la grille et permettent la représentation et la transmission d'informations ;
- des canons à planeurs, qui émettent des flux de planeurs ;

- différentes sortes de collisions entre planeurs, dont les effets permettent la réalisation d'opérations logiques, la redirection, la synchronisation et la duplication de flux d'information.

(Sapin, et al., 2003) ont proposé un couplage d'algorithmes évolutionnaires et d'automates cellulaires pour rechercher des automates cellulaires universels. Ils ont considéré un automate cellulaire binaire 2D et un système d'interaction de Moore de rayon $R=1$.

L'espace des fonctions de transition Ω est de taille 2^{512} . Pour des raisons de concision et de lisibilité, une convention utilisant l'espace de Bayes (Bayes 1987) a été adoptée pour représenter les fonctions de transition de l'automate. L'espace de Bayes est défini comme suit :

- une cellule ayant l'état « 0 » passe à l'état « 1 » si et seulement si elle possède entre F_b et F_h cellules voisines à l'état « 1 » ;
- une cellule reste à l'état « 1 » si et seulement si elle a entre E_b et E_h cellules à l'état « 1 ».

Les automates de cet espace sont notés E_bE_h/F_bF_h . La fonction de transition du jeu de la vie est représentée dans cet espace par $F = (23/33)$. Cet espace est réduit de 2^{512} à 1296 fonctions de transition possibles.

La fonction de transition d'un automate cellulaire A est notée :

$$F = (E_bE_h/F_bF_h)\{m_k\}_{0 < k \leq s} \quad (24)$$

où (E_bE_h/F_bF_h) est la restriction à l'espace de Bayes de l'automate cellulaire A , $\{m_k\}$ est l'ensemble des éléments modifiés de F par rapport à cette restriction et s le nombre de ces modifications. Une modification est une inversion de l'état d'un élément de la fonction de transition F (passage de « 0 » à « 1 » et vice-versa).

Plusieurs approches ont été proposées par (Sapin, et al., 2003) permettant de chercher une fonction de transition de l'automate donnant des planeurs ou des canons à planeurs. L'algorithme évolutionnaire utilisé pour trouver la fonction de transition de l'automate acceptant des planeurs utilise une fonction objectif qui prend en compte l'évolution spatiale, le nombre de cellules vivantes et la présence de planeurs. A chaque itération de l'algorithme évolutionnaire, un ensemble de cellules à l'état « 1 » est généré dans un carré de taille C dans une grille carrée de taille L . Pendant cette évolution deux scores S_1 , S_2 et un coefficient c sont calculés :

- Pour calculer le premier score, la surface du rectangle englobant est prise en compte. Elle représente la dimension du plus petit rectangle contenant toutes les cellules vivantes. La dimension de ce rectangle augmente en présence des planeurs. Ce score représente donc la capacité de l'automate à s'étendre dans l'espace pour un nombre de générations L .
- Le deuxième score tient compte du nombre de cellules vivantes, qui tend à rester stable dans le jeu de la vie.
- Le coefficient c tient compte du déplacement du centre de gravité des cellules vivantes.

Pour détecter les planeurs, chaque groupe de cellules vivantes connexe est isolé et mis dans un univers de test, qui est une grille vide pour suivre son évolution. Un planeur est détecté :

- si la forme initiale est réapparue dans un autre endroit de la grille c'est un planeur ;
- si la forme est réapparue à la position de départ c'est une forme cyclique (périodique) ;
- sinon elle est en cours d'évolution.

La fonction objectif est donnée par l'équation suivante :

$$F = (S_1 * S_2 / cL^2) * (1 + \nu + \rho / 100) \quad (25)$$

En désignant par ν le nombre de planeurs ρ et le nombre de formes périodiques.

Pour construire la population initiale de l'algorithme évolutionnaire, une fonction de transition F est générée aléatoirement dans l'espace de Bayes. La population initiale est de taille N_c (taille de la fonction de transition). Elle est composée de $N_c / 5$ fonctions de transition F , $N_c / 5$ fonctions de transition F avec une mutation (un seul élément modifié), $A/5$ fonctions de transition F avec deux mutations, $N_c / 5$ fonctions de transition F avec trois mutations et $N_c / 5$ fonctions de transition F avec quatre mutations.

Un croisement à un point central est choisi. La mutation utilisée consiste à modifier une règle d'évolution dans la fonction de transition. La réinsertion dans la population est réalisée par conservation de certaines meilleures fonctions de transition.

Les étapes de l'algorithme évolutionnaire utilisé sont les suivantes :

- génération d'une population initiale ;
- évaluation de la population par la fonction objectif ;
- conservation de X meilleures fonctions de transition ;
- création de Y fonctions de transition par le choix de Y fonctions de transition auxquelles une mutation est appliquée ;
- création de $(A-X-Y)$ fonctions de transition par croisement appliqué à des couples de fonctions de transition choisis aléatoirement dans X .

Ainsi, une nouvelle population est obtenue et est composée des solutions conservées et des solutions obtenues par mutation et croisement.

Les planeurs obtenus avec cet algorithme ne se déplacent que dans un seul sens, ceci est dû à la mutation qui transforme un automate isotrope en un automate quelconque. Un automate cellulaire est isotrope lorsqu'il se déplace de la même façon dans toutes les directions.

Pour prendre en compte cette notion, (Sapin, 2003) a pris en compte la symétrie axiale et rotationnelle pour obtenir des planeurs qui se déplacent de la même manière dans toutes les directions et des canons à planeurs. Aucune information sur la façon dont est exprimée cette symétrie n'est donnée.

III.2.2.5 Extraction de la fonction de transition d'un automate cellulaire à partir d'un diagramme spatio-temporel

(Ichise & Ishida, 2008) ont proposé une méthode pour extraire la fonction de transition d'un automate cellulaire 1D à partir d'un diagramme spatio-temporel (espace-temps) et de définir le rayon du système d'interaction R . Ils ont étudié deux automates cellulaires : un automate cellulaire déterministe et un automate cellulaire stochastique.

Le rayon du système d'interaction peut évoluer au cours du temps dans (Ichise & Ishida, 2008). $E_t(c)$ est l'état à l'instant t de la cellule à la position x et $E(\sigma_k^R, t+1)$ l'état futur $(t+1)$ de la cellule en fonction de la combinaison k du système d'interaction de rayon R . $k \in [1, 2^{2R+1}]$.

Les étapes d'extraction de la fonction de transition sont données dans l'Algorithme 1. Deux cellules sont en contradiction lorsque, pour la même combinaison d'états du système d'interaction, elles possèdent deux états différents.

La qualité de la fonction de transition donnée par le taux d'identification T_{id} est évaluée en fonction du nombre de règles d'évolution locales inconnues, elle est donnée par :

$$T_{id} = 1 - \frac{\text{nombre de règles d'évolution locales inconnues}}{\text{nombre total de règles d'évolution locales}} \quad (26)$$

Algorithme III.1 : Algorithme d'extraction de la fonction de transition à partir du diagramme spatio-temporel

1. $R \leftarrow 0$
 2. **Répéter**
 3. $E(\sigma_k^R, t+1) \leftarrow \text{« inconnu » } \forall k = 1, 2^{2R+1}$
 4. **Pour** x de 1 à L **faire**
 5. Récupérer la combinaison k du système d'interaction de la cellule x
 6. **Si** $E(\sigma_k^R, t+1) = \text{« inconnu »}$ **alors**
 7. $E(\sigma_k^R, t+1) \leftarrow E_{t+1}(c);$
 8. **Sinon Si** $E(\sigma_k^R, t+1) \neq E_{t+1}(c)$ **alors**
 9. $E(\sigma_k^R, t+1) \leftarrow \text{contradiction}$
 10. **Fin Si**
 11. **Fin Si**
 12. **Fin Pour**
 13. $R \leftarrow R+1$
 14. **Jusqu'à** $E(\sigma_k^R, t+1) \neq \text{contradiction } \forall k = 1, 2^{2R+1}$
-

(Ichise & Ishida, 2008) ont utilisé cette méthode pour déterminer les règles de quelques automates cellulaires de Wolfram. Les taux d'identification T_{id} obtenus varient entre 0,875 et 0,937.

III.2.2.6 Problème de génération de formes

Dans cette partie, quelques problèmes de générations de formes seront présentés. Un modèle de génération de forme est présenté pour chaque type d'approche : approche chimique et approche par croissance cellulaire (automates cellulaires).

a. Génération de formes par approche chimique et algorithme de programmation génétique

a.1 Définition du problème

Bai et al (Bai, et al., 2008) se sont intéressés à la génération de formes 2D à partir d'un ensemble de cellules initialement dispersées qui se regroupent entre elles. Chaque cellule est un agent autonome appelé « primitive morphogénique ». Les cellules émettent dans leur environnement des substances chimiques ou morphogènes. Chaque cellule répond à cette stimulation par un déplacement dans la direction des gradients des champs chimiques cumulatifs de ses cellules voisines détectés sur sa surface. Ces champs chimiques sont définis par une fonction mathématique. Chaque fonction permet aux cellules de réagir différemment et de donner une forme.

a.2 Méthode de résolution (Bai, et al., 2008)

(Bai, et al., 2008) avaient pour objectif principal d'utiliser un processus de programmation génétique distribué pour déterminer la fonction des champs chimiques locaux qui permettraient aux cellules de se réarranger et donner la forme recherchée. Ces champs chimiques sont définis comme une fonction de distance d et d'angle θ entre deux primitives morphogéniques et en fonction de l'âge t de chaque primitive. Pour déterminer l'expression mathématique de ces champs chimiques, certains opérateurs sont utilisés, tels que la division, l'addition, le logarithme,....

A chaque instant, chaque primitive morphogénique émet une substance chimique dans son environnement. Ensuite, le champ chimique cumulé, ainsi que le gradient chimique local ∇C à sa surface, sont calculés. Ce gradient est utilisé pour déterminer la vitesse v de déplacement de cette primitive morphogénique, qui est proportionnelle à ce gradient chimique.

$$v = \lambda \cdot \nabla C \quad (27)$$

où λ est une constante.

Le déplacement de la primitive obtenu à chaque instant t est de :

$$\nabla x = v \cdot \Delta t \quad (28)$$

La comparaison de la forme obtenue par une fonction et celle recherchée par une fonction *fitness* est précédée par une étape d'alignement. Cette phase d'alignement permet d'aligner les centres de gravités de ces deux formes par une opération de translation et de rotation. La fonction *fitness* utilisée est basée sur le nombre de cellules communes dans les deux formes et le nombre de cellules totales dans la forme recherchée.

$$f = \frac{\sum p_i, p_i \in Sh \cap Sh^*}{\sum p_j, p_j \in Sh^*} \quad (29)$$

avec :

- p_i est une cellule ayant le même état dans la forme obtenue par une fonction d'agrégation Sh et la forme recherchée Sh^* ;
- p_j les cellules de la forme recherchée Sh^* .

Les paramètres du programme génétique sont :

- chaque individu est un arbre de taille maximale de 17 ;
- fonctions = {+, -, *, /, exp, log, sin, cos} ;
- terminaux = {E, d, t, θ }. E est un nombre réel, d est la distance entre deux primitives morphogéniques, t est le temps de simulation et θ est l'angle entre deux primitives ;
- sélection par tournoi avec 7 individus ;
- croisement uni-point avec une probabilité de 0,9 et mutation d'un sous-arbre avec une probabilité de 0,1 ;
- population de 100 individus ;
- nombre de générations 50.

a.3 Résultats et conclusion

Cet algorithme a été utilisé pour générer des formes pleines, telles que le carré, le losange, l'ellipse, le sablier, des ondes, ...etc. La Figure III.20 présente les formes de (Bai, et al., 2008). La première forme à gauche représente la forme initiale. Les autres formes sont les formes qu'ils ont recherchées.



Figure III.20 : Formes de (Bai, et al., 2008).

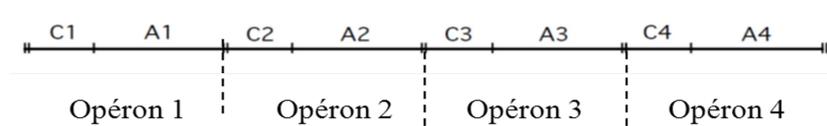


Figure III.22 : Exemple d'un chromosome à quatre opérons (De Garis & Brain Builder Group , 1999).

(De Garis & Brain Builder Group , 1999) ont proposé une fonction de *fitness* permettant de comparer deux formes donnée par l'équation :

$$fit = \frac{ins - 0.5 \cdot outs}{des} \quad (30)$$

- *ins* est le nombre de cellules ayant l'état « 1 » simultanément dans les deux formes ;
- *outs* est le nombre de cellules ayant l'état « 1 » dans la forme à comparer à l'extérieur de la forme recherchée ;
- *des* le nombre de cellules ayant l'état « 1 » dans la forme recherchée.

Cette méthode a été utilisée pour générer des formes pleines 2D et des formes en plusieurs phases.

b.2 Travaux de (Chavoya & Duthen, 2006)

(Chavoya & Duthen, 2006) ont proposé un modèle basé sur un automate cellulaire pour la génération de formes 2D pleines (carré, triangle, losange et cercle) à partir d'une cellule placée au centre de la grille.

La fonction de transition ainsi que le nombre de générations de l'automate cellulaire, sont déterminés par un algorithme génétique. Ce modèle est implémenté sur un environnement programmable de modélisation NetLogo qui est basé sur des systèmes multi-agents. Cet environnement est utilisé pour modéliser et simuler certains phénomènes naturels et sociaux.

Le modèle est un automate cellulaire binaire 2D. Ses cellules peuvent avoir deux états {0,1} pour une cellule vide et une cellule occupée. La fonction de transition de l'automate cellulaire est appliquée uniquement aux cellules vides, les cellules occupées restent occupées. Cette fonction de transition ne s'applique pas à toutes les cellules vides en même temps (système de mise à jour asynchrone). Le nouvel état d'une cellule vide est déterminé par une fonction de transition qui prend en considération uniquement les états précédents de ses cellules voisines définies selon différents systèmes d'interaction. Un algorithme génétique est utilisé pour retrouver la fonction de transition, ainsi que le nombre de générations de l'automate cellulaire.

Un chromosome est composé de deux champs codés en binaire : un champ d'action et un champ de contrôle (Figure III.23). Le champ d'action code la fonction de transition qui permet à l'automate d'évoluer. Quant au champ de contrôle, il représente le nombre de générations N_g que l'automate cellulaire applique cette fonction de transition.

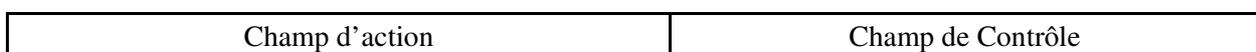


Figure III.23 : Schéma d'un chromosome de (Chavoya, 2007).

Le champ de contrôle donne le nombre de générations que l'automate doit réaliser. La taille de ce champ est donnée par le nombre de bits nécessaires pour coder le nombre de générations maximal donné par la moitié de la grille de l'automate cellulaire.

(Chavoya & Duthen, 2006; Chavoya, 2007) ont utilisé la fonction de *fitness* proposée dans (De Garis & Brain Builder Group, 1999) et ont considéré un système de mise à jour asynchrone. Par conséquent, deux mises à jour de l'automate cellulaire avec la même fonction de transition ne donnent pas forcément les mêmes évolutions.

Les méthodes proposées par (Chavoya & Duthen, 2006) ont été utilisées pour générer des formes 2D pleines binaires (Figure III.24). Ils ont utilisé quatre systèmes d'interaction :

- Von Neumann de rayon $R = 1$;
- Von Neumann de rayon $R = 2$;
- Moore de rayon $R = 1$;
- Margolus.

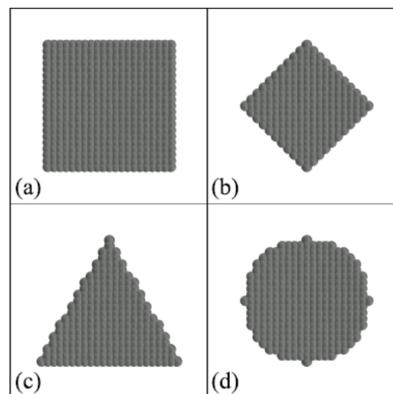


Figure III.24 : Formes pleines 2D de (Chavoya & Duthen, 2006).

Les résultats qu'ils ont obtenus sont présentés dans le Tableau III.3. Pour chaque forme, ils ont lancé 100 fois leur algorithme avec la fonction de transition obtenue par l'algorithme génétique. La moyenne et l'écart-type obtenus pour chaque forme avec les 100 exécutions de l'algorithme sont donnés.

	Von Neumann $R = 1$		Moore $R = 1$		Von Neumann $R = 2$		Margolus	
	Moyenne	Ecart-type	Moyenne	Ecart-type	Moyenne	Ecart-type	Moyenne	Ecart-type
(a)	0,738	0,008	0,993	0,003	0,887	0,015	1	0
(b)	1	0	0,805	0,040	0,773	0,028	0,880	0,018
(c)	0,580	0	0,950	0,011	0,909	0,023	0,860	0,010
(d)	0,868	0	0,932	0,013	0,875	0,017	0,928	0,006

Tableau III.3 : Résultats de Chavoya pour la génération de formes 2D pleines binaires.

b.3 Travaux de (Breukelaar & Back, 2005)

(Breukelaar & Back, 2005) se sont également intéressés à la génération de formes 2D et 3D par automate cellulaire et algorithme génétique. Parmi les formes à générer, ils ont traité le problème du damier.

Le problème du damier consiste à chercher la fonction de transition de l'automate cellulaire permettant de générer la forme d'un damier à partir d'une configuration initiale aléatoire de l'automate cellulaire. La forme du damier est une forme où deux cellules adjacentes au sens de Von Neumann doivent avoir un état différent (Figure III.25-(a)).

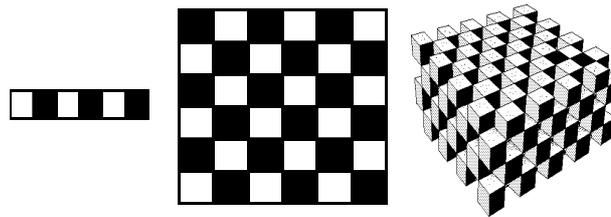
Breukelaar et Back (Breukelaar et Back 2005) ont proposé d'utiliser un algorithme génétique pour déterminer la fonction de transition de l'automate. La grille de l'automate cellulaire doit être périodique, c'est-à-dire que les cellules de la première ligne et celles de la dernière ligne sont adjacentes. En conséquence, ces cellules doivent respecter la règle du damier. Ceci s'applique aussi aux cellules de la première et de la dernière colonne. Pour construire un damier dans ces conditions, la taille de la grille de l'automate cellulaire doit être paire.

Trois automates cellulaires ont été utilisés dont les caractéristiques sont :

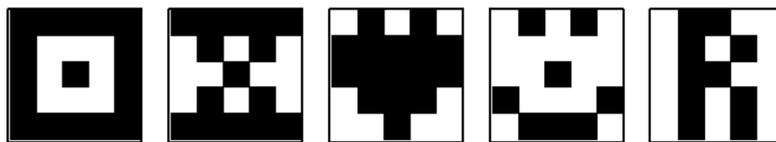
- automate cellulaire 1D de taille 150, 2D de taille 12*12 et 3D de taille 6*6*6,
- le nombre d'itérations de l'automate cellulaire est fixé à 320,
- un système d'interaction de Von Neumann de rayon r de taille 3 en 1D et 1 pour les automates 2D et 3D,
- un système de mise à jour synchrone.

La fonction de *fitness* consiste à compter le nombre de cellules de la grille de l'automate cellulaire qui respectent les conditions d'un damier.

Breukelaar et Back (Breukelaar et Back 2005) ont également été intéressés à la génération d'autres formes 2D par des automates cellulaires 2D. Les formes 2D recherchées sont présentées dans la Figure III.25. La première forme représente un carré avec un creux, la deuxième a la forme d'un sablier, la troisième a la forme d'un cœur, la quatrième est le « Smiley » et la cinquième est la lettre « R ».



(a) : Damier en 1D, 2D et 3D.



(b) : Formes 2D.

Figure III.25: Formes proposées par Back et Breukelaar (Breukelaar et Back 2005).

Un automate cellulaire à deux états $\{0, 1\}$ est utilisé. L'état « 0 » représente une cellule vide et l'état « 1 » désigne une cellule occupée. Les différentes formes sont générées à partir d'une cellule occupée placée au centre de la grille de l'automate cellulaire. Ce choix a été réalisé après avoir testé plusieurs autres configurations initiales de l'automate cellulaire. Pour générer des formes 2D, un automate cellulaire de taille 5*5 et un système d'interaction de Von Neumann sont considérés. Les conditions aux limites de l'automate cellulaire sont périodiques. Le nombre de générations maximal $N_{g_{max}}$ de l'automate cellulaire est fixé a priori par l'utilisateur.

b.4 Travaux de (Richards et al, 1990)

Une méthode d'extraction des règles d'évolution de l'automate cellulaire à partir de données expérimentales a été développée par Richards *et al* en 1990. Ces données représentent des formes qui évoluent dans un espace discret dans un temps discret. Elles représentent les formes

observées lors de la solidification du NH_4Br à partir une solution aqueuse saturée. Un algorithme génétique est utilisé pour rechercher la fonction de transition dans l'espace des fonctions de transition aléatoire de l'automate cellulaire permettant de reproduire au mieux le comportement observé dans ces données. Dans ce travail, la notion de multi-échelle et dynamique est prise en compte.

Les données sont des séquences d'images microscopiques de taille 512×512 pixels codées en niveaux de gris du processus de solidification du NH_4Br . Ces images ont été binarisées pour permettre à chaque pixel de prendre deux valeurs possibles : « 0 » si le pixel est une solution et « 1 » si c'est un solide. Elles ont été ensuite redimensionnées en taille 480×480 pixels. Ces dernières ont été réduites en taille 128×128 pixels. Ceci a permis d'avoir des images à deux échelles.

Un automate cellulaire 2D à deux états $\{0, 1\}$ est considéré. Afin d'inclure la dynamique à différentes échelles de temps et à différentes tailles d'échelle, l'état de chaque cellule de cette automate est déterminé non seulement en fonction des états de ses cellules voisines à l'instant t mais aussi à l'instant $t-1$. Aussi, pour intégrer la notion multi-échelle dans l'évolution de l'automate.

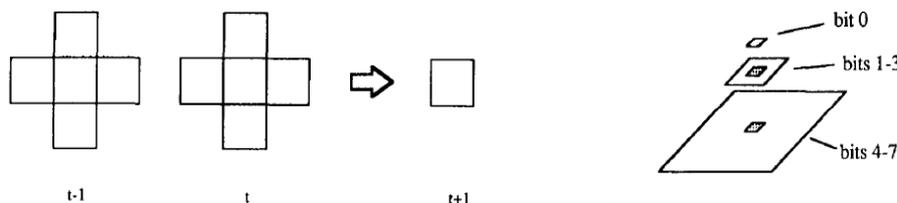


Figure III.26 : Système d'interaction de (Richards et al, 1990).

Pour capturer l'effet dynamique à différentes échelles, les données sont représentées en forme de pyramide. Chaque niveau de la pyramide représente les informations concernant cette échelle (contenue dans l'image correspondant à cette échelle). Au niveau bas de la pyramide, l'état de la cellule est considéré. Le second niveau contient les informations des cellules définies sur un voisinage de taille 3×3 . Le dernier niveau contient les informations des 9×9 cellules voisines.

Le codage des données est réalisé sur huit bits numérotés de 0 à 7. Le bit 0 représente l'état de la cellule considérée. Les bits de 1 à 3 sont les 3 bits significatifs de somme codée en binaire des cellules voisines (3×3) du niveau deux de la pyramide. Et enfin les 4 derniers bits représentent les bits significatifs de la somme des valeurs des cellules voisines (9×9) à la dernière échelle de la pyramide.

L'état d'une cellule $c_{i,j}$ est déterminé par une fonction de transition F .

$$E_{t+1}(c_{i,j}) = F(E_t(c_{i,j}), E'_t(c_{i,j}), E''_t(c_{i,j}), E_{t-1}(c_{i,j}), E'_{t-1}(c_{i,j}), E''_{t-1}(c_{i,j})) \quad (31)$$

Cette fonction prend en considération :

- l'état de la cellule à l'instant $t-1$ ($E_{t-1}(c)$), et à l'instant t ($E_t(c)$) ;
- l'état de la cellule à l'instant $t-1$ ($E'_{t-1}(c)$), et à l'instant t ($E'_t(c)$) à la première échelle ;
- l'état de la cellule à l'instant $t-1$ ($E''_{t-1}(c)$), et à l'instant t ($E''_t(c)$) à la deuxième échelle.

Cette méthode a permis de retrouver la fonction de transition de l'automate cellulaire permettant de créer des images proches de celles obtenues lors du processus de solidification du NH_4Br .

c. Génération de formes par réseaux de gènes régulateurs, automate cellulaire et algorithme génétique

(Chavoya, 2007) a proposé un modèle de croissance cellulaire synchronisé par un Réseau Artificiel Régulateur (RAR). Ce modèle a été utilisé pour résoudre le problème de génération de la forme du drapeau français par des automates cellulaires. Le modèle du drapeau français (*French flag model*) a été créé par Lewis Wolpert (Wolpert, 1969) à la fin des années soixante. Il consiste à étudier le phénomène de morphogénèse et à déterminer comment les cellules embryogéniques reçoivent et répondent aux informations génétiques et se différencient en différentes formes.

L'objectif de ce travail est de générer une forme en plusieurs phases (Figure III.27). Une phase correspond à l'application d'une fonction de transition pendant un certain nombre de générations à une forme initiale. Cette forme initiale est la forme finale de la phase qui la précède. Un gène régulateur est associé à chaque fonction de transition. Lorsqu'un gène régulateur est activé, la fonction de transition qui lui est associée est appliquée. L'activation des gènes régulateurs est réalisée grâce à un algorithme génétique.

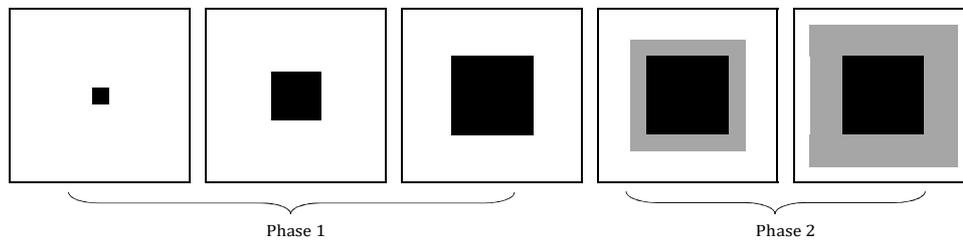


Figure III.27 : Génération d'une forme en deux phases.

Les génomes sont des référentiels de l'information génétique des organismes vivants. Ils sont composés d'une ou plusieurs chaînes d'ADN qui interagissent avec d'autres molécules, telles que l'ARN et les protéines. Un génome artificiel est une chaîne binaire, composée d'une série de gènes régulateurs constituant un RAR, suivis d'une série de gènes structurels.

La série des gènes structurels est un ensemble de fonctions de transition permettant d'obtenir des formes simples (carré, triangle, losange et un cercle). A chaque gène structurel correspond un ou plusieurs gènes régulateurs qui vont l'activer et lui permettre de s'exprimer.

Chavoya et Duthen ont utilisé une variante du modèle RAR proposé par (Banzhaf, 2003). Un génome est composé d'un ensemble de gènes régulateurs suivi de gènes structurels. Chaque gène régulateur possède un site d'activation, un site d'inhibition et une série de régions codantes de la protéine régulatrice. Les sites d'activation et d'inhibition ont pour objectif d'activer ou d'inhiber le gène régulateur. La série des régions permettent de produire une protéine régulatrice. Ces régions interagissant avec les sites régulateurs de tous les gènes régulateurs (Figure III.28). Le site d'activation d'un gène régulateur est activé lorsque la concentration de la protéine régulatrice qu'il produit est supérieure à celles des autres gènes régulateurs.

Le site d'activation, le site d'inhibition, les régions codantes de la protéine régulatrice, ainsi que la protéine régulatrice sont des chaînes de bits de même taille.

La taille du génome *TailleGenome* dans ce cas est donnée comme suit :

$$TailleGenome = n \times [(2 + k) \times r] + (m \times s) \quad (32)$$

- n : nombre de gènes régulateurs,

- k : nombre de régions codantes de la protéine régulatrice dans un gène régulateur,
- r : nombre de bits du site activateur et inhibiteur et d'une région codante de la protéine régulatrice,
- m : nombre de gènes structuraux ($m < n$),
- s : taille d'un gène structural (taille de la fonction de transition de l'automate).

Les valeurs des paramètres que Chavoya et Duthen ont utilisées sont : $n = 10$, $k = 5$, $r = 32$ et $s = 256$ (système d'interaction de Moore). Le nombre de gènes régulateurs n prend les valeurs 1, 2, 3 ou 6.

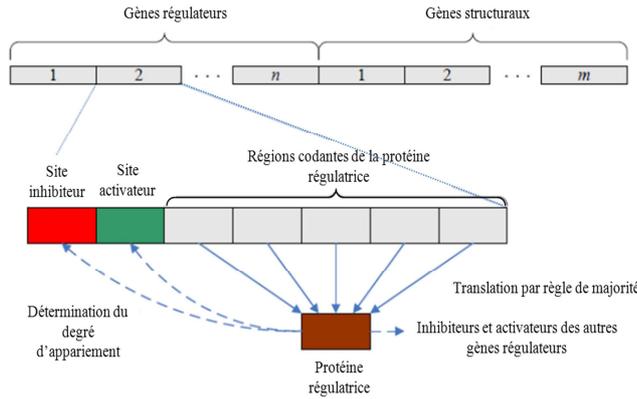


Figure III.28 : Structure d'un génome du modèle basique.

La synthèse de la protéine régulatrice est réalisée par l'application d'une règle de majorité bit à bit (premiers bits des régions codantes, ensuite deuxièmes bits, et ainsi de suite) sur l'ensemble des régions codantes, c'est-à-dire si le nombre de bits à « 1 » est majoritaire alors le bit correspondant de la protéine régulatrice prend la valeur « 1 », et vice-versa, si le nombre de « 0 » est majoritaire, le bit correspondant de la protéine régulatrice prend la valeur « 0 ».

Une fois la protéine régulatrice obtenue, elle va être comparée bit à bit avec tous les sites activateurs et les sites inhibiteurs des gènes régulateurs, afin de déterminer le degré d'interaction du réseau régulateur. La comparaison est réalisée avec une fonction XOR, qui est à « 1 » uniquement si les deux bits comparés sont complémentaires. Le nombre d'appariements entre une protéine et un site activateur est le nombre de fois où la comparaison par la fonction XOR est égale à « 1 » ; en d'autres termes, le nombre d'appariements représente le nombre de bits qui sont différents entre le site inhibiteur et la région codante.

La force d'activation (signe +) en_i ou d'inhibition (signe -) in_i du gène i avec $i = 1, n$ est définie comme suit :

$$en_i = \frac{1}{n} \sum_{j=1}^n c_j e^{\beta(u_{ij}^+ - u_{\max}^+)} \tag{33}$$

$$in_i = \frac{1}{n} \sum_{j=1}^n c_j e^{\beta(u_{ij}^- - u_{\max}^-)} \tag{34}$$

avec :

- n : nombre de gènes régulateurs,
- c_j : concentration de la protéine j ,

- β : constante permettant un réglage précis de de la force d'appariement,
- u_{ij}^+ : nombre d'appariements entre la protéine j et l'activateur i ,
- u_{ij}^- : nombre d'appariements entre la protéine j et l'inhibiteur i ,
- u_{\max}^+ : nombre d'appariements réalisables entre une protéine et un activateur,
- u_{\max}^- : nombre d'appariements réalisables entre une protéine et un inhibiteur.

Une fois les valeurs en et in obtenues pour tous les gènes régulateurs, les concentrations des protéines sont mises à jour par la formule suivante :

$$\frac{dc_i}{dt} = \delta(en_i - in_i)c_i \quad (35)$$

Les valeurs que Chavoya et Duthen ont utilisées sont : $\beta = 1$ et $\delta = 10^6$. δ est une constante qui régule le degré de changement de la protéine régulatrice.

D'autres extensions de ce modèle ont été proposées (Chavoya, et al., 2010).

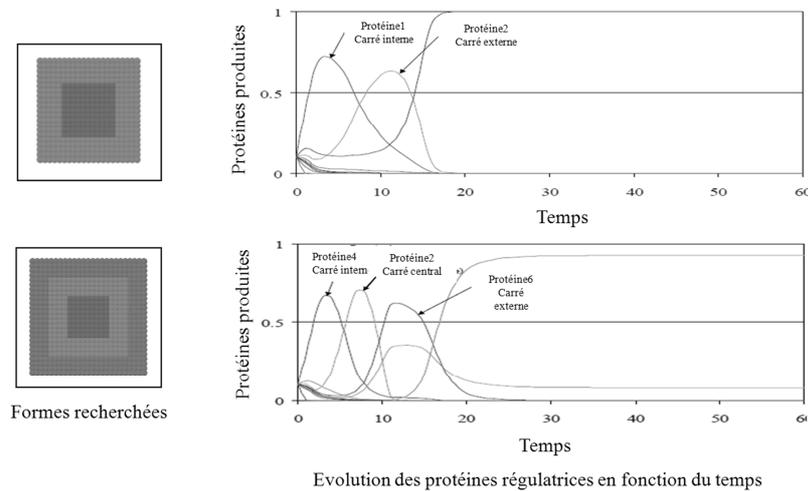


Figure III.29 : Formes créées par réseaux de gènes régulateurs, automate cellulaire et algorithme génétique (Chavoya, et al., 2010).

Nous constatons qu'une forme générée au cours d'une phase ne peut plus être modifiée lors de l'évolution de l'automate cellulaire à la phase suivante. L'évolution continue sans remettre en cause les formes obtenues pendant les phases précédentes.

III.2.2.7 Synthèse des travaux de la littérature

Dans le Tableau III.4, un résumé des travaux de la littérature est donné. La description de chaque problème, ainsi que l'automate cellulaire, le système d'interaction, le codage et la méthode d'optimisation qui sont utilisés par les auteurs, sont donnés.

Problème	Références	Description du problème	Automate cellulaire	Système d'interaction	Codage	Méthode d'optimisation	Application
Filtrage et segmentation d'images	(Rosin, 2006)	Chercher la fonction de transition de l'automate cellulaire permettant de filtrer, segmenter et déterminer les enveloppes convexes des objets présents des images binaires	Binaire 2D Fonction de transition additive	Moore	Chaîne binaire.	Recherche déterministe basée sur la méthode de Hill Climbing	Segmentation et extraction des objets d'une image binaire
Problème de majorité et de synchronisation	(Gacs, et al., 1978; Mitchell, 1993; Andre, et al., 1996; Breukelaar & Back, 2005)	- Problème de majorité : chercher une fonction de transition de l'automate cellulaire permettant à un ensemble de configurations initiales de l'automate cellulaire de converger vers des « 0 » si l'état « 0 » est majoritaire dans ces configurations initiales et à « 1 » si la majorité des cellules de ces configurations initiales sont à « 1 ». - Problème de synchronisation : cherche la fonction de transition de l'automate cellulaire qui, après un nombre de générations important, sa configuration finale oscille entre une configuration ne comportant que des « 0 » et une configuration ne comportant que des « 1 »	Conditions aux limites périodiques 1D de taille 149 2D de taille 13*13	1D : Von Neumann de rayon R = 3. 2D : Von Neumann et Moore	Chaîne binaire.	Algorithme génétique	
Recherche d'automates cellulaires universels	(Sapin, et al., 2003)	- Chercher la fonction de transition d'un automate cellulaire universel. Un automate cellulaire est universel lorsqu'il permet de générer des planeurs et des canons à planeurs	2D avec des conditions aux limites périodiques	Moore	Chaîne binaire. Avec utilisation de la symétrie axiale et rotationnelle. Simplification de l'espace de recherche par la règle de Bayes utilisant deux seuils sur la somme des états des cellules voisines pour choisir l'état suivant.	Algorithme évolutionnaire	Un automate cellulaire universel peut être utilisé comme une machine de Turing
Identification de la fonction de transition d'un automate cellulaire à partir d'un diagramme spatio-temporel	(Ichise & Ishida, 2008)	Extraction de la fonction de transition d'un automate cellulaire 1D à partir d'un diagramme spatio-temporel	1D déterministe et probabiliste	Rayon de Von Neumann variable	Chaîne binaire	Heuristique	

Génération de formes par système multi-agents	(Bai, et al., 2008)	Générer des formes 2D par systèmes multi-agents et programmation génétique	Représentation 2D	-	Ensemble de fonctions : l'addition, la division, le logarithme, les fonctions trigonométriques,...	Programmation génétique	Comprendre le processus de réarrangement des cellules permettant de créer des colonies de différentes formes
Génération de formes 2D et 3D binaires simples par automate cellulaire	(Chavoya & Duthen, 2006) (Breukelaar & Back, 2005)	Déterminer la fonction de transition de l'automate cellulaire et le nombre de générations qui permettent de donner une forme recherchée	2D	Pour les AC 2D. Von Neumann, Moore, 2D-radial et Margolus. 3D : Von Neumann et Margolus	Chaîne binaire	Algorithme génétique	
Génération de formes 2D et 3D à plusieurs états par automate cellulaire	(Chavoya, 2007)	Déterminer les fonctions de transition donnant des formes simples pour générer des formes complexes contrôlées par un réseau de gènes régulateurs	2D et 3D	2D : Von Neumann, Moore, 2D-radial et Margolus. 3D : Von Neumann et Margolus	Les chromosomes sont composés de deux parties principales : les gènes régulateurs et les gènes structurels. Les gènes régulateurs permettent d'activer ou d'inhiber les gènes structurels. Les gènes structurels sont les fonctions de transition de l'automate cellulaire permettant d'obtenir une forme simple	Algorithme génétique	
Problème d'embryogénèse et de différenciation cellulaire	(De Garis & Brain Builder Group, 1999)	Déterminer la fonction de transition de l'automate cellulaire, ainsi que les conditions correspondant à la mise à jour d'une cellule (direction, activation de la règle d'évolution locale ou pas,...)	2D	Von Neumann	Chaîne binaire. Le chromosome est composé de plusieurs gènes appelés opérons. Chaque opéron est composé de deux champs : un champ de conditions et un champ d'activation	Algorithme génétique	

Problème du drapeau français	(Gordon & Bentley, 2005)		Espace 2D		Utilisation d'un système multi-agents pour générer la forme du drapeau français	Algorithme génétique	
Extraction des règles d'évolution à partir de données expérimentales	(Richards et al, 1990)	Déterminer la fonction de transition de l'automate cellulaire à partir des données expérimentales représentant les formes observées lors de la solidification du NH4Br	2D	Von Neumann	Chaîne binaire	Algorithme génétique	Processus de solidification du NH4Br
Lignes de fusiliers	(Mazoyer, 1987; Yunès, 1993; Umeo, et al., 2004)	Ce problème consiste à partir d'une configuration initiale avec des cellules qui sont toutes à un état de « repos » sauf ,celles qui sont aux bords, trouver un ensemble d'états, ainsi que la fonction de transition de l'automate qui permettent à toutes les cellules d'arriver en même temps à un état jamais apparu avant, appelé état de « feu »	1D	Von Neumann de rayon 1.	Les règles sont définies par des symboles ou par des valeurs entières	Heuristique	
Identification des règles de l'automate cellulaire	(Adamatzky, 1994)	Déterminer la fonction de transition de l'automate cellulaire	1D	Von Neumann de rayon 1	Représentation booléenne de la fonction de transition de l'automate cellulaire		
Identification des règles et du système d'interaction de l'automate cellulaire	(Zhao & Billings, 2006)	Déterminer la fonction de transition de l'automate cellulaire, ainsi que le système d'interaction à utiliser	Binaire 1D et 2D	Von Neumann et Moore	Représentation booléenne et polynomiale de la fonction de transition de l'automate cellulaire	Algorithme orthogonal des moindres carrés (<i>Orthogonal least square algorithm</i>)	Segmentation d'images et simulation des ondes d'excitation cardiaque
Recherche des règles du L-système et de l'automate cellulaire	(Koza, 1993)	Déterminer la fonction de transition de l'automate cellulaire et les règles du L-Système	1D et 2D	Von Neumann de rayon 1	La fonction de transition est une fonction booléenne	Programmation génétique	

Tableau III.4 : Synthèse des travaux de la littérature.

III.2.2.8 Bilan

Nous avons recensé peu de travaux qui traitent le problème inverse des automates cellulaires dans la littérature. De plus, la plupart de ces travaux traitent le problème particulier de majorité.

La plupart des travaux de la littérature présentés utilisent les algorithmes évolutionnaires pour la recherche de la fonction de transition de l'automate cellulaire.

Tous les travaux de génération de formes que nous avons recensés sont appliqués pour la génération de formes pleines et ne peuvent pas être utilisés tels quels pour générer des formes quelconques.

Les modèles de générations de formes en plusieurs phases permettent d'obtenir des formes où les différentes phases sont bien distinctes dans l'espace. A la fin de chaque phase d'évolution, une autre phase prend le relais sans remettre en cause les phases d'évolution précédentes. Cependant, les formes obtenues aux phases précédentes ne changent pas et leurs cellules restent au même état. Des formes complexes où les cellules continuent à évoluer au cours du temps ne peuvent pas être générées avec ce type de modèle.

Conclusion

Dans ce chapitre, nous avons présenté les automates cellulaires. Ce sont des systèmes très utilisés pour simuler des systèmes dynamiques complexes. Ils sont basés sur le principe d'auto-organisation qui est une caractéristique de plusieurs systèmes biologiques. Ils permettent de faire émerger des comportements complexes, à partir d'un ensemble de règles locales simples.

Un état de l'art des problèmes inverses des automates cellulaires a été également présenté dans ce chapitre. Ces problèmes partagent l'objectif de déterminer les règles d'évolution locales (fonction de transition) permettant un comportement global souhaité. La plupart de ces travaux proposent de résoudre ce problème par des algorithmes évolutionnaires.

Parmi ces travaux, nous nous intéressons en particulier au problème inverse de génération de formes par automate cellulaire. Les modèles présentés dans la littérature pour la génération de formes sont des modèles qui s'intéressent à la génération de formes pleines et les méthodes d'optimisation utilisées pour chercher la fonction de transition de l'automate cellulaire sont des algorithmes évolutionnaires.

Dans cette thèse, nous proposons de résoudre ces problèmes par des méthodes d'optimisation à base de métaheuristiques. Ces méthodes seront présentées dans le chapitre suivant.

Chapitre IV. MODELISATION MICROSCOPIQUE : PROPOSITION DE METHODES DE RESOLUTION DU PROBLEME INVERSE

Introduction

Les automates cellulaires sont des modèles très utilisés pour la modélisation microscopique du processus de morphogénèse où l'on cherche à comprendre comment les cellules interagissent et communiquent entre elles au niveau local pour former un équilibre global donnant une certaine forme.

Pour comprendre ces processus, nous nous intéressons dans ce chapitre à la génération de formes quelconques par automates cellulaires. L'objectif principal est de déterminer les règles locales appliquées aux cellules du modèle d'automate cellulaire permettant de générer des formes prédéfinies à partir d'une forme initiale ainsi que le temps d'évolution de l'automate cellulaire (nombre de générations) (voir Figure IV.1).

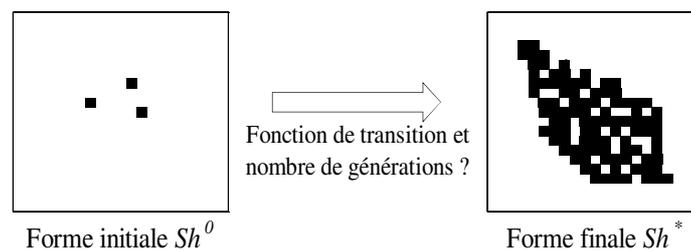


Figure IV.1 : Problème traité.

Ce chapitre est consacré aux différentes propositions pour la résolution du problème inverse des automates cellulaires. Il est scindé en plusieurs parties :

- Dans la première partie, nous définissons le problème traité. Les différentes notations qui seront utilisées sont données.
- Dans la deuxième partie, les critères de similarité utilisés pour comparer deux formes seront présentés. Le choix des critères de similarité s'est porté sur des mesures de distance entre les deux formes (le nombre de cellules identiques, l'écart quadratique et adaptation du critère de De Garis), l'information mutuelle pour mesurer la corrélation et les moments géométriques de Legendre pour leur invariance par échelle et par rotation dans l'espace.
- Dans la troisième partie, nous proposons une formalisation mathématique de ce problème, donnée pour les trois types d'automates cellulaires : 1D, 2D et 3D.
- Dans la quatrième partie, deux codages des solutions donnant trois approches de résolution du problème seront présentés. Dans le premier codage, la fonction de transition et le nombre de générations de l'automate cellulaire sont codés dans la solution et le problème est résolu

de façon simultanée. Dans le deuxième codage, seule la fonction de transition est codée dans la solution. Ainsi, le problème est décomposé. Ce codage a donné naissance à deux approches. Dans la première approche, le nombre de générations est calculé pour chaque fonction de transition. Dans la deuxième approche, la meilleure fonction de transition est recherchée pour chaque nombre de générations. Pour réduire les temps d'exécution, nous proposons d'évaluer les solutions voisines uniquement en cas de besoin.

- Pour résoudre ce problème inverse des automates cellulaires, nous proposons d'utiliser des méthodes d'optimisation à base d'heuristiques. Deux heuristiques d'amélioration (recherche locale déterministe avec le premier voisin et recherche locale déterministe avec le meilleur voisin) et un algorithme glouton sont présentées dans la cinquième partie.
- Les méthodes à base de métaheuristiques proposées concernent l'algorithme du Kangourou (recherche locale itérée), le recuit simulé et l'algorithme génétique. Ces métaheuristiques sont présentées dans la sixième partie.
- Une hybridation des heuristiques entre-elles et entre les heuristiques et métaheuristiques est proposée dans la septième partie.
- La huitième partie est consacrée aux cas particuliers des formes symétriques. Un codage exploitant cette propriété est proposé afin de réduire la taille de l'espace de recherche.

IV.1 Définition du problème

Soit un automate cellulaire de dimension d . Nous supposons que la taille L de la grille de cet automate cellulaire est la même pour chaque dimension.

Nous appelons une forme Sh , toute configuration d'un automate cellulaire. Elle est donnée pour un automate cellulaire 3D par :

$$Sh: G \rightarrow \sigma$$

$$c_{i,j,k} \mapsto Sh_{i,j,k} = E(c_{i,j,k}) \quad (36)$$

Avec :

G : la grille de l'automate cellulaire ;

σ : l'ensemble des états de cellules $\sigma = \{\sigma_1, \dots, \sigma_{|\sigma|}\}$;

$Sh_{i,j,k}$: l'état d'une cellule dans la forme Sh avec $i, j, k = 1, L$.

Nous notons :

Sh_i : l'état d'une cellule dans la forme Sh ($i = 1, L$) dans le cas d'un automate cellulaire 1D

$Sh_{i,j}$: l'état d'une cellule dans la forme Sh ($i, j = 1, L$) pour un automate cellulaire 2D.

Sh^0 : la forme initiale de l'automate cellulaire.

Sh^* : la forme que l'on cherche à obtenir.

Soit $Sh_{(F,Ng)}$ une forme obtenue par une fonction de transition F et un nombre de générations Ng , à partir d'une forme initiale Sh^0 . Elle est donnée par :

$$Sh_{(F,Ng)} = F^{Ng} (Sh^0) \quad (37)$$

Le problème traité consiste à trouver une fonction de transition F^* dans l'espace des fonctions de transition Π et un nombre de générations Ng^* qui maximise le critère de similarité C entre la forme recherchée Sh^* et la forme $Sh_{(F,Ng)}$.

$$\text{Rechercher } F^* \in \Pi \text{ et } Ng^* \in [0, Ng_{\max}] \text{ tels que } C(Sh_{(F^*, Ng^*)}, Sh^*) = \max_{\substack{F \in \Pi \\ Ng \in [0, Ng_{\max}]}} C(Sh_{(F, Ng)}, Sh^*) \quad (38)$$

Dans ce travail, afin d'éviter les effets de bords, le nombre de générations maximal Ng_{\max} est choisi de l'ordre de la moitié de la taille de la grille de l'automate cellulaire et le système de mise à jour de l'automate cellulaire est synchrone. Deux systèmes d'interaction sont considérés : Von Neumann et Moore.

IV.2 Critères de similarité

Pour mesurer la similarité entre la forme obtenue et la forme désirée nous avons proposé trois types de critères qui sont basés sur les mesures de distances et de quantité d'information. Ces critères sont : le nombre de cellules identiques, l'écart quadratique, l'adaptation du critère de De Garis aux formes creuses, l'information mutuelle et les moments de Legendre.

La formule de chaque critère est présentée pour le cas général d'un automate cellulaire 3D.

IV.2.1 Nombre de cellules identiques

Le premier critère de similarité que nous proposons pour mesurer la ressemblance entre deux formes est le nombre de cellules identiques. Il représente le nombre de cellules ayant le même état dans les deux formes à comparer. Nous utilisons le symbole de Kronecker $\delta_{i,j}$ pour calculer cette somme.

$$\delta_i^j = \begin{cases} 1 & \text{si } i = j \\ 0 & \text{si } i \neq j \end{cases} \quad (39)$$

Le nombre de cellules identiques entre une forme $Sh1$ et une forme $Sh2$ est donné par :

$$ID (Sh1, Sh2) = \sum_{i=1}^L \sum_{j=1}^L \sum_{k=1}^L \delta_{Sh1_{i,j,k}}^{Sh2_{i,j,k}} \quad (40)$$

Ce critère a pour avantage d'être facile à calculer et surtout à interpréter car la valeur du critère donne directement une indication sur le pourcentage de la ressemblance entre les deux formes.

IV.2.2 Ecart quadratique

Lorsque l'on s'intéresse uniquement aux automates cellulaires binaires, l'écart quadratique peut être utilisé comme mesure de similarité. L'objectif consiste à minimiser la différence $Equad$ entre les deux formes à comparer $Sh1$ et $Sh2$.

$$Equad(Sh1 - Sh2) = \sum_{i=1}^L \sum_{j=1}^L \sum_{k=1}^L (Sh1_{i,j,k} - Sh2_{i,j,k})^2 \quad (41)$$

IV.2.3 Adaptation du critère de De Garis aux formes creuses

Pour comparer des formes quelconques (pleines et creuses) à deux ou plusieurs états, nous proposons une adaptation de la fonction *fitness* proposée dans les travaux de (De Garis et Brain Builder Group, 1999; Chavoya, 2007) pour la génération de formes pleines binaires. L'adaptation du critère de De Garis pour comparer deux formes *Sh1* et *Sh2* est donnée par l'équation suivante :

$$fit(Sh1, Sh2) = \frac{(ins(Sh1, Sh2) - out(Sh1, Sh2))}{des(Sh1)} \quad (42)$$

Avec :

- $ins(Sh1, Sh2) = Card\left(Sh1_{i,j,k} \text{ tel que } (Sh1_{i,j,k} \neq 0) \wedge (Sh1_{i,j,k} = Sh2_{i,j,k})\right)$: le nombre de cellules ayant le même état dans les deux formes *Sh1* et *Sh2*.
- $out(Sh1, Sh2) = Card\left(Sh1_{i,j,k} \text{ tel que } (Sh1_{i,j,k} \neq Sh2_{i,j,k})\right)$: le nombre de cellules qui sont différentes dans les deux formes.
- $des(Sh1) = Card\left(Sh1_{i,j,k} \text{ tel que } (Sh1_{i,j,k} \neq 0)\right)$ le nombre de cellules non vides dans la forme *Sh1*.

IV.2.4 Information mutuelle

L'information mutuelle est une mesure utilisée en théorie de l'information (Gray, 1990). Elle permet de quantifier l'information apportée d'une configuration d'un système par rapport à une autre configuration. L'information mutuelle est très utilisée en traitement d'images et du signal en mesurant la corrélation entre deux images ou deux signaux.

L'information mutuelle considère toutes les transitions d'un état à un autre. Ces transitions d'états sont exprimées en termes de probabilités conjointes. Elle est donnée par l'équation suivante :

$$IM(Sh1, Sh2) = \sum_{e_1 \in \sigma} \sum_{e_2 \in \sigma} p(e_1, e_2, Sh1, Sh2) \ln \left(\frac{p(e_1, e_2, Sh1, Sh2)}{p(e_1, Sh1) p(e_2, Sh2)} \right) \quad (43)$$

$p(e_1, e_2, Sh1, Sh2) = Card\left(Sh1_{i,j,k} \text{ tel que } ((Sh1_{i,j,k} = e_1) \wedge (Sh2_{i,j,k} = e_2))\right) / L^3$: est la probabilité conjointe d'avoir une cellule ayant l'état e_1 dans la forme *Sh1* et un état e_2 dans la forme *Sh2*.

$p(e, Sh) = Card\left(Sh_{i,j,k} \text{ tel que } (Sh_{i,j,k} = e)\right) / L^3$: est la probabilité marginale d'avoir l'état e dans la forme *Sh*.

L'information mutuelle peut être formulée en fonction des entropies marginales ($H(Sh1)$ et $H(Sh2)$) et de l'entropie conjointe $H(Sh1, Sh2)$:

$$IM(Sh1, Sh2) = H(Sh1) + H(Sh2) - H(Sh1, Sh2) \quad (44)$$

L'entropie $H(Sh)$ est donnée par :

$$H(Sh) = - \sum_{e \in \sigma} p(e, Sh) \ln(p(e, Sh)) \quad (45)$$

L'entropie conjointe $H(Sh1, Sh2)$ de deux formes $Sh1$ et $Sh2$ se calcule par :

$$H(Sh1, Sh2) = - \sum_{e1 \in \sigma} \sum_{e2 \in \sigma} p(e1, e2, Sh1, Sh2) \ln(p(e1, e2, Sh1, Sh2)) \quad (46)$$

Les étapes de calcul de l'information mutuelle sont présentées dans l'Algorithme IV.1.

Algorithme IV.1 : Algorithme de calcul de l'information mutuelle

1. $p(e_1, e_2) \leftarrow 0; \forall e_1 = 0, |\sigma| - 1; \forall e_2 = 0, |\sigma| - 1$
 2. $p(e_1, Sh1) \leftarrow 0, \forall e_1 = 0, |\sigma| - 1$
 3. $p(e_2, Sh2) \leftarrow 0, \forall e_2 = 0, |\sigma| - 1$
 4. $p(Sh1_{i,j,k}, Sh2_{i,j,k}) \leftarrow p(Sh1_{i,j,k}, Sh2_{i,j,k}) + 1, \forall i, j, k = 1, L$
 5. $p(Sh1_{i,j,k}, Sh1) \leftarrow p(Sh1_{i,j,k}, Sh1) + 1, \forall i, j, k = 1, L$
 6. $p(Sh2_{i,j,k}, Sh2) \leftarrow p(Sh2_{i,j,k}, Sh2) + 1, \forall i, j, k = 1, L$
 7. $H(Sh1, Sh2) \leftarrow 0$
 8. $\forall e_1 = 0, |\sigma| - 1; \forall e_2 = 0, |\sigma| - 1$
 9. **Si** $p(e_1, e_2) \neq 0$ **faire**
 10. $p(e_1, e_2) \leftarrow p(e_1, e_2) / L^3$
 11. $H(Sh1, Sh2) \leftarrow H(Sh1, Sh2) + p(e_1, e_2) \ln(p(e_1, e_2))$
 12. **Fin si**
 13. $H(Sh1) \leftarrow 0; H(Sh2) \leftarrow 0$
 14. $\forall e = 0, |\sigma| - 1$
 15. **Si** $p(e, Sh1) \neq 0$ **faire**
 16. $p(e, Sh1) \leftarrow p(e, Sh1) / L^3$
 17. $H(Sh1) \leftarrow H(Sh1) + p(e, Sh1) \ln(p(e, Sh1))$
 18. **Fin Si**
 19. **Si** $p(e, Sh2) \neq 0$ **faire**
 20. $p(e, Sh2) \leftarrow p(e, Sh2) / L^3$
 21. $H(Sh2) \leftarrow H(Sh2) + p(e, Sh2) \ln(p(e, Sh2))$
 22. **Fin Si**
 23. $IM(Sh1, Sh2) \leftarrow H(Sh1) + H(Sh2) - H(Sh1, Sh2)$
-

IV.2.5 Moments de Legendre

Les moments de Legendre ont été introduits par (Teague, 1980) pour l'analyse d'images. Depuis, ils ont été très utilisés dans le domaine du traitement d'images, reconnaissance des formes, compression d'images, ... Le principal avantage des moments de Legendre, est qu'ils sont invariants par translation, par échelle et par rotation.

Dans ce travail, nous proposons d'utiliser ces moments de Legendre pour comparer deux formes. La formule des moments de Legendre présentée dans (Shu, et al., 2000; Hosny, 2010) utilisée pour le traitement d'images peut être adaptée au cas des automates cellulaire en 3D en remplaçant le pixel de l'image par une cellule de l'automate cellulaire et l'intensité d'un pixel par l'état d'une cellule.

Pour prendre en compte le changement d'échelle, les polynômes de Legendre se calculent dans l'intervalle $[-1, 1]$. Les coordonnées de chaque cellule $(i, j, k) \in L^3$ sont reportées et normalisées

dans l'intervalle $[-1, 1]$. Pour chaque cellule de coordonnée (i, j, k) , nous associons les coordonnées normalisées (x_i, y_j, z_k) telles que :

$$x_i = \frac{2i}{L} - 1 ; y_j = \frac{2j}{L} - 1 \text{ et } z_k = \frac{2k}{L} - 1 \quad (47)$$

Les moments de Legendre d'ordre $(p + q + r)$ d'une forme Sh peuvent être calculés par cette équation :

$$M_{pqr}(Sh) = \frac{(2p+1)(2q+1)(2r+1)}{L^3} \sum_{i=1}^L \sum_{j=1}^L \sum_{k=1}^L P_p(x_i) P_q(y_j) P_r(z_k) Sh_{i,j,k} \quad (48)$$

p, q et r sont les ordres des polynômes de Legendre dans chaque dimension. Les ordres des polynômes de Legendre sont strictement supérieurs à 1.

$P_p(x_i)$, $P_q(y_j)$ et $P_r(z_k)$ sont respectivement les polynômes de Legendre d'une cellule avec les coordonnées normalisées (x_i, y_j, z_k) .

Le polynôme de Legendre $P_p(x)$ d'ordre p d'une cellule à la position x dans une dimension peut être exprimé par la fonction récursive suivante :

$$P_{p+1}(x) = \frac{(2p+1)}{(p+1)} x \cdot P_p(x) - \frac{p}{(p+1)} P_{p-1}(x) \quad (49)$$

Avec : $P_0(x) = 1$, $P_1(x) = x$. Les polynômes de Legendre $P_q(y)$ et $P_r(z)$ d'ordre q et r se calculent de la même façon que le polynôme de Legendre $P_p(x)$.

Deux formes identiques ont les mêmes moments de Legendre. Le problème d'optimisation consiste à trouver la forme qui a la valeur des moments de Legendre la plus proche de celle des moments de Legendre de la forme recherchée.

Maximiser la ressemblance entre deux formes $Sh1$ et $Sh2$ en utilisant ces moments de Legendre revient à minimiser l'écart ΔM entre les valeurs des moments géométriques de Legendre des deux formes à comparer.

$$\Delta M = M_{pqr}(Sh1) - M_{pqr}(Sh2) \quad (50)$$

Les étapes de calcul des moments géométriques de Legendre d'une forme Sh de dimension $d=3$ sont présentées dans l'Algorithme IV.2.

IV.3 Formalisation mathématique

Dans cette partie, nous proposons une formalisation mathématique du problème inverse des automates cellulaires. Nous commençons par le cas le plus simple d'un automate cellulaire 1D puis une généralisation aux cas 2D et 3D est présentée.

Dans le cas des automates cellulaires 1D et 2D, une première formalisation du problème au cas binaire avec des systèmes d'interaction de rayon $R = 1$ est donnée. Ensuite, une formalisation généralisée aux cas à plusieurs états est donnée.

Algorithme IV.2 : Algorithme de calcul des moments de Legendre.

1. $M_{pqr} \leftarrow 0; p > 1; q > 1; r > 1$
 2. **Pour** i de 1 à L **faire**
 3. **Pour** j de 1 à L **faire**
 4. **Pour** k de 1 à L **faire**
 5. $x_i \leftarrow \frac{2i}{L} - 1, y_j \leftarrow \frac{2j}{L} - 1, z_k \leftarrow \frac{2k}{L} - 1$
 6. $P_0(x_i) \leftarrow 1, P_1(x_i) \leftarrow x_i$
 7. $P_0(y_j) \leftarrow 1, P_1(y_j) \leftarrow y_j$
 8. $P_0(z_k) \leftarrow 1, P_1(z_k) \leftarrow z_k$
 9. **Pour** u de 1 à $p-1$ **faire**
 11. $P_{u+1}(x_i) \leftarrow \frac{(2u+1)}{(u+1)} x_i \cdot P_u(x_i) - \frac{u}{(u+1)} P_{u-1}(x_i)$
 12. **Fin Pour**
 13. **Pour** h de 1 à $q-1$ **faire**
 14. $P_{h+1}(y_j) \leftarrow \frac{(2h+1)}{(h+1)} y_j \cdot P_h(y_j) - \frac{h}{(h+1)} P_{h-1}(y_j)$
 15. **Fin Pour**
 16. **Pour** v de 1 à $r-1$ **faire**
 17. $P_{v+1}(z_k) \leftarrow \frac{(2v+1)}{(v+1)} z_k \cdot P_v(z_k) - \frac{v}{(v+1)} P_{v-1}(z_k)$
 18. **Fin Pour**
 19. $M_{pqr} \leftarrow M_{pqr} + P_p(x_i) \cdot P_q(y_j) \cdot P_r(z_k) \cdot Sh_{i,j,k}$
 20. **Fin Pour**
 21. **Fin Pour**
 22. **Fin Pour**
 23. $M_{pqr} \leftarrow \frac{(2p+1)(2q+1)(2r+1)}{L^3} \cdot M_{pqr}$
-

IV.3.1 Automate cellulaire 1D

Le problème inverse des automates cellulaires pour la génération de formes peut être formulé sous forme d'un problème d'optimisation avec des variables, des contraintes et une fonction objectif.

IV.3.1.1 Cas binaire

Dans le cas d'une automate cellulaire 1D binaire de taille L , le problème inverse peut être formulé comme suit :

a. Données

L'ensemble d'états $\sigma = \{0, 1\}$.

La forme initiale $Sh^0 = (Sh_i^0); i = 1, L$.

La forme recherchée $Sh^* = (Sh_i^*); i = 1, L$.

Le nombre de générations maximal $Ngmax$.

Le système d'interaction N utilisé pour faire évoluer l'automate cellulaire au cours du temps. Ce système d'interaction est caractérisé par son rayon d'interaction R . Chacune des cellules du système d'interaction x ($x = 1, 2R+1$) est affectée par un coefficient de pondération p_x . Lorsque

l'ordre des cellules dans le système d'interaction se fait de gauche à droite (la cellule à gauche a le poids le plus fort 2^{2R} et la cellule à droite le poids le plus faible $2^0 = 1$). Nous considérons dans cette partie $R = 1$. La matrice des coefficients de pondération P peut être donnée par :

$$P = [p1 \ p2 \ p3] = [4 \ 2 \ 1] \quad (51)$$

b. Variables

Le nombre de générations de l'automate cellulaire Ng .

La fonction de transition $F = (f(e))$ qui est un vecteur de taille 2^3 ($e = 0, 7$). Elle est donnée par l'ensemble des règles locales $f(e)$.

c. Variables intermédiaires

Les configurations de l'automate cellulaire au cours du temps. Notons, Sh^t la forme obtenue à la génération t par l'automate cellulaire. $Sh^t = (Sh_i^t); i = 1, L$ et $t \in [1, Ngmax]$.

d. Contraintes

Les contraintes concernent le nombre de générations Ng , les règles d'évolution locales qui forment la fonction de transition F et les contraintes liées à l'évolution de l'automate cellulaire au cours du temps. Les contraintes peuvent être formulées comme suit :

Contrainte sur le nombre de générations : $0 \leq Ng \leq Ngmax$.

Contraintes sur la fonction de transition : les éléments de la fonction de transition doivent être binaires.

L'état d'une cellule à la génération t dépend des états de ses cellules voisines délimitées par le rayon R qui est déterminé en fonction :

$$Sh_i^{t+1} = f(4Sh_{i-1}^t + 2Sh_i^t + Sh_{i+1}^t); \forall i = 2, L-1; t = 1, Ngmax .$$

e. Objectif

L'objectif consiste à maximiser le critère de similarité C entre la forme $Sh_{(F, Ng)}$ obtenue par la fonction de transition F à la génération Ng et la forme recherchée Sh^* .

$$\text{Maximiser} \left(C \left(Sh_{(F, Ng)}, Sh^* \right) \right)$$

IV.3.1.2 Cas à plusieurs états

Dans le cas d'une automate cellulaire 1D de taille L à plusieurs états, le problème inverse peut être formulé comme suit :

a. Données

L'ensemble d'états σ .

La forme initiale $Sh^0 = (Sh_i^0); i = 1, L$.

La forme recherchée $Sh^* = (Sh_i^*); i = 1, L$.

Le nombre de générations maximal $Ngmax$.

Le système d'interaction N et son rayon d'interaction R . La matrice des coefficients de pondération P peut être donnée par :

$$P = [p_1 \quad \dots \quad p_x \quad \dots \quad p_{2R+1}] = [|\sigma|^{2R} \quad \dots \quad |\sigma|^{2R+1-x} \quad \dots \quad |\sigma|^0] \quad (52)$$

b. Variables

Les variables concernent :

Le nombre de générations de l'automate cellulaire Ng .

La fonction de transition $F = (f(e))$ qui est un vecteur de taille 2^{2R+1} ($e = 0, 2^{2R+1}-1$). Elle est donnée par l'ensemble des règles locales $f(e)$.

c. Variables intermédiaires

Les configurations de l'automate cellulaire au cours du temps $Sh^t = (Sh_i^t); i = 1, L$ et $t \in [1, Ngmax]$.

d. Contraintes

Les contraintes peuvent être formulées comme suit :

Nombre de générations $0 \leq Ng \leq Ngmax$.

Les éléments de la fonction de transition $f(e)$ doivent appartenir à l'ensemble d'états ($f(e) \in \sigma$).

L'état d'une cellule à l'instant t respecte la contrainte suivante :

$$\forall i = R+1, L-R; t = 1, Ng_{max}$$

$$Sh_i^{t+1} = f\left(\sum_{x=1}^{2R+1} Sh_{i+x-(R+1)}^t \cdot p_x\right) = f\left(\sum_{x=1}^{2R+1} Sh_{i+x-(R+1)}^t \cdot |\sigma|^{2R+1-x}\right) \quad (53)$$

e. Objectif

L'objectif consiste à maximiser le critère de similarité C entre la forme $Sh_{(F, Ng)}$ obtenue par la fonction de transition F à la génération Ng et la forme recherchée Sh^* .

$$\text{Maximiser}\left(C\left(Sh_{(F, Ng)}, Sh^*\right)\right) \quad (54)$$

IV.3.2 Automate cellulaire 2D

En passant d'un automate cellulaire 1D à un automate cellulaire 2D de taille L^2 , seuls les systèmes d'interaction avec les matrices des coefficients de pondération changent par rapport aux données. Les contraintes liées au nombre de générations Ng ainsi que l'objectif, restent les mêmes quelle que soit la dimension de l'automate cellulaire.

IV.3.2.1 Cas binaire

a. Données

L'ensemble d'états $\sigma = \{0, 1\}$.

La forme initiale $Sh^0 = (Sh_{i,j}^0); i = 1, L; j = 1, L$.

La forme recherchée $Sh^* = (Sh_{i,j}^*); i = 1, L; j = 1, L$.

Le nombre de générations maximal $Ngmax$.

Dans le cas d'un automate cellulaire binaire 2D et avec un système d'interaction de rayon R , nous appelons P_{VN} et P_{Moore} les matrices des coefficients de pondération des systèmes

d'interactions de Von Neumann et de Moore. Lorsque $R = 1$, la matrice des coefficients de pondération P_{VN} est donnée comme suit :

$$P_{VN} = \begin{bmatrix} 0 & 2^3 & 0 \\ 2^2 & 2^4 & 2^0 \\ 0 & 2^1 & 0 \end{bmatrix} \quad (55)$$

Dans le cas du système d'interaction de Moore, la matrice des coefficients de pondération P_{Moore} est donnée par :

$$P_{Moore} = \begin{bmatrix} 2^7 & 2^6 & 2^5 \\ 2^4 & 2^8 & 2^3 \\ 2^2 & 2^1 & 2^0 \end{bmatrix} \quad (56)$$

b. Variables

Les variables sont :

Le nombre de générations de l'automate cellulaire Ng .

La fonction de transition $F = (f(e))$, $\forall e = 0, 2^n - 1$, avec n la taille du système d'interaction. Dans le cas du système d'interaction de Von Neumann, n vaut 5 et dans le cas du système de Moore n vaut 9. n correspond au nombre de coefficients de pondération non nuls.

c. Variables intermédiaires

Les configurations de l'automate cellulaire $Sh^t = (Sh_{i,j}^t); i = 1, L, j = 1, L$ et $t \in [1, Ngmax]$.

d. Contraintes

Les contraintes sont données par :

Nombre de générations $0 \leq Ng \leq Ngmax$.

Les éléments de la fonction de transition $f(e)$ doivent appartenir à l'ensemble d'états $(f(e) \in \sigma)$.

L'état d'une cellule à l'instant t respecte la contrainte suivante :

Cas du système d'interaction de Von Neumann :

$$\begin{cases} \forall i = 2, L-1; j = 2, L-1; t = 1, Ng_{\max} \\ Sh_{i,j}^{t+1} = f(16Sh_{i,j}^t + 8Sh_{i-1,j}^t + 4Sh_{i,j-1}^t + 2Sh_{i+1,j}^t + Sh_{i,j+1}^t) \end{cases} \quad (57)$$

Cas du système d'interaction de Moore :

$$\begin{cases} \forall i = 2, L-1; j = 2, L-1; t = 1, Ng_{\max} \\ Sh_{i,j}^{t+1} = f \left(\begin{array}{l} 256Sh_{i,j}^t + 128Sh_{i-1,j-1}^t + 64Sh_{i-1,j}^t + 32Sh_{i-1,j+1}^t \\ + 16Sh_{i,j-1}^t + 8Sh_{i,j+1}^t + 4Sh_{i+1,j-1}^t + 2Sh_{i+1,j}^t + Sh_{i+1,j+1}^t \end{array} \right) \end{cases} \quad (58)$$

e. Objectif

Avec pour objectif : maximiser $\left(C \left(Sh_{(F, Ng)}, Sh^* \right) \right)$.

IV.3.2.2 Cas à plusieurs états

a. Données

L'ensemble d'états σ .

La forme initiale $Sh^0 = (Sh_{i,j}^0); i = 1, L; j = 1, L$.

La forme recherchée $Sh^* = (Sh_{i,j}^*); i = 1, L; j = 1, L$.

Le nombre de générations maximal $Ngmax$.

Les matrices des coefficients de pondération P_{VN} du système d'interaction de Von Neumann lorsque $R = 1$ est donnée comme suit :

$$P_{VN} = \begin{bmatrix} 0 & |\sigma|^3 & 0 \\ |\sigma|^2 & |\sigma|^4 & |\sigma|^0 \\ 0 & |\sigma|^1 & 0 \end{bmatrix} \quad (59)$$

Dans le cas du système d'interaction de Moore, la matrice des coefficients de pondération P_{Moore} est donnée par :

$$P_{Moore} = \begin{bmatrix} |\sigma|^7 & |\sigma|^6 & |\sigma|^5 \\ |\sigma|^4 & |\sigma|^8 & |\sigma|^3 \\ |\sigma|^2 & |\sigma|^1 & |\sigma|^0 \end{bmatrix} \quad (60)$$

b. Variables

Les variables sont :

Le nombre de générations de l'automate cellulaire Ng .

La fonction de transition $F = (f(e)), \forall e = 0, |\sigma|^n - 1$, avec n le nombre de coefficients de pondération non nuls.

c. Variables intermédiaires

Les configurations de l'automate cellulaire $Sh^t = (Sh_{i,j}^t); i = 1, L, j = 1, L$ et $t \in [1, Ngmax]$.

d. Contraintes

Les contraintes sont :

Nombre de générations $0 \leq Ng \leq Ngmax$.

Les éléments de la fonction de transition $f(e)$ doivent appartenir à l'ensemble d'états ($f(e) \in \sigma$).

Quel que soit le système d'interaction (Von Neumann ou Moore), l'état d'une cellule à l'instant t respecte la contrainte suivante :

$$\forall i, j = R+1, L-R; \forall t = 1, Ngmax$$

$$Sh_{i,j}^{t+1} = f \left(\sum_{x=1}^{2R+1} \sum_{y=1}^{2R+1} Sh_{i+x-(R+1), j+y-(R+1)}^t \cdot P_{x,y} \right) \quad (61)$$

e.Objectif

L'objectif consiste à maximiser $\left(C \left(Sh_{(F, Ng)}, Sh^* \right) \right)$.

IV.3.3 Automate cellulaire 3D à plusieurs états

Dans le cas général d'un automate cellulaire 3D à plusieurs états, le problème est formulé comme suit :

IV.3.3.1 Données

L'ensemble d'états σ .

La forme initiale $Sh^0 = (Sh_{i,j,k}^0); i = 1, L; j = 1, L; k = 1, L$.

La forme recherchée $Sh^* = (Sh_{i,j,k}^*); i = 1, L; j = 1, L; k = 1, L$.

Le nombre de générations maximal $Ngmax$.

Le système de d'interaction N de rayon R est utilisé pour faire évoluer l'automate cellulaire au cours du temps. Chacune des cellules du système d'interaction est affectée par un coefficient de pondération $p_{x,y,z}$. $P = (p_{x,y,z}); x = 1, 2R + 1; y = 1, 2R + 1; z = 1, 2R + 1$.

Lorsqu'une cellule n'est pas considérée dans le système d'interaction, son coefficient de pondération $p_{x,y,z}$ est nul. A l'inverse, lorsqu'une cellule est considérée dans le système d'interaction, son coefficient de pondération $p_{x,y,z}$ est non nul et dépend essentiellement du nombre d'états σ , de la taille du système d'interaction n et de m l'ordre de cette cellule dans le système d'interaction (avec $m < n$). Son coefficient de pondération est donné par $|\sigma|^{n-1-m}$.

IV.3.3.2 Variables

Les variables du problème sont :

Le nombre de générations de l'automate cellulaire Ng .

La fonction de transition $F = (f(e)), \forall e = 0, |\sigma|^n - 1$, avec n le nombre de coefficients de pondération non nuls.

IV.3.3.3 Variables intermédiaires

Les configurations de l'automate cellulaire :

$$Sh^t = (Sh_{i,j,k}^t); i = 1, L; j = 1, L; k = 1, L \text{ et } t \in [1, Ngmax]$$

IV.3.3.4 Contraintes

Nombre de générations $0 \leq Ng \leq Ngmax$.

Les éléments de la fonction de transition $f(e)$ doivent appartenir à l'ensemble d'états ($f(e) \in \sigma$).

Les états des cellules à l'instant $t+1$ respectent la contrainte suivante :

$$\left\{ \begin{array}{l} \forall i = R + 1, L - R; j = R + 1, L - R; k = R + 1, L - R; t = 1, Ngmax \\ Sh_{i,j,k}^{t+1} = f \left(\sum_{x=1}^{2R+1} \sum_{y=1}^{2R+1} \sum_{z=1}^{2R+1} Sh_{i+x-(R+1), j+y-(R+1), k+z-(R+1)}^t \cdot P_{x,y,z} \right) \end{array} \right. \quad (62)$$

Les contraintes de ce problème ne sont pas linéaires et leur nombre dépend de la taille de la grille de l'automate cellulaire, du rayon R et du nombre de générations $Ngmax$.

IV.3.3.5 Objectif

L'objectif reste le même et ne change pas quel que soit l'automate cellulaire, nous cherchons à maximiser le critère de similarité C entre la forme $Sh_{(F,Ng)}$ obtenue par la fonction de transition F à l'instant Ng et la forme recherchée Sh^* .

$$\text{Maximiser} \left(C \left(Sh_{(F,Ng)}, Sh^* \right) \right) \tag{63}$$

IV.4 Codages d'une solution et approches de résolution

Nous proposons deux codages des solutions. Ces codages ont donné naissance à trois approches basées sur le couplage méthodes d'optimisation et automate cellulaire. Les méthodes d'optimisation proposées sont des heuristiques et des métaheuristiques.

IV.4.1 Détermination simultanée de la fonction transition (F) et du nombre de générations (Ng)

Ce codage est une extension de la représentation du chromosome utilisé dans (Chavoya & Duthen, 2006) pour générer non seulement des formes pleines mais aussi des formes creuses deux états ou plus.

IV.4.1.1 Codage de la solution

Comme Chavoya et Duthen, une solution $X = (F, Ng)$ est codée en base σ en deux parties (voir Figure IV.2).

- la première partie concerne la fonction de transition de l'automate cellulaire F . Chaque élément représente l'état d'une cellule c à l'instant $t+1$ en fonction de la combinaison (m) de son système d'interaction à l'instant t . Ces combinaisons sont données dans un ordre lexicographique. La taille de cette partie est donnée par Nc le nombre de combinaisons des états du système d'interaction.
- La seconde partie de la solution donne le nombre de générations Ng de l'automate cellulaire. Sa taille est donnée par le nombre d'éléments nécessaires pour coder le nombre de générations maximal $Ngmax$ en base σ .

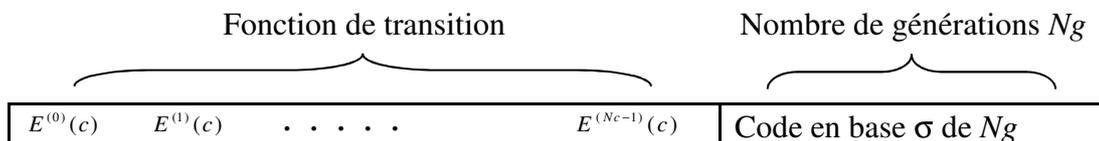


Figure IV.2 : Codage de solution $X = (F, Ng)$.

La taille d'une solution $TSol$ est la somme par la taille de la fonction de transition Nc et de la taille de la partie qui code le nombre de générations Ng .

Exemple

Un exemple de solution avec cette approche est illustré par la Figure IV.3 dans le cas d'un automate cellulaire 2D binaire et avec le système d'interaction de Von Neumann. La taille de la fonction de transition est de 2^5 . Les nombre de fonctions de transition total est de 2^{32} .

- Dans le cas d'un automate cellulaire binaire et comme nous considérons la cellule observée dans le système d'interaction, la taille de la fonction de transition est multipliée par 2 comparée à celle de Chavoya et Duthen, car elle décrit non seulement l'état d'une cellule lorsqu'elle est vide (cas de Chavoya et Duthen) mais aussi quand elle est à l'état « 1 ». La fonction de transition détermine la règle locale appliquée pour chaque combinaison d'états du système d'interaction. Les seize premières combinaisons sont équivalentes à celles de Chavoya et Duthen. Les seize combinaisons suivantes décrivent les règles d'évolution locales lorsque la cellule observée est occupée (état « 1 »). Le numéro de chacune des combinaisons est donné entre parenthèses. Pour chaque combinaison, l'état de la cellule observée et ceux de ses cellules voisines sont détaillés.
- Le nombre de générations Ng est égal à 4 et la taille de cette solution est égale à 35.

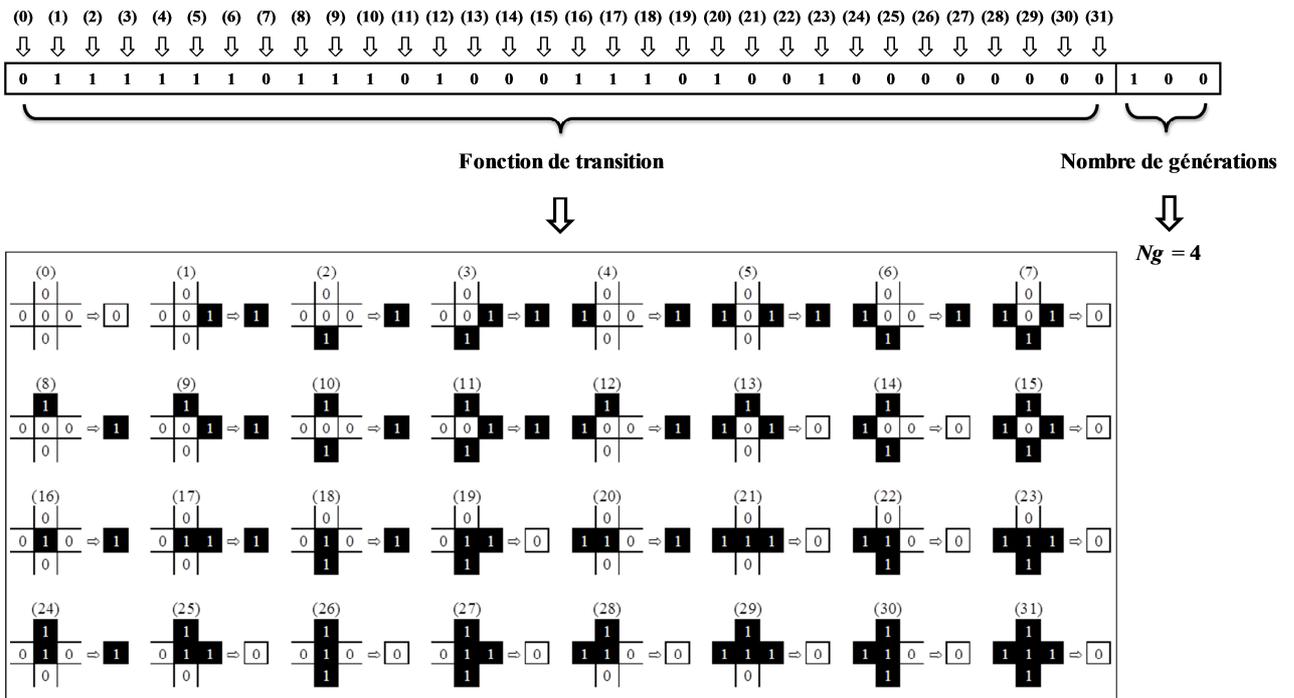


Figure IV.3: Exemple de solution.

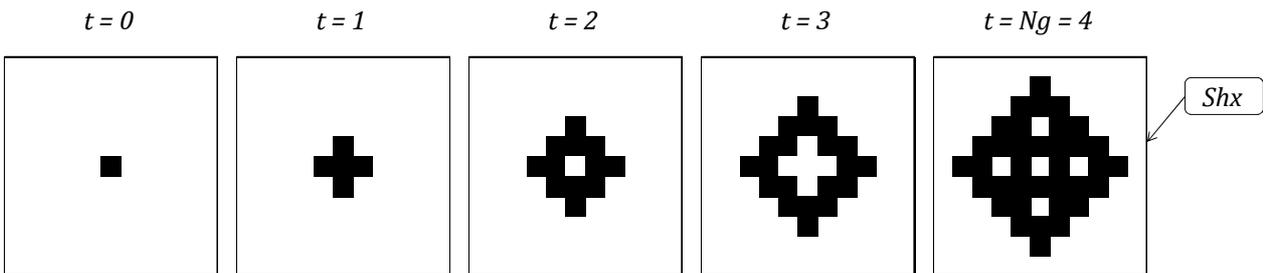


Figure IV.4 : Evolution de l'automate au cours du temps.

La Figure IV.4 donne les étapes d'évolution de l'automate au cours du temps en utilisant la solution $X = (F, Ng)$ de l'exemple de la Figure IV.3. La forme Sh_X est la forme obtenue par $Ng = 4$ applications de la fonction de transition F de la solution X . On peut également la noter $Sh_{(F,Ng)}$.

Ce codage considérant simultanément la fonction de transition F et le nombre de générations Ng donne naissance à une première approche (F, Ng).

IV.4.1.2 Approche (F, Ng)

Le principe de cette approche est décrit dans la Figure IV.5 (Aïboud, et al., 2010 ; Aïboud, et al., 2011b). La méthode d'optimisation part d'une solution initiale $X = (F, Ng)$ décrivant la fonction de transition F et le nombre de générations Ng . Cette solution est utilisée par l'automate cellulaire pour générer une forme à partir de la forme initiale par l'application de Ng fois la fonction de transition F . La forme obtenue Sh_X (ou $Sh_{(F,Ng)}$) est comparée à la forme recherchée Sh^* par le critère de similarité $C(Sh_{(F,Ng)}, Sh^*)$. A chaque itération, la méthode d'optimisation génère une nouvelle solution. A la fin, elle retourne la meilleure solution rencontrée. L'algorithme s'arrête lorsque le nombre d'itérations maximal est atteint ou lorsque la solution optimale est trouvée. Le critère d'acceptation d'une solution dépend de la méthode d'optimisation utilisée.

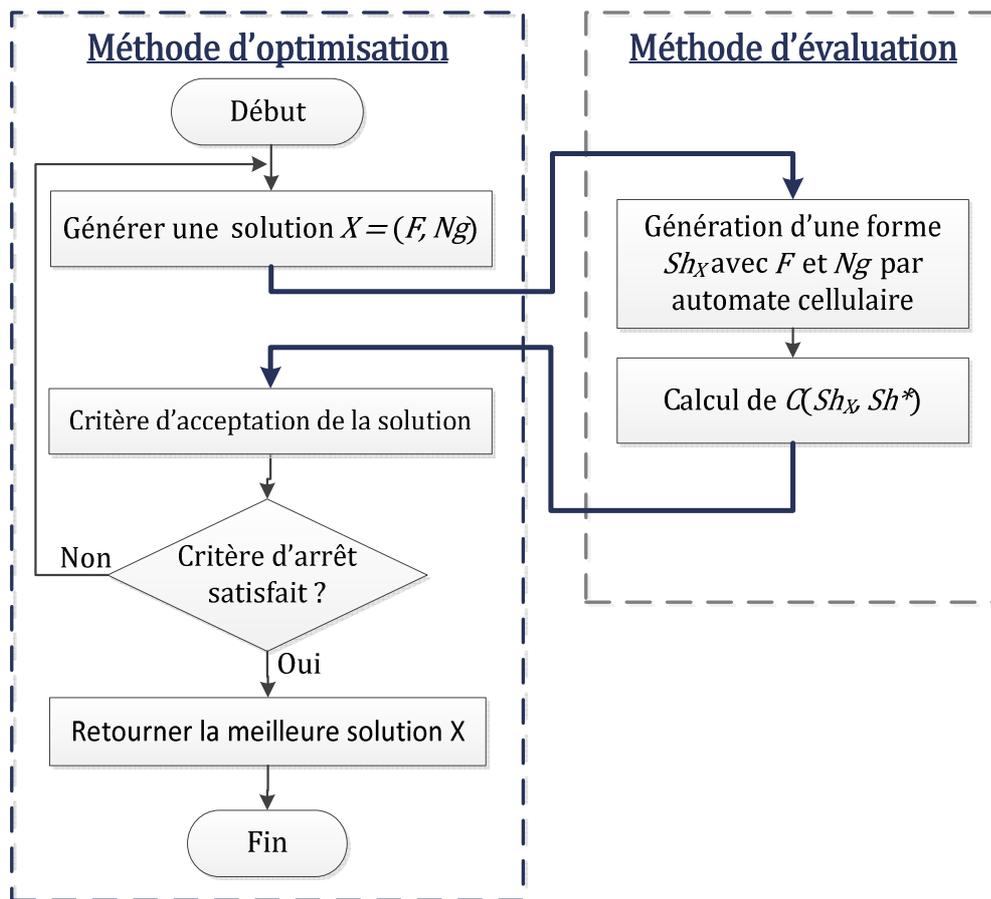


Figure IV.5 : Principe de l'approche (F, Ng).

Dans cette approche, la taille de l'espace des solutions Ω est donnée par le produit de l'espace des fonctions de transition et le nombre de générations maximal Ng_{max} donné par la deuxième partie d'une solution. La taille de la fonction de transition Nc est donnée par $|\sigma|^n$. La taille de

l'espace des fonctions de transition de l'automate cellulaire augmente de façon exponentielle avec le nombre d'états $|\sigma|$ et de la taille du système d'interaction n .

$$|\Omega| = |\sigma|^{Nc} \times Ngmax \quad (64)$$

La Tableau IV.1 présente la taille de l'espace des solutions Ω en fonction du nombre d'états $|\sigma|$, la taille du système d'interaction considéré avec un rayon $R = 1$ (Von Neumann ou Moore) et le nombre de générations maximal $Ngmax$.

	Von Neumann (5 cellules voisines)				Moore (9 cellules voisines)			
	$Ngmax = 3$	$Ngmax = 7$	$Ngmax = 9$	$Ngmax = 15$	$Ngmax = 3$	$Ngmax = 7$	$Ngmax = 9$	$Ngmax = 15$
$ \sigma = 2$	1,72E+10	3,44E+10	3,87E+10	6,87E+10	5,36E+154	1,07E+155	1,21E+155	2,15E+155
$ \sigma = 3$	7,85E+116	2,35E+117	7,85E+116	7,06E+117	0,15E+9392	∞	∞	∞
$ \sigma = 4$	∞	∞	∞	∞	∞	∞	∞	∞

Tableau IV.1 : Taille de l'espace des solutions Ω .

IV.4.2 Détermination séparée de F et Ng

Dans cette approche, nous proposons de dissocier le problème et de déterminer un paramètre connaissant le second.

IV.4.2.1 Codage de la solution

Une solution $X = (F)$ est restreinte à la fonction de transition de l'automate cellulaire (voir Figure IV.6). Sa taille $TSol$ est donnée par la taille d'une fonction de transition Nc .

Fonction de transition

$E_{t+1}^{(0)}(c)$	$E_{t+1}^{(1)}(c)$	• • • • •	$E_{t+1}^{(Nc-1)}(c)$
--------------------	--------------------	-----------	-----------------------

Figure IV.6 : Codage de la solution $X = (F)$

Avec ce codage, l'espace des solutions Ω représente l'espace des fonctions de transition de l'automate cellulaire.

$$|\Omega| = |\sigma|^{Nc} \quad (65)$$

La taille de l'espace des solutions Ω ne dépend plus du nombre de générations maximal $Ngmax$.

Ce codage donne deux approches :

- La première approche est appelée (Ng/F) . Une partie de la complexité du problème est déléguée à l'automate cellulaire qui détermine le nombre de générations Ng qui donne la meilleure fonction de transition F .
- La deuxième approche est appelée (F/Ng) . Nous cherchons la meilleure fonction de transition F pour chaque nombre de générations Ng .

Dans ces deux approches, le nombre de générations Ng est déterminé par énumération.

IV.4.2.2 Détermination de Ng connaissant F (Approche (Ng/F))

Les étapes de notre algorithme sont présentées dans la Figure IV.7 (Aïboud, et al., 2011a; Aïboud, et al., 2011c). A chaque itération, la méthode d'optimisation produit à l'automate cellulaire une ou plusieurs solutions $X = (F)$. L'automate cellulaire génère plusieurs formes par

l'application de $Ngmax$ fois la fonction de transition F à la forme initiale. A chaque génération de l'automate cellulaire $t \in [1, Ngmax]$, nous comparons la forme obtenue $Sh_X = Sh_{(F,t)}$ avec la forme désirée Sh^* . La meilleure valeur du critère $C(Sh_{(F,Ng)}, Sh^*)$ obtenue est retenue comme critère de la solution X ainsi que le nombre de générations Ng auquel nous obtenons la meilleure valeur du critère. Le nombre de générations Ng est déterminé tel que :

$$C(Sh_{(F,Ng)}, Sh^*) = \max_{t=1, Ng_{max}} (C(Sh_{(F,t)}, Sh^*)) \tag{66}$$

Ensuite l'automate cellulaire retourne à la méthode d'optimisation, la valeur du critère ainsi que le nombre de générations associés à chaque solution X .

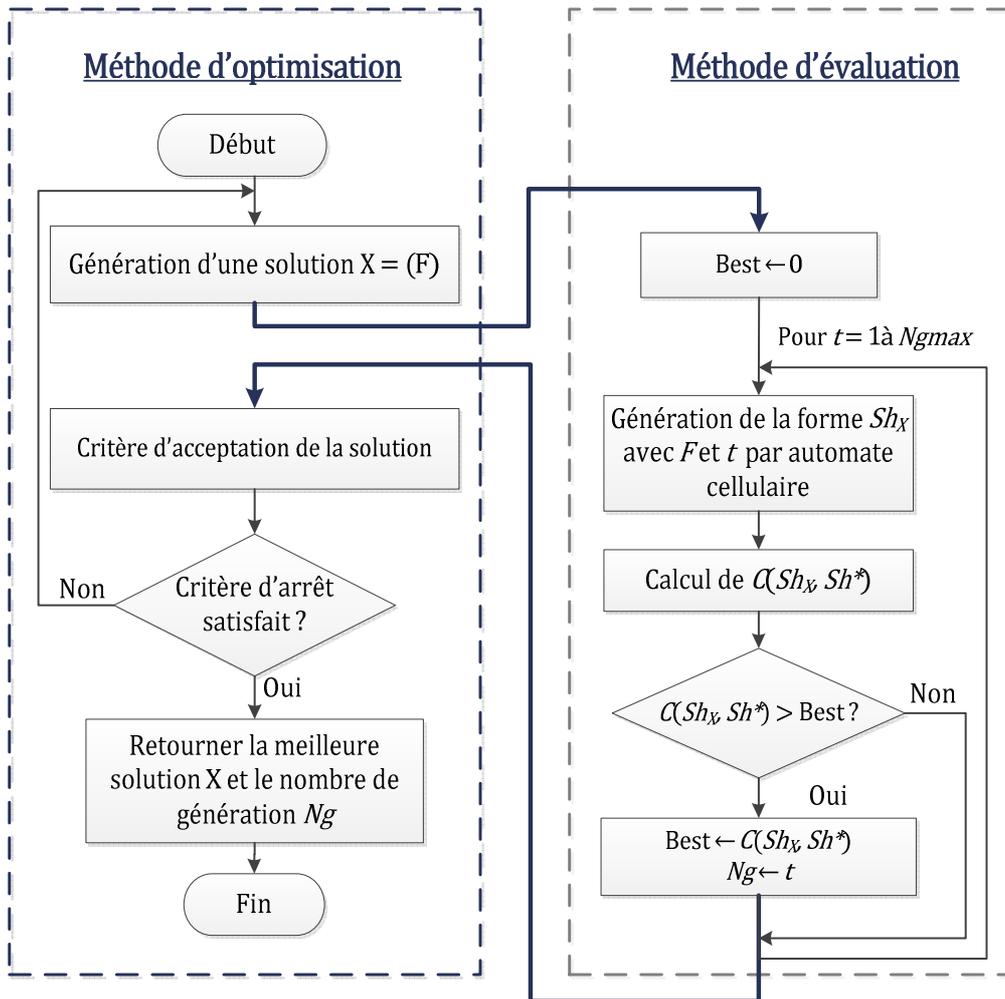


Figure IV.7: Principe de l'approche (Ng/F).

IV.4.2.3 Détermination de F connaissant Ng (Approche (F/Ng))

Dans cette approche (Figure IV.8), le nombre de générations t de l'automate cellulaire est initialisé à « 1 » et la méthode d'optimisation cherche la fonction de transition F qui donne la meilleure valeur du critère de similarité. Si la méthode d'optimisation ne trouve pas la solution optimale, le nombre de générations t de l'automate cellulaire est incrémenté et la méthode d'optimisation est relancée. Cette procédure est répétée jusqu'à ce que le nombre de génération $Ngmax$ soit atteint ou la solution optimale est atteinte (Aïboud, et al., 2013a).

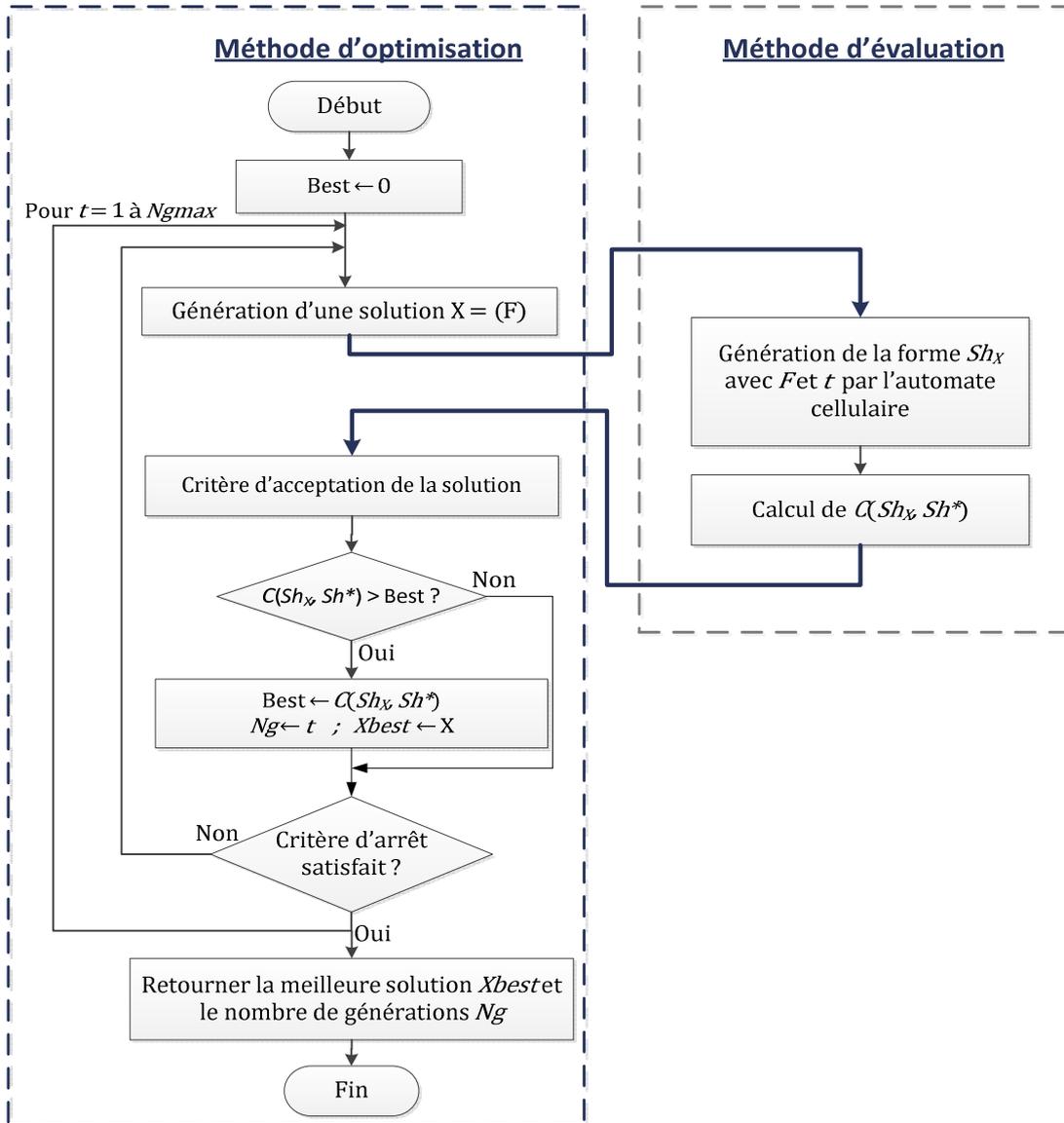


Figure IV.8. Principe de l'approche (F/Ng).

Nous résumons dans la Figure IV.9, les trois approches proposées :

- Approche (F,Ng) : détermination simultanée de F et Ng.
- Approche (Ng/F) : détermination de Ng connaissant F.
- Approche (F/Ng) : détermination de F connaissant Ng.

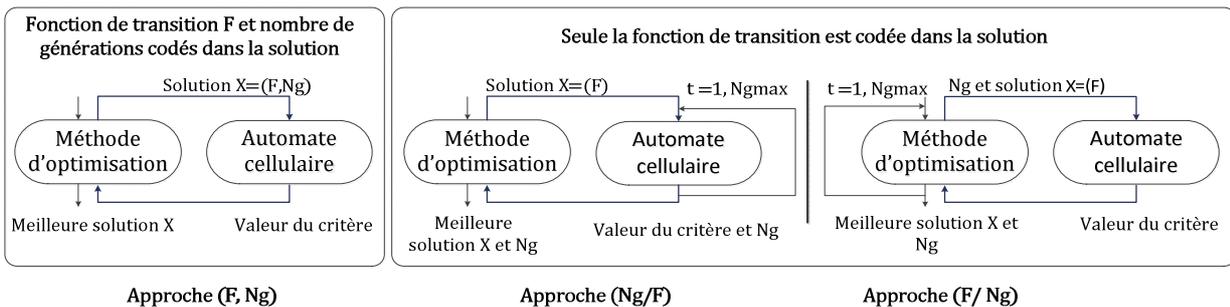


Figure IV.9 Principes des approches proposées.

La taille de l'espace de recherche Ω en fonction du nombre d'états est donnée par le Tableau IV.2. La taille de l'espace de recherche reste du même ordre d'une approche à une autre.

		Automate cellulaire 2D				Automate cellulaire 3D	
		(F, Ng)		(F/Ng) et (Ng/F)		(F, Ng)	(F/Ng) et (Ng/F)
		Von Neumann	Moore	Von Neumann	Moore	Von Neumann	Von Neumann
σ	2	3.87E+10	1.21E+155	4.29E+09	1.34E+154	3.06E+39	3.40E+38
	3	7.85E+116	∞	8.72E+115	∞	∞	∞

Tableau IV.2 : Taille de l'espace des solutions Ω en fonction des approches.

IV.4.2.4 Réduction de temps d'exécution

Dans la première approche proposée (F,Ng), à chaque évaluation d'une solution, l'automate cellulaire applique Ng fois la fonction de transition alors que dans les deux dernières approches, la fonction de transition est appliquée $Ngmax$ fois. Ces deux approches permettent de réduire l'espace de recherche mais elles augmentent les temps de calcul.

Face à cet inconvénient, nous proposons de réduire le nombre de fois où nous faisons appel à l'automate cellulaire (Aïboud, et al., 2013a). En fait, durant chaque itération de la méthode d'optimisation, la fonction de transition est appliquée $Ngmax$ fois et durant cette itération, il existe des règles locales qui ne sont jamais utilisées. Cependant, au lieu de faire appel à l'automate cellulaire à chaque itération, nous proposons d'utiliser l'automate cellulaire uniquement lorsque la modification de la fonction de transition peut avoir une conséquence sur le critère de similarité.

Pour cela, lorsque la fonction de transition est appliquée, les règles d'évolution locale utilisées et non utilisées sont identifiées. Si la règle d'évolution locale est non utilisée et si cette règle est sélectionnée pour être modifiée lors de la génération d'une nouvelle fonction de transition, la méthode d'optimisation accepte directement cette solution sans utiliser l'automate cellulaire car son critère n'a pas changé.

IV.5 Heuristiques

Nous proposons dans ce travail trois heuristiques. La première heuristique est un algorithme glouton. La deuxième et la troisième heuristique sont des heuristiques d'amélioration ou d'exploration locale.

IV.5.1 Algorithme glouton

Dans cette heuristique, la solution est construite étape par étape. Nous fixons le premier élément à l'état $\sigma_1 = 0$ et nous générons aléatoirement $Nrep$ solutions à l'exception du premier élément. Chacune de ces solutions est évaluée avec l'un des critères de similarité proposés et la moyenne $M(\sigma_1)$ des valeurs du critère est calculée. Ensuite, nous passons l'état du premier élément de toutes les $Nrep$ solutions, à l'état $\sigma_2 = 1$, nous réévaluons chacune de ces solutions et nous calculons la moyenne $M(\sigma_2)$ des valeurs du critère. Nous répétons cette procédure jusqu'à ce que l'état du premier élément soit de $\sigma_{|\sigma|}$ et $M(\sigma_{|\sigma|})$ soit calculée. A la fin l'état e qui donne la moyenne maximale $M(e)$ est attribué au premier élément de la solution à construire.

Pour générer le 2^{ème} élément, nous recommençons la procédure précédente en générant $Nrep$ solutions aléatoirement. L'état du premier élément ne change pas. Nous recalculons les moyennes $M(\sigma_1)$ à $M(\sigma_{|\sigma|})$ et nous attribuons l'état qui donne la valeur maximale de la moyenne au 2^{ème} élément. La même procédure est répétée pour déterminer les autres éléments de la solution.

Les étapes de cette heuristique sont données par l'Algorithme IV.3. $Nrep$ est le nombre solutions générées à chaque étape de l'algorithme glouton.

Algorithme IV.3 : Principe de l'algorithme glouton

1. **Pour** i de 1 à $TSol$ **faire**
2. $M(e) \leftarrow 0, \forall e \in \{\sigma_1, \sigma_{|\sigma|}\}$
3. **Pour** j de 1 à $Nrep$ **faire**
4. **Pour** l de $i+1$ à $TSol$ **faire**
5. Choisir uniformément et aléatoirement $X(l)$
6. **Fin pour**
7. **Pour** e de σ_1 à $\sigma_{|\sigma|}$ **faire**
8. $X(i) \leftarrow e$
9. $M(e) \leftarrow M(e) + C(Sh_x, Sh^*)$
10. **Fin Pour**
11. **Fin Pour**
12. $M(e) \leftarrow M(e) / Nrep, \forall e \in \{\sigma_1, \sigma_{|\sigma|}\}$
13. $X(i) \leftarrow e^*$ tel que $e^* = \max_{e \in \{\sigma_1, \sigma_{|\sigma|}\}}(M(e))$
14. **Fin Pour**
15. Retourner X

Exemple

Nous expliquons dans cet exemple les étapes de construction d'une solution dans le cas d'un automate cellulaire binaire. Nous attribuons l'état « 0 » au premier élément de la solution, les autres éléments sont générés aléatoirement et nous évaluons la solution obtenue. Ensuite nous modifions l'état de ce premier élément (passage à « 1 ») et nous réévaluons la solution ainsi obtenue. Nous répétons cette procédure $Nrep$ fois. A la fin, nous calculons la moyenne $M0$ des valeurs des critères obtenues avec les solutions où le premier élément est à « 0 » et la moyenne $M1$ des valeurs des critères où le premier élément est à « 1 ». L'état qui donne la meilleure moyenne sera attribué au premier élément (Figure IV.10).

Lorsqu'un état est attribué au premier élément (exemple état «1»), nous appliquons la même procédure pour le choix de l'état du deuxième élément sans modifier l'état du premier élément. La Figure IV.11 illustre la procédure pour déterminer l'état du deuxième élément. Le nombre de fois $Nrep$ où les éléments manquants sont générés aléatoirement décroît de « 1 » en allant du premier élément au dernier.

Cette procédure est répétée jusqu'à ce que tous les éléments de la solution soient déterminés. Le nombre $Nrep$ peut être soit constant, soit il décroît en partant du premier au dernier élément.

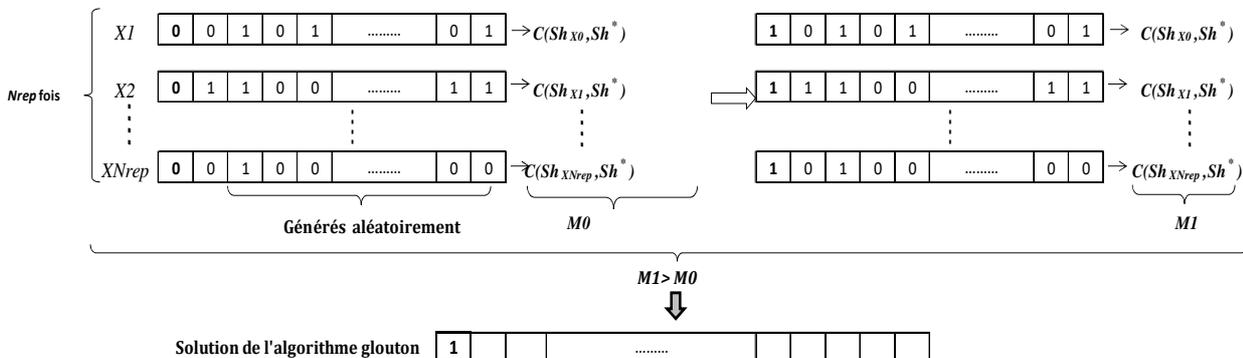


Figure IV.10 : Détermination de l'état du premier élément.

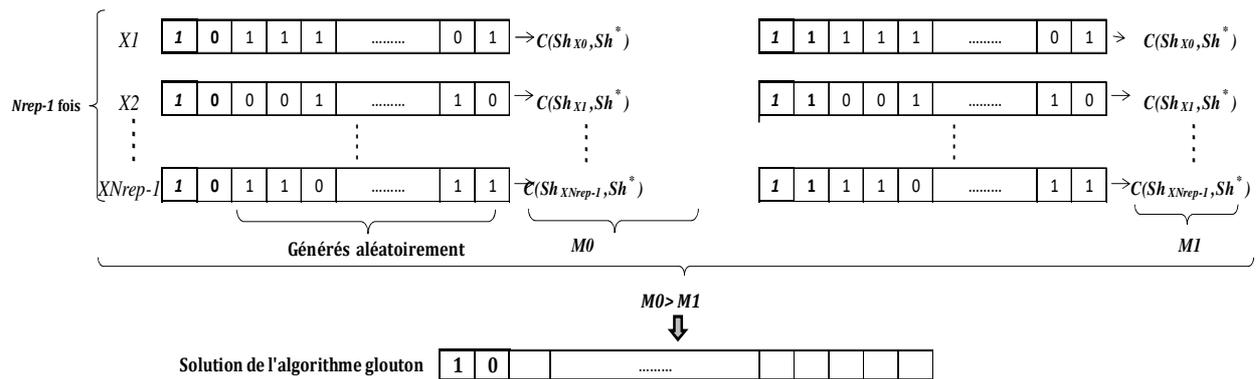


Figure IV.11 : Détermination du deuxième élément.

IV.5.2 Recherche locale déterministe avec le meilleur voisin

Cette méthode consiste à générer une solution initiale aléatoirement. A chaque itération de l'heuristique, toutes les solutions voisines selon le système de voisinage sont générées et la meilleure d'entre-elles est retenue. Ce processus est réitéré jusqu'à ce qu'on n'améliore plus la solution (optimum local). Les étapes de cette heuristique, lorsque nous utilisons un système de voisinage qui modifie un seul élément de la solution, sont données par l'Algorithme IV.4.

Ces solutions voisines sont générées soit avec le système de voisinage $\Gamma1$ soit avec le système de voisinage $\Gamma2$. Ces deux systèmes de voisinage seront décrits dans la section sur les métaheuristiques à voisinage.

Algorithme IV.4 : Principe de la recherche locale déterministe avec le meilleur voisin

1. Génération d'une solution initiale X de taille $TSol$
 2. $C_{Best} \leftarrow C(Sh_X, Sh^*)$
 3. $C_{obj} \leftarrow C(Sh^*, Sh^*)$
 4. Amelioration \leftarrow vrai
 5. **Tant que** $(C(Sh_X, Sh^*) < C_{obj})$ et $(Amelioration = vrai)$ **faire**
 6. Amelioration \leftarrow faux
 7. **Pour** l de 1 à $TSol$ **faire**
 8. **Pour** e de σ_1 à $\sigma_{|\sigma|}$ **faire**
 9. $X' \leftarrow X$
 10. $X'(i) \leftarrow e$
 11. **si** $(C(Sh_{X'}, Sh^*) > C_{Best})$ **alors**
 12. $I_{Best} \leftarrow l$
 13. $C_{Best} \leftarrow C(Sh_{X'}, Sh^*)$
 14. Amelioration \leftarrow vrai
 15. $E_{Best} \leftarrow e$
 16. **Fin si**
 17. **Fin Pour**
 18. **Fin Pour**
 19. **Si** $(Amelioration = vrai)$
 20. $X(I_{Best}) \leftarrow E_{Best}$
 21. **Fin si**
 22. **Fin tant que**
 23. Retourner X
-

L'algorithme de la recherche locale déterministe avec le meilleur voisin est donné en annexe A pour le cas où deux éléments sont modifiés.

IV.5.3 Recherche locale déterministe avec le premier voisin

Le principe de cette heuristique est le même que celui de l'heuristique précédente mais, à chaque itération, on énumère les solutions voisines selon le système de voisinage $\Gamma(X)$ de la solution courante jusqu'à ce que l'on trouve une solution qui améliore la solution courante. L'Algorithme 15 donne le principe de cette heuristique lorsqu'un seul élément de la solution est perturbé.

L'algorithme de la recherche locale déterministe avec le premier voisin est donné en annexe A pour le cas où deux éléments sont modifiés.

Algorithme IV.5 : Principe de la recherche déterministe avec le premier voisin avec Γ_1

1. Génération aléatoire d'une solution X de taille $TSol$
 3. $C_{obj} \leftarrow C(Sh^*, Sh^*)$
 4. Amélioration \leftarrow vrai
 5. **Tant que** $(C(Sh_X, Sh^*) < C_{obj})$ et $(Amélioration = vrai)$ **faire**
 6. $l \leftarrow 1$
 7. Amélioration \leftarrow faux
 8. **Tant que** $((l < TSol)$ et $(Amélioration = faux))$ **faire**
 9. $e \leftarrow \sigma_1$
 10. **Tant que** $(e \leq \sigma_{|e|})$ et $(Amélioration = faux)$ **faire**
 11. $X' \leftarrow X$
 12. $X'(i) \leftarrow e$
 13. **Si** $(C(Sh_{X'}, Sh^*) > C(Sh_X, Sh^*))$ **alors**
 14. $X(i) \leftarrow e$
 15. Amélioration \leftarrow vrai
 16. **Fin si**
 17. $e \leftarrow e + 1$
 18. **Fin Tant que**
 19. **Fin tant que**
 20. $l \leftarrow l + 1$
 21. **Fin tant que**
 22. Retourner X
-

IV.6 Métaheuristiques

Dans cette partie, nous décrivons les métaheuristiques que nous avons utilisées pour résoudre le problème de génération de formes avec des automates cellulaires. Nous proposons d'utiliser un algorithme génétique, une recherche locale itérée et recuit simulé inhomogène.

IV.6.1 Algorithme génétique

Un algorithme génétique est mis en œuvre en générant d'abord une population initiale de $NIND$ chromosomes. Chaque chromosome représente une solution du problème à traiter. Les valeurs de la fonction de *fitness* (critère de similarité) sont calculées pour chacun des chromosomes.

Les opérateurs génétiques utilisés sont : la sélection par roulette, le croisement uni-point et la réinsertion élitiste.

IV.6.1.1 Sélection par roulette

A chaque chromosome X_l , un secteur dont l'angle est proportionnel à la valeur de son critère (fonction *fitness*) $C(Sh_{X_l}, Sh^*)$ est attribué. Un secteur est ensuite sélectionné suite à un tirage au sort. La probabilité de sélection d'un chromosome X_l est proportionnelle à la valeur de sa fonction *fitness*. Cette probabilité vaut :

$$P(X_l) = \frac{C(Sh_{X_l}, Sh^*)}{\sum_{v=1}^{NIND} C(Sh_{X_v}, Sh^*)} \tag{67}$$

IV.6.1.2 Croisement

Le croisement consiste à choisir deux chromosomes aléatoirement appelés « *parents* ». A partir de ces deux parents nous construisons deux autres chromosomes appelés « *enfants* ». Chacun de ces enfants hérite d'une partie de chacun de ses parents.

La Figure IV.12 présente un exemple d'un opérateur de croisement uni-point dans le cas d'un automate cellulaire 2D binaire et un système d'interaction de Von Neumann de rayon $R = 1$. Pour chaque chromosome (parent et enfant), la configuration de l'automate cellulaire avec $N_g = 7$ est donnée.



Figure IV.12 : Exemple d'un opérateur de croisement uni-point.

IV.6.1.3 Mutation

Chaque chromosome est muté avec une probabilité de mutation Pm . La mutation consiste à modifier un ou plusieurs éléments d'un chromosome.

Nous présentons dans la Figure IV.13 un exemple de mutation d'un élément du chromosome pour un automate cellulaire 2D binaire avec un système d'interaction de Von Neumann de rayon $R = 1$.

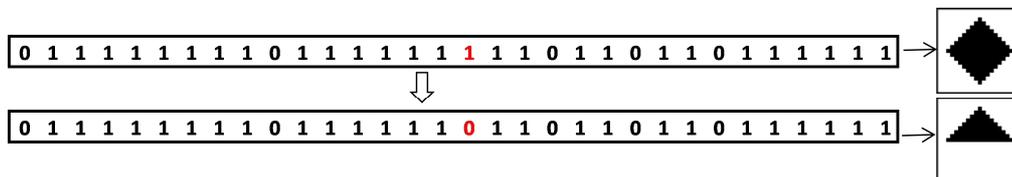


Figure IV.13 : Exemple de solution voisine en utilisant le système $\Gamma 1$.

IV.6.1.4 Remplacement

D'une itération à une autre de l'algorithme génétique, nous utilisons un mécanisme de remplacement élitiste. Il consiste à conserver les meilleurs individus d'une population à l'itération suivante de l'algorithme génétique.

Les paramètres de l'algorithme génétique sont présentés dans le Tableau IV.3. Les valeurs de ces paramètres ont été choisies après avoir réalisé plusieurs tests.

Nombre d'itérations N_{iter}	5000
Taille de la population N_{IND}	1000
Pourcentage des chromosomes sélectionnés	90%
Probabilité de croisement P_c	0.7
Probabilité de mutation P_m	0.3
Remplacement élitiste	20%

Tableau IV.3 : Paramètres de l'algorithme génétique

IV.6.2 Systèmes de voisinage

Un système de voisinage consiste à modifier un ou plusieurs éléments de la solution courante. Nous avons défini trois systèmes de voisinage qui dépendent du nombre d'éléments modifiés pour générer la solution voisine.

La taille η d'une solution dépend de l'approche utilisée :

- Dans l'approche (F, Ng), η est la somme de la taille de la fonction de transition N_c et le nombre de bits nécessaires utilisé pour coder le nombre de générations N_g .
- Dans les approches (Ng/F) et (F/Ng), η est donnée par la taille de la fonction de transition N_c .

IV.6.2.1 Système de voisinage Γ_1

L'Algorithme IV.6 donne le principe du système de voisinage Γ_1 . Un élément y de la solution est choisi uniformément et aléatoirement puis modifié. Dans ce système de voisinage, tous les éléments de la solution ont la même probabilité d'être sélectionnés. La taille de ce système de voisinage (nombre de solutions voisines possibles) est donnée par : $Card(\Gamma_1) = (|\sigma|-1) \cdot \eta$.

Algorithme IV.6 : Algorithme de génération d'une solution voisine avec le système Γ_1

1. Soit X une solution
 2. $X' \leftarrow X$
 3. Choisir aléatoirement et uniformément un élément y de la solution X
 4. Choisir aléatoirement et uniformément un état $e \in \sigma - \{X(y)\}$
 5. $X'(y) \leftarrow e$
-

Exemple

Dans le cas d'un automate cellulaire binaire, une modification d'un élément y de la solution X , pour obtenir une solution voisine X' , consiste à le remplacer par son complément : $X'(y) = 1 - X(y)$.

IV.6.2.2 Système de voisinage Γ_2

L'algorithme de principe de ce système de voisinage est donné dans l'Algorithme IV.7. Deux éléments différents sont choisis de manière aléatoire et uniforme puis modifiés. Le premier élément est choisi parmi η éléments et le deuxième parmi $(\eta-1)$ éléments. La taille de ce système de voisinage dans ce cas est donnée par :

$$Card(\Gamma_2) = (|\sigma| - 1)^2 \cdot \frac{\eta \cdot (\eta - 1)}{2} \quad (68)$$

Algorithme IV.7 : Algorithme de génération d'une solution voisine avec le système Γ_2

1. Soit X une solution
 2. $X' \leftarrow X$
 3. Choisir aléatoirement et uniformément deux éléments y et w de la solution X ($y \neq w$)
 4. Choisir aléatoirement et uniformément un état $e_1 \in \sigma\{-X(y)\}$
 5. $X'(y) \leftarrow e_1$
 6. Choisir aléatoirement et uniformément un état $e_2 \in \sigma\{-X(w)\}$
 7. $X'(w) \leftarrow e_2$
-

IV.6.2.3 Système de voisinage Γ_M

Contrairement aux deux systèmes de voisinage précédents, dans ce système de voisinage, le nombre de règles d'évolution locales à modifier ainsi que la taille du système de voisinage Γ_M varient d'une solution à une autre. Nous fixons M le nombre maximal d'éléments qui peuvent être modifiés. Ensuite, nous choisissons $m \in [1, M]$ comme étant le nombre d'éléments à modifier. La taille de ce système de voisinage est donnée par le nombre de combinaisons des règles d'évolution locales m sélectionnées parmi tous les éléments de la solution, elle est donnée par :

$$Card(\Gamma_M) = (|\sigma| - 1)^m \cdot \frac{\eta!}{(\eta - m)! m!} \quad (69)$$

Algorithme IV.8 : génération d'une solution voisine avec le système Γ_M

1. Soit X une solution
 2. $X' \leftarrow X$
 3. Choisir aléatoirement et uniformément $m \in [1, M]$
 4. Choisir aléatoirement m éléments différents de X à modifier
 5. **Pour** y de 1 à m **faire**
 6. Choisir aléatoirement et uniformément un état $e \in \sigma\{-X(y)\}$
 7. $X'(y) \leftarrow e$
 8. **Fin pour**
-

IV.6.2.4 Propriétés d'accessibilité et réversibilité

Les systèmes de voisinage proposés respectent les propriétés d'accessibilité et de réversibilité suivantes :

- *Accessibilité* : pour chaque paire de solutions (X, Y) , il existe une séquence finie d'états (μ_0, \dots, μ_l) avec $\mu_0 = X$ et $\mu_l = Y$ et $\mu_c \in \Gamma(\mu_{c-1})$, $c = 1, l$.
- *Réversibilité* : $Y \in \Gamma(X) \Leftrightarrow X \in \Gamma(Y)$, $\forall X, \forall Y$.

IV.6.3 Recherche locale itérée

Le principe de l'algorithme de la recherche locale itérée est donné par l'Algorithme IV.9. Deux systèmes de voisinage sont utilisés, le premier Γ (Γ_1 , Γ_2 ou Γ_M) pour la recherche locale et le deuxième Γ' (application de Γ 10 fois), généralement plus grand que Γ pour la perturbation. A est le maximum d'itérations sans amélioration, *best* correspond à la meilleure solution rencontrée, k est le nombre d'itérations à partir de la dernière amélioration.

Algorithme IV.9 : Algorithme de principe de la recherche locale itérée

```

1. Générer aléatoirement et uniformément une solution initiale X
2. Xbest ← X
3. i ← 1, k ← 0
4. Tant que i ≤ NIter faire
5.     Générer aléatoirement et uniformément X' ∈ Γ(X)
6.     Si C(ShX, Sh*) ≥ C(ShX, Sh*) alors
7.         X ← X'
8.         Si C(ShX, Sh*) ≠ C(ShX, Sh*) alors
9.             k ← 0
10.        Fin Si
11.    Fin Si
12.    k ← k+1
13.    Si k ≥ A alors
14.        Générer aléatoirement et uniformément X' ∈ Γ'(X)
14.        Si C(ShX, Sh*) ≠ C(ShX, Sh*) alors
15.            X ← X'
16.            k ← 0
17.        Fin Si
18.    Fin Si
19.    Si C(ShX, Sh*) > C(Best, Sh*) alors
20.        Xbest ← X
21.    Fin Si
22.    i ← i+1
23. Fin tant que
24. Retourner la solution record Xbest

```

Les paramètres de la recherche locale itérée concernent le nombre d'itérations qui est fixé à 5 millions quelle que soit l'approche utilisée et la taille du palier A qui dépend du système de voisinage utilisé. Avec le système de voisinage Γ_1 , la taille du palier A respecte la condition de convergence $A = 0.7Card(\Gamma)$ donnée dans (Fleury, 1993). Dans le cas de Γ_2 et Γ_3 , en respectant la condition de Fleury, la taille du palier pour Γ_2 et Γ_3 est très importante. Cependant, la taille du palier A est fixée à 10000 pour Γ_2 et 50000 pour Γ_3 .

IV.6.4 Recuit simulé

Dans ce travail, nous utilisons un recuit non homogène qui ne marque pas de paliers à chaque baisse de la température. La température décroît à chaque itération du recuit simulé. Les différentes étapes d'évolution de cette méthode sont présentées dans l'Algorithme IV.10. L'efficacité du recuit simulé dépend de ses paramètres qui sont :

- La température initiale du recuit simulé T_0 .
- La température d'arrêt du recuit simulé T_a .

Le facteur de décroissance de température T_f est donné par: $T_f = \sqrt[N_{Iter}]{T_a / T_0}$ où N_{Iter} est le nombre d'itérations du recuit simulé.

Le choix des paramètres du recuit simulé est primordial pour sa convergence vers les optima globaux. Comme dans notre cas, le recuit simulé utilisé est inhomogène, la température d'arrêt dépend uniquement de la température initiale et du facteur de décroissance de la température qui à son tour dépend du nombre d'itérations du recuit. Le nombre d'itérations a été fixé à 5 millions pour toutes les méthodes à voisinage quelle que soit l'approche, il ne reste qu'à déterminer la température initiale du recuit simulé.

Algorithme IV.10: Algorithme de principe du recuit simulé non homogène

1. Choisir aléatoirement une solution initiale X
2. $T \leftarrow T_0$
3. $X_{\text{best}} \leftarrow X$
4. **Tant que** $T > T_a$ **faire**
5. Choisir aléatoirement une solution voisine $X' \in \Gamma(X)$
6. **Si** $C(\text{Sh}_{X'}, \text{Sh}^*) \geq C(\text{Sh}_X, \text{Sh}^*)$ **alors**
7. $X \leftarrow X'$
8. **Sinon**
9. Choisir aléatoirement $\alpha \in [0, 1]$
10. **Si** $\alpha \leq \exp [-(C(\text{Sh}_X, \text{Sh}^*) - C(\text{Sh}_{X'}, \text{Sh}^*))/T]$ **alors**
11. $X \leftarrow X'$
12. **Fin si**
13. **Fin si**
14. **Si** $C(\text{Sh}_X, \text{Sh}^*) > C(\text{Sh}_{X_{\text{Best}}}, \text{Sh}^*)$ **alors**
14. $X_{\text{best}} \leftarrow X$
15. **Fin si**
16. $T \leftarrow T * T_f$
17. **Fin tant que**
18. Retourner la solution record X_{best}

Pour choisir la température initiale, nous avons pris le plan de réglage décrit dans (Siarry, 1995). Le principe (Algorithme IV.1) est de choisir une solution initiale, puis de réaliser 100 perturbations de cette solution et calculer la moyenne $\overline{\Delta C}$ des variations ΔC correspondantes. Choisir un taux initial δ_0 d'acceptation des solutions dégradantes selon la qualité de la solution initiale. Lorsque la solution est de très mauvaise qualité, on peut prendre un taux d'acceptation élevé (exemple : $\delta_0 = 0.8$). Si la solution initiale est de bonne qualité, le taux d'acceptation de solutions dégradantes est faible (exemple $\delta_0 = 0.1$). La température initiale peut être déduite par la relation suivante :

$$\delta_0 = \exp\left(-\frac{\overline{\Delta C}}{T_0}\right) \Rightarrow T_0 = \frac{\overline{\Delta C}}{\ln(\delta_0^{-1})} \quad (70)$$

Algorithme IV.11 : Algorithme 1 de calcul de la température du recuit simulé

1. Générer une solution initiale X aléatoirement
2. Choisir δ_0 en fonction de la qualité de X
3. $\Delta C \leftarrow 0$
4. **Pour** i de 1 à m **faire**
5. Générer aléatoirement et uniformément une solution voisine $X' \in \Gamma(X)$
6. $\Delta C \leftarrow \Delta C + (C(\text{Sh}_X, \text{Sh}^*) - C(\text{Sh}_{X'}, \text{Sh}^*))$
7. **Fin pour**
8. $\overline{\Delta C} \leftarrow \Delta C / m$
9. $T_0 = \frac{\overline{\Delta C}}{\ln(\delta_0^{-1})}$

Dans le deuxième cas (Algorithme IV.12), nous réalisons les mêmes étapes que précédemment mais on prend en considération uniquement les perturbations dégradantes ΔC^- pour le calcul de $\overline{\Delta C}$. Le calcul de la température initiale est donné par :

$$T_0 = \frac{\overline{\Delta C^-}}{\ln(\delta_0^{-1})} \quad (71)$$

Algorithme IV.12 : Algorithme 2 de calcul de la température du recuit simulé

1. Générer une solution initiale X aléatoirement
 2. Choisir δ_0 en fonction de la qualité de X
 3. $\Delta C^- \leftarrow 0$; $k \leftarrow 0$
 4. **Pour** i de 1 à m **faire**
 5. Générer aléatoirement et uniformément une solution voisine $X' \in \Gamma(X)$
 6. **Si** $(C(\text{Sh}_X, \text{Sh}^*) - C(\text{Sh}_{X'}, \text{Sh}^*)) < 0$ **alors**
 7. $\Delta C^- \leftarrow \Delta C^- + (C(\text{Sh}_X, \text{Sh}^*) - C(\text{Sh}_{X'}, \text{Sh}^*))$
 8. $k \leftarrow k+1$
 9. **Fin si**
 10. **Fin pour**
 11. $\overline{\Delta C^-} \leftarrow \Delta C^- / k$
 12. $T_0 = \frac{\overline{\Delta C^-}}{\ln(\delta_0^{-1})}$
-

En général, la solution initiale est de très mauvaise qualité et le taux d'acceptation de solutions dégradantes est élevé.

IV.7 Hybridation des méthodes

L'efficacité des méthodes d'optimisation que nous proposons dépend de la performance de la solution de départ qui est choisie aléatoirement et qui peut être de très mauvaise qualité. Afin d'améliorer nos algorithmes et d'éviter d'explorer l'espace de recherche à partir d'une solution de très mauvaise qualité, nous proposons dans cette partie, une hybridation d'heuristiques et de métaheuristiques.

Le type d'hybridation choisi est celui de coopération relais de haut niveau. Il consiste à exécuter deux méthodes ou plusieurs en séquentielle pour exploiter les performances de chacune d'elle pour améliorer les résultats.

Les hybridations proposées (voir Figure IV.14) sont : entre heuristiques et entre heuristiques et métaheuristiques.

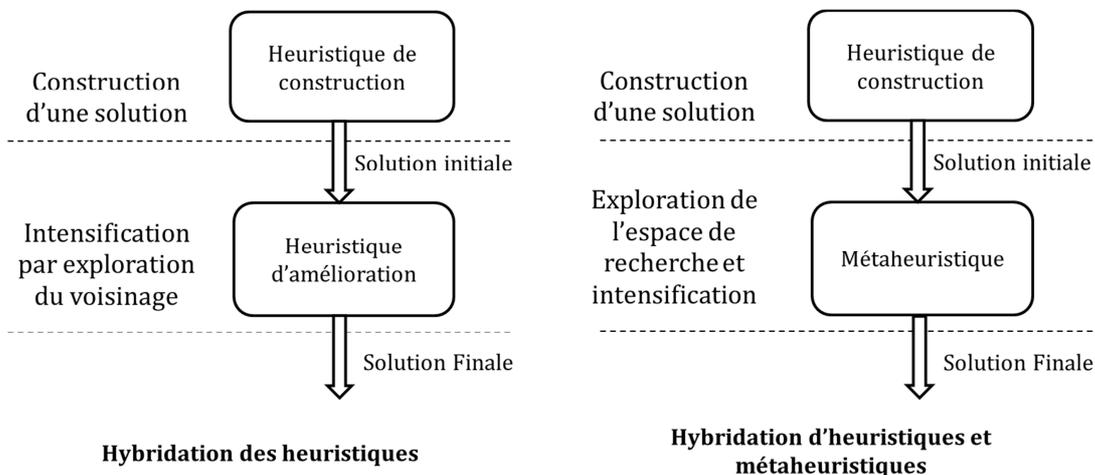


Figure IV.14 : Hybridation de méthodes.

L'hybridation entre les heuristiques se fait entre une heuristique de construction (algorithme glouton) et une heuristique d'amélioration (recherche locale déterministe avec le meilleur voisin ou recherche locale déterministe avec le premier voisin). L'heuristique de construction génère une solution de départ pour l'heuristique d'amélioration.

IV.8 Cas particuliers des formes symétriques

La symétrie vient du mot grec « *Symmetria* » pour désigner une situation d'harmonie dans les proportions. Puis la définition s'est étendue à des notions plus formelles : mise en coïncidence d'un objet avec lui-même (ou un autre), par déplacement (translation ou rotation) ou par réflexion dans un miroir. Puis apparaît la notion d'invariance par symétrie et la recherche d'ordre dans la diversité.

Dans le domaine des automates cellulaires, quelques travaux exploitent la symétrie. (Mainzer, 2005) s'est intéressé aux différentes symétries dans le diagramme spatio-temporel des automates cellulaires 1D mais il ne traite pas le problème inverse des automates cellulaires. (Sapin, et al., 2003) a exploité la symétrie des automates cellulaires pour la recherche d'automates cellulaires universels permettant de générer des « *planeurs* » qui se déplacent dans toutes les directions de l'espace de l'automate cellulaire au cours du temps. Dans ces travaux, la symétrie horizontale, verticale et rotationnelle (Figure IV.15) est exploitée. (Batoche, et al., 2006) ont également exploité le même type de symétrie pour le problème de segmentation d'image (détection des contours des objets). Pour le cas de génération de formes par automates cellulaires, il nous a semblé intéressant d'exploiter les différents types de symétrie.

Nous avons identifié les différents types de symétrie (voir Figure IV.15) :

- Cas (a): symétrie verticale,
- Cas (b): symétrie horizontale,
- Cas (c): symétrie horizontale et verticale,
- Cas (d): symétrie rotationnelle de 90° ,
- Cas (e): symétrie rotationnelle de 180°
- Cas (f): symétrie axiale (horizontale et verticale) et rotationnelle.

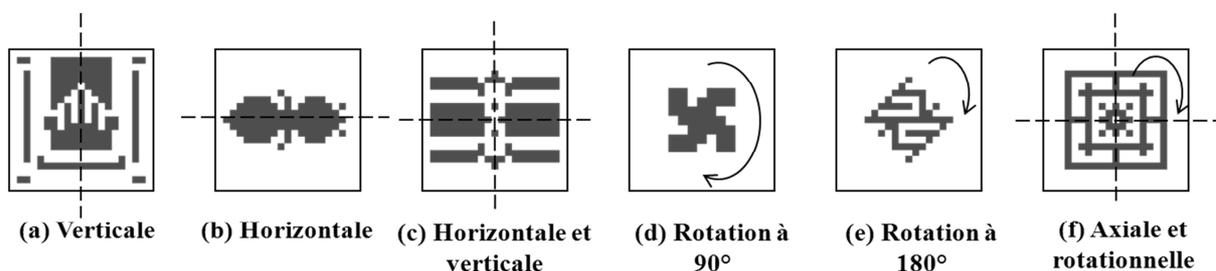


Figure IV.15. Types de symétrie.

Les formes symétriques, horizontalement et verticalement, sont également symétriques par rotation de 180° . Les formes symétriques par une rotation de 90° sont également symétriques par rotation de 180° .

La symétrie dans un automate cellulaire peut se traduire par l'invariance d'une règle d'évolution locale par transformation spatiale T du système d'interaction.

Pour exploiter cette symétrie, les combinaisons du système d'interaction qui sont symétriques à la génération t doivent avoir le même état à la génération $t+1$. A cet effet, une seule règle d'évolution locale est appliquée à un groupe de combinaisons symétriques.

$$\forall c \in L, \quad f \left(E_t (N(c)) \right) = f \left(E_t \left((T(N(c))) \right) \right) \quad (72)$$

La fonction de transition de l'automate cellulaire est l'ensemble de ces règles d'évolution locales. Par conséquent, la taille de la fonction de transition ainsi que celle de l'espace de recherche dépendent non seulement du nombre d'états et de la taille du système d'interaction mais aussi du type de symétrie de la forme recherchée.

Afin de réduire l'espace de recherche et augmenter la vitesse de convergence des métaheuristiques, nous proposons un codage de la fonction de transition de l'automate cellulaire qui tient compte du type de symétrie présent dans la forme désirée (Aiboud, et al., 2013a ; Aiboud, et al., 2013b).

Dans les parties suivantes, nous présentons le codage exploitant la symétrie horizontale et le cas de la symétrie axiale et rotationnelle (horizontale, verticale et rotationnelle). Les différents codages de la fonction de transition de l'automate cellulaire en fonction des autres types de symétrie sont présentés dans l'annexe B.

IV.8.1 Symétrie horizontale

La Figure IV.16 montre les combinaisons symétriques horizontalement dans le cas du système d'interaction de Von Neumann et de Moore.

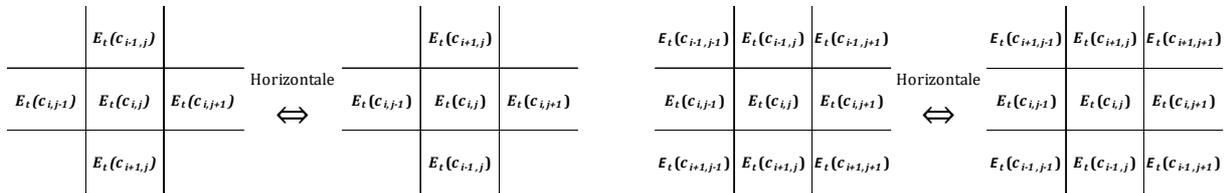


Figure IV.16. Symétrie horizontale avec Von Neumann (à gauche) et Moore (à droite).

La transformation T_H qui définit les combinaisons symétriques horizontalement est donnée dans le cas de Von Neumann par :

$$T_H : \sigma^n \rightarrow \sigma^n \quad (73)$$

$$E_t(c_{i,j}, c_{i-1,j}, c_{i,j-1}, c_{i+1,j}, c_{i,j+1}) \mapsto E_t(c_{i,j}, c_{i+1,j}, c_{i,j-1}, c_{i-1,j}, c_{i,j+1})$$

Avec le système d'interaction de Moore la transformation d'équivalence des combinaisons du système d'interaction est donnée par :

$$T_H \left(E_t(c_{i-1,j-1}, c_{i-1,j}, c_{i-1,j+1}, c_{i,j-1}, c_{i,j}, c_{i,j+1}, c_{i+1,j-1}, c_{i+1,j}, c_{i+1,j+1}) \right) \quad (74)$$

$$= \left(E_t(c_{i+1,j-1}, c_{i+1,j}, c_{i+1,j+1}, c_{i,j-1}, c_{i,j}, c_{i,j+1}, c_{i-1,j-1}, c_{i-1,j}, c_{i-1,j+1}) \right)$$

La Figure IV.17 montre les combinaisons équivalentes par symétrie horizontale. La Figure IV.18 montre un exemple de simplification de la fonction de transition de l'automate cellulaire avec le système d'interaction de Von Neumann par symétrie horizontale.

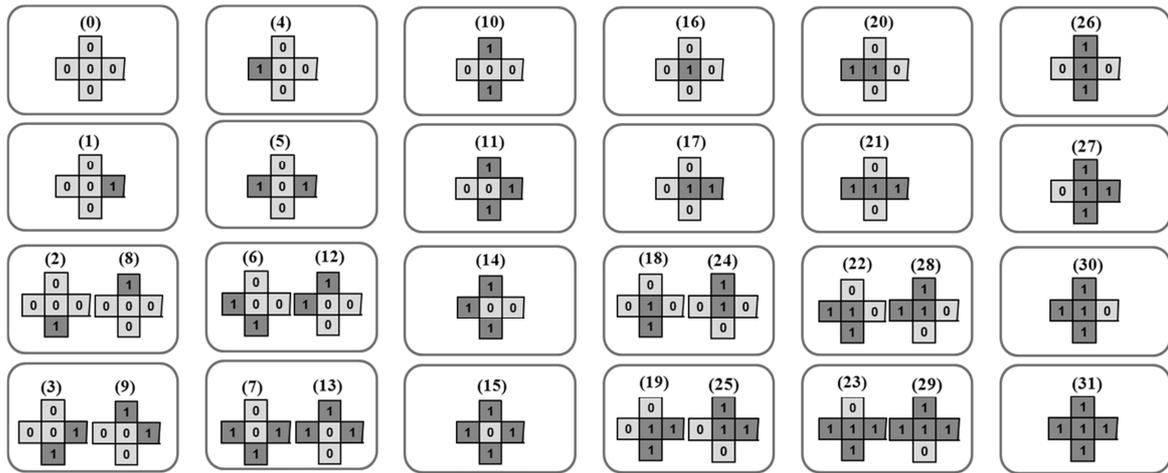


Figure IV.17 : Combinaisons d'état du système d'interaction équivalentes par symétrie horizontale.

		(0)	(1)	(2)	(8)	(3)	(9)	(4)	(5)	(6)	(12)	(7)	(13)	(10)	(11)	(14)	(15)	(16)	(17)	(18)	(24)	(19)	(25)	(20)	(21)	(22)	(28)	(23)	(29)	(26)	(27)	(30)	(31)
Etat du système d'interaction	$E_t(c_{ij})$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	$E_t(c_{i+1,j})$	0	0	0	1	0	1	0	0	0	1	0	1	1	1	1	1	0	0	0	0	1	0	1	0	0	0	1	0	1	1	1	1
	$E_t(c_{i,j+1})$	0	0	0	0	0	0	1	1	1	1	1	0	0	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	1	1
	$E_t(c_{i+1,j+1})$	0	0	1	0	1	0	0	0	1	0	1	0	1	1	1	0	0	1	0	0	1	0	0	0	1	0	1	0	1	0	1	1
	$E_t(c_{i,j})$	0	1	0	0	1	1	0	1	0	0	1	1	0	1	0	1	0	1	0	0	1	1	0	1	0	0	1	1	0	1	0	1
↓																																	
Nouvelle état de $c_{i,j}$	$E_{i+1}(c_{ij})$	0	1	1	1	1	1	1	1	1	1	1	0	0	0	1	1	0	1	1	0	1	1	0	1	0	1	0	0	0	0	0	

Figure IV.18. Fonction de transition simplifiée par symétrie horizontale (cas du système d'interaction de Von Neumann).

Symétrie axiale et rotationnelle

Dans ce cas, la forme est compose de quatre parties identiques et symétriques par rotation (à 90° et à 180°). Les conditions de symétries axiale et rotationnelle doivent être simultanément satisfaites.

La Figure IV.19 présente les combinaisons équivalentes par symétrie axiale et rotationnelle dans le cas du système d'interaction de Von Neumann.

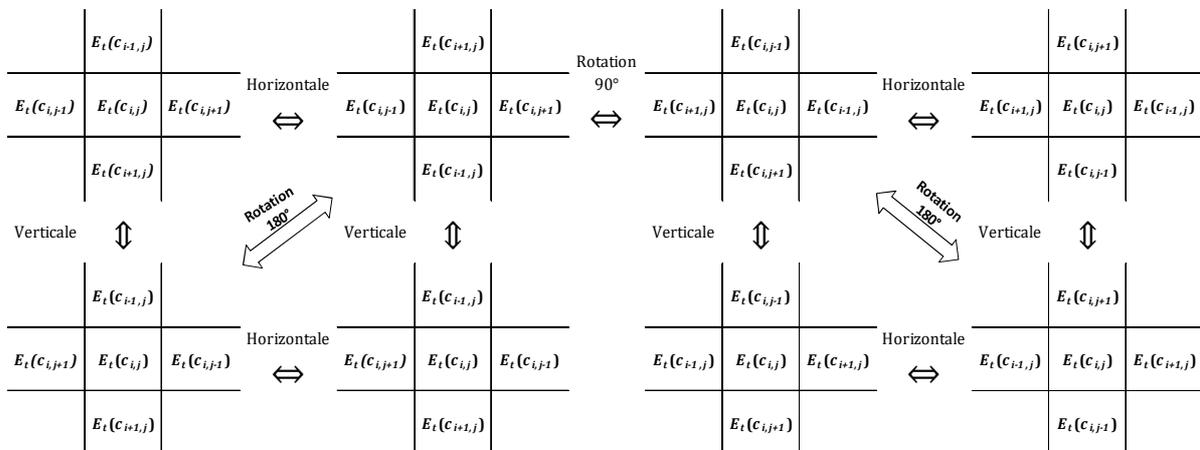


Figure IV.19. Symétrie axiale et rotationnelle avec le système d'interaction de Von Neumann.

Les combinaisons équivalentes par symétrie axiale et rotationnelle sont données par :

$$\begin{aligned}
 f(E_t(c_{i,j}, c_{i-1,j}, c_{i,j-1}, c_{i+1,j}, c_{i,j+1})) &= f(E_t(c_{i,j}, c_{i+1,j}, c_{i,j-1}, c_{i-1,j}, c_{i,j+1})) \\
 &= f(E_t(c_{i,j}, c_{i-1,j}, c_{i,j+1}, c_{i+1,j}, c_{i,j-1})) = f(E_t(c_{i,j}, c_{i+1,j}, c_{i,j-1}, c_{i-1,j}, c_{i,j-1})) \\
 &= f(E_t(c_{i,j}, c_{i,j-1}, c_{i+1,j}, c_{i,j+1}, c_{i-1,j})) = f(E_t(c_{i,j}, c_{i,j-1}, c_{i-1,j}, c_{i,j+1}, c_{i+1,j})) \\
 &= f(E_t(c_{i,j}, c_{i,j+1}, c_{i+1,j}, c_{i,j-1}, c_{i-1,j})) = f(E_t(c_{i,j}, c_{i,j+1}, c_{i-1,j}, c_{i,j-1}, c_{i+1,j}))
 \end{aligned}
 \tag{75}$$

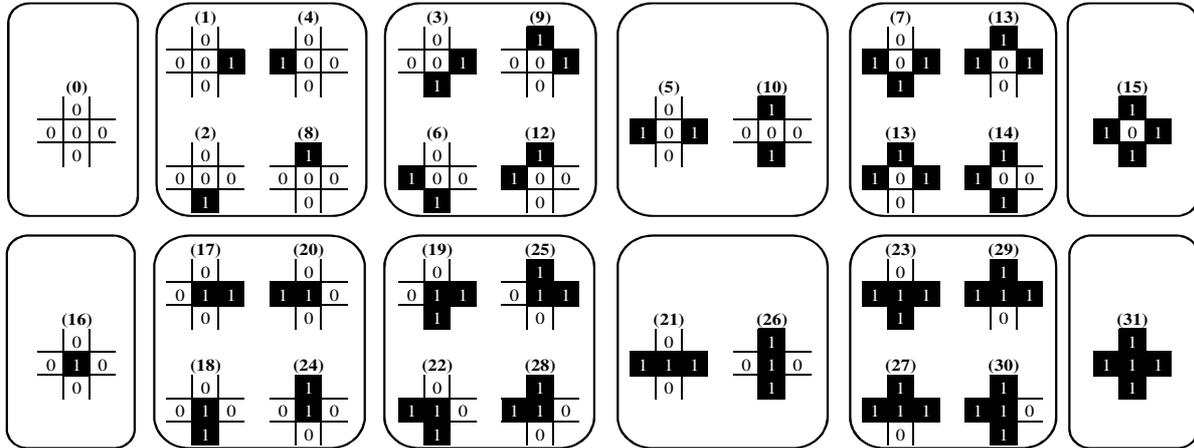


Figure IV.20 : Combinaisons d'états du système d'interaction équivalentes par symétrie axiale et rotationnelle.

Les combinaisons équivalentes par symétrie axiale et rotationnelle sont données dans le cas du système d'interaction de Moore dans la Figure 50.

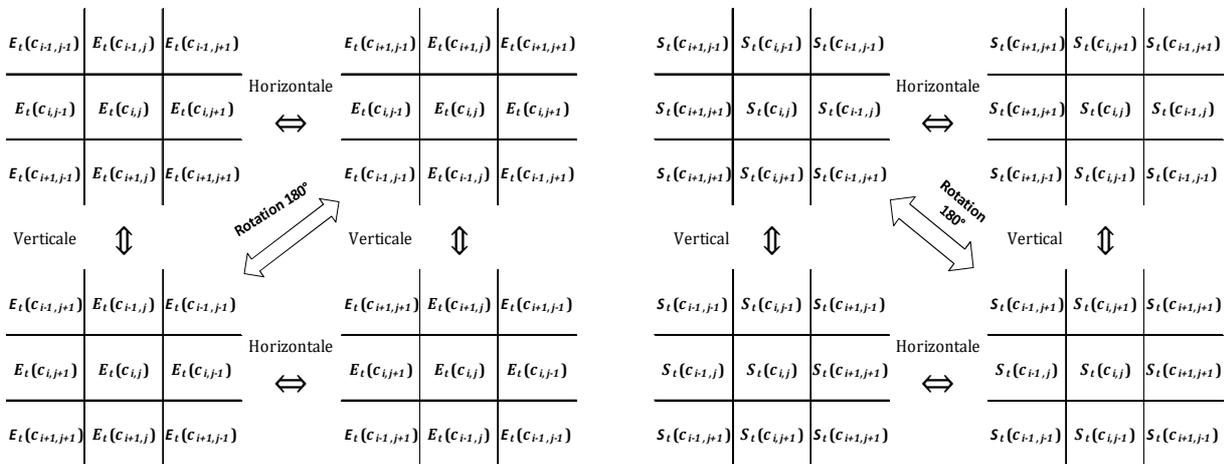


Figure IV.21 : Symétrie axiale et rotationnelle avec le système d'interaction de Moore.

La transformation utilisée avec le système d'interaction de Moore avec une symétrie axiale et rotationnelle est :

$$\begin{aligned}
 & f\left(E_t\left(c_{i-1,j-1}, c_{i-1,j}, c_{i-1,j+1}, c_{i,j-1}, c_{i,j}, c_{i,j+1}, c_{i+1,j-1}, c_{i+1,j}, c_{i+1,j+1}\right)\right) \\
 &= f\left(E_t\left(c_{i+1,j-1}, c_{i+1,j}, c_{i+1,j+1}, c_{i,j-1}, c_{i,j}, c_{i,j+1}, c_{i-1,j-1}, c_{i-1,j}, c_{i-1,j+1}\right)\right) \\
 &= f\left(E_t\left(c_{i-1,j+1}, c_{i-1,j}, c_{i-1,j-1}, c_{i,j+1}, c_{i,j}, c_{i,j-1}, c_{i+1,j+1}, c_{i+1,j}, c_{i+1,j-1}\right)\right) \\
 &= f\left(E_t\left(c_{i+1,j+1}, c_{i+1,j}, c_{i+1,j-1}, c_{i,j+1}, c_{i,j}, c_{i,j-1}, c_{i-1,j+1}, c_{i-1,j}, c_{i-1,j-1}\right)\right) \\
 &= f\left(E_t\left(c_{i-1,j-1}, c_{i,j-1}, c_{i+1,j-1}, c_{i-1,j}, c_{i,j}, c_{i+1,j}, c_{i-1,j+1}, c_{i,j+1}, c_{i+1,j+1}\right)\right) \\
 &= f\left(E_t\left(c_{i+1,j-1}, c_{i,j-1}, c_{i-1,j-1}, c_{i+1,j}, c_{i,j}, c_{i-1,j}, c_{i+1,j+1}, c_{i,j+1}, c_{i-1,j+1}\right)\right) \\
 &= f\left(E_t\left(c_{i-1,j+1}, c_{i,j+1}, c_{i+1,j+1}, c_{i-1,j}, c_{i,j}, c_{i+1,j}, c_{i-1,j-1}, c_{i,j-1}, c_{i+1,j-1}\right)\right) \\
 &= f\left(E_t\left(c_{i+1,j+1}, c_{i,j+1}, c_{i-1,j+1}, c_{i+1,j}, c_{i,j}, c_{i-1,j}, c_{i+1,j-1}, c_{i,j-1}, c_{i-1,j-1}\right)\right)
 \end{aligned} \tag{76}$$

En utilisant le système d'interaction de Von Neumann, la symétrie axiale et rotationnelle est équivalente à une simple rotation à 90°.

		(0)	(1)	(2)	(4)	(8)	(3)	(6)	(9)	(12)	(5)	(10)	(7)	(11)	(13)	(14)	(15)	(16)	(17)	(18)	(20)	(24)	(19)	(22)	(25)	(28)	(21)	(26)	(23)	(27)	(29)	(30)	(31)
Etat du système d'interaction	$E_t(c_{ij})$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	$E_t(c_{i+1,j})$	0	0	0	0	1	0	0	1	1	0	1	0	1	1	1	1	0	0	0	0	1	0	0	1	1	0	1	0	1	1	1	1
	$E_t(c_{i,j+1})$	0	0	0	1	0	0	1	0	1	1	0	1	0	1	1	1	0	0	0	1	0	0	1	0	1	1	0	1	1	1	1	1
	$E_t(c_{i+1,j+1})$	0	0	1	0	0	1	1	0	0	0	1	1	1	0	1	1	0	0	1	0	0	1	1	0	0	0	1	1	1	0	1	1
	$E_t(c_{ij+1})$	0	1	0	0	0	1	0	1	0	1	0	1	1	1	0	1	0	1	0	0	0	1	0	1	0	1	0	1	1	1	0	1

↓

Nouvelle état de c_{ij}	$E_{i+1}(c_{ij})$	0	1	0	1	0	0	1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0		
		0	1	0	1	0	0	1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0

Figure IV.22 : Exemple de fonction de transition simplifiée par symétrie axiale et rotationnelle avec le système d'interaction de Von Neumann.

IV.8.2 Espace de recherche

La taille des fonctions de transition et celle de l'espace de recherche dépendent non seulement du nombre d'états et de la taille du système d'interaction, mais aussi du type de symétrie dans la forme recherchée. Les facteurs de réduction de la taille de l'espace de recherche sont également donnés. Chaque facteur de réduction est le rapport entre la taille de l'espace de recherche sans prise en compte de la symétrie et la taille de l'espace de recherche avec prise en compte de la symétrie.

Nous présentons dans le Tableau IV.4 la taille des fonctions de transition et la taille de l'espace de recherche pour un automate cellulaire binaire. La taille de l'espace des fonctions de transition est largement réduite par exploitation de la symétrie. Dans le cas du système d'interaction de Von Neumann, la taille de l'espace de recherche est largement réduite. Ceci permet d'énumérer toutes les fonctions de transition de l'automate cellulaire.

Le Tableau IV.5 donne la taille des fonctions de transition et de l'espace de recherche en fonction du type de symétrie pour des automates cellulaires à 3 états. Les facteurs de réduction de l'espace de recherche sont donnés pour chaque type de symétrie et système d'interaction.

La taille de l'espace de recherche avec le système d'interaction de Moore reste très importante malgré l'importante réduction de l'espace de recherche.

	Von Neumann			Moore		
	Taille de la fonction de transition	Taille de l'espace de recherche	Facteur de réduction	Taille de la fonction de transition	Taille de l'espace de recherche	Facteur de réduction
Sans symétrie	32	4.29E+09	-	512	1.34E+154	-
Symétrie horizontale	24	1.68E+07	2.56E+02	288	4.97E+86	2.70E+67
Symétrie verticale	24	1.68E+07	2.56E+02	288	4.97E+86	2.70E+67
Symétrie horizontale et verticale	18	2.62E+05	1.64E+04	168	3.74E+50	3.58E+103
Rotation à 90°	12	4.10E+03	1.05E+06	140	1.39E+42	9.62E+111
Rotation à 180°	20	1.05E+06	4.10E+03	272	7.59E+81	1.77E+72
Symétrie axiale et rotationnelle	12	4.10E+03	1.05E+06	102	5.07E+30	2.64E+123

Tableau IV.4 : Tailles et nombres de fonctions de transition en fonction de la symétrie pour un automate cellulaire binaire.

	Von Neumann			Moore		
	Taille de la fonction de transition	Taille de l'espace de recherche	Facteur de réduction	Taille de la fonction de transition	Taille de l'espace de recherche	Facteur de réduction
Sans symétrie	243	8.7E+115	-	19683	15E+9390	-
Symétrie horizontale	162	2.0E+77	4.43E+38	10205	10E+4868	15E+4524
Symétrie verticale	162	2.0E+77	4.43E+38	10205	10E+4868	10E+4524
Symétrie horizontale et verticale	108	3.4E+51	2.58E+64	5346	49E+2549	10E+6841
Rotation à 90°	63	1.1E+30	7.62E+85	4995	16E+2384	31E+7008
Rotation à 180°	135	2.6E+64	3.38E+51	9963	36E+4752	16E+4640
Symétrie axiale et rotationnelle	63	1.1E+30	7.62E+85	2862	33E+1364	36E+8028

Tableau IV.5: Tailles et nombres de fonctions de transition en fonction de la symétrie pour un automate cellulaire à 3 états.

Conclusion

Dans ce chapitre, nous avons présenté des couplages d'automates cellulaires et des méthodes d'optimisation pour le problème inverse des automates cellulaires. Une formalisation mathématique de ce problème d'optimisation a été proposée. Comme ce problème non linéaire est de taille très importante, des méthodes exactes ne peuvent pas être utilisées pour le résoudre.

Les méthodes d'optimisation à base d'heuristiques (algorithme glouton et recherche locale déterministe) et de métaheuristiques (algorithme glouton et recherche locale déterministe) ont été proposées. Elles ont pour rôle de déterminer la fonction de transition F de l'automate cellulaire et ainsi que son nombre de générations Ng permettant d'obtenir une forme très proche de la forme recherchée.

Des critères de similarité ont été proposés pour la comparaison des formes. Ces critères mesurent soit la distance entre les deux formes (nombre de cellules identiques, écarts quadratiques ou moments de Legendre) soit l'information apportée mutuellement par les formes à comparer (information mutuelle).

Deux codages de solution ont été proposés. Ces codages ont donné trois approches : dans la première approche (F, Ng), la fonction de transition F de l'automate ainsi que son nombre de générations Ng sont codés dans la solution. Dans les deux autres approches, seule la fonction de transition est codée dans la solution. Dans l'approche (Ng/F), le nombre de générations Ng est

calculé pour chaque fonction de transition F . Dans l'approche (F/Ng), pour chaque nombre de générations Ng , la meilleure fonction de transition F est déterminée.

Pour réduire la taille de l'espace de recherche, nous avons proposé un codage qui exploite les propriétés de la forme recherchée. Ce codage est appliqué aux cas des formes symétriques.

Chapitre V. MODELISATION MICROSCOPIQUE :

MISE EN ŒUVRE ET RESULTATS

Introduction

Dans ce chapitre, nous présentons les résultats de la mise en œuvre des méthodes proposées pour la résolution du problème inverse des automates cellulaires. Ces méthodes sont à base d'heuristiques (algorithme glouton et recherches locales déterministes) et de métaheuristiques (algorithme génétique, recherche locale itérée et recuit simulé). Pour chacune de ces méthodes, les différentes approches (approche (F, Ng), approche (Ng/F) et approche (F/Ng)) ont été considérées. Les critères de similarité (nombre de cellules identiques, information mutuelle, écart quadratique, extension du critère de De Garis et moments de Legendre) seront étudiés. Les méthodes à voisinage utilisent trois systèmes de voisinage (Γ_1 , Γ_2 et Γ_M).

Ces méthodes ont été testées dans un premier temps sur des formes pleines et quelconques de la littérature, puis sur des formes générées aléatoirement formant une bibliothèque d'instances. Les résultats obtenus sur ces formes sont donnés. Une utilisation de cette méthode pour un problème de segmentation d'images est présentée.

Ce chapitre est réparti en plusieurs parties :

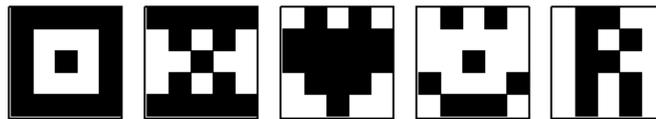
- Dans la première partie de ce chapitre, la bibliothèque des instances est donnée. Ces formes sont soit quelconques soit symétriques. Des formes 2D et 3D sont créées avec deux systèmes d'interaction (Von Neumann ou Moore de rayon $R = 1$). Ces formes sont soit binaires (à deux états), soit à trois états.
- Dans la deuxième partie, dans le but de sélectionner parmi l'ensemble des critères proposés ceux qui sont les plus efficaces, des comparaisons des critères de similarité sont données. Ces comparaisons permettent de retenir deux critères (information mutuelle et nombre de cellules identiques) pour tous les tests qui seront effectués.
- Dans la troisième partie, une justification de l'utilisation des métaheuristiques en se basant sur l'espace de la fonction objectif est présentée.
- Dans la quatrième partie, une comparaison des méthodes proposées sur des formes pleines (Chavoya & Duthen, 2006) et des formes quelconques (Breukelaar & Back, 2005) est donnée.
- Les résultats obtenus avec les différentes méthodes et différents types de formes sont présentés dans les quatre parties suivantes : une partie est consacrée pour des formes binaires quelconques, une deuxième partie pour des formes binaires 2D symétriques, une autre partie, pour des formes 3D symétriques et une dernière partie pour des formes 2D à trois états.
- La neuvième et dernière partie est consacrée à l'application des méthodes proposées pour résoudre d'autres problèmes (cas de la segmentation des contours d'images par automate cellulaire).

V.1 Bibliothèque d'instances

Les méthodes proposées ont été appliquées pour rechercher des formes de la littérature : formes pleines (Figure V.1-(a)) proposées par (Chavoya & Duthen, 2006) et formes creuses (Figure V.1-(b)) proposées par (Breukelaar & Back, 2005).



(a) Formes proposées par Chavoya et Duthen.



(a) Formes proposées par (Breukelaar & Back, 2005).

Figure V.1: Formes de la littérature.

Un ensemble de formes a été créé pour tester les méthodes proposées. Ces formes diffèrent les unes des autres par la dimension de l'automate cellulaire (2D ou 3D), le nombre d'états (2 ou 3 états), le système d'interaction (Von Neumann ou Moore) et leurs propriétés (quelconques ou symétriques).

Toutes ces formes sont de même tailles $L = 21$ dans chaque dimension (21×21 pour les formes 2D et $21 \times 21 \times 21$ pour les formes 3D).

Comme dans la plupart des travaux de la littérature (Chavoya & Duthen, 2006; Breukelaar & Back, 2005), nous considérons une forme initiale composée d'une cellule à l'état « 1 » placée au centre de la grille de l'automate cellulaire.

V.1.1 Formes 2D binaires quelconques

Différentes formes binaires 2D quelconques ont été générées (Figure V.2). Ces formes sont obtenues par la génération de fonctions de transition de façon aléatoire. La fonction de transition utilisée pour générer chacune de ces formes correspond soit à un système d'interaction de Von Neumann (formes appelées VN1 à VN10) ou bien au système d'interaction de Moore (formes appelées M1 à M10). Le nombre de générations utilisé pour obtenir une forme donnée est également choisi aléatoirement. Ce nombre de générations est choisi de telle sorte que les bords de l'automate cellulaire ne soient pas atteints.

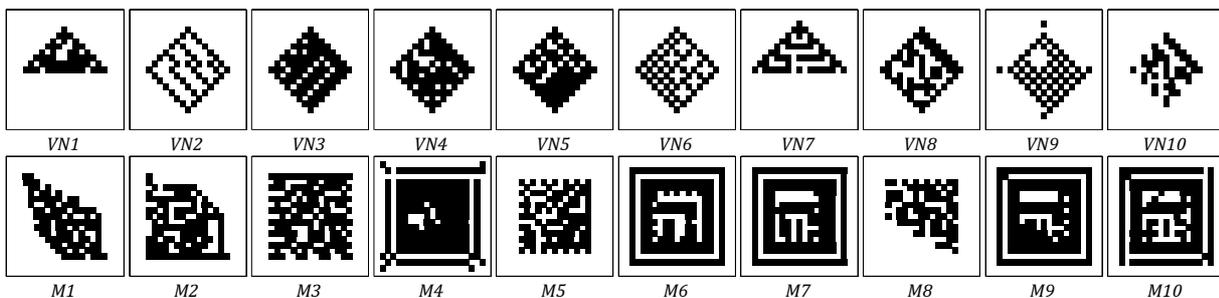


Figure V.2 : Formes 2D binaire quelconques.

Le principe est décrit dans l'Algorithme V.1. Sh^{Ng} constitue une forme de la bibliothèque.

Algorithme V.1 : Principe de génération d'une forme

1. Choisir une fonction de transition F aléatoirement
 2. Choisir un nombre de générations $Ng \leq Ngmax$
 3. Sh^0 une forme initiale
 4. **Pour** t de 1 à Ng **faire**
 5. Appliquer la fonction de transition F à la forme Sh^{t-1}
 6. **Fin pour**
-

V.1.2 Formes 2D binaires symétriques

Nous avons testé notre algorithme sur des formes symétriques. Nous avons généré cinq formes pour chaque type de symétrie en utilisant les deux systèmes d'interaction Von Neumann et Moore. Le nom de chaque forme est composé du nom du système d'interaction pour la générer (VN pour Von Neumann et M pour Moore) suivi du type de symétrie identifié dans la forme désirée et enfin le numéro de la forme. Comme nous l'avons précisé dans le chapitre précédent, dans le cas du système d'interaction de Von Neumann, la symétrie horizontale, verticale et rotationnelle est équivalente à la symétrie par rotation à 90° . Par conséquent, nous ne générons pas de formes pour la symétrie par rotation à 90° avec le système d'interaction de Von Neumann.

Ces formes ont été générées en utilisant l'Algorithme V.2. Au départ, une fonction de transition F_s qui prend en compte un type de symétrie est générée aléatoirement. Chaque élément de cette fonction de transition est une règle d'évolution locale correspondant à un groupe de combinaisons d'états symétriques du système d'interaction. Comme cette fonction de transition ne peut pas s'appliquer directement à l'automate cellulaire, une étape d'extraction de la fonction de transition F à appliquer à l'automate cellulaire est nécessaire. Le nombre de génération Ng est généré aléatoirement.

Algorithme V.2 : Principe de génération d'une forme symétrique

1. Choisir aléatoirement une fonction de transition F_s
 2. Choisir un nombre de générations $Ng \leq Ngmax$
 3. Sh^0 une forme initiale
 4. **Pour** k de 1 à *taille de F_s* **faire**
 5. **Pour** i de 1 à Nombre de combinaisons symétriques du groupe k **faire**
 6. $F(i) \leftarrow F_s(k)$
 7. **Fin pour**
 8. **Fin pour**
 9. **Pour** t de 1 à Ng **faire**
 10. Appliquer la fonction de transition F à la forme Sh^{t-1}
 11. **Fin pour**
-

En ce qui concerne le type de symétrie, nous utilisons les notations suivantes :

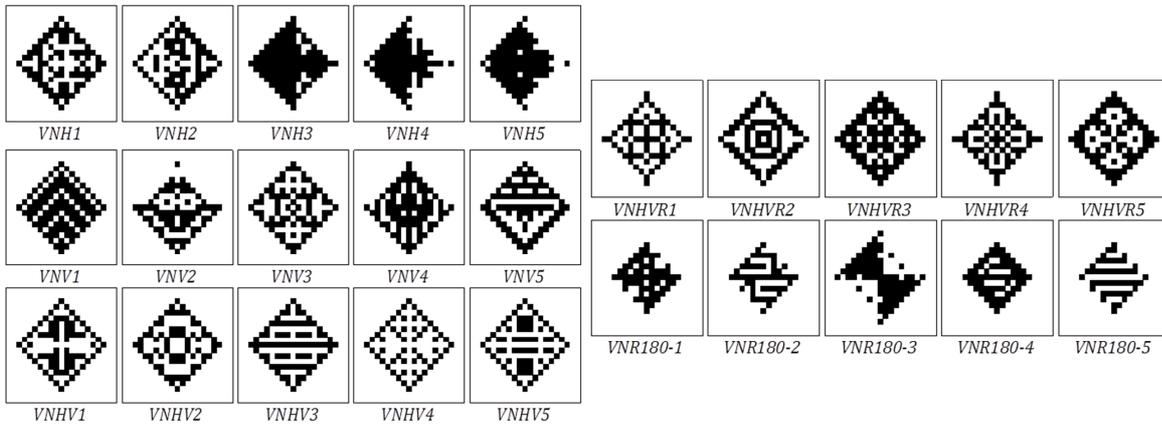
- H : symétrie horizontale.
- V : symétrie verticale.
- HV : symétrie horizontale et verticale.
- HVR : symétrie axiale et rotationnelle.
- R180 : rotation de 180° .
- R90 : rotation de 90° .

V.1.3 Formes 3D binaires symétriques

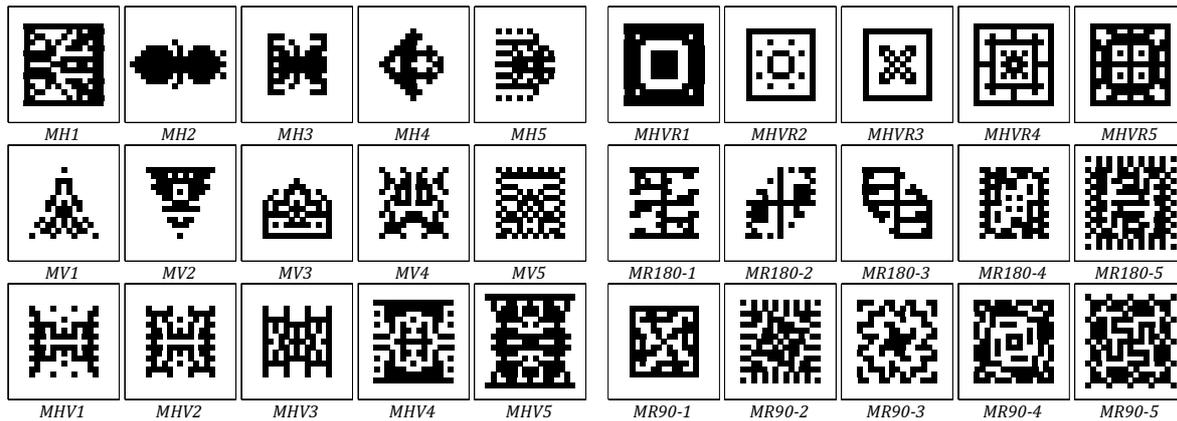
Des formes 3D avec le système de Von Neumann à 7 cellules voisines ont été générées (Figure V.4). Les deux premières formes (Losange3D et Pyramide3D) sont des formes pleines. La dernière forme Flocon3D est une forme creuse. Les deux premières formes ont été générées

manuellement en donnant un état à chaque cellule de la grille de l'automate cellulaire. La dernière forme a été générée en utilisant une règle additive :

- Si la somme des cellules est égale à 1 ou 4 ou 5, la cellule prend l'état « 1 ».
- Dans les autres cas, la cellule prend l'état « 0 ».



(b) : Formes générées avec le système d'interaction de Von Neumann.



(c) : Formes générées avec le système d'interaction de Moore

Figure V.3: Formes 2D binaires symétriques.

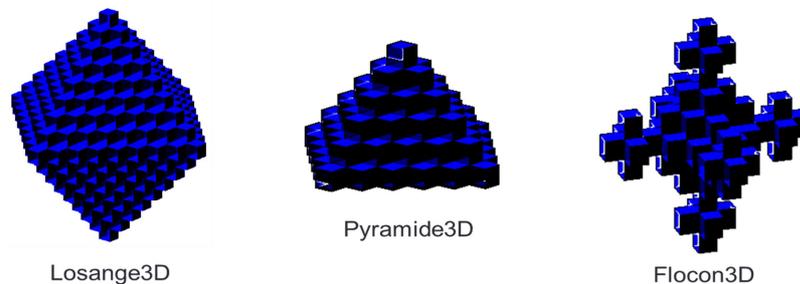


Figure V.4 : Formes 3D binaires.

V.1.4 Formes 2D à trois états symétriques

Des formes symétriques 2D à 3 états (Figure V.5) ont été testées. Une forme est générée pour chaque type de symétrie (horizontale « H », verticale « V », horizontale et verticale « HV », rotationnelle à 180° « R180 » et rotationnelle à 90° « R90 »). Ces formes sont générées soit avec

le système d'interaction de Von Neumann soit en utilisant le système d'interaction Moore. Ces formes ont été générées en utilisant l'algorithme présenté dans l'Algorithme V.2.

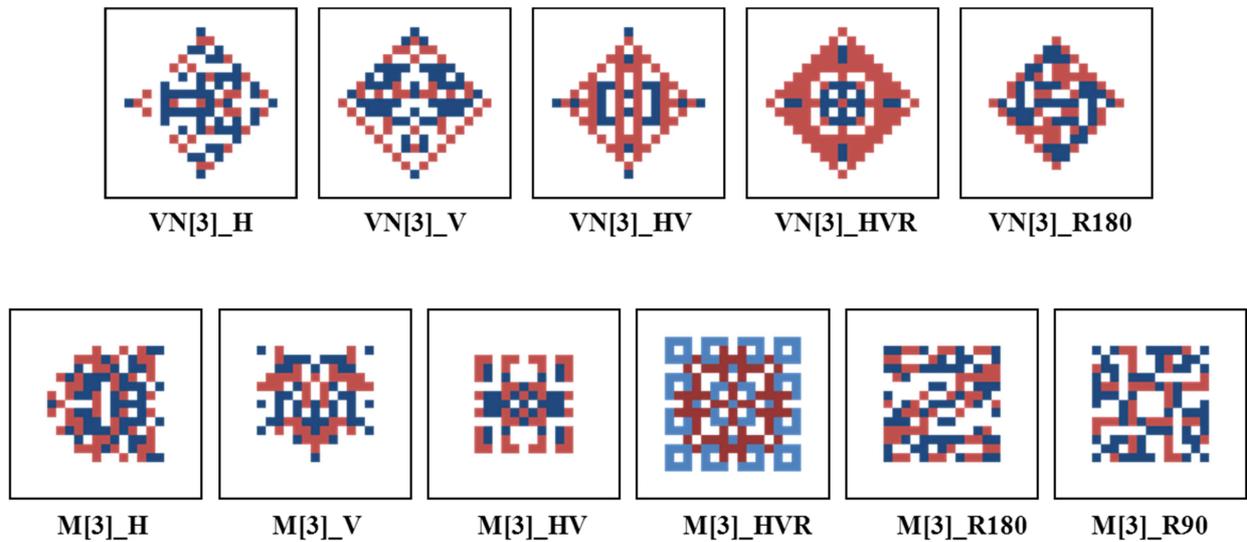


Figure V.5 : Formes Symétriques à 3 états.

V.2 Comparaison et choix des critères de similarité

Dans cette partie, nous présentons une comparaison des critères de similarité permettant de sélectionner ceux qui sont les plus efficaces.

Dans le Tableau V.1, nous comparons les critères de similarités proposés (le critère de De Garis « Fit », le nombre de cellules identiques « ID », l'écart quadratique « EQuad », l'extension du critère de De Garis « Adaptation Fit », l'information mutuelle « IM » et les moments de Legendre « ML »).

		Fit	ID	EQuad	Adaptation fit	IM	ML
		√	√	√	√	√	√
			√	√	√	√	±
						√	
							√
							√

Tableau V.1 : Comparaison des critères de similarité.

L'efficacité de chacun de ces critères dépend des formes comparées :

- Lorsque les deux formes comparées sont pleines, tous les critères peuvent être utilisés (formes de la première ligne).

- Lorsque les deux formes comparées sont quelconques (formes de la deuxième ligne), le nombre de cellules identiques, l'information mutuelle et l'extension du critère de De Garis peuvent être utilisés. L'erreur quadratique peut être utilisée dans ce cas, si ces formes sont binaires. Les moments de Legendre sont moins efficaces sur ce type de formes.
- Lorsque les formes comparées sont les mêmes mais les états qui les forment sont différents (formes de la troisième ligne), seule l'information mutuelle détermine la ressemblance entre ces deux formes.
- Lorsque les deux formes comparées sont les mêmes mais dans des positions différentes par rotation (formes de la quatrième ligne), ou ces formes sont de tailles différentes (formes de la cinquième ligne), seuls les moments de Legendre peuvent être utilisés.

Parmi ces critères, nous n'avons retenu dans un premier temps que trois critères :

- Nombre de cellules identiques : car il est le critère le plus simple à interpréter.
- L'information mutuelle : car elle est très utilisée dans le domaine du traitement d'image pour comparer deux images. Ce critère peut être utilisé de la même façon dans le domaine des automates cellulaires.
- Les moments de Legendre : le seul critère invariant par rotation et par échelle.

Dans la section suivante, une comparaison de ces trois critères est donnée.

V.2.1 Comparaison des critères : nombre de cellules identiques, information mutuelle et moments de Legendre

Nous avons considéré deux formes 2D à 2 états : Sh^* comme forme de référence (Figure V.6) de taille 21*21 et Sh comme forme de référence. Cette dernière forme est initialement la même que la forme de référence. Nous avons ensuite modifié toutes les cellules de cette forme (passage de « 0 » à « 1 » et vice-versa) jusqu'à ce que tous ses états soient inversés. A chaque modification d'une cellule, nous calculons les trois critères de similarité (IM, ID et ML) entre la forme de référence Sh^* et la forme Sh modifiée.

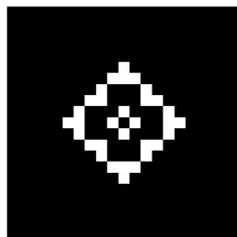


Figure V.6 : Forme de référence.

Les étapes de cette comparaison sont données dans l'Algorithme V.3. $ER_C(Sh^*, Sh)$ désigne l'écart relatif absolu entre la forme de référence Sh^* et Sh pour le critère C . Ces écarts-relatifs ont été calculés pour chaque critère C comme suit :

$$ER_C(Sh^*, Sh) = \left| \frac{C(Sh^*, Sh^*) - C(Sh^*, Sh)}{C(Sh^*, Sh^*)} \right| \quad (77)$$

Algorithme V.3 : Algorithme de comparaison des critères de similarité

1. Choisir une forme Sh^*
2. $Sh \leftarrow Sh^*$
3. **Pour** $i = 1$ à L **faire**
4. **Pour** $j = 1$ à L **faire**
5. $Sh(i,j) \leftarrow 1 - Sh(i,j)$
6. Calculer $ER_{ID}(Sh^*, Sh)$
7. Calculer $ER_{IM}(Sh^*, Sh)$
8. Calculer $ER_{ML}(Sh^*, Sh)$
9. **Fin Pour**
10. **Fin Pour**

Nous présentons dans la Figure V.7 les écarts relatifs obtenus pour chaque critère. Le nombre de cellules modifiées représente le nombre de cellules qui sont différentes entre les deux formes Sh^* et Sh . Les moments de Legendre sont calculés avec les ordres $p = 2$ et $q = 2$.

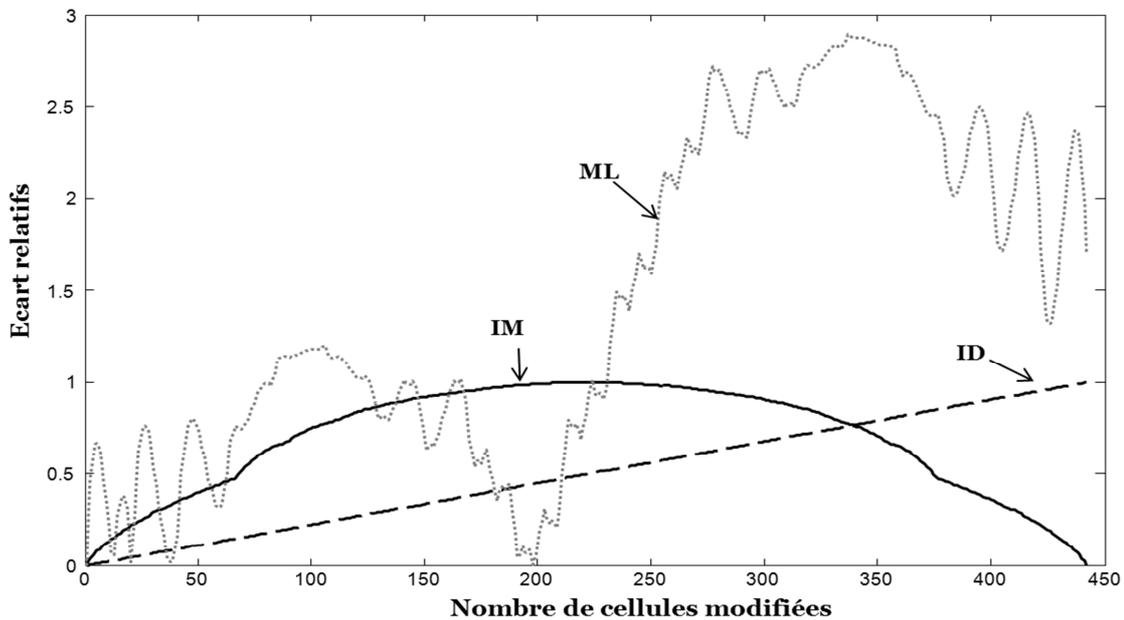


Figure V.7 : Ecarts Relatifs en fonction du critère.

- Le nombre de cellules identiques évolue de façon linéaire. L'écart relatif est nul lorsque le nombre de cellules modifiées est nul (les deux formes sont identiques) et il est maximal lorsque toutes les cellules de Sh sont modifiées.
- Avec l'information mutuelle, l'écart est nul lorsque le nombre de cellules modifiées est nul (les deux formes sont identiques) mais aussi lorsque toutes les cellules sont différentes (les deux formes sont inversées).
- Avec les moments de Legendre, l'écart relatif varie en fonction des ordres (p et q) choisis. On constate de nombreuses fluctuations de la courbe. Pour avoir une courbe lisse des écarts relatifs avec un écart relatif nul lorsque les deux formes sont identiques, les valeurs des ordres p et q doivent être importantes (Figure V.8).

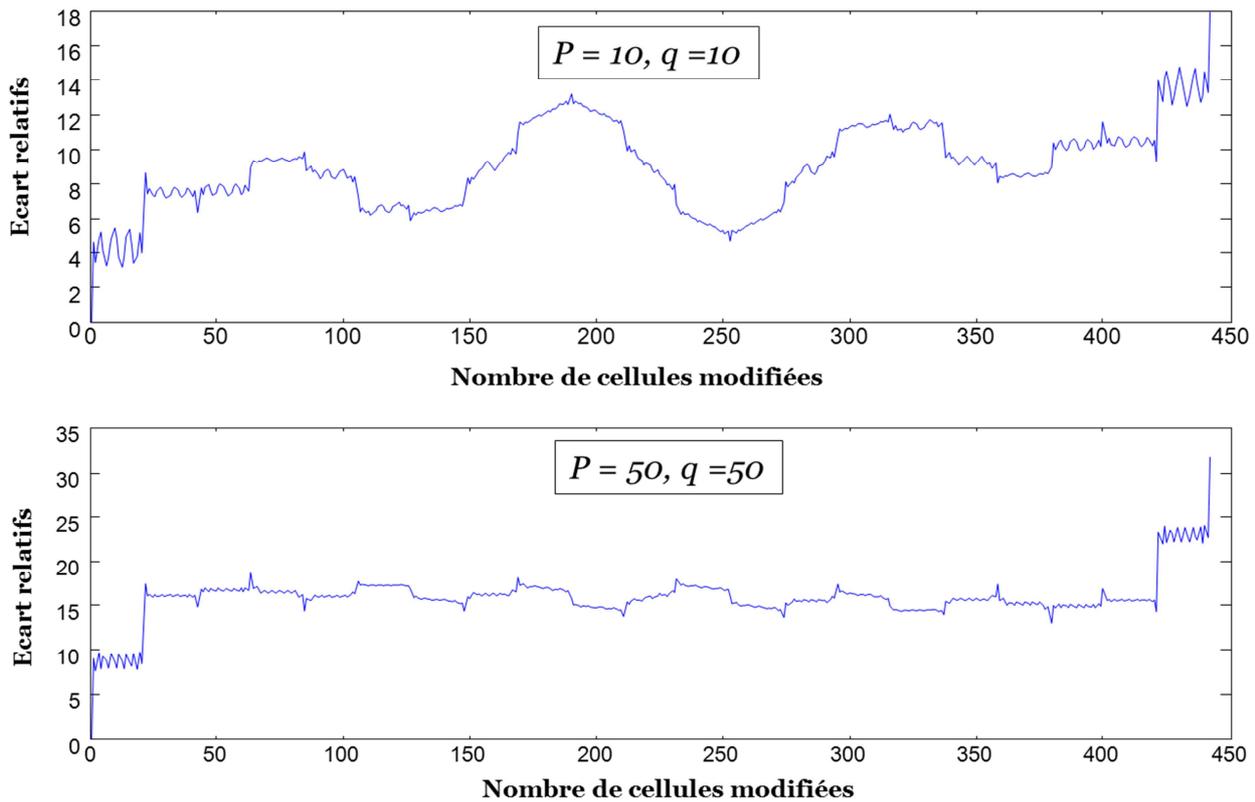


Figure V.8 : Evolution de l'écart relatif en fonction des ordres des polynômes de Legendre.

L'utilisation des moments de Legendre est très contraignante car d'une part, il faudra déterminer les ordres des polynômes pour chaque forme testée. D'autre part, ces ordres doivent être très grands ce qui rend le calcul de ce critère très long.

Une comparaison des deux critères information mutuelle et nombre de cellules identiques est donnée dans section suivante.

V.2.2 Comparaison du nombre de cellules identiques et de l'information mutuelle

Cette comparaison est basée sur des constatations visuelles du comportement des deux critères. Elle consiste à générer une forme de référence (Figure V.11) de taille 21×21 avec une fonction de transition F . Ensuite, toutes les fonctions de transition voisines F' de F sont générées par le système de voisinage Γ_1 . Chaque forme obtenue par une fonction de transition voisine est comparée à la forme de référence, en utilisant les deux critères : le nombre de cellules identiques et l'information mutuelle. Un tri dans un ordre croissant des valeurs de chacun des deux critères est réalisé.

La Figure V.9 et la Figure V.10 donnent les classements réalisés pour les deux critères. Le nombre de cellules identiques donne un classement de formes différent de celui de l'information mutuelle. Visuellement, les dernières formes classées par l'information mutuelle (IM élevé) ressemblent plus à la forme de référence, contrairement à celles du nombre de cellules identiques. L'information mutuelle peut être considérée comme un critère qualitatif.

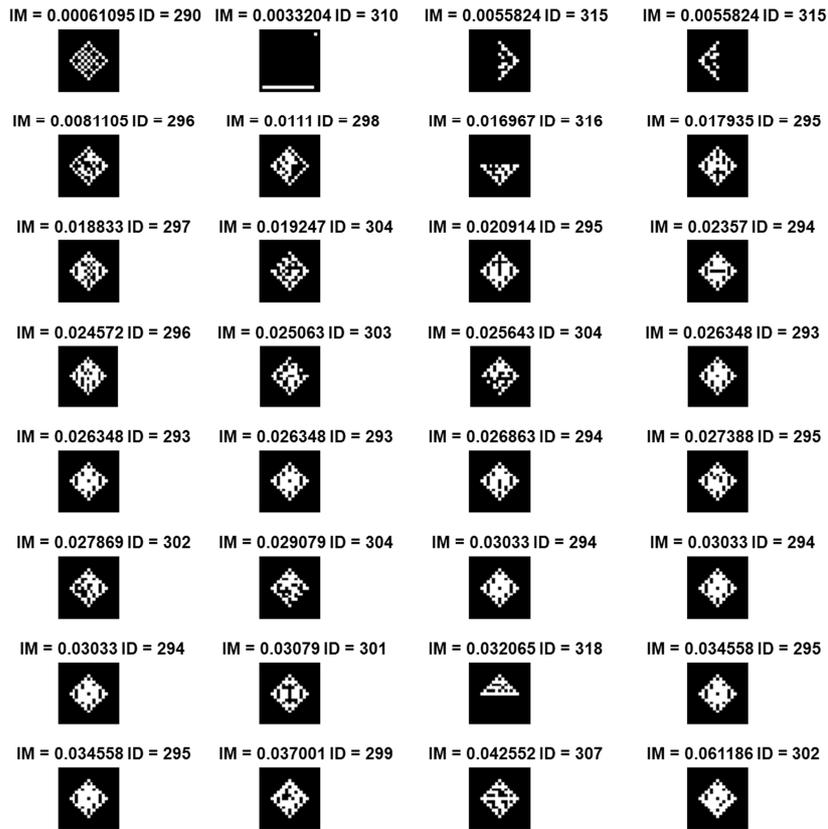


Figure V.9 : Classement des formes par l'information mutuelle.

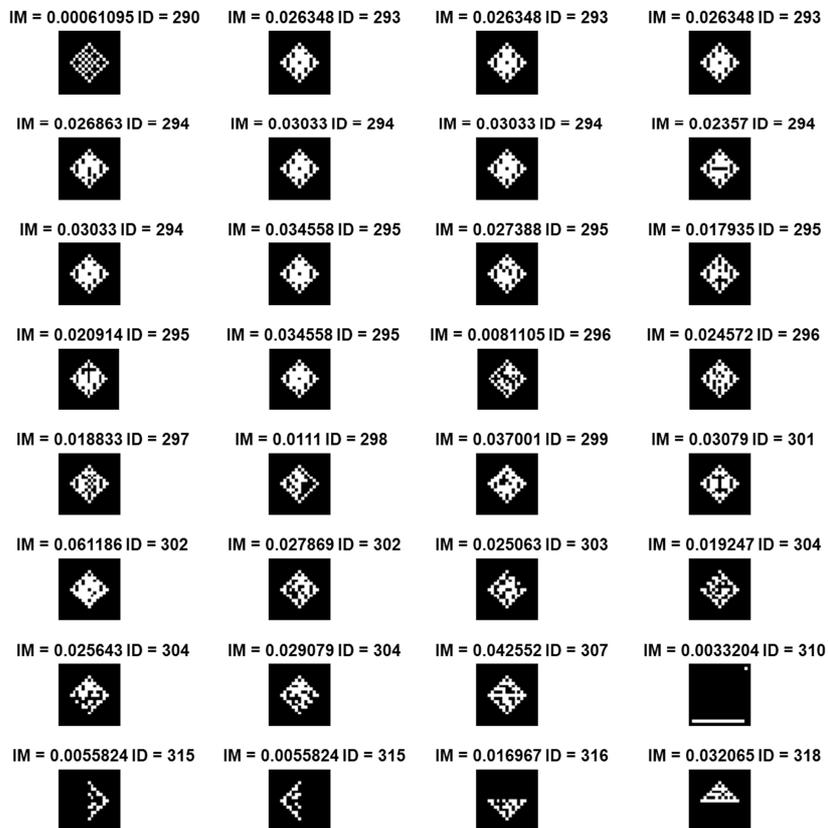


Figure V.10 : Classement des formes par nombre de cellules identiques.

De cette étude, nous avons choisi de prendre le nombre de cellules identiques et l'information mutuelle comme critères de similarité pour tous les tests des méthodes proposées.

V.3 Justification de l'utilisation des métaheuristiques

Dans cette partie, nous nous intéressons à la pertinence des métaheuristiques pour rechercher les règles d'un automate cellulaire. L'automate cellulaire est un système complexe et son comportement est souvent chaotique et donc difficile de prédire. Pour ce faire, nous avons réalisé les tests suivants :

- Premièrement, un échantillon de 5 millions de solutions est généré avec l'approche (F, Ng) et l'approche (Ng/F). Pour l'approche (Ng/F), le calcul du nombre de générations est réalisé en utilisant les deux critères de similarité (information mutuelle (Ng/F)_IM et nombre de cellules identiques (Ng/F)_ID).

Dans le Tableau V.2, pour chaque forme, le minimum (Min), la moyenne (Moy), le maximum (Max), l'écart-type (ET) et la valeur optimale (IMObj et IDObj) de l'information mutuelle et du nombre de cellules identiques sont donnés.

		Information mutuelle					Nombre de cellules identiques				
		IMObj	Min	Moy	Max	ET	IDObj	Min	Moy	Max	ET
ASVN1	(F, Ng)	0.35352	0.00000	0.01620	0.23401	0.01324	361	114	292.96	349	26.01
	(Ng/F)_IM		0.00000	0.03165	0.23401	0.01817		183	297.98	349	16.46
	(Ng/F)_ID		0.00000	0.01798	0.23401	0.01512		308	313.47	349	2.49
ASM2	(F, Ng)	0.58083	0.00000	0.01962	0.18735	0.01755	361	120	244.77	300	12.98
	(Ng/F)_IM		0.00000	0.04433	0.21522	0.02339		157	255.84	305	8.69
	(Ng/F)_ID		0.00000	0.04199	0.21522	0.02350		241	257.41	305	7.50

Tableau V.2: Résultats de l'échantillon de 5 millions de solutions.

Le Tableau V.3 montre qu'un échantillon aléatoire de l'espace de recherche ne permet pas d'obtenir des solutions de bonne qualité.

- Deuxièmement, toutes les solutions voisines de la solution optimale sont générées avec les deux systèmes de voisinage (Γ_1 et Γ_2) pour chaque forme avec chacune des deux approches (F, Ng) et (Ng/F). Les résultats sont présentés dans le Tableau V.3.

			Information mutuelle					Nombre de cellules identiques				
			IMObj	Min	Moy	Max	ET	IDObj	Min	Moy	Max	ET
ASVN1	(F, Ng)	Γ_1	0.35352	0.03628	0.33696	0.35352	0.04961	361	308	359.05	361	6.81
		Γ_2		0.01137	0.32176	0.35352	0.06589		275	357.22	361	9.22
	(Ng/F)_IM	Γ_1		0.06293	0.33869	0.35352	0.04434		314	359.34	361	5.69
		Γ_2		0.01449	0.32504	0.35352	0.05894		275	357.75	361	7.90
	(Ng/F)_ID	Γ_1		0.06293	0.33869	0.35352	0.04434		314	359.34	361	5.69
		Γ_2		0.01015	0.32471	0.35352	0.06040		313	357.78	361	7.70
ASM2	(F, Ng)	Γ_1	0.58083	0.02696	0.52185	0.58083	0.10251	361	254	353.77	361	15.43
		Γ_2		0.00709	0.47335	0.58083	0.12561		223	347.41	361	19.94
	(Ng/F)_IM	Γ_1		0.10113	0.52464	0.58083	0.09606		279	354.36	361	13.47
		Γ_2		0.03856	0.47841	0.58083	0.11731		257	348.52	361	17.27
	(Ng/F)_ID	Γ_1		0.10113	0.52464	0.58083	0.09606		279	354.36	361	13.47
		Γ_2		0.03856	0.47841	0.58083	0.11732		257	348.52	361	17.27

Tableau V.3 : Résultats des solutions voisines de la solution optimale.

- Troisièmement, comme il existe quelques règles d'évolution locales qui ne sont jamais utilisées, plusieurs solutions voisines sont identiques à la solution optimale en termes de critère de similarité. Dans le Tableau V.4, nous présentons les résultats obtenus avec uniquement les solutions voisines qui dégradent la solution optimale.

			Information mutuelle					Nombre de cellules identiques				
			IMObj	Min	Moy	Max	ET	IDObj	Min	Moy	Max	ET
ASVN1	(F, Ng)	$\Gamma1$	0.35352	0.03628	0.25455	0.34236	0.08095	361	308	349.35	360	12.81
		$\Gamma2$		0.01137	0.25004	0.34236	0.08201		275	348.70	360	13.12
	(Ng/F) – IM	$\Gamma1$		0.06293	0.26220	0.34236	0.07154		314	350.78	360	10.58
		$\Gamma2$		0.01449	0.25817	0.34236	0.07248		275	350.12	360	11.21
	(Ng/F) – ID	$\Gamma1$		0.06293	0.26220	0.34236	0.07154		314	350.78	360	10.58
$\Gamma2$		0.01015	0.25706	0.34236	0.07542	313	350.23	360	10.83			
ASM2	(F, Ng)	$\Gamma1$	0.58083	0.02696	0.45759	0.56774	0.11848	361	254	345.90	360	19.46
		$\Gamma2$		0.00709	0.43332	0.56774	0.12550		223	342.35	360	21.25
	(Ng/F) – IM	$\Gamma1$		0.10113	0.46268	0.56774	0.10992		279	347.03	360	16.71
		$\Gamma2$		0.03856	0.43965	0.56774	0.11618		257	343.79	360	18.16
	(Ng/F) – ID	$\Gamma1$		0.10113	0.46268	0.56774	0.10992		279	347.03	360	16.71
$\Gamma2$		0.03856	0.43965	0.56774	0.11619	257	343.79	360	18.16			

Tableau V.4 : Résultats des solutions voisines dégradantes.

Comme (Bagnoli, et al., 1992) l'ont démontré, la Tableau V.3 montre qu'autour d'une solution optimale, il existe des solutions de très mauvaise qualité. Mais en moyenne, le critère avec des solutions voisines de la solution optimale est largement plus important que celui obtenu avec un échantillon aléatoire.

Nous pouvons conclure de ces résultats, que les métaheuristiques à voisinage peuvent être utilisées pour résoudre ce problème inverse.

V.4 Comparaison avec les formes de la littérature

Les méthodes proposées ont été appliquées pour rechercher des formes pleines et des formes de la littérature. Ces formes sont des formes pleines proposées par (Chavoya & Duthen, 2006) et des formes creuses proposées par (Breukelaar & Back, 2005).

V.4.1 Comparaison avec les formes de Chavoya et Duthen

Nous avons appliqué l'algorithme génétique (AGs) et le recuit simulé proposés, pour générer les formes pleines proposées dans (Chavoya & Duthen, 2006) en utilisant l'approche (F, Ng), (Ng/F) et (F/Ng). Le critère de similarité considéré est l'information mutuelle. A la fin d'exécution de la méthode, la solution retournée est évaluée en utilisant le critère proposé par (Chavoya & Duthen, 2006).

Dans (Chavoya & Duthen, 2006), un système de mise à jour asynchrone est utilisé. Par conséquent, deux applications d'une fonction de transition obtenue par leur algorithme génétique pour mettre à jour un automate cellulaire, ne donnent pas forcément les mêmes formes. La Tableau V.5 présente la moyenne et l'écart type obtenus par (Chavoya & Duthen, 2006) avec 100 applications de la fonction de transition retournée par leur algorithme génétique, et la moyenne et l'écart type obtenus avec 5 répliques de l'algorithme génétique que nous avons proposé (Aïboud, et al., 2011). Ces résultats montrent que les méthodes que nous proposons donnent des résultats satisfaisants sur des formes pleines. Sur toutes les instances, en moyenne nous obtenons de meilleurs résultats que (Chavoya & Duthen, 2006).

	Von Neumann				Moore			
	Chavoya and Duthen		Extension results		Chavoya and Duthen		Extension results	
	Moyenne	Ecart-type	Moyenne	Ecart-type	Moyenne	Ecart-type	Moyenne	Ecart-type
Losange	1	0	1	0	0,805	0,040	1	0
Triangle	0,580	0	1	0	0,950	0,011	0,993	0,022
Carré	0,738	0,008	0,794	0,012	0,993	0,003	1	0
<i>Moyenne</i>	<i>0,772</i>	<i>0,002</i>	<i>0,931</i>	<i>0,004</i>	<i>0,916</i>	<i>0,018</i>	<i>0,997</i>	<i>0,007</i>

Tableau V.5 : Comparaison des résultats sur des formes pleines générées par (Chavoya & Duthen, 2006).

V.4.2 Résultats sur les formes de Breukelaar et Back

Nous avons également testé nos algorithmes sur des formes 2D (Figure 52) de taille 5*5 proposées par (Breukelaar et Back 2005). La première forme représente un carré avec un creux, la deuxième a la forme d'un sablier, la troisième a la forme d'un cœur, la quatrième est le « Smily » et la cinquième est la lettre « R ».

(Breukelaar et Back 2005) ont utilisé un algorithme génétique avec une population de taille 100 et un nombre d'itérations de 1000 pour rechercher ces différentes formes. Pour cette raison, nous avons pris les mêmes paramètres pour notre algorithme génétique. Pour la recherche locale itérée et le recuit simulé, le nombre d'itérations maximal est fixé à 50000. Nous avons considéré le système de voisinage Γ_1 .

Pour chacune de ces formes, nous avons appliqué chacun de nos algorithmes 100 fois. Les résultats obtenus sont présentés dans le Tableau V.6 avec chaque méthode, approche et critère. Pour chaque forme, le nombre de fois où la solution optimale est obtenue est donné. Les valeurs en gras représentent les cas où les méthodes proposées donnent des résultats équivalents ou meilleurs que ceux de (Breukelaar & Back, 2005).

Nos algorithmes ont donné de bons résultats en particulier avec la recherche locale itérée. En augmentant le nombre d'itérations, nous retrouvons à 100% toutes les formes avec toutes les approches et méthodes.

Le manque de détails concernant la méthode proposée par (Breukelaar & Back, 2005), en particulier la fonction objectif et l'approche de couplage de leur automate cellulaire de leur algorithme génétique, n'a pas permis de faire une comparaison des méthodes.

	(Breukelaar & Back, 2005)	AGs						RLI						RS					
		(F, Ng)		(Ng/F)		(F/Ng)		(F, Ng)		(Ng/F)		(F/Ng)		(F, Ng)		(Ng/F)		(F/Ng)	
		IM	ID	IM	ID	IM	ID	IM	ID	IM	ID	IM	ID	IM	ID	IM	ID	IM	ID
Carré	<i>100</i>	93	94	99	100	98	97	100	100	100	100								
Sablier	<i>96</i>	78	83	96	99	100	94	99	99	100	100	99							
Coeur	<i>55</i>	36	46	65	62	73	92	94	99	100	100	100	100	44	49	82	79	82	98
Smily	<i>29</i>	12	4	24	20	27	11	46	34	80	75	95	96	10	9	31	23	31	38
R	<i>18</i>	1	10	20	26	26	44	16	28	36	74	68	85	5	8	14	29	14	54

Tableau V.6 : Résultats sur les formes (Breukelaar & Back, 2005).

V.5 Résultats des formes 2D binaires quelconques

Cette partie concerne les résultats obtenus avec les formes quelconques. Nous allons présenter d'abord les résultats en utilisant les heuristiques puis ceux obtenus avec les métaheuristiques.

V.5.1 Heuristiques

Dans cette partie, les résultats des heuristiques sont présentés dans une première partie, puis les résultats de l'hybridation de ces méthodes sont donnés.

V.5.1.1 Résultats des heuristiques

Les heuristiques proposées ont été testées pour générer les formes 2D binaires quelconques. Pour les formes générées avec les systèmes d'interaction de Von Neumann, nous considérons les deux systèmes d'interaction de Von Neumann et Moore. Pour les formes générées avec le système d'interaction de Moore, nous considérons uniquement le système d'interaction de Moore. Comme Moore est une extension du système d'interaction de Von Neumann, il ne semble pas

intéressant de considérer le système de Von Neumann pour rechercher les formes générées avec le système d'interaction de Moore.

La Figure V.11 montre les différentes combinaisons des systèmes d'interaction, méthodes, approches, critères, et système d'interaction pour chacune des formes recherchées. Nous considérons :

- *Trois heuristiques* : la recherche locale déterministe avec le meilleur voisin (RLDM), la recherche locale déterministe avec le premier voisin (RLDP) et l'algorithme glouton (AGIt).
- *Trois approches* : approche (F, Ng), approche (Ng/F) et approche (F/Ng).
- *Deux critères de similarité* : le nombre de cellules identiques (ID) et l'information mutuelle (IM).
- *Deux systèmes de voisinages* : deux systèmes de voisinage Γ_1 et Γ_2 sont considérés pour les recherches locales déterministes (RLDP et RLDM).

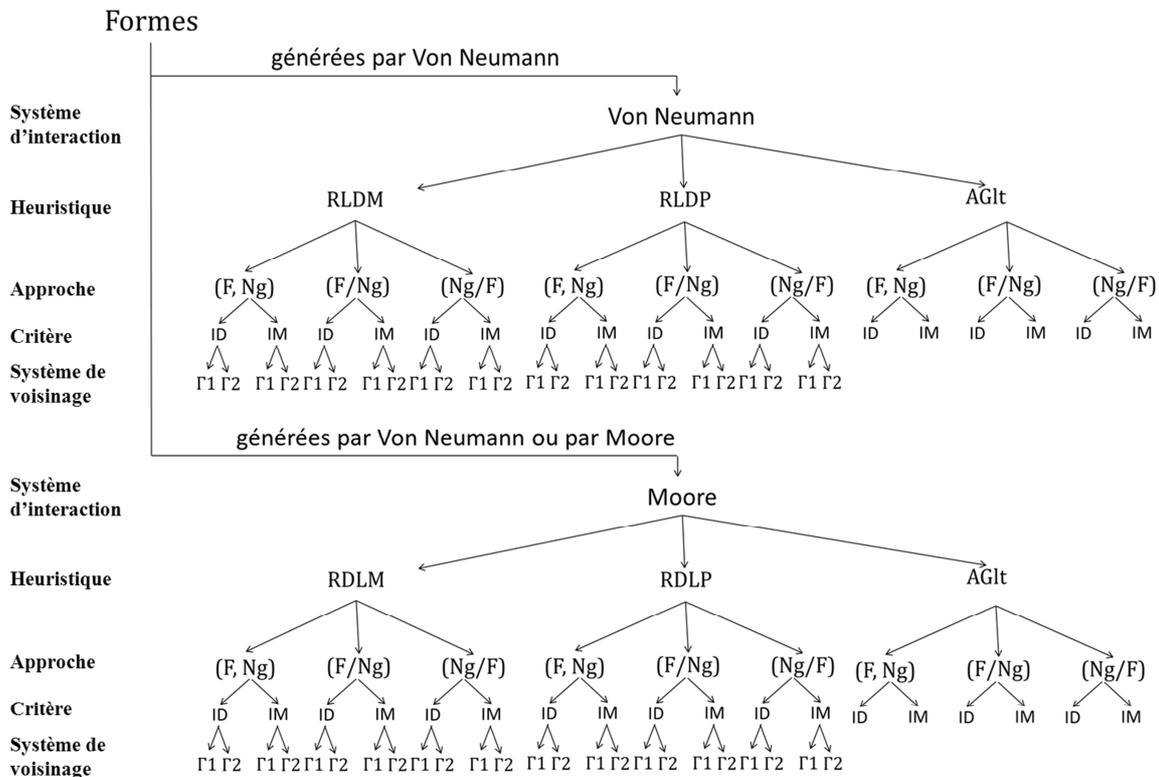


Figure V.11 : Combinaisons possibles de systèmes d'interaction, heuristiques, approches et critères.

Comme l'algorithme glouton possède un comportement stochastique et la solution initiale des recherches locales déterministes est choisie aléatoirement, nous lançons notre algorithme cinq fois pour chaque forme. Le nombre total de tests réalisés est de 50 avec le système d'interaction de Von Neumann pour chacune des feuilles de l'arbre de la Figure V.11 et il est de 100 avec le système d'interaction de Moore. Ce nombre est obtenu par le nombre de formes multiplié par le nombre de réplifications.

La forme Sh obtenue par la solution retournée pour chacun des cas est comparée avec la forme recherchée Sh^* en utilisant les deux critères (information mutuelle et nombre de cellules identiques). Pour comparer nos algorithmes, nous proposons de calculer le ratio R entre le

nombre de cellules identiques entre la forme retournée par l'algorithme et la forme recherchée. Ce ratio est calculé en fonction du nombre de cellules identiques par :

$$R(Sh, Sh^*) = ID(Sh, Sh^*) / ID(Sh^*, Sh^*) \quad (78)$$

Le Tableau V.7 présente les résultats obtenus avec les trois heuristiques et les différentes approches. Ces résultats sont répartis en quatre classes en fonction de la valeur du ratio R . Pour chaque classe $[a, b[$, nous donnons le pourcentage de solutions possédant un ratio R compris entre a et b . La dernière classe inclut le nombre de solutions optimales où le ratio $R = 100\%$. Les meilleurs résultats obtenus pour chaque méthode sont en gras. Pour sélectionner les meilleurs résultats, nous nous basons sur la somme des pourcentages des classes $[90\%, 95\%[$ et $[95\%, 100\%]$. Dans le cas d'égalité entre deux résultats, nous choisissons comme meilleur résultat celui qui possède le plus grand pourcentage dans la dernière classe $[95\%, 100\%]$.

- Pour les formes générées avec le système d'interaction de Von Neumann :
 - Avec le système d'interaction de Von Neumann, les méthodes de recherche locales ont donné de meilleurs résultats que l'algorithme glouton. Dans la plupart des cas, l'approche (Ng/G) avec le système de voisinage Γ_2 a donné les meilleurs résultats en particulier avec le critère nombre de cellules identiques.
 - Avec le système d'interaction de Moore, l'algorithme glouton avec l'approche (Ng/F) a donné de meilleurs résultats que les méthodes de recherches locales déterministes avec l'information mutuelle. Avec le nombre de cellules identiques, les meilleurs résultats sont obtenus avec la recherche locale avec le premier voisin. Les meilleurs résultats des recherches locales sont obtenus avec l'approche (F/Ng) et le système de voisinage Γ_2 .
- Pour les formes générées avec le système d'interaction de Moore, les meilleurs résultats sont obtenus avec l'algorithme glouton avec l'approche (Ng/F).

Dans tous les cas, le nombre de cellules identiques a donné de meilleurs résultats que l'information mutuelle.

		Formes générées avec Von Neumann								Formes générées avec Moore																	
		Von Neumann				Moore				Moore																	
		IM		ID		IM		ID		IM				ID													
		0%, 80%[80%, 90%[90%, 95%[95%, 100%]	0%, 80%[80%, 90%[90%, 95%[95%, 100%]	0%, 80%[80%, 90%[90%, 95%[95%, 100%]	0%, 80%[80%, 90%[90%, 95%[95%, 100%]										
AGH	(F, Ng)	-	0.40	0.06	0.08	0.46	0.00	0.48	0.24	0.28	0.34	0.04	0.40	0.40	0.22	0.00	0.72	0.28	0.00	0.88	0.00	0.10	0.02	0.10	0.58	0.30	0.02
	(Ng/F)	-	0.42	0.00	0.20	0.38	0.00	0.00	0.12	0.88	0.14	0.00	0.18	0.68	0.00	0.12	0.34	0.54	0.26	0.32	0.32	0.10	0.00	0.28	0.52	0.20	
	(F/Ng)	-	0.30	0.70	0.00	0.00	0.00	0.00	0.20	0.80	0.30	0.70	0.00	0.00	0.00	0.48	0.22	0.30	1.00	0.00	0.00	0.00	0.00	0.00	0.54	0.40	0.06
RLDP	(F, Ng)	Γ_1	0.26	0.04	0.30	0.40	0.00	0.24	0.20	0.56	0.04	0.72	0.20	0.04	0.02	0.78	0.12	0.08	0.72	0.24	0.04	0.00	0.64	0.36	0.00	0.00	
		Γ_2	0.20	0.08	0.10	0.62	0.00	0.02	0.06	0.92	0.14	0.28	0.24	0.34	0.00	0.40	0.30	0.30	0.50	0.28	0.18	0.04	0.16	0.46	0.24	0.14	
	(Ng/F)	Γ_1	0.42	0.02	0.24	0.32	0.00	0.02	0.28	0.70	0.10	0.50	0.30	0.10	0.00	0.62	0.22	0.16	0.52	0.42	0.06	0.00	0.48	0.46	0.06	0.00	
		Γ_2	0.00	0.00	0.12	0.88	0.00	0.00	0.02	0.98	0.00	0.36	0.30	0.34	0.00	0.36	0.32	0.32	0.38	0.46	0.16	0.00	0.32	0.50	0.18	0.00	
	(F/Ng)	Γ_1	0.04	0.00	0.38	0.58	0.00	0.02	0.26	0.72	0.04	0.40	0.34	0.22	0.00	0.42	0.44	0.14	0.42	0.50	0.06	0.02	0.30	0.58	0.10	0.02	
		Γ_2	0.00	0.00	0.18	0.82	0.00	0.00	0.08	0.92	0.02	0.20	0.30	0.48	0.00	0.06	0.46	0.48	0.34	0.44	0.20	0.02	0.32	0.50	0.18	0.00	
RLDM	(F, Ng)	Γ_1	0.28	0.14	0.24	0.34	0.00	0.32	0.26	0.42	0.10	0.66	0.14	0.10	0.00	0.80	0.18	0.02	0.58	0.42	0.00	0.00	0.54	0.38	0.08	0.00	
		Γ_2	0.26	0.08	0.06	0.60	0.00	0.02	0.08	0.90	0.08	0.32	0.44	0.16	0.00	0.50	0.38	0.12	0.52	0.34	0.10	0.04	0.32	0.54	0.10	0.04	
	(Ng/F)	Γ_1	0.38	0.02	0.28	0.32	0.00	0.06	0.36	0.58	0.24	0.48	0.22	0.06	0.00	0.56	0.34	0.10	0.62	0.32	0.04	0.02	0.48	0.48	0.02	0.02	
		Γ_2	0.04	0.00	0.16	0.80	0.00	0.00	0.06	0.94	0.02	0.24	0.32	0.42	0.00	0.30	0.34	0.36	0.38	0.40	0.18	0.04	0.26	0.40	0.26	0.08	
	(F/Ng)	Γ_1	0.02	0.04	0.44	0.50	0.00	0.00	0.34	0.66	0.04	0.64	0.26	0.06	0.00	0.40	0.18	0.00	0.00	0.54	0.40	0.06	0.40	0.50	0.08	0.02	
		Γ_2	0.00	0.00	0.18	0.82	0.00	0.00	0.08	0.92	0.00	0.18	0.42	0.40	0.00	0.10	0.50	0.40	0.26	0.48	0.26	0.00	0.28	0.46	0.26	0.00	

Tableau V.7 : Résultats des heuristiques sur les formes quelconques.

Afin de pouvoir comparer les résultats des trois approches, nous avons regroupé tous les résultats des heuristiques (algorithme glouton et ceux des deux recherches locales déterministes avec $\Gamma 1$ et $\Gamma 2$).

Les résultats des heuristiques groupées par approche sont présentés dans le Tableau V.8. Pour chaque approche, nous donnons le pourcentage des cas en fonction des valeurs du ratio R. Ces résultats montrent que les approches (F/Ng) et (Ng/F) donnent de meilleurs résultats que l'approche (F, Ng).

	Formes générées par Von Neumann												Formes générées par Moore											
	Von Neumann						Moore						Moore											
	IM				ID				IM				ID				IM				ID			
	[0%, 80%]	[80%, 90%]	[90%, 95%]	[95%, 100%]	[0%, 80%]	[80%, 90%]	[90%, 95%]	[95%, 100%]	[0%, 80%]	[80%, 90%]	[90%, 95%]	[95%, 100%]	[0%, 80%]	[80%, 90%]	[90%, 95%]	[95%, 100%]	[0%, 80%]	[80%, 90%]	[90%, 95%]	[95%, 100%]				
(F, Ng)	0.28	0.08	0.16	0.48	0.00	0.22	0.17	0.62	0.14	0.40	0.28	0.17	0.00	0.64	0.25	0.10	0.64	0.26	0.08	0.02	0.35	0.46	0.14	0.04
(Ng/F)	0.25	0.01	0.20	0.54	0.00	0.02	0.17	0.82	0.10	0.32	0.26	0.32	0.00	0.39	0.31	0.30	0.43	0.38	0.15	0.03	0.31	0.42	0.21	0.06
(F/Ng)	0.07	0.15	0.24	0.54	0.00	0.00	0.19	0.80	0.08	0.42	0.26	0.23	0.08	0.29	0.36	0.26	0.40	0.39	0.18	0.02	0.26	0.52	0.20	0.02

Tableau V.8 : Résultats des heuristiques groupés par approche.

Des classements en utilisant le ratio de l'information mutuelle, donnent des résultats comparables à ceux du classement avec le ratio calculé par le nombre de cellules identiques.

V.5.1.2 Résultats de l'hybridation des heuristiques

Afin d'améliorer nos résultats par heuristiques, nous avons fait une hybridation, entre l'algorithme glouton et la recherche locale déterministe avec le premier voisin. Le système de voisinage $\Gamma 2$ est considéré. L'hybridation est de type coopération relais de haut niveau. L'algorithme glouton est utilisé pour générer une solution initiale pour la recherche locale déterministe.

Les résultats obtenus avec l'hybridation sont résumés dans le Tableau V.9. Cette hybridation entre l'algorithme glouton et la recherche locale déterministe améliore considérablement les résultats des heuristiques, en particulier, ceux obtenus avec le système d'interaction de Moore. Les meilleurs résultats sont obtenus dans la plupart des cas avec l'approche (Ng/F).

	Formes générées par Von Neumann												Formes générées par Moore											
	Von Neumann						Moore						Moore											
	IM				ID				IM				ID				IM				ID			
	[0%, 80%]	[80%, 90%]	[90%, 95%]	[95%, 100%]	[0%, 80%]	[80%, 90%]	[90%, 95%]	[95%, 100%]	[0%, 80%]	[80%, 90%]	[90%, 95%]	[95%, 100%]	[0%, 80%]	[80%, 90%]	[90%, 95%]	[95%, 100%]	[0%, 80%]	[80%, 90%]	[90%, 95%]	[95%, 100%]				
(F, Ng)	0.02	0.00	0.08	0.90	0.00	0.00	0.00	1.00	0.20	0.04	0.16	0.60	0.00	0.24	0.40	0.36	0.34	0.10	0.14	0.42	0.00	0.14	0.62	0.24
(Ng/F)	0.00	0.00	0.10	0.90	0.00	0.00	0.00	1.00	0.14	0.00	0.06	0.80	0.00	0.06	0.24	0.70	0.12	0.18	0.48	0.22	0.00	0.06	0.68	0.26
(F/Ng)	0.00	0.00	0.16	0.84	0.00	0.00	0.06	0.94	0.00	0.00	0.10	0.90	0.00	0.18	0.28	0.54	0.00	0.54	0.30	0.16	0.00	0.44	0.42	0.14

Tableau V.9 : Résultats de l'hybridation de l'algorithme glouton et la recherche locale déterministe avec le premier voisin.

V.5.2 Métaheuristiques

Les métaheuristiques proposées ont été mises en œuvre avec les différentes approches, système d'interaction, critère et systèmes de voisinage.

V.5.2.1 Paramétrage

Les paramètres de l'algorithme génétique sont :

- *Taille de la population initiale* : la population initiale est composée de 5000 chromosomes quelle que soit l'approche utilisée ;
- *Probabilité de sélection* : chaque chromosome est sélectionné avec une probabilité de 0.9 pour se reproduire ;
- *Probabilité de croisement* : la probabilité de croisement $P_c = 0.7$;
- *Probabilité de mutation* : chaque chromosome est muté avec une probabilité $P_c = 0.3$;
- *Remplacement élitiste* : à chaque itération de l'algorithme génétique, 20% des meilleurs chromosomes de la population de l'itération précédente sont conservés.

Les paramètres de la recherche locale itérée (RLI) concernent :

- *le nombre d'itérations* : quelle que soit l'approche utilisée, le nombre d'itérations de la recherche locale itérée est de $N_{Iter} = 5.10^6$.
- *La taille du palier* : avec le système de voisinage Γ_1 , la taille du palier $A = 0.7 \text{Card}(\Gamma)$. Pour les systèmes Γ_2 et Γ_3 , elle est respectivement égale à 10000 et 50000.

Les paramètres du recuit simulé (RS) sont :

- *le nombre d'itérations* : quelle que soit l'approche utilisée, le nombre d'itérations du recuit simulé est de $N_{Iter} = 5.10^6$.
- *la température initiale* : $T_0 = 10$;
- *la température d'arrêt* : $T_a = 10^{-4}$.
- *Le facteur de décroissance de température* : il est obtenu par la relation $T_f = N_{Iter} \sqrt{T_a / T_0}$.

V.5.2.2 Résultats

Tout comme pour les heuristiques, les formes de Von Neumann ont été testées avec le système d'interaction de Von Neumann et de Moore. Les formes de Moore sont testées uniquement avec le système d'interaction de Moore.

Chacune des formes est testée avec :

- *Trois métaheuristiques* : la recherche locale itérée (RLI), le recuit simulé (RS) et l'algorithme génétique).
- *Trois approches* : (F, Ng), (Ng/F) et (F/Ng).
- *Trois systèmes de voisinages* : les systèmes de voisinage Γ_1 , Γ_2 et Γ_M sont considérés pour la recherche locale itérée et le recuit simulé.
- *Deux critères de similarité* : le nombre de cellules identiques (ID) et l'information mutuelle (IM).

Les différentes combinaisons entre les systèmes d'interaction, les métaheuristiques, les approches, les systèmes de voisinage et les critères sont données dans la Figure V.12.

Comme les métaheuristiques ont un comportement stochastique, nous avons réalisé cinq réplifications pour chaque forme. Le nombre total d'exécutions pour chacune des feuilles de l'arbre présenté dans la Figure V.12 est de 50 pour le système d'interaction de Von Neumann et

100 pour le système d'interaction de Moore (nombre de réplifications multiplié par le nombre de formes testées).

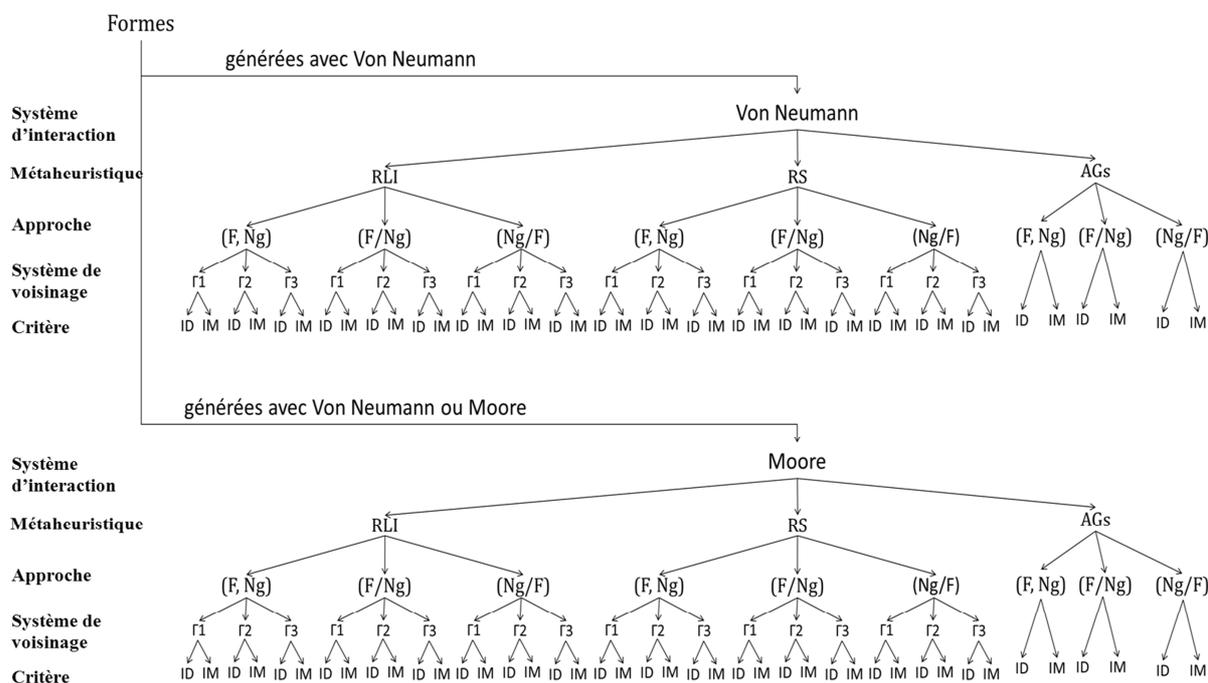


Figure V.12 : Combinaisons de métaheuristiques, systèmes d'interaction, approches, systèmes de voisinage et critères.

		Formes générées par Von Neumann								Formes générées par Moore												
		Von Neumann				Moore				Moore												
		IM		ID		IM		ID		IM		ID		IM		ID						
		0%, 80%]	[80%, 90%]	[90%, 95%]	[95%, 100%]	0%, 80%]	[80%, 90%]	[90%, 95%]	[95%, 100%]	0%, 80%]	[80%, 90%]	[90%, 95%]	[95%, 100%]	0%, 80%]	[80%, 90%]	[90%, 95%]	[95%, 100%]					
AGs	(F, Ng)	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.20	0.52	0.28	0.00	0.00	0.36	0.64	
	(Ng/F)	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.18	0.42	0.36	0.00	0.00	0.02	0.98	
	(F/Ng)	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.16	0.58	0.26	0.00	0.00	0.30	0.70	
RLI	(F, Ng)	Γ1	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.24	0.76	0.00	0.00	0.14	0.86
		Γ2	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.02	0.00	0.98	0.00	0.04	0.30	0.66	0.00	0.02	0.16	0.82
		Γ3	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.04	0.12	0.84	0.00	0.00	0.00	1.00	0.00	0.00	0.22	0.78
	(F, Ng)	Γ1	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.18	0.82	0.00	0.00	0.14	0.86
		Γ2	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.26	0.74	0.00	0.00	0.16	0.84
		Γ3	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.26	0.66	0.00	0.00	0.16	0.84
	(F, Ng)	Γ1	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.14	0.86	0.00	0.00	0.08	0.92
		Γ2	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.14	0.86	0.00	0.00	0.08	0.92
		Γ3	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.16	0.84	0.00	0.00	0.06	0.94
RS	(F, Ng)	Γ1	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.04	0.14	0.12	0.70	0.00	0.02	0.00	0.98
		Γ2	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.02	0.16	0.14	0.68	0.00	0.02	0.00	0.98
		Γ3	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.04	0.24	0.02	0.70	0.00	0.00	0.00	1.00
	(F, Ng)	Γ1	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.02	0.76	0.00	0.00	0.00	1.00
		Γ2	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.06	0.16	0.02	0.76	0.00	0.00	0.00	1.00
		Γ3	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.04	0.18	0.06	0.72	0.00	0.00	0.00	1.00
	(F, Ng)	Γ1	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.18	0.04	0.78	0.00	0.00	0.04	0.96
		Γ2	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.16	0.02	0.82	0.00	0.00	0.08	0.92
		Γ3	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.16	0.06	0.78	0.00	0.00	0.00	1.00

Tableau V.10 : Résultats des métaheuristiques sur les formes 2D binaires quelconques.

Le Tableau V.10 donne les résultats des métaheuristiques. Ces résultats sont répartis en quatre intervalles selon les valeurs du ratio R. ces classes sont les mêmes que celles utilisées pour les

heuristiques. Chaque intervalle $[a, b[$ donne le pourcentage des cas où le ratio R est compris entre a et b .

Les résultats obtenus avec les métaheuristiques sont meilleurs que ceux obtenus avec les heuristiques. Pour les formes générées avec le système d'interaction de Von Neumann, dans la plupart des cas, les solutions sont dans l'intervalle $[95\%, 100\%]$. Pour les formes de Moore, la plupart des solutions sont concentrées dans l'intervalle $[90\%, 100\%]$.

Pour affiner ces résultats et pouvoir comparer les résultats des métaheuristiques, nous proposons de présenter la répartition des solutions dans l'intervalle $[90\%, 100\%]$ (Tableau V.11). Ces résultats sont regroupés dans quatre classes $[90\%, 95\%[$, $[95\%, 98\%[$ et $[98\%, 100\%[$. La dernière classe appelée 100%, donne le nombre de solutions optimales. La somme des pourcentages de chacune de ces classes peut être différente de 1 (100%), car il existe des cas où le ratio R est inférieur à 90%.

		Formes générées par Von Neumann												Formes générées par Moore												
		Von Neumann						Moore						Moore												
		IM			ID			IM			ID			IM				ID								
		$[90\%, 95\%[$	$[95\%, 98\%[$	$[98\%, 100\%[$	100%	$[90\%, 95\%[$	$[95\%, 98\%[$	$[98\%, 100\%[$	100%	$[90\%, 95\%[$	$[95\%, 98\%[$	$[98\%, 100\%[$	100%	$[90\%, 95\%[$	$[95\%, 98\%[$	$[98\%, 100\%[$	100%	$[90\%, 95\%[$	$[95\%, 98\%[$	$[98\%, 100\%[$	100%					
AGs	(F, Ng)	0.00	0.12	0.04	0.84	0.00	0.02	0.04	0.94	0.00	0.12	0.74	0.14	0.00	0.10	0.74	0.16	0.52	0.28	0.00	0.00	0.36	0.58	0.06	0.00	
	(Ng/F)	0.00	0.14	0.00	0.86	0.00	0.04	0.00	0.96	0.00	0.22	0.66	0.12	0.00	0.06	0.78	0.16	0.42	0.26	0.10	0.00	0.02	0.56	0.42	0.00	
	(F/Ng)	0.00	0.10	0.00	0.90	0.00	0.04	0.06	0.90	0.00	0.16	0.62	0.22	0.00	0.02	0.60	0.38	0.58	0.26	0.00	0.00	0.30	0.66	0.04	0.00	
RLI	(F, Ng)	$\Gamma 1$	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.12	0.84	0.04	0.00	0.08	0.76	0.16	0.24	0.54	0.22	0.00	0.14	0.44	0.42	0.00
		$\Gamma 2$	0.00	0.00	0.08	0.92	0.00	0.00	0.08	0.92	0.00	0.18	0.70	0.10	0.00	0.08	0.56	0.36	0.30	0.26	0.32	0.08	0.16	0.38	0.32	0.12
		$\Gamma 3$	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.12	0.12	0.66	0.06	0.00	0.12	0.56	0.32	0.32	0.22	0.34	0.08	0.22	0.32	0.42	0.04
	(Ng/F)	$\Gamma 1$	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.10	0.84	0.06	0.00	0.08	0.74	0.18	0.18	0.30	0.48	0.04	0.14	0.34	0.50	0.02
		$\Gamma 2$	0.00	0.00	0.02	0.98	0.00	0.00	0.08	0.92	0.00	0.02	0.58	0.40	0.00	0.04	0.52	0.44	0.26	0.22	0.40	0.12	0.16	0.32	0.38	0.14
		$\Gamma 3$	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.60	0.40	0.02	0.06	0.50	0.42	0.26	0.18	0.32	0.16	0.16	0.32	0.34	0.18
	(F/Ng)	$\Gamma 1$	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.10	0.82	0.08	0.00	0.04	0.70	0.26	0.14	0.36	0.46	0.04	0.08	0.40	0.40	0.12
		$\Gamma 2$	0.00	0.00	0.04	0.96	0.00	0.00	0.04	0.96	0.00	0.04	0.60	0.36	0.00	0.04	0.70	0.26	0.14	0.34	0.36	0.16	0.08	0.40	0.32	0.20
		$\Gamma 3$	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.12	0.62	0.26	0.00	0.00	0.46	0.54	0.16	0.32	0.38	0.14	0.06	0.42	0.36	0.16
RS	(F, Ng)	$\Gamma 1$	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.10	0.48	0.42	0.00	0.10	0.18	0.72	0.12	0.14	0.30	0.26	0.00	0.10	0.56	0.32
		$\Gamma 2$	0.00	0.00	0.06	0.94	0.00	0.00	0.04	0.96	0.00	0.00	0.32	0.68	0.00	0.04	0.14	0.82	0.14	0.10	0.28	0.30	0.00	0.08	0.56	0.34
		$\Gamma 3$	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.26	0.74	0.00	0.10	0.06	0.84	0.02	0.14	0.32	0.24	0.00	0.08	0.68	0.24
	(Ng/F)	$\Gamma 1$	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.02	0.44	0.54	0.00	0.00	0.20	0.80	0.02	0.18	0.38	0.20	0.00	0.10	0.62	0.28
		$\Gamma 2$	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.02	0.26	0.72	0.00	0.00	0.16	0.84	0.02	0.16	0.36	0.24	0.00	0.16	0.56	0.28
		$\Gamma 3$	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.24	0.76	0.00	0.02	0.06	0.92	0.06	0.14	0.32	0.26	0.00	0.12	0.56	0.32
	(F/Ng)	$\Gamma 1$	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.04	0.46	0.50	0.00	0.00	0.26	0.74	0.04	0.16	0.46	0.16	0.04	0.36	0.40	0.20
		$\Gamma 2$	0.00	0.00	0.06	0.94	0.00	0.00	0.08	0.92	0.00	0.00	0.40	0.60	0.00	0.00	0.12	0.88	0.02	0.12	0.50	0.20	0.08	0.34	0.32	0.26
		$\Gamma 3$	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.28	0.72	0.00	0.00	0.10	0.90	0.06	0.10	0.58	0.10	0.00	0.40	0.36	0.24

Tableau V.11 : Résultats des métaheuristiques.

Dans tous les cas, les résultats obtenus avec l'algorithme génétique sont de moins bonne qualité que ceux obtenus avec la recherche locale itérée et le recuit simulé.

- Pour les formes générées avec le système d'interaction de Von Neumann, le ratio R est dans la majorité des cas égal à 100% avec le système d'interaction de Von Neumann, et supérieur à 98% avec le système d'interaction de Moore :
 - Avec le système d'interaction de Von Neumann, les résultats de la recherche locale itérée et ceux du recuit simulé sont comparables. Le système de voisinage $\Gamma 2$ a donné de moins bons résultats que les deux autres systèmes de voisinage.
 - Avec le système d'interaction de Moore, les résultats du recuit simulé sont meilleurs que ceux de la recherche locale itérée. Les meilleurs résultats sont obtenus dans la plupart des cas avec l'approche (F/Ng).

- Pour les formes générées avec le système d’interaction de Moore, les résultats du recuit simulé sont meilleurs que ceux de la recherche locale itérée. L’approche (F/Ng) a donné dans la plupart des cas les meilleurs résultats.

Les résultats des formes générées par le système d’interaction de Moore sont plus difficiles à retrouver que celles générées avec le système de Von Neumann.

Le Tableau V.12 donne les résultats des métaheuristiques groupés par approche.

- Pour l’algorithme génétique :
 - Avec les formes générées avec le système d’interaction de Von Neumann, les meilleurs résultats sont obtenus avec l’approche (F, Ng) et (F/Ng).
 - Avec les formes de Moore, l’approche (Ng/F) donne les meilleurs résultats.
- Pour la recherche locale itérée :
 - Avec les formes générées avec le système d’interaction de Von Neumann, les meilleurs résultats sont obtenus avec l’approche (Ng/F) et (F/Ng).
 - Avec les formes de Moore, l’approche (F/Ng) donne les meilleurs résultats.
- Pour le recuit simulé :
 - Avec les formes générées avec le système d’interaction de Von Neumann, l’approche (Ng/F) donne les meilleurs résultats avec le système d’interaction de Von Neumann et l’approche (F/Ng) donne les meilleurs résultats avec le système d’interaction de Moore.
 - Avec les formes générées avec le système d’interaction de Moore, les meilleurs résultats sont ceux de l’approche (F/Ng) en utilisant le critère d’information mutuelle. Avec le critère nombre de cellules identiques, l’approche (F, Ng) donne les meilleurs résultats.

		Formes générées par Von Neumann												Formes générées par Moore											
		Von Neumann				Moore				Moore															
		IM		ID		IM		ID		IM				ID											
		90%, 95%[95%, 98%[98%, 100%[100%	90%, 95%[95%, 98%[98%, 100%[100%	90%, 95%[95%, 98%[98%, 100%[100%	90%, 95%[95%, 98%[98%, 100%[100%								
AGs	(F, Ng)	0.00	0.12	0.04	0.84	0.00	0.02	0.04	0.94	0.00	0.12	0.74	0.14	0.00	0.10	0.74	0.16	0.52	0.28	0.00	0.00	0.36	0.58	0.06	0.00
	(Ng/F)	0.00	0.14	0.00	0.86	0.00	0.04	0.00	0.96	0.00	0.22	0.66	0.12	0.00	0.06	0.78	0.16	0.42	0.26	0.10	0.00	0.02	0.56	0.42	0.00
	(F/Ng)	0.00	0.10	0.00	0.90	0.00	0.04	0.06	0.90	0.00	0.16	0.62	0.22	0.00	0.02	0.60	0.38	0.58	0.26	0.00	0.00	0.30	0.66	0.04	0.00
RLI	(F, Ng)	0.00	0.00	0.03	0.97	0.00	0.00	0.03	0.97	0.04	0.14	0.73	0.07	0.00	0.09	0.63	0.28	0.29	0.34	0.29	0.05	0.17	0.38	0.39	0.05
	(Ng/F)	0.00	0.00	0.01	0.99	0.00	0.00	0.03	0.97	0.00	0.04	0.67	0.29	0.01	0.06	0.59	0.35	0.23	0.23	0.40	0.11	0.15	0.33	0.41	0.11
	(F/Ng)	0.00	0.00	0.01	0.99	0.00	0.00	0.01	0.99	0.00	0.09	0.68	0.23	0.00	0.03	0.62	0.35	0.15	0.34	0.40	0.11	0.07	0.41	0.36	0.16
RS	(F, Ng)	0.00	0.00	0.02	0.98	0.00	0.00	0.01	0.99	0.00	0.03	0.35	0.61	0.00	0.08	0.13	0.79	0.09	0.13	0.30	0.27	0.00	0.09	0.60	0.30
	(Ng/F)	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.01	0.31	0.67	0.00	0.01	0.14	0.85	0.03	0.16	0.35	0.23	0.00	0.13	0.58	0.29
	(F/Ng)	0.00	0.00	0.02	0.98	0.00	0.00	0.03	0.97	0.00	0.01	0.38	0.61	0.00	0.00	0.16	0.84	0.04	0.13	0.51	0.15	0.04	0.37	0.36	0.23

Tableau V.12 : Résultats des métaheuristiques regroupés par approche.

Le Tableau V.13 donne les résultats des métaheuristiques à individus regroupés par système de voisinage. Pour chaque approche, le pourcentage des cas en fonction des valeurs du ratio R sont donnés. Nous constatons que :

- Avec le système d’interaction de Von Neumann, les systèmes de voisinage Γ_1 et Γ_3 , on retrouve dans tous les cas la solution optimale quel que soit le critère utilisé.
- Avec le système d’interaction de Moore, Γ_2 et Γ_3 donnent les meilleurs résultats.

		Formes générées par Von Neumann								Formes générées par Moore															
		Von Neumann				Moore				Moore															
		IM		ID		IM		ID		IM				ID											
		[90%, 95%]	[95%, 98%]	[98%, 100%]	100%	[90%, 95%]	[95%, 98%]	[98%, 100%]	100%	[90%, 95%]	[95%, 98%]	[98%, 100%]	100%	[90%, 95%]	[95%, 98%]	[98%, 100%]	100%								
RLI	Γ1	0,00	0,00	0,00	1,00	0,00	0,00	0,00	1,00	0,00	0,11	0,83	0,06	0,00	0,07	0,73	0,20	0,19	0,40	0,39	0,03	0,12	0,39	0,44	0,05
	Γ2	0,00	0,00	0,05	0,95	0,00	0,00	0,07	0,93	0,00	0,08	0,63	0,29	0,00	0,05	0,59	0,35	0,23	0,27	0,36	0,12	0,13	0,37	0,34	0,15
	Γ3	0,00	0,00	0,00	1,00	0,00	0,00	0,00	1,00	0,04	0,08	0,63	0,24	0,01	0,06	0,51	0,43	0,25	0,24	0,35	0,13	0,15	0,35	0,37	0,13
RS	Γ1	0,00	0,00	0,00	1,00	0,00	0,00	0,00	1,00	0,00	0,05	0,46	0,49	0,00	0,03	0,21	0,75	0,06	0,16	0,38	0,21	0,01	0,19	0,53	0,27
	Γ2	0,00	0,00	0,04	0,96	0,00	0,00	0,04	0,96	0,00	0,01	0,33	0,67	0,00	0,01	0,14	0,85	0,06	0,13	0,38	0,25	0,03	0,19	0,48	0,29
	Γ3	0,00	0,00	0,00	1,00	0,00	0,00	0,00	1,00	0,00	0,00	0,26	0,74	0,00	0,04	0,07	0,89	0,05	0,13	0,41	0,20	0,00	0,20	0,53	0,27

Tableau V.13 : Résultats des métaheuristiques à individus regroupés par système de voisinage

En résumé, les meilleurs résultats sont obtenus avec le recuit simulé avec l'approche (F/Ng). Comme système de voisinage, Γ3 a donné dans la plupart des cas les meilleurs résultats.

V.6 Résultats des formes 2D binaires symétriques

Dans cette partie, nous avons appliqué uniquement les métaheuristiques pour rechercher les formes symétriques. Pour voir l'efficacité du codage en exploitant la symétrie présente dans la forme recherchée, nous avons considéré le codage sans symétrie et le codage exploitant la symétrie.

Pour chacune des formes symétriques Sh^* , nous lançons chaque méthode (algorithme génétique, recherche locale itérée et recuit simulé) en utilisant comme critère de similarité soit l'information mutuelle (IM), soit le nombre de cellules identiques (ID). L'algorithme génétique, la recherche locale itérée et le recuit simulé sont lancés cinq fois. Le nombre total de tests réalisés (nombre de répliquions \times le nombre de type de symétrie \times le nombre de formes par type de symétrie) est de 125 pour les formes générées avec Von Neumann et de 150 pour les formes générées avec le système d'interaction de Moore.

Dans cette partie, les résultats sont donnés pour chaque méthode. Les deux codages sont considérés : codage sans prise en compte de la symétrie et codage avec prise en compte de la symétrie.

V.6.1 Algorithme génétique

Tout comme pour les formes quelconques, les formes symétriques générées par le système d'interaction de Moore sont testées avec le système d'interaction de Moore, et les formes symétriques générées avec Von Neumann sont testées avec les deux systèmes d'interactions (Von Neumann et Moore). Chaque forme est testée avec

- Les deux types de codages : sans prise en compte de la symétrie et avec prise en compte de la symétrie ;
- Les trois approches : approche (F, Ng), approche (Ng/F) et approche (F/Ng) ;
- les deux critères : information mutuelle (IM) et nombre de cellules identiques (ID).

Le Tableau V.14 donne les résultats de l'algorithme génétique obtenus avec les formes 2D binaires symétriques. Les résultats de chaque type de symétrie (horizontale (H), verticale (V), horizontale et verticale (HV), horizontale, verticale et rotationnelle (HVR), rotation à 90° (Rot_90°) et rotation à 180° (Rot_180°) sont donnés.

Comme nous l’avons expliqué dans le chapitre précédent, dans le cas du système d’interaction de Von Neumann, le codage exploitant la symétrie horizontale, verticale et rotationnelle est équivalent au codage exploitant la symétrie par rotation à 90°. Dans le Tableau V.14, nous présentons uniquement des résultats pour la symétrie horizontale, verticale et rotationnelle avec le système de Von Neumann.

		Formes générées avec Von Neumann												Formes générées avec Moore												
		Von Neumann						Moore						Moore												
		IM			ID			IM			ID			IM				ID								
		[90%, 95%]	[95%, 98%]	[98%, 100%]	[90%, 95%]	[95%, 98%]	[98%, 100%]	[90%, 95%]	[95%, 98%]	[98%, 100%]	[90%, 95%]	[95%, 98%]	[98%, 100%]	[90%, 95%]	[95%, 98%]	[98%, 100%]	[90%, 95%]	[95%, 98%]	[98%, 100%]							
Codage sans symétrie	H	(F, Ng)	0.00	0.12	0.00	0.68	0.00	0.00	0.00	1.00	0.24	0.28	0.44	0.04	0.00	0.52	0.28	0.20	0.07	0.37	0.23	0.00	0.20	0.17	0.47	0.00
		(Ng/F)	0.00	0.04	0.00	0.80	0.00	0.00	0.00	1.00	0.24	0.48	0.24	0.04	0.00	0.52	0.44	0.04	0.13	0.43	0.13	0.00	0.27	0.27	0.30	0.00
		(F/Ng)	0.00	0.20	0.08	0.64	0.00	0.00	0.00	1.00	0.28	0.32	0.28	0.08	0.00	0.52	0.36	0.12	0.20	0.40	0.17	0.00	0.13	0.37	0.33	0.00
	V	(F, Ng)	0.00	0.12	0.00	0.68	0.00	0.00	0.00	1.00	0.24	0.28	0.44	0.04	0.04	0.40	0.36	0.20	0.07	0.37	0.23	0.00	0.03	0.33	0.43	0.03
		(Ng/F)	0.16	0.24	0.04	0.56	0.00	0.00	0.00	1.00	0.36	0.24	0.16	0.20	0.00	0.56	0.24	0.20	0.10	0.47	0.27	0.00	0.00	0.53	0.30	0.00
		(F/Ng)	0.24	0.20	0.04	0.52	0.00	0.00	0.00	1.00	0.20	0.52	0.08	0.20	0.00	0.48	0.32	0.20	0.10	0.63	0.10	0.00	0.00	0.57	0.27	0.00
	HV	(F, Ng)	0.08	0.12	0.00	0.64	0.08	0.00	0.00	0.92	0.12	0.56	0.32	0.00	0.24	0.40	0.24	0.00	0.17	0.17	0.17	0.00	0.23	0.23	0.37	0.00
		(Ng/F)	0.12	0.08	0.00	0.64	0.08	0.00	0.00	0.92	0.12	0.52	0.28	0.08	0.16	0.48	0.32	0.04	0.10	0.30	0.20	0.03	0.17	0.30	0.30	0.00
		(F/Ng)	0.04	0.16	0.00	0.68	0.12	0.00	0.00	0.88	0.28	0.56	0.16	0.00	0.00	0.44	0.40	0.16	0.37	0.30	0.17	0.00	0.13	0.33	0.30	0.00
	HVR	(F, Ng)	0.12	0.04	0.00	0.84	0.16	0.00	0.00	0.84	0.08	0.72	0.20	0.00	0.04	0.60	0.36	0.00	0.07	0.13	0.13	0.00	0.37	0.47	0.00	0.00
		(Ng/F)	0.04	0.08	0.00	0.88	0.00	0.08	0.00	0.92	0.08	0.84	0.08	0.00	0.00	0.64	0.36	0.00	0.17	0.43	0.07	0.00	0.40	0.43	0.00	0.00
		(F/Ng)	0.04	0.16	0.00	0.80	0.00	0.00	0.00	1.00	0.08	0.76	0.16	0.00	0.00	0.56	0.44	0.00	0.53	0.30	0.00	0.00	0.37	0.47	0.00	0.00
	Rot_180°	(F, Ng)	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.12	0.76	0.12	0.12	0.24	0.40	0.24	0.17	0.37	0.13	0.00	0.17	0.37	0.30	0.00
		(Ng/F)	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.08	0.76	0.16	0.00	0.20	0.72	0.08	0.33	0.40	0.10	0.00	0.13	0.43	0.23	0.00
		(F/Ng)	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.20	0.72	0.08	0.00	0.12	0.48	0.40	0.27	0.50	0.07	0.00	0.10	0.50	0.23	0.00
	Rot_90°	(F, Ng)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0.03	0.17	0.03	0.00	0.23	0.23	0.10	0.00
		(Ng/F)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0.27	0.17	0.07	0.00	0.27	0.30	0.03	0.00
		(F/Ng)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0.33	0.20	0.00	0.00	0.23	0.37	0.00	0.00
Codage avec symétrie	H	(F, Ng)	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.04	0.28	0.48	0.20	0.00	0.24	0.56	0.20	0.13	0.33	0.33	0.03	0.17	0.17	0.40	0.10
		(Ng/F)	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.04	0.32	0.44	0.20	0.00	0.16	0.64	0.20	0.10	0.27	0.43	0.00	0.03	0.30	0.43	0.07
		(F/Ng)	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.56	0.36	0.08	0.00	0.52	0.32	0.16	0.10	0.37	0.37	0.00	0.13	0.20	0.50	0.00
	V	(F, Ng)	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.04	0.28	0.48	0.20	0.00	0.12	0.60	0.28	0.13	0.33	0.33	0.03	0.00	0.10	0.50	0.23
		(Ng/F)	0.00	0.04	0.08	0.88	0.00	0.00	0.00	1.00	0.00	0.36	0.40	0.24	0.00	0.12	0.60	0.28	0.00	0.23	0.50	0.10	0.00	0.10	0.60	0.13
		(F/Ng)	0.12	0.04	0.04	0.80	0.00	0.00	0.00	1.00	0.04	0.24	0.52	0.20	0.00	0.44	0.36	0.20	0.03	0.37	0.43	0.00	0.00	0.10	0.70	0.03
	HV	(F, Ng)	0.00	0.00	0.00	0.60	0.40	0.00	0.00	0.60	0.00	0.04	0.32	0.64	0.00	0.00	0.28	0.72	0.10	0.13	0.30	0.23	0.07	0.10	0.47	0.20
		(Ng/F)	0.00	0.00	0.00	0.60	0.40	0.00	0.00	0.60	0.00	0.12	0.20	0.68	0.00	0.04	0.24	0.72	0.07	0.10	0.33	0.33	0.03	0.17	0.43	0.20
		(F/Ng)	0.00	0.00	0.00	0.60	0.40	0.00	0.00	0.60	0.00	0.32	0.28	0.40	0.00	0.08	0.44	0.48	0.07	0.23	0.43	0.03	0.13	0.23	0.37	0.07
	HVR	(F, Ng)	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.28	0.72	0.00	0.00	0.28	0.72	0.00	0.00	0.13	0.70	0.00	0.00	0.10	0.73
		(Ng/F)	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.04	0.96	0.00	0.00	0.07	0.77	0.00	0.00	0.30	0.77
		(F/Ng)	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.40	0.60	0.00	0.00	0.08	0.92	0.00	0.17	0.30	0.37	0.00	0.03	0.30	0.50
	Rot_180°	(F, Ng)	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.04	0.36	0.60	0.00	0.04	0.12	0.84	0.03	0.40	0.33	0.00	0.03	0.37	0.43	0.00
		(Ng/F)	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.04	0.16	0.80	0.00	0.08	0.12	0.80	0.07	0.30	0.43	0.00	0.03	0.33	0.40	0.07
		(F/Ng)	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.12	0.48	0.40	0.00	0.08	0.28	0.64	0.10	0.57	0.13	0.00	0.07	0.50	0.27	0.00
	Rot_90°	(F, Ng)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0.17	0.33	0.13	0.17	0.10	0.33	0.13	0.27
		(Ng/F)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0.10	0.33	0.27	0.13	0.03	0.23	0.30	0.27
		(F/Ng)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0.27	0.27	0.20	0.07	0.13	0.50	0.13	0.07

Tableau V.14 : Résultats de l’algorithme génétique sur les formes symétriques binaires 2D.

Pour les formes générées avec le système d’interaction de Von Neumann :

- Avec le système d’interaction de Von Neumann, les résultats du codage avec prise en compte de la symétrie sont meilleurs que ceux obtenus sans prise en compte de la symétrie sauf pour les formes avec la symétrie horizontale et verticale.
- Avec le système d’interaction de Moore, dans tous les cas, les résultats obtenus avec le codage avec la symétrie sont meilleurs que ceux du codage sans symétrie quel que soit le type de symétrie. les meilleurs pourcentages de solutions optimales sont obtenus avec les formes possédant une symétrie horizontale, verticale et rotationnelle et les formes symétriques avec une rotation de 180°.

- Pour les formes générées avec le système d’interaction de Moore :
 - Le codage qui exploite la symétrie donne de meilleurs résultats que le codage qui n’exploite pas la symétrie.
 - Avec le codage avec symétrie, l’approche (Ng/F) a donné de meilleurs résultats que les deux autres approches.

Les plus grands pourcentages sont obtenus pour les formes avec une symétrie (HVR).

Le Tableau V.15 résume les résultats de l’algorithme génétique sur les formes 2D binaires symétriques. Avec le codage exploitant la symétrie, l’approche (Ng/F) permet d’obtenir de meilleurs résultats en particulier avec le système d’interaction de Moore.

RLI		Formes générées par Von Neumann																Formes générées par Moore							
		Von Neumann								Moore								Moore							
		IM				ID				IM				ID				IM				ID			
		[90%, 95%]	[95%, 98%]	[98%, 100%]	100%	[90%, 95%]	[95%, 98%]	[98%, 100%]	100%	[90%, 95%]	[95%, 98%]	[98%, 100%]	100%	[90%, 95%]	[95%, 98%]	[98%, 100%]	100%	[90%, 95%]	[95%, 98%]	[98%, 100%]	100%				
Sans symétrie	(F, Ng)	0.04	0.08	0.00	0.77	0.05	0.00	0.00	0.95	0.14	0.39	0.43	0.04	0.09	0.43	0.33	0.13	0.10	0.26	0.15	0.00	0.21	0.30	0.28	0.01
	(Ng/F)	0.06	0.09	0.01	0.78	0.02	0.02	0.00	0.97	0.16	0.43	0.30	0.10	0.03	0.48	0.42	0.07	0.18	0.37	0.14	0.01	0.21	0.38	0.19	0.00
	(F/Ng)	0.06	0.14	0.02	0.73	0.02	0.00	0.00	0.98	0.17	0.47	0.28	0.07	0.00	0.42	0.40	0.18	0.30	0.39	0.09	0.00	0.16	0.44	0.19	0.00
Avec symétrie	(F, Ng)	0.00	0.00	0.00	0.92	0.08	0.00	0.00	0.92	0.02	0.13	0.38	0.47	0.00	0.08	0.37	0.55	0.09	0.25	0.26	0.19	0.06	0.18	0.34	0.26
	(Ng/F)	0.00	0.01	0.02	0.90	0.08	0.00	0.00	0.92	0.01	0.17	0.24	0.58	0.00	0.08	0.33	0.59	0.06	0.21	0.34	0.22	0.02	0.19	0.37	0.25
	(F/Ng)	0.02	0.01	0.01	0.88	0.08	0.00	0.00	0.92	0.01	0.25	0.41	0.34	0.00	0.22	0.30	0.48	0.10	0.33	0.31	0.08	0.08	0.26	0.38	0.11

Tableau V.15 : Résultats de l’algorithme génétique sur les formes symétriques binaires 2D groupés par approche.

V.6.2 Recherche locale itérée

Dans cette partie, chacune des formes est testée en utilisant :

- les deux types de codages : sans prise en compte de la symétrie et avec prise en compte de la symétrie ;
- les trois approches : approche (F, Ng), approche (Ng/F) et approche (F/Ng) ;
- les deux critères : information mutuelle (IM) et nombre de cellules identiques (ID).
- les trois systèmes de voisinage : Γ_1 , Γ_2 et Γ_M avec $M = 10$.

Nous présentons les résultats pour chaque approche en fonction du type de symétrie.

V.6.2.1 Approche (F, Ng)

Nous présentons dans le Tableau V.16 les résultats de la recherche locale itérée en utilisant l’approche (F, Ng).

- Pour les formes générées avec Von Neumann :
 - Avec le système d’interaction de Von Neumann, les résultats du codage avec la symétrie sont meilleurs que ceux du codage sans symétrie à l’exception des formes avec une symétrie horizontale et verticale.
 - Avec le système d’interaction de Moore, dans tous les cas, les résultats obtenus avec le codage avec symétrie sont meilleurs que ceux du codage sans symétrie quel que soit le type de symétrie. Le système de voisinage Γ_1 a donné dans la majorité des cas, de meilleurs résultats que les deux autres systèmes de voisinage.

Pour les formes de Moore, avec le codage avec symétrie, le système de voisinage Γ_2 a donné de meilleurs résultats que les deux autres systèmes de voisinage.

		Formes générées avec Von Neumann								Formes générées avec Moore																
		Von Neumann				Moore				Moore																
		IM		ID		IM		ID		IM		ID														
		[90%, 95%]	[95%, 98%]	[98%, 100%]	100%	[90%, 95%]	[95%, 98%]	[98%, 100%]	100%	[90%, 95%]	[95%, 98%]	[98%, 100%]	100%													
Codage sans symétrie	H	Γ_1	0.00	0.12	0.00	0.68	0.00	0.00	0.00	1.00	0.24	0.28	0.44	0.04	0.00	0.52	0.28	0.20	0.07	0.37	0.23	0.00	0.20	0.17	0.47	0.00
		Γ_2	0.00	0.04	0.00	0.80	0.00	0.00	0.00	1.00	0.24	0.48	0.24	0.04	0.00	0.52	0.44	0.04	0.13	0.43	0.13	0.00	0.27	0.27	0.30	0.00
		Γ_3	0.00	0.20	0.08	0.64	0.00	0.00	0.00	1.00	0.28	0.32	0.28	0.08	0.00	0.52	0.36	0.12	0.20	0.40	0.17	0.00	0.13	0.37	0.33	0.00
	V	Γ_1	0.00	0.12	0.00	0.68	0.00	0.00	0.00	1.00	0.24	0.28	0.44	0.04	0.04	0.40	0.36	0.20	0.07	0.37	0.23	0.00	0.03	0.33	0.43	0.03
		Γ_2	0.16	0.24	0.04	0.56	0.00	0.00	0.00	1.00	0.36	0.24	0.16	0.20	0.00	0.56	0.24	0.20	0.10	0.47	0.27	0.00	0.00	0.53	0.30	0.00
		Γ_3	0.24	0.20	0.04	0.52	0.00	0.00	0.00	1.00	0.20	0.52	0.08	0.20	0.00	0.48	0.32	0.20	0.10	0.63	0.10	0.00	0.00	0.57	0.27	0.00
	HV	Γ_1	0.08	0.12	0.00	0.64	0.08	0.00	0.00	0.92	0.12	0.56	0.32	0.00	0.24	0.40	0.24	0.00	0.17	0.17	0.17	0.00	0.23	0.23	0.37	0.00
		Γ_2	0.12	0.08	0.00	0.64	0.08	0.00	0.00	0.92	0.12	0.52	0.28	0.08	0.16	0.48	0.32	0.04	0.10	0.30	0.20	0.03	0.17	0.30	0.30	0.00
		Γ_3	0.04	0.16	0.00	0.68	0.12	0.00	0.00	0.88	0.28	0.56	0.16	0.00	0.00	0.44	0.40	0.16	0.37	0.30	0.17	0.00	0.13	0.33	0.30	0.00
	HVR	Γ_1	0.12	0.04	0.00	0.84	0.16	0.00	0.00	0.84	0.08	0.72	0.20	0.00	0.04	0.60	0.36	0.00	0.07	0.13	0.13	0.00	0.37	0.47	0.00	0.00
		Γ_2	0.04	0.08	0.00	0.88	0.00	0.08	0.00	0.92	0.08	0.84	0.08	0.00	0.00	0.64	0.36	0.00	0.17	0.43	0.07	0.00	0.40	0.43	0.00	0.00
		Γ_3	0.04	0.16	0.00	0.80	0.00	0.00	0.00	1.00	0.08	0.76	0.16	0.00	0.00	0.56	0.44	0.00	0.53	0.30	0.00	0.00	0.37	0.47	0.00	0.00
	Rot_180°	Γ_1	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.12	0.76	0.12	0.12	0.24	0.40	0.24	0.17	0.37	0.13	0.00	0.17	0.37	0.30	0.00
		Γ_2	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.08	0.76	0.16	0.00	0.20	0.72	0.08	0.33	0.40	0.10	0.00	0.13	0.43	0.23	0.00
		Γ_3	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.20	0.72	0.08	0.00	0.12	0.48	0.40	0.27	0.50	0.07	0.00	0.10	0.50	0.23	0.00
	Rot_90°	Γ_1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0.03	0.17	0.03	0.00	0.23	0.23	0.10	0.00
		Γ_2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0.27	0.17	0.07	0.00	0.27	0.30	0.03	0.00
		Γ_3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0.33	0.20	0.00	0.00	0.23	0.37	0.00	0.00
Codage avec symétrie	H	Γ_1	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.04	0.28	0.48	0.20	0.00	0.24	0.56	0.20	0.13	0.33	0.33	0.03	0.17	0.17	0.40	0.10
		Γ_2	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.04	0.32	0.44	0.20	0.00	0.16	0.64	0.20	0.10	0.27	0.43	0.00	0.03	0.30	0.43	0.07
		Γ_3	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.56	0.36	0.08	0.00	0.52	0.32	0.16	0.10	0.37	0.37	0.00	0.13	0.20	0.50	0.00
	V	Γ_1	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.04	0.28	0.48	0.20	0.00	0.12	0.60	0.28	0.13	0.33	0.33	0.03	0.00	0.10	0.50	0.23
		Γ_2	0.00	0.04	0.08	0.88	0.00	0.00	0.00	1.00	0.00	0.36	0.40	0.24	0.00	0.12	0.60	0.28	0.00	0.23	0.50	0.10	0.00	0.10	0.60	0.13
		Γ_3	0.12	0.04	0.04	0.80	0.00	0.00	0.00	1.00	0.04	0.24	0.52	0.20	0.00	0.44	0.36	0.20	0.03	0.37	0.43	0.00	0.00	0.10	0.70	0.03
	HV	Γ_1	0.00	0.00	0.00	0.60	0.40	0.00	0.00	0.60	0.00	0.04	0.32	0.64	0.00	0.00	0.28	0.72	0.10	0.13	0.30	0.23	0.07	0.10	0.47	0.20
		Γ_2	0.00	0.00	0.00	0.60	0.40	0.00	0.00	0.60	0.00	0.12	0.20	0.68	0.00	0.04	0.24	0.72	0.07	0.10	0.33	0.33	0.03	0.17	0.43	0.20
		Γ_3	0.00	0.00	0.00	0.60	0.40	0.00	0.00	0.60	0.00	0.32	0.28	0.40	0.00	0.08	0.44	0.48	0.07	0.23	0.43	0.03	0.13	0.23	0.37	0.07
	HVR	Γ_1	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.28	0.72	0.00	0.00	0.28	0.72	0.00	0.00	0.13	0.70	0.00	0.00	0.10	0.73
		Γ_2	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.04	0.96	0.00	0.00	0.07	0.77	0.00	0.00	0.07	0.77
		Γ_3	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.40	0.60	0.00	0.00	0.08	0.92	0.00	0.17	0.30	0.37	0.00	0.03	0.30	0.50
	Rot_180°	Γ_1	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.04	0.36	0.60	0.00	0.04	0.12	0.84	0.03	0.40	0.33	0.00	0.03	0.37	0.43	0.00
		Γ_2	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.04	0.16	0.80	0.00	0.08	0.12	0.80	0.07	0.30	0.43	0.00	0.03	0.33	0.40	0.07
		Γ_3	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.12	0.48	0.40	0.00	0.08	0.28	0.64	0.10	0.57	0.13	0.00	0.07	0.50	0.27	0.00
	Rot_90°	Γ_1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0.17	0.33	0.13	0.17	0.10	0.33	0.13	0.27
		Γ_2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0.10	0.33	0.27	0.13	0.03	0.23	0.30	0.27
		Γ_3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0.27	0.27	0.20	0.07	0.13	0.50	0.13	0.07

Tableau V.16 : Résultats de la recherche locale itérée sur les formes symétriques binaires 2D avec l'approche (F, Ng).

V.6.2.2 Approche (Ng/F)

Nous présentons dans le Tableau V.17 les résultats de la recherche locale itérée en utilisant l'approche (Ng/F).

- Pour les formes générées avec Von Neumann :
 - Avec le système d'interaction de Von Neumann, la solution optimale est toujours atteinte avec les deux codages (sans et avec symétrie).
 - Avec le système d'interaction de Moore, dans tous les cas, les résultats obtenus avec le codage avec symétrie sont meilleurs que ceux du codage sans symétrie quel que soit le type de symétrie. Les meilleurs résultats ont été obtenus avec le système de voisinage Γ_2 pour les deux types de codages.
- Pour les formes de Moore :

- Avec le codage avec symétrie, le système de voisinage Γ_2 a donné de meilleurs résultats que les deux autres systèmes de voisinage dans la majorité des cas.

Les meilleurs pourcentages sont obtenus avec les formes ayant une symétrie horizontale, verticale et rotationnelle.

			Formes générées avec Von Neumann												Formes générées avec Moore											
			Von Neumann						Moore						Moore				Moore							
			IM			ID			IM			ID			IM		ID		IM		ID					
			[90%, 95%]	[95%, 98%]	[98%, 100%]	100%	[90%, 95%]	[95%, 98%]	[98%, 100%]	100%	[90%, 95%]	[95%, 98%]	[98%, 100%]	100%	[90%, 95%]	[95%, 98%]	[98%, 100%]	100%	[90%, 95%]	[95%, 98%]	[98%, 100%]	100%				
Codage sans symétrie	H	Γ_1	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.56	0.44	0.00	0.00	0.44	0.56	0.00	0.30	0.20	0.33	0.00	0.23	0.27	0.33	0.00
		Γ_2	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.36	0.64	0.00	0.00	0.40	0.56	0.04	0.20	0.27	0.37	0.00	0.27	0.23	0.33	0.00
		Γ_3	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.08	0.52	0.36	0.04	0.04	0.60	0.32	0.04	0.27	0.23	0.33	0.00	0.33	0.17	0.33	0.00
	V	Γ_1	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.56	0.44	0.00	0.00	0.68	0.28	0.04	0.30	0.20	0.33	0.00	0.00	0.47	0.37	0.00
		Γ_2	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.08	0.56	0.28	0.08	0.04	0.52	0.36	0.04	0.00	0.47	0.37	0.00	0.00	0.43	0.37	0.03
		Γ_3	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.68	0.32	0.00	0.00	0.68	0.16	0.08	0.00	0.67	0.17	0.00	0.00	0.60	0.23	0.00
	HV	Γ_1	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.80	0.20	0.00	0.08	0.92	0.00	0.00	0.03	0.60	0.03	0.00	0.10	0.50	0.13	0.00
		Γ_2	0.00	0.00	0.00	0.96	0.00	0.00	0.00	1.00	0.04	0.68	0.24	0.04	0.56	0.32	0.04	0.00	0.13	0.50	0.07	0.00	0.13	0.50	0.10	0.00
		Γ_3	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.04	0.64	0.16	0.04	0.44	0.48	0.00	0.00	0.23	0.37	0.07	0.00	0.20	0.40	0.07	0.00
	HVR	Γ_1	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.48	0.52	0.00	0.00	0.48	0.52	0.00	0.27	0.57	0.00	0.00	0.20	0.63	0.00	0.00
		Γ_2	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.32	0.68	0.00	0.12	0.36	0.52	0.00	0.43	0.40	0.00	0.00	0.23	0.57	0.03	0.00
		Γ_3	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.08	0.32	0.60	0.00	0.16	0.32	0.48	0.00	0.37	0.43	0.03	0.00	0.37	0.47	0.00	0.00
	Rot_180°	Γ_1	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.16	0.76	0.08	0.00	0.12	0.80	0.08	0.17	0.57	0.10	0.00	0.13	0.57	0.10	0.00
		Γ_2	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.12	0.64	0.24	0.16	0.16	0.52	0.16	0.17	0.50	0.13	0.00	0.17	0.60	0.07	0.00
		Γ_3	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.12	0.20	0.52	0.16	0.08	0.24	0.48	0.20	0.10	0.53	0.10	0.00	0.23	0.57	0.03	0.00
	Rot_90°	Γ_1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0.37	0.30	0.00	0.00	0.30	0.37	0.00	0.00
		Γ_2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0.53	0.20	0.00	0.00	0.43	0.30	0.00	0.00
		Γ_3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0.37	0.30	0.00	0.00	0.40	0.27	0.00	0.00
Codage avec symétrie	H	Γ_1	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.72	0.28	0.00	0.00	0.68	0.32	0.00	0.33	0.37	0.13	0.00	0.30	0.30	0.23
		Γ_2	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.48	0.52	0.00	0.00	0.56	0.44	0.00	0.17	0.33	0.33	0.00	0.30	0.20	0.33
		Γ_3	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.04	0.52	0.44	0.00	0.00	0.64	0.36	0.03	0.30	0.27	0.23	0.00	0.23	0.27	0.33
	V	Γ_1	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.72	0.28	0.00	0.00	0.72	0.28	0.00	0.33	0.37	0.13	0.00	0.00	0.53	0.30
		Γ_2	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.04	0.56	0.40	0.00	0.00	0.64	0.36	0.00	0.00	0.57	0.27	0.00	0.00	0.37	0.47
		Γ_3	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.08	0.60	0.32	0.00	0.04	0.64	0.32	0.00	0.03	0.53	0.27	0.00	0.00	0.47	0.37
	HV	Γ_1	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.13	0.10	0.60	0.00	0.17	0.00	0.67
		Γ_2	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.03	0.13	0.03	0.63	0.00	0.13	0.03	0.67
		Γ_3	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.07	0.07	0.07	0.63	0.03	0.13	0.03	0.63
	HVR	Γ_1	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.83	0.00	0.00	0.00	0.83
		Γ_2	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.83	0.00	0.00	0.00	0.83
		Γ_3	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.83	0.00	0.00	0.00	0.83
	Rot_180°	Γ_1	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.04	0.96	0.00	0.17	0.63	0.03	0.00	0.17	0.60	0.07
		Γ_2	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.04	0.96	0.00	0.00	0.00	1.00	0.00	0.13	0.53	0.17	0.07	0.10	0.40	0.27
		Γ_3	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.08	0.92	0.00	0.00	0.12	0.88	0.13	0.07	0.53	0.10	0.07	0.10	0.47	0.20
	Rot_90°	Γ_1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0.00	0.13	0.13	0.57	0.00	0.13	0.13	0.57
		Γ_2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0.00	0.13	0.20	0.50	0.00	0.10	0.17	0.57
		Γ_3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0.00	0.17	0.27	0.40	0.00	0.17	0.07	0.60

Tableau V.17 : Résultats de la recherche locale itérée sur les formes symétriques avec l'approche (Ng/F).

V.6.2.3 Approche (F/Ng)

Nous présentons dans le Tableau V.18 les résultats de la recherche locale itérée avec l'approche (F/Ng).

			Formes générées avec Von Neumann												Formes générées avec Moore											
			Von Neumann						Moore						Moore				Moore							
			IM			ID			IM			ID			IM		ID		IM		ID					
			[90%, 95%]	[95%, 98%]	[98%, 100%]	100%	[90%, 95%]	[95%, 98%]	[98%, 100%]	100%	[90%, 95%]	[95%, 98%]	[98%, 100%]	100%	[90%, 95%]	[95%, 98%]	[98%, 100%]	100%	[90%, 95%]	[95%, 98%]	[98%, 100%]	100%				
e sans symétrie	H	Γ_1	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.56	0.44	0.00	0.00	0.36	0.64	0.00	0.30	0.23	0.30	0.00	0.20	0.27	0.37	0.00
		Γ_2	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.48	0.52	0.00	0.00	0.16	0.80	0.04	0.20	0.27	0.37	0.00	0.20	0.27	0.37	0.00

Codage avec symétrie	V	$\Gamma 3$	0.00 0.00 0.00 1.00	0.00 0.00 0.00 1.00	0.00 0.44 0.48 0.08	0.00 0.32 0.68 0.00	0.23 0.27 0.33 0.00	0.23 0.27 0.30 0.03	
		$\Gamma 1$	0.00 0.00 0.00 1.00	0.00 0.00 0.00 1.00	0.00 0.56 0.44 0.00	0.00 0.68 0.28 0.04	0.30 0.23 0.30 0.00	0.00 0.40 0.43 0.00	
		$\Gamma 2$	0.00 0.00 0.00 1.00	0.00 0.00 0.00 1.00	0.00 0.64 0.36 0.00	0.00 0.44 0.48 0.08	0.00 0.37 0.47 0.00	0.00 0.37 0.47 0.00	
	HV	$\Gamma 3$	0.00 0.04 0.00 0.96	0.00 0.00 0.00 1.00	0.08 0.52 0.32 0.08	0.00 0.60 0.40 0.00	0.07 0.53 0.23 0.00	0.03 0.43 0.37 0.00	
		$\Gamma 1$	0.00 0.00 0.00 1.00	0.00 0.00 0.00 1.00	0.00 0.64 0.36 0.00	0.00 0.64 0.36 0.00	0.13 0.47 0.10 0.00	0.07 0.53 0.10 0.00	
		$\Gamma 2$	0.00 0.00 0.00 1.00	0.00 0.00 0.00 1.00	0.00 0.56 0.44 0.00	0.00 0.28 0.52 0.20	0.17 0.53 0.00 0.00	0.27 0.43 0.10 0.00	
	HVR	$\Gamma 3$	0.00 0.00 0.00 1.00	0.00 0.00 0.00 1.00	0.00 0.60 0.36 0.04	0.00 0.44 0.40 0.16	0.23 0.50 0.00 0.00	0.10 0.47 0.13 0.00	
		$\Gamma 1$	0.00 0.00 0.00 1.00	0.00 0.00 0.00 1.00	0.00 0.48 0.52 0.00	0.00 0.32 0.68 0.00	0.33 0.43 0.07 0.00	0.20 0.63 0.00 0.00	
		$\Gamma 2$	0.00 0.00 0.00 1.00	0.00 0.00 0.00 1.00	0.00 0.28 0.72 0.00	0.00 0.04 0.96 0.00	0.27 0.47 0.10 0.00	0.23 0.57 0.03 0.00	
	Rot_180°	$\Gamma 3$	0.00 0.00 0.00 1.00	0.00 0.00 0.00 1.00	0.00 0.32 0.68 0.00	0.00 0.12 0.88 0.00	0.33 0.33 0.17 0.00	0.33 0.50 0.00 0.00	
		$\Gamma 1$	0.00 0.00 0.00 1.00	0.00 0.00 0.00 1.00	0.00 0.20 0.68 0.12	0.00 0.08 0.72 0.20	0.17 0.57 0.07 0.00	0.13 0.67 0.00 0.00	
		$\Gamma 2$	0.00 0.00 0.00 1.00	0.00 0.00 0.00 1.00	0.00 0.16 0.52 0.32	0.00 0.04 0.44 0.52	0.10 0.57 0.07 0.00	0.10 0.60 0.07 0.00	
	Rot_90°	$\Gamma 3$	0.00 0.00 0.00 1.00	0.00 0.00 0.00 1.00	0.00 0.16 0.48 0.36	0.00 0.16 0.32 0.52	0.17 0.47 0.20 0.00	0.03 0.60 0.07 0.00	
		$\Gamma 1$	- - - -	- - - -	- - - -	- - - -	0.40 0.27 0.00 0.00	0.43 0.33 0.00 0.00	
		$\Gamma 2$	- - - -	- - - -	- - - -	- - - -	0.43 0.20 0.03 0.00	0.47 0.33 0.00 0.00	
	Codage sans symétrie	H	$\Gamma 3$	- - - -	- - - -	- - - -	- - - -	0.47 0.27 0.00 0.00	0.40 0.30 0.00 0.00
			$\Gamma 1$	0.00 0.00 0.00 1.00	0.00 0.00 0.00 1.00	0.00 0.00 0.56 0.44	0.00 0.00 0.64 0.36	0.00 0.30 0.33 0.20	0.00 0.33 0.20 0.30
			$\Gamma 2$	0.00 0.00 0.00 1.00	0.00 0.00 0.00 1.00	0.00 0.00 0.48 0.52	0.00 0.00 0.32 0.68	0.00 0.27 0.23 0.33	0.00 0.20 0.30 0.33
		V	$\Gamma 3$	0.00 0.00 0.00 1.00	0.00 0.00 0.00 1.00	0.00 0.00 0.60 0.40	0.00 0.00 0.68 0.32	0.00 0.33 0.23 0.27	0.00 0.30 0.27 0.27
			$\Gamma 1$	0.00 0.00 0.00 1.00	0.00 0.00 0.00 1.00	0.00 0.00 0.56 0.44	0.00 0.00 0.80 0.20	0.00 0.30 0.33 0.20	0.00 0.00 0.53 0.30
			$\Gamma 2$	0.00 0.00 0.00 1.00	0.00 0.00 0.00 1.00	0.00 0.00 0.60 0.40	0.00 0.00 0.60 0.40	0.00 0.00 0.50 0.33	0.00 0.00 0.33 0.50
		HV	$\Gamma 3$	0.00 0.00 0.00 1.00	0.00 0.00 0.00 1.00	0.00 0.00 0.72 0.28	0.00 0.00 0.56 0.44	0.00 0.00 0.53 0.30	0.00 0.00 0.43 0.40
			$\Gamma 1$	0.00 0.00 0.00 1.00	0.00 0.13 0.07 0.63	0.00 0.13 0.03 0.67			
			$\Gamma 2$	0.00 0.00 0.00 1.00	0.00 0.17 0.00 0.67	0.00 0.13 0.03 0.67			
HVR		$\Gamma 3$	0.00 0.00 0.00 1.00	0.03 0.13 0.03 0.63	0.00 0.13 0.10 0.60				
		$\Gamma 1$	0.00 0.00 0.00 1.00	0.00 0.00 0.00 0.83	0.00 0.00 0.00 0.83				
		$\Gamma 2$	0.00 0.00 0.00 1.00	0.00 0.00 0.00 0.83	0.00 0.00 0.00 0.83				
Rot_180°		$\Gamma 3$	0.00 0.00 0.00 1.00	0.00 0.00 0.83	0.00 0.00 0.00 0.83				
		$\Gamma 1$	0.00 0.00 0.00 1.00	0.00 0.00 0.00 1.00	0.00 0.00 0.04 0.96	0.00 0.00 0.00 1.00	0.07 0.10 0.50 0.17	0.03 0.13 0.43 0.23	
		$\Gamma 2$	0.00 0.00 0.00 1.00	0.03 0.17 0.37 0.27	0.07 0.07 0.50 0.20				
Rot_90°		$\Gamma 3$	0.00 0.00 0.00 1.00	0.00 0.00 0.00 1.00	0.00 0.00 0.04 0.96	0.00 0.00 0.04 0.96	0.13 0.07 0.53 0.10	0.03 0.20 0.43 0.17	
		$\Gamma 1$	- - - -	- - - -	- - - -	- - - -	0.00 0.03 0.40 0.40	0.00 0.10 0.17 0.57	
		$\Gamma 2$	- - - -	- - - -	- - - -	- - - -	0.00 0.07 0.33 0.43	0.00 0.10 0.17 0.57	
Rot_90°		$\Gamma 3$	- - - -	- - - -	- - - -	- - - -	0.00 0.10 0.27 0.47	0.00 0.17 0.23 0.43	

Tableau V.18 : Résultats de la recherche locale itérée sur les formes symétriques avec l'approche (F/Ng).

- Pour les formes générées avec Von Neumann :
 - Avec le système d'interaction de Von Neumann, la solution optimale est toujours atteinte avec le codage exploitant la symétrie.
 - Avec le système d'interaction de Moore, la recherche locale itérée retrouve pratiquement toutes les solutions optimales des formes avec symétrie horizontale et verticale ou une symétrie horizontale, verticale et rotationnelle ou une rotation à 180°. Les meilleurs résultats ont été obtenus avec le système de voisinage $\Gamma 2$ pour les deux types de codages.
- Pour les formes de Moore :
 - Le système de voisinage $\Gamma 2$ a donné de meilleurs résultats que les deux autres systèmes de voisinage.

Les meilleurs pourcentages sont obtenus avec les formes ayant une symétrie horizontale, verticale et rotationnelle.

V.6.3 Recuit simulé

Dans cette partie, les résultats du recuit simulé sont présentés.

V.6.3.1 Approche (F, Ng)

Nous présentons dans le Tableau V.19 les résultats du recuit simulé en utilisant l'approche (F, Ng).

- Pour les formes générées avec le système d'interaction de Von Neumann :
 - Avec le système d'interaction de Von Neumann, la solution optimale est toujours atteinte avec le codage exploitant la symétrie.
 - Avec le système d'interaction de Moore, les meilleurs résultats du recuit simulé ont été obtenus avec les systèmes de voisinage Γ_2 et Γ_3 pour les deux types de codages.
- Pour les formes générées avec le système d'interaction de Moore :
 - Le système de voisinage Γ_2 a donné de meilleurs résultats que les deux autres systèmes de voisinage dans la plupart des cas.

		Formes générées avec Von Neumann												Formes générées avec Moore												
		Von Neumann						Moore						Moore												
		IM			ID			IM			ID			IM				ID								
		[90%, 95%]	[95%, 98%]	[98%, 100%]	100%	[90%, 95%]	[95%, 98%]	[98%, 100%]	100%	[90%, 95%]	[95%, 98%]	[98%, 100%]	100%	[90%, 95%]	[95%, 98%]	[98%, 100%]	100%	[90%, 95%]	[95%, 98%]	[98%, 100%]	100%					
Codage sans symétrie	H	Γ_1	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.52	0.48	0.00	0.04	0.08	0.72	0.16	0.07	0.27	0.43	0.00	0.03	0.27	0.47	0.07
		Γ_2	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.24	0.60	0.16	0.00	0.04	0.88	0.08	0.13	0.37	0.30	0.03	0.00	0.23	0.43	0.17
		Γ_3	0.00	0.04	0.00	0.96	0.00	0.00	0.00	1.00	0.00	0.24	0.60	0.16	0.00	0.08	0.84	0.08	0.23	0.13	0.43	0.03	0.00	0.37	0.40	0.07
	V	Γ_1	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.52	0.48	0.00	0.00	0.08	0.44	0.48	0.07	0.27	0.43	0.00	0.00	0.07	0.60	0.17
		Γ_2	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.04	0.28	0.44	0.24	0.00	0.04	0.60	0.36	0.00	0.20	0.57	0.07	0.00	0.03	0.67	0.13
		Γ_3	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.04	0.28	0.36	0.32	0.00	0.08	0.52	0.40	0.00	0.30	0.50	0.03	0.00	0.03	0.67	0.13
	HV	Γ_1	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.04	0.20	0.60	0.16	0.36	0.04	0.44	0.16	0.23	0.20	0.27	0.07	0.07	0.20	0.43	0.13
		Γ_2	0.00	0.00	0.00	0.96	0.00	0.00	0.00	1.00	0.00	0.32	0.52	0.16	0.20	0.04	0.44	0.32	0.07	0.30	0.23	0.10	0.10	0.13	0.33	0.27
		Γ_3	0.04	0.08	0.00	0.84	0.00	0.04	0.00	0.96	0.04	0.20	0.60	0.16	0.16	0.00	0.60	0.24	0.03	0.33	0.33	0.03	0.17	0.13	0.30	0.23
	HVR	Γ_1	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.08	0.92	0.00	0.04	0.04	0.84	0.08	0.13	0.23	0.30	0.00	0.27	0.33	0.23	0.00
		Γ_2	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.04	0.16	0.76	0.04	0.04	0.04	0.88	0.04	0.13	0.17	0.30	0.07	0.10	0.50	0.23	0.00
		Γ_3	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.28	0.68	0.04	0.04	0.04	0.84	0.08	0.23	0.10	0.33	0.03	0.17	0.40	0.27	0.00
	Rot_180°	Γ_1	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.16	0.24	0.60	0.00	0.12	0.28	0.60	0.03	0.37	0.43	0.00	0.00	0.10	0.57	0.17
		Γ_2	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.08	0.28	0.64	0.04	0.12	0.24	0.60	0.07	0.37	0.40	0.00	0.07	0.17	0.57	0.03
		Γ_3	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.08	0.40	0.52	0.00	0.04	0.24	0.72	0.07	0.27	0.47	0.03	0.07	0.10	0.57	0.10
	Rot_90°	Γ_1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0.23	0.33	0.17	0.00	0.33	0.13	0.33	0.03
		Γ_2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0.23	0.23	0.27	0.00	0.13	0.30	0.23	0.10
		Γ_3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0.40	0.17	0.10	0.00	0.20	0.27	0.33	0.03
Codage avec symétrie	H	Γ_1	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.72	0.28	0.00	0.00	0.64	0.36	0.00	0.27	0.37	0.20	0.00	0.23	0.33	0.27
		Γ_2	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.64	0.36	0.00	0.00	0.60	0.40	0.03	0.27	0.13	0.40	0.03	0.20	0.27	0.33
		Γ_3	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.64	0.36	0.00	0.00	0.60	0.40	0.03	0.23	0.20	0.37	0.00	0.27	0.20	0.37
	V	Γ_1	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.72	0.28	0.00	0.00	0.20	0.80	0.00	0.27	0.37	0.20	0.00	0.00	0.43	0.40
		Γ_2	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.04	0.32	0.64	0.00	0.04	0.32	0.64	0.00	0.00	0.43	0.40	0.00	0.00	0.23	0.60
		Γ_3	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.04	0.36	0.60	0.00	0.00	0.40	0.60	0.00	0.00	0.40	0.43	0.00	0.03	0.13	0.67
	HV	Γ_1	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.24	0.76	0.00	0.00	0.04	0.96	0.00	0.13	0.13	0.57	0.00	0.07	0.10	0.67
		Γ_2	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.08	0.92	0.00	0.00	0.00	1.00	0.00	0.07	0.10	0.67	0.00	0.13	0.03	0.67
		Γ_3	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.04	0.96	0.00	0.00	0.00	1.00	0.00	0.10	0.13	0.60	0.00	0.07	0.10	0.67
	HVR	Γ_1	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.83	0.00	0.00	0.00	0.83
		Γ_2	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.83	0.00	0.00	0.00	0.83
		Γ_3	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.83	0.00	0.00	0.00	0.83
	Rot_180°	Γ_1	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.20	0.80	0.00	0.00	0.16	0.84	0.03	0.17	0.47	0.17	0.03	0.07	0.40	0.33
		Γ_2	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.16	0.84	0.00	0.00	0.20	0.80	0.00	0.07	0.47	0.30	0.00	0.07	0.33	0.43
		Γ_3	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.20	0.80	0.00	0.00	0.12	0.88	0.00	0.10	0.40	0.33	0.03	0.03	0.47	0.30
	Rot_90°	Γ_1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0.00	0.13	0.17	0.53	0.00	0.07	0.20	0.57
		Γ_2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0.00	0.07	0.13	0.63	0.00	0.17	0.10	0.57
		Γ_3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0.00	0.03	0.30	0.50	0.03	0.03	0.17	0.60

Tableau V.19 : Résultats du recuit simulé sur les formes symétriques avec l'approche (F, Ng).

V.6.3.2 Approche (Ng/F)

Le Tableau V.20 donne les résultats du recuit simulé obtenus pour chaque type de symétrie, codage (sans symétrie et avec symétrie) et système de voisinage Γ_1 , Γ_2 et Γ_3 .

- Pour les formes générées avec Von Neumann :

- Avec le système d'interaction de Von Neumann, la solution optimale est toujours atteinte avec le codage exploitant la symétrie.
- Avec le système d'interaction de Moore, les meilleurs résultats du recuit simulé ont été obtenus avec les systèmes de voisinage Γ_2 et Γ_3 pour les deux types de codages.
- Pour les formes de Moore :
- Les systèmes de voisinage Γ_1 et Γ_2 ont donné les meilleurs résultats dans la plupart des cas.

		Formes générées avec Von Neumann												Formes générées avec Moore												
		Von Neumann						Moore						Moore												
		IM			ID			IM			ID			IM				ID								
		90%, 95%]	95%, 98%]	98%, 100%]	100%	90%, 95%]	95%, 98%]	98%, 100%]	100%	90%, 95%]	95%, 98%]	98%, 100%]	100%	90%, 95%]	95%, 98%]	98%, 100%]	100%	90%, 95%]	95%, 98%]	98%, 100%]	100%					
Codage sans symétrie	H	Γ_1	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.12	0.32	0.40	0.16	0.00	0.00	0.92	0.08	0.13	0.13	0.53	0.00	0.00	0.33	0.37	0.13
		Γ_2	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.08	0.32	0.44	0.16	0.00	0.08	0.80	0.12	0.13	0.17	0.47	0.03	0.00	0.23	0.60	0.00
		Γ_3	0.00	0.00	0.04	0.96	0.00	0.00	0.00	1.00	0.00	0.28	0.52	0.20	0.00	0.00	0.84	0.16	0.13	0.20	0.47	0.03	0.03	0.17	0.57	0.07
	V	Γ_1	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.12	0.32	0.40	0.16	0.00	0.00	0.68	0.32	0.13	0.13	0.53	0.00	0.00	0.07	0.60	0.17
		Γ_2	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.08	0.20	0.48	0.24	0.00	0.12	0.48	0.40	0.03	0.20	0.47	0.13	0.00	0.00	0.67	0.17
		Γ_3	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.08	0.28	0.40	0.24	0.00	0.08	0.52	0.40	0.00	0.13	0.63	0.07	0.00	0.03	0.70	0.10
	HV	Γ_1	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.32	0.56	0.12	0.16	0.20	0.52	0.12	0.13	0.37	0.17	0.03	0.13	0.03	0.50	0.17
		Γ_2	0.00	0.00	0.00	0.96	0.00	0.04	0.00	0.96	0.00	0.28	0.44	0.28	0.12	0.12	0.56	0.20	0.23	0.17	0.27	0.03	0.13	0.13	0.27	0.30
		Γ_3	0.00	0.04	0.00	0.96	0.00	0.04	0.00	0.96	0.00	0.32	0.28	0.40	0.08	0.00	0.52	0.40	0.07	0.23	0.40	0.00	0.07	0.23	0.40	0.13
	HVR	Γ_1	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.04	0.28	0.68	0.00	0.00	0.00	0.92	0.08	0.07	0.13	0.30	0.07	0.20	0.30	0.30	0.03
		Γ_2	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.24	0.68	0.08	0.00	0.00	0.88	0.12	0.13	0.30	0.23	0.07	0.27	0.27	0.27	0.03
		Γ_3	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.12	0.88	0.00	0.04	0.00	0.80	0.16	0.13	0.20	0.33	0.00	0.13	0.20	0.47	0.03
Rot_180°	Γ_1	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.12	0.44	0.44	0.00	0.04	0.20	0.76	0.07	0.50	0.27	0.00	0.03	0.13	0.57	0.10	
	Γ_2	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.08	0.28	0.64	0.00	0.00	0.12	0.88	0.00	0.37	0.43	0.03	0.00	0.17	0.60	0.07	
	Γ_3	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.12	0.28	0.60	0.00	0.00	0.24	0.76	0.00	0.23	0.47	0.13	0.03	0.07	0.57	0.17	
Rot_90°	Γ_1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0.37	0.27	0.13	0.00	0.30	0.13	0.33	0.07	
	Γ_2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0.33	0.23	0.17	0.00	0.20	0.27	0.37	0.00	
	Γ_3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0.23	0.33	0.20	0.00	0.20	0.27	0.30	0.07	
Codage avec symétrie	H	Γ_1	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.68	0.32	0.00	0.00	0.56	0.44	0.03	0.27	0.23	0.30	0.00	0.23	0.33	0.27
		Γ_2	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.44	0.56	0.00	0.00	0.68	0.32	0.00	0.23	0.27	0.33	0.00	0.17	0.33	0.33
		Γ_3	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.64	0.36	0.00	0.00	0.48	0.52	0.00	0.13	0.43	0.27	0.00	0.27	0.27	0.30
	V	Γ_1	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.68	0.32	0.00	0.00	0.20	0.80	0.03	0.27	0.23	0.30	0.00	0.00	0.23	0.60
		Γ_2	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.04	0.32	0.64	0.00	0.04	0.28	0.68	0.00	0.00	0.50	0.33	0.00	0.00	0.17	0.67
		Γ_3	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.28	0.72	0.00	0.00	0.28	0.72	0.00	0.00	0.43	0.40	0.00	0.00	0.13	0.70
	HV	Γ_1	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.12	0.88	0.00	0.00	0.00	1.00	0.00	0.13	0.10	0.60	0.00	0.17	0.03	0.63
		Γ_2	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.04	0.96	0.00	0.00	0.00	1.00	0.03	0.00	0.17	0.63	0.00	0.10	0.10	0.63
		Γ_3	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.04	0.96	0.00	0.00	0.04	0.96	0.03	0.07	0.07	0.67	0.00	0.13	0.03	0.67
	HVR	Γ_1	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.83	0.00	0.00	0.00	0.83
		Γ_2	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.83	0.00	0.00	0.00	0.83
		Γ_3	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.83	0.00	0.00	0.00	0.83
Rot_180°	Γ_1	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.16	0.84	0.00	0.00	0.16	0.84	0.00	0.17	0.43	0.23	0.00	0.13	0.40	0.30	
	Γ_2	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.12	0.88	0.00	0.00	0.08	0.92	0.00	0.10	0.27	0.47	0.00	0.10	0.30	0.43	
	Γ_3	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.16	0.84	0.00	0.00	0.16	0.84	0.03	0.07	0.37	0.37	0.00	0.03	0.33	0.47	
Rot_90°	Γ_1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0.00	0.13	0.23	0.47	0.00	0.03	0.20	0.60	
	Γ_2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0.00	0.07	0.20	0.57	0.00	0.10	0.17	0.57	
	Γ_3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0.00	0.03	0.23	0.57	0.00	0.07	0.07	0.70	

Tableau V.20 : Résultats du recuit simulé sur les formes symétriques avec l'approche (Ng/F).

V.6.3.3 Approche (F/Ng)

Nous présentons dans le Tableau V.21 les résultats du recuit simulé en utilisant l'approche (F/Ng).

- Pour les formes générées avec Von Neumann :
- Avec le système d'interaction de Von Neumann, la solution optimale est toujours atteinte avec le codage exploitant la symétrie.

- Avec le système d'interaction de Moore, les meilleurs résultats du recuit simulé ont été obtenus avec les systèmes de voisinage Γ_2 et Γ_3 pour les deux types de codages.
- Pour les formes de Moore :
- Les systèmes de voisinage Γ_2 et Γ_3 ont donné de meilleurs résultats que le système de voisinage Γ_1 .

		Formes générées avec Von Neumann																Formes générées avec Moore								
		Von Neumann								Moore								Moore								
		IM				ID				IM				ID				IM				ID				
		[90%, 95%[[95%, 98%[[98%, 100%[100%	[90%, 95%[[95%, 98%[[98%, 100%[100%	[90%, 95%[[95%, 98%[[98%, 100%[100%	[90%, 95%[[95%, 98%[[98%, 100%[100%	[90%, 95%[[95%, 98%[[98%, 100%[100%					
Codage sans symétrie	H	Γ_1	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.04	0.32	0.52	0.12	0.00	0.08	0.76	0.16	0.03	0.40	0.37	0.03	0.00	0.37	0.40	0.07
		Γ_2	0.00	0.08	0.16	0.76	0.00	0.16	0.32	0.52	0.08	0.28	0.56	0.08	0.00	0.04	0.76	0.20	0.00	0.43	0.33	0.07	0.00	0.23	0.60	0.00
		Γ_3	0.00	0.04	0.00	0.96	0.00	0.00	0.00	1.00	0.00	0.40	0.48	0.12	0.00	0.00	0.88	0.12	0.00	0.33	0.47	0.03	0.00	0.30	0.40	0.13
	V	Γ_1	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.04	0.32	0.52	0.12	0.00	0.04	0.60	0.36	0.03	0.40	0.37	0.03	0.00	0.13	0.63	0.07
		Γ_2	0.04	0.00	0.40	0.56	0.00	0.00	0.28	0.72	0.04	0.28	0.36	0.32	0.00	0.04	0.56	0.40	0.00	0.17	0.57	0.10	0.00	0.07	0.60	0.17
		Γ_3	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.04	0.16	0.56	0.24	0.00	0.12	0.48	0.40	0.00	0.30	0.50	0.03	0.00	0.00	0.57	0.27
	HV	Γ_1	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.36	0.52	0.12	0.00	0.08	0.44	0.48	0.20	0.20	0.20	0.13	0.13	0.10	0.43	0.17
		Γ_2	0.00	0.12	0.12	0.72	0.00	0.16	0.12	0.72	0.00	0.16	0.72	0.12	0.00	0.04	0.52	0.44	0.10	0.17	0.40	0.07	0.13	0.10	0.33	0.27
		Γ_3	0.00	0.04	0.00	0.96	0.00	0.00	0.00	1.00	0.00	0.24	0.52	0.24	0.00	0.04	0.48	0.48	0.07	0.40	0.17	0.10	0.13	0.17	0.43	0.10
	HVR	Γ_1	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.04	0.36	0.60	0.00	0.00	0.00	0.92	0.08	0.07	0.47	0.27	0.03	0.20	0.20	0.43	0.00
		Γ_2	0.00	0.08	0.16	0.76	0.00	0.12	0.28	0.60	0.00	0.28	0.64	0.08	0.00	0.00	0.76	0.24	0.07	0.37	0.33	0.03	0.13	0.27	0.40	0.03
		Γ_3	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.04	0.96	0.00	0.00	0.00	0.72	0.28	0.10	0.30	0.33	0.07	0.17	0.30	0.37	0.00
	Rot_180°	Γ_1	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.12	0.48	0.40	0.00	0.04	0.24	0.72	0.07	0.33	0.37	0.07	0.03	0.07	0.67	0.07
		Γ_2	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.48	0.52	0.00	0.04	0.20	0.76	0.00	0.37	0.47	0.00	0.03	0.13	0.47	0.20
		Γ_3	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.08	0.24	0.68	0.00	0.04	0.20	0.76	0.00	0.27	0.47	0.03	0.03	0.10	0.57	0.13
	Rot_90°	Γ_1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0.30	0.30	0.23	0.00	0.17	0.33	0.33	0.00
		Γ_2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0.27	0.40	0.10	0.00	0.17	0.27	0.23	0.13
		Γ_3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0.30	0.40	0.07	0.00	0.17	0.30	0.27	0.07
Codage avec symétrie	H	Γ_1	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.56	0.44	0.00	0.00	0.72	0.28	0.00	0.20	0.40	0.23	0.00	0.20	0.33	0.30
		Γ_2	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.52	0.48	0.00	0.00	0.68	0.32	0.00	0.20	0.33	0.30	0.00	0.23	0.27	0.33
		Γ_3	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.72	0.28	0.00	0.00	0.48	0.52	0.00	0.30	0.30	0.23	0.00	0.10	0.33	0.40
	V	Γ_1	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.56	0.44	0.00	0.00	0.36	0.64	0.00	0.20	0.40	0.23	0.00	0.00	0.17	0.67
		Γ_2	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.04	0.32	0.64	0.00	0.00	0.28	0.72	0.00	0.00	0.53	0.30	0.00	0.00	0.37	0.47
		Γ_3	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.04	0.32	0.64	0.00	0.00	0.36	0.64	0.00	0.03	0.37	0.43	0.00	0.00	0.27	0.57
	HV	Γ_1	0.00	0.00	0.00	1.00	0.00	0.10	0.10	0.63	0.00	0.10	0.13	0.60												
		Γ_2	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.04	0.96	0.00	0.00	0.00	1.00	0.00	0.13	0.03	0.67	0.00	0.10	0.07	0.67
		Γ_3	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.08	0.92	0.00	0.00	0.00	1.00	0.00	0.00	0.17	0.67	0.00	0.07	0.10	0.67
	HVR	Γ_1	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.83	0.00	0.00	0.00	0.83												
		Γ_2	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.83	0.00	0.00	0.00	0.83												
		Γ_3	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.83	0.00	0.00	0.00	0.83												
	Rot_180°	Γ_1	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.20	0.80	0.00	0.00	0.12	0.88	0.00	0.20	0.43	0.20	0.03	0.03	0.40	0.37
		Γ_2	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.20	0.80	0.00	0.00	0.12	0.88	0.00	0.10	0.40	0.33	0.00	0.03	0.27	0.53
		Γ_3	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.20	0.80	0.00	0.00	0.16	0.84	0.03	0.03	0.47	0.30	0.00	0.10	0.40	0.33
	Rot_90°	Γ_1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0.00	0.17	0.20	0.47	0.00	0.03	0.13	0.67
		Γ_2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0.00	0.03	0.20	0.60	0.00	0.10	0.13	0.60
		Γ_3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0.00	0.07	0.20	0.57	0.00	0.00	0.27	0.57

Tableau V.21 : Résultats du recuit simulé sur les formes symétriques avec l'approche (F/Ng).

V.6.4 Résultats groupés par approches

Dans le Tableau V.22 , les résultats de la recherche locale itérée et du recuit simulé regroupés par approche et type de codage sont présentés. Avec la recherche locale itérée et le recuit simulé, les résultats du codage exploitant la symétrie sont meilleurs que ceux du codage sans symétrie quel que soit le critère et l'approche utilisée.

Les meilleurs résultats de la recherche locale itérée sont obtenus avec l'approche (F/Ng) avec ou sans prise en compte de la symétrie. Pour le recuit simulé, avec le codage sans symétrie, les

meilleurs résultats sont obtenus avec l'approche (Ng/F) et (F/Ng). Avec la prise en compte de la symétrie, les meilleurs résultats du recuit simulé sont obtenus avec l'approche (Ng/F).

		Formes générées par Von Neumann												Formes générées par Moore												
		Von Neumann						Moore						Moore												
		IM			ID			IM			ID			IM				ID								
		[90%, 95%]	[95%, 98%]	[98%, 100%]	100%	[90%, 95%]	[95%, 98%]	[98%, 100%]	100%	[90%, 95%]	[95%, 98%]	[98%, 100%]	100%	[90%, 95%]	[95%, 98%]	[98%, 100%]	100%	[90%, 95%]	[95%, 98%]	[98%, 100%]	100%					
RLI	Sans symétrie	(F, Ng)	0.00	0.00	0.00	0.99	0.00	0.00	0.00	1.00	0.10	0.50	0.36	0.02	0.14	0.47	0.33	0.03	0.32	0.48	0.12	0.00	0.31	0.46	0.16	0.00
		(Ng/F)	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.03	0.46	0.45	0.05	0.11	0.45	0.37	0.05	0.28	0.49	0.16	0.00	0.25	0.53	0.17	0.00
	(F/Ng)	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.01	0.44	0.49	0.07	0.00	0.31	0.57	0.12	0.29	0.46	0.19	0.00	0.23	0.53	0.19	0.00	
	(F, Ng)	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.01	0.28	0.70	0.00	0.02	0.29	0.69	0.04	0.17	0.33	0.46	0.01	0.17	0.32	0.51	
RS	Avec symétrie	(Ng/F)	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.01	0.25	0.74	0.00	0.00	0.27	0.73	0.02	0.15	0.33	0.50	0.01	0.14	0.27	0.58
		(F/Ng)	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.24	0.76	0.00	0.00	0.24	0.76	0.02	0.14	0.31	0.53	0.01	0.13	0.28	0.58
	(F, Ng)	0.00	0.01	0.00	0.98	0.00	0.00	0.00	1.00	0.01	0.24	0.53	0.21	0.06	0.06	0.59	0.29	0.16	0.31	0.42	0.03	0.11	0.25	0.51	0.12	
	(Ng/F)	0.00	0.00	0.00	0.99	0.00	0.01	0.00	0.99	0.03	0.24	0.48	0.25	0.03	0.04	0.60	0.33	0.16	0.29	0.43	0.04	0.12	0.20	0.56	0.12	
RS	Avec symétrie	(F/Ng)	0.00	0.02	0.06	0.91	0.00	0.03	0.07	0.90	0.02	0.23	0.54	0.21	0.00	0.04	0.57	0.39	0.11	0.40	0.40	0.06	0.10	0.23	0.54	0.12
		(F, Ng)	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.01	0.29	0.71	0.00	0.00	0.22	0.78	0.01	0.13	0.28	0.59	0.01	0.10	0.23	0.66
	(Ng/F)	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.25	0.75	0.00	0.00	0.19	0.80	0.01	0.11	0.28	0.60	0.00	0.10	0.21	0.69	
	(F/Ng)	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.01	0.25	0.75	0.00	0.00	0.22	0.78	0.00	0.12	0.30	0.58	0.00	0.07	0.24	0.68	

Tableau V.22 : Résultats de la RLI et le RS regroupés par approche.

V.6.5 Résultats groupés par système de voisinage

Dans le Tableau V.23 les résultats de la recherche locale itérée et du recuit simulé sont regroupés par systèmes de voisinage. Avec le système d'interaction de Von Neumann, le système de voisinage Γ_1 permet d'obtenir toutes les solutions optimales quel que soit le codage, la méthode et le critère utilisé. Les meilleurs résultats de la recherche locale itérée sont obtenus avec le système de voisinage Γ_2 . Pour le recuit simulé, les meilleurs résultats sont obtenus avec Γ_2 ou Γ_3 . Ces résultats montrent que dans le cas du système d'interaction de Moore, la perturbation d'un seul élément ne suffit pas pour converger vers l'optimum global et que l'on se bloque très rapidement dans un optimum local.

		Formes générées par Von Neumann												Formes générées par Moore												
		Von Neumann						Moore						Moore												
		IM			ID			IM			ID			IM				ID								
		[90%, 95%]	[95%, 98%]	[98%, 100%]	100%	[90%, 95%]	[95%, 98%]	[98%, 100%]	100%	[90%, 95%]	[95%, 98%]	[98%, 100%]	100%	[90%, 95%]	[95%, 98%]	[98%, 100%]	100%	[90%, 95%]	[95%, 98%]	[98%, 100%]	100%					
RLI	Sans symétrie	Γ_1	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.54	0.43	0.02	0.01	0.52	0.43	0.03	0.31	0.47	0.14	0.00	0.23	0.54	0.17	0.00
		Γ_2	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.04	0.45	0.46	0.05	0.11	0.32	0.47	0.08	0.28	0.48	0.17	0.00	0.27	0.50	0.19	0.00
		Γ_3	0.00	0.01	0.00	0.99	0.00	0.00	0.00	1.00	0.09	0.41	0.40	0.06	0.13	0.39	0.37	0.08	0.30	0.48	0.15	0.00	0.28	0.48	0.15	0.00
	Avec symétrie	Γ_1	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.29	0.71	0.00	0.00	0.30	0.70	0.01	0.21	0.33	0.46	0.00	0.15	0.33	0.52
		Γ_2	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.01	0.22	0.77	0.00	0.01	0.23	0.77	0.02	0.14	0.30	0.54	0.01	0.13	0.26	0.60
		Γ_3	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.01	0.26	0.73	0.00	0.02	0.27	0.71	0.05	0.13	0.33	0.49	0.02	0.16	0.28	0.55
RS	Sans symétrie	Γ_1	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.03	0.29	0.52	0.16	0.04	0.06	0.59	0.31	0.16	0.35	0.38	0.03	0.13	0.22	0.55	0.11
		Γ_2	0.00	0.02	0.06	0.91	0.00	0.03	0.07	0.90	0.02	0.21	0.51	0.25	0.03	0.05	0.58	0.34	0.13	0.33	0.42	0.06	0.10	0.23	0.52	0.14
		Γ_3	0.00	0.02	0.00	0.98	0.00	0.01	0.00	0.99	0.01	0.21	0.52	0.26	0.02	0.03	0.58	0.36	0.13	0.31	0.44	0.04	0.10	0.23	0.54	0.12
	Avec symétrie	Γ_1	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.32	0.68	0.00	0.00	0.21	0.79	0.01	0.19	0.28	0.52	0.00	0.09	0.26	0.65
		Γ_2	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.01	0.21	0.78	0.00	0.01	0.22	0.78	0.00	0.09	0.28	0.63	0.00	0.10	0.21	0.69
		Γ_3	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.01	0.25	0.75	0.00	0.00	0.21	0.79	0.01	0.08	0.30	0.61	0.00	0.08	0.22	0.70

Tableau V.23 : Résultats de la RLI et le RS regroupés par système de voisinage.

V.6.6 Temps de calcul

Le codage exploitant la symétrie donne non seulement de meilleurs résultats que le codage sans symétrie mais il permet aussi de réduire considérablement le temps de calcul.

V.6.6.1 Temps de calcul moyen

Le temps de calcul moyen a été calculé pour chaque forme. Il représente la moyenne des temps des cinq réplifications de chaque forme. A partir de ces temps moyens de chaque forme, le temps moyen par type de symétrie est calculé. Il représente la moyenne des temps moyens de calcul des cinq formes de même type de symétrie.

Le Tableau V.24 donne les temps de calcul moyens par type de symétrie en seconde pour les formes générées avec le système d'interaction de Von Neumann. Pour chaque type de symétrie, le temps moyen minimal (Min) et le temps moyen maximal (Max) sont donnés pour chaque critère. Le temps de l'énumération avec le codage exploitant la symétrie est également donné.

	Formes générées avec Von Neumann																	
	Von Neumann								Moore									
	Avec symétrie				Sans symétrie				Avec symétrie				Sans symétrie					
	Enumération		IM		ID		IM		ID		IM		ID		IM		ID	
	IM	ID	min	max	min	max	min	max	min	max	min	max	min	max	min	max	min	max
H	1479.50	1305.44	852	1662	4	234	9	1095	66	99	1513	1976	463	1828	1178	2008	237	1788
v	1353.55	1219.24	2	1250	0	210	0	1092	3	119	1295	1934	231	1773	1133	2004	38	1786
HV	19.48	18.63	2	1649	0	1502	0	189	0	21	1391	1922	268	1735	7	1993	4	1752
HVR	0.22	0.20	663	1202	1	275	0	3	0	1	1579	1942	598	1756	36	1319	0	267
R_180	81.70	71.47	167	1816	2	211	0	146	1	9	1406	1867	361	1723	1088	1976	170	1728

Tableau V.24 : Temps de calcul moyen par type de symétrie pour les formes générées avec le système d'interaction de Von Neumann.

Le Tableau V.25 donne les temps de calcul moyens pour les formes générées avec le système d'interaction de Moore.

Formes générées avec Moore							
Moore							
Sans symétrie				Avec symétrie			
IM		ID		IM		ID	
min	max	min	max	min	max	min	max
1496	3661	534	1807	1341	2003	296	1807
1529	2707	517	1656	1343	1917	337	1640
1578	1995	537	1777	1251	1989	142	1757
1615	1973	518	1714	14	1253	1	163
1666	2018	548	1776	1406	2018	354	1766
1858	2002	641	1743	1208	2022	85	1758

Tableau V.25 : Temps de calcul moyen par type de symétrie pour les formes générées avec Moore.

V.6.6.2 Facteur de réduction du temps de calcul

Dans cette partie, nous présentons les facteurs de réduction du temps du recuit simulé avec l'approche (F, Ng) et (Ng/F).

a.Approche (F, Ng)

Le Tableau V.26 donne les facteurs de réduction de temps. Le facteur de réduction de temps est obtenu en divisant le temps de calcul moyen obtenu lorsque nous utilisons le codage sans prise en compte de la symétrie sur le temps de calcul moyen obtenu avec le codage exploitant la symétrie. Lorsque ce facteur est supérieur à 1, il signifie que le temps de calcul est réduit en utilisant le codage exploitant la symétrie et vice versa, s'il est inférieur à 1, l'algorithme devient plus long en exploitant la symétrie. Ce facteur dépend des formes utilisées (générées avec Von Neumann ou Moore), du type de symétrie dans la forme recherchée, du système d'interaction et du critère utilisé pour chercher la forme désirée.

Un facteur de 1.82 signifie que si le recuit simulé met 100s pour retrouver une forme en exploitant la symétrie, il met 182s pour le retrouver avec le codage sans symétrie.

		Von Neumann		Moore	
		IM	ID	IM	ID
Formes générées avec le système de Von Neumann	Symétrie horizontale	8.76	9.01	1.39	1.82
	Symétrie verticale	28.23	5.31	1.55	3.33
	Symétrie horizontale et verticale	87.42	50.49	9.42	10.18
	Symétrie axiale et rotationnelle	721.88	686.66	14.56	64.29
	Rotation à 180°	31.10	53.88	1.79	2.16
Formes générées avec le système de Moore	Symétrie horizontale	-	-	1.37	1.75
	Symétrie verticale	-	-	1.65	2.48
	Symétrie horizontale et verticale	-	-	2.56	3.78
	Symétrie axiale et rotationnelle	-	-	24.42	174.48
	Rotation à 180°	-	-	1.45	1.93
	Rotation à 90°	-	-	3.48	4.38

Tableau V.26 : Facteurs de réduction de temps.

Ce tableau montre une réduction du temps à chaque type de symétrie en particulier les formes générées avec le système d’interaction de Von Neumann. Dans le cas des formes générées avec le système d’interaction de Moore, ces facteurs dépendent du type de symétrie, ils sont plus importants dans le cas de la symétrie axiale et rotationnelle et plus faibles pour les autres types de symétries.

b.Approche (Ng/F)

Le Tableau V.27 donne les facteurs de réduction de temps. Le facteur de réduction de temps varie en fonction du critère, du système d’interaction, du type de symétrie identifié dans la forme recherchée et du système d’interaction utilisé.

		Von Neumann		Moore	
		IM	ID	IM	ID
Formes générées avec le système de Von Neumann	Symétrie horizontale	11.06	12.15	1.45	1.27
	Symétrie verticale	13.16	19.76	1.67	1.53
	Symétrie horizontale et verticale	1140.39	958.63	11.64	17.41
	Symétrie axiale et rotationnelle	2262.62	1908.40	88.46	405.64
	Rotation à 180°	503.63	205.64	1.56	1.31
Formes générées avec le système de Moore	Symétrie horizontale	-	-	1.28	1.92
	Symétrie verticale	-	-	1.63	2.37
	Symétrie horizontale et verticale	-	-	2.33	2.89
	Symétrie axiale et rotationnelle	-	-	97.94	193.54
	Rotation à 180°	-	-	1.37	1.71
	Rotation à 90°	-	-	2.59	4.21

Tableau V.27 : Facteurs de réduction de temps.

Le temps de calcul en exploitant la symétrie est réduit pour chaque type de symétrie.

V.7 Résultats des formes binaires 3D

Comme le recuit simulé avec l’approche (Ng/F) est la méthode qui a donné les meilleurs résultats pour les formes 2D (avec ou sans symétrie), nous avons testé uniquement cette méthode pour la recherche de ces formes 3D. Le critère considéré est le nombre de cellules identiques.

Le recuit simulé est lancé cinq fois pour chaque forme. Les résultats obtenus sont présentés dans le Tableau V.28. Pour chaque forme, le minimum (Min), le maximum (Max), la moyenne (Moy) et l’écart-type (SD) ainsi que la valeur optimale (IDObj) du critère nombre de cellules identiques sont donnés.

L'information mutuelle arrive à retrouver des formes identiques à celle recherchée, mais avec des états différents (Figure V.13). L'information mutuelle est donc maximale mais le nombre de cellules identiques est faible.

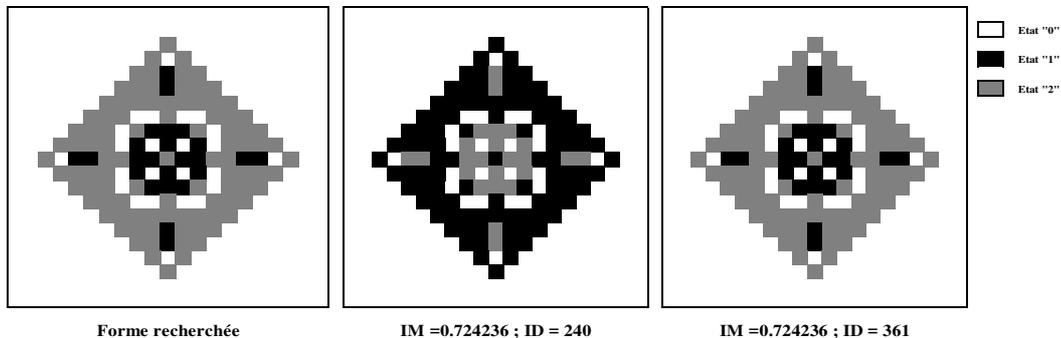


Figure V.13 : Formes à 3 états obtenues par l'information mutuelle.

L'information mutuelle permet de déterminer la fonction de transition qui donne une forme qui est la même que celle recherchée mais qui ne possède pas les mêmes états. Ce qui rend l'utilisation de ce critère difficile lors de la modélisation du processus de morphogénèse où les états cellulaires doivent être respectés dans la formation de la structure globale de l'organe. Pour pouvoir utiliser l'information mutuelle dans ce cas, il va falloir permuter les états de la fonction de transition et faire une correspondance entre les règles d'évolution locales.

V.9 Autres applications

Dans cette section, les méthodes proposées pour le problème inverse de génération de formes par automates cellulaires sont appliquées aux cas de la segmentation d'images en particulier la détection des contours.

La segmentation d'image par automates cellulaires et métaheuristiques consiste à déterminer la fonction de transition de l'automate cellulaire permettant de déterminer les contours des différents objets de cette image. Un objet représente une zone homogène de l'image.

Pour déterminer une fonction de transition de l'automate cellulaire permettant de déterminer les contours des objets, il va falloir une configuration initiale de l'automate cellulaire qui représente l'image à segmenter et une configuration finale de l'automate que l'on cherche à obtenir. Cette dernière représente l'image segmentée de l'image de la configuration initiale (Batouche, et al., 2006).

Pour la segmentation d'images, seul le recuit simulé avec l'approche (Ng/F) a été utilisé. Nous avons préconisé le système d'interaction de Moore, car en général les filtres utilisés pour la segmentation d'images prennent en compte toutes les directions locales. Les deux critères de similarité (information mutuelle et nombre de cellules identiques) ont été utilisés. Comme système de voisinage, nous avons considéré le système Γ_1 .

Logiquement, l'automate cellulaire devrait permettre d'obtenir l'image segmentée après une seule application de la fonction de transition sur l'image de départ. La configuration obtenue (image segmentée) à la première génération n'évoluera pas au cours du temps.

Deux formes simples contenant un seul objet et celles de leurs contours ont été générées (voir Figure V.14). Leurs tailles est de 21*21. Les contours de ces images ont été réalisés manuellement sans filtre particulier. Le recuit simulé a trouvé dans les deux cas la fonction de transition optimale permettant de segmenter correctement ces deux images.

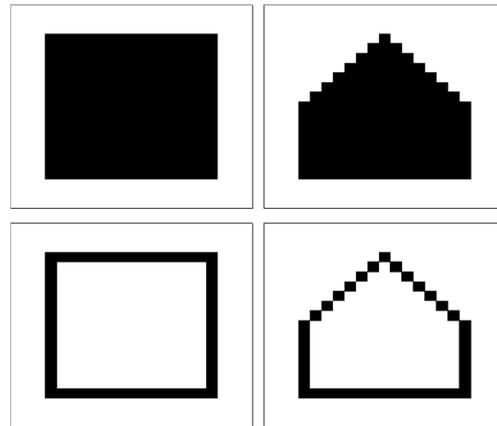


Figure V.14 : Images simples avec un seule objet et leurs contours.

Nous avons ensuite recherché d'autres images binaires contenant plusieurs objets qui ont été testées (voir Figure V.15). Ces images sont binaires de taille 128*128. Les contours de ces images ont été obtenus en utilisant le filtre de Canny (Canny, 1986; Cocquerez & Philipp, 1997).

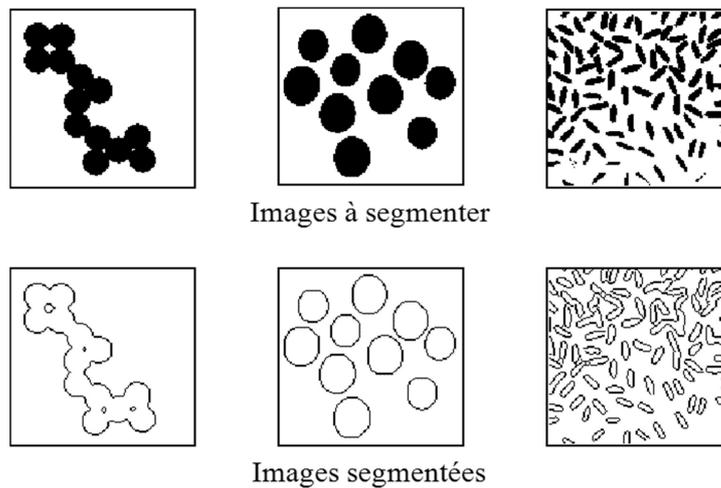


Figure V.15 : Images à segmenter et images de leurs contours.

Le recuit simulé a été utilisé pour déterminer la fonction de transition permettant de segmenter chacune de ces formes. La Figure V.16 donne les résultats obtenus pour les images segmentées par l'automate cellulaire.

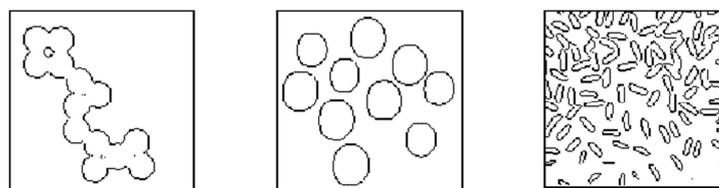


Figure V.16 : Images segmentées par automate cellulaire.

Le Tableau V.30 donne les valeurs du ratio R pour chacune de ces images. Ce ratio représente le rapport du nombre de cellules identiques (obtenu entre l'image segmentée par le filtre de Canny et celle obtenue par l'automate cellulaire) et le nombre de cellules identique optimal (nombre de pixel dans l'image).

	Information mutuelle	Nombre de cellules identiques
Image 1	99.39%	99.39%
Image 2	99.50%	99.50%
Image 3	93.40%	93.41%

Tableau V.30 : Résultats des formes à plusieurs objets.

A la fin de la segmentation, nous avons utilisé la fonction de transition obtenue par l'automate cellulaire permettant de segmenter la seconde image, et elle a été appliquée directement pour segmenter les deux autres images. Cette fonction de transition a permis de bien déterminer les contours des objets des deux autres images.

Ces résultats montrent que les méthodes proposées sont aussi efficaces pour la résolution du problème de segmentation d'images. Pour les trois images, le ratio R est supérieur à 99%. A partir d'un ensemble de tests sur un ensemble d'images comprenant plusieurs objets, une seule fonction de transition peut être extraite et utiliser pour segmenter n'importe quelle autre image.

Conclusion

Dans ce chapitre, les méthodes proposées pour la résolution du problème inverse des automates cellulaires sont appliquées pour générer différentes formes (quelconques et symétriques, binaire et à 3 états, 2D et 3D).

Une étude montre que les métaheuristiques à voisinage peuvent être utilisées pour résoudre ce problème inverse des automates cellulaires.

Les critères de similarité choisis pour comparer les formes sont le nombre de cellules identiques et l'information mutuelle.

Les résultats des méthodes proposées sont satisfaisants. Ils peuvent être comparés selon :

Les systèmes d'interaction

Sur l'ensemble des résultats, le système d'interaction de Von Neumann donne de meilleurs résultats que Moore sur les formes générées par Von Neumann. Ces résultats peuvent s'expliquer par la taille de l'espace de recherche qui est beaucoup plus importante dans le cas du système d'interaction de Moore.

Les formes générées par le système de Moore sont obtenues uniquement avec Moore.

Les approches proposées

Des résultats semblables sont obtenus avec les deux approches sur les formes générées par Von Neumann. Sur les formes générées par le système de Moore, l'approche (Ng/F) donne de meilleurs résultats dans la majorité des cas avec le système d'interaction de Moore.

Avec la recherche locale itérée, l'approche (F/Ng) a donné dans la plupart des cas de meilleurs résultats que les deux autres approches (F, Ng) et (Ng/F).

Les systèmes de voisinage

L'efficacité des systèmes de voisinage proposés dépend de la méthode utilisée, de l'approche, du critère et du système d'interaction.

Les méthodes

Les heuristiques ont donné des résultats satisfaisants. L'algorithme glouton a donné de meilleurs résultats que les recherches locales déterministes. La recherche locale déterministe avec le

premier voisin a donné de meilleurs résultats que la recherche locale avec le meilleur voisin. L'hybridation de l'algorithme glouton et de la recherche locale déterministe avec le premier voisin a permis d'améliorer les résultats de ces deux heuristiques.

Les métaheuristiques ont donné de meilleurs résultats que les heuristiques. Avec les métaheuristiques, les résultats obtenus avec le système d'interaction de Von Neumann sont similaires pour la plupart des méthodes. Les méthodes basées sur un individu ont permis d'obtenir de meilleurs résultats que les algorithmes génétiques. Le recuit simulé a donné, sur la plupart des formes testées, de meilleurs résultats que la recherche locale itérée.

Exploitation des propriétés des formes symétriques

Le codage exploitant la symétrie donne non seulement de meilleurs résultats que le codage sans symétrie, mais il permet aussi de réduire considérablement le temps de calcul.

Nombre d'états

Les formes binaires sont plus faciles à retrouver que les formes à 3 états. Ceci est dû à la taille de l'espace de recherche, qui est beaucoup plus importante en augmentant le nombre d'états.

CONCLUSION ET PERSPECTIVES

Dans ces travaux de thèse, nous nous sommes intéressées à la modélisation des systèmes biologiques afin de comprendre les processus naturels de la morphogénèse. Cette modélisation se fait à différents niveaux: macroscopique par des équations différentielles et microscopique par des automates cellulaires. Les difficultés auxquelles les scientifiques sont souvent confrontés résident dans le choix des paramètres des modèles permettant de reproduire les fonctionnements global et local du système. Ces problèmes inverses sont des problèmes d'optimisation difficile. Les problèmes traités dans cette thèse sont : l'identification paramétrique des systèmes décrits par équations différentielles et le problème inverse des automates cellulaires. Pour les résoudre, nous avons proposé des méthodes à base de métaheuristiques.

Dans le cas de la modélisation macroscopique, l'identification des paramètres est réalisée à partir des observations temporelles du système. Le modèle des équations différentielles qui décrit le système, ainsi que les conditions initiales, sont connus. Les méthodes proposées à base de recuit simulé et algorithme génétique ont été appliquées pour identifier les paramètres d'un système physique oscillant. Comme fonction objectif, nous avons proposé de minimiser la somme des écarts quadratiques entre les observations temporelles réelles et celles obtenues par le modèle. Plusieurs systèmes avec différents paramètres ont été traités. Un système de voisinage du recuit simulé perturbant tous les paramètres de la solution est proposé. Pour l'algorithme génétique, un opérateur de mutation est utilisé pour rendre réalisable toute solution non réalisable. Dans un premier temps, afin de valider les méthodes de résolution proposées, nous les avons testées sur des systèmes avec des paramètres entiers. Ces méthodes ont été ensuite utilisées pour identifier des paramètres réels. Les résultats des deux méthodes proposées sont satisfaisants, mais le recuit simulé a permis d'obtenir de meilleurs résultats que l'algorithme génétique.

Dans le cas de la modélisation microscopique par automates cellulaires, nous avons traité le problème inverse des automates cellulaires appliqué à la génération de formes. L'objectif était de déterminer la fonction de transition de l'automate cellulaire et son nombre de générations. Nous avons proposé deux codages de solutions qui ont donné trois approches. Dans la première approche nous avons proposé de déterminer simultanément la fonction de transition et le nombre de générations de l'automate cellulaire (approche (F, Ng)). Dans les deux autres approches, nous avons proposé de décomposer le problème, soit nous cherchons le nombre de générations pour chaque fonction de transition (approche (Ng/F)), soit nous déterminons la fonction de transition, connaissant le nombre de générations (approche (F/Ng)). Nous avons proposé plusieurs critères de similarité permettant de comparer les formes et nous avons retenu deux critères : l'information mutuelle et le nombre de cellules identiques. Des heuristiques (algorithme glouton, recherche locale déterministe avec le premier voisin et recherche locale déterministe avec le meilleur voisin) et des métaheuristiques (algorithme génétique, recherche locale itérée et recuit simulé) ont été proposées. Trois systèmes de voisinages (Γ_1 , Γ_2 et Γ_M) ont été testés. Ces méthodes ont été appliquées sur des formes de la littérature et sur une bibliothèque de formes que nous avons générées. Elles se distinguent les unes des autres en fonction : du type de formes (quelconques ou symétriques), du nombre d'états (binaire ou 3 états), de la dimension de l'automate cellulaire (2D ou 3D) et du système d'interaction utilisé pour la générer (Von

Neumann ou Moore). Afin de réduire la taille de l'espace de recherche, nous avons proposé un codage qui exploite les propriétés des formes. Ce codage a été appliqué au cas des formes symétriques.

Les résultats obtenus sont satisfaisants et montrent que les méthodes proposées permettent de résoudre ce problème. Une comparaison des résultats selon : l'approche utilisée, la méthode d'optimisation, le système d'interaction, le système de voisinage et les critères de similarité est donnée. En général, l'approche (Ng/F) où le nombre de générations est déterminé pour chaque fonction de transition a donné de meilleurs résultats que les deux autres approches proposées. En termes de méthodes d'optimisation, les métaheuristiques ont donné de meilleurs résultats que les heuristiques. Le recuit simulé est la méthode qui a permis d'obtenir les meilleurs résultats dans la plupart des cas. Pour les systèmes de voisinage, il nous a été difficile de déterminer celui qui fonctionne le mieux sur toutes les méthodes, car les résultats varient en fonction de la méthode utilisée, des critères de similarité, des formes et des méthodes. Dans le cas des formes symétriques, le codage exploitant la symétrie a permis d'obtenir non seulement de meilleurs résultats que ceux du codage sans prise en compte de la symétrie, mais aussi de réduire les temps de calcul. Nous avons appliqué ces méthodes pour le problème de segmentation d'images et les premiers résultats sont satisfaisants.

Dans le cadre du projet de collaboration (LIMOS, IBC), les méthodes que nous avons proposées pour la modélisation macroscopique serviront de premiers travaux pour la modélisation des systèmes biologiques. Elles peuvent être adaptées à d'autres systèmes décrits par des équations différentielles, telle que la modélisation de la pression artérielle et les déformations d'une forme biologique. Ces méthodes peuvent également être adaptées à l'identification des paramètres du modèle de croissance bactérienne permettant d'obtenir des courbes de croissance des bactéries dans différents milieux (naturel, exposé au stress radioactif). Dans le cas de la modélisation microscopique, les méthodes proposées pour la résolution du problème inverse des automates cellulaires pourraient avoir des applications non seulement dans le domaine de la génération automatique d'organes, mais aussi dans les nombreux domaines faisant intervenir des modèles à base d'automates cellulaires (sciences du vivant, physique des matériaux, trafic urbain, ...).

En termes de perspectives, plusieurs améliorations peuvent être apportées aux méthodes proposées pour la modélisation macroscopique :

- pour le recuit simulé : tester d'autres systèmes de voisinage, qui ne modifient pas tous les paramètres de la solution courante et proposer un système de voisinage avec un rayon qui décroît au cours du temps et qui diffère d'un paramètre à un autre.
- pour l'algorithme génétique : un opérateur de croisement permettant de générer uniquement des solutions admissibles peut être étudié et un couplage entre l'algorithme génétique et des méthodes de recherche locales peut être envisagé.
- envisager d'autres méthodes qui sont plus adaptées pour l'optimisation continue, telles que la méthode de gradient et les essais particuliers.

Les méthodes proposées pour la modélisation microscopique donnent des résultats satisfaisants. Pour les améliorer, nous proposons de :

- tester d'autres codages permettant de réduire la taille de l'espace de recherche. Car la taille de l'espace de recherche augmente exponentiellement avec le nombre d'états et la taille du système d'interaction.
- proposer d'autres systèmes de voisinage permettant de mieux explorer l'espace de recherche.

- utiliser d'autres méthodes, telles que les essaims particulaires et les colonies de fourmis ou des méthodes exactes tels qu'une procédure par séparation et évaluation.

Pour modéliser certains phénomènes biologiques tels que la croissance cellulaire, l'évolution des tumeurs ou la morphogénèse, il serait intéressant de :

- générer des formes en plusieurs phases d'évolution. A chaque phase, la configuration initiale, la configuration finale, la fonction de transition et le nombre de générations de l'automate cellulaire changent.
- modifier le système de mise à jour de l'automate cellulaire, car, dans la plupart des processus biologiques, les cellules n'évoluent pas en même temps. La mise à jour asynchrone n'est donc pas adéquate pour modéliser ces processus.
- combiner les méthodes proposées pour la modélisation microscopique et macroscopique pour la génération des modèles multi-échelles.

ANNEXE A. RECHERCHE LOCALE DETERMINISTE

L'Algorithme A.1 présente le principe de la recherche locale déterministe avec le meilleur voisin en utilisant le système d'interaction Γ_2 .

Algorithme A.1 : Principe de la recherche locale déterministe avec le meilleur voisin et Γ_2

```

1. Génération d'une solution initiale X de taille TSol aléatoirement
2.  $C_{Best} \leftarrow C(Sh_X, Sh^*)$ 
3.  $C_{obj} \leftarrow C(Sh^*, Sh^*)$ 
4. Amelioration  $\leftarrow$  vrai
5. Tant que ( $C_{Best} < C_{obj}$ ) et (Amelioration = vrai) faire
6.   Amelioration  $\leftarrow$  faux
7.   Pour i de 1 à TSol-1 faire
8.     Pour e1 de 0 à  $\sigma-1$  faire
9.        $X' \leftarrow X$ 
10.       $X'(i) \leftarrow e1$ 
11.      Pour j de i+1 à TSol faire
12.        Pour e2 de 0 à  $\sigma-1$  faire
13.           $X'' \leftarrow X'$ 
14.           $X''(j) \leftarrow e2$ 
15.          Si ( $C(Sh_{X''}, Sh^*) > C_{Best}$ ) alors
16.             $I_{best} \leftarrow i$ 
17.             $J_{best} \leftarrow j$ 
18.             $C_{Best} \leftarrow C(Sh_{X''}, Sh^*)$ 
19.            Amelioration  $\leftarrow$  vrai
20.             $E_{best1} \leftarrow e1$ 
21.             $E_{best2} \leftarrow e2$ 
22.          Fin si
23.        Fin Pour
24.      Fin Pour
25.    Fin Pour
26.  Fin Pour
27.  Si (Amelioration = vrai)
28.     $X(I_{best}) \leftarrow E_{best1}$ 
29.     $X(J_{best}) \leftarrow E_{best2}$ 
30.  Fin si
31. Fin tant que
32. Retourner X

```

L'Algorithme A.2 présente le principe de la recherche locale déterministe avec le premier voisin en utilisant le système d'interaction Γ_2 .

Algorithme A.2 : Principe de la recherche déterministe avec le premier voisin avec $\Gamma 2$.

```

1. Génération aléatoire d'une solution X
2. Evaluation de la solution X et calcul de  $C(\text{Sh}_X, \text{Sh}^*)$ 
3.  $C_{\text{obj}} \leftarrow C(\text{Sh}^*, \text{Sh}^*)$ 
4. Amelioration  $\leftarrow$  vrai
5. Tant que ( $C(\text{Sh}_X, \text{Sh}^*) < C_{\text{obj}}$ ) et (Amelioration = vrai) faire
6.      $i \leftarrow 1$ 
7.     Amelioration  $\leftarrow$  faux
8.     Tant que ( $(i < \text{TSol}-1)$  et (Amelioration = faux)) faire
9.          $e1 \leftarrow 0$ 
10.        Tant que ( $(e1 < \sigma-1)$  et (Amelioration = faux)) faire
11.             $X' \leftarrow X$ 
12.             $X'(i) \leftarrow e1$ 
13.             $j \leftarrow i+1$ 
14.            Tant que ( $(j < \text{TSol})$  et (Amelioration = faux)) faire
15.                 $X'' \leftarrow X'$ 
16.                 $e2 \leftarrow 0$ 
17.                Tant que ( $(e2 < \sigma-1)$  et (Amelioration = faux)) faire
18.                     $X'' \leftarrow e2$ 
19.                    Si ( $C(\text{Sh}_X, \text{Sh}^*) > C(\text{Sh}_X, \text{Sh}^*)$ ) alors
20.                         $X(i) \leftarrow e1$ 
21.                         $X(j) \leftarrow e2$ 
22.                        Amelioration  $\leftarrow$  vrai
23.                    Fin si
24.                     $e2 \leftarrow e2+1$ 
25.                Fin Tant que
26.                 $j \leftarrow j+1$ 
27.            Fin tant que
28.             $e1 \leftarrow e1+1$ 
29.        Fin Tant que
30.         $i \leftarrow i+1$ 
31.    Fin tant que
32.    Retourner X

```

ANNEXE B. CODAGE PAR TYPES DE SYMETRIE

Identification du type de symétrie

La symétrie d'une forme

Pour identifier le type de symétrie dans la forme recherchée, nous utilisons l'Algorithme B.1.

Algorithme B.1 : Algorithme d'identification du type de symétrie dans la forme

1. Type de symétrie est "Non symétrique"
2. **Si** (forme symétrique horizontalement) **alors**
3. Type de symétrie est "Symétrie Horizontale"
4. **Si** (forme symétrique verticalement) **alors**
5. Type de symétrie est " Symétrie Horizontale et verticale"
6. **Si** (forme symétrique par rotation de 90°) **alors**
7. Type de symétrie est " Symétrie Horizontale et verticale et rotationnelle"
8. **Fin Si**
9. **Fin Si**
10. **Si non Si** (forme symétrique verticalement) **alors**
11. Type de symétrie est "Symétrie verticale"
12. **Si non Si** (forme symétrique par rotation de 180°) **alors**
13. Type de symétrie est "Rotation à 180°"
14. **Si** (forme symétrique par rotation de 90°) **alors**
15. Type de symétrie est "Rotation à 90°"
16. **Fin Si**
17. **Fin Si**
18. **Fin Si**
19. **Fin Si**

Les règles équivalentes par symétrie

Les règles équivalentes par symétrie dépendent du type de symétrie considéré.

Symétrie verticale

Une forme est verticalement si sa partie à droite est symétrique à sa partie à gauche. La Figure B.1 donne les combinaisons symétriques verticalement.

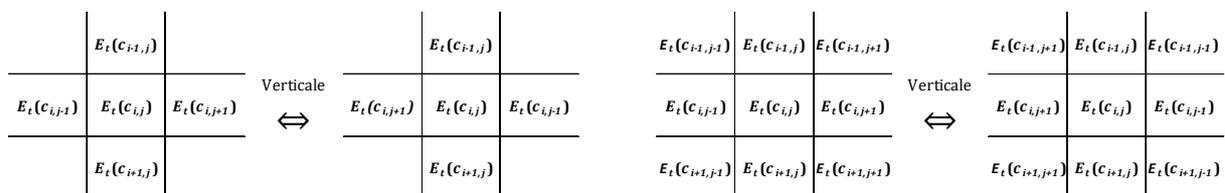


Figure B.1. Symétrie verticale avec Von Neumann (à gauche) et Moore (à droite).

La transformation T permettant de déterminer les combinaisons symétriques verticalement dans le cas de Von Neumann est donnée par:

$$T_V \left(E_t \left(c_{i,j}, c_{i-1,j}, c_{i,j-1}, c_{i+1,j}, c_{i,j+1} \right) \right) = E_t \left(c_{i,j}, c_{i-1,j}, c_{i,j+1}, c_{i+1,j}, c_{i,j-1} \right) \quad (79)$$

Avec le système de Moore la transformation T est donnée par :

$$\begin{aligned} T_V \left(E_t \left(c_{i-1,j-1}, c_{i-1,j}, c_{i-1,j+1}, c_{i,j-1}, c_{i,j}, c_{i,j+1}, c_{i+1,j-1}, c_{i+1,j}, c_{i+1,j+1} \right) \right) \\ = E_t \left(c_{i-1,j+1}, c_{i-1,j}, c_{i-1,j-1}, c_{i,j+1}, c_{i,j}, c_{i,j-1}, c_{i+1,j+1}, c_{i+1,j}, c_{i+1,j-1} \right) \end{aligned} \quad (80)$$

La Figure B.2 présente un exemple de fonction de transition simplifiée par symétrie verticale dans le cas du système d'interaction de Von Neumann.

		(0)	(1)	(4)	(2)	(3)	(6)	(5)	(7)	(8)	(9)	(12)	(10)	(11)	(14)	(13)	(15)	(16)	(17)	(20)	(18)	(19)	(22)	(21)	(23)	(24)	(25)	(28)	(26)	(27)	(30)	(29)	(31)
Etat du système d'interaction	$E_t(c_{ij})$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	$E_t(c_{i+1j})$	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
	$E_t(c_{i-1j})$	0	0	1	0	0	1	1	1	0	0	1	0	0	1	1	1	0	0	1	0	0	1	1	1	0	0	1	0	0	1	1	1
	$E_t(c_{i+1j})$	0	0	0	1	1	1	0	1	0	0	0	1	1	1	0	1	0	0	0	1	1	1	0	1	0	0	0	1	1	1	0	1
	$E_t(c_{i-1j})$	0	1	0	0	1	0	1	1	0	1	0	0	1	0	1	1	0	1	0	0	1	0	1	1	0	1	0	0	1	0	1	1

↓

Nouvelle état de c_{ij}	$E_{i+1}(c_{ij})$	0	1	1	0	1	0	1	1	1	0	0	0	1	1	1	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0
---------------------------	-------------------	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Figure B.2. Exemple de fonction de transition simplifiée par symétrie verticale avec le système d'interaction de Von Neumann.

Symétrie horizontale et verticale

Les différentes combinaisons symétriques horizontalement et verticalement sont présentées dans la Figure B.3.

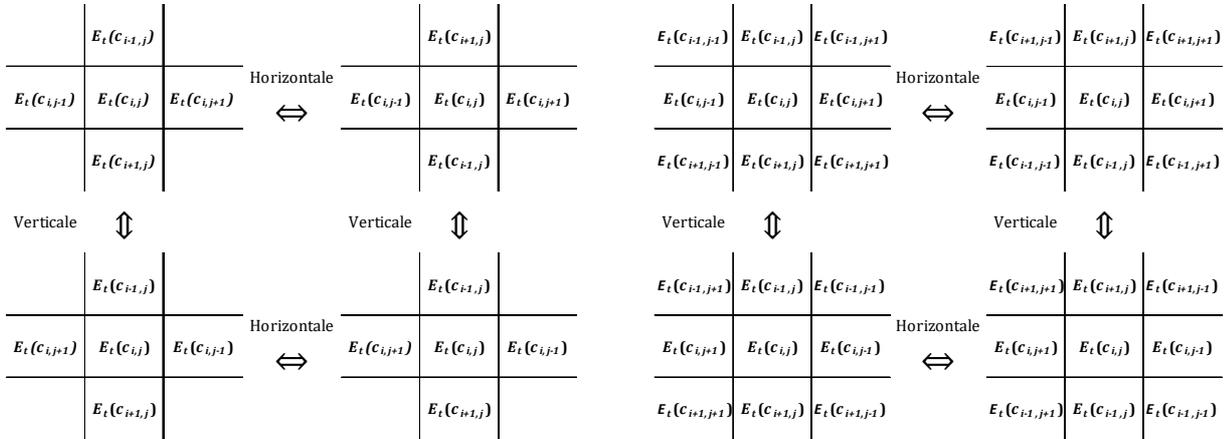


Figure B.3. Symétrie horizontale et verticale avec Von Neumann (à gauche) et Moore (à droite).

La règle d'évolution locale dans le cas de la symétrie horizontale et verticale avec Von Neumann doit respecter la condition suivante :

$$\begin{aligned} f \left(E_t \left(c_{i,j}, c_{i-1,j}, c_{i,j-1}, c_{i+1,j}, c_{i,j+1} \right) \right) &= f \left(E_t \left(c_{i,j}, c_{i+1,j}, c_{i,j-1}, c_{i-1,j}, c_{i,j+1} \right) \right) \\ &= f \left(E_t \left(c_{i,j}, c_{i-1,j}, c_{i,j+1}, c_{i+1,j}, c_{i,j-1} \right) \right) = f \left(E_t \left(c_{i,j}, c_{i+1,j}, c_{i,j+1}, c_{i-1,j}, c_{i,j-1} \right) \right) \end{aligned} \quad (81)$$

Avec le système de Moore, la même règle est appliquée aux combinaisons équivalentes par symétrie comme suit :

$$\begin{aligned}
& f \left(E_t \left(c_{i-1,j-1}, c_{i-1,j}, c_{i-1,j+1}, c_{i,j-1}, c_{i,j}, c_{i,j+1}, c_{i+1,j-1}, c_{i+1,j}, c_{i+1,j+1} \right) \right) \\
&= f \left(E_t \left(c_{i+1,j-1}, c_{i+1,j}, c_{i+1,j+1}, c_{i,j-1}, c_{i,j}, c_{i,j+1}, c_{i-1,j-1}, c_{i-1,j}, c_{i-1,j+1} \right) \right) \\
&= f \left(E_t \left(c_{i-1,j+1}, c_{i-1,j}, c_{i-1,j-1}, c_{i,j+1}, c_{i,j}, c_{i,j-1}, c_{i+1,j+1}, c_{i+1,j}, c_{i+1,j-1} \right) \right) \\
&= f \left(E_t \left(c_{i+1,j+1}, c_{i+1,j}, c_{i+1,j-1}, c_{i,j+1}, c_{i,j}, c_{i,j-1}, c_{i-1,j+1}, c_{i-1,j}, c_{i-1,j-1} \right) \right)
\end{aligned} \tag{82}$$

La Figure B.4 montre un exemple de fonction transition simplifiée symétrie horizontale et verticale pour un automate cellulaire binaire.

		(0)	(1)	(4)	(2)	(8)	(3)	(6)	(9)	(12)	(5)	(7)	(13)	(10)	(11)	(14)	(15)	(16)	(17)	(20)	(18)	(24)	(19)	(22)	(25)	(28)	(21)	(23)	(29)	(26)	(27)	(30)	(31)			
Etat du système d'interaction	$E_t(c_{ij})$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1			
	$E_t(c_{i+1,j})$	0	0	0	0	1	0	0	1	1	0	0	1	1	1	1	1	0	0	0	0	1	0	0	1	1	0	0	1	1	1	1	1			
	$E_t(c_{i,j+1})$	0	0	1	0	0	0	1	0	1	0	1	1	0	0	1	1	0	0	1	0	0	0	1	0	1	1	1	1	0	0	1	1			
	$E_t(c_{i+1,j+1})$	0	0	0	1	0	1	1	0	0	0	1	0	1	1	1	1	0	0	0	1	0	1	1	0	0	0	1	0	1	1	1	1			
	$E_t(c_{i+1,j-1})$	0	1	0	0	0	1	0	1	0	1	1	1	0	1	0	1	0	1	0	0	0	0	1	0	1	1	0	1	0	1	0	1			
		↓																																		
Nouvelle état de c_{ij}	$E_{t+1}(c_{ij})$	0	1	0		1		0		0		0		0		1		1		0		0		1				0		1		1		0		1

Figure B.4. Exemple de fonction de transition simplifiée par symétrie verticale avec le système d'interaction de Von Neumann

Rotation à 90°

Une forme a une symétrie rotationnelle si elle reste la même après une rotation avec un angle donné.

La Figure B.5 donne les combinaisons symétriques par rotation à 90° avec les systèmes d'interaction de Von Neumann et Moore.

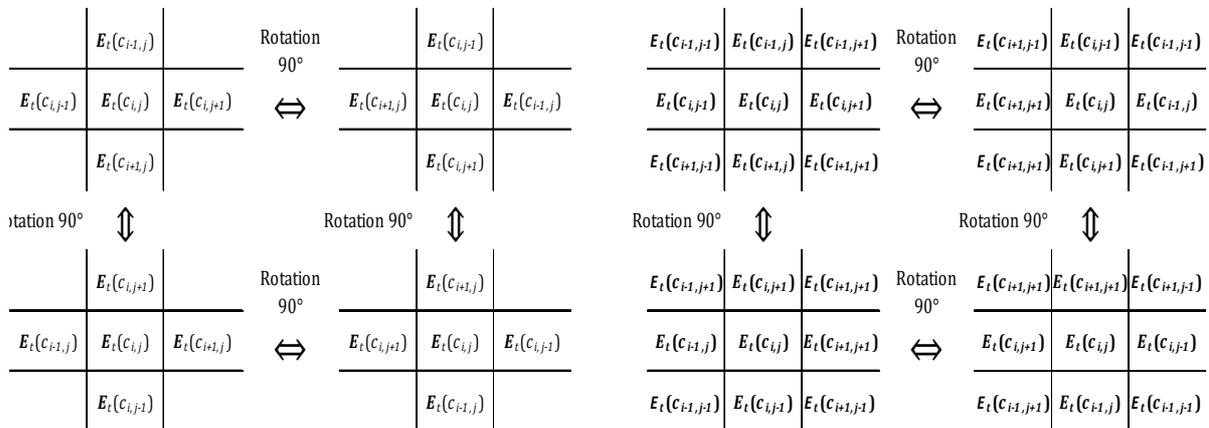


Figure B.5. Symétrie rotationnelle à 90° avec Von Neumann (à gauche) et Moore (à droite).

Les règles symétriques avec le système d'interaction de Von Neumann pour la rotation à 90° respectent les conditions suivantes :

$$\begin{aligned}
& f \left(E_t \left(c_{i,j}, c_{i-1,j}, c_{i,j-1}, c_{i+1,j}, c_{i,j+1} \right) \right) = f \left(E_t \left(c_{i,j}, c_{i,j-1}, c_{i+1,j}, c_{i,j+1}, c_{i-1,j} \right) \right) \\
&= f \left(E_t \left(c_{i,j}, c_{i+1,j}, c_{i,j+1}, c_{i-1,j}, c_{i,j-1} \right) \right) = f \left(E_t \left(c_{i,j}, c_{i,j+1}, c_{i-1,j}, c_{i,j-1}, c_{i+1,j} \right) \right)
\end{aligned} \tag{83}$$

Avec le système d'interaction de Moore la condition appliquée aux combinaisons équivalentes est la suivante :

$$\begin{aligned}
& f\left(E_t\left(c_{i-1,j-1}, c_{i-1,j}, c_{i-1,j+1}, c_{i,j-1}, c_{i,j}, c_{i,j+1}, c_{i+1,j-1}, c_{i+1,j}, c_{i+1,j+1}\right)\right) \\
&= f\left(E_t\left(c_{i+1,j-1}, c_{i,j-1}, c_{i-1,j-1}, c_{i+1,j}, c_{i,j}, c_{i-1,j}, c_{i+1,j+1}, c_{i,j+1}, c_{i-1,j+1}\right)\right) \\
&= f\left(E_t\left(c_{i+1,j+1}, c_{i+1,j}, c_{i+1,j-1}, c_{i,j+1}, c_{i,j}, c_{i,j-1}, c_{i-1,j+1}, c_{i-1,j}, c_{i-1,j-1}\right)\right) \\
&= f\left(E_t\left(c_{i-1,j+1}, c_{i,j+1}, c_{i+1,j+1}, c_{i-1,j}, c_{i,j}, c_{i+1,j}, c_{i-1,j-1}, c_{i,j-1}, c_{i+1,j-1}\right)\right)
\end{aligned} \tag{84}$$

Nous donnons dans la Figure B.6 un exemple de fonction de transition simplifiée par rotation à 90° , en utilisant le système d'interaction de Von Neumann.

		(0)	(1)	(2)	(4)	(8)	(3)	(6)	(9)	(12)	(5)	(10)	(7)	(11)	(13)	(14)	(15)	(16)	(17)	(18)	(20)	(24)	(19)	(22)	(25)	(28)	(21)	(26)	(23)	(27)	(29)	(30)	(31)	
Etat du système d'interaction	$E_t(c_{ij})$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
	$E_t(c_{i+1,j})$	0	0	0	0	1	0	0	1	1	0	1	0	1	1	1	1	1	0	0	0	0	1	0	0	1	1	0	1	0	1	1	1	1
	$E_t(c_{i,j+1})$	0	0	0	1	0	0	1	0	1	1	1	0	1	0	1	1	1	1	0	0	0	1	0	0	1	0	1	0	1	1	1	1	1
	$E_t(c_{i+1,j+1})$	0	0	1	0	0	1	1	0	0	0	0	1	1	1	0	1	1	0	1	0	0	0	1	1	1	0	0	1	1	1	0	1	1
	$E_t(c_{ij+1})$	0	1	0	0	0	1	0	1	0	1	0	1	1	1	1	0	1	0	1	0	0	0	1	0	1	0	1	0	1	1	1	0	1

↓

Nouvelle état de c_{ij}		$E_{i+1}(c_{ij})$	0	1	0	1	0	0	1	1	0	1	1	0	1	1	0	1	1	0	0	1	0	1	1	0	1	1	0	1	0	0
---------------------------	--	-------------------	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Figure B.6 : Exemple de fonction de transition simplifiée par rotation à 90° avec le système d'interaction de Von Neumann

Rotation à 180°

La Figure B.7 donne les combinaisons symétriques dans le cas d'une rotation à 180° . La forme dans ce cas est invariante par une rotation de 180° .

	$E_t(c_{i+1,j})$		Rotation 180°		$E_t(c_{i,j+1})$			$E_t(c_{i+1,j+1})$	$E_t(c_{i+1,j})$	$E_t(c_{i+1,j+1})$	Rotation 180°		$E_t(c_{i+1,j+1})$	$E_t(c_{i+1,j})$	$E_t(c_{i+1,j+1})$
$E_t(c_{i,j+1})$	$E_t(c_{ij})$	$E_t(c_{i,j+1})$	↔	$E_t(c_{i+1,j})$	$E_t(c_{ij})$	$E_t(c_{i+1,j})$		$E_t(c_{i,j+1})$	$E_t(c_{ij})$	$E_t(c_{i,j+1})$	↔	$E_t(c_{i,j+1})$	$E_t(c_{ij})$	$E_t(c_{i,j+1})$	
	$E_t(c_{i+1,j})$				$E_t(c_{i,j+1})$			$E_t(c_{i+1,j+1})$	$E_t(c_{i+1,j})$	$E_t(c_{i+1,j+1})$			$E_t(c_{i+1,j+1})$	$E_t(c_{i+1,j})$	$E_t(c_{i+1,j+1})$

Figure B.7. Symétrie rotationnelle à 180° avec Von Neumann (à gauche) et Moore (à droite).

Pour le système d'interaction de Von Neumann, la règle d'évolution locale pour une rotation à 180° , respecte la condition suivante :

$$f\left(E_t\left(c_{i,j}, c_{i-1,j}, c_{i,j-1}, c_{i+1,j}, c_{i,j+1}\right)\right) = f\left(E_t\left(c_{i,j}, c_{i+1,j}, c_{i,j+1}, c_{i-1,j}, c_{i,j-1}\right)\right) \tag{85}$$

Avec le système de Moore, elle doit vérifier la condition suivante :

$$\begin{aligned}
& f\left(E_t\left(c_{i-1,j-1}, c_{i-1,j}, c_{i-1,j+1}, c_{i,j-1}, c_{i,j}, c_{i,j+1}, c_{i+1,j-1}, c_{i+1,j}, c_{i+1,j+1}\right)\right) \\
&= f\left(E_t\left(c_{i+1,j+1}, c_{i+1,j}, c_{i+1,j-1}, c_{i,j+1}, c_{i,j}, c_{i,j-1}, c_{i-1,j+1}, c_{i-1,j}, c_{i-1,j-1}\right)\right)
\end{aligned} \tag{86}$$

Nous donnons la Figure B.8 un exemple de fonction de transition simplifiée par rotation à 180° , en utilisant le système de Von Neumann, pour un automate cellulaire binaire.

		(0)	(1)	(4)	(2)	(8)	(3)	(12)	(5)	(6)	(9)	(7)	(13)	(10)	(11)	(14)	(15)	(16)	(17)	(20)	(18)	(24)	(19)	(28)	(21)	(22)	(25)	(23)	(29)	(26)	(27)	(30)	(31)	
Etat du système d'interaction	$E_t(c_{ij})$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	$E_t(c_{i+1,j})$	0	0	0	0	1	0	1	0	0	1	0	1	1	1	1	1	0	0	0	0	1	0	1	0	0	1	0	1	1	1	1	1	1
	$E_t(c_{i,j+1})$	0	0	0	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	0	0	0	0	1	1	1	0	1	1	0	0	1	1
	$E_t(c_{i+1,j+1})$	0	1	0	0	0	1	0	1	0	1	1	1	0	1	1	1	1	0	1	0	0	1	0	1	0	0	1	0	1	1	1	1	1
		↓																																
Nouvelle état de c_{ij}	$E_{i+1}(c_{ij})$	0	1	1	0	1	1	0	1	0	1	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1		

Figure B.8: Exemple de fonction de transition simplifiée par rotation à 180° avec le système d'interaction de Von Neumann.

REFERENCES BIBLIOGRAPHIQUES

- Aarts, E. & Van Laarhoven, P., 1985. *Statistical cooling: A general approach to combinatorial optimization problems*. Philips Journal of Research, Volume 40, pp. 193-226.
- Abdellaoui, M., 2003. *Approche des systèmes distribués par automates cellulaires. Application en mécanique des milieux déformables*, Thèse de Doctorat, Perpignan: Université de Perpignan.
- Adachi, S., 2004. *The Game of Life at finite temperature*. Physica D : Nonlinear phenomena, 198(3-4), pp. 182-196.
- Adamatzky, A., 1994. *Identification of cellular Automata*. London, éd : Taylor&Francis.
- Aïboud, f., Grangeon, N. & Norre, S., 2010. *Using cellular automata and genetic algorithm to generate 2D forms*. Djerba, International Conference on Metaheuristics and Nature Inspired Computing.
- Aïboud, F., Grangeon, N. & Norre, S., 2011a. *Couplage métaheuristique à base de recuit simulé et automate cellulaire pour la génération de formes 2D*. Saint-Etienne, Société française de Recherche Opérationnelle et d'Aide à la Décision.
- Aïboud, F., Grangeon, N. & Norre, S., 2011b. *Generation of 2D Shapes with Cellular Automaton and Genetic Algorithm: Extension of Chavoya and Duthen Work*. Udine, Metaheuristics International Conference.
- Aïboud, F., Grangeon, N. & Norre, S., 2011c. *Simulated annealing based metaheuristics and binary cellular automata to generate 2D shapes*. Angers, Evolution Artificial 2011.
- Aïboud, F., Grangeon, N. & Norre, S., 2013a. *Improving efficiency of metaheuristics for cellular automaton inverse problem*. Bordeaux, Artificial Evolution 2013.
- Aïboud, F., Grangeon, N. & Norre, S., 2013b. *Couplage d'automate cellulaire et métaheuristique pour la génération de formes 2D symétriques*. Troyes, Société française de Recherche Opérationnelle et d'Aide à la Décision.
- Andre, D., Bennett, F. H. & Koza, J. R., 1996. *Discovery by genetic programming of cellular automata rule that is better than any known rule of the majority classification problem*. Cambridge, Proceedings of the First Annual Conference on Genetic Programming '96, pp. 3-11.
- Ashyraliev, M., Fomekong-Nanfack, Y., Kaandorp, J. & Blom, J. G., 2009. *Systems biology: parameter estimation for biochemical models*. FEBS journal, Volume (276), pp. 886-902.
- Augerat, P., 1995. *Approche polyédrale du problème de tournées de véhicules*, Thèse de doctorat. Grenoble: Institut Polytechnique de Grenoble.
- Bagnoli, F., Rechtman, R. & Ruffo, S., 1992. *Damage spreading and Lyapunov exponents in cellular automata*. Physics Letters, 172(A), pp. 34-38.

- Bai, L., Eyiurekli, M. & Breen, E., 2008. *Automated shape composition based on cell biology and distributed genetic programming*. Georgia, Proceedings of the 10th annual conference on genetic and evolutionary computation, pp. 1179-1186.
- Balzer, R., 1967. *An 8-state minimal time solution of the firing squad synchronisation problem*. Information and Control, Volume 10, pp. 22-42.
- Banzhaf, W., 2003. *Artificial regulatory networks and genetic programming*. Dans: Rick, Riolo & Worzel, éd. Genetic Programming Theory and Practice. Kluwer: Springer US, pp. 43-61.
- Batouche, S., Meshoul, S. & Abassene, A., 2006. *On solving edge detection by emergence*. Advanced in Applied Artificial Intelligence, Volume 4031, pp. 800-808.
- Bayes, C., 1987. *Candidates for the Game of Life in Three Dimensions*. Complex Systems, Volume 1, pp. 373-400.
- Bellman, R., 1957. *Dynamic Programming*. New Jersey, Princeton University Press.
- Berec, L., 2002. *Techniques of spatially explicit individual-based models: construction, simulation, and mean-field analysis*. Ecological Modelling, Volume 150, pp. 55-81.
- Berlekamp, E. R., Conway, J. H. & Guy, R., 1982. *Winning ways for your mathematical theory*. New York: Academic Press.
- Beyer, W., Sellers, P. & Waterman, M., 1985. Stanislaw M. *Ulam's contributions to theoretical theory*. Letters in Mathematical Physics, Volume 10, pp. 231-242.
- Bourgine, P. & Lesne, A., 2006. *Morphogénèse : L'origine des formes*. éd. : Collection Echelles.
- Breukelaar, R. & Back, T., 2005. *Using a Genetic Algorithm to Evolve Behavior in Multi Dimensional Cellular Automata: Emergence of Behavior*. Proceedings of the 2005 conference on Genetic and evolutionary computation, pp. 107-114.
- Canny, J., 1986. *Computational approach to edge detection*. IEEE Transaction on Pattern Analysis and Machine Intelligence, Volume 8, pp. 679-714.
- Cavalier, L., 2011. *Inverse problems in statistics*. Dans: Inverse Problems and High-Dimensional Estimation. Springer Berlin Heidelberg, pp. 3-96.
- Champman, S. & Jones, G. W., 2012. *Modeling growth in biological materials*. Society for Industrial and Applied Mathematics, Volume (54), pp. 52-118.
- Chandru, V. & Rao, M., 1998. *Linear programming*. CRC Press. Chapitre 31 éd. Alallah, M.J.
- Chavoya, A., 2007. *Un modèle de développement artificiel pour la génération de structures cellulaires*, Toulouse: Thèse de Doctorat en Informatique. Université de Toulouse.
- Chavoya, A., Andalon-Garcia, I. R., Lopez-Martin, C. & Meda-Campana, M., 2010. *Use of evolved artificial regulatory networks to simulate 3D cell differentiation*. Biosystems, Volume (102), pp. 41-48.
- Chavoya, A. & Duthen, Y., 2006. *Evolving Cellular Automata for 2D Form Generation*. New York, Proceedings of the Ninth International Conference on Computer Graphics and Artificial Intelligence, pp. 129-137.
- Chvatal, V., 1979. *A greedy heuristic for the set of covering problem*. Mathematics of operations research, 4(3), pp. 233-235.
- Clarkson, K., 1983. *A modification of the greedy algorithm for vertex cover*. Informations Proceeding Letters, Volume 16, pp. 23-25.

- Cocquerez, J.-P. & Philipp, S., 1997. *Analyse d'images : filtrage et segmentation d'images*. éd : Masson.
- Cormen, T. H., Leiserson, C. E., Rivest, R. L. & Stein, C., 1990. *Introduction to algorithms*. MIT Press and McGraw-Hill. 1st Edition.
- Creutz, M., 1983. *Microcanonical Monte Carlo simulation*. Physical Review Letters, 50(19), pp. 1411-1414.
- Culik, K., Hurd, L. & Yu, S., 1990a. *Computation theoretic aspects of cellular automata*. Physica D: Nonlinear phenomena, 45(1-3), pp. 357-378.
- Culik, K., Hurd, L. & Yu, S., 1990b. *Formal languages and global cellular automaton behavior*. Physica D: Nonlinear phenomena, 45(1-3), pp. 396-403.
- D'Arcy, T., 1942. *On Growth and Form*. Press Cambridge University.
- De Garis, H. & Brain Builder Group, 1999. *Artificial embryology and cellular Differentiation*. Dans: M. K. Pub, éd. Evolutionary Design by Computers. San Francisco: Editorial and Sales Office, pp. 281-295.
- Deutsch, A. & Dormann, S., 2005. *Cellular automaton modelling biological pattern formation. Characterization, Application and analysis*. Boston. éd: Berkhäuser.
- Domany, E. & Kinzel, W., 1984. *Equivalence of cellular automata to Ising models and directed percolation*. Physical Review Letters, 53(4), pp. 311-314.
- Doner, J., 1988. *Cellular automata to application: Information Processing in cellular automata*. Maths Comput. Modelling, Volume 11, pp. 351-356.
- Dorigo, M., 1992. *Optimization, Learning and Natural Algorithms*, thèse de doctorat. Milano: Politecnico di Milano.
- Dréo, J., Pétrowski, A., Siarry, P. & Taillard, E., 2003. *Métaheuristiques pour l'optimisation difficile*. éd: Eyrolles.
- Dueck, G., 1993. *New optimization heuristics, the great deluge and the record to record travel*. Journal of Computational physics, Volume (104), pp. 86-92.
- Dueck, G. & Scheuer, T., 1989. *Threshold accepting*. IBM éd. Zebtrum Heidelberg.
- Eberhart, R. & Kennedy, J., 1995. *A new optimizer using particles swarm theory*. Nagoya, Proceedings of the Sixth International Symposium on Micro Machine and Human Science, pp. 39-43.
- El Yacoubi, S., Jacewicz, P. & Ammor, N., 2003. *Analyse et contrôle par automates cellulaires*. Annals of University of Craiova, Mathematics and Computer Science Series., Volume 30, pp. 210-221.
- Fatès, N., 2001. *Les automates cellulaires : vers une nouvelle épistémologie*, Rapport de DEA. Paris: Université de Paris1.
- Feldman, A. B., Chernyak, Y. B. & Cohen, R. J., 1999. *Cellular automata model of cardiac excitation waves*. Herzschrittmachertherapie und Elektrophysiologie, 10(2), pp. 92-104.
- Fleury, 1993. *Méthodes stochastiques et déterministes pour les problèmes NP-difficiles*, Clermont Ferrand: Thèse de Doctorat en Informatique. Université Blaise Pascal.
- Flocchini, P. & Cezar, V., 2001. *Radial view continuous cellular automata*. Fundamenta Informaticae, pp. 1001-1018.

- Gacs, P., Kurdyumov, G. & Levin, L., 1978. *One dimensional uniform arrays that wash out finite islands*. *Problemy Peredachi Informatsi*, Volume 12, pp. 92-98.
- Ganguly, N., Sikdar B. K., Deutsch, A., Canright, G., Chaudhuri, P. P., 2003. *A survey on cellular automata*, Technical Report, Centre for High Performance Computing, Dresden University of Technology.
- Gardner, M., 1970. *The fantastic combinations of John Conway's new solitaire game 'life'*. *Scientific American* n° 223, pp. 120-123.
- Gennemark, P. & Wedelin, D., 2009. *Benchmarks for identification of ordinary differential equations from time series data*. *Bioinformatics*, 25(6), pp. 780-786.
- Gibas, C. & Jambeck, P., 2001. *Developing Bioinformatics Computer Skills*. éd: O'Reilly Media.
- Glover, F., 1977. *Heuristics for integer programming using surrogate constraints*. *Decision Sciences*, Volume 8, pp. 156-166.
- Glover, F., 1986. *Future Paths for integer programming and links to artificial intelligence*. Volume 5, pp. 533-549.
- Glover, F., 1989. *Tabu Search*. *ORSA Journal on Computing*, Part-I, pp. 190-206.
- Goldberg, D. E., 1989. *Genetic Algorithms in Search, Optimization and Machine Learning*. Boston: Addison-Wesley Longman Publishing.
- Gordon, T. & Bentley, P., 2005. *Bias and scalability in evolutionary development of cell patterns : Characterizing the space of cell chemistry genetic regulatory networks*. Genetic and Evolutionary Computation Conference, pp. 57-66.
- Gosalvez, M., Xing, Y., Sato, K. & Nieminem, R., 2009. *Discrete and continuous cellular automata for the simulation of propagating surfaces*. *Sensors and Actuators A: Physical*, Volume 155, pp. 98-112.
- Goto, E., 1962. *A minimal time solution of the firing squad problem*, Harvard University: Course notes for Applied Mathematics 298.
- Graner, F. & Sawada, Y., 1993. *Can surface adhesion drive cell-rearrangement ? Part II: A geometrical model*. *Journal of Theoretical Biology*, Volume 164, pp. 455-476.
- Grangeon, N., 2001. *Métaheuristiques et modèles d'évaluation pour le problème du Flow-Shop Hybride hiérarchisé: Contexte déterministe et contexte stochastique*, Thèse de Doctorat en Informatique. Clermont-Ferrand, Université Blaise Pascal.
- Gray, R., 1990. *Entropy and Information Theory*. New york, Springer Verlag.
- Grefenstette, J., 1987. *Incorporating problem specific knowledge into genetic algorithms*. éd. : Morgan Kaufmann Publishers.
- Grinstein, G., Jayaprakash, C. & He, Y., 1985. *Statistical mechanics of probabilistic cellular automata*. *Physical review letters*, 55(25), pp. 2527-2530.
- Guéret, C., Prins, C. & Sevaux, M., 2000. *Programmation linéaire*. éd : Eyrolles.
- Gunji, Y.-P. & Ono, R., 2012. *Sociality of an agent during morphogenetic canalization: Asynchronous*. *BioSystems*, Volume 109, pp. 420-429.
- Hadamard, J., 1932. *Le problème de Cauchy et les équations aux dérivées partielles hyperboliques*. Herman éd. Paris: EDP Sciences.
- Hajek, B. & Sasaki, G., 1989. *Simulated annealing-to cool or not*. *Systems and Control Letters*, 12(5), pp. 443-447.

- Hansen, P., 1986. *Steepest Ascent Midest Heuristic for Combinatorial Programming*. Capri, In Proceedings of the Congress on Numerical Methods in Combinatorial Optimization.
- Hao, J.-K., Galinier, P. & Habib, M., 1999. *Métaheuristiques pour l'optimisation combinatoire et l'affectation sous contraintes*. Revue D'Intelligence Artificielle, 2(13), pp. 283-324.
- Hodgkin, A. & Huxley, A., 1952. *A quantitative description of membrane current an dits application to conduction and excitation in nerve*. Journal of Physiology, Volume (117), pp. 500-544.
- Holland, J., 1975. *Adaptation in Natural and Artificial Systems: an introductory analysis with applications to biology, control, and artificial intelligence*. Michigan Press éd. Oxford: University of Michigan Press.
- Honda, H., 1978. *Description of cellular patterns by dirichlet domains: the two dimentional case*. Journal of Theoretical Biology, Volume 72, pp. 523-543.
- Hosny, K. M., 2010. *Refined translation and scale Legendre invariants*. Pattern Recognition Letters, Volume 31, pp. 533-538.
- Ichise, Y. & Ishida, Y., 2008. *Reverse engineering of spatial patterns in cellular automata*. 13th International Symposium on Artificial Life and Robotics, pp. 172-175.
- Jaeger, J. & Reinitz, J., 2006. *On the dynamic nature of positional information*. Bioessays, 28(11), pp. 1102-1111.
- Jourdan, L., 2010. *Métaheuristiques coopératives : du déterministe au stochastique*, Université de Lille1: Habilitation à diriger des Recherches.
- Jourdan, L., Basseur, M. & Talbi, E., 2009. *Hybridizing exact methods and metaheuristics : a taxonomy*. European Journal of Operational research, 199(3), pp. 620-629.
- Kaandorp, J., 1992. *Modeling growth forms of biological objects using fractals*, Amsterdam: Thèse de doctorat. University of Amsterdam.
- Kagaris, D. & Tragoudas, S., 2001. *Von Neumann hybrid cellular automata for generating deterministic tests sequences*. On Design Automation of Electronic Systems, 6(3), pp. 308-321.
- Karalylidis, I., 1999. *Acceleration of cellular automata algorithms using genetic algorithms*. Advances in Engineering Software, pp. 419-437.
- Keller, J. B., 1976. *Inverse problems*. The American Mathematical Monthly, 83(2), pp. 107-118.
- Kennedy, J. & Eberhart, R. C., 1997. *A discrete binary version of the particule swarm algorithm*. International Conference On Systems, Man, And Cybernetics. Volume 9, pp. 4104-4108.
- Kenneth, O., Stanly, K. & Miikkulainen, R., 2003. *A Taxonomy for Artificial Embryogeny*. Artificial Life Journal, 9(2), pp. 93-130.
- Kirkpatrick, S., Gelatt, C. & Vecchi, M., 1983. *Optimization by Simulated Annealing*. Science, 220(4598), pp. 671-680.
- Korniyak, V., 2007. *Symmetric cellular automata*. Programming and computer software 33(2), pp. 87-93.
- Koza, J. R., 1993. *Discovery of Rewrite Rules in Lindenmayer Systems and State Transition Rules in Cellular Automata via Genetic Programming*. California, Symposium on Pattern Formation.

- Lebowitz, J., Maes, C. & Speer, E., 1990. *Statistical mechanics of probabilistic cellular automata*. Journal of statistical physics, Volume 59. pp 117-170.
- Lima, C. & Lobo, F., 2004. *Parameter-Less Optimization with the Extended Compact Genetic Algorithm and Iterated Local Search*. Proceedings of the Genetic and Evolutionary Computation Conference, Part I, volume 3102, pp. 1328-1339.
- Lindenmayer, A., 1968. *Mathematical models for cellular interaction in development*. Journal of Theoretical Biology (Parts I and II), pp. 300-315.
- Lokta, A., 1925. *Elements of physical biology*. Baltimore, éd Williams and Wilkins.
- Lourenço, H. R., Martin, O. C. & Stützle, T., 2001. *Iterated local search*. : Handbook on MetaHeuristics.
- Mainzer, K., 2005. *Symmetry and complexity in dynamical systems*. 13(2), pp. 29-48.
- Margolus, N., 1984. *Physics-like models of computation*. Physica 10D, pp. 81-95.
- Mazoyer, J., 1987. *A six states minimal time solution to the firing squad synchronization problem*. Theoretical Computer Science, 50(2), pp. 183-238.
- Mazoyer, J., 1999. *Inducing an order on cellular automata by a grouping operation*. Discrete applied mathematics 91, pp. 177-196.
- Meinhardt, H., 1976. *Morphogenesis of lines and nets*. Differentiation, 6(2), pp. 117-123.
- Metropolis, N., Rosenbluth, A., Teller, A. & Teller, E., 1953. *Equation of State Calculations by Fast Computing Machines*. Journal of Chemical Physics. 21(6), pp. 1087-1095.
- Minsky, M., 1967. *Computation: Finite and infinite machines*. Prentice Hall, pp. 28-29.
- Mitchell, M., 1993. *Revisiting the Edge of Chaos: Evolving Cellular Automata to Perform Computations*. Complex Systems n°7, pp. 89-130.
- Mitchell, M., Crutchfield, J. P. & Das, R., 1996. *Evolving Cellular Automata with Genetic Algorithms: A review of recent work*. 1st International Conference on Evolutionary Computation and its Applications. Russia, pp. 1-15.
- Moore, 1964. *Sequential machines*. Wesley Longman éd. Essex: Series In Computer Science and Information Processing.
- Moreira, J. & Deutsch, A., 2002. *Cellular automaton models of tumor development: a critical review*. Advances in Complex Systems, Volume 2, pp. 247-267.
- Mühlenbein, H., Gorges-Schleuter, M. & Krämer, O., 1988. *Evolution algorithms in combinatorial optimization*. Parallel Computing, Volume 7, pp. 65-85.
- Nemhauser, G. & Wolsey, L., 1988. *Integer and combinatorial optimization*. John Wiley & Sons éd. : Wiley Series in Discrete Mathematics and Optimization .
- Neumann, J. V., 1966. *Theory of Self Reproducing Automata*. University of Illinois Press, pp. 64-87.
- Newell, A., Shaw, J. & Simon, H., 1957. *Emperical explorations with the logic theory machine*. Proceeding of the Western Joint Computer Conference, Volume 15, pp. 218-239.
- Ollinger, N., 2002. *Automates cellulaires: structures*, Lyon: Thèse de Doctorat, Ecole normale supérieure de Lyon.
- Prusinkiewicz, P., 1993. *Modeling and Visualization of Biological Structures*. Proceeding of Graphics Interface, pp. 128-137.

- Puchinger, J. & Raidl, G. R., 2005. *Combining metaheuristics and exact algorithms in combinatorial optimization: A survey and classification*. Artificial Intelligence and Knowledge Engineering Applications: A Bioinspired Approach, Volume 3562. pp. 41-53.
- Randell, P., 2001. *Turing Universality of the game of life*. Dans: A. Adamatzky, éd. Collision-Based Computing. London: Springer, pp. 513-539.
- Romanycia, M. H., 1985. *What is a heuristic?*. Computational Intelligence, 1(1), pp. 47-58.
- Romeo, F. & Sangiovanni-Vincentelli, A., 1985. *Probabilistic hill climbing algorithms: Properties and applications*. Electronics Research Laboratory, College of Engineering, University of California, Proceeding of 1985 Chapel Hill Conference on VLSI, pp. 393-417.
- Rosin, P., 2006. *Training cellular automata for images processing*. IEEE Transactions on Image Processing, 15(7), pp. 2076-2087.
- Rosin, P., 2010. *Image processing using 3-state cellular automata*. Computer Vision and Image Understanding, Volume 114, pp. 790-802.
- Sapin, E., 2003. *Recherche par algorithmes évolutionnaires d'automates cellulaires universels*, Dijon: Thèse de Doctorat, Université de Bourgogne.
- Sapin, E., Bailleux, O. & Channier, J., 2003. *Research of cellular automaton simulating logic gates by evolutionary algorithms*. Proceedings of the 6th European conference on Genetic programming, Springer-Verlag Berlin Heidelberg, pp. 414-423.
- Schönfisch, B. & De Roos, A., 1999. *Synchronous and asynchronous updating in cellular automata*. BioSystems, Volume 51, pp. 123-143.
- Shin, J., 2010. *Atlas of Patterns from One-rule Firing Cellular Automata*. Odense, Proceedings of the Alife XII Conference.
- Shu, H., Luo, L., Bao, X. & Yu, W., 2000. *An efficient Method for Computation of Legendre Moments*. Graphical Models, Volume 62, pp. 237-262.
- Siarry, P., 1995. *La méthode du recuit simulé : Théorie et Applications*. Automatique-productique informatique industrielle. 29(4-5), pp. 535-561.
- Siarry, P., Berthiau, F., Durbin, F. & Haussey, J., 1997. *Enhanced simulated annealing for globally minimizing functions of many continuous variables*. ACM Transactions on Mathematical Software, 23(2), pp. 209-228.
- Sipper, M. & Tomassini, M., 1999. *Computation in artificially evolved, non-uniform cellular automata*. Theoretical Computer Science, Volume 217, pp. 81-98.
- Soriano, P. & Gendreau, M., 1997. *Fondements et applications des méthodes de recherche avec tabous*. Revue Française d'automatique, d'informatique et de recherche opérationnelle. 31(2), pp. 133-159.
- Stauffer, A. & Sipper, M., 1998. *On the relationship between cellular automata and L-systems: The self-replication case*. Volume D 116, pp. 71-80.
- Talbi, E., 2002. *A taxonomy of hybrid metaheuristics*. Journal of heuristics, 8(2), pp. 541-564.
- Tanese, R., 1987. *Parallel genetic algorithm for hypercube*. Hillsdale, Proceedings of the Second International Conference on Genetic Algorithms on Genetic algorithms and their application, pp. 177-183.
- Teague, M. R., 1980. *Image analysis via the general theory of moments*. Journal of the Optical Society of America, 70(8), pp. 920-930.

- Turing, A., 1948. *Intelligent Machinery*, Report for National Physical Laboratory: National Physical Laboratory.
- Turing, A., 1952. *The chemical basis of Morphogenesis*. Philosophical Transactions of the royal Society of London. Series B, Biological Sciences, 237(641), pp. 37-72.
- Tyler Bonner, J., 2009. *Forme et croissance*. Critique, Edition du Seuil. pp. 764-765.
- Umeo, H., Hisaoka, M. & Sogabe, T., 2004. *A survey on optimum-time firing squad synchronization algorithms for one-dimensional cellular automaton*. International Journal of Unconventional Computing, Volume 1, pp. 403-426.
- Von Neumann, J. & Burks, A. W., 1966. *Theory of Self-Reproducing Automata*. Urbana and London: University of Illinois press.
- Waksman, A., 1966. *An optimum solution to the firing squad synchronization problem*. Information and Control, Volume 9, pp. 66-78.
- Westerhof, N., Lankhaar, J.-W. & Westerhof, B. E., 2009. *The arterial WindKessel*. Medical and Biomedical Engineering Computing, 47(2), pp. 131-141.
- Whale, J., Neubert, L., Esser, J. & Schrechenberg, 2001. *A cellular automaton traffic flow model for online simulation of traffic*. Elsevier Parallel Computing, Volume 27, pp. 719-735.
- Widmer, M., 2001. *Les métaheuristiques : des outils performants pour les problèmes industriels*. Troyes, 3ème Conférence Francophone de MODélisation et Simulation (Conception et analyse des systèmes industriels).
- Witten, T. & Sander, L., 1983. *Diffusion-limited aggregation*. Physical Review B, B(29), pp. 5686-5697.
- Wolfram, S., 1984. *Universality and complexity in cellular automata*. Physica 10D, pp. 1-35.
- Wolfram, S., 1986. *Random Sequence Generation by Cellular Automata*. Advances in applied mathematics, Volume 7, pp. 123-169.
- Wolpert, D. H. & Macready, W. G., 1997. *No Free Lunch Theorems for Optimization*. IEEE Transactions on Evolutionary Computation, 1(1), pp. 67-82.
- Wolpert, L., 1969. *Positional Information and the Spatial Pattern of Cellular Differentiation*. Journal of Theoretical Biology, Volume 25, pp. 1-47.
- Yang, Y. & Billings, S., 2003. *Identification of neighbourhood and CA rules from spatio-temporal CA patterns*. IEEE Transactions on systems man and Cybernetics Part B , 30(2), pp. 332-339.
- Yunès, J.-B., 1993. *Synchronisation et automates cellulaires : la ligne de fusiliers*, Paris: Thèse de doctorat, Université de Paris7.
- Zhao, Y. & Billings, S., 2006. *The identification of cellular automata*, Sheffield: Research Report, The University of Sheffield.