

République Algérienne Démocratique et Populaire
Ministère de l'enseignement Supérieur et de la Recherche Scientifique
Université Mentouri de Constantine
Faculté des Sciences de l'ingénieur
Département d'informatique

Année : 2011

N° d'ordre :/TS/2011

N° de série :/INF/2011

Thèse

Pour l'obtention du diplôme de Docteur en Sciences
Spécialité : Informatique

Modèles pour les systèmes complexes adaptatifs pour la résolution de problèmes :

Automates cellulaires apprenants quantiques et évolution différentielle quantique

Amer Draâ

Soutenue le/2011 devant le jury composé de :

Pr. Benmohammed Mohamed, Président, Université Mentouri de Constantine

Pr. Chikhi Salim, Directeur de thèse, Université Mentouri de Constantine

Dr. Atmani Baghdad, Examineur, Université d'Oran

Dr. Bouamrane Karim, Examineur, Université d'Oran

Dr. Chaoui Allaoua, Examineur, Université Mentouri de Constantine

**Modèles pour les systèmes complexes
adaptatifs pour la résolution de problèmes :**
**Automates cellulaires apprenants quantiques et
évolution différentielle quantique**



Amer Draâ

*A mes parents, mes enseignants, mes amis,
mes collègues et mes étudiants ...*

Amer ...

Remerciements

Tout d'abord, je tiens à exprimer ma profonde gratitude à mon directeur de thèse, Professeur Salim Chikhi pour m'avoir encadré, orienté et guidé afin d'achever ce travail. Son soutien moral, ses qualités humaines et son esprit scientifique ont été pour moi de vrais propulseurs pour réaliser ce travail de recherche.

Je remercie également tous les Professeurs membres du jury. Qu'ils veuillent trouver ici toute ma reconnaissance pour avoir accepté l'examen et l'évaluation de ce travail.

Un remerciement spécial va à Dr. Souham Meshoul, pour avoir aidé à l'émergence de cette thèse à travers ses conseils et ses contributions de haute qualité scientifique.

Je tiens aussi à remercier Pr. Mohammed Batouche, qui a bien aidé à enrichir ce travail par ces commentaires scientifiques appréciables.

Je remercie aussi tous les membres de ma famille qui m'ont soutenu et accompagné durant mes années d'étude, ainsi que toutes les personnes qui m'ont apporté un soutien moral de loin ou de près.

Je remercie toutes les personnes qui ont contribué à la réalisation de cette thèse, notamment Hichem Talbi, Nabil Belala, Abdesslem Layeb, Mustapha Khouadjia et Mme. Fergani.

Je n'oublie pas tous mes amis pour leurs encouragements en particulier Lamine Boukrouche, Hassen Chaker, Hichem Khelifi et Mehdi Talhi.

Finalement, je remercie tout le personnel du laboratoire MISC et du département d'informatique de l'université Mentouri de Constantine.

Résumé

Les systèmes complexes adaptatifs (Complex Adaptive Systems (CAS)) constituent une sous classe des systèmes complexes dans laquelle la propriété d'adaptation joue un rôle important dans le processus d'émergence. Pour pouvoir exploiter les atouts des CASs, les chercheurs dans le domaine ont proposé plusieurs modèles d'implémentation de leurs propriétés. Ces modèles sont tous inspirés de la nature : systèmes biologiques, systèmes sociaux et même systèmes non vivants. Parmi ces modèles on trouve les automates cellulaires, les systèmes multi-agents, les colonies de fourmis, les essaims de particules, les boids de Reynolds, et les réseaux de neurones artificiels. Vu l'absence d'une base théorique pour les CASs dans le domaine informatique, ces modèles souffrent de plusieurs limites du point de vue modélisation et mise en œuvre des CASs. Dans cette optique nous proposons dans cette thèse deux modèles pour les CASs, à savoir les automates cellulaires apprenants quantiques et l'évolution différentielle quantique. Le premier modèle est le résultat de l'enrichissement de l'hybridation des automates cellulaires apprenants et des boids de Reynolds par les concepts de l'informatique quantique. Ce modèle a été validé à travers la résolution du problème du voyageur de commerce et du problème d'optimisation de fonctions numériques. La deuxième contribution, l'évolution différentielle quantique, est une nouvelle version de l'évolution différentielle qui utilise les concepts de l'informatique quantique pour offrir plus de diversité à la population, en plus d'opérateurs quantiques permettant de bien guider la recherche. Ce modèle a été également utilisé pour résoudre deux problèmes d'optimisation : le problème des N reines et le problème d'optimisation de fonctions numériques. Les résultats expérimentaux obtenus des deux modèles ont montré une excellente convergence du système et une meilleure adaptation soit au niveau micro (des agents) ou au niveau macro (du système).

Mots-clés Systèmes complexes adaptatifs, Automates cellulaires apprenants, Évolution différentielle, Informatique quantique, Adaptation, Émergence, Complexité

Abstract

A Complex Adaptive System (CAS) is a collection of adaptive interconnected elements called agents. These agents have the capacity to change and learn from their experience. Because of the adaptation of its agents, the whole system is also adaptive. Complex adaptive systems have been studied in many fields, such as sociology, philosophy, biology, ecology, economics, and, recently, computer science. Though the large use of complex adaptive systems in the field of computer science, a unified definition of such systems and their properties has not yet been established. As a result, many previously proposed approaches to CAS modelling and implementing have been very limited in terms of exploiting the authentic properties of such systems, namely adaptation, complexity and emergence. In this topic, two new frameworks for CAS modelling and implementing are proposed in this thesis. The first one is a quantum-inspired learning cellular automaton, which is the result of the hybridisation between learning cellular automata, Reynolds' boids and quantum computing. This model has been validated through its application to solve two optimisation problems : the travelling salesman problem and the numerical function optimisation problem. In the second model, traditional differential evolution has been enriched with the use of a quantum representation of the individuals and some quantum-inspired operators. This approach has been also approved through using it to solve the N queens problem and the function optimisation problem. The obtained results are very promising, and the convergence of both algorithms is guaranteed. Comparisons with conventional approaches are given to prove the superiority of the proposed models in CAS modelling and problem solving.

Keywords Complex adaptive systems, Learning cellular automata, Differential evolution, Quantum computing, Adaptation, Emergence, Complexity

ملخص

تتكون الأنظمة المعقدة التآلفية من مجموعة من العناصر البسيطة القابلة للتأقلم والمتراطة فيما بينها والتي تسمى وكلاء. يستطيع هؤلاء الوكلاء التعلم من محيطهم والتأقلم معه استنادا إلى تجاربهم السابقة. ونظرا لإمكانيات التأقلم لدى الوكلاء، فإن النظام ككل يكتسب القدرة على التأقلم مع محيطه الخارجي. لقد تمت دراسة الأنظمة المعقدة التآلفية في عدة مجالات كعلم الاجتماع وعلم الحيوان والبيئة وعلوم الاقتصاد، ومؤخرا، علوم الحاسوب، وهذا ما عدّ من تنوع تعاريفها ومفاهيمها. تتميز هذه الأنظمة بعدة خصائص، أهمها: البروغ، التنظيم الذاتي والتأقلم. ومنذ ظهورها كجال قائم بذاته، تم اقتراح عدة نماذج اصطناعية لدراسة واستغلال هذه الأنظمة، لعل أهمها التلقائيات الخلوية، الأنظمة متعددة الوكلاء، الأنظمة المناعية الاصطناعية، بويدات رينولدز، والأسراب. ولى الرغم من الانتشار الواسع لهذه النماذج في مجال الحوسبة، فإن كلا منها يبقى محدودا من حيث امكانية نمذجة واستغلال الأنظمة المعقدة التآلفية. ومن هذا المنطلق، نهدف في هذه الأطروحة، إلى اقتراح نموذجين جديدين لمحاكاة واستغلال الأنظمة المعقدة التآلفية. أما النموذج الأول فهو عبارة عن تلقائية خلوية تعليمية كمية، وهي حاصل تمديد الخلايا التعليمية بفكرتين: تتمثل الأولى في استعمال عمليات مستمدة من نموذج رينولدز لتجسيد خاصية الذكاء الجماعي، وترتكز الثانية على استغلال التمثيل الكمي لخلايا التلقائية من أجل تجاوز تعقيد المشكلة المعالجة. بحل مشكلة التاجر المتجول ومشكلة تحسين الدوال، تمكنا من إثبات نجاعة هذا النموذج. وأما النموذج الثاني فيتلخص في خوارزم تطوري تفاضلي كمي، ويعتمد على استعمال تمثيل كمي للأفراد واستخدام عمليتين كميتين مستنبطتين من الحوسبة الكمية: القياس والتداخل، إضافة إلى الطفرة التفاضلية. ولقد مكّنتنا تجسيد النموذج الثاني لحل مشكلة الوزراء المعقدة ومشكلة تحسين الدوال من التحصل على نتائج جد مشجعة. إضافة إلى هذا، قمنا باقتراح امتداد لهذا النموذج بتعريف عملية جديدة تتمثل في إعادة التوليد والتي تسمح بتفادي الوقوع في النضج المبكر للخوارزم، وبالتالي جعله خصبا وزيادة من قدراته التآلفية. كلا النموذجين أعطى نتائج جد مشجعة مقارنة مع نموذجين كلاسيكيين: الأسراب الكمية والخوارزمات التطورية. كخلاصة، يمكن القول أننا تمكنا إلى حد بعيد من نمذجة الأنظمة التآلفية المعقدة واستغلالها لحل المشكلات.

كلمات مفتاحية: الأنظمة المعقدة التآلفية، التلقائيات الخلوية التعليمية، الخوارزمات التطورية التفاضلية، الحوسبة الكمية، التأقلم، البروغ، التعقيد

Table des matières

Remerciements	ii
Résumé	iii
Abstract	iv
Introduction générale	1
I Concepts de base et état de l’art	6
1 Systèmes complexes adaptatifs	7
1.1 Introduction	9
1.2 Système complexe adaptatif : définitions	9
1.2.1 Systèmes complexes	10
1.2.2 Systèmes complexes adaptatifs	11
1.3 Propriétés des systèmes complexes adaptatifs	13
1.3.1 Complexité	14
1.3.2 Émergence	15
1.3.3 Auto-organisation et systèmes auto-organisés	17
1.3.4 Attracteurs	19
1.3.5 Adaptation	21
1.3.6 Apprentissage	22
1.3.7 Différence entre apprentissage et adaptation	23
1.4 Structure et comportement d’un CAS	23
1.4.1 Relations entre agents dans un CAS	24
1.5 La théorie des systèmes complexes adaptatifs	25
1.6 Conclusion	26
2 Modèles artificiels pour les systèmes complexes adaptatifs	27
2.1 Introduction	29
2.2 Systèmes multi-agents	29

2.2.1	Types d'agents dans un SMA	29
2.2.2	Systèmes multi-agents et systèmes complexes adaptatifs	30
2.3	Automates cellulaires	31
2.3.1	Jeu de la vie	32
2.3.2	Automates cellulaires et systèmes complexes adaptatifs	33
2.4	Algorithmes évolutionnaires	34
2.4.1	Principe d'un algorithme évolutionnaire	35
2.4.2	Algorithmes génétiques	36
2.4.3	Algorithmes évolutionnaires et systèmes complexes adaptatifs	38
2.5	Réseaux de neurones artificiels	39
2.5.1	Réseaux de neurones et systèmes complexes adaptatifs	40
2.6	Conclusion	41
 II Éléments de base pour les modèles proposés		42
 3 Automates cellulaires apprenants		43
3.1	Introduction	45
3.2	Apprentissage par renforcement	45
3.3	Automates apprenants	47
3.3.1	Automate apprenant à structure fixe	48
3.3.2	Automate apprenant à structure variable	48
3.3.3	Signal de renforcement dans un VSLA	49
3.4	Automates cellulaires apprenants	50
3.5	Automates cellulaires apprenants irréguliers	51
3.6	LCAs et systèmes complexes adaptatifs	52
3.7	Conclusion	52
 4 Évolution différentielle		54
4.1	Introduction	56
4.2	Principe de l'évolution différentielle	56
4.2.1	Mutation différentielle	57
4.2.2	Croisement différentiel	57
4.2.3	Sélection	58
4.2.4	Critère d'arrêt	58
4.3	Variantes pour l'évolution différentielle	59
4.4	Avantages et limites de l'évolution différentielle	60
4.4.1	Avantages de l'évolution différentielle	60
4.4.2	Limites de l'évolution différentielle	61

4.5	Évolution différentielle et systèmes complexes adaptatifs	61
4.6	Conclusion	62
5	Informatique inspirée du quantique	63
5.1	Introduction	65
5.2	Informatique quantique	65
5.2.1	Bit quantique	65
5.2.2	Registre quantique	66
5.2.3	Les cinq principes de l'informatique quantique	67
5.2.4	Calcul quantique et algorithmes quantiques	68
5.2.5	Avantages et limites de l'informatique quantique	70
5.3	Algorithmes évolutionnaires quantiques	70
5.3.1	Individu quantique	71
5.3.2	Mesure quantique d'individus	72
5.3.3	La porte D	72
5.3.4	Discussion de l'algorithme évolutionnaire quantique	73
5.4	Autres algorithmes inspirés du quantique	74
5.4.1	Algorithmes génétiques quantiques	74
5.4.2	Boids de Reynolds quantiques	76
5.4.3	Réseaux de neurones quantiques	76
5.5	Avantages des algorithmes inspirés du quantique	77
5.6	Conclusion	77
III	Contributions	79
6	Automate cellulaire apprenant quantique	81
6.1	Introduction	83
6.2	Motivations et idée de base	83
6.3	Structure de l'automate cellulaire apprenant quantique	84
6.3.1	Une représentation quantique des cellules	84
6.3.2	Voisinage	85
6.3.3	Matrice de probabilités d'actions	85
6.4	Comportement de l'automate cellulaire apprenant quantique	87
6.4.1	Boids de Reynolds	87
6.4.2	Mesure quantique	90
6.4.3	Évaluation	90
6.4.4	Comparaison et mise à jour	91
6.4.5	Cohésion quantique et séparation quantique	91

6.4.6	Alignement quantique	93
6.4.7	Mise à jour de la matrice de probabilités d'actions	93
6.5	Applications	94
6.5.1	L'LCA quantique pour la résolution du TSP	94
6.5.2	L'LCA quantique pour l'optimisation de fonctions numériques	99
6.6	Conclusion	100
7	Validation de l'automate cellulaire apprenant quantique pour l'optimisation	101
7.1	Introduction	103
7.2	Automate cellulaire apprenant quantique pour le TSP	103
7.2.1	Bechmarks	103
7.2.2	Résultats expérimentaux	103
7.3	LCA quantique pour l'optimisation de fonctions	105
7.3.1	Fonctions bechmarks	105
7.3.2	Résultats expérimentaux	106
7.4	Étude des opérateurs du LCA quantique	110
7.5	Conclusion	113
8	Évolution différentielle quantique	114
8.1	Introduction	116
8.2	Idée de base	116
8.3	Algorithme à évolution différentielle quantique	116
8.3.1	Une représentation quantique des individus	117
8.3.2	Opérateurs quantiques	117
8.3.3	Autres opérateurs	120
8.3.4	Voisinage et sous-populations	120
8.3.5	Déroulement de l'algorithme différentiel quantique	121
8.4	EDQ avec régénération de la population	121
8.5	Applications	122
8.5.1	EDQ pour la résolution du problème des N reines	122
8.5.2	ADQR pour l'optimisation de Fonctions numériques	124
8.6	Colclusion	124
9	Validation de l'évolution différentielle quantique pour l'optimisation	126
9.1	Introduction	128
9.2	L'EDQ pour le problème des N reines	128
9.2.1	EDQ sans régénération de la population	128
9.2.2	Étude des opérateurs	129

9.2.3	EDQ avec régénération de la population	132
9.3	EDQ avec régénération pour l'optimisation de Fonctions	133
9.4	Discussion	133
9.4.1	Taille de la population	136
9.4.2	Taux de régénération	136
9.4.3	Taux de différence de la mutation différentielle	137
9.4.4	Taux d'interférences locale et globale	138
9.4.5	Coefficients d'interférence	139
9.5	Conclusion	139
	Conclusion générale	140
	Bibliographie	154

Table des figures

1.1	L'émergence	17
1.2	Types d'émergence	18
1.3	Types d'attracteurs dans un système dynamique	20
1.4	Structure et comportement d'un CAS	24
2.1	Les quatre classes des automates cellulaires	33
2.2	Démarche d'un algorithme génétique	37
2.3	Réseau de neurones artificiels	39
3.1	Apprentissage par renforcement	46
3.2	Automate apprenant et son environnement	47
3.3	Automate cellulaire apprenant irrégulier	52
4.1	Mutation différentielle	58
4.2	Croisement différentiel	59
5.1	Bit classique et bit quantique	66
5.2	Mesure quantique	67
5.3	Mesure quantique d'individus	72
5.4	Effet de la porte D sur un qubit	73
5.5	Croisement quantique	75
5.6	Mutation quantique	76
6.1	Représentation quantique d'un LCA	85
6.2	Voisinage de Von Neumann et voisinage de Moore	86
6.3	Matrice de probabilités d'actions pour un LCA quantique	86
6.4	Comportements de boids de Reynolds	88
6.5	L'émergence dans le modèle de boids	89
6.6	Mesure quantique d'une cellule	90
6.7	Évaluation d'une solution binaire dans le cas du TSP	91
6.8	Opérateurs de boids quantiques	94
6.9	Le circuit Hamiltonien du TSP	96

6.10	Matrice de distances pour TSP symétrique	96
6.11	Une matrice de solution pour le TSP	97
6.12	Codage quantique des cellules de l'automate pour le TSP	97
6.13	Permutation quantique	99
7.1	Évolution de la fitness dans un QLCA pour le TSP	105
7.2	La fonction Sphere dans l'espace bidimensionnel	106
7.3	La fonction Griewank dans l'espace bidimensionnel	106
7.4	La fonction Rosenbrock dans l'espace bidimensionnel	107
7.5	Évolution de fitness dans l'LCA quantique pour l'optimisation de fonctions	110
7.6	Evolution des fitness dans un QLCA avec et sans cohésion	111
7.7	Evolution des fitness dans un QLCA avec et sans séparation	112
7.8	Evolution des fitness dans un QLCA avec et sans alignement	112
7.9	Évolution de fitness pour différents taux de permutation	113
8.1	Individu quantique sous une forme matricielle	117
8.2	Mutation différentielle quantique	119
8.3	Une solution du problème des 8 reines	123
8.4	Une représentation binaire d'une solution du problème des 4 reines	123
8.5	Une configuration des reines avec fitness égale à 5	124
9.1	Evolutin de la fitness dans l'ADQ	129
9.2	AG classique et ADQ pour le problème des N reines	130
9.3	Opérateur de permutation dans l'ADQ pour les N reines	130
9.4	Apport de l'interférence locale dans l'ADQ	131
9.5	Comparaison de la solution globale avec et sans interférence locale	131
9.6	Comparaison de la solution globale avec et sans interférence globale	132
9.7	Comparaison de la solution globale avec et sans mutation différentielle . . .	132
9.8	Apport de l'application de la régénération dans l'ADQ	135
9.9	Une recherche avec régénération Vs. une recherche sans régénération	135
9.10	Taux de régénération dans l'ADQR	137
9.11	Taux de différence dans l'ADQR	138

Liste des tableaux

2.1	Une comparaison entre ACs et SMA pour la modélisation des CASs	34
5.1	Direction de la rotation	73
6.1	Les fonction benchmarks	99
7.1	Résultats expérimentaux de l’LCA quantique pour le TSP	104
7.2	Paramètres de l’LCA quantique pour l’optimisation de fonctions	108
7.3	Résultats expérimentaux de l’LCA quantique pour l’optimisation de fonctions numériques	109
7.4	Résultats pour 10 exécutions avec et sans l’utilisation de chaque opérateur	111
9.1	Résultats expérimentaux de l’EDQ quantique pour le problème des N reines	128
9.2	Résultats expérimentaux de l’ADQR pour le problème des reines	133
9.3	Résultats expérimentaux de l’ADQR pour l’optimisation de fonctions numériques	134
9.4	Résultats obtenus selon différents taux d’interférences locale et globale . .	139
9.5	Résultats obtenus selon différents coefficients d’interférences locale et globale	139

Liste d'algorithmes

2.1	Algorithme évolutionnaire	36
3.1	Automate apprenant à structure variable	50
3.2	Automate cellulaire apprenant (LCA)	51
4.1	Évolution différentielle (ED)	57
5.1	Algorithme évolutionnaire quantique (AEQ)	74
6.1	Boids de Reynolds (dans une scène 2D)	89
6.2	Cohésion et séparation quantiques	92
6.3	Automate cellulaire apprenant quantique	95
6.4	Mesure quantique pour le TSP	98

Introduction générale

Pendant plusieurs années, les scientifiques ont vu le monde comme un espace linéaire géré par le principe de "causes et effets". Ils ont présenté l'univers comme une grande machine et ils ont pensé qu'ils peuvent comprendre le tout en comprenant ses parties chacune à part. Ils ont pensé aussi que les composants de l'univers peuvent être présentés comme des machines et donc ils ont espéré que si on rend chacune des parties meilleure, on peut par conséquent rendre le tout meilleur. C'est par ce principe de causes et effets que les scientifiques espéraient que l'univers et ses parties peuvent être prévus et contrôlés. Cependant, ils ont échoué de trouver les composants manquants pour que l'image soit complète.

Parallèlement, le monde de la physique a montré une nouvelle stratégie de voir le lien entre les causes et les effets. Et puisque toute entité dans l'univers est constituée de la matière physique, ceci a mené les chercheurs à penser d'une nouvelle manière à l'univers d'où la naissance de *la théorie de la complexité* et du domaine des systèmes complexes. Pour la description des systèmes de l'univers, cette nouvelle théorie se base sur les relations entre les parties de ces systèmes, l'émergence de phénomènes étranges dans ces systèmes et les patterns constitués dans et par ces systèmes.

La théorie de la complexité énonce que l'univers est un tout complet constitué de systèmes : système écologique, système social, système immunitaire, etc. et que ces systèmes sont complexes et s'adaptent continuellement à leurs environnements. Aussi, les systèmes complexes, depuis la cellule jusqu'à l'écosphère, résultent de processus d'évolution et d'adaptation. Ce lien entre la complexité et l'adaptation a guidé les chercheurs à voir les systèmes complexes d'un autre angle, en donnant plus d'intérêt à la notion d'adaptation à ces systèmes, pour obtenir une nouvelle classe appelée *systèmes adaptatifs complexes* (en anglais : Complex Adaptive Systems, CASs).

Les CASs constituent une sous classe des systèmes complexes. Dans cette sous-classe, la propriété d'adaptation joue un rôle important dans la survie du système aux changements de son environnement. Ceci ne signifie pas que dans les systèmes complexes conventionnels cette propriété (l'adaptation) est absente, mais elle est généralement incluse sous d'autres concepts tels que l'apprentissage et l'auto-organisation.

De nouveaux courants de recherche pour le traitement de l'information sont apparus ces dernières années dont le but est de satisfaire les nouveaux besoins à des traitements plus sophistiqués. Parmi ces nouveaux domaines qui ont connu un intérêt grandissant, on trouve l'informatique quantique qui est le résultat d'un rapprochement multidisciplinaire incluant la physique, la chimie, l'informatique et les mathématiques. Cette révolution qui s'inspire des principes de la physique quantique permettrait de diminuer remarquablement la complexité algorithmique grâce notamment au calcul parallèle. Une telle possibilité du parallélisme peut être exploitée pour les problèmes d'optimisation qui manipulent une grande quantité d'information.

L'informatique quantique permet donc de concevoir des algorithmes parallèles plus puissants. Cependant, ces algorithmes ne sont réalisables que sur des machines quantiques dont la construction est toujours en état de recherche. En attendant la construction de machines quantiques puissantes, l'idée de simuler les algorithmes quantiques sur des ordinateurs classiques ou de les combiner à des techniques conventionnelles est apparue.

Les recherches dans le domaine de l'hybridation des modèles bio-inspirés et de l'informatique quantique ont commencé à la fin des années 1990. Le but principal de cette hybridation est l'amélioration du rendement de chacun de ces deux paradigmes en s'inspirant mutuellement l'un de l'autre. Pour atteindre ce but, deux axes de recherche ont vu le jour : Le premier vise la génération de nouveaux algorithmes quantiques en utilisant les techniques de programmation automatique (programmation génétique par exemple), ceci est motivé par la difficulté de concevoir de nouveaux algorithmes quantiques. L'autre se base sur le calcul classique inspiré des fondements quantiques et destiné à être exécuté sur des ordinateurs classiques. C'est une branche d'étude dans le calcul conventionnel où on a introduit certains principes de la mécanique quantique, tels que : l'interférence, la décohérence et la superposition d'états.

Problématique

Pour pouvoir exploiter les atouts des systèmes complexes, plusieurs modèles d'implémentation de leurs propriétés ont été proposés. Ces modèles sont tous inspirés de la nature : systèmes biologiques, systèmes sociaux et même systèmes non vivants. Parmi ces modèles, on trouve les automates cellulaires, les systèmes multi-agents, les colonies de fourmis, les essaims de particules, les boïds de Reynolds et les réseaux de neurones artificiels.

Malgré leurs capacités à résoudre des problèmes complexes, ces modèles restent très limités dans le contexte des CASs pour les raisons suivantes :

1. Ces modèles ne donnent pas une grande importance à la notion d'adaptation permettant à ces système de survivre dans des environnement ouverts et aléatoires.

2. Malgré que ces modèles sont tous inspirés de systèmes naturels puissants et intelligents, plusieurs aspects comportementaux ne sont pas inclus dans la version artificielle, ce qui la limite à des domaines d'applications et des environnements très spécifiques.
3. Les modèles artificiels existants des CASs souffrent généralement de deux problèmes : la convergence prématurée et la lenteur de convergence. Ceci est dû à la difficulté de trouver un bon compromis entre l'exploitation et l'exploration. Trop d'exploitation va causer une convergence prématurée du système, alors que trop d'exploration cause la lenteur de convergence.
4. Le réglage de paramètres est un autre problème dans ce type de systèmes. par exemple, malgré qu'ils ont été utilisés pour plus que deux décennies, les probabilités d'application de la mutation et du croisement dans les algorithmes génétiques sont fixés d'une manière empirique. Ceci mène à une grande implication du programmeur dans l'orientation du comportement du système.
5. Une autre limite très importante des modèles conventionnels des CASs est la négligence de l'environnement influençant l'évolution du système. Ceci a une influence négative non seulement sur l'exploitation du modèle pour des fins de calcul, mais aussi sur l'étude et la simulation des systèmes complexes adaptatifs naturels.

Notre travail dans le cadre de cette thèse consiste à proposer de nouvelles approches pour la simulation et l'exploitation des systèmes complexes adaptatifs. Ces modèles doivent exploiter le maximum des atouts des CASs, et bien sûr, doivent être capables de surmonter les limites trouvées dans les modèles conventionnels.

Contributions

Nos contributions peuvent se résumer en la propositions de nouvelles approches pour la mise en œuvre des CASs et ceci à travers l'extension de quelques modèles classiques, notamment les automates cellulaires et l'évolution différentielle, par des concepts inspirés d'autres domaines informatiques pour surmonter leurs limites. Pour ce faire, trois points clé ont été introduits :

1. l'hybridation entre deux modèles des CASs : l'automate cellulaire apprenant (en anglais : Learning Cellular Automaton, LCA) et les boids de Reynolds (BRs).
2. l'exploitation des concepts de l'informatique quantique par l'adoption d'une représentation quantique des individus et l'introduction d'opérateurs quantiques dans cette hybridation et dans le modèle d'évolution différentielle.

3. l'introduction d'autres opérateurs, pour le cas de l'évolution différentielle, telle que la régénération, pour augmenter la diversité au sein de la population.

Ainsi, deux modèles sont proposés pour la simulation et l'exploitation des propriétés des CASs :

1. Un automate cellulaire apprenant quantique : C'est le résultat de l'enrichissement de l'hybridation des automates cellulaires apprenants et des boîs de Reynolds (deux modèles des CASs) par les concepts de l'informatique quantique. Ce modèle a été validé à travers la résolution du problème du voyageur de commerce et du problème d'optimisation de fonctions numériques.
2. Un algorithme à Évolution Différentielle Quantique (EDQ) : C'est une nouvelle version de l'évolution différentielle qui utilise les concepts quantiques pour offrir plus de diversité à la population, en plus d'opérateurs quantiques permettant de bien guider la recherche. Ce modèle a été également utilisé pour résoudre deux problèmes d'optimisation : le problème des N reines et le problème d'optimisation de fonctions numériques.

Plan du document

Le travail présenté dans ce document est organisé en neuf chapitres regroupés en trois parties, respectivement : concepts de base et état de l'art, éléments de base pour les modèles proposés et contributions.

La première partie a été consacrée aux concepts de base et à l'état de l'art. Elle est constituée de deux chapitres. Le chapitre 1 donne une introduction aux systèmes complexes adaptatifs et introduit leurs concepts de base tels que l'adaptation, la complexité, l'émergence, l'apprentissage et l'auto-organisation. Le chapitre 2 présente quelques modèles artificiels pour la modélisation et la mise en œuvre des systèmes complexes adaptatifs. Les automates cellulaires et les systèmes multi-agents constituent la base de ce chapitre. Les modèles présentés sont discutés du point de vue modélisation des systèmes complexes adaptatifs à base des caractéristiques clé des CASs : l'adaptation, la complexité et l'émergence.

La seconde partie présente les éléments de base pour les modèles proposés, à savoir les approches de bases de nos contributions. Trois chapitres complets sont consacrés à cette partie importante :

Le chapitre 3 présente les automates cellulaires apprenants et démontre leur capacité à mettre en œuvre les concepts clé des systèmes complexes adaptatifs, notamment

l'adaptation et l'émergence. Les fondements du modèle sont également présentés dans ce chapitre : l'apprentissage par renforcement et les automates apprenants.

Dans le chapitre 4, les algorithmes à évolution différentielle sont présentés. Le modèle de base proposé par Storn et Price (1997) est discuté, ainsi que quelques variantes de ce modèle.

Le chapitre 5 ou troisième et dernier chapitre de cette partie, introduit l'informatique quantique comme domaine émergent en informatique. Il présente ses principes de base, ses avantages et ses limites. Il expose également des algorithmes inspirés du quantique tout en présentant trois modèles qui ont un lien fort avec nos contributions : les algorithmes évolutionnaires quantiques, les algorithmes génétiques quantiques et les boîds de Reynolds quantiques. Nous terminons ce chapitre par une synthèse des avantages des algorithmes inspirés du quantique.

Dans la troisième partie, nous mettons l'accent sur nos propres contributions. Nous avons utilisé quatre chapitres pour exposer et valider nos contributions :

Le chapitre 6 décrit dans sa première partie le comportement générique de l'automate cellulaire apprenant quantique proposé et explique les opérateurs quantiques utilisés par ce modèle. Ensuite, l'application du modèle proposé pour la résolution des deux problèmes, le problème du voyageur de commerce et le problème d'optimisation de fonctions, est détaillée.

Dans le chapitre 7, les résultats expérimentaux de l'application de l'automate cellulaire quantique pour l'optimisation sont expliqués et commentés.

Le chapitre 8 décrit en premier lieu l'hybridation entre l'évolution différentielle et l'informatique quantique. Ensuite, il présente son application pour la résolution de deux problèmes d'optimisation : le problème des N reines et le problème d'optimisation de fonctions numérique. Finalement, une extension de l'EDQ par un opérateur de régénération de la population est introduite.

Dans le chapitre 9, dernier chapitre de cette troisième partie et de la totalité de la thèse, le modèle à évolution différentielle quantique est validé à travers l'analyse des résultats expérimentaux obtenus de son application pour la résolution du problème des N reines et pour l'optimisation de fonctions numériques. Les opérateurs de cette approche ont été aussi étudiés en montrant l'apport de chacun d'eux sur le comportement global du système.

Première partie

Concepts de base et état de l'art

Chapitre 1

Systemes complexes adaptatifs

“The study of complex systems is the study of the world as it is,
not as we want it to be.” - Brenda Zimmerman, Edgeware

Sommaire

1.1	Introduction	9
1.2	Système complexe adaptatif : définitions	9
1.2.1	Systèmes complexes	10
1.2.2	Systèmes complexes adaptatifs	11
1.3	Propriétés des systèmes complexes adaptatifs	13
1.3.1	Complexité	14
1.3.2	Émergence	15
1.3.3	Auto-organisation et systèmes auto-organisés	17
1.3.4	Attracteurs	19
1.3.5	Adaptation	21
1.3.6	Apprentissage	22
1.3.7	Différence entre apprentissage et adaptation	23
1.4	Structure et comportement d'un CAS	23
1.4.1	Relations entre agents dans un CAS	24
1.5	La théorie des systèmes complexes adaptatifs	25
1.6	Conclusion	26

1.1 Introduction

Un système complexe adaptatif (en Anglais : Complex Adaptive System, CAS) est une collection d'éléments adaptatifs inter-connectés entre eux appelés agents. Ces agents ont la capacité de changer leurs comportements et d'apprendre de leurs expériences. A cause de l'adaptation de ses agents, le système global devient lui même adaptatif. Les systèmes complexes adaptatifs ont été largement étudiés dans plusieurs domaines et spécialités telles que la sociologie, la philosophie, la biologie, l'écologie, l'économie, et récemment, l'informatique.

Historiquement, l'apparition du domaine des systèmes complexes adaptatifs comme une nouvelle école de pensée remonte au milieu des années 1980 avec la formation de l'SFI (Santa Fe Institute), une institution formée principalement par les ex-membres du laboratoire LANL (Los Alamos National Laboratory). Les participants dans cette institution sont venus de diverses disciplines : l'économie, la physique, la biologie, l'écologie et l'archéologie. Le SFI a fourni beaucoup de produits scientifiques autour de la science de complexité au cours des deux dernières décennies. Plus de détails sur les activités de l'SFI peuvent être trouvés sur son site Web trouvé à l'adresse : <http://www.santafe.com/diverses>.

En plus de l'SFI, on trouve également le NECSI (*New England Complex Systems Institute*). C'est une institution de recherche indépendante dédiée à l'étude des systèmes complexes. Le NECSI joint les efforts intellectuels des institutions de la Nouvelle-Angleterre d'une façon indépendante au dehors des frontières institutionnelles et académiques. La liste de ses sponsors reflète le rôle offert par une telle institution où on trouve : *l'NSF, Boeing, Microsoft Research, LeBaron Foundation, Group Health Cooperative, Pan Agora Asset Management et World Bank* [NECSI, 2011]. Le site Web du NECSI se trouve sur l'url : <http://necsi.edu/>.

Malgré leur utilisation intense en informatique, une standardisation de définition des systèmes complexes adaptatifs et de leurs concepts reste loin d'être achevée. Dans ce chapitre, on définit ces systèmes, on introduit leurs concepts de base tels que l'adaptation, la complexité, l'émergence, l'apprentissage et l'auto-organisation, on présente la structure générique ainsi que le comportement d'un CAS dans son environnement, et on aborde la théorie des systèmes complexes adaptatifs.

1.2 Système complexe adaptatif : définitions

A cause de leur nature multidisciplinaire, on ne peut pas trouver une définition unique des systèmes complexes adaptatifs. Plusieurs définitions de tels systèmes ont été proposées,

chacune selon le contexte de son utilisation. Dans cette section, on présente quelques définitions des CASs tels que vus par certains chercheurs, mais avant, une définition des systèmes complexes (CSs) s'impose.

1.2.1 Systèmes complexes

Historiquement, le domaine des systèmes complexes a évolué selon les trois phases suivantes :

1. Des années 40 aux années 60 : Apparition de la science de la complexité grâce aux initiatives de plusieurs chercheurs provenant de disciplines variées. Ceci était le résultat de l'insatisfaction de plusieurs chercheurs des outils offerts par la science réductionniste pour la modélisation des phénomènes rencontrés dans leurs domaines de recherche. Les premières initiatives ont été les travaux du cybernéticien Ashby sur les propriétés du cerveau humain et des informaticiens Turing et Kolmogorov pour la conception d'une machine universelle, ainsi que les travaux de Von Neumann et Hofstadter portant sur la vie et l'intelligence artificielle [Turing, 1950; Ashby, 1956; Neumann, 1966].

2. Au cours des années 70 et 80 : Les mathématiciens et les physiciens du non-linéaire et de la mécanique statistique se sont intéressés aux phénomènes d'auto-organisation en généralisant aux systèmes biologiques l'idée que les propriétés globales émergentes de ce type de systèmes ne pouvaient être obtenues en effectuant un moyennage des propriétés des éléments en interaction, comme le font les méthodes classiques de la physique [Henon et Weisbuch, 2003].

3. A partir des années 90 : Dans cette période, l'apparition des approches bio-inspirées a très bien contribué dans la compréhension et la mise en œuvre des systèmes complexes. Parmi ces approches, on cite : les systèmes multi-agents, les approches basées essaims et les systèmes immunitaires artificiels [Langton, 1990; Henon et Weisbuch, 2003; Ianik, 1995].

Dans ce qui suit, on aborde deux définitions de base pour les systèmes complexes. Chacune est suivie d'une petite analyse de son contenu vis-à-vis sa capacité d'englober les propriétés des CSs naturels.

Définition 1.1 (Système complexe) *Un système complexe peut être vu comme un système dynamique composé de plusieurs systèmes simples et de parties interagissant d'une manière non linéaire [Ilachinski, 2004].*

Dans cette définition des systèmes complexes, Ilachinski ne prend pas en compte une caractéristique clé des systèmes complexes : c'est la différence de comportement existante

entre le bas niveau (composants et parties) et le haut niveau du système (le système lui-même). Cette différence est bien cernée dans la définition suivante.

Définition 1.2 (Système complexe) *Un système complexe est un système composé d'un ensemble d'éléments homogènes ou hétérogènes, interagissant entre eux de façon non linéaire (interactions rétroactives), mettant ainsi en œuvre une dynamique permettant à l'ensemble du système d'exister comme un tout, différent de la simple somme de ses composants. [Hassas, 2003].*

On voit bien que cette définition, donnée par Hassas, étend celle de Ilachinski par la mise en relief de la possibilité que les éléments du système peuvent être hétérogènes, et par le fait que le bas niveau (niveau micro) et le haut niveau (niveau macro) sont différents l'un de l'autre. Ici, on doit noter que l'existence de différence entre les niveaux macro et micro n'est pas obligatoire, et même si elle existe, des propriétés clé des composants sont toujours présentes au niveau globale. Un bon exemple de l'existence de similitude entre le niveau micro et le niveau macro est le système social : des caractéristiques clé des composants (soit humains, entreprises, etc) dans un système social sans toujours inhérentes dans le système lui-même.

Dans un système complexe, on distingue deux niveaux : un niveau micro, représentant le niveau des composants, avec des propriétés locales à chacun d'eux ; et un niveau macro, représentant l'ensemble du système, avec des propriétés nouvelles, que l'on ne retrouve dans aucun des composants pris individuellement. On parle alors d'émergence de nouvelles propriétés. Le caractère complexe d'un système tient aux émergences induites par les interactions rétroactives. Dans la dynamique du système, ces interactions permettent de mettre en œuvre les mécanismes de son adaptation et de son évolution par rapport aux évolutions de son environnement. On parlera alors de systèmes complexes adaptatifs [Hassas, 2003].

1.2.2 Systèmes complexes adaptatifs

Dans cette section, on présente quelques définitions des systèmes complexes adaptatifs. Après chaque définition, on donne notre point de vue sur les concepts des CASs abordés dans la définition en question.

Définition 1.3 (Système complexe adaptatif) *Un système complexe adaptatif est un système complexe dont les parties peuvent évoluer et s'adapter à un environnement variable [Ilachinski, 2004].*

Donc, l'adaptation du système est liée au fait que l'environnement est variable. Sachant que ce dernier dans la majorité des cas est constitué d'autres systèmes complexes

adaptatifs qui agissent et évoluent pour leurs propres objectifs qui peuvent être en conflit avec ceux des autres, on trouve que l'adaptation peut nécessiter non seulement le changement du comportement des agents ou du système, mais aussi le changement de sa structure. Cette possibilité de changement de structure n'est pas cernée dans la majorité des définitions des CASs existantes dans la littérature.

Définition 1.4 (Système complexe adaptatif) *Un système complexe adaptatif est une société d'un grand nombre d'agents rationnels [Namatame et Sasaki, 1998].*

Cette définition introduit deux notions élémentaires des CASs :

- Un CAS est constitué d'un grand nombre d'agents. En effet, le grand nombre d'agents en soit ne mène pas à l'émergence du comportement complexe. Il faut que ces agents soient fortement interconnectés pour que le système soit complexe.
- Il existe une certaine intelligence au niveau d'agents du système. Cette intelligence n'est pas une condition nécessaire, au contraire, elle peut exclure le système de la classe des systèmes complexes adaptatifs. Par exemple, dans une colonie de fourmis, le comportement simple au niveau individuel prouve que les composants ne sont pas assez intelligents, mais l'ensemble de composants comme un tout offre un comportement complexe.

Définition 1.5 (Système complexe adaptatif) *Un système complexe adaptatif est une collection d'agents hétérogènes et indépendants agissant d'une manière adaptative chacun avec l'autre et chacun avec son environnement dans le but de créer une entité distincte de haut niveau [Anderson, 1999].*

De cette définition, on obtient les caractéristiques suivantes des CASs :

- Une hétérogénéité au niveau micro du système. Cette caractéristique se voit bien dans les systèmes sociaux, notamment chez les êtres humains. Malgré leurs caractères, mentalités et cultures différentes, les sociétés d'humains sont fortement organisées.
- Les agents sont indépendants les uns des autres. Ceci offrent une certaine robustesse au système : si un agent ou une partie du système tombe en panne, le tout continue à exister et de fonctionner.

Définition 1.6 (Système complexe adaptatif) *Un système complexe adaptatif est un système composé de plusieurs parties interactives semi-autonomes (généralement appelées agents). Chaque partie a des comportements individuels simples. Quand ces parties sont agrégées, elles produisent des comportements émergents de haute complexité [Dooley, 1996].*

Cette définition introduit la semi-autonomie des agents. Ceci reflète bien le fait que les agents ne doivent pas nécessairement avoir une grande intelligence (voir commentaire sur la définition 1.4).

Dans son livre *Hidden Order*, Holland [1996] donne une définition plus complète des systèmes complexes adaptatifs, il les définit comme suit.

Définition 1.7 (Système complexe adaptatif) *Les systèmes complexes adaptatifs sont des systèmes composés d'agents interagissant les uns avec les autres qui répondent au stimulus [de leur environnement] sous forme d'un comportement pouvant être défini par des règles. Les agents [d'un CAS] s'adaptent par le changement de leurs règles de comportement résultant du cumul d'expérience. Ils peuvent être agrégés en méta-agents dont le comportement peut être émergent, i.e. un comportement ne pouvant pas être déterminé par l'analyse d'agents de bas niveau [Holland, 1996].*

De cette définition, on peut extraire ce qui suit comme caractéristiques des CASs :

- Les agents du système réagissent aux stimulus de l'environnement. Donc, selon la définitions, ils ne sont pas autonomes, ils sont réactifs.
- Le comportement des agents peut être défini par des règles. Dans le contexte des CASs, ces règles doivent être très simples.
- L'adaptation des agents à leur environnement se fait à travers le changement de règles de leurs comportements.
- C'est le cumul de l'expérience au niveau des agents qui mène à leur adaptation.
- Holland introduit ici la notion de méta-agent, qui peut être lui même un système complexe adaptatifs.
- Le comportement global du système ne peut pas être déduit des comportement des agents, c'est la notion d'émergence qu'on verra plus tard dans ce chapitre.

1.3 Propriétés des systèmes complexes adaptatifs

Dans cette section, on introduit les propriétés les plus pertinentes des systèmes complexes adaptatifs, principalement extraites des définitions données dans la section précédente. Même si elles ne sont pas décrites ici, d'autres propriétés telles que la hiérarchie des composants et le non-équilibre existent toujours dans un CAS.

1.3.1 Complexité

Il n'existe pas à ce jour une définition unique et acceptée par tous de la notion de complexité. Ceci n'est pas seulement dû à la jeunesse du domaine des systèmes complexes, mais aussi sa pluridisciplinarité. Chaque domaine scientifique étudie les systèmes complexes avec ses propres outils et prérogatives de sorte que des définitions propres à chaque discipline ont été formulées. Cependant, on trouve la définition suivante qui synthétise à un certain degré quelques définitions connues dans la littérature de la notion de complexité.

Définition 1.8 (Complexité) *La complexité est la propriété d'un modèle qui rend difficile la formulation de son comportement global même s'il y a assez d'information pour décrire ses parties et les relations entre elles [Edmonds, 2000].*

La complexité dans un CAS, comme dans une CS, est due aux interactions locales. Elle résulte principalement de la diversité des agents du système et du grand nombre de connexions entre ces agents. Selon Morin [2008] (Sociologue français, qui a développé un esprit interdisciplinaire dans le cadre de la théorie de la complexité) : "Penser en termes de complexité signifie reconnaître le *Principe Holistique* selon lequel *l'unité de l'organisme n'est pas réductible à la somme de ses parties constitutives*". Il propose la formulation suivante pour la description d'un système complexe :

$$\text{Système} = \sum \text{parties} + \sum \text{relations entre parties}$$

Le Moigne [1999] et Clergue [1997] font la distinction entre systèmes complexes et systèmes compliqués. Ils disent que les systèmes compliqués sont des systèmes que l'on peut réduire en éléments plus simples que l'on peut analyser séparément pour comprendre le système global, par exemple un avion. Alors que dans le cas des systèmes complexes, la somme des éléments fait émerger de nouvelles propriétés qui ne sont pas présentes dans les éléments composants du système, par exemple le système immunitaire.

• Types de complexité

Dodder et Dare [2000] distinguent trois types de base de la complexité :

1. Complexité statique : C'est l'ensemble d'aspects statiques de la complexité du système. Ça inclue les notions de hiérarchie, la connectivité, les détails, la complication, la variété et les niveaux/forces des interactions. Ce type de complexité est facilement visualisé comme un réseau de patterns complexes de liens et de nœuds. La complexité statique est à un certain degré dépendante du contexte car elle va apparaître très différemment aux niveaux micro et macro, et change quand les frontières du système changent.

2. Complexité dynamique : Ce type de complexité englobe les idées de complexité liées au comportement, aux processus de causes et effets, aux feedbacks, aux variations et stabilité, aux cycles et aux échelles de temps. Essentiellement, la théorie des CASs se base sur l'étude de cette notion de changement de comportement au fil du temps qui est liée à un aspect important : *la complexité évolutionnaire*. Cette évolution de la complexité dynamique peut résulter de l'adaptation du système et de ses agents. La complexité d'un système peut évoluer sans l'existence d'une adaptation au niveau de ses agents.
3. Complexité informationnelle : Elle représente une notion un peu plus abstraite (liée à la mesure de la complexité), qui peut être vue comme la complexité impliquée dans la description ou l'évaluation d'un système complexe. Elle peut refléter la complexité statique (e.g. la complication d'un réseau d'agents) ainsi que la complexité dynamique (e.g. la complexité des processus impliqués dans la création du système). D'un point de vue d'ingénierie, on considère aussi *la complexité d'évaluation*, qui peut être une forme de la complexité informationnelle requise pour décrire et évaluer la fonctionnalité, la performance et le succès du système.

• Complexité et traitement de l'information

La majorité des efforts développés dans la littérature concernant la complexité a été fourni dans les domaines des sciences naturelles et humaines. Ces domaines se sont concentrés plus sur l'étude descriptive et non prescriptive de la complexité. Pour cela, le passage vers la science de complexité dans le domaine d'ingénierie, malgré son importance pour ce domaine, est une tâche difficile ; on passe de la compréhension des systèmes à l'intervention dans ces systèmes.

Seth [1999] a bien exprimé ce problème dans la conférence d'ingénierie de la complexité qui a eu lieu en 1999 à l'MIT (Massachusetts Institute of Technology) quand il a dit : "L'objet de la conférence est de déterminer les implications que cette *complexification* va apporter à la société, et de développer des techniques et stratégies permettant l'ingénierie de systèmes qui seront complexes mais fiables". Malgré que la résolution de ce problème a été l'objet de plusieurs recherches depuis plus d'une décennie, ça reste toujours ouvert.

1.3.2 Émergence

Comme le résumait Ali et Zimmer [1997] dans leur article *The Question Concerning Emergence* [Ali et Zimmer, 1997], l'origine de l'émergence pourrait bien être le postulat datant de la Grèce antique : "**le tout est plus que la somme de ses parties**". On retrouve bien plus tard la même idée dans les écrits de *J. W. Von Goethe (1749-1832)*.

Ce poète, philosophe et scientifique élabore la ‘*théorie de la Gesta*’ (silhouette, forme) qui fait référence à une entité naturel qui est le point final d’un développement à partir d’un chaos primordial. Dès lors, de nombreux philosophes, mathématiciens, biologistes et physiciens participent à ce courant de pensée. Au milieu du XIXe siècle un nouveau courant est apparu autour du terme de l’émergence [McLaughlin et Varela, 1980]. Ce nouveau courant a été appelé plus tard le *Proto-émergentisme*.

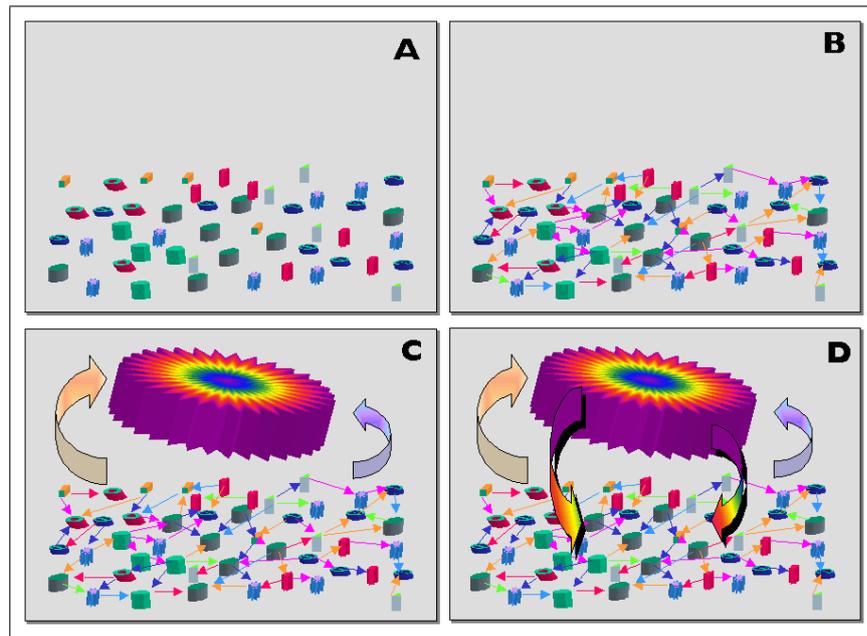
En 1875, le philosophe anglais *G. H. Lewis* fait la distinction entre *résultant* et *émergent*. Il explique que pour le résultant la séquence d’étapes qui produisent un phénomène est traçable, alors que pour *l’émergent* nous ne pouvons pas tracer les étapes du processus. Ainsi, nous ne pouvons pas voir dans le produit le mode d’opération de chaque facteur. L’émergence ne peut pas être réduite ni à la somme ni à la différence des forces co-opérantes [Kim, 2000]. Les proto-émergentistes cherchaient surtout à définir l’émergence afin de reconnaître un phénomène émergent et le différencier de phénomènes explicables grâce à d’autres théories ou modèles [George, 2003]. Dans ce qui suit, on donne quelques définitions formelles de l’émergence suivies d’une brève explication de leurs contenus.

Définition 1.9 (Émergence) *La propriété d’émergence réfère à l’apparition inattendue de patterns spatiaux et temporels dans la structure et la dynamique du système [Parrott, 2002]. Ceci est équivalent au fait que le comportement du système n’est pas déduit des comportements de ses parties.*

Définition 1.10 (Émergence) *La notion d’émergence peut être définie d’une manière intuitive comme une propriété macroscopique d’un système qui ne peut pas être inférée à partir de son fonctionnement microscopique [Gleizes, 2004].*

Définition 1.11 (Émergence) *Le principe de l’émergence correspond à l’apparition d’un tout cohérent à partir d’interactions locales d’agents non explicitement informés de la manière d’obtenir ce tout cohérent [Gleizes, 2004], comme illustré en Figure 1.1.*

En effet, le comportement global d’un système complexe ne peut être compris par la simple somme des comportements individuels des entités qui le composent. Il est impossible de comprendre les systèmes complexes sans penser que les agents simples d’une manière ou d’une autre vont donner des comportements complexes. Donc, dans l’étude de l’émergence dans les CASs, nous devons distinguer l’émergence locale de l’émergence globale. La première concerne une partie du système et elle est simple à analyser. Cependant la deuxième concerne les propriétés observées seulement au niveau du système global [George, 2003].



Un système d'agents divers (A), richement connecté (B), fait apparaître un pattern émergent (C), qui renvoie un feedback au système (D) [Seel, 2000].

Figure 1.1: L'émergence

• Types d'émergence

Fromm [2005] identifie quatre types de base de l'émergence. Selon lui, Dans le premier type, aucun 'feedback' existe au niveau du système, il y en a que des 'feedforwards'. La propriété majeure de l'émergence de type 2 est l'existence de feedbacks simples, soit positifs ou négatifs. Les feedbacks multiples, l'apprentissage et l'adaptation sont des points clé dans le troisième type de l'émergence. On trouve ce type d'émergence dans les CASs intelligents ayant une interaction intensive avec leurs environnements externes. Finalement, le quatrième type d'émergence est caractérisé par le multi-niveau et la diversité des patterns créés (Figure 1.2).

1.3.3 Auto-organisation et systèmes auto-organisés

Selon Parrott [2002], l'auto-organisation est le mécanisme responsable de l'émergence. C'est le processus par lequel l'effet collectif des interactions locales entre les entités du système, bien qu'apparemment désorganisé, forme une structure et un comportement ordonné émanant au niveau global.

L'auto-organisation peut aussi être vue comme une collaboration entre les entités

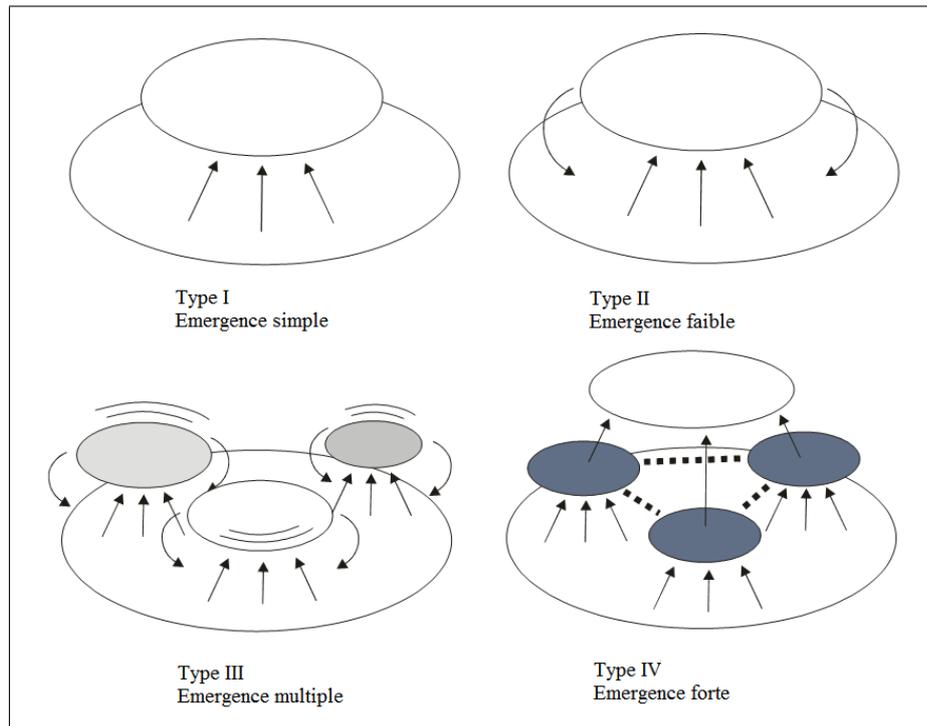


Figure 1.2: Types d'émergence

[Fromm, 2005]

du système qui modifient leur structure interne dans le but d'améliorer la viabilité et l'efficacité des relations que ce dernier entretient avec son environnement [Manson, 2001].

Il n'existe pas une hiérarchie de commandes et de contrôle dans un système complexe adaptatif. Donc, il n'y a pas de planning du comportement global et pas de gestion des tâches. Mais, il existe une auto-organisation continue agissant pour donner le meilleur rendement par rapport aux changements de l'environnement. Le système s'auto-organise par les processus d'émergence et de feedback.

• **Différence entre émergence et auto-organisation**

Selon Fromm [2006], les termes 'émergence' et 'auto-organisation' sont utilisés des fois d'une façon interchangeable, mais la différence entre les deux concepts derrière peut être cernée comme suit. Le terme auto-organisation réfère plus à un processus de dynamique résidant au frontières entre le système et son environnement. Tandis que le terme 'émergence' est concerné par les frontières entre le niveau micro et le niveau macro du système. En pratique, les deux phénomène se trouvent ensemble comme caractéristiques de plusieurs systèmes complexes naturels.

1.3.4 Attracteurs

Les trajectoires des systèmes complexes adaptatifs dans les espaces des phases montrent des patterns qui dépendent de plusieurs facteurs tels que : les conditions initiales, le type et les rôles des agents du CAS, le type des relations entre ces agents, les feedbacks, les règles communes d'agents, les valeurs, croyances et modèles internes des agents, etc. En effet, ces patterns ont la tendance de converger vers des régions spécifiques dans les espaces de phases au fil du temps. Ces régions sont connues sous le nom d'attracteurs [Couture et Valcartier, 2007].

Définition 1.12 (Attracteur) *Les attracteurs sont des patterns typiques de comportement, dynamiques, indépendants, de dimensionnalité limitée, et taillés d'un espace plus large de patterns et de dimensions possibles [Juarrero, 2010].*

Ces patterns qui émergent de l'interaction des composants du système peuvent être caractérisés comme des collections émergentes [Juarrero, 2010].

Goldstein [1999] fait un lien fort entre l'émergence et la notion d'attracteurs. Il annonce que "l'émergence dans les systèmes dynamiques est associée à l'apparition de nouveaux attracteurs" et que "ce sont ces attracteurs qui permettent l'émergence de quelque chose radicalement nouveau". Couture et Valcartier [2007] abordent les points suivants comme caractéristiques d'un attracteur dans un CAS.

- Les CASs sont organisés autour d'attracteurs qui sont des points (ou régions) vers lesquels les systèmes tendent à se déplacer intentionnellement ou à cause de contraintes établies par les paramètres du système.
- Ces attracteurs attirent toutes les trajectoires émanant de certains rangs de conditions de départ.
- Un attracteur est une caractérisation de long terme d'un système dynamique dissipatif (ouvert). A travers de longues périodes de temps, l'espace d'états (ou espace de phases) de certains systèmes dynamiques va se contracter vers cette région (attracteur).
- Les attracteurs déterminent les états qu'un système dynamique va tendre à adopter au fil du temps. Selon les conditions initiales, le système va suivre une trajectoire spécifique pour atteindre un attracteur. C'est la présence des attracteurs dans un système qui permet son auto-organisation et son comportement ordonné, au lieu d'un comportement complètement aléatoire.
- Des fois les attracteurs sont appelés des bassins d'attractions.

- Les attracteurs agissent comme des pôles d'attraction auto-organisés de comportement pour les interactions d'agents.

Williams [2001] distingue 4 types d'attracteurs, comme illustré en Figure 1.3 :

- **Attracteur ponctuel, ou à point fixe** : il correspond à un état stationnaire du système (pas d'évolution).
- **Attracteur périodique, ou avec cycle limite** : il est associé à un comportement périodique du système.
- **Attracteur tore** : il est caractérisé par un régime quasi-périodique ayant r fréquences de base indépendantes.
- **Attracteur étrange ou chaotique** : se trouve dans un système loin de l'équilibre : chaotique, non aléatoire, mais qui reste géré par des contraintes spécifiques.

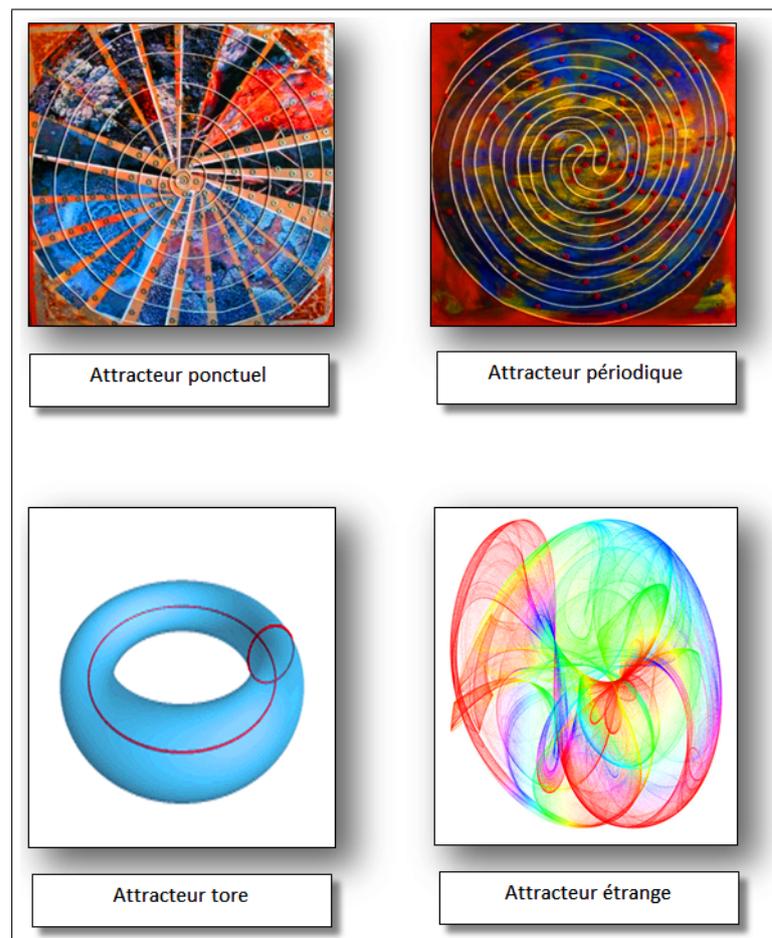


Figure 1.3: Types d'attracteurs dans un système dynamique

1.3.5 Adaptation

Le terme ‘adaptation’ est défini différemment selon le domaine considéré. Dans ce qui suit, on adopte deux définitions extraites respectivement des sciences de la nature et des sciences sociales. Ce choix est basé sur notre intérêt dans le cadre de cette thèse au systèmes inspirés de systèmes sociaux naturels.

Définition 1.13 (Adaptation) *L’adaptation dans les systèmes naturels réfère au développement des caractéristiques génétiques et comportementales qui permettent aux organismes ou systèmes de faire face aux changements de l’environnement afin de survivre et se reproduire [Smit et Wandel, 2006].*

Définition 1.14 (Adaptation) *Le processus d’adaptation est “un processus par lequel des groupes d’êtres humains ajoutent des méthodes nouvelles et améliorées pour faire face à l’environnement de leur répertoire culturel” O’Brien et Holland [1992].*

De manières plus ou moins vagues, d’autres définitions existent pour le terme adaptation. Pour ne pas s’étaler dans la définition du terme dans ce document, on préfère citer l’HDR de Rosnet [1999] dans laquelle l’auteur donne une bonne synthèse des définitions de l’adaptation selon différents points de vue philosophiques. Aussi, dans sa thèse, Elie-Dit-Cosaque [2009] fait appel à plusieurs définition de l’adaptation tout en s’appuyant sur une position plus générique que celle de Rosnet [1999] qui donne plus de spécificité selon les domaines.

On distingue principalement deux types d’adaptation : l’adaptation spontanée et l’adaptation planifiée. Dans la première, le système s’adapte d’une manière immédiate et non réfléchi d’un point de vue stratégique. Par contre, l’adaptation planifiée résulte de décisions stratégiques intentionnelles, fondées sur une perception claire des conditions qui ont changé - ou qui sont sur le point de changer - et sur les mesures qu’il convient de prendre pour revenir, s’en tenir ou parvenir à la situation souhaitée [EcoFr, 2011].

• Système adaptatif

Selon Grisogono [Grisogono, 2010], pour qu’un système soit adaptatif, il doit satisfaire les conditions suivantes :

1. l’existence de la notion de succès (techniquement connu sous le nom de ‘fitness’) et d’échec pour le système dans son environnement.
2. l’existence d’une source de variation dans les détails du système.
3. l’existence d’un processus de sélection au niveau des variations offertes pour l’amélioration de la fitness du système. Ceci nécessite :

4. l'existence d'un outil d'évaluation de l'impact d'une variation sur la fitness du système qui est généralement réalisée à travers un cycle d'interaction et feedback.

Tout le temps, le système génère et intériorise des variations qui tendent à augmenter sa fitness ou succès. Ce processus exploite la diversité, la variabilité, et la sélection basée succès, sous l'influence inhérente des changements de l'environnement affectant le système et son environnement.

Pour l'utilisation pratique de la notion d'adaptation dans le cadre de l'ingénierie des systèmes complexes, on doit bien savoir choisir les décisions à prendre dans les premiers stades et celles à laisser ouvertes pour une sélection adaptative réalisée par le système en court d'exécution (quand plus d'information sera disponible sur l'environnement du système). Plus de détails sur les principes guidant ces méta-décisions (décisions concernant les décisions) peuvent être trouvés dans l'article de Unewisse et Grisogono [2006] sur l'adaptabilité.

1.3.6 Apprentissage

L'apprentissage est une notion très liée au domaine de psychologie. Donc, il est clair que dans ce domaine on peut trouver les meilleures définitions de ce concept. Dans ce qui suit, on donne deux définitions de l'apprentissage et ses types, et on présente la différence entre l'apprentissage et l'adaptation.

Définition 1.15 (Apprentissage) *L'apprentissage est le processus continu de transformation de l'information et de l'expérience en connaissances, compétences, comportements et attitudes [Cobb, 2011].*

Définition 1.16 (Apprentissage) *L'apprentissage est défini comme un changement relativement permanent dans le comportement comme un résultat de l'expérience [Learn-Theo, 2011].*

• Types d'apprentissage

On distingue principalement trois types d'apprentissage : l'apprentissage supervisé, l'apprentissage non supervisé et l'apprentissage par renforcement. Dans le premier type, un superviseur externe est présent. Ce dernier fournit les réponses désirées (exemples et contre-exemples) au système. Dans le deuxième type, non supervisé, l'apprentissage se fait sans guide, l'apprenant fait des hypothèses, puis les tests, et ensuite il les évalue selon certains critères. Le troisième type d'apprentissage, apprentissage par renforcement, est réalisé à travers une séquence de perceptions de l'environnement, d'actions du système et de récompenses comme feedback de l'environnement dans lequel le système évolue [Cornuéjols et Miclet, 2002].

1.3.7 Différence entre apprentissage et adaptation

A partir des définitions ci-dessus de l'apprentissage et de l'adaptation, on peut bien voir qu'il existe un chevauchement entre les deux concepts du point de vue but avec des différences au niveau façon et durabilité. En ce qui concerne la similitude, le système (ou l'agent) dans les deux cas s'adapte ou apprend pour pouvoir survivre ou être plus performant dans un environnement donné [Eberhart et Shi, 2007].

En ce qui concerne les différences, on peut citer les suivantes. L'apprentissage se fait au niveau global du système, ce qui le rend intelligent, mais l'adaptation s'applique seulement à une partie du système. Donc, un système intelligent apprend et un système adaptatif s'adapte. Aussi, une différence importante entre les deux concepts se voit dans le fait que l'apprentissage se fait avec une présence de la conscience alors que l'adaptation ne nécessite pas sa présence. Une autre différence est que l'apprentissage est relativement durable, alors que l'adaptation ne l'est pas [Cobb, 2011; LearnTheo, 2011; Eberhart et Shi, 2007].

1.4 Structure et comportement d'un CAS

Dooley [1996] donne une bonne description du comportement d'un système complexe adaptatif qui englobe les caractéristiques trouvées dans les travaux de Holland [1996], Prigogine et Stengers [1984], et Maturana et Varela [1992]. Il annonce qu'un CAS se comporte et évolue selon les trois principes suivants.

- L'ordre est émergent et non pré-déterminé. L'historique du système est irréversible. Aussi, le comportement futur du système est souvent imprévisible.
- Les blocs de base composant un CAS sont les agents. Ces agents sont des éléments semi-autonomes qui cherchent à maximiser une certaine mesure de fitness en évoluant au fil du temps. Ils scannent leur environnement et développent des schémas représentant des règles d'interprétation et d'action. Ces schémas sont souvent développés à partir de schémas plus basiques.
- Ces schémas d'agents sont rationnellement bornés : ils sont potentiellement non-déterministes à cause de l'incomplétude ou l'incertitude de l'information. Ils dépendent de l'observateur car il est souvent difficile de séparer un phénomène de son contexte. Aussi, ces schémas peuvent être contradictoires les uns aux autres. Ils existent en multitudes et rivalisent pour survivre.

Donc, un système complexe adaptatif reçoit l'information de son environnement, identifie des régularités dans cette information, et condensent ces régularités sous forme de

schémas ou modèles à adopter pour pouvoir gérer son monde. Il existe toujours une variété de modèles et de schémas d'interprétation et d'action offerts au système. Le résultat de l'action du système va stimuler des feedbacks de l'environnement qui vont influencer les futures choix des schémas [Robin, 2003]. La figure 1.4 synthétise la structure et le comportement d'un CAS.

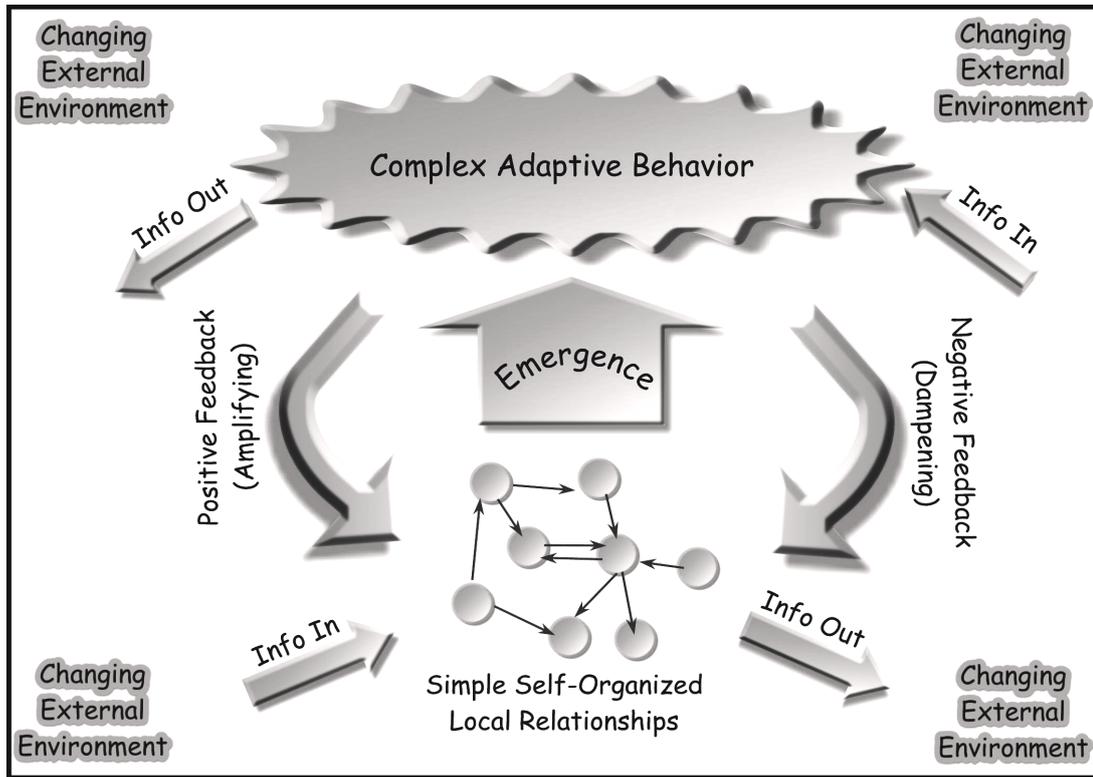


Figure 1.4: Structure et comportement d'un CAS

[Reus, 2010]

1.4.1 Relations entre agents dans un CAS

Selon Gharajedaghi [1999], on peut trouver quatre classes d'inter-relations entre agents (ou ensemble d'agents) dans un CAS. Ces classes sont déterminées par le niveau de compatibilité entre les moyens et finalités des agents. On peut les résumer comme suit, sachant que la survie d'un CAS dans un environnement ouvert peut nécessiter n'importe quel type de ces inter-relations.

- **Coopération** : la coopération (ou collaboration) entre les agents d'un CAS est possible quand les moyens des agents ainsi que leurs finalités sont compatibles.

- **Coalition** : elle peut avoir lieu quand les moyens d'agents sont compatibles mais leurs finalités sont incompatibles.
- **Compétition** : la compétition entre agents apparaît quand les moyens des agents sont incompatibles mais leurs finalités sont compatibles.
- **Conflit** : le conflit entre agents d'un CAS apparaît quand les moyens des agents ainsi que leurs finalités sont incompatibles. Ce type d'inter-relation entre agents peut être dangereux car les agents vont souvent combattre les uns contre les autres avec une intention de destruction.

1.5 La théorie des systèmes complexes adaptatifs

Avant d'introduire la théorie des CASs, il s'avère nécessaire de parler de la théorie des systèmes. Cette dernière est une théorie qui représente le monde comme un ensemble de stocks et de flux régis par une variété de processus de feedbacks. C'est une approche analytique dans laquelle les représentations des systèmes peuvent faire l'objet d'analyse mathématique dont le but est de comprendre le comportement du système en question [Hartvigsen *et al.*, 1998].

La théorie des CASs se base sur la théorie des systèmes avec une représentation explicite de la diversité et l'hétérogénéité que la théorie des systèmes a essayé d'agréger dans les flux et stocks homogènes. La théorie des CASs se concentre sur la non-linéarité de la dynamique du système. Ceci a permis de bien analyser les systèmes naturels car cette non-linéarité est une caractéristique clé dans l'analyse et la prédiction de ces systèmes. De plus, la théorie des CASs diffère de celle des systèmes, par la mise explicite d'accent sur le rôle de l'adaptation qui gouverne la dynamique et les réponses d'agents hétérogènes à leurs voisinages directs ou à l'environnement externe [Mitchell, 1998; Goldberg, 1989].

L'introduction de la notion d'adaptation dans la théorie des CASs a permis une très bonne compréhension de la façon par laquelle les patterns et les processus émergent et interagissent à différents niveaux d'organisation biologiques et à différentes échelles temporelles et spatiales [Hartvigsen *et al.*, 1998]. La théorie des systèmes complexes adaptatifs a développé plusieurs frameworks analytiques pour aider à comprendre et à gérer les systèmes complexes adaptatifs [Higgs, 2001].

Une théorie voisine qui est souvent confondue avec la théorie des CASs est celle du Chaos. La théorie des CASs diffère de la théorie du Chaos du fait qu'elle étudie les systèmes présentant un certain degré d'ordre : des propriétés émergentes apparaissent à partir d'un background chaotique. La différence en termes d'approche est que les chercheurs dans le domaine des CASs sont intéressés à l'ordre émergent plus qu'au désordre.

1.6 Conclusion

Dans ce chapitre, on a donné une synthèse sur les systèmes complexes adaptatifs. On a introduit quelques définitions de ces systèmes ainsi que leurs propriétés de base. Après avoir introduit ces notions, on a présenté la théorie des systèmes complexes adaptatifs comme outil de base de l'étude de ces systèmes dans différents domaines scientifiques.

Dans le prochain chapitre, on présentera quelques modèles artificiels pour la modélisation et la mise en œuvre des systèmes complexes adaptatifs. On présentera aussi une étude analytique de ces modèles artificiels vis-à-vis les propriétés des systèmes complexes adaptatifs.

Chapitre 2

Modèles artificiels pour les systèmes complexes adaptatifs

“A year spent in artificial intelligence is enough to make one believe in God.” - Alan J. Perlis

Sommaire

2.1	Introduction	29
2.2	Systèmes multi-agents	29
2.2.1	Types d'agents dans un SMA	29
2.2.2	Systèmes multi-agents et systèmes complexes adaptatifs	30
2.3	Automates cellulaires	31
2.3.1	Jeu de la vie	32
2.3.2	Automates cellulaires et systèmes complexes adaptatifs	33
2.4	Algorithmes évolutionnaires	34
2.4.1	Principe d'un algorithme évolutionnaire	35
2.4.2	Algorithmes génétiques	36
2.4.3	Algorithmes évolutionnaires et systèmes complexes adaptatifs	38
2.5	Réseaux de neurones artificiels	39
2.5.1	Réseaux de neurones et systèmes complexes adaptatifs	40
2.6	Conclusion	41

2.1 Introduction

Selon Fromm [2004], les Systèmes Multi-Agents (SMA) et les Automates Cellulaires (ACs) sont les deux outils les plus adéquats pour la modélisation des systèmes complexes adaptatifs. Il annonce aussi que : “les SMA et les ACs sont utilisés ces jours pour expliquer les systèmes adaptatifs complexes et compliqués”. Dans ce chapitre, on présente les principes de base de ces deux modèles ainsi qu’autres modèles artificiels inspirés de la nature. On se concentre plus sur les automates cellulaires qui sont utilisés comme une base, en plus des approches basées population, de l’une de nos contributions dans cette thèse.

2.2 Systèmes multi-agents

Le domaine des systèmes multi-agents est né à la fin des années 70 et début des années 80, de l’idée de distribuer les connaissances et le contrôle dans les systèmes d’intelligence Artificielle. Cette idée a émergé d’une part du besoin de faire face à la complexité croissante de ces systèmes et a été favorisée d’une autre part par l’émergence des modèles et machines parallèles, rendant possible la mise en œuvre opérationnelle de la distribution [Hassas, 2003].

Définition 2.1 (Agent) *Un agent est une “entité computationnelle”, comme un programme informatique ou un robot, qui peut être vue comme percevant et agissant d’une façon autonome sur son environnement. On peut parler d’autonomie parce que son comportement dépend au moins partiellement de son expérience [Weiss, 1999].*

Définition 2.2 (Système multi-agents) *Un système multi-agents est un ensemble d’entités (physiques ou virtuelles) appelées agents, partageant un environnement commun (physique ou virtuel), qu’elles sont capables de percevoir et sur lequel elles peuvent agir [Bonabeau et al., 1998].*

Les perceptions permettent aux agents d’acquérir des informations sur l’évolution de leur environnement, et leurs actions leur permettent entre autres de modifier l’environnement. Les agents interagissent entre eux directement ou indirectement, et exhibent des comportements corrélés créant ainsi une synergie permettant à l’ensemble des agents de former un collectif organisé.

2.2.1 Types d’agents dans un SMA

Potgieter [2004] distingue trois types d’agents dans un SMA :

- **Agent simple** : n’importe quelle partie d’un programme assez simple à comprendre, même si les interactions entre groupes formés de ce type d’agents sont largement très difficiles à comprendre.

- **Agent autonome** : c'est un programme informatique situé dans un environnement, qui est capable de faire des actions autonomes flexibles afin de réaliser ses objectifs.
- **Agent adaptatif** : c'est un agent qui peut s'améliorer au fil du temps, i.e. il devient meilleur avec le temps dans l'atteinte de ses objectifs. Donc, c'est un agent capable de changer et d'améliorer son comportement au fil du temps.

Actuellement, il existe deux tendances de recherche dans le domaine des multi-agents : la tendance cognitive, influencée par l'IA symbolique, et la tendance réactive influencée par la vie artificielle et l'intelligence 'sans représentation', appelée aussi *embodied intelligence* [Hassas, 2003]. Cette dernière approche est la plus adaptée pour la modélisation des systèmes complexes adaptatifs car elle inclue l'émergence et l'imprévisibilité, caractéristiques clé dans les CASs.

Le point clé des systèmes multi-agents réside dans la formalisation de la coordination entre les agents. La recherche sur les agents est ainsi une recherche sur [Cormas, 2010] :

1. **La décision** : On cherche à répondre aux questions suivantes : Quels sont les mécanismes de décision de l'agent ? Quelle est la relation entre les perceptions, les représentations et les actions des agents ? Comment décomposent-ils leurs buts et tâches ? Comment construisent-ils leurs représentations ?
2. **Le contrôle** : On cherche à répondre aux questions suivantes : Quelles sont les relations entre les agents ? Comment sont-ils coordonnés ?
3. **La communication** : On cherche à répondre aux questions suivantes : Quels types de message s'envoient-ils ? A quelle syntaxe obéissent ces messages ?

Les SMAs ont été largement appliqués pour la résolution de problèmes, la simulation de systèmes intelligents, la construction de mondes synthétiques et la robotique. Ils ont également des applications dans le domaine de l'intelligence artificielle où ils permettent de réduire la complexité de la résolution d'un problème en divisant le savoir nécessaire en sous-ensembles, en associant un agent intelligent indépendant à chacun de ces sous-ensembles et en coordonnant l'activité de ces agents. On parle ainsi d'intelligence artificielle distribuée [Cormas, 2010].

2.2.2 Systèmes multi-agents et systèmes complexes adaptatifs

Les motivations derrière le choix des systèmes multi-agents comme étant un modèle puissant pour la mise en œuvre des systèmes complexes adaptatifs ne sont pas souvent introduites dans la littérature des SMAs pour la raison principale que ces travaux sont orientés application. Une autre raison qui n'a pas permis l'adoption des SMAs comme modèle des

CASs est la jeunesse de la théorie des CASs et de la non standardisation des concepts. Un bon exemple de cette non standardisation est que chacun des domaines d'étude et d'utilisation des CASs utilisent ses propres définitions et outils, par contre au domaine des SMAs qui est très lié à l'informatique. Dans ce qui suit, on essaie de mettre en relief le lien entre les SMAs et les CASs à travers l'analyse des définitions et des caractéristiques des SMAs données par les sections précédentes. En analysant la définition des SMAs donnée par Bonabeau *et al.* [1998] (Définition 2.2) on peut extraire les points suivants :

- **L'agent** : Comme dans un CAS, l'agent dans un SMA est un composant du système (soit un individu, une partie ou même un CAS composant du système). Cependant, il existe une différence principale, comme décrit par Fromm [2004], entre l'agent d'un CAS et celui d'un SMA : un agent dans un CAS est très simple qu'il l'est dans un SMA. Cette différence est l'une des raisons qui nous a fait éviter le modèle des SMAs dans la composition des contributions proposées dans cette thèse pour la mise en œuvre des CASs.
- **L'environnement** : c'est la même notion d'environnement dans les deux types de systèmes. Selon la définition de Bonabeau *et al.* [1998] (Section 2.2), il existe une possibilité que les agents change l'environnement dans lequel ils existent, en plus qu'ils sont influencés par lui. Ceci nous rappelle de la notion de 'co-évolution' d'un CAS avec son environnement.
- **Les interactions** : l'interaction entre les agents du système à bas niveau dans les deux modèles mène à une complexité au niveau global.
- **L'organisation** : Ici, on doit stresser le fait que dans un SMA, il doit exister une auto-organisation et pas une organisation pour qu'on peut parler d'un CAS. Cette auto-organisation doit être créée par le système et pas imposée par un agent ou une entité externe, par exemple un être humain.

2.3 Automates cellulaires

Les Automates cellulaires ont été inventés par *Stanislaw Ulam* (1909-1984) et *John Von Neumann* (1903-1957) à la fin des années quarante, à Los Alamos National Laboratory (États-Unis) [Langton, 1986]. Un automate cellulaire est une grille D -dimensionnelle composée d'agents appelés cellules. Ces cellules interagissent et évoluent dans le temps en changeant leurs états continuellement. Les règles de transition de chaque cellule sont locales et simples. Elles ne prennent en compte que l'état des cellules voisines.

Formellement, un automate cellulaire est un quintuplet $A = (L, D, S, N, R)$. L est la grille cellulaire D dimensionnelle. A chaque instant discret t , chaque cellule ne peut avoir

qu'un seul état d'un ensemble fini S . A chaque étape de temps, les états de toutes les cellules sont mis à jour simultanément. Le processus de transition est guidé par une règle de mise à jour locale R , qui est la même pour toutes les cellules (Équation 2.1). Cette règle reçoit comme entrée le voisinage local N de la cellule (états des cellules voisines), et produit l'état futur de la cellule [Ganguly *et al.*, 2003].

$$S_{i,j}^{t+1} = R(N(S_{i,j}^t)) \quad (2.1)$$

• Classes d'automates cellulaires

Les automates cellulaires peuvent être classés en 4 catégories selon le type de comportement à savoir [Wolfram, 1983, 2002]. Ces classes sont illustrées dans la figure 2.1.

- Classe 1 - *attracteur ponctuel* : le système dans cette classe converge vers un état fixe après une courte période de temps.
- Classe 2 - *cycles limites (périodique)* : le système développe des comportements périodiques qui se répètent continuellement.
- Classe 3 - *chaotique* : le système devient non-périodique se changeant continuellement de façons imprévisibles et aléatoires.
- Classe 4 - *structuré* : le système peut se développer à des formes fortement structurées mais non-stables. Cette classe d'automates se situe entre les classes 2 et 3.

2.3.1 Jeu de la vie

Le jeu de la vie est un automate cellulaire inventé par John Conway en 1970 dont l'objet était la simulation d'une forme très basique de la vie. Le principe du jeu de la vie est le suivant "pour pouvoir vivre, il faut avoir des voisins ; mais quand il y en a trop, on étouffe". L'automate cellulaire simulant ce comportement fonctionne comme suit. Les cellules de l'automate peuvent se trouver dans l'un des états 'vivant' ou 'mort'. L'automate commence d'une configuration initiale dans laquelle toutes les cellules sont dans l'état 'mort' sauf que pour un nombre fini d'entre elles. Cet automate évolue en suivant les règles suivantes :

1. Une cellule inactive entourée de 3 cellules actives devient active ("naît") ;
2. Une cellule active entourée de 2 ou 3 cellules actives reste active ;
3. Dans tous les autres cas, la cellule "meurt" ou reste inactive.

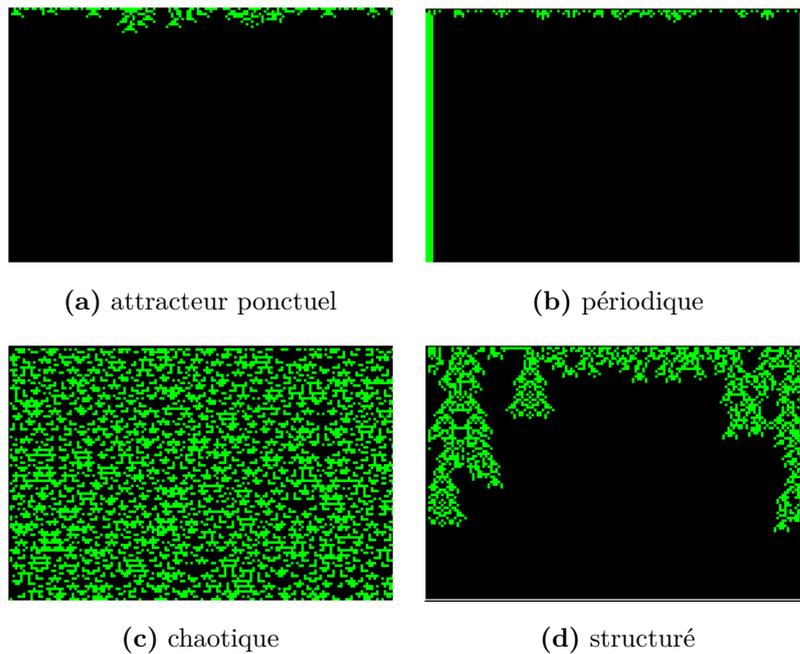


Figure 2.1: Les quatre classes des automates cellulaires

2.3.2 Automates cellulaires et systèmes complexes adaptatifs

Pour simuler le comportement des systèmes complexes, les automates cellulaires ont été utilisées comme un modèle mathématique puissant [Langton, 1986]. Ils ont l'avantage d'être faciles à comprendre et à mettre en œuvre [Fromm, 2004]. Dans un contexte informatique pour la simulation et l'exploitation des CASs, les automates cellulaires ont les avantages suivants par rapport aux techniques conventionnelles :

1. Ils offrent des coûts faibles de calcul pour des modèles à haute complexité.
2. Ils sont faciles à programmer.
3. Ils permettent une contrôlabilité élevée du modèle.
4. Ils sont faciles à visualiser.
5. Ils permettent une possibilité de paralléliser un grand nombre d'entrées.
6. Ils ont une grande portabilité d'autres modèles mathématiques de calcul.
7. Ils sont bien adaptés à la 'siliconisation'.

Le "jeu de la vie", décrit ci-dessus est l'un des modèles des ACs les plus proches du comportement d'un CAS. l'évolution des objets manipulés, bien que totalement déterminée par les fonctions de transition, est hautement imprévisible. On remarque aussi que le simple fait d'ajouter ou d'enlever une cellule dans une configuration change son évolution d'une façon radicale, c'est exactement le même phénomène observé dans le système écologique dans lequel n'importe quelle petite modification, même très locale, va influencer tout le système d'une façon imprévisible.

Pour synthétiser les points forts et les points faibles des deux modèles, ACs et SMAs, le tableau 2.1 (extrait de Fromm [2004]) donne une petite comparaison entre les deux approches pour la modélisation des CASs. Ce tableau reflète bien les avantages des ACs par rapport aux SMAs pour la simulation et l'exploitation des CASs.

Table 2.1: Une comparaison entre ACs et SMA pour la modélisation des CASs

Propriétés/Système	Automate cellulaire (CA)	Système multi-agents (SMA)
Phénomènes décrit	Patterns et structures complexes	Structures simples
Agents, Objets et Composants	Agents simples, machines d'états finis (MEF) ou automates	Agents complexes, la complexité dépend de la structure interne
Règle locales de base	Table de transition	Langage de communication et intentions entre agents
Phénomènes émergents	Structures préparées	Culture puissante, etc.

En plus des systèmes multi-agents et des automates cellulaires, d'autres modèles exploitant les caractéristiques des CASs pour les besoins de calcul ont été largement utilisés, notamment : les algorithmes évolutionnaires (AEs) et les Réseaux de Neurones Artificiels (RNAs).

2.4 Algorithmes évolutionnaires

Les Algorithmes Evolutionnaires (AEs) font partie du champ de l'intelligence artificielle. Ils sont apparus à la fin des années 1950 [Fraser, 1957]. Il s'agit d'une Intelligence de bas niveau inspirée de l'intelligence naturelle. Ces algorithmes sont basés sur la théorie de l'évolution et de la sélection naturelle élaborée par Charles Darwin, qui annonce que dans un environnement quelconque, seules les espèces les mieux adaptées survivent, les autres étant condamnées à disparaître. Au sein de chaque espèce, le renouvellement des populations est essentiellement dû aux meilleurs individus de l'espèce.

Les AEs constituent une approche originale : il ne s'agit pas de trouver une solution analytique exacte, ou une bonne approximation numérique, mais de trouver des solutions satisfaisant au mieux certains critères, souvent contradictoires. S'ils ne permettent pas de trouver à coup sûr la solution optimale de l'espace de recherche, du moins peut-on

constater que les solutions fournies sont généralement meilleures que celles obtenues par des méthodes plus classiques, pour un même temps de calcul. Trois classes d'algorithmes évolutionnaires ont été développées indépendamment, entre la moitié des années 1960 et 1970 :

- *les algorithmes génétiques (AGs)* [Holland, 1962, 1975; Goldberg, 1989; Goldberg et Corruble, 1994] : on les verra en détail dans la section 2.4.2. Une autre catégorie des AEs est généralement incluse dans les AGs, c'est la programmation génétique ayant comme espace de recherche un ensemble d'instructions et de programmes au lieu de bits ou de réels [Koza, 1996].
- *la programmation évolutionnaire* [Fogel *et al.*, 1966, 1995] : conçu comme une méthode d'intelligence artificielle pour la conception d'automates à état fini.
- *les stratégies évolutionnaires* [Rechenberg, 1965; Beyer et Schwefel, 2002].

Dans les années 90, ces trois champs de recherche ont été regroupés dans ce qui est devenu "Evolutionary Computation" [Fogel, 1997].

2.4.1 Principe d'un algorithme évolutionnaire

Un algorithme évolutionnaire opère sur une population d'individus représentant des solutions potentielles du problème à résoudre. A chaque génération, l'algorithme applique les trois opérations suivantes en 4 phases :

1. **La sélection** : choisir les individus les plus performants pour la reproduction et la survie.
2. **La reproduction** : permet le brassage, la recombinaison et les variations des caractères héréditaires des parents pour former les descendants.
3. **L'évaluation** : offre à chaque individu une qualité d'adaptation qu'on appelle 'fitness'.

La sélection est appliquée en premier lieu pour choisir des individus à reproduire, et puis elle est appliquée à la fin de l'itération pour choisir les individus qui vont survivre, comme montré par l'algorithme 2.1.

Dans ce qui suit, on présente une de ces classes des AEs. Il s'agit des algorithmes génétiques.

Algorithme 2.1 Algorithme évolutionnaire

```
 $t \leftarrow 0$   
Initialiser la population  $P(t)$   
Évaluer  $P(t)$   
Tant que (le critère d'arrêt n'est pas atteint) Faire  
   $t \leftarrow t + 1$   
   $P'(t) \leftarrow$  Sélectionner-parents( $P(t)$ )  
  Reproduire( $P'(t)$ )  
  Evaluer( $P'(t)$ )  
   $P(t + 1) \leftarrow$  Sélectionner-suivants( $P(t), P'(t)$ )  
Fin Tant que
```

2.4.2 Algorithmes génétiques

Les algorithmes génétiques sont inspirés du processus Darwinien d'évolution des espèces qui fait intervenir des mécanismes de sélection naturelle et de reproduction. La sélection permet au individus les mieux adaptés de survivre et de se reproduire. La reproduction permet le brassage des gènes parentaux. Ce phénomène, réitéré sur une population nombreuse, produit après plusieurs générations des individus très bien adaptés au milieu dans lequel ils vivent. Ces algorithmes sont itératifs dont le but est d'optimiser une fonction appelée 'fonction d'adaptation' ou 'fonction objectif' qui associe un coût à chaque solution. Ils sont fondés sur les principes suivants [Benachba, 2005] :

- Une représentation chromosomique des solutions du problème.
- Une méthode pour générer une population initiale de solutions.
- Une fonction d'évaluation qui joue le rôle de l'environnement, elle classe les solutions en fonction de leurs aptitudes.
- La reproduction qui consiste en des opérateurs génétiques qui définissent la manière dont les caractéristiques des parents sont transmises aux descendants

Les algorithmes génétiques s'appuient alors sur trois opérations de base [Benbouzid Sitayeb, 2005] :

- **La sélection** : Elle permet de favoriser les individus qui ont une meilleure fitness (souvent, c'est la valeur de la fonction objectif de la solution associée à l'individu).
- **Le croisement** : Dans cette opération, on combine deux solutions parents pour former un ou deux enfants (offspring) en essayant de conserver les bonnes caractéristiques des solutions parents.
- **La mutation** : Elle permet d'ajouter de la diversité à la population en mutant certaines caractéristiques (gènes) d'une solution.

Le principe générale de fonctionnement d'un algorithme génétique classique est illustré par la figure 2.2.

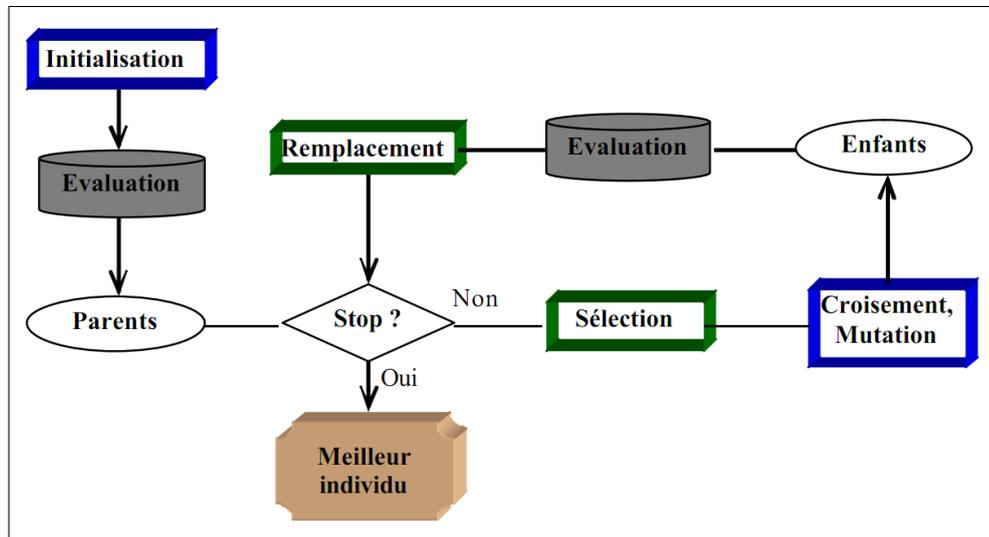


Figure 2.2: Démarche d'un algorithme génétique
[Ghali, 2005]

• Avantages et inconvénients des algorithmes génétiques

Il est évident que les algorithmes génétiques sont particulièrement adaptés à de nombreuses applications liées à des problèmes d'optimisation dans plusieurs domaines. On peut citer les avantages suivants de ce type d'algorithmes [Moutairou, 2009; Whitley, 1994; Yoon *et al.*, 1994].

- La recherche de l'optimum est globale
- Ces algorithmes opèrent d'une manière parallèle (traitement de plusieurs données en parallèle).
- Ils peuvent manipuler des entités arbitraires.
- Ils nécessitent un minimum d'information à priori.
- Ils font un bon équilibre entre l'exploration et l'exploitation de l'espace de recherche.
- Du fait de leur caractère aléatoire, ils offrent une certaine souplesse dans la recherche de la meilleure combinaison. Pour le même problème et à l'aide des mêmes variables d'entrée, on peut obtenir différents jeux de paramètres après plusieurs essais successifs.

Cependant, l'utilisation des algorithmes génétiques pose certains problèmes. On peut citer les limites suivantes [Moutairou, 2009; Whitley, 1994; Yoon *et al.*, 1994].

- La qualité des résultats obtenus est tributaire de l'ajustement des paramètres de l'algorithme (à savoir, les probabilités de croisement et de mutation) qui est souvent empirique.
- Les algorithmes génétiques convergent rapidement vers une solution locale dans les premiers stades, mais la convergence vers une solution optimale peut nécessiter plus de temps.
- La taille du chromosome peut être très grande, et donc, l'algorithme sera algorithmiquement plus complexe (e.g. le cas d'un codage binaire).

2.4.3 Algorithmes évolutionnaires et systèmes complexes adaptatifs

Dans cette section, on étudie les caractéristiques des AEs les permettant de modéliser les CASs.

- **Adaptation** : L'adaptation des individus en utilisant l'opération de reproduction est un point clé du point de vue modélisation des CASs. Dans un AE, on trouve que les conditions pour que le système soit adaptatif vues dans la section 1.3.5 sont satisfaites :
 - l'existence de la notion de succès (techniquement connu sous le nom de 'fitness') et d'échec pour le système dans son environnement.
 - l'existence d'une source de variation dans les détails du système : due aux opérateurs de reproduction.
 - l'existence d'un processus de sélection au niveau des variations offertes pour l'amélioration de la fitness du système.
- **Complexité** : On ne peut pas considérer les AEs comme des systèmes complexes adaptatifs que si la reproduction se fait à base de plusieurs individus de la population, et donc il y aura une interaction entre les éléments du système. Cette propriété est présente dans la majorité des variantes des AEs, notamment les AGs et l'évolution différentielle, qu'on verra en détail dans le chapitre 4. C'est cette interaction entre éléments d'un AE qui mène à l'apparition de la complexité au niveau global du système.
- **Émergence** : Dans un AE, l'émergence du phénomène de haut niveau qui s'agit de la solution du problème à résoudre apparaît à partir de comportements de base appliqués sur les individus (agents) : reproduction et sélection.

2.5 Réseaux de neurones artificiels

Le cerveau humain est constitué d'environ 10^{10} neurones connectés chacun à 10^4 neurones. Ce système organique permet une bonne adaptation aux tâches complexes, il est robuste aux pannes (il continue à fonctionner même en cas de lésion quelque fois importante) et possède des capacités d'apprentissage, de généralisation et d'auto organisation [Changeux, 2009]. A partir de ce système complexe adaptatif naturel, un modèle artificiel s'est inspiré pour accomplir des tâches de calcul dans des environnements ouverts et dynamiques, c'est le Réseau de Neurones Artificiel (RNA).

Essentiellement utilisés pour faire de la classification, les réseaux de neurones peuvent être représentés par une boîte noire à l'entrée de laquelle on présente un vecteur à n dimensions, représentant les données du problème, et à la sortie de laquelle on récupère un vecteur de dimension m qui représente la solution déterminée par le système [Balmisse, 2002]. Une définition plus formelle des réseaux de neurones est la suivante.

Définition 2.3 (Réseau de neurones) *Un réseau de neurones est un processus distribué parallèle qui a la propriété de stocker de la connaissance et la rendre disponible à l'utilisation. Les RNAs ressemblent au cerveau humain dans deux aspects : 1. La connaissance est acquise grâce à un processus d'apprentissage. 2. Les poids des connexions inter-neuronales, ou poids synoptiques sont utilisés pour stocker la connaissance [Haykin, 1994].*

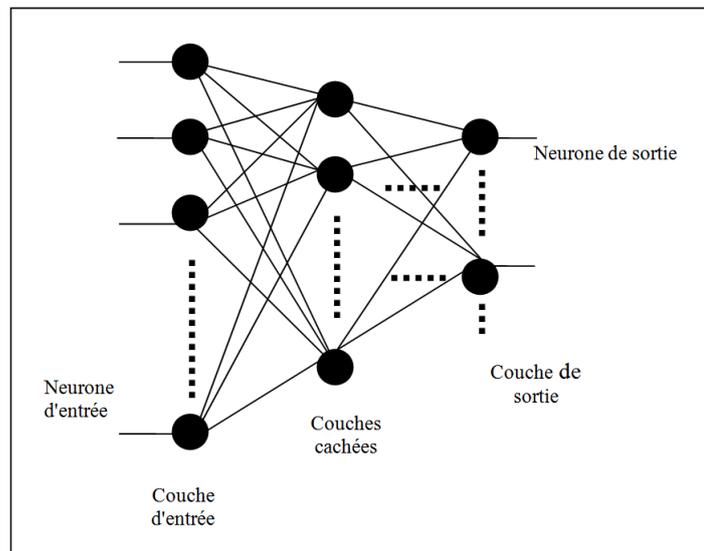


Figure 2.3: Réseau de neurones artificiels

Dans un RNA, chaque élément de traitement a une sortie unique branchée à plusieurs connexions collatérales, qui transmettent le même signal, le signal de sortie du neurone.

Tout le traitement effectué dans un élément de traitement doit être complètement local, c'est-à-dire, il ne dépend que des valeurs courantes des signaux d'entrée et des données stockées dans ce même local. On peut définir donc un réseau de neurones comme un ensemble de neurones associés en couches et fonctionnant en parallèle. L'information donnée au réseau (soit binaire ou réelle) va se propager couche par couche, de la couche d'entrée à la couche de sortie, en passant soit ; par : aucune, une ou plusieurs couches intermédiaires (dites couches cachées) [Meghriche, 2008].

Les deux principales méthodes d'apprentissage dans les RNAs sont l'apprentissage non supervisé et l'apprentissage supervisé [Milgram, 1993]. Les neurones peuvent être organisés de différentes manières, c'est ce qui définit l'architecture et le modèle du réseau. L'architecture la plus courante est celle du perceptron multicouche [Balmisse, 2002].

Le grand avantage des réseaux de neurones réside dans leur capacité d'apprentissage automatique, ce qui permet de résoudre des problèmes sans nécessiter l'écriture de règles complexes, tout en étant tolérant aux erreurs [Balmisse, 2002]. Ces deux caractéristiques, simplicité de règles gérant les éléments et tolérance aux erreurs, font du réseau de neurones un bon modèle pour l'exploitation des systèmes complexes adaptatifs. Cependant, le fait qu'un réseau de neurone est une boîte noire, ça ne permet pas de déterminer la cause d'erreurs dans de tels systèmes. Ça va limiter la possibilité de maintenance d'un tel système artificiel.

2.5.1 Réseaux de neurones et systèmes complexes adaptatifs

En analysant les réseaux de neurones artificiels, on trouve qu'ils sont très proche d'être des CASs. Ils contiennent les éléments suivants d'un CAS.

- **Agents** : Ce sont les neurones du réseau.
- **Voisinage** : Le voisinage d'un neurone (agent) dans un RNA est l'ensemble des neurones connectés à ce neurone.
- **Environnement** : Les nouvelles connaissances, et plus spécifiquement les exemples d'apprentissage sont la base de l'environnement du système globale. Aussi, la notion de biais existante dans certaines architectures des RNAs modélise bien l'aspect aléatoire de l'environnement.
- **Complexité, adaptation et émergence** : La complexité est une caractéristique inhérente dans un RNA : à partir d'interactions très simples entre neurones (gérées par des règles simples), le réseau génère un comportement complexe, sous la forme de reconnaissance ou de classification. Ce comportement globale émerge comme un

phénomène inexplicable des phénomènes locaux vu que le RNA est une boîte noire. Les RNAs sont souvent adaptatifs, étant guidé par les réponses désirées sous forme d'exemples, le RNA adapte les poids des connexions existantes entre neurones, et donc adapte son comportement global.

2.6 Conclusion

Dans ce chapitre, on a présenté quelques modèles artificiels des systèmes complexes adaptatifs : les systèmes multi-agents, les automates cellulaires, les algorithmes évolutionnaires et les réseaux de neurones artificiels. Ces modèles ont été analysés selon une perspective des systèmes complexes adaptatifs à travers la recherche des caractéristiques des CASs dans ces systèmes.

Dans le prochain chapitre, les automates cellulaires apprenant seront présentés. En plus du modèle de base de ce type d'automates, de ses différentes variantes et ses applications, la puissance de ce modèle pour l'étude et la mise en œuvre des CASs sera décrite.

Deuxième partie

Éléments de base pour les modèles proposés

Chapitre 3

Automates cellulaires apprenants

“Learning by experience often is painful- and the more it hurts, the more you learn.” - Ralph Banks

Sommaire

3.1	Introduction	45
3.2	Apprentissage par renforcement	45
3.3	Automates apprenants	47
3.3.1	Automate apprenant à structure fixe	48
3.3.2	Automate apprenant à structure variable	48
3.3.3	Signal de renforcement dans un VSLA	49
3.4	Automates cellulaires apprenants	50
3.5	Automates cellulaires apprenants irréguliers	51
3.6	LCAs et systèmes complexes adaptatifs	52
3.7	Conclusion	52

3.1 Introduction

Après avoir introduit dans la première partie les systèmes complexes adaptatifs et avoir présenté quelques modèles artificiels de ces systèmes, notamment les automates cellulaires et les algorithmes évolutionnaires, on détaille dans ce chapitre et le chapitre qui le suit deux autres modèles artificiels pour la mise en œuvre de ces systèmes. Il s'agit des automates cellulaires apprenants (en anglais : Learning Cellular Automata, LCA) et l'évolution différentielle respectivement.

Les automates cellulaires apprenants sont le résultat d'une hybridation entre deux paradigmes de recherche : les automates cellulaires d'une part et les automates apprenants d'une autre part. Dans la section 2.3 du chapitre 2, les automates cellulaires ont été présentés tout en citant leurs points forts du point de vue modélisation des CASs. Malgré ses avantages, ce modèle reste limité et souvent non adéquat pour fonctionner dans un environnement ouvert d'où la nécessité de le doter de la capacité d'adaptation pour le permettre de survivre face aux changements de l'environnement externe. Cette adaptabilité a été trouvée dans les automates apprenants (en anglais : Learning Automata, LA) qui se basent sur l'utilisation de l'apprentissage par renforcement.

L'LCA a été présenté pour la première fois comme étant un modèle d'émergence par Fei et Hironori [1996], Dans Fei *et al.* [2001], les LCAs ont été considérés comme un bon modèle artificiel des CASs. Cependant, les auteurs de ces deux travaux ont concentré sur le côté applicatif du modèle pour la résolution de problèmes d'optimisation et n'ont pas fourni assez d'analyse de ce modèle comme étant un CAS. Dans leur proposition d'un LCA pour la post-classification des images satellitaires, Mojaradi *et al.* [2004] ont bien rapproché les LCAs au domaine des CASs, en considérant les LCAs comme étant des CASs.

Ce chapitre a comme objet de présenter les notions de bases des LCAs et les étudier d'un point de vue modélisation des CASs. Pour ce faire, on devra d'abord aborder deux approches de base constituant les LCAs, ce sont l'apprentissage par renforcement et les automates apprenants. Ensuite, quelques applications des LCAs pour la résolution de problèmes académiques et réels sont présentées. Finalement, une section est dédiée pour l'analyse des LCAs du point de vue modélisation des CASs.

3.2 Apprentissage par renforcement

L'apprentissage par renforcement est un type d'apprentissage qui permet de trouver, par un processus d'essais et d'erreurs, l'action optimale à effectuer pour chacune des situations que l'apprenant va percevoir afin de maximiser ses *récompenses*. C'est une méthode

d'apprentissage orientée objectif qui va conduire à une situation optimale pour la tâche spécifiée en utilisant un système basé récompenses. L'apprentissage par renforcement est une méthode particulièrement adaptée à l'IA. Le problème de l'apprentissage par renforcement pour un agent se définit en utilisant les éléments suivants [Filliat, 2004] :

- Un ensemble d'états S correspondant à la perception que l'agent a de l'environnement,
- Un ensemble d'actions possibles A , et
- Une fonction de récompense $R : S, A \rightarrow \mathbb{R}$

L'agent va interagir avec son environnement par pas de temps discrets, en percevant l'état de l'environnement s_t , en choisissant une action a_t en fonction de cet état et en recevant la récompense r_{t+1} associée (Figure 3.1)

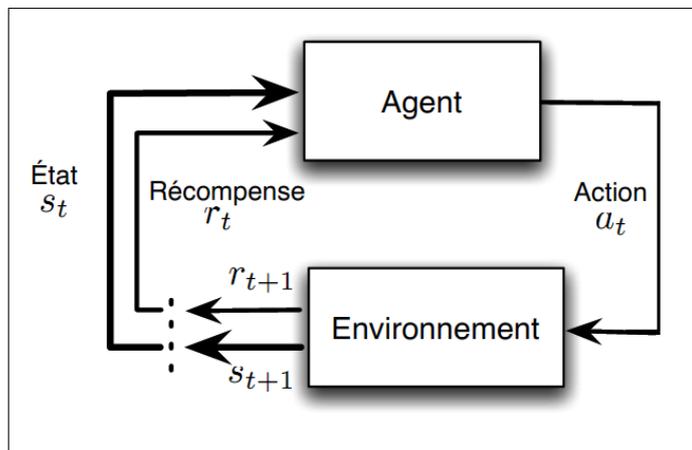


Figure 3.1: Apprentissage par renforcement
[Gagné, 2010]

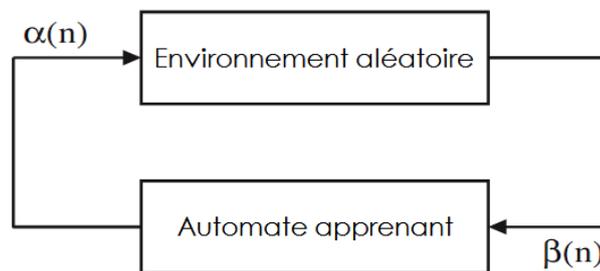
Dans la majorité des cas intéressants et difficiles, les actions de l'agent ne vont pas seulement influencer la récompense immédiate de l'environnement, mais elles vont changer ce dernier et donc, elles vont changer les futures récompenses. Cette caractéristique de l'apprentissage par renforcement et le principe d'essais et d'erreurs sont les deux particularités de ce type d'apprentissage [Martinez *et al.*, 2010].

Un avantage majeur de l'apprentissage par renforcement, en le comparant aux autres types d'apprentissage, est qu'on a pas besoin d'information de l'environnement à part le signal de renforcement. Cependant, un système d'apprentissage par renforcement est plus lent que les autres approches pour la majorité des applications : chaque action doit être testée un nombre de fois afin d'attendre une performance satisfaisante [Unsal, 1998].

Au cours des 50 dernières années, l'apprentissage par renforcement a été largement utilisé pour résoudre des problèmes d'apprentissage. On cite les domaines d'application suivants [Garcia, 2011] : les jeux (les dames américaines, Backgammon, Tetris), la robotique (apprentissage de tâches d'assemblage, robotique mobile, navigation) et l'industrie (routage de paquets, contrôle et maintenance de machines, conduite de systèmes agricoles).

3.3 Automates apprenants

Un automate apprenant (LA) est un outil d'optimisation puissant qui a été développé comme étant un modèle de systèmes apprenants capables d'apprendre dans un environnement stochastique ouvert et inconnu. Un LA essaie de déterminer, d'une manière itérative, l'action optimale à appliquer dans un environnement donné à partir d'un ensemble d'actions à sa disposition. L'environnement renvoie à l'automate un signal de renforcement reflétant la qualité relative de l'action effectuée par l'automate (Figure 3.3). Selon ce signal, l'automate ajuste son comportement à travers un algorithme d'apprentissage [Rastegar *et al.*, 2004]. Avant de détailler la politique de comportement des LAs, on donne les définitions suivantes d'un LA.



$\alpha(n)$: l'action de l'automate, $\beta(n)$: la récompense ou la pénalité de l'environnement

Figure 3.2: Automate apprenant et son environnement

[Akbari Torkestani et Meybodi, 2011]

Définition 3.1 (Automate apprenant) *Un automate apprenant est un automate qui interagit avec un environnement aléatoire en essayant d'apprendre l'action optimale offerte par cet environnement [Papadimitriou et al., 2004].*

Définition 3.2 (Automate apprenant) *Un automate apprenant est une unité de décision qui améliore ses performances en apprenant à choisir l'action optimale à partir d'un*

ensemble fini d'actions autorisées à travers une interaction itérative avec un environnement aléatoire, via un processus d'apprentissage [Rastegar et al., 2004].

Dans le cadre de cette thèse on s'intéresse à cette deuxième définition car la notion d'environnement aléatoire est cruciale dans les systèmes complexes adaptatifs.

Les automates apprenants peuvent être classés en deux catégories de base : automates apprenants à structure fixe et automates apprenants à structure variable. Dans le premier type, les matrices (fonctions) de transition et de sorties sont toutes les deux constantes dans le temps, alors que dans le deuxième elles changent avec le temps [Mojaradi et al., 2004].

3.3.1 Automate apprenant à structure fixe

Formellement, un LA à structure fixe (en anglais : Fixed Structure Learning Automaton, FSLA) est décrit comme suit [Mojaradi et al., 2004].

Il s'agit d'un quintuplet $A = \{\alpha, \beta, F, G, \varphi\}$ avec : $\alpha = \{\alpha_1, \dots, \alpha_r\}$ est un ensemble de r actions offertes par l'environnement parmi lesquelles l'automate doit choisir. $\beta = \{0, 1\}$ est l'ensemble d'entrée provenant de l'environnement sous forme de récompense ou de pénalité, φ est l'ensemble d'états internes de l'automate, F est la fonction de mise à jour de l'état interne de l'automate selon l'état courant de l'automate et la pénalité ou récompense provenant de l'environnement et G est l'action choisie par l'automate en se basant sur le nouveau état de l'automate.

3.3.2 Automate apprenant à structure variable

Un LA à structure variable (en anglais : Variable Structure Learning Automaton, VSLA) utilise des matrices de transition et de sortie qui changent dans le temps [Mojaradi et al., 2004]. Formellement, c'est un quadruplet :

$$A = \{\alpha, P, b, T\} \tag{3.1}$$

Où :

$\alpha = \{\alpha_1, \dots, \alpha_r\}$ est un ensemble de r actions offertes par l'environnement parmi lesquelles l'automate doit choisir.

$P = [p_1(n) \dots p_r(n)]$ est le vecteur de probabilités d'actions, tel que p_i est la probabilité de choisir l'action α_i à l' n -ème point de temps.

$\beta = \{0, 1\}$ est l'ensemble d'entrées provenant de l'environnement, où 0

représente une récompense et 1 représente une pénalité.

$T : P \times \beta \rightarrow P$ est la fonction de mise à jour. Elle spécifie l'algorithme de mise à jour des probabilités d'actions selon les entrées provenant de l'environnement. C'est le signal de renforcement de l'LA.

3.3.3 Signal de renforcement dans un VSLA

Après avoir effectué une action α_i , sa probabilité est mise à jour en se basant sur la récompense $r \in \{0, 1\}$ provenant de l'environnement et ceci selon la première formule du système d'équations 3.2. Les autres probabilités p_j (pour toute action α_j $j \neq i$) sont ajustées d'une façon qui garantit que la somme de probabilités soit égale à 1 ($\sum_i p_i = 1$), comme montré par la deuxième formule du système d'équations 3.2.

$$\begin{cases} p_i \leftarrow p_i + a * r(1 - p_i) - b(1 - r)p_i \\ p_j \leftarrow p_j - a * r * p_j + b(1 - r)(\frac{1}{n-1} - p_j), \quad \forall j \neq i. \end{cases} \quad (3.2)$$

Les paramètres a et b sont respectivement les taux de récompense et de pénalité. Dans la littérature des LAs, trois schémas de renforcement sont définis selon les valeurs a et b [Martinez *et al.*, 2010] :

1. Automate avec récompense-inaction linéaire (en anglais : Linear Reward-Inaction, L_{R-I}) pour $b = 0$.
2. Automate avec récompense-pénalité linéaire (en anglais : Linear Reward-Penalty, L_{R-P}) pour $a = b$.
3. Automate avec récompense- ϵ -pénalité (en anglais : Linear Reward- ϵ -Penalty, $L_{R-\epsilon P}$) pour $b \ll a$.

Le pseudo-code 3.1 résume le principe de fonctionnement d'un automate apprenant à structure variable [Rezvanian et Meybodi, 2010].

Les LAs ont l'avantage de pouvoir bien agir dans des situations dans lesquelles l'information sur l'environnement est incomplète, i.e, l'automate ne peut pas prévoir la réaction de son l'environnement. Ils sont aussi très efficaces dans des environnements complexes, dynamiques et aléatoires ayant un grand degré d'incertitude [Akbari Torkestani et Meybodi, 2011].

Les automates apprenants ont été utilisés avec succès dans plusieurs applications. On peut citer : la résolution de problèmes NP-Complets [Oommen et de St. Croix, 1996], les réseaux mobiles [Beigy et Meybodi, 2005], l'allocation de ressources [Oommen et Roberts, 2000], et la conception de réseaux de neurones [Meybodi et Beigy, 2002].

Algorithme 3.1 Automate apprenant à structure variable

Initialiser le vecteur de probabilités d'actions $P : P \leftarrow [1/r, 1/r, \dots, 1/r]$,

{ r est le nombre d'actions offertes à l'automate.}

Tant que (le critère d'arrêt n'est pas atteint) **Faire**

 Sélectionner une action α_i selon une probabilité p_i .

 Évaluer cette action et renvoyer à l'automate un signal de renforcement β .

 Mettre à jour le vecteur de probabilités d'actions P selon l'algorithme d'apprentissage.

Fin Tant que

3.4 Automates cellulaires apprenants

Un automate cellulaire apprenant (en anglais : a Learning Cellular Automaton, LCA) est un automate cellulaire dans lequel à chaque cellule est associé un ou plusieurs automates apprenants définissant sa dynamique [Hariri *et al.*, 2005]. Dans un LCA, les LAs voisins d'un LA donné constituent son environnement local. Les vecteurs de probabilités d'actions des LAs voisins changent au fil du temps (durant l'évolution de l'LCA), et donc, l'environnement locale est non stationnaire. Ceci modélise bien la co-evolution d'agents d'un CAS avec son environnement local. Formellement, un LCA d -dimensionnel est un quintuplet [Akbari Torkestani et Meybodi, 2011] :

$$Aut = (Z^d, \Phi, A, N, f) \tag{3.3}$$

Où :

Z^d est un treillis (une grille) de d -tuples de nombres entiers.

Φ est un ensemble fini d'états.

A est l'ensemble des automates apprenants assignés aux cellules de l'automate cellulaire.

$N = \{\bar{x}_1, \bar{x}_2, \dots, \bar{x}_m\}$ est un sous-ensemble fini de Z^d appelé le vecteur de voisinage, où : $\bar{x}_i \in Z^d$.

$f : \Phi^m \rightarrow \beta$ est la règle locale de l'LCA, où : β est l'ensemble de valeurs que le signal de renforcement peut prendre. Elle donne le signal de récompense (renforcement) à chaque LA en se basant sur les actions courante choisies par les LAs voisins.

Les LCAs ont été utilisés pour résoudre des problèmes dans plusieurs domaines tels que : le traitement d'images [Meybodi *et al.*, 2000], la modélisation de la diffusion de rumeurs [Meybodi et Taherkhani, 2001], la modélisation des réseaux de commerces [Meybodi et Khojasteh, 2001] les réseaux cellulaires [Beigy et Meybodi, 2003], les réseaux

mobiles [Meybodi et Mehdipour, 2003] et pour la résolution des problèmes NP-difficiles [Enami Eraghi *et al.*, 2011].

Les LCAs peuvent être classés en deux catégories : les LCAs synchrones (SLCA) et les LCAs asynchrones (ALCA). Dans le premier type, LCA synchrones, toutes les cellules sont synchronisées par une horloge globale et s'exécutent (prennent actions) au même temps. Donc, dans un LCA synchrone, tous les LAs (dans toutes les cellules) sont activés au même temps et en parallèle. Dans la deuxième catégorie, LCAs asynchrones, à un moment donné, seulement quelques LAs sont activés indépendamment les uns des autres [Beigy et Meybodi, 2004]. Le principe de fonctionnement d'un LCA est donné par le pseudo-code 3.2 [Abin *et al.*, 2008].

Algorithme 3.2 Automate cellulaire apprenant (LCA)

1. Spécifier l'état interne de chaque cellule de l'automate à la base des vecteurs de probabilités d'actions des LAs résidants dans ces cellules.
{ l'état initiale peut être choisi aléatoirement ou à la base l'expérience de l'automate. }
 2. Déterminer le signal de renforcement des LAs pour chaque cellule selon la règle de l'AC.
 3. Chaque LA met à jour son vecteur de probabilités d'actions selon le signal de renforcement reçu et l'action choisie.
 4. répéter le processus de 1 à 3 jusqu'à la satisfaction d'une condition d'arrêt.
-

3.5 Automates cellulaires apprenants irréguliers

Dans ce type d'LCAs, la condition que la grille de cellules soit rectangulaire est enlevée. ceci a été motivé par le fait qu'il existe des applications dont les agents du système ne peuvent pas être modélisés sous forme rectangulaire, par exemple : les réseaux sans fil et les applications basées théorie des graphes. Un LCA irrégulier est défini comme un graphe non orienté dans lequel chaque sommet représente une cellule équipée d'un automate apprenant. Le voisinage dans ce type d'LCAs est constitué de l'ensemble des sommets liés par des arrêtes à une cellule (sommet) donnée [Esnaashari et Meybodi, 2008; Akbari Torkestani et Meybodi, 2011].

Les LCAs sont plus performants que les ACs et les LAs pour les raisons suivantes. Un LCA est capable d'apprendre de son expérience, chose qui n'est pas possible dans un AC traditionnel. Aussi, il permet une interaction intensive entre les agents constituants du système ce qui donne lieu à l'apparition de phénomènes émergents, et donc, à une intelligence collective, ce qui le rend meilleur par rapport aux LAs individuels dans le contexte des CASs [Beigy et Meybodi, 2008].

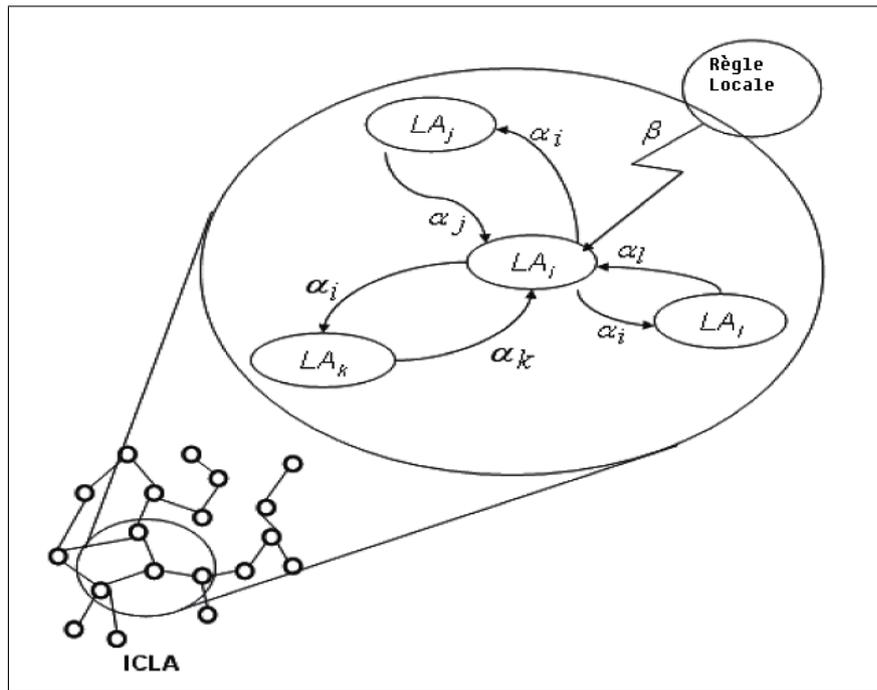


Figure 3.3: Automate cellulaire apprenant irrégulier
[Esnaashari et Meybodi, 2008]

3.6 LCAs et systèmes complexes adaptatifs

Dans le chapitre 2 on a introduit les atouts des automates cellulaires du point de vue modélisation et mise en oeuvre des CASs (Section 2.3.2 et tableau 2.1). Cependant, le modèle des automates cellulaires conventionnels n’exploite pas la notion d’environnement que localement : le voisinage de chaque cellule constitue son environnement local, mais l’automate comme un tout est totalement isolé de son environnement. Ceci limite les capacités d’adaptation ou d’apprentissage de l’automate dans un environnement ouvert.

Cette limite est réglée dans l’automate cellulaire apprenant quantique : l’environnement et son interaction avec la grille de cellule constituent la base du modèle. Cette intégration de la notion d’environnement à l’automate permet bien la simulation des CASs naturels, où le changement d’information et d’énergie entre l’environnement et le système jouent un rôle primordial pour la survie de ce dernier.

3.7 Conclusion

Dans ce chapitre, les automates cellulaires apprenants ont été présentés. Ce type d’automates présente une bonne simulation des systèmes complexes adaptatifs vu qu’il se base non seulement sur les interactions entre les cellules avec leurs voisines, mais aussi sur l’in-

teraction de l'automate comme un tout avec son environnement externe, d'où l'existence de la co-évolution de l'automate avec son environnement. En plus des concepts de base des LCAs, une discussion de leur capacité de simuler les CASs a été abordée.

Dans le chapitre suivant, on présente un autre modèle artificiel pour les CASs. Il s'agit de l'évolution différentielle. Le principe de base de ce modèle, ses points forts et limites et ses variantes seront présentés. De plus, le comportement global du système à évolution différentielle autant que CAS sera évalué.

Chapitre 4

Évolution différentielle

“If evolution really works, how come mothers only have two hands?” -
Milton Berle

Sommaire

4.1	Introduction	56
4.2	Principe de l'évolution différentielle	56
4.2.1	Mutation différentielle	57
4.2.2	Croisement différentiel	57
4.2.3	Sélection	58
4.2.4	Critère d'arrêt	58
4.3	Variantes pour l'évolution différentielle	59
4.4	Avantages et limites de l'évolution différentielle	60
4.4.1	Avantages de l'évolution différentielle	60
4.4.2	Limites de l'évolution différentielle	61
4.5	Évolution différentielle et systèmes complexes adaptatifs	61
4.6	Conclusion	62

4.1 Introduction

L'Évolution Différentielle (ED) est un type d'Algorithmes Evolutionnaires (AEs) inventé par Storn et Price en 1995. C'est un algorithme d'optimisation globale simple mais très efficace. L'ED a montré des performances meilleures que celles des AEs traditionnels soit pour des fonctions benchmarks ou pour des applications réelles [Yang *et al.*, 2007]. L'ED combine deux mécanismes de recherche : l'évolution naturel d'une part et la génération de directions multiples de recherche à base de la distribution des solutions dans la population en cours d'une autre part [Mezura-Mentes et Palmeque-Ortiz, 2009].

Dans ce chapitre, les principes de base de l'évolution différentielle seront présentés. Ensuite, quelques variantes de l'ED seront introduites. Les avantages et les limites des algorithmes à évolution différentielle sont discutés. Finalement, on analyse le modèle à évolution différentielle du point de vue modélisation et mise en œuvre des CASs.

4.2 Principe de l'évolution différentielle

L'idée de base de cet algorithme est l'utilisation d'une nouvelle stratégie de mutation. Elle consiste en l'ajout de la différence vectorielle pondérée entre deux individus à un troisième. Après l'application de cette mutation, un croisement entre l'individu mutant et l'individu correspondant de la population aura lieu. Finalement, une sélection gloutonne entre ces deux individus, le père et le fils, va décider lequel des deux individus survivra dans la prochaine génération. Dans sa version de base, il existe trois paramètres stratégiques de l'évolution différentielle : la taille de la population NP , le taux de croisement CR et le facteur de différence F .

Pour chaque génération du processus d'évolution, chaque individu, qu'on appelle *individu cible*, rivalise avec un nouveau individu, appelé *individu essai*, pour survivre. Dans la prochaine génération le meilleur des deux, ayant la meilleure fitness, survivra. L'individu essai est créé en combinant l'individu cible avec un autre créé par mutation c'est *l'individu mutant*.

Formellement, les individus dans un Algorithme à Évolution Différentielle (AED) sont représentés par des vecteurs D -dimensionnels de valeurs réelles $x_i \forall i \in 1, \dots, NP$, où : D est le nombre des paramètres à optimiser et NP est la taille de la population. Selon la description de Storn et Price [1997], un AED fonctionne comme décrit dans le pseudo code 4.1.

Dans ce qui suit, on détaille les éléments d'un AED, notamment la mutation et le croisement différentiels, la sélection et le critère d'arrêt.

Algorithme 4.1 Évolution différentielle (ED)

$G \leftarrow 1$

choisir NP , F et CR

$P_G \leftarrow$ initialiser la population aléatoirement

Tant que (le critère d'arrêt n'est pas atteint) **Faire**

Pour (chaque individu $X_{i,G}$ de P_G) **Faire**

 choisir les parents auxiliaires $X_{r_1,G}$, $X_{r_2,G}$ et $X_{r_3,G}$

 créer l'individu enfant $U_{i,G}$ en utilisant la mutation et le croisement différentiels

 appliquer la sélection : $P_{G+1} \leftarrow P_G \cup \text{Meilleur}(U_{i,G}, X_{i,G})$

Fin Pour

$G \leftarrow G + 1$

Fin Tant que

4.2.1 Mutation différentielle

Un mutant v_i correspondant à un individu cible x_i est créé à partir de trois autres individus choisis aléatoirement de la population selon l'équation 4.1 [Yang *et al.*, 2007; Das *et al.*, 2009; Das et Suganthan, 2011]. Le principe de cette mutation est illustré dans la figure 4.1.

$$v_i = x_{i_1} + F \cdot (x_{i_2} - x_{i_3}) \quad (4.1)$$

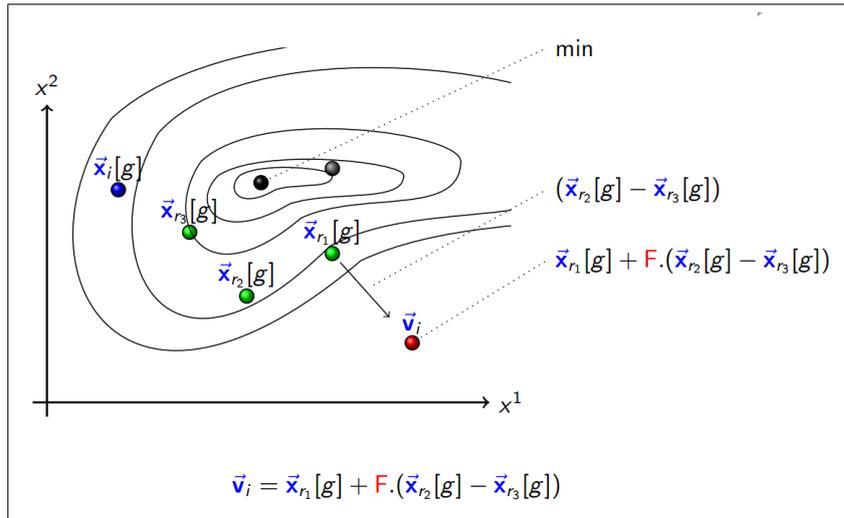
Où : $i, i_1, i_2, i_3 \in \{1, \dots, NP\}$ sont des entiers mutuellement différents, et $F > 0$ est un facteur constant réel positif servant à contrôler la variation différentielle $d_i = x_{i_2} - x_{i_3}$.

4.2.2 Croisement différentiel

Le croisement dans le cadre d'un AED permet de générer un individu essai u_i à partir de la combinaison entre un mutant v_i et l'individu cible x_i . Ceci est fait selon le processus décrit par l'équation 4.2 [Yang *et al.*, 2007; Das *et al.*, 2009; Das et Suganthan, 2011].

$$u_i(j) = \begin{cases} v_i(j), & \text{si } U_j(0, 1) < CR \\ x_i(j), & \text{sinon} \end{cases} \quad (4.2)$$

Où : $U_j(0, 1)$ est un nombre aléatoire entre 0 et 1, et $CR \in [0, 1]$ est le taux de croisement.


Figure 4.1: Mutation différentielle

[Dutrech, 2010]

En plus de ce type de croisement appelé *croisement binomial*, il existe un autre type fréquemment utilisée dans des variantes plus récentes de l'ED que celle de Storn et Price, c'est le *croisement exponentiel*. Dans ce type de croisement, à partir d'une position choisie aléatoirement, L composants successives de l'individu mutant, sont transmis à l'individu d'essai [Das et Suganthan, 2011]. Ces deux types de croisement sont illustrés par la figure 4.2, croisement binomial à gauche et croisement exponentiel à droite.

4.2.3 Sélection

Comme on l'a déjà mentionné, la sélection permet de choisir un individu parmi l'individu cible x_i et l'individu essai u_i selon la qualité de leurs fitnesses comme indiqué dans l'équation 4.3 [Yang *et al.*, 2007; Das *et al.*, 2009; Das et Suganthan, 2011].

$$x'_i = \begin{cases} u_i, & \text{si } f(u_i) < f(x_i) \\ x_i, & \text{sinon} \end{cases} \quad (4.3)$$

Où : x'_i est l'enfant remplaçant x_i dans la prochaine génération.

4.2.4 Critère d'arrêt

Le critère d'arrêt d'un AED est généralement l'un des suivants.

1. Un nombre fixe d'itérations G_{max} avec une valeur largement suffisante à la résolution du problème en question.

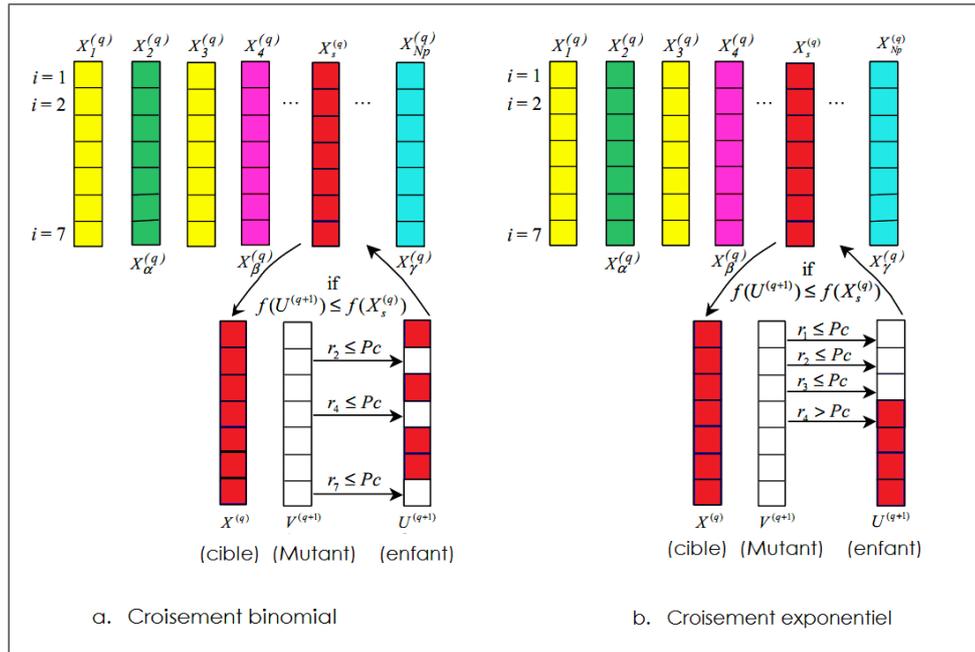


Figure 4.2: Croisement différentiel : binomial à gauche, exponentiel à droite
 [Purcina et Saramago Pereira, 2008]

2. L'absence d'une évolution remarquable dans la valeur de la fitness de la population pour un certain nombre d'itération successives.
3. L'obtention de la qualité requise de la fonction d'adaptation.

4.3 Variantes pour l'évolution différentielle

L'algorithme de Storn et Price [1997] décrit ci-dessus est connu sous le nom de DE/rand/1. Avec l'utilisation d'un croisement binomial, cette variante est appelée DE/rabd/1/bin. De cette façon, les différentes variantes d'AEDs sont nommées. La convention générale est DE/x/y/z, où : 'x' est le vecteur base à être perturbé, 'y' est le nombre de vecteurs de différences utilisés dans la perturbation de 'x', et 'z' est une abréviation du type de croisement utilisé (exp :exponentiel, bin : binomial).

Les variantes d'AEDs diffèrent généralement entre eux par le type de mutation utilisé. En plus de la mutation de base de leur algorithme, Storn et Price [1997] proposent quatre autres schémas de mutation présentés par les équations 4.4 à 4.7.

$$\text{"DE/best/1"} : \vec{V}_{i,G} = \vec{X}_{best,G} + F \cdot (\vec{X}_{r_1,G} - \vec{X}_{r_2,G}) \quad (4.4)$$

$$\begin{aligned} \text{"DE/target-to-best/1"} : \quad \vec{V}_{i,G} &= \vec{X}_{i,G} + F.(\vec{X}_{best,G} - \vec{X}_{i,G}) \\ &+ F.(\vec{X}_{r_1^i,G} - \vec{X}_{r_2^i,G}) \end{aligned} \quad (4.5)$$

$$\begin{aligned} \text{"DE/best/2"} : \quad \vec{V}_{i,G} &= \vec{X}_{best,G} + F.(\vec{X}_{r_1^i,G} - \vec{X}_{r_2^i,G}) \\ &+ F.(\vec{X}_{r_3^i,G} - \vec{X}_{r_4^i,G}) \end{aligned} \quad (4.6)$$

$$\begin{aligned} \text{"DE/rand/2"} : \quad \vec{V}_{i,G} &= \vec{X}_{r_1^i,G} + F.(\vec{X}_{r_2^i,G} - \vec{X}_{r_3^i,G}) \\ &+ F.(\vec{X}_{r_4^i,G} - \vec{X}_{r_5^i,G}) \end{aligned} \quad (4.7)$$

Les indices $r_1^i, r_2^i, r_3^i, r_4^i$ et r_5^i sont des entiers mutuellement exclusifs choisis aléatoirement de l'ensemble $\{1, \dots, NP\}$ et ce sont tous différents de l'indice de base i . Ces indices sont générés aléatoirement pour chaque opération de mutation.

En plus de ces variantes de l'AED proposée par Storn et Price, les chercheurs dans le domaine de l'optimisation combinatoire ne cessent de créer de nouvelles variantes d'AEDs. Ces variantes diffèrent généralement par le type de mutation ou de croisement utilisés.

Dans une revue récente de la littérature de l'ED, Das et Suganthan [2011] donnent une analyse très riche de plusieurs variantes d'AED. Ils discutent les instances suivantes : ED avec mutation trigonométrique, ED avec croisement arithmétique, DE/rand/1/Either-Or, ED à base d'opposition, ED avec mutation basée voisinage, ED avec une sélection adaptative de mutation, ED adaptative avec une mutation DE/Current-to-pbest. De plus, ils présentent dans le même article des hybridations entre l'ED et autres meta-heuristiques [Das et Suganthan, 2011].

4.4 Avantages et limites de l'évolution différentielle

4.4.1 Avantages de l'évolution différentielle

Depuis la fin des années 1990, l'ED suscite un vif intérêt dans plusieurs domaines académiques ou industriels pour les raisons suivantes [Das et Suganthan, 2011].

- L'ED est plus simple à implémenter qu'autres approches évolutionnaires. Cette simplicité de programmation est très importante pour des non-informaticiens qui peuvent ne pas être des experts mais cherchent un algorithme simple à programmer et à paramétrer pour la résolution de problèmes d'optimisation rencontrés dans leurs domaines. A noter que malgré que le PSO est aussi simple à programmer et à paramétrer, il reste moins performant que l'ED.

- Malgré sa simplicité, l'évolution différentielle offre des performances remarquablement meilleures que celles offertes par plusieurs autres approches d'optimisation, notamment les AEs et le PSO.
- Le nombre de paramètres de contrôle dans l'AED est bien limité (CR , F et NP dans la version de base) ce qui permet un bon contrôle de l'algorithme. De plus, les effets de ces paramètres sur le comportement de l'algorithme sont bien étudiés. Donc, l'application d'un AED ne va pas causer beaucoup de difficultés pour les programmeurs.
- La complexité spatiale de l'ED est réduite, quand on la compare à d'autres algorithmes performants célèbres tel que le CMA-ES. Cette caractéristique permet d'étendre l'ED pour résoudre des problèmes d'optimisation de grandes dimensions.

4.4.2 Limites de l'évolution différentielle

Malgré son efficacité et sa robustesse, l'ED souffre principalement des limites suivantes.

- La performance d'AED est très sensible des valeurs des paramètres F , CR et NP [Liu et Lampinen, 2005]. Donc, un mauvais choix de ces paramètres peut mener à des résultats médiocres de l'algorithme [Noman *et al.*, 2011].
- Malgré l'existence de quelques recommandations pour le choix des paramètres dans la littérature, ça reste très dépendant du problème à résoudre [Noman *et al.*, 2011].
- L'ED n'est pas complètement loin des problèmes de lenteur de convergence ou de convergence prématurée [Das *et al.*, 2009].
- Comme tout autre AE, la performance d'un AED se détériore avec la croissance de la dimension de l'espace de recherche [Das et Suganthan, 2011].
- Dans, l'évolution différentielle, le concept de voisinage, concept clé dans les stratégies d'évolution est manquant [Yang *et al.*, 2007].

4.5 Évolution différentielle et systèmes complexes adaptatifs

En plus des caractéristiques d'adaptation dues de sa nature évolutionnaire (comme vu dans la section : 2.4.3), un système à évolution différentielle offre les possibilités suivantes du point de vue modélisation des CASs.

- **Complexité** : La reproduction dans un algorithme à évolution différentielle implique l'interaction entre plusieurs individus de la population, quatre dans le cas de

la mutation et deux pour le croisement. Ceci donne une forte inter-connexion entre les agents du système, ce qui résulte à un comportement complexe au niveau global du système.

- **Adaptation** : Plus d'adaptation du modèle est offerte dans quelques une de ses variantes, e.g., ED avec une sélection adaptative de mutation et ED adaptative avec une mutation DE/Current-to-pbest.
- **Voisinage** : La variante de l'ED avec mutation basée voisinage est un bon exemple de la prise en compte de la notion de voisinage, notion clé dans les CASs.

Malgré que quelques variantes de l'ED offrent une bonne modélisation de l'adaptation et du voisinage des CASs, ceci n'est pas offert dans la version de Storn et Price [1997], et donc l'ED reste limitée du point de vue modélisation des CASs. Dans cette optique, notre deuxième contribution (chapitre 8), prend en compte la notion de voisinage comme élément de base du modèle proposé.

4.6 Conclusion

Dans ce chapitre, on a présenté l'évolution différentielle, ses principes et le comportement générique d'un algorithme à évolution différentielle. Aussi, quelques variantes de l'évolution différentielle ont été abordées. Finalement, une d'analyse des propriétés de l'évolution différentielle permettant son exploitation pour la simulation et la mise en œuvre des systèmes complexes adaptatifs a été donnée.

Dans le chapitre suivant, un troisième champ de recherche sur lequel on s'est basé dans nos contributions sera présenté. Il s'agit de l'informatique inspirée du quantique.

Chapitre 5

Informatique inspirée du quantique

“If quantum mechanics hasn’t profoundly shocked you,
you haven’t understood it yet.” - Niels Bohr

Sommaire

5.1	Introduction	65
5.2	Informatique quantique	65
5.2.1	Bit quantique	65
5.2.2	Registre quantique	66
5.2.3	Les cinq principes de l'informatique quantique	67
5.2.4	Calcul quantique et algorithmes quantiques	68
5.2.5	Avantages et limites de l'informatique quantique	70
5.3	Algorithmes évolutionnaires quantiques	70
5.3.1	Individu quantique	71
5.3.2	Mesure quantique d'individus	72
5.3.3	La porte D	72
5.3.4	Discussion de l'algorithme évolutionnaire quantique	73
5.4	Autres algorithmes inspirés du quantique	74
5.4.1	Algorithmes génétiques quantiques	74
5.4.2	Boids de Reynolds quantiques	76
5.4.3	Réseaux de neurones quantiques	76
5.5	Avantages des algorithmes inspirés du quantique	77
5.6	Conclusion	77

5.1 Introduction

L'informatique quantique est un nouveau champs de recherche en informatique dont l'objectif est de tirer profit des possibilités offertes par la physique quantique. Elle se base sur l'utilisation des spécificités de la physique quantique pour le traitement de l'information. La particularité de cette nouvelle voie de recherche réside dans l'utilisation d'un nouveau type de support de stockage et de traitement de l'information : l'ordinateur quantique. Un ordinateur quantique est un nouveau type de machines de calculs caractérisé par sa puissance de calcul grâce aux possibilités de parallélisme offertes par les principes issues de la physique quantique.

Malgré une certaine maturité de la théorie de l'informatique quantique et l'apparition d'algorithmes quantiques remarquablement moins complexes que leurs contreparties classiques pour la résolution de problèmes, une difficulté cruciale pour l'adoption de ce nouveau modèle de traitement de l'information est l'absence d'ordinateurs quantiques puissants jusqu'à nos jours. Les modèles déjà construits ne sont que des modèle préliminaires et ils sont toujours dans des phases expérimentales.

En attendant la réalisation d'ordinateurs quantiques puissants, deux axes de recherche ont vu la jours. Le premier vise la simulation d'algorithmes quantiques sur machine classique. Cette solution réduit les performances des algorithmes quantiques et n'est efficace que pour leur étude. L'autre se base sur l'exploitation des principes de l'informatique quantique, tels que le bit quantique, la superposition d'états et les portes quantiques, par des algorithmes conventionnels afin d'augmenter les performances de ces derniers.

Dans ce chapitre, on donne une brève introduction à l'informatique quantique. Ensuite, on décrit les Algorithmes Evolutionnaires Quantiques (AEQ). Finalement, on donne une petite synthèse sur autres hybridations entre l'informatique quantique et les approches d'optimisation.

5.2 Informatique quantique

Dans cette section, on présente les principes de l'informatique quantique, on donne quelques exemples de ses domaines d'applications, et on présente ses avantages et ses limites.

5.2.1 Bit quantique

Un "bit" représente l'unité élémentaire d'information dans un ordinateur classique. Il peut se trouver dans l'un de deux états, le '1' et le '0', à l'instar d'un interrupteur, ouvert ou fermé. Appelée "qubit" (de l'anglais : QUantum BIT), la version quantique

d'un bit d'information peut être dans les deux états en même temps. Ces états peuvent être représentés par une flèche, qui pointe une position à la surface d'une sphère comme montré dans la figure 5.1 [Deutsch et ekert, 2006; Hey, 1999; Nielsen et Chuang, 2000].

Un qubit est un système physique basé sur deux valeurs logiques distinctes : 0 et 1, ou "oui" et "non". Ces deux valeurs représentent les états de base notés par $|0\rangle$ et $|1\rangle$. L'état du système φ , qui est une superposition des deux états fondamentaux, est donné par l'équation 5.1 [Rieffel et Polak, 2000].

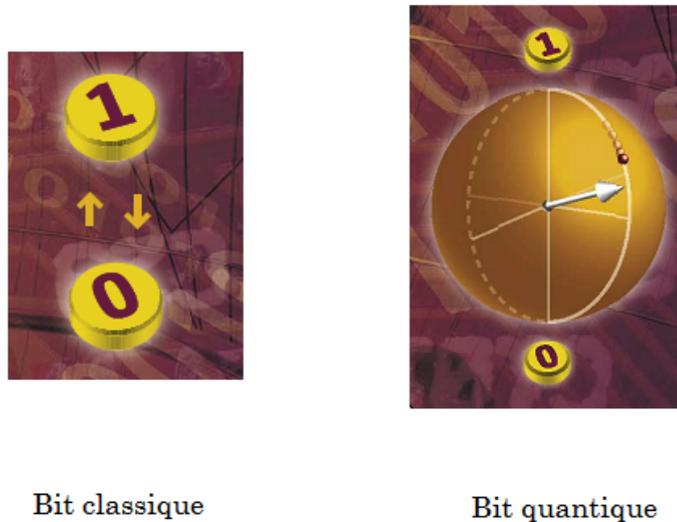


Figure 5.1: Bit classique et bit quantique
[Kempe *et al.*, 2006]

$$\varphi = \alpha|0\rangle + \beta|1\rangle \tag{5.1}$$

L'observation de cet état va donner la valeur 0 avec une probabilité égale à $|\alpha|^2$ et la valeur 1 avec une probabilité égale à $|\beta|^2$, où α et β sont deux nombres complexes appelés *amplitudes* et vérifiant la contrainte donnée par l'équation 5.2.

$$|\alpha|^2 + |\beta|^2 = 1 \tag{5.2}$$

5.2.2 Registre quantique

Un registre quantique est un ensemble de qubits, une superposition arbitraire de n qubits. L'état d'un registre quantique peut être donné par l'équation 5.3 [Deutsch et ekert, 2006; Rieffel et Polak, 2000].

$$\psi = \sum_{x=0}^{2^n-1} C_x |x\rangle \quad (5.3)$$

Les amplitudes C_x gouvernant les probabilités des différents états satisfont la propriété donnée par l'équation 5.4.

$$\sum_{x=0}^{2^n-1} |C_x|^2 = 1 \quad (5.4)$$

Dans un système classique, un registre de n bits peut contenir une seule valeur parmi les 2^n valeurs possibles. Par contre, un registre quantique représente toutes les 2^n valeurs en même temps [Deutsch et ekert, 2006].

• **Mesure quantique**

La mesure de l'état d'un qubit le force à pointer soit vers le '1', soit vers le '0'. Le résultat dépend des amplitudes du qubit. La figure 5.2 donne un exemple de mesure d'un qubit qui aura 70% de chances de se trouver dans l'état '1' et 30% de se trouver dans l'état '0'.

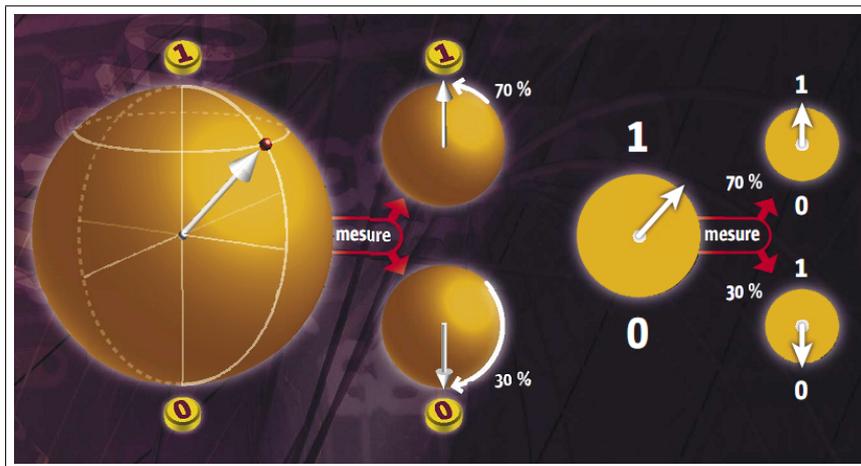


Figure 5.2: Mesure quantique
[Kempe *et al.*, 2006]

5.2.3 Les cinq principes de l'informatique quantique

L'informatique quantique est basée principalement sur les cinq principes suivants qui sont inspirés de la mécanique quantique : la superposition d'états, l'interférence, l'enchevêtrement, le non déterminisme et la non duplication, et la réversibilité [Bellac, 2003].

- **Superposition d'états**

Un qubit peut être dans l'état 0 ou 1, mais également les deux en même temps. Donc, un registre de n qubits pourra contenir 2^n valeurs distinctes. Une conséquence importante de cette propriété, la superposition d'états, est que le système peut manipuler 2^n valeurs en parallèle, ce qui offre une rapidité énorme aux machines quantiques.

- **Interférence**

C'est l'action qui permet d'augmenter ou diminuer la probabilité d'avoir un état. Il existe deux types d'interférence : *interférence constructive* et *interférence destructive* ; l'interférence constructive permet d'augmenter la probabilité d'obtenir un état, alors que l'interférence destructive permet de diminuer la probabilité d'obtenir un état.

- **Enchevêtrement ou intrication**

Deux bits quantiques sont dits enchevêtrés si l'un dépend de l'autre et toute modification de l'un affecte l'autre.

- **Non déterminisme et non duplication**

La mécanique quantique est non déterministe : le hasard quantique est authentique et irréductible, c'est une loi fondamentale. La copie des qubits n'est pas toujours possible. Un état inconnu ne peut pas être dupliqué .

- **Réversibilité**

L'évolution dans le temps d'un système quantique isolé est décrite par une transformation unitaire. En d'autres termes, ça veut dire que l'aboutissement d'une condition à une action peut aussi mener à l'action inverse dans des circonstances inverses. Pour préserver la superposition et l'intrication on a besoin d'évolutions unitaires. Pour ceci, une condition nécessaire est que les opérations soient réversibles.

5.2.4 Calcul quantique et algorithmes quantiques

Il s'agit d'utiliser des processus quantiques pour réaliser des opérations logiques. Les unités de calcul sont des qubits évoluant dans des superpositions d'états [Preskill, 1998; Rieffel et Polak, 2000].

L'état d'un qubit est un vecteur dans un espace vectoriel à deux dimensions. Pour manipuler ces données quantiques (représentées sous forme de bits quantiques), on dis-

tingue deux types d'opérations : opérations simples dites portes quantiques simples et successions d'opérations quantiques appelées circuits quantiques [Talbi, 2009].

• Portes quantiques

Les portes quantiques sont des opérations élémentaires opérant sur un qubit ou un ensemble de qubits (registre quantique). Les portes quantiques célèbres permettant des transformations unitaires sur des qubits sont les suivantes [Rieffel et Polak, 2000] :

- les portes quantiques simples : la transformation identité, la négation, le décalage en phase, la combinaison du décalage en phase et de la négation.
- la porte racine carrée du NOT (\sqrt{NOT}).
- la porte controlled-NOT (CNOT).

• Circuits quantiques

Un circuit quantique est une combinaison de deux ou plusieurs portes quantiques permettant d'effectuer un traitement plus compliqué sur un système quantique.

Pour une étude plus approfondie des portes et circuits quantiques, le lecteur peut se référer à l'article *An Introduction to Quantum Computing for Non-Physicists* de Rieffel et Polak [2000].

• Algorithmes quantiques

Un algorithme quantique est une succession d'application d'opérations quantiques sur des systèmes quantiques. Ces opérations quantiques sont soit des portes logiques quantiques simples, soit des circuits quantiques. Les algorithmes quantiques offrent une complexité algorithmique nettement inférieure à celle des algorithmes classiques, grâce au concept de superposition d'états. Cette diminution de complexité a attiré l'attention de beaucoup de chercheurs qui ont fourni de grands efforts pour concevoir des algorithmes quantiques résolvant des problèmes connus du monde réel, citant l'algorithme de Shor [1994] pour la factorisation de grands nombres, l'algorithme de Grover [1996] pour la recherche dans une base de données non triée et l'algorithme de Kitaev pour la factorisation [Rieffel et Polak, 2000; Magniez, 2002].

Dans le contexte de l'informatique quantique, on ne peut pas vraiment parler des domaines d'applications présents vu l'absence d'ordinateurs quantiques puissants. Malgré ça, les futures applications de l'informatique quantique, vu les algorithmes quantiques déjà disponibles et les modèles préliminaires d'ordinateurs quantiques, sont très prometteuses. Les domaines les plus actifs selon cette perspective sont la recherche de l'information (recherche de sous groupes cachés) et la cryptographie quantique [Magniez, 2002].

5.2.5 Avantages et limites de l'informatique quantique

L'informatique quantique offre des possibilités de calcul et des fonctionnalités qui sont rare dans sa contrepartie classique. On peut les résumer dans les points suivants [Bellac, 2003].

- une puissance de calcul énorme grâce à la superposition d'états et des opérateurs quantiques permettant le traitement d'une grande quantité d'information en parallèle.
- l'indéterminisme et la non duplication, caractérisant ce paradigme, ont des applications très utiles dans le domaine de la cryptographie.
- une complexité algorithmique remarquablement diminuée.

Malgré les possibilités offertes par l'informatique quantique, les limites suivantes restent à résoudre pour pouvoir exploiter ses atouts. Parmi les limites actuelles de l'informatique quantique, on trouve les suivantes [Nedjah *et al.*, 2008; Bellac, 2003].

- les algorithmes quantiques ne sont réalisables que sur des machines quantiques dont la construction est toujours en état de recherche.
- la décohérence, l'interférence destructive, cause des problèmes au processus de calcul, la moindre perturbation du système causera l'effondrement du processus de calcul et du système lui même. Pour cette raison, l'ordinateur quantique doit être totalement isolé de son environnement dans la phase de traitement.
- la difficulté de correction d'erreurs : ceci est dû à la nature quantique de la machine. Donc, plus d'effort doit être fourni pour la correction d'erreurs, causées par la décohérence, pour chaque algorithme.
- l'opération de lecture (mesure) elle même peut changer l'état du système ou causer sa corruption. Pour ceci, on doit utiliser des mécanismes bien spécifiés pour lire les résultats d'un traitement quantique avant qu'ils soient perdus. Malgré que ceci a été fourni par Grover [1996], ça reste très basique et demande un effort considérable pour pouvoir être utilisé au niveau algorithmique.

5.3 Algorithmes évolutionnaires quantiques

L'exploitation de toutes les opportunités qu'offre l'informatique quantique ne sera possible qu'après la mise en place de vrais ordinateurs quantiques. En attendant la construction de machines quantiques puissantes, l'idée de simulation des algorithmes quantiques sur

ordinateurs classiques ou de les combiner à d'autres méthodes conventionnelles est apparue. Parmi ces hybridations, qui ont prouvé d'être meilleures que leurs contreparties classiques, on cite : les réseaux de neurones quantiques, les colonies de fourmis quantiques, les essaims de particules quantiques et les algorithmes évolutionnaires quantiques [Nedjah *et al.*, 2008] qu'on présente dans cette section.

Définition 5.1 (AEQ) *Un algorithme évolutionnaire quantique est un algorithme évolutionnaire enrichi par les concepts et les principes de l'informatique quantique, tel que le bit quantique, la superposition d'états et les opérateurs quantiques.*

Historiquement, le premier algorithme s'inspirant de l'informatique quantique a été celui de Narayanan et Moore [1996] qui ont proposé un algorithme génétique s'inspirant des principes de l'informatique quantique. Cependant, cet algorithme n'a pas suscité beaucoup d'attention à cause de son interprétation différente de l'interprétation standard de l'informatique quantique et à cause de la pauvreté de formalisme et d'évidence offerts par les auteurs.

Quatre ans plus tard, Han et Kim [2000] ont proposé un nouveau modèle d'algorithmes génétiques s'inspirant des concepts de l'informatique quantique tels que le bit quantique (qubit), la superposition d'états et l'opérateur quantique (la porte D). Ils ont appliqué l'algorithme proposé sur un problème d'optimisation combinatoire, celui du *sac-à-dos binaire (0/1 knapsack)* et ont surtout conclut que l'hybridation entre algorithmes génétiques et l'informatique quantique est une voie très prometteuse.

Cet algorithme de Han et Kim n'utilise ni mutation ni croisement. Selon les auteurs, l'utilisation de ces deux opérations conduit vers une détérioration des performances. Dans ce qui suit, l'algorithme original de Han et Kim sera présenté. Autres variantes des algorithmes évolutionnaires quantiques peuvent être trouvés dans [Han et Kim, 2002, 2004; Da Cruz *et al.*, 2005, 2007; Zhang *et al.*, 2003].

5.3.1 Individu quantique

Han et Kim [2000] utilisent dans leur algorithme des individus quantiques représentés par des registres quantiques ; une chaîne de bits quantiques (voir formule 5.5).

$$\begin{pmatrix} \alpha_1 & \alpha_2 & \dots & \alpha_m \\ \beta_1 & \beta_2 & \dots & \beta_m \end{pmatrix} \quad (5.5)$$

Où : α_i et β_i représentent les amplitudes des qubits, et donc, doivent vérifier la condition $|\alpha_i|^2 + |\beta_i|^2 = 1$

Grâce à cette superposition d'états, un individu quantique peut représenter en même temps toute une population d'individus, chacun avec une probabilité. Donc, il ne sera pas nécessaire dans une telle représentation d'individus d'avoir une grande taille de la population, ce qui offre plus de diversité à l'AEQ, tout en utilisant une petite population. Un peu plus tard, Han [2003] démontre que les performances offertes par un nombre d'individus un peu supérieur sont meilleures que celles données par un seul individu.

En plus de la mise à jour de la meilleure solution trouvée, deux opérations dites quantiques sont appliquée d'une façon itérative sur les individus quantiques jusqu'à l'atteinte d'une condition d'arrêt. Il s'agit de la *mesure quantique* et la *porte D*.

5.3.2 Mesure quantique d'individus

Cet opérateur permet d'extraire un individu classique à partir d'un individu quantique en appliquant des mesures quantiques individuelles (comme vu dans la section 5.2.2) sur ses qubits. La figure 5.3 donne une illustration de la mesure quantiques appliquée à un individu quantique.

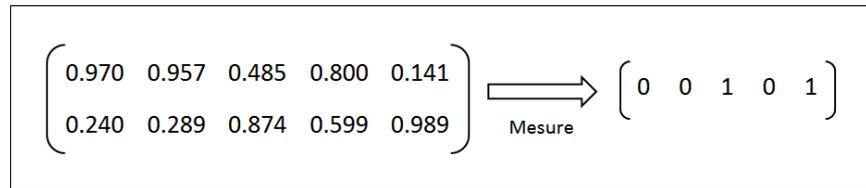


Figure 5.3: Mesure quantique d'individus

5.3.3 La porte D

A chaque itération, l'actuelle meilleure solution sert comme guide pour rechercher de nouvelles solutions qui peuvent s'avérer meilleures. Cela est implémenté via une porte quantique, appelée la porte *D*, qui effectue une rotation du qubit en question de façon à augmenter la probabilité d'avoir la valeur binaire du bit correspondant dans la meilleure solution actuelle (Figure 5.4).

La porte *D* prend la forme matricielle donnée par la formule 5.6.

$$\begin{pmatrix} \cos \delta\theta & -\sin \delta\theta \\ \sin \delta\theta & \cos \delta\theta \end{pmatrix} \tag{5.6}$$

Le tableau ci-dessous donne la direction de rotation en fonction de α_i, β_i et de la valeur du bit correspondant dans la meilleure solution actuelle.

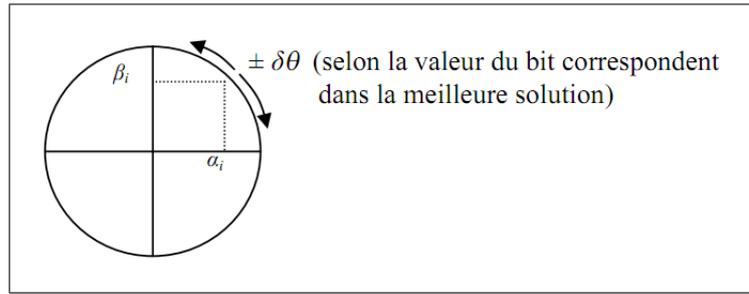


Figure 5.4: Effet de la porte D sur un qubit

Le pseudo-code de l’algorithme de Han et Kim [2000] est donné dans l’algorithme 5.1, où : $Q(t)$ est la population composée de chromosomes quantiques à l’instant t , $B(t)$ est la meilleure solution jusqu’à l’instant t , et $P(t)$ est l’ensemble de solutions binaires issues de la mesure de $Q(t)$.

5.3.4 Discussion de l’algorithme évolutionnaire quantique

Tout algorithme évolutionnaire doit être capable de bien équilibrer entre la diversification et le renforcement, souvent connus sous les noms : exploration et exploitation respectivement. Dans ce contexte, on introduit ici une petite analyse de l’AEQ de Han et Kim [2000] vis-à-vis ces deux critères [Talbi, 2009].

Dans l’algorithme évolutionnaire quantique de Han et Kim [2000], le codage quantique des chromosomes constitue le moyen employé par les auteurs pour garantir la diversification de la recherche. Un qubit pourrait, dans la majorité des situations, produire lors de sa mesure la valeur “0” ou la valeur “1”. Donc, un chromosome pourrait potentiellement couvrir tout l’espace de recherche.

Table 5.1: Direction de la rotation

α	β	valeur du bit de référence	angle de rotation
> 0	> 0	1	$+ \delta\theta$
> 0	> 0	0	$- \delta\theta$
> 0	< 0	1	$- \delta\theta$
> 0	< 0	0	$+ \delta\theta$
< 0	> 0	1	$- \delta\theta$
< 0	> 0	0	$+ \delta\theta$
< 0	< 0	1	$+ \delta\theta$
< 0	< 0	0	$- \delta\theta$

Algorithme 5.1 Algorithme évolutionnaire quantique (AEQ)

$t \leftarrow 0$
Produire $P(t)$ par observation de $Q(t)$
Evaluer $P(t)$
Stocker la meilleure solution parmi $P(t)$ dans $B(t)$
Tant que (critère d'arrêt non satisfait) **Faire**
 $t \leftarrow t + 1$
 Produire $P(t)$ par observation de $Q(t - 1)$
 Evaluer $P(t)$
 Générer $Q(t)$ en appliquant la porte D sur $Q(t - 1)$
 Stocker la meilleure solution parmi $B(t - 1)$ et $P(t)$ dans $B(t)$
Fin Tant que

L'opérateur de rotation, ou la porte D , est l'outil garantissant le renforcement de la recherche autour de la meilleure solution actuelle. Au cours des itérations, cet opérateur augmente, pour chaque qubit, la probabilité d'avoir la valeur du bit correspondant dans la meilleure solution. Dans la majorité des situations, on garde la possibilité, même affaiblie, d'avoir la valeur inverse qui pourrait contribuer à la découverte d'une nouvelle meilleure solution.

Cette stratégie n'est pas tout le temps gagnante. Le problème majeur est qu'en cas de non découverte d'une nouvelle meilleure solution après un certain nombre d'itérations, la probabilité pour un qubit de générer la valeur binaire inverse de la valeur du bit correspondant dans la meilleure solution sera amplement réduite, voir même mise à 0. La généralisation de ce phénomène sur l'ensemble des qubits entraînera l'effondrement de la superposition d'états et le blocage des chromosomes dans un optimum local.

5.4 Autres algorithmes inspirés du quantique

En se basant sur le framework de l'algorithme évolutionnaire quantique de Han et Kim [2000] et en s'inspirant d'autres approches évolutionnaires, d'autres instances et hybridation d'AEQ ont été créées et appliquées avec succès pour la résolution de problèmes d'optimisation. Dans ce qui suit on donne une brève description de quelques algorithmes s'inspirant de l'informatique quantique.

5.4.1 Algorithmes génétiques quantiques

Les AEQs ont prouvé leur efficacité pour la résolution de problèmes d'optimisation combinatoire. Cependant, un problème qui les limite est la perte de diversité avec le temps. Pour

pouvoir faire un bon équilibre entre l'exploration et l'exploitation, une variante des AEQs a été proposée et utilisée pour résoudre divers problèmes, tels que le recalage d'images [Talbi *et al.*, 2004a; Draa, 2004; Talbi *et al.*, 2006], le problème du voyageur de commerce [Talbi *et al.*, 2004b], la vérification formelle [Layeb et Saidouni, 2007], la bio-informatique [Layeb *et al.*, 2008] et la segmentation d'images [Talbi *et al.*, 2007]. Dans cette variante, en plus des opérateurs quantiques de mesure et de mise à jour, des opérateurs de croisement et de mutation quantiques sont utilisés.

• **Croisement quantique**

Ce croisement a le même principe qu'un croisement classique. Mais, il opère sur des chromosomes quantiques. Donc, c'est un croisement de matrices de probabilités qui génère comme résultat des matrices de probabilités [Draa, 2004]. Le croisement quantique entre deux individus (parents) en un point donné permet de générer deux nouveaux individus (descendants) dont les gènes proviennent de leurs deux parents (Figure 5.5).

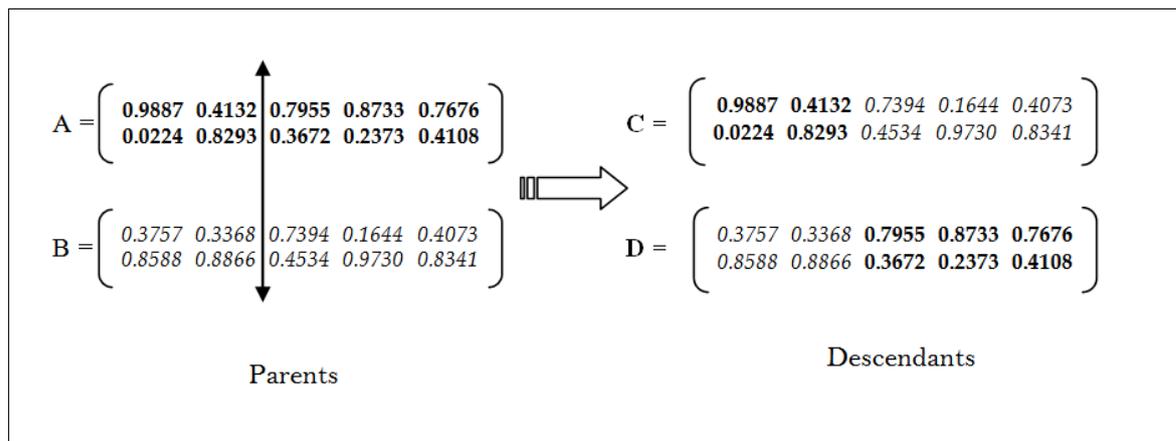


Figure 5.5: Croisement quantique

• **Mutation quantique**

La mutation classique opère comme une petite perturbation qui inverse le bit muté. Dans une mutation quantique, il y a aussi une perturbation, mais elle opère sur les probabilité d'un qubit du chromosome en question de la façon suivante. Soit un qubit $A = \alpha|0\rangle + \beta|1\rangle$. La mutation quantique du qubit A génère le qubit $B = \beta|0\rangle + \alpha|1\rangle$. La figure 5.6 donne un exemple dans lequel une mutation quantique est appliquée au troisième point du chromosome d'entrée.

Après une opération de croisement quantique, le nombre d'individus augmente. Donc, pour maintenir la taille initiale de la population, une opération de sélection quantique s'im-

0.2383	0.1079	0.5057	0.8337
0.9712	0.9942	0.8627	0.5522
		↓	
0.2383	0.1079	0.8627	0.8337
0.9712	0.9942	0.5057	0.5522

Figure 5.6: Mutation quantique

pose. Il est difficile, bien que faisable, de considérer les caractéristiques des chromosomes quantiques pour formuler un critère de sélection. Ceci est dû au fait qu'un chromosome quantique représente potentiellement une multitude de solutions éparpillées dans l'espace de recherche.

On a donc suggéré de se baser sur la qualité de la solution binaire issue d'une opération de mesure pour évaluer un chromosome. Une opération de mesure est appliquée sur l'ensemble des chromosomes. Une solution est obtenue de chaque chromosome et les chromosomes desquels émanent les meilleures solutions sont choisis pour former la nouvelle génération [Talbi, 2009].

5.4.2 Boids de Reynolds quantiques

Les boids de Reynolds quantiques sont le résultat de l'enrichissement du modèle classique des boids de Reynolds [1987] (vu dans la section 6.4.1 du chapitre 2) par les concepts de l'informatique quantique tels que la superposition d'états et la mesure quantique.

Un boids de Reynolds quantique est une suite de qubits qui peut être représentée sous la forme d'un registre quantique. Par conséquence, les opérations de base du modèle de Reynolds sont adaptées pour créer des opérations de boids quantiques qui sont nommées : *cohésion quantique*, *séparation quantique* et *alignement quantique*. Ces opérations seront décrites en détail dans le chapitre 6 vu qu'il constituent une base de l'une des contributions de notre thèse : l'LCA quantique.

La représentation quantique des boids a offert à ce modèle plus de diversité par rapport aux représentations classiques (binaire ou réelle) du fait que chaque boid représente une multitude de boids en superposition. Les boids de Reynolds quantiques ont été appliqués pour la résolution du problème du voyageur de commerce et ont donné des résultats très prometteurs [Boufenar *et al.*, 2005].

5.4.3 Réseaux de neurones quantiques

S'inspirant de l'informatique quantique, l'approche des réseaux de neurones quantiques vise à améliorer les performances des réseaux de neurones formels classiques avec de

nouveaux algorithmes quantiques [Sandford, 2007]. Ventura et Martinez [1997] ont proposé un modèle basé sur la superposition des poids de connexions du réseau, le principe est simple : lors de l'utilisation du réseau, une mesure est effectuée et les poids de connexions sont fixés de manière probabiliste (phénomène de décohérence), l'exécution du réseau s'effectue ensuite classiquement comme dans un réseau de neurones classique. Les intérêts possibles sont le gain de performances dû aux algorithmes quantiques d'une part et la possibilité offerte par l'aspect probabiliste de la mesure quantique d'effectuer certaines fonctions inaccessibles aux réseaux de neurones classiques telles que le problème de l'XOR d'une autre part [Ventura et Martinez, 1997; Al-Rabadi et Lendaris, 2003].

Les réseaux de neurones quantiques offrent des gains non négligeables, tant au niveau de la capacité de stockage que de la rapidité. Les propriétés de superposition d'états d'intrication et de décohérence ont permis aux réseaux de neurones quantiques d'être meilleurs que leur contrepartie classique [Sandford, 2007].

5.5 Avantages des algorithmes inspirés du quantique

Comme on l'a déjà vu dans les sections ci-dessus, le paradigme des algorithmes inspirés du quantique est basé sur l'exploitation des concepts quantiques permettant de surmonter les limites des approches d'optimisation classiques, notamment la convergence prématurée et la lenteur de convergence. Ceci a été relativement atteint dans ce type d'algorithmes. On peut résumer les points forts des approches inspirés du quantique dans les points suivants.

- La représentation quantique des individus (boids dans les boids de Reynolds quantiques, poids dans les RNAs quantiques, ou chromosomes dans les AGs quantiques) permet à un algorithme inspiré du quantique une meilleure exploration : une bonne diversité est offerte, ce qui permet de couvrir une grande partie de l'espace de recherche.
- Ces algorithmes sont caractérisés par une complexité réduite : très peu d'individus sont nécessaires pour une bonne représentation de l'espace de recherche.
- La porte D (connu des fois sous le nom d'*interférence*) qui constitue la base de comportement dans la majorité des algorithmes inspirés du quantique permet une bonne exploitation des solutions déjà trouvées.

5.6 Conclusion

Dans ce chapitre, on a vu une introduction à l'informatique quantique comme domaine émergent en informatique. On a présenté ses principes de base et ses avantages et li-

mites. On a également parlé des algorithmes inspirés du quantique tout en présentant trois modèles qui ont un lien fort avec nos contributions : les algorithmes évolutionnaires quantiques, les algorithmes génétiques quantiques et les boîtes de Reynolds quantiques. On a aussi introduit les réseaux de neurones quantiques, vus comme un bon modèle artificiel des systèmes complexes adaptatifs. Finalement, une petite synthèse des avantages des algorithmes inspirés du quantique a été donnée.

Dans le prochain chapitre, un algorithme inspiré du quantique, constituant notre première contribution, sera présenté. C'est l'automate cellulaire apprenant quantique.

Troisième partie

Contributions

Résumé des contributions

Dans la première partie de cette thèse on a présenté les concepts de base des systèmes complexes adaptatifs (Chapitre 1) et quelques modèles artificiels pour la simulation et l'exploitation de ces concepts pour des fins de calcul (Chapitre 2). La deuxième partie a été dédiée à l'étude des éléments de base des modèles de CASs proposés dans cette thèse : les automates cellulaires apprenants (Chapitre 3), l'évolution différentielle (Chapitre 4) et l'informatique inspirée du quantique (Chapitre 5).

Cette troisième partie est consacrée à la propositions de deux nouvelles approches pour la mise en œuvre des CASs et ceci à travers l'extension des modèles vus dans les deux premières parties de cette thèse pour surmonter leurs limites. Pour ce faire trois points clé ont été introduits :

1. l'hybridation entre deux modèles des CASs : l'automate cellulaire apprenant et les boids de Reynolds.
2. l'exploitation des concepts de l'informatique quantique à travers l'adoption d'une représentation quantique des individus et l'introduction d'opérateurs quantiques dans cette hybridation et dans le modèle à évolution différentielle.
3. l'introduction d'autre opérateurs, pour le cas de l'évolution différentielle, telle que la régénération, pour augmenter la diversité au sein de la population.

Chapitre 6

Automate cellulaire apprenant quantique

“Research is to see what everybody else has seen, and to think what nobody else has thought.” - Albert Szent-Gyorgyi

Sommaire

6.1	Introduction	83
6.2	Motivations et idée de base	83
6.3	Structure de l'automate cellulaire apprenant quantique	84
6.3.1	Une représentation quantique des cellules	84
6.3.2	Voisinage	85
6.3.3	Matrice de probabilités d'actions	85
6.4	Comportement de l'automate cellulaire apprenant quantique	87
6.4.1	Boids de Reynolds	87
6.4.2	Mesure quantique	90
6.4.3	Évaluation	90
6.4.4	Comparaison et mise à jour	91
6.4.5	Cohésion quantique et séparation quantique	91
6.4.6	Alignement quantique	93
6.4.7	Mise à jour de la matrice de probabilités d'actions	93
6.5	Applications	94
6.5.1	L'LCA quantique pour la résolution du TSP	94
6.5.2	L'LCA quantique pour l'optimisation de fonctions numériques	99
6.6	Conclusion	100

6.1 Introduction

Dans les parties précédentes de cette thèse, on a présenté quelques modèles artificiels pour la simulation et l'exploitation des concepts des systèmes complexes adaptatifs. Malgré leurs capacités à résoudre des problèmes académiques ou de la vie réelle, ces modèles souffrent de plusieurs inconvénients, comme déjà vu dans ces chapitres, qui limitent leurs performances et les préviennent de bien simuler les propriétés clé des CASs. Les deux limites commune entre ces modèles sont généralement la convergence prématurée et la lenteur de convergence.

Dans cette optique, on présente dans ce chapitre un nouveau modèle pour la mise en œuvre des CASs. Il s'agit de l'enrichissement par des concepts quantiques de l'hybridation entre les LCAs et les boids de Reynolds. Ces derniers ont été utilisés pour implémenter le comportement des cellules de l'LCA qui a été choisi pour la raison principale d'être adaptatif au changements de l'environnement. Une représentation quantique est utilisée pour offrir plus de diversité aux états des cellules de l'automate. Donc, de nouvelles opérations dites opérations quantiques ont été définies et utilisées.

L'approche proposée, L'LCA quantique (en anglais : Quantum Learning Cellular Automaton, QLCA), a été utilisée pour résoudre deux problèmes connus dans le domaine de l'optimisation : le problème du voyageur de commerce et l'optimisation des fonctions numériques. Les résultats obtenus ainsi que leur interprétation seront introduits dans le chapitre 7.

Dans ce chapitre on donne, dans une première partie, l'idée de base et la description du comportement global de l'LCA quantique et de ses opérations quantiques, ainsi qu'une description du modèle des Boids de Reynolds (BRs), l'un des éléments du modèle proposé. Ensuite, l'application de l'LCA quantique proposé pour la résolution des deux problèmes en question est détaillée dans les sections qui suivent. Finalement, une discussion du comportement de l'LCA quantique et de ses limites est donnée avant de conclure.

6.2 Motivations et idée de base

La puissance des automates cellulaires apprenants (vus dans le chapitre 3) est due de leur capacité de s'adapter et de survivre dans un environnement ouvert. Ceci est réalisé à travers la mise à jour des probabilités d'actions offertes au cellules de l'automate selon un schéma d'apprentissage par renforcement. Donc, les LCAs offrent de grandes possibilités du point de vue modélisation de l'adaptation, notion clé dans un système complexe adaptatif. Cependant, les deux problèmes suivants limitent l'utilisation des LCAs pour la modélisation des CASs :

1. Le comportement de base des agents doit être très simple mais capable d'offrir un phénomène global complexe.
2. La taille de la grille des cellules doit être suffisamment grande pour couvrir une bonne portion de l'espace de recherche. Ceci cause cause la lenteur de convergence.

Pour surmonter ces deux limites on a opté pour l'utilisation du modèle des boîds de Reynolds pour modéliser le comportement local des cellules de l'automate, en plus de l'utilisation d'une représentation quantique inspirée de l'informatique quantique. Les BRs ont été choisis vu leur capacité de simuler l'émergence de phénomènes complexes au niveau macro du système à partir d'interactions et comportements locaux basiques des composants. L'utilisation des notions de l'informatique quantique, notamment la superposition d'états, non seulement réduit la complexité algorithmique du modèle, mais aussi, offre une meilleure diversité au système et donc, une meilleure couverture de l'espace de recherche.

L'idée de base du modèle proposé, l'LCA quantique, est d'enrichir le modèles des LCAs par une représentation quantique des individus et d'utiliser des opérations inspirées du modèle de Reynolds comme opérations offertes aux cellules. Le modèle des boîds de Reynolds sera présenté en détail dans la section 6.4.

6.3 Structure de l'automate cellulaire apprenant quantique

6.3.1 Une représentation quantique des cellules

Malgré les performances d'adaptation des LCAs classiques, ils souffrent du problème de manque de diversité. Les cellules dans ce type de modèles ne représentent qu'une partie très limitée de l'espace de recherche. Pour augmenter leurs diversité, les LCAs classiques doivent contenir un grand nombre de cellules, ce qui cause l'augmentation de la complexité algorithmique.

Pour résoudre ce problème, une représentation quantique des cellules est adoptée dans notre modèle. Cette représentation, comme déjà expliqué dans le chapitre 5, offrent une bonne diversité au modèle : une cellule, *appelée cellule quantique*, peut contenir toutes les valeurs possibles de cellules en même temps. Donc, avec un nombre très réduit de cellules, on peut couvrir une très grande partie de l'espace de recherche.

La figure 6.1 donne un exemple d'un LCA quantique de 3×3 cellules quantiques dans le cas du problème du voyageur de commerce (en anglais : Travelling Salesman Problem, TSP). Chaque cellule de notre automate dans ce cas est une superposition d'états de solutions classiques, qui se modélisent généralement sous la forme d'une matrice binaire, comme on le verra dans la section 6.5.1 (Figure 6.11).

Donc, la population sur laquelle un LCA quantique opère, est une grille de cellules quantiques qui sont soit des matrices quantiques, soit des vecteurs quantiques. Les éléments constituant ces matrices et vecteurs quantiques ne sont que des bits quantiques (qubits).

6.3.2 Voisinage

La relation de voisinage, une caractéristique clé des systèmes complexes adaptatifs, est essentielle dans notre modèle. Le voisinage choisi dans notre QLCA est le voisinage de *Von Neumann* en 2D (Figure 6.2(a)). On a préféré ce voisinage par rapport à celui de Moore (Figure 6.2(b)) à cause de sa simplicité qui reflète la perception limitée de l'agent de son environnement dans un CAS.

6.3.3 Matrice de probabilités d'actions

Comme on l'a déjà vu dans le chapitre 3, à chaque cellule d'un LCA est associé un vecteur de probabilités représentant les probabilités d'applications des actions offertes à la cellule. De la même manière, pour chaque cellule de notre LCA quantique est associé un vecteur de probabilités d'actions. Pour des raisons de simplification de représentation et d'implémentation, on a regroupé ces vecteurs dans une matrice qu'on appelle *ActProb* (Figure 6.3).

Dans notre LCA quantique, on a utilisé les boîtes de Reynolds comme base du comportement des agents du système, cellules dans notre cas. Donc, trois opérations quantiques

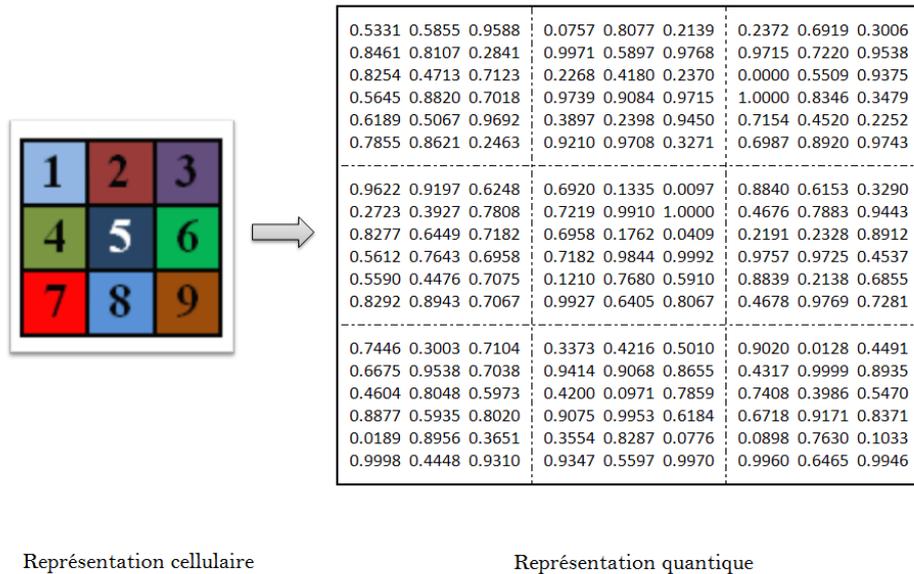


Figure 6.1: Représentation quantique d'un LCA

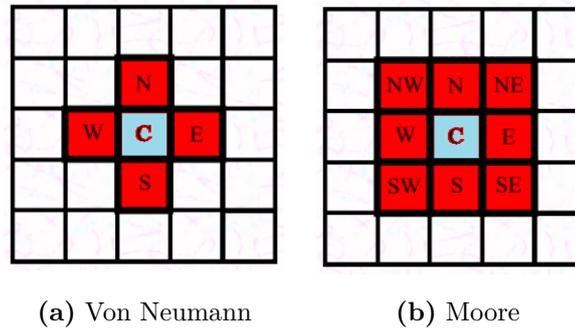


Figure 6.2: Voisinage de Von Neumann et voisinage de Moore

similaires à celles des boids classiques sont définies et utilisées. Ce sont : *la séparation quantique* (Q-séparation), *l'alignement quantique* (Q-alignement) et *la cohésion quantique* (Q-Cohésion). On verra ces opérations en détail dans la section 6.4.

Ce qui nous intéresse ici est de savoir que chaque élément de la matrice des probabilités d'actions *ActProb* est un vecteur à 2 composantes représentant la probabilité d'appliquer des cohésion des voisines vers la cellule en question et la probabilité de séparer les voisines de cette cellule, respectivement. L'alignement quantique est une opération qui ne dépend pas des réponses de l'environnement, et donc, elle n'est pas sujette d'adaptation via le processus d'apprentissage par renforcement.

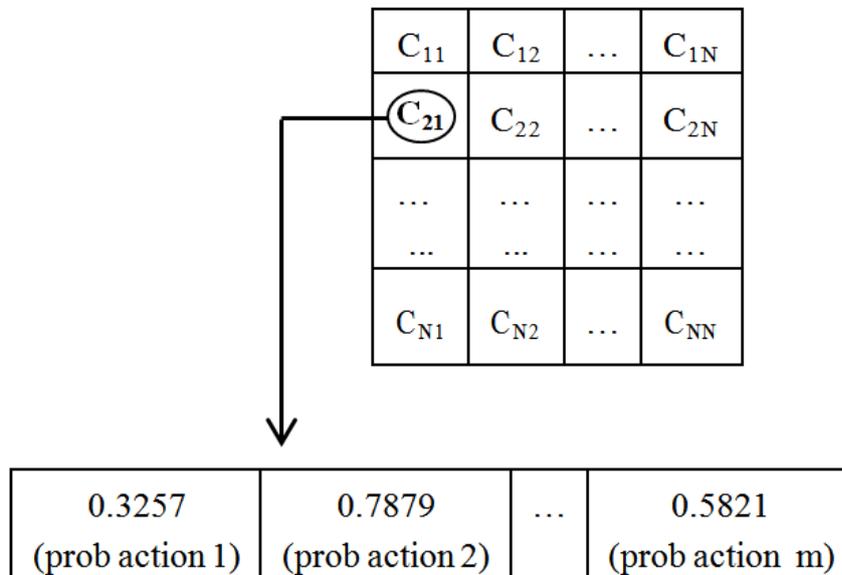


Figure 6.3: Matrice de probabilités d'actions pour un LCA quantique

6.4 Comportement de l'automate cellulaire apprenant quantique

A base de la structure de l'LCA quantique, et de la représentation quantique décrites dans la section précédente, on décrit dans cette section la stratégie d'évolution de notre LCA quantique. Le comportement générique de l'LCA quantique est très similaire à celui du LCA classique (décrit dans la section 3.4 du chapitre 3) avec la particularité que les opérations dans notre version sont des opérations quantiques.

L'idée de base est de guider la sélection d'une action par l'historique de l'automate et par les récompenses ou pénalités reçues de l'environnement. Initialement, on commence d'une grille de cellules quantiques, et on applique cycliquement l'ensemble des opérations suivantes pour chaque cellule jusqu'à satisfaire une condition d'arrêt, généralement un nombre spécifique d'itérations ou l'obtention d'un optimum global [Draa et Meshoul, 2010] :

1. Mesure quantique
2. Évaluation
3. Comparaison et mise à jours
4. Cohésion quantique ou séparation quantique
5. Alignement quantique
6. Mise à jours de la matrice de probabilités d'actions

Tout au long du processus d'évolution de l'automate, la meilleure solution trouvée est sauvegardée, et utilisée pour guider les réponses de l'environnement aux actions de l'automate à chaque itération. Ces actions sont jugées par l'environnement par rapport à la meilleure solution trouvée et à la moyenne des solutions de l'itération en cours.

Les opérations offertes aux cellules sont inspirées du modèle de boids de Reynolds, mentionné dans la section motivations. Pour ceci, on va d'abord présenter le modèle des boids de Reynolds.

6.4.1 Boids de Reynolds

En 1986, Reynolds [1987] créa un programme basé sur des créatures artificielles appelées '*boids*' visant à simuler le comportement d'essaims naturels. Ce modèle qui a été nommé par la suite '*Boids de Reynolds*' a constitué la base pour plusieurs modèles plus puissants d'intelligence collective, notamment les essaims de particules (PSO).

Définition 6.1 (Boid) *Un boid est une représentation d'un oiseau dans une simulation d'essaim, notamment dans le contexte du modèle de boids de Reynolds et des modèles s'inspirant de ce modèle [Erneholm, 2011].*

Chaque individu dans le modèle des boids de Reynolds, caractérisé par sa position et sa vitesse, évolue dans un espace tridimensionnel, percevant les autres boids dans un voisinage donné. Cette évolution est gérée par des règles comportementales simples permettant au boids de se déplacer en groupe et d'éviter les obstacles [De Romemon, 2010]. Chaque boid a l'accès direct à la description géométrique de la scène entière, mais il réagit seulement aux autres boids dans un voisinage autour de lui-même. Les boids en dehors de ce voisinage local sont ignorés [Montalvo, 2005].

Les trois règles de base proposées par Reynolds [1987] dans son modèle sont les suivantes, illustrées dans la figure 6.4.

- **Cohésion** : orienter pour se déplacer vers la position moyenne des boids locaux.
- **Alignement** : orienter vers la direction moyenne des boids locaux.
- **Séparation** : orienter pour éviter de serrer les boids locaux.

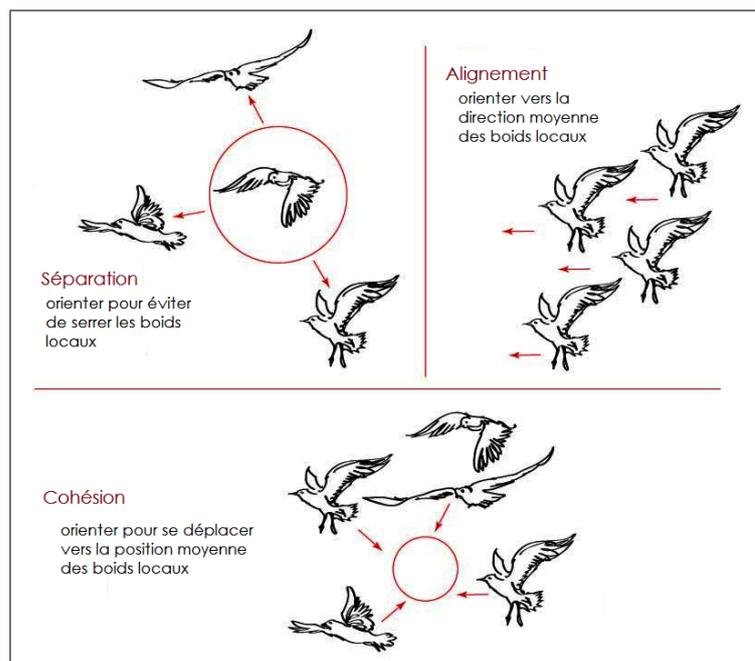


Figure 6.4: Comportements de boids de Reynolds

Depuis son apparition, ce modèle a été essentiellement orienté vers des applications graphiques : jeux, cinéma, etc, où un ensemble d'objets, e.g soldats ou monstres, déplacent dans des patterns complexes [Davison, 2005]. Le pseudo-code 6.1 présente une description de l'application du modèle de base pour la simulation du comportement naturel d'oiseaux [Erneholm, 2011].

Algorithme 6.1 Boids de Reynolds (dans une scène 2D)

Pour chaque scène **Faire**

Pour chaque boid **Faire**

 Séparation(boid, voisinage)

 Alignement(boid, voisinage)

 Cohésion(boid, voisinage)

Fin Pour

Pour chaque boid **Faire**

$boid.x \leftarrow \cos(boid.direction) * b.vélocité * dTime;$

$boid.y \leftarrow \sin(boid.direction) * b.vélocité * dTime;$

Fin Pour

Fin Pour

• **Avantages et limites du modèle de Reynolds**

Le premier avantage à citer des boids de Reynolds est sa simplicité : trois comportements simples à implémenter constituent la base du modèle. Un autre avantage très important dans le champs des systèmes complexes adaptatifs est l'émergence de phénomènes complexes à partir de comportements très basiques (Figure 6.5). Cet avantage est un point clé dans l'exploitation des boids pour des fins de calcul dans le domaine d'optimisation. De plus, le modèle de Reynolds constitue la base de tous les modèles d'essaims (PSO) qui ont été utilisés avec succès pour la résolution d'une grande masse de problèmes académiques ou réels.

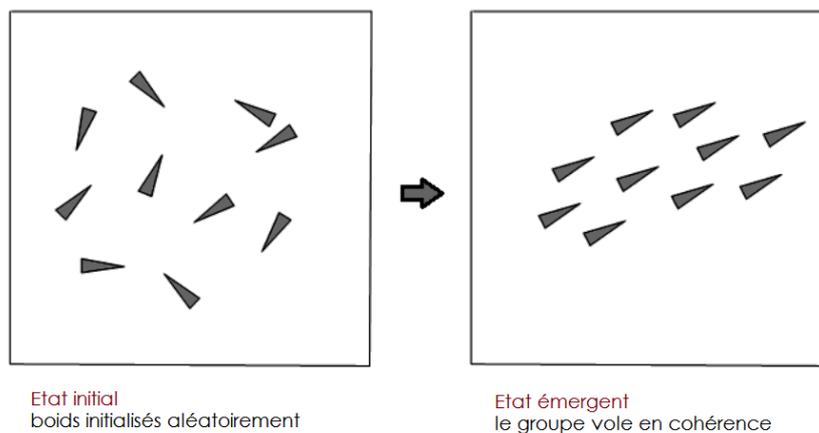


Figure 6.5: L'émergence dans le modèle de boids

[Davison, 2005]

Malgré ces avantages, les boids de Reynolds souffrent de deux problèmes majeurs. Le premier problème est l'absence d'une base théorique solide pour le modèle, comparé à d'autres modèles d'essaims, notamment le PSO. L'autre limite est le fait que les deux règles de séparation et de cohésion sont contradictoires et un paramétrage difficile de leur application est nécessaire pour pouvoir les contrôler tout en préservant leurs apports au comportement global émergent du système.

6.4.2 Mesure quantique

La mesure quantique qui se base sur les valeurs des bits quantiques est liée à la représentation quantique adoptée dans cette approche. Le but de cette opération est d'observer une cellule quantique pour obtenir une cellule binaire. L'avantage de cette opération par rapport à sa contrepartie classique est qu'elle conserve l'état quantique de la cellule. Le résultat de la mesure est une cellule binaire à part.

L'opération de mesure quantique se déroule comme suit. Pour chaque qubit de la cellule en question, une comparaison entre un nombre généré aléatoirement r et le carré de la composante β du qubit, la probabilité d'avoir le 1, est établie : si $\beta^2 > r$ alors le bit correspondant de la matrice résultat est mis à 1, sinon il est mis à 0. La figure 6.6 donne un exemple du résultat de la mesure d'une cellule quantique de l'automate.

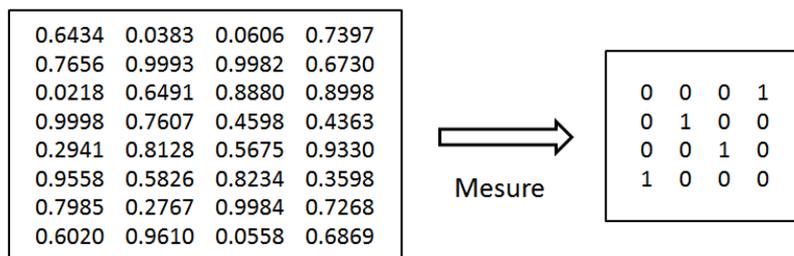


Figure 6.6: Mesure quantique d'une cellule

6.4.3 Évaluation

Dans l'étape d'évaluation, on calcule la qualité de la solution codée par la solution obtenue de la mesure. Ceci se fait bien sûr selon la formule de la fonction objectif du problème à résoudre. Dans le cas du TSP par exemple, comme on le verra par la suite, la qualité d'une solution binaire est la longueur du chemin équivalent. Pour l'LCA quantique, cette valeur, de la fonction objectif, joue le rôle de sortie de la cellule mesurée vers l'environnement. Un exemple de l'opération d'évaluation est donnée dans la figure 6.7 où la matrice à gauche

est la solution binaire pour le TSP, et la matrice à droite est la matrice de distances entre les villes.

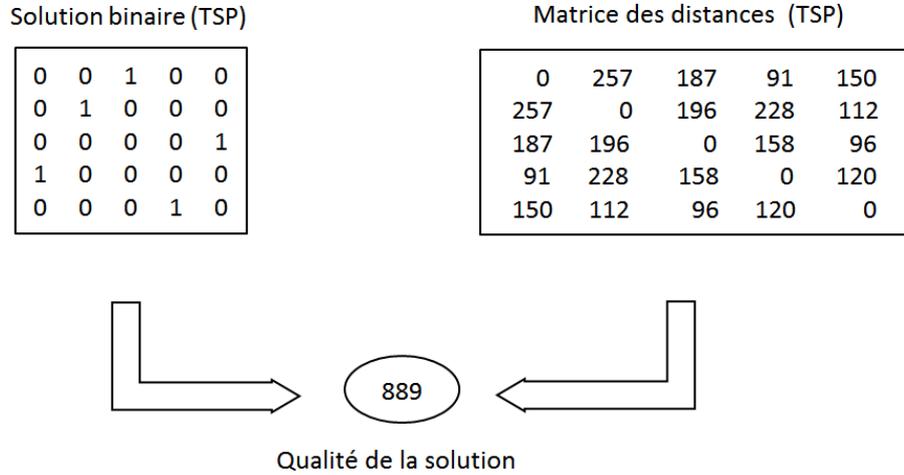


Figure 6.7: Évaluation d'une solution binaire dans le cas du TSP

6.4.4 Comparaison et mise à jour

Dans cette étape, on met à jour la meilleure solution globale suite à une comparaison entre la valeur de la qualité de la solution obtenue de l'étape d'évaluation avec la qualité de la meilleure solution globale.

6.4.5 Cohésion quantique et séparation quantique

La cohésion et la séparation quantiques sont appliquées selon les probabilités correspondantes dans la matrice des probabilités d'actions *ActionProb*. Dans cette dernière, chaque paire de probabilités représente respectivement la probabilité d'appliquer une Q-cohésion des cellules voisines vers la cellule en question et la probabilité d'appliquer une Q-séparation des cellules voisines de cette cellule. Il est à noter que les éléments de la matrice de probabilités d'actions sont initialisés par des vecteurs de valeurs $(1/2, 1/2)$ car on a deux actions parmi lesquelles chaque LA correspondant à une cellule peut choisir.

Les opérations de cohésion et séparation quantiques sont décrites dans le pseudo-code de l'algorithme 6.2, où r est un nombre généré aléatoirement entre 0 et 1, $ActionProb(2 * i - 1, j)$ et $ActionProb(2 * i, j)$, sont les probabilités de cohésion des cellules voisines vers ou de leur séparation de la cellule $Aut(i, j)$.

Algorithme 6.2 Cohésion et séparation quantiques

Si $r < ActionProb(2 * i - 1, j)$ **Alors**

Q-cohésion($Voisines(i,j), Aut(i,j)$)

Fin Si

Si $r < ActionProb(2 * i, j)$ **Alors**

Q-séparation($Voisines(i,j), Aut(i,j)$)

Fin Si

• **Cohésion quantique**

Dans le cas de la Q-cohésion, chaque qubit dans les cellules voisines est mis à jour pour s'approcher du qubit correspondant dans la cellule en question. Cette mise à jour peut être décrite comme suit. Soit $A\langle\alpha_A, \beta_A\rangle$ le qubit à mettre à jour vers le qubit $B\langle\alpha_B, \beta_B\rangle$, et soit CC un entier qu'on appelle coefficient de cohésion. Les composantes α_C, β_C du qubit résultant de la cohésion $C\langle\alpha_C, \beta_C\rangle$ sont calculées selon l'ensemble d'équations 6.1.

$$\begin{cases} \alpha_C = \frac{CC \times \alpha_A + \alpha_B}{CC + 1} \\ \beta_C = \sqrt{1 - \alpha_C^2} \end{cases} \quad (6.1)$$

• **Séparation quantique**

D'une manière similaire de la Q-cohésion, la Q-séparation permet d'éloigner tous les qubits dans les cellules de voisinage des qubit correspondants dans la cellule en question. Dans ce cas, les composantes α_C et β_C du qubit résultat $C\langle\alpha_C, \beta_C\rangle$ sont déterminées de deux manières différentes, selon la position du qubit $A\langle\alpha_A, \beta_A\rangle$ par rapport au qubit $B\langle\alpha_B, \beta_B\rangle$ (Figure 6.8). Si $|\alpha_A| > |\alpha_B|$, on utilise la paire d'équations 6.2, sinon on utilise la paire 6.3. Dans les deux paires d'équations, SC est un nombre entier représentant le coefficient de séparation.

$$\begin{cases} \alpha_C = \frac{1 + SC \times \alpha_A}{CC + 1} \\ \beta_C = \sqrt{1 - \alpha_C^2} \end{cases} \quad (6.2)$$

$$\begin{cases} \alpha_C = \frac{SC \times \alpha_A}{CC + 1} \\ \beta_C = \sqrt{1 - \alpha_C^2} \end{cases} \quad (6.3)$$

Comme on le voit dans ces deux paires d'équations (6.2 et 6.3), si la composante α du qubit à éloigner A est supérieure à celle du qubit de référence B , ça veut dire que la probabilité d'avoir un 0 ($|\alpha|^2$) dans le qubit à éloigner est supérieur à celle du qubit de référence. Donc, il est logique qu'une séparation du qubit A de B est d'augmenter sa probabilité d'avoir le zéro et donc l'équation 6.2 est utilisée. Dans le cas contraire, la probabilité d'avoir le 0 est supérieur dans le qubit de référence, la séparation est faite par le rapprochement du qubit à séparer vers le 1, et donc, on utilise l'équation 6.3.

6.4.6 Alignement quantique

L'opération d'alignement permet de guider les qubits de la cellule en question vers leurs correspondants binaires dans la meilleure solution binaire trouvée. Elle opère comme suit. Soit $A\langle\alpha_A, \beta_A\rangle$ le qubit à aligner et AC un entier représentant le coefficient d'alignement. Les amplitudes α_R et β_R du qubit résultat R sont calculées selon la paire d'équations 6.4, où $best(i, j)$ est la valeur du bit correspondant dans la meilleure solution binaire.

$$\begin{cases} \alpha_R = \frac{(1 - best(i, j)) + AC * \alpha_A}{AC + 1} \\ \beta_R = \sqrt{1 - \alpha_R^2} \end{cases} \quad (6.4)$$

Si la valeur de $best(i, j)$, le bit correspondant dans la solution binaire est égale à 1, alors, la probabilité d'avoir le 0 ($|\alpha_A|^2$) dans le qubit à aligner est diminuée, sinon, elle est augmentée.

La cohésion, la séparation et l'alignement quantiques sont les opérations offertes aux cellules de l'automate. Les deux premières opérations sont appliquées selon les probabilités correspondantes dans la matrice de probabilités d'actions, alors que l'alignement quantique est appliqué indépendamment des réponses de l'environnement, selon une probabilité fixe. Ces trois opérations sont illustrées par la figure 6.8

6.4.7 Mise à jour de la matrice de probabilités d'actions

La dernière opération à appliquer est la mise à jour de la matrice de probabilités d'actions. Cette opération permet l'adaptation de l'automate à son environnement externe, chose qui n'est pas disponible dans les automates cellulaires conventionnels. Elle joue un rôle très important dans la dynamique globale du système. Selon le résultat de l'évaluation d'une solution donnée par l'automate à l'environnement, ce dernier renvoie soit une récompense soit une pénalité à l'automate. A base de cette réponse, la probabilité d'appliquer l'opération qui a donné ce résultat est ajustée : augmentée dans le cas de récompense, ou diminuée dans le cas de pénalité.

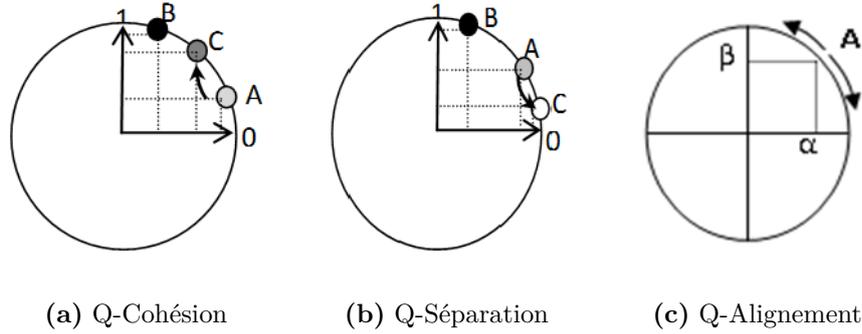


Figure 6.8: Opérateurs de boids quantiques

La mise à jour des probabilités d'actions se fait selon la paire d'équations 6.5. Il s'agit d'un apprentissage par renforcement dans un automate à structure variable, déjà vu dans le chapitre 3 (section 3.3.3), où $r \in \{0, 1\}$ est le signal reçu de l'environnement ($r = 0$ dans le cas d'une récompense ou $r = 1$ dans le cas d'une pénalité). Les paramètres a et b sont respectivement les taux de récompense et de pénalité. Ils sont choisis empiriquement selon le type du problème à résoudre.

$$\begin{cases} p_i \leftarrow p_i + a * r(1 - p_i) - b(1 - r)p_i \\ p_j \leftarrow p_j - a * r * p_j + b(1 - r)(\frac{1}{n-1} - p_j), \quad \forall j \neq i. \end{cases} \quad (6.5)$$

Pour synthétiser, l'LCA quantique présenté dans ce chapitre fonctionne selon le pseudo-code donné par l'algorithme 6.3.

6.5 Applications

Après avoir présenté la structure générique de l'LCA quantique et le comportement gérant son évolution, dans cette section, on présente l'application du modèle proposé pour la résolution de deux problèmes célèbres dans le domaine de l'optimisation : le TSP et l'optimisation de fonctions numériques.

6.5.1 L'LCA quantique pour la résolution du TSP

• Formulation du problème

Le problème du voyageur de commerce (TSP) vise à trouver le chemin le plus court que doit suivre un voyageur de commerce pour visiter N villes. Ceci est équivalent à trouver le cycle Hamiltonien le plus court reliant N sommets. Le TSP est l'un des problèmes d'optimisation combinatoire les plus étudiés, c'est un outil très efficace pour l'évaluation

Algorithme 6.3 Automate cellulaire apprenant quantique

Initialiser la grille de cellules quantiques {En mettant tous les qubits de toutes les cellules à la valeur $\langle\sqrt{2}/2, \sqrt{2}/2\rangle$ }

Initialiser la matrice de probabilités d'actions {En mettant toutes les valeurs à $(1/2, 1/2)$ }

Tant que (critère d'arrêt non atteint) **Faire**

Pour (chaque cellule de l'automate) **Faire**

1. Mesure quantique
2. Évaluation
3. Comparaison et mise à jours
4. Cohésion quantique et séparation quantique
5. Alignement quantique
6. Mise à jours de la matrice de probabilités d'actions

Fin Pour

Fin Tant que

d'algorithmes d'optimisation. L'évaluation d'une solution pour le TSP est effectuée par le calcul de la longueur du chemin représenté par cette solution. Le but du processus d'optimisation est de minimiser cette longueur [Draa, 2004].

En d'autres termes, un représentant commercial doit démarcher des clients dans N villes. Il connaît les distances séparant les villes entre elles mais il n'a pas encore arrêté son itinéraire exact. Ce dernier doit être aussi court que possible, afin de minimiser la distance totale parcourue, mais doit passer une et une seule fois par chacune des villes. La difficulté du TSP réside dans le fait que le nombre de solutions croît d'une manière exponentielle avec le nombre de villes. Soit les circuits représentés dans la figure 6.9 : le circuit Hamiltonien le plus court doit être celui ayant la longueur 27.

Les applications du TSP sont nombreuses. D'une part, certains problèmes d'optimisation de parcours en robotique ou en conception de circuits électroniques ainsi que certains problèmes de séquençement (passage de trains sur une voie, atterrissage d'avions, processus de fabrication en industrie chimique, etc.) s'expriment directement sous forme de TSP. D'autre part, et c'est sans doute la famille d'application la plus importante, les problèmes de transport sont généralement plus complexes que le TSP, mais comportent des sous problèmes de type TSP [Laburthe, 1998].

Le problème du voyageur de commerce est dit symétrique quand le coût pour aller de la ville i à la ville j est le même pour aller de j vers la ville i . Autrement, le problème est dit asymétrique [Talbi, 2009]. Dans ce qui suit, on ne s'intéresse qu'à la variante symétrique :

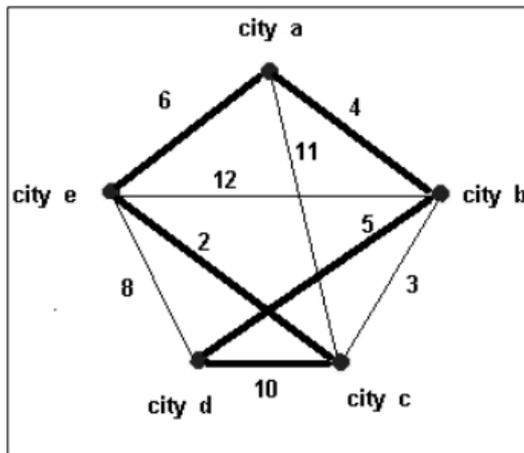


Figure 6.9: Le circuit Hamiltonien du TSP

le TSP en soit n'est pas l'objet de cette thèse, on l'utilise seulement autant qu'un outil de validation du modèle conçu, l'LCA quantique.

Les distances entre les villes sont représentées par une matrice D de dimension $N \times N$, tel que $D(i, j)$ est la distance entre la ville i et la ville j . La matrice de la figure 6.10 représente les distances mutuelles dans un ensemble de 5 villes.

0	257	187	91	150
257	0	196	228	112
187	196	0	158	96
91	228	158	0	120
150	112	96	120	0

Figure 6.10: Matrice de distances pour TSP symétrique

Une solution du TSP pour N villes est un circuit qui relie ces villes en visitant chaque ville une et une seule fois. Donc, on peut représenter la solution par une matrice A de taille $N \times N$ associant à chaque ville son rang dans le circuit : Si $A(i, j) = 1$, alors i est la j ème ville visitée, sinon, i est dans un autre rang. Par exemple, la figure 6.11 représente le circuit : ABDCA.

Dans notre algorithme, on a opté pour une représentation quantique des solutions. Donc, la grille des cellules dans notre QLCA est composée de $M \times M$ cellules. Chaque cellule est une matrice quantique contenant $N \times N$ qubits (Figure 6.12). Cette représentation quantique offre à chaque solution (cellule dans notre automate) la possibilité de

1	0	0	0
0	1	0	0
0	0	0	1
0	0	1	0

Figure 6.11: Une matrice de solution pour le TSP

contenir toutes les solutions possibles en même temps, solutions non acceptables y comprises. Ceci offre une diversité énorme pour l’automate : avec un nombre très réduit de cellules, une très grande portion de l’espace de recherche est couverte.

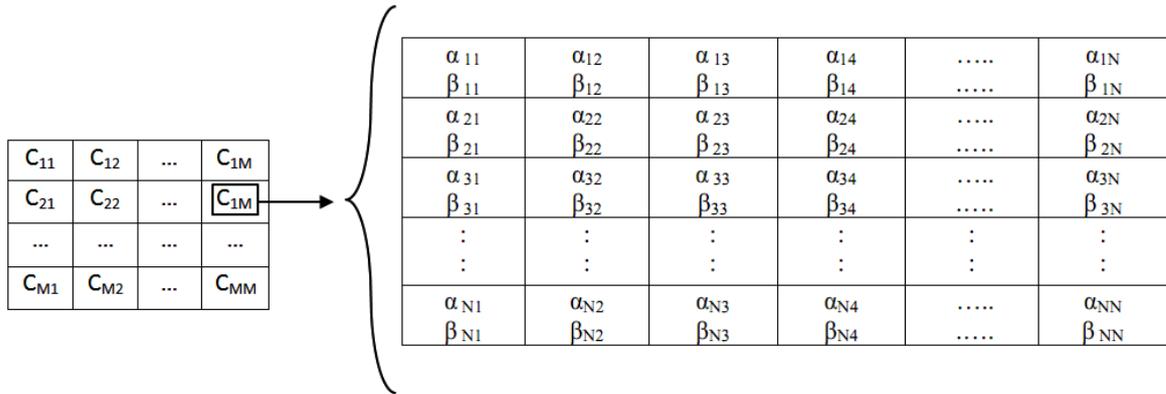


Figure 6.12: Codage quantique des cellules de l’automate pour le TSP

• **Spécificités liées au TSP**

Afin de pouvoir utiliser l’LCA quantique pour résoudre le problème du voyageur de commerce, quelques contraintes sont à respecter. Une opération de permutation est aussi introduite. Ces contraintes et cette opération sont expliquées dans ce qui suit.

- **Une mesure quantique avec contrainte**

La mesure dans le cas du problème du voyageur de commerce diffère de la mesure ordinaire (déjà vue dans la section 6.4.2) dans le fait qu’une solution binaire n’est acceptable dans le cas du TSP que si elle vérifie la contrainte d’avoir “un seul 1 par ligne et par colonne”. Le pseudo-code de l’algorithme 6.4 donne la variante de la mesure quantique qui vérifie cette contrainte tout en s’appuyant sur les valeurs des amplitudes des qubits.

Algorithme 6.4 Mesure quantique pour le TSP

Initialiser la matrice solution Sol avec des zéros
 Créer une copie $CellSave$ de la cellule $Cell$ {Pour préserver les valeurs originales}
 Choisir aléatoirement un entier $r \in \{1, \dots, N\}$
Pour ($i \leftarrow 1$ à $[(r + N - 1) \text{ modulo } N]$) **Faire**
 Générer un nombre aléatoire $j \in [1, N]$
 Générer un nombre aléatoire $p \in [0, 1]$
 Tant que ($p > |\beta_{ij}|^2$) **Faire**
 $j \leftarrow j + 1 \text{ modulo } N$
 Générer un nombre aléatoire $p \in [0, 1]$
 Fin Tant que
 $Sol(i, j) \leftarrow 1$
 Pour ($k \leftarrow 1$ à N) **Faire**
 $\beta_{kj} \leftarrow 0$
 $\alpha_{kj} \leftarrow 1$
 $\beta_{ik} \leftarrow 0$
 $\alpha_{ik} \leftarrow 1$
 Fin Pour
Fin Pour

- Opérateur de permutation

Cet opérateur est spécifique au TSP. Une interversion entre deux blocs de colonnes de même taille se fait dans quelques cellules de l'automate selon une probabilité qu'on appelle ici $permProb$. A noter qu'il s'agit ici d'une permutation quantique, opérant sur des matrices quantiques (cellules de l'automate quantique).

Cette opération joue un rôle important dans ce type de problème car il suffit des fois d'inter-verser l'ordre de deux villes pour avoir la solution optimale. La figure 6.13 donne un exemple de l'effet de l'opération de permutation quantique.

- Fonction objectif

La fonction objectif dans le cas du problème de voyageur de commerce est la longueur globale du chemin désigné par la matrice binaire des solutions. Cette longueur est calculée selon des matrices de distances fournies par le site Web des benchmarks TSPLIB [TspLib, 1995]. Ces matrices contiennent les distances entre chaque couple des villes.

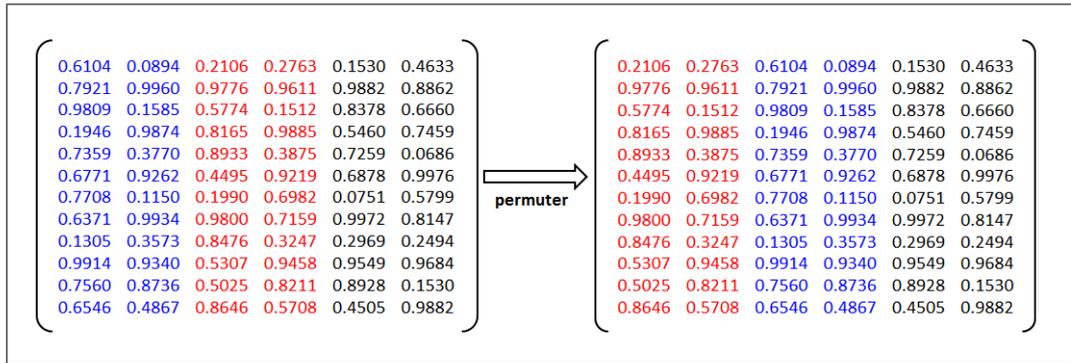


Figure 6.13: Permutation quantique

6.5.2 L’LCA quantique pour l’optimisation de fonctions numériques

Après avoir présenté l’application du LCA quantique pour la résolution du problème du voyageur de commerce, dans cette section on présente l’application du LCA quantique pour la résolution du problème d’optimisation de fonctions numériques.

- **Formulation du problème**

Le problème d’optimisation de fonctions numériques consiste en l’optimisation d’un ensemble de fonctions références dans le domaine de l’optimisation numérique. Il s’agit des fonctions *Sphere*, *Rosenbrock* et *Griewank*. La première est une fonction uni-modale, n’ayant qu’un seul minimum : minimum global, alors que les deux autres sont multi-modales, ayant plusieurs minima (locaux et globaux). Ces trois fonctions sont formulées comme présenté dans le tableau 6.1, où : n est la dimension de la fonction. Plus de détails sur les caractéristiques de ces fonctions seront abordées dans le chapitre 7.

Table 6.1: Les fonction benchmarks

Fonction	Formule	Variable	Optimum
Rosenbrock	$f_1(x) = \sum_{i=1}^{n-1} [(1 - x_i)^2 + 100(x_{i+1} - x_i^2)^2]$	$x_i \in [-30, +30]$	$\vec{x}^* = (0, 0, \dots, 0),$ $f(\vec{x}^*) = 0$
Griewank	$f_2(x) = \frac{1}{4000} \left(\sum_{i=1}^n (x_i - 100)^2 \right) - \prod_{i=1}^n \cos \left(\frac{x_i - 100}{\sqrt{i}} \right) + 1$	$x_i \in [-600, +600]$	$\vec{x}^* = (100, 100, \dots, 100),$ $f(\vec{x}^*) = 0$
Sphere	$f_3(x) = \sum_{i=1}^n (x_i)^2$	$x_i \in [-100, +100]$	$\vec{x}^* = (0, 0, \dots, 0),$ $f(\vec{x}^*) = 0$

• Spécificités liées au problème d'optimisation de fonctions numériques

Dans sa conception, L'LCA quantique pour l'optimisation de fonctions diffère de celui conçu pour le TSP dans trois points. Premièrement, les cellules sont des registres quantiques et non pas des matrices quantiques. Ces registres quantiques sont des superpositions de représentations binaires à virgule fixe des variables des fonctions. On a utilisé cette représentation pour pouvoir exploiter les concepts quantiques déjà cités ci-dessus.

La deuxième différence est l'existence de l'opération de conversion du binaire vers le réel qui est appliquée juste après l'opération de mesure quantique et avant l'opération d'évaluation. Troisièmement, l'opérateur de permutation n'existe pas dans cette variante destinée à la résolution du problème d'optimisation de fonctions numériques.

Fonction objectif

Dans le cas du problème d'optimisation de fonctions numériques, la fonction objectif est elle-même la fonction en cours d'optimisation. Pour les fonctions benchmarks choisies (Tableaux 6.1), il s'agit d'un problème de minimisation, mais l'algorithme reste le même pour des fonctions à maximiser à part une petite modification de l'étape de comparaison et mise-à-jour.

6.6 Conclusion

Dans ce chapitre, on a présenté notre première contribution pour la simulation et l'implémentation des systèmes complexes adaptatifs : l'automate cellulaire apprenant quantique. Ce modèle est le résultat de l'hybridation entre un LCA et l'informatique quantique. Une représentation quantiques des cellules de l'automate a été adoptée afin d'augmenter la diversité du système tout en diminuant la complexité algorithmique. Des opérations inspirées du modèle des boîds de Reynolds et de l'informatique quantique ont été conçues pour mettre en œuvre l'interaction locale des cellules de l'automate. Ce sont la cohésion, la séparation et l'alignement quantiques.

Dans le chapitre suivant, le modèle proposé dans ce chapitre, LCA quantique, sera validé en présentant les résultats expérimentaux obtenus de son application pour le TSP et l'optimisation de fonctions numériques respectivement. Aussi, une étude de l'influence de chaque opérateur sur le comportement global du système sera abordée.

Chapitre 7

Validation de l'automate cellulaire apprenant quantique pour l'optimisation

“If you steal from one author it's plagiarism ; if you steal from many it's
research.” - Wilson Mizner

Sommaire

7.1	Introduction	103
7.2	Automate cellulaire apprenant quantique pour le TSP	103
7.2.1	Bechmarks	103
7.2.2	Résultats expérimentaux	103
7.3	LCA quantique pour l'optimisation de fonctions	105
7.3.1	Fonctions bechmarks	105
7.3.2	Résultats expérimentaux	106
7.4	Étude des opérateurs du LCA quantique	110
7.5	Conclusion	113

7.1 Introduction

Dans le chapitre précédent, on a présenté notre première contribution qui consiste en la proposition d'un nouveau modèle de systèmes complexes adaptatifs, c'est l'automate cellulaire apprenant quantique. On a également introduit deux applications de notre approche pour les problèmes d'optimisation, le TSP et l'optimisation de fonctions numériques.

Dans ce chapitre, les benchmarks utilisés dans les deux problèmes, les résultats expérimentaux, ainsi que l'analyse des opérateurs du modèle proposé seront présentés.

7.2 Automate cellulaire apprenant quantique pour le TSP

7.2.1 Benchmarks

L'LCA quantique proposé a été testé pour résoudre différentes instances du TSP symétrique obtenues du site de référence TSPLIB [TspLib, 1995]. On a testé l'algorithme sur 4 instances du problème. On a opté pour un apprentissage de type récompense-inaction. Il y a deux raisons principales derrière ce choix. Premièrement, un schéma d'apprentissage de type récompense-pénalité ne sera pas assez puissant à cause de la contradiction existante entre les deux actions de base offertes aux cellules : cohésion et séparation, leur application avec le même taux va résulter à une stagnation de l'algorithme. Deuxièmement, malgré qu'un schéma de type récompense- ϵ -pénalité est aussi efficace, il n'atteint pas les performances d'un schéma récompense-inaction. On a exécuté l'algorithme 10 fois pour chaque instance. À chaque fois, on calcule l'erreur par rapport à la solution optimale connue de l'instance. Les paramètres utilisés dans ces tests sont les suivants.

- Nombre d'itérations : selon l'instance du problème.
- Probabilité de permutation $permProb \in [0.01, 0.1]$
- Probabilité d'alignement $alProb \in [0.15, 1]$
- Coefficient de cohésion CC entre 5 et 30.
- Coefficient de séparation SC entre 5 et 30.
- Coefficient de alignement AC entre 5 et 100.
- Taux de récompense a : selon l'instance du problème.

7.2.2 Résultats expérimentaux

Les résultats obtenus de l'application de ce premier modèle des CASs sur un ensemble de tests sont donnés dans le tableau 7.1. De ces résultats, on voit que l'LCA quantique a donné des bons résultats avec des taux d'erreur très réduits par rapport à la solution optimale connue de la base TSPLIB [TspLib, 1995].

Table 7.1: Résultats expérimentaux de l'LCA quantique pour le TSP

Instance	"GR24"		"GR17"		"BURMA14"		"GR21"	
	Résultat	Déviatio	Résultat	Déviatio	Résultat	Déviatio	Résultat	Déviatio
1	1377	8.25%	2168	3.98%	3087	0.00%	2707	0.00%
2	1473	15.80%	2190	5.03%	3087	0.00%	2707	0.00%
3	1355	6.52%	2213	6.13%	3120	1.06%	2707	0.00%
4	1290	1.4%	2166	3.88%	3087	0.00%	2851	5.31%
5	1289	1.33%	2167	3.93%	3087	0.00%	2815	3.98%
6	1272	0.00%	2162	3.69%	3155	2.2%	2707	0.00%
7	1278	0.47%	2085	0.00%	3087	0.00%	2707	0.00%
8	1349	6.05%	2095	0.47%	3087	0.00%	2707	0.00%
9	1323	4.00%	2163	3.74%	3087	0.00%	2707	0.00%
9	1372	7.86%	2085	0.00%	3087	0.00%	2707	0.00%
Sol.Opt.	1272		2085		3087		2707	
a	1/10		1/5		1/50		0.4	
It.Min.	4362		1534		814		1186	
MOY.	1337.8		2149.4		3090.3		2757.4	
Erreur	5.17%		3.08%		0.32%		1.86%	
Nbr.It.	5000		2000		4200		2500	
Prob.Perm	0.1		0.05		0.1		0.12	

Sol.Opt. : Solution optimale dans TSPLIB ; Moy. : Moyenne des solutions ;
 It.Min. : Nombre minimal d'itérations où la meilleure solution a été obtenue ;
 Erreur : Erreur vis-à-vis la solution optimale ; a : Taux de récompense ;
 Prob.Perm. : Probabilité de permutation ; Nbr.It. : Nombre d'itération ;

La figure 7.1 donne une illustration de l'évolution de la fitness des solutions pour l'application du QLCA pour le TSP, le meilleur global en rouge, la moyenne locale en bleu et le minimum local en vert. On voit clairement la convergence au niveau des trois types de fitness, locale, moyenne et globale tout en gardant un certain degré de diversité.

L'analyse des opérateurs du LCA quantique et de leur influence sur la convergence de l'algorithme est détaillée dans la section 7.4 qui étudie les opérateurs un par un et donne des illustrations des évolutions des fitness.

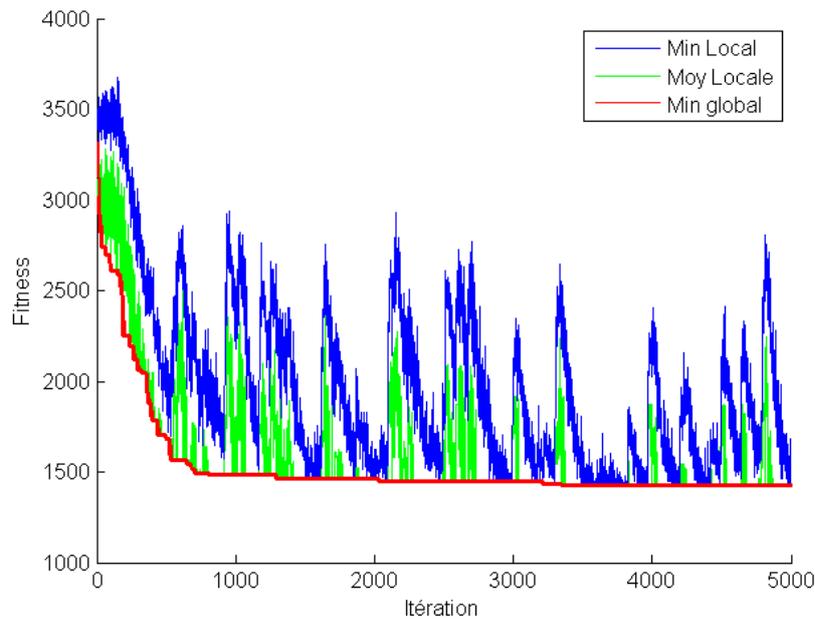


Figure 7.1: Évolution de la fitness dans un QLCA pour le TSP, meilleur local (en vert), moyenne locale (en bleu) et meilleur global (en rouge).

7.3 Automate cellulaire apprenant quantique pour l'optimisation de fonctions numériques

7.3.1 Fonctions bechmarks

Dans le chapitre 6, une formulation de base des fonctions de test a été donnée (Tableau 6.1). Dans cette section, plus de détails sur les caractéristiques de ces fonctions sont présentés.

La fonction *Sphere*, aussi appelée *la première fonction de De Jong* ou bien *F1 de De Jong*, est une fonction convexe, continue et uni-modale. Elle travaille sur des variables

indépendantes. Vu ces propriétés, cette fonction présente une classe de problèmes relativement simples à résoudre. La figure 7.2 donne une illustration de la fonction *Sphere* à deux dimensions.

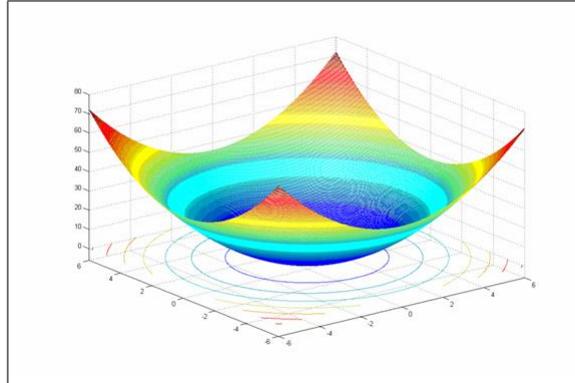


Figure 7.2: La fonction Sphere dans l'espace bidimensionnel

La fonction *Griewank* présente un problème d'optimisation hautement multi-modal avec nombreux minima locaux distribués régulièrement (Figure 7.3).

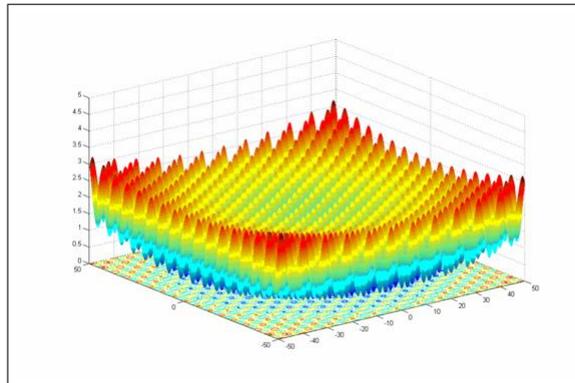


Figure 7.3: La fonction Griewank dans l'espace bidimensionnel

La fonction *Rosenbrock*, aussi connue sous le nom de *la fonction de banane*, est caractérisée par un optimum global situé à l'intérieur d'une vallée aplatie et parabolique. Elle a des variables fortement dépendantes avec un gradient qui ne pointe pas vers son meilleur global. Ceci rend cette fonction difficile à optimiser (Figure 7.4).

7.3.2 Résultats expérimentaux

On a testé l'algorithme (l'LCA quantique) sur deux instances pour chaque fonction : pour un nombre de variables égale à 10 et 30 respectivement, ce qui donnent un ensemble de 12 instances du problème. La taille de la grille de cellules a été mise à 6×6 et puis à

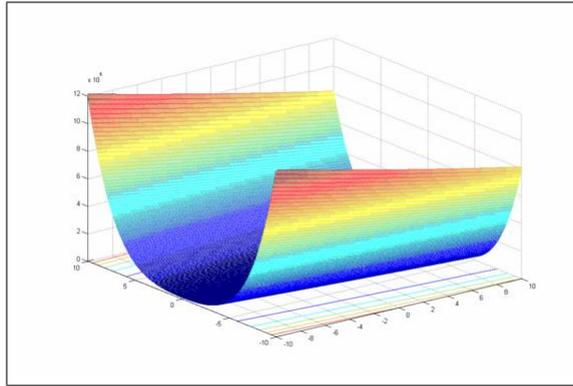


Figure 7.4: La fonction Rosenbrock dans l'espace bidimensionnel

9×9 cellules. Ce choix a été fait à base de deux raisons. Premièrement, on a voulu faire une comparaison des résultats avec ceux de Huang *et al.* [2008], et donc, on a choisi les mêmes dimensions des fonctions avec un nombre d'agents très proche de celui utilisé par ces auteurs. Deuxièmement, on a voulu vérifier si l'automate est capable de préserver la capacité d'adaptation pour des grandes tailles de population. Les autres paramètres ont été choisis comme montré dans le tableau 7.2.

Le nombre d'évaluations maximum de la fonction objectif a été mis à 400000 d'une façon similaire au travail de Huang *et al.* [2008]. Le programme s'est exécuté 50 fois pour chaque instance du problème. La moyenne des 50 exécutions et l'écart type sont calculés et comparés avec les résultats obtenus des algorithmes évolutionnaires quantiques et ceux du PSO quantique extraits de l'article de Huang *et al.* [2008]. Les résultats expérimentaux sont décrits dans le tableau 7.3.

La première chose à remarquer dans le tableau des résultats (Tableau 7.3) est le succès de l'LCA à donner de meilleurs résultats dans les 12 instances (marqués en gras) soit en termes de moyenne des meilleures solutions ou en termes de l'écart type. L'LCA quantique a même pu trouver l'optimum global pour toutes les instances de la fonction *Sphere*.

La deuxième chose qu'on remarque dans ces résultats est que l'instance avec une grille de 9×9 cellules donne toujours des résultats meilleurs que ceux donnée par l'instance de 6×6 cellules. Ceci signifie que l'augmentation de la taille de l'automate ne cause pas la perte des performances, au contraire elle les améliore.

La figure 7.5 donne une illustration de l'évolution de la fitness des solutions de l'LCA quantique pour l'optimisation de l'instance de la fonction *Griewank* avec 10 variables. On voit l'évolution du meilleur global en rouge, du minimum de chaque itération en vert et de la moyenne de chaque itération en bleu. De cette figure, il est clair que la recherche de

Table 7.2: Paramètres de l'LCA quantique pour l'optimisation de fonctions

	Rosenbrock						Griewank						Sphere							
	10	30	10	30	10	30	10	30	10	30	10	30	10	30	10	30	10	30	10	30
Dim	10	30	10	30	10	30	10	30	10	30	10	30	10	30	10	30	10	30	10	30
Size	6×6	9×9																		
<i>a</i>	1/5	1/20	1/5	1/5	1/10	1/10	1/10	1/10	1/10	1/10	1/10	1/10	1/5	1/5	1/5	1/5	1/5	1/5	1/5	1/5
<i>b</i>	1/100	1/1000	1/100	1/100	1/100	1/100	1/100	1/100	1/100	1/100	1/100	1/100	1/100	1/100	1/100	1/100	1/100	1/100	1/100	1/100
<i>AC</i>	100	1000	100	100	300	300	300	300	300	300	300	300	100	100	100	100	100	100	100	100
<i>CC</i>	100	1000	100	100	5000	5000	5000	5000	5000	5000	5000	5000	1000	1000	1000	1000	1000	1000	1000	1000
<i>SC</i>	100	1000	100	100	5000	5000	5000	5000	5000	5000	5000	5000	1000	1000	1000	1000	1000	1000	1000	1000

Table 7.3: Résultats expérimentaux de l'LCA quantique pour l'optimisation de fonctions numériques

Fonction	Dim	Taille	Max.It	QLCA		QSEA		CEA	
				Moy.Best	Ecart	Moy.Best	Ecart	Moy.Best	Ecart
Rosenbrock	10	6 × 6	1111	8.1373	0.2298	10.423	14.479	71.023	174.110
	30	6 × 6	2222	33.7450	9.0768	59.028	63.485	289.583	478.625
	10	9 × 9	987	7.9116	0.3641	8.634	16.589	37.365	57.431
	30	9 × 9	1975	28.3638	0.1908	51.565	40.843	202.456	289.968
Griewank	10	6 × 6	1111	0.0038	0.0098	0.069	0.051	0.085	0.073
	30	6 × 6	2222	0.0055	0.0229	0.012	0.017	0.013	0.015
	10	9 × 9	987	2.05E-004	0.0015	0.034	0.033	0.075	0.071
	30	9 × 9	1975	1.93E-006	7.44E-006	0.011	0.009	0.013	0.014
Sphere	10	6 × 6	1111	0	0	8.25E-72	5.82E-71	3.11E-23	8.01E-23
	30	6 × 6	2222	0	0	1.85E-28	6.71E-28	2.26E-10	5.08E-10
	10	9 × 9	987	0	0	3.09E-100	2.09E-99	6.13E-28	2.64E-27
	30	9 × 9	1975	0	0	1.09E-48	2.66E-48	2.47E-12	7.16E-12

Dim : Dimension de la fonction ; Taille : la taille de l'automate ; Max.It : Nombre maximum d'itérations ;
QLCA : Résultats du QLCA (notre approche) ; QSEA : Résultats du PSO quantique [Huang *et al.*, 2008] ;
CEA : Résultats de l'AE classique ; Moy.Best : la moyenne des meilleures solutions obtenues dans 50 exécutions ;
Ecart : L'écart type statistique

l'algorithme est très bien guidée. On observe que la moyenne et le minimum dans chaque itération évoluent en suivant l'évolution de la meilleure solution globale. Aussi, on voit qu'il y a un bon équilibre entre exploitation et exploration dans l'algorithme.

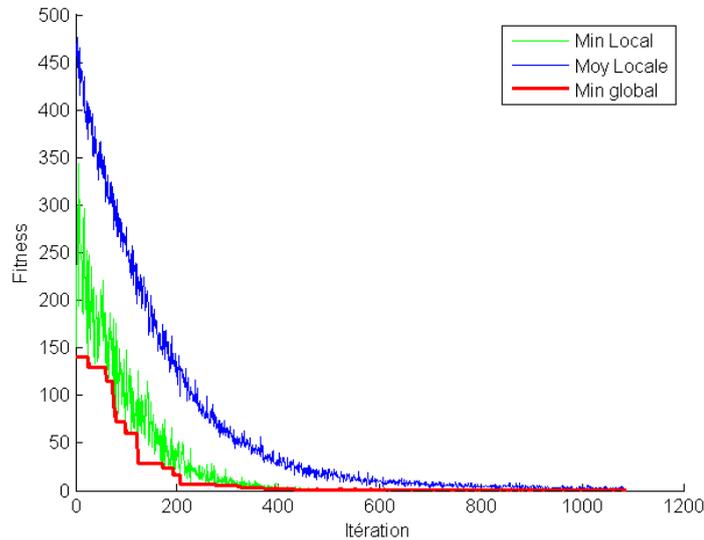


Figure 7.5: Évolution de fitness dans l'LCA quantique pour l'optimisation de fonctions : Meilleur global (en rouge), minimum local (en vert) et moyenne locale (en bleu)

7.4 Étude des opérateurs du LCA quantique

Pour savoir l'utilité de chaque opérateur dans la convergence globale de l'algorithme, on a procédé par une série de tests qui consiste en l'exécution de l'algorithme 10 fois avec l'opérateur en question d'étude et 10 fois sans son utilisation. Chaque fois, la moyenne des meilleures solutions pour les 10 exécutions est calculée. Pour tous les opérateurs, le tableau 7.4 présente une comparaison entre les résultats obtenus de la présence et l'absence de l'opérateur en question. Tous les tests ont été effectués sur l'instance de la fonction *Rosenbrock* avec 10 variables, à part la permutation qui est spécifique au TSP et qui a été testée sur l'instance *GR21* du TSP.

De l'analyse du tableau 7.4, il est clair que chaque opérateur joue un rôle crucial dans la convergence de l'algorithme. En plus de cette comparaison, les figures 7.6 à 7.9 résument le comportement de l'LCA quantique dans la présence de l'opérateur en question et dans le cas de son absence.

Ce qui est important à noter est que l'absence de la cohésion ou de la séparation peut ne pas causer une mauvaise convergence de l'algorithme comme les montre les sous-figures

Table 7.4: Résultats pour 10 exécutions avec et sans l'utilisation de chaque opérateur

Opérateur	Avec l'opérateur en question		Sans l'opérateur en question	
	Moyenne	Ecartype	Moyenne	Ecartype
Cohésion	8.3225	0.3614	58.1856	26.7706
Séparation	8.3225	0.3614	26.9527	16.7527
Alignement	8.3225	0.3614	1.3617E+006	7.5005E+005
Permutation(TSP)	Avec Permutation		Sans Permutation	
	Moyenne	Erreur	Moyenne	Erreur
	3029.60	11.92	4986.10	84.19

7.6(b) et 7.7(b), mais ceci se voit au niveau de plusieurs exécutions comme montré dans le tableaux 7.4.

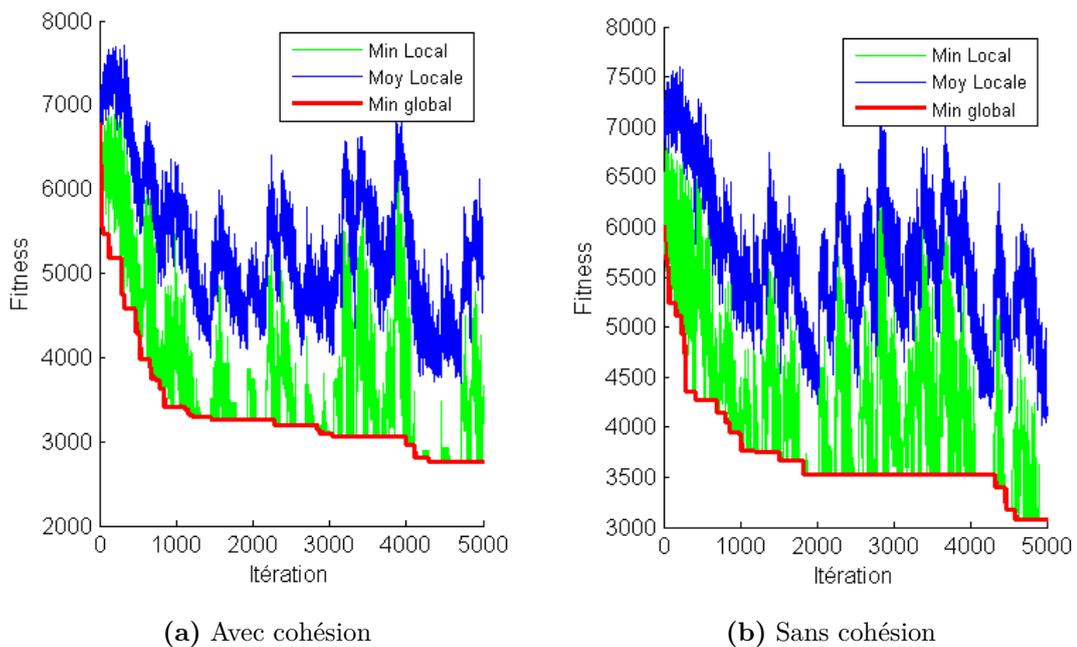


Figure 7.6: Evolution des fitness dans un LCA quantique avec cohésion (à gauche) et sans cohésion (à droite)

Au contraire de la cohésion et de la séparation, l'alignement et la permutation ont une grande influence sur la convergence. On observe une grande différence entre la présence de l'opérateur (Figures 7.8(a) et 7.9(a) et son absence (Figures 7.8(b) et 7.9(c)).

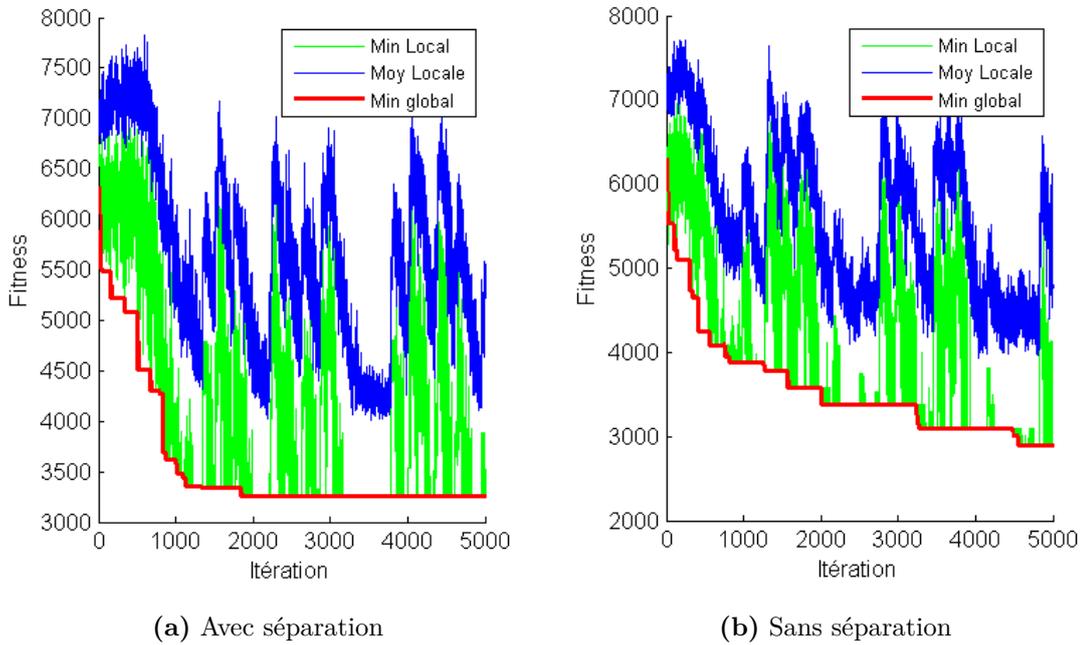


Figure 7.7: Evolution des fitness dans un LCA quantique avec séparation (à gauche) et sans séparation (à droite)

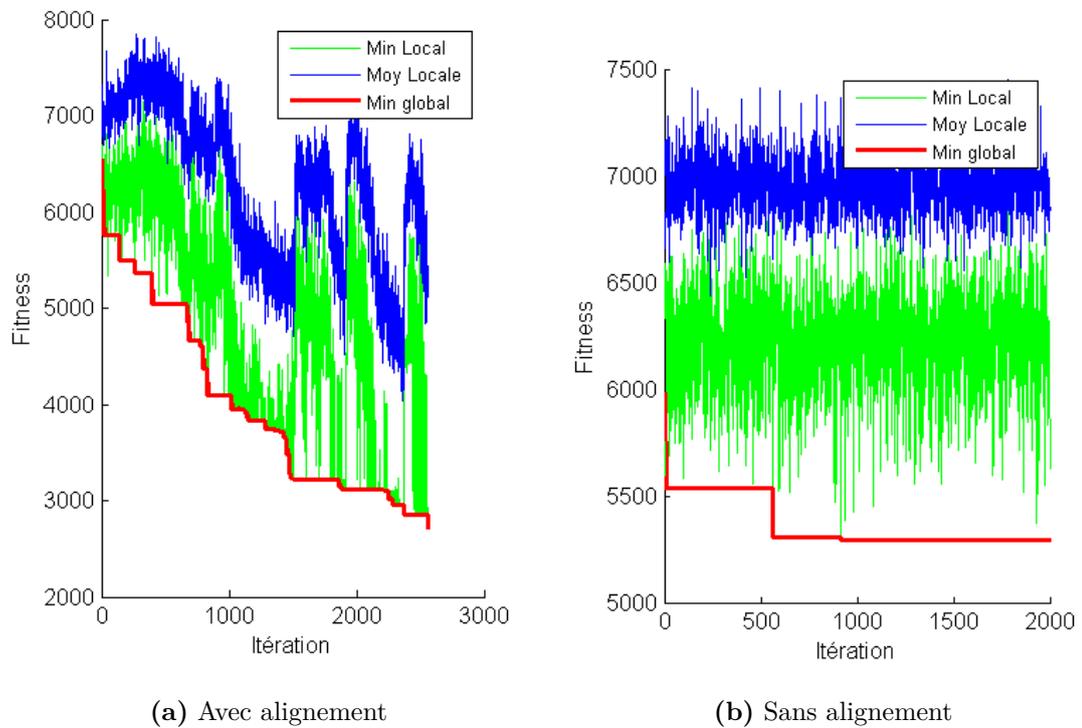


Figure 7.8: Evolution des fitness dans un LCA quantique avec alignement (à gauche) et sans alignement (à droite)

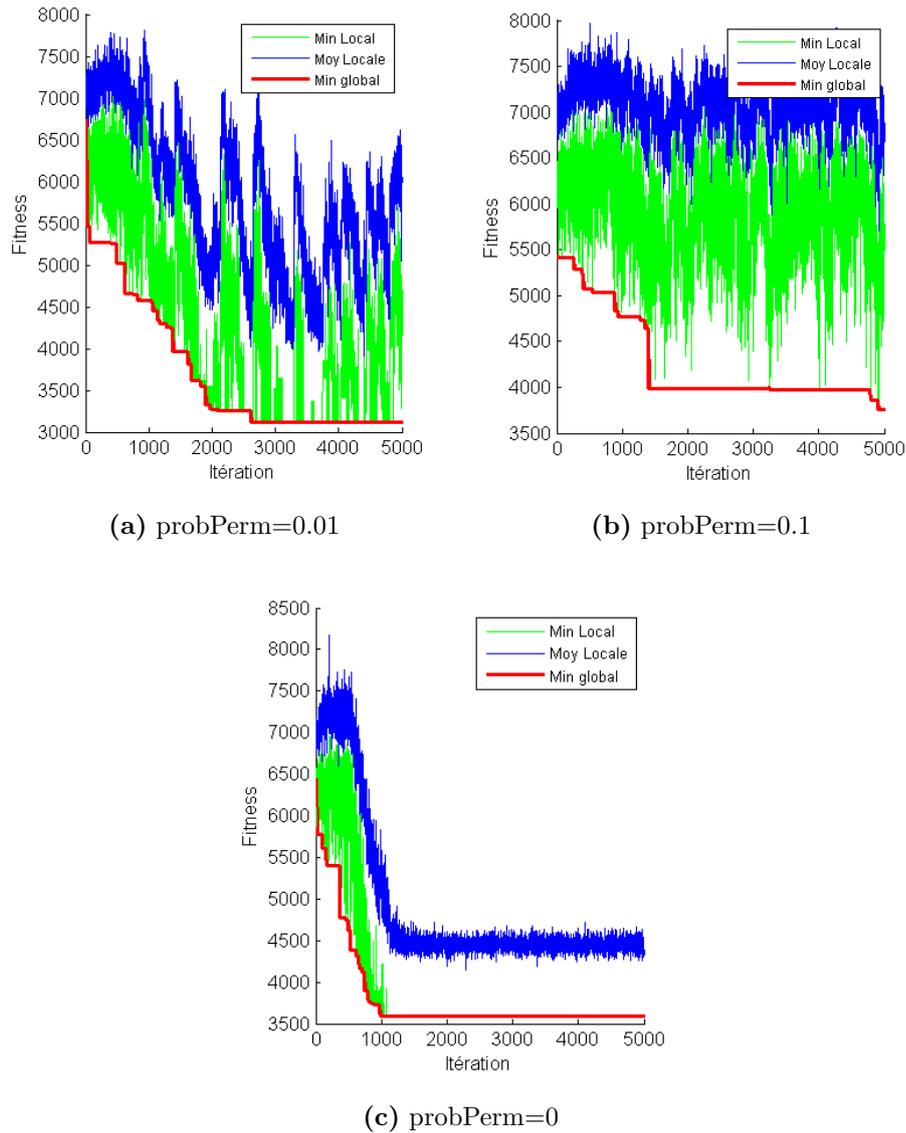


Figure 7.9: Evolution des fitness dans un LCA quantique avec taux de permutation 1%, 10% et sans permutation

7.5 Conclusion

Dans ce chapitre, les différents benchmarks utilisés pour la validation du LCA quantique proposé ainsi que les résultats expérimentaux obtenus de son application pour l'optimisation ont été expliqués et commentés. Aussi, une étude des différents opérateurs et paramètres du modèle a été abordée.

Dans le prochain chapitre, notre deuxième contribution dans cette thèse, l'évolution différentielle quantique, sera présentée. Le modèle générique de l'approche sera expliqué, et les spécificités liée aux problèmes utilisés pour la validation seront introduites.

Chapitre 8

Évolution différentielle quantique

“The greatest challenge to any thinker is stating the problem in a way that will allow a solution.” - Bertrand Russell

Sommaire

8.1	Introduction	116
8.2	Idée de base	116
8.3	Algorithme à évolution différentielle quantique	116
8.3.1	Une représentation quantique des individus	117
8.3.2	Opérateurs quantiques	117
8.3.3	Autres opérateurs	120
8.3.4	Voisinage et sous-populations	120
8.3.5	Déroulement de l'algorithme différentiel quantique	121
8.4	EDQ avec régénération de la population	121
8.5	Applications	122
8.5.1	EDQ pour la résolution du problème des N reines	122
8.5.2	ADQR pour l'optimisation de Fonctions numériques	124
8.6	Colclusionion	124

8.1 Introduction

Dans les deux chapitres précédents, on a proposé un modèle hybride pour la mise en œuvre des concepts des systèmes complexes adaptatifs, notamment l'émergence et l'adaptation : l'LCA quantique. Ce dernier a été utilisé pour la résolution de deux problèmes fameux dans le domaine de l'optimisation : le TSP et l'optimisation de fonctions numériques. Malgré les bons résultats qu'a donné l'LCA quantique, une limite majeure de ce modèle doit être prise en considération. C'est la difficulté de paramétrage. Le nombre des paramètres d'un LCA quantique est relativement élevé : cinq paramètres au minimum qui doivent être très bien gérés, sinon la convergence sera perdue.

Dans cette optique, on présente dans ce chapitre notre deuxième contribution, un autre modèle pour la simulation des CASs qui est moins complexe et plus simple à paramétrer. Il s'agit d'une hybridation entre l'évolution différentielle et l'informatique quantique [Draa *et al.*, 2010].

Dans ce chapitre, on décrit en premier lieu notre hybridation entre l'ED et l'informatique quantique : l'Évolution Différentielle Quantique (EDQ). Ensuite, on présente son application pour la résolution de deux problèmes d'optimisation : le problème des N reines et le problème d'optimisation de fonctions numériques. Finalement, une extension de l'EDQ par un opérateur de régénération de la population sera introduite.

8.2 Idée de base

L'évolution Différentielle est l'un des modèles puissants de simulation des CASs. Récemment, cette nouvelle variante des Algorithmes évolutionnaires a été intensivement utilisée pour la résolution de problèmes de calcul et a prouvé d'être meilleure que plusieurs autres approches évolutionnaires. Cependant, quelques limitations restent à régler pour que cette approche soit encore meilleure (voir Section 4.4.2 du chapitre 4).

Pour surmonter ces limites, on propose l'enrichissement de l'ED par les concepts de l'informatique quantique, notamment la superposition d'états et l'interférence (porte D). Une représentation quantique offre plus de diversité à la population, et donc, une meilleure exploration, alors que l'interférence permet de bien guider la recherche autour des meilleurs locaux et globaux, et donc, une bonne exploitation.

8.3 Algorithme à évolution différentielle quantique

Dans cette section, le framework d'un algorithme à évolution différentielle quantique sera présenté. Avant de donner le comportement global de l'algorithme, la représentation adop-

tée, l'ensemble des opérations utilisées et la relation de voisinage choisie seront expliquées.

8.3.1 Une représentation quantique des individus

L'évolution différentielle classique utilise une représentation vectorielle ou matricielle des solutions. Pour couvrir une grande partie de l'espace de recherche, le nombre d'individus représentant les solutions peut, dans certains contextes, être relativement grand. Ceci cause soit la lenteur de convergence (dans le cas d'une grande taille de la population), soit la convergence prématurée (dans le cas d'un nombre insuffisant d'individus). Pour résoudre ce problème, une représentation quantique est utilisée dans notre approche.

Un Algorithme Différentielle Quantique (ADQ) opère sur une population d'individus quantiques. Ces derniers sont des superpositions d'individus classique (des matrice ou des vecteurs). Donc, un individu quantique est représenté par un registre, ou une matrice, quantique composé de qubits, comme le montre la formule 8.1, où : α_i et β_i représentent les amplitudes des qubits, et donc, doivent vérifier la condition $|\alpha_i|^2 + |\beta_i|^2 = 1$.

$$\begin{pmatrix} \alpha_1 & \alpha_2 & \dots & \alpha_m \\ \beta_1 & \beta_2 & \dots & \beta_m \end{pmatrix} \quad (8.1)$$

Cette représentation offre une énorme diversité car un seul individu peut représenter toutes les solutions possibles en même temps. Initialement, tous les qubits d'un individu sont mis à la valeur $\sqrt{2}/2$. Ceci permet de représenter toutes les solutions possibles par un seul individu.

La figure 8.1 donne un exemple d'une représentation quantique d'une solution sous une forme matricielle.

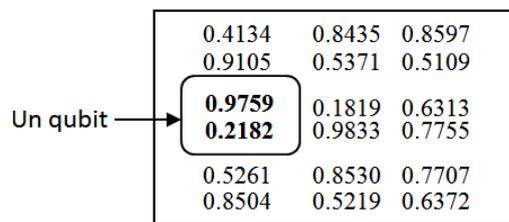


Figure 8.1: Individu quantique sous une forme matricielle

8.3.2 Opérateurs quantiques

Les opérateurs de base utilisés dans notre algorithme différentiel quantique sont la mesure quantique, l'interférence quantique et l'opérateur de mutation différentielle quantique. Ce

dernier est la version quantique qu'on a conçu de la mutation différentielle (vue dans la section 4.2.1 du chapitre 4). En plus de ces opérateurs, on utilise également, l'évaluation, et la mise à jour.

• Mutation différentielle quantique

Cet opérateur est la version quantique de l'opérateur de mutation différentielle classique. Tout en gardant le même principe de la version classique, la mutation quantique opère sur des individus quantiques avec une petite particularité liée à la nature quantique de l'individu. Un nouvel individu quantique est créé en perturbant un individu choisi aléatoirement par la différence pondérée entre trois autres individus quantiques aussi choisis aléatoirement.

Pour préserver la contrainte que la somme des carrés des amplitudes soit égale à 1, cet opérateur est appliqué à une seule composante du qubit en question, l'autre composante est déduite de la première, comme le montre la paire d'équations 8.2.

$$\begin{cases} A_{i1} = B_{i1} + F(C_{i1} - D_{i1}) & i = 1..N \\ A_{i2} = \sqrt{1 - A_{i1}^2} \end{cases} \quad (8.2)$$

Où :

$F \in [0, 2]$ est le facteur de différence.

B , C et D sont des individus quantiques choisis aléatoirement de la population.

A est l'individu résultat.

N est la longueur des individus.

Cette mutation est basée sur le schéma DE/rand/1 de mutation différentielle vu dans le chapitre 4, avec la différence que la notre opère sur des individus quantiques, et utilise le même facteur de différence F pour tous les qubits : le quantique donne assez de diversité et donc on aura pas besoin de changer le facteur F pour chaque qubit.

La figure 8.2 donne une illustration du principe de la mutation différentielle quantique. Dans cette figure, l'individu quantique D est créé en perturbant l'individu C par la différence pondérée entre les individus A et B . Le facteur de différence dans cet exemple a été mis à 0.01.

• Interférence quantique

L'opération d'interférence quantique permet de guider la population d'individus vers le voisinage des meilleurs locaux et ces derniers vers le meilleur global. Donc, on distingue

$$\left. \begin{array}{l}
 \left[\begin{array}{ccc}
 \mathbf{A} \\
 0.6398 & 0.8011 & 0.3001 \\
 0.7685 & 0.5985 & 0.9539
 \end{array} \right] \\
 \left[\begin{array}{ccc}
 \mathbf{B} \\
 0.6561 & 0.4730 & 0.3531 \\
 0.7547 & 0.8811 & 0.9356
 \end{array} \right] \\
 \left[\begin{array}{ccc}
 \mathbf{C} \\
 0.4721 & 0.6568 & 0.2085 \\
 0.8816 & 0.7541 & 0.9780
 \end{array} \right]
 \end{array} \right\} \xrightarrow[\text{mut diff}]{F=1/100} \left[\begin{array}{ccc}
 \mathbf{D=C+F*(A-B)} \\
 0.4719 & 0.6600 & 0.2079 \\
 0.8816 & 0.7512 & 0.9781
 \end{array} \right]$$

Figure 8.2: Mutation différentielle quantique

deux types d'interférence dans notre approche : *l'interférence locale* et *l'interférence globale*. La première utilise comme référence une solution quantique qui représente le meilleur d'un voisinage local d'individus. La référence pour le deuxième type est une solution binaire : c'est la meilleure solution binaire trouvée. Pour préserver la contrainte que la somme des carrés des amplitudes de chaque qubit soit égale à 1, les deux opérateurs opèrent seulement sur une composante dans chaque qubit, l'autre composante est déduite de la première.

Les paires d'équations 8.3 et 8.4 représentent respectivement comment un qubit est calculé pour une interférence locale et pour une interférence globale, où : $A \langle \alpha_A, \beta_A \rangle$ est le qubit à interférer, $Loc \langle \alpha_{Loc}, \beta_{Loc} \rangle$ est le qubit correspondant dans le meilleur individu dans le voisinage de l'individu du qubit A , $best(i, j)$ est la valeur binaire correspondante à A dans la meilleure solution globale, et $R \langle \alpha_R, \beta_R \rangle$ est le qubit résultat. CL et CG sont respectivement le coefficient d'interférence locale et le coefficient d'interférence globale.

$$\begin{cases}
 \alpha_R = \frac{\alpha_{Loc} + CL * \alpha_A}{CL + 1} \\
 \beta_C = \sqrt{1 - \alpha_C^2}
 \end{cases} \quad (8.3)$$

$$\begin{cases}
 \alpha_R = \frac{(1 - \alpha_{best}) + CG * \alpha_A}{CG + 1} \\
 \beta_R = \sqrt{1 - \alpha_R^2}
 \end{cases} \quad (8.4)$$

A travers ces deux opérateurs, on peut bien guider l'ensemble de la population pour exploiter les bonnes solutions déjà trouvées soit localement ou globalement. Le choix des

valeurs des coefficients dépend de la nature du problème à résoudre. On le fait essentiellement d'une façon empirique.

8.3.3 Autres opérateurs

En plus de la mutation différentielle quantique et des deux types d'interférence quantique vus dans la section précédente, notre algorithme utilise d'autres opérateurs quantiques et non-quantiques telles que la mesure quantique, l'évaluation et la mise à jour des meilleures solutions. Ces opérateurs ont été déjà décrits en détail dans le chapitre 6 (section 6.4). La seule différence dans l'EDQ est que ces opérateurs dans le cas de l'EDQ opèrent sur des individus quantiques au lieu de cellules. Donc, pour éviter la redondance, on ne donne dans ce qui suit que le rôle de chaque opérateur.

- **Mesure quantique**

La mesure quantique permet d'extraire un individu classique à partir d'un individu quantique. Elle se fait à base des amplitudes des qubits constituant les individus de la population quantique.

- **Évaluation**

L'évaluation permet de calculer la qualité de la solution binaire en se basant sur la fonction objectif du problème à résoudre.

- **Mise à jour des meilleures solutions**

Dans cette étape, les meilleures solutions locales et globales sont mises à jour. Le meilleur local pour chaque groupe d'individus, sous-population, est remplacé par un individu quantique de la même sous population si et seulement si l'évaluation de la solution binaire obtenue de la mesure de ce dernier donne un résultat meilleur que celui du meilleur local. Le meilleur global est remplacé par une solution binaire (extraite de la mesure d'un individu quantique) si et seulement si l'évaluation donne une meilleure valeur que celle trouvée jusqu'à la génération courante.

8.3.4 Voisinage et sous-populations

Une sous-population est définie à travers une relation de voisinage entre individus. Cette relation peut être de n'importe quel type, par exemple : selon les fitness des individus, selon leurs évolution, ou simplement d'une façon aléatoire. Dans notre approche, on opté pour le choix aléatoire pour la simple raison que tous les individus sont initialisés de la même valeur initiale : $\alpha = \beta = \sqrt{2}/2$, et donc, tous les individus sont identiques.

La population dans notre algorithme différentiel quantique est subdivisée en k sous-populations, avec l individus par sou-population. Ces sous populations n'ont en commun que le meilleur global vers lequel les meilleurs locaux sont interférés.

8.3.5 Déroulement de l'algorithme différentiel quantique

Après avoir expliqué chaque opérateur utilisé par notre approche, on donne la structure générique de notre algorithme. Après une étape d'initialisation de la population, des meilleurs locaux et du meilleur global, l'AED quantique applique cycliquement les opérations suivantes jusqu'à la satisfaction d'une condition d'arrêt.

1. Mesure quantique
2. Évaluation
3. Comparaison et mise à jours des meilleures solutions
4. Interférence (locale/globale)
5. Mutation différentielle quantique

Ici, on doit noter que les interférences locale et globale se font avec des probabilités : $locProb$ et $gloProb$, et que le facteur de différence dépend du problème à résoudre, et est choisi empiriquement.

8.4 EDQ avec régénération de la population

Comme on le verra dans la section 9.2.2 du prochain chapitre, malgré que l'algorithme différentiel quantique donne des bons résultats, il face des difficultés à s'échapper des minima locaux. Ceci est principalement causé par le fait qu'avec le temps, les individus quantiques deviennent égaux à cause de l'impact de l'opérateur d'interférence (local ou global), et donc, la mutation différentielle sera incapable de créer la diversité au sein de la population : la différence entre deux individus similaires donne toujours des zéros.

Pour pouvoir résoudre ce problème et s'échapper des minima locaux, on a proposé une extension de l'AED quantique enrichie par un opérateur de régénération de la population [Draa *et al.*, 2011]. Cette extension est appelée Algorithme Différentiel Quantique avec Régénération (ADQR).

Le principe de l'opérateur de régénération est simple : selon une probabilité $regProb$ une portion de la population (généralement le tiers ou la moitié) est régénérée en remplaçant des individus choisis aléatoirement par de nouveaux individus dont la valeur des amplitudes des qubits est égale à $\sqrt{2}/2$. Cette régénération offre une nouvelle diversité à l'algorithme et donc permet de s'échapper des minima locaux. Notre ADQR se comporte

alors de la même manière qu'un AEDQ (vu à la section 8.3.5) avec l'introduction de l'opérateur de régénération. Donc, après une étape d'initialisation, on applique les étapes suivantes jusqu'à la satisfaction d'une condition d'arrêt.

1. Régénération { Avec une probabilité $regProb$ }
2. Mesure quantique
3. Évaluation
4. Comparaison et mise à jours des meilleures solutions
5. Interférence (locale/globale)
6. Mutation différentielle quantique

8.5 Applications

Pour valider les performances du modèle proposé, deux applications ont été implémentées. La première est la résolution du problème des N reines. Dans la deuxième, l'optimisation des fonctions numériques a été considérée. Dans cette section, ces deux applications sont présentées.

8.5.1 EDQ pour la résolution du problème des N reines

• Formulation du problème

Le problème des N reines (NQP : de l'anglais N-Queens Problem) est une généralisation du problème des 8 reines, un problème d'optimisation combinatoire introduit en premier lieu par Max Bezzel, un joueur d'échecs, en 1848. Depuis 1850, ce problème a attiré l'attention de fameux mathématiciens, notamment Gauss, Polya et Lucas [Talbi, 2009].

Le problème des N reines vise, comme son nom l'indique, à placer N reines sur un échiquier de dimensions $N \times N$ de façon à satisfaire la contrainte que deux reines ne peuvent pas être prises l'une par l'autre, i.e., toute paire de reines ne peut pas être sur la même ligne, colonne ou diagonale [Berger, 2010]. La figure 8.3 Donne un exemple d'une solution du problème des 8 reines. Une des formes classiques de résoudre ce problème est d'utiliser le backtracking, mais sa complexité est presque exponentielle à la taille du problème, ce qui fait que d'autres méthodes ont été développées pour le résoudre comme le branch and bound entre autres [Tazi et Pablo, 2008].

Pour résoudre ce problème, on a utilisée l'AED quantique décrit dans les sections précédentes avec une représentation matricielle des individus quantiques et en adoptant la fonction objectif décrite dans ce qui suit.

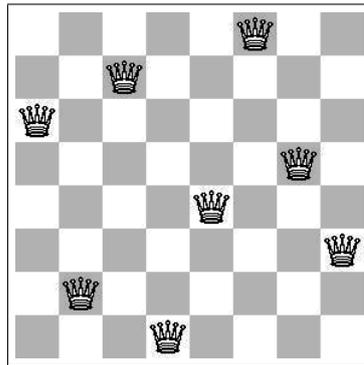


Figure 8.3: Une solution du problème des 8 reines

- **Représentation des solutions**

Une représentation d'une solution pour le problème des N reines peut être une matrice de taille $N \times N$ contenant N uns et des zéros (Figure 8.4). Les "uns" représentent les reines de l'échiquier.

0	0	1	0
1	0	0	0
0	0	0	1
0	1	0	0

Figure 8.4: Une représentation binaire d'une solution du problème des 4 reines

- **Fonction objectif**

La pénalité d'une reine est égale au nombre de reines en conflit avec elle : dans une même ligne, colonne ou diagonale. La fitness d'une configuration donnée de reines est égale à la somme des pénalités de toutes les reines divisée par 2 (pour éliminer la redondance). Donc, la fonction objectif utilisée pour évaluer un résultat est égale au nombre de paires de reines pouvant capturer l'une l'autre. Par exemple, la fitness de la configuration présentée par la figure 8.5 est égale à 5.

- **Une mesure quantique avec une contrainte**

La mesure quantique utilisée pour ce problème, permettant d'extraire une solution binaire à partir d'un individu quantique, doit satisfaire la condition que le nombre des "uns" soit égale à N .

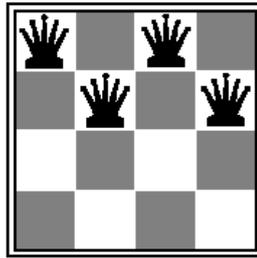


Figure 8.5: Une configuration des reines avec fitness égale à 5

• Opérateur de permutation

C'est le même opérateur utilisé par l'LCA quantique pour le TSP (Chapitre 6, section 6.5.1) : une interversion entre deux blocs de colonnes quantiques de même taille se fait dans quelques individus quantiques selon une probabilité qu'on appelle ici *permProb*.

Les résultats expérimentaux obtenus de l'application de l'EDQ pour résoudre le problème des N reines, ainsi que l'étude des différents opérateurs de l'algorithme seront abordés dans le chapitre suivant.

8.5.2 ADQR pour l'optimisation de Fonctions numériques

La variante de l'EDQ utilisant l'opérateur de régénération, ADQR, a été utilisée pour résoudre le problème d'optimisation de fonctions numériques, déjà décrit dans les chapitres 6 et 7. Les fonction benchmarks ont été choisies d'être les mêmes utilisées dans le chapitre précédent (Tableau 6.1). Les résultats expérimentaux de cette application seront détaillés dans le chapitre suivant.

8.6 Colclusion

Dans ce chapitre, notre deuxième contribution pour la modélisation et la mise en œuvre des CASs a été présentée ; l'évolution différentielle quantique. On a décrit la façon d'hybridation entre l'informatique quantique et l'évolution différentielle. Malgré sa capacité d'adaptation, ce modèle lui même a été limité par le problème de la convergence prématurée, et donc la stagnation de l'algorithme dans les minima locaux. Pour surmonter cette limite, une nouvelle variante du modèle proposé a été conçue, il s'agit d'algorithme à évolution différentielle quantique avec régénération de la population.

L'évolution différentielle quantique (l'EDQ) et sa variante avec régénération (l'ADQR) ont été utilisées pour la résolution de deux problème célèbres d'optimisation, le problème

des N reines et l'optimisation de fonctions numériques. Aussi, les spécificités du modèle liées aux problèmes en question ont été présentées.

Dans le chapitre suivant, les résultats expérimentaux obtenus de l'application de l'EDQ avec ses deux variantes seront présentés. De plus, une discussion des différents opérateurs et paramètres de l'algorithme sera abordée.

Chapitre 9

Validation de l'évolution différentielle quantique pour l'optimisation

“If we knew what it was we were doing, it would not be called research,
would it? ” - Albert Einstein

Sommaire

9.1	Introduction	128
9.2	L'EDQ pour le problème des N reines	128
9.2.1	EDQ sans régénération de la population	128
9.2.2	Étude des opérateurs	129
9.2.3	EDQ avec régénération de la population	132
9.3	EDQ avec régénération pour l'optimisation de Fonctions	133
9.4	Discussion	133
9.4.1	Taille de la population	136
9.4.2	Taux de régénération	136
9.4.3	Taux de différence de la mutation différentielle	137
9.4.4	Taux d'interférences locale et globale	138
9.4.5	Coefficients d'interférence	139
9.5	Conclusion	139

9.1 Introduction

Dans le chapitre précédent, on a présenté notre deuxième contribution : une approche différentielle quantique pour la mise en œuvre des systèmes complexes adaptatifs. On a présenté l'approche générique du modèle et les deux applications utilisées comme base de validation de ce modèle : le problème des N reines et le problème d'optimisation de fonctions numériques.

Dans ce chapitre, la validation expérimentale de l'approche différentielle quantique sera détaillée. On aborde les résultats expérimentaux obtenus et on discute les opérateurs des deux variantes proposées de cette l'approche.

9.2 Evolution différentielle quantique pour la résolution du problème des N reines

9.2.1 EDQ sans régénération de la population

Notre algorithme différentiel quantique pour la résolution du problème des N reines utilise la même structure décrite dans la section 8.3.5 avec l'inclusion de l'opérateur de permutation. On a appliqué l'algorithme pour résoudre différentes instances du problème des reines pour $N=4$ jusqu'à $N=12$ (pour $n=1$ à 3 , le problème est mathématiquement insolvable). Les résultats obtenus ont été très prometteurs comme le montre le tableau 9.1, où : N est le nombre de reines, CL est le coefficient d'interférence locale, CG est le coefficient d'interférence globale, et $Max.Gen$ est le nombre maximum de générations [Ammi *et al.*, 2011].

Table 9.1: Résultats expérimentaux de l'EDQ quantique pour le problème des N reines

N	CL	CG	Max.Gen	% Solutions optimales
4	30	30	10000	100%
5	30	30	10000	100%
6	35	30	30000	100%
7	35	30	50000	100%
8	30	30	80000	100%
9	30	30	150000	100%
10	30	30	100000	88%
11	30	30	100000	39%
12	30	30	100000	26%

Comme on le voit dans le tableau 9.1, l'algorithme donne une bonne convergence, surtout pour les petites instances du problème : $N=4..10$. Pour les grandes instances, $N>11$ ça nécessite plus de temps. La figure 9.1 présente l'évolution de la meilleure fitness globale (en rouge), de la meilleure fitness de chaque génération (en vert) et de la moyenne des fitness de chaque génération (en bleu). On voit bien que la fitness moyenne et minimale sont guidées par la fitness globale, c'est l'apport des opérateurs d'interférence (locale et globale).

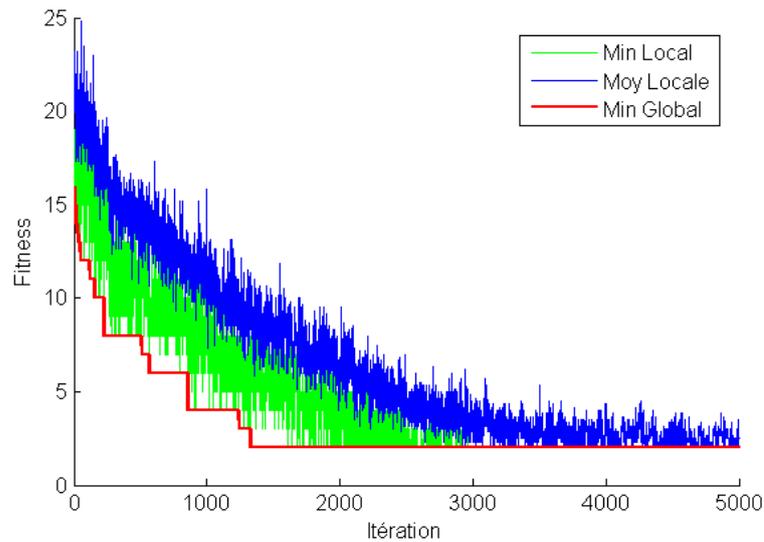


Figure 9.1: Évolution de la fitness des solutions : meilleur local (en vert), moyenne des fitness de chaque itération (en bleu) et meilleur global (en rouge)

La figure 9.2 compare l'évolution de la fitness (locale et globale) dans un algorithme génétique classique (en bleu) et dans notre approche différentielle quantique (en rouge).

9.2.2 Étude des opérateurs

Pour une analyse plus fine des opérateurs de notre algorithme, on exécute chaque fois ce dernier avec et sans application du paramètre en question. La figures 9.3 représente une comparaison entre l'évolution du meilleur global dans une recherche avec permutation (en rouge) et sans permutation (en bleu).

Dans les figures 9.4 et 9.5 on compare entre une recherche avec l'application d'une interférence locale et une recherche sans son application. La première figure montre la différence en terme de l'évolution du meilleur local et de la moyenne par rapport au meilleure global. On voit bien que dans le cas de l'absence de cet opérateur (à gauche), la recherche n'est pas bien guidée, comme l'est dans le cas de sa présence (à droite).

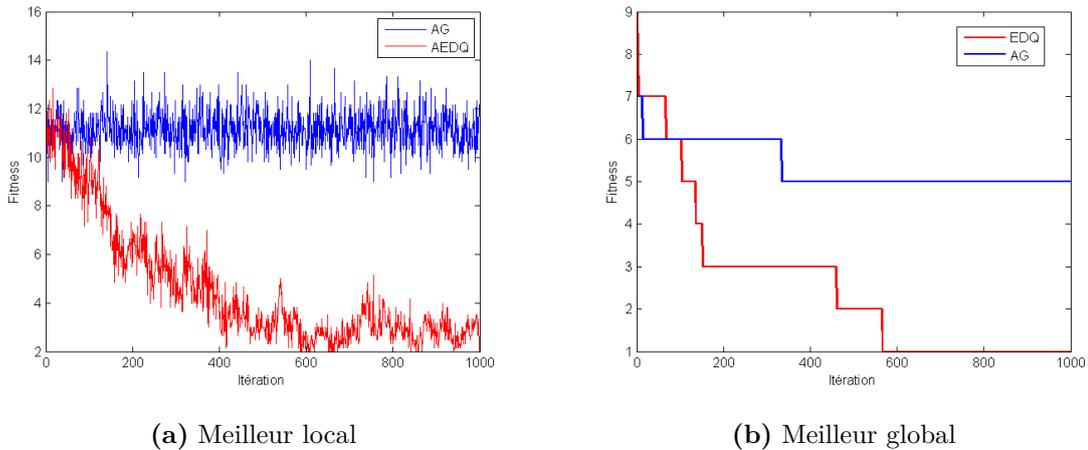


Figure 9.2: Une comparaison entre l'algorithme génétique classique et l'algorithme différentiel quantique pour le problème des N reines

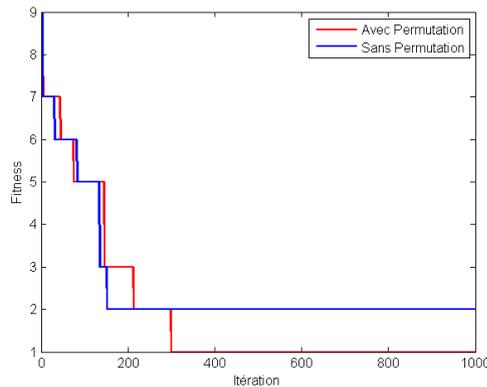


Figure 9.3: Comparaison entre une recherche avec permutation (en rouge) et une recherche sans permutation (en bleu)

Dans la deuxième figure, l'effet de l'application et de l'absence de l'interférence locale en terme d'évolution du meilleur global est illustré.

De la même façon, les figures 9.6 et 9.7 offrent des comparaisons entre évolutions des fitness dans le cas de l'application et de l'absence de l'interférence globale et de la mutation différentielle respectivement.

Après l'analyse de ces figures, on peut bien observer l'apport qu'apporte chaque opérateur sur la convergence de l'algorithme, notamment l'opérateur d'interférence. Ce dernier permet de guider la recherche et d'exploiter l'espace de recherche dans le voisinage du meilleur local (dans le cas d'interférence locale) ou du meilleur global (dans le cas d'interférence globale).

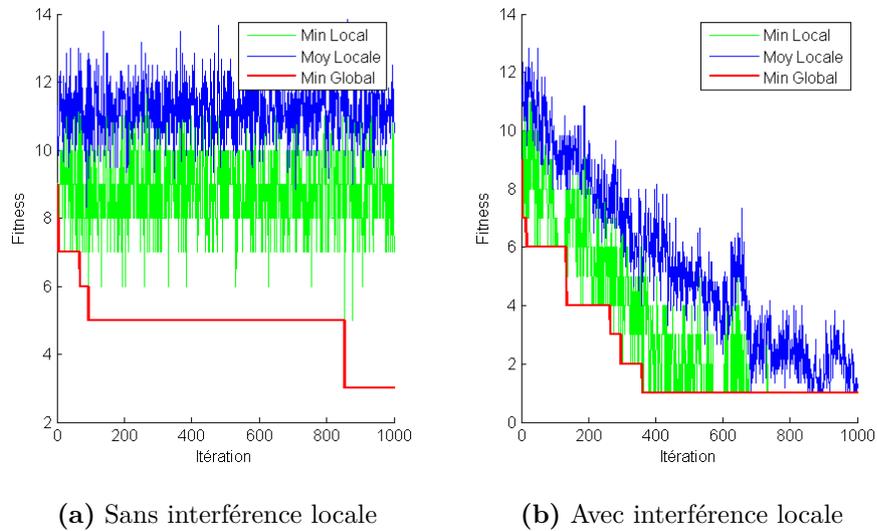


Figure 9.4: Une recherche sans interférence locale (à gauche) et une recherche avec interférence locale (à droite)

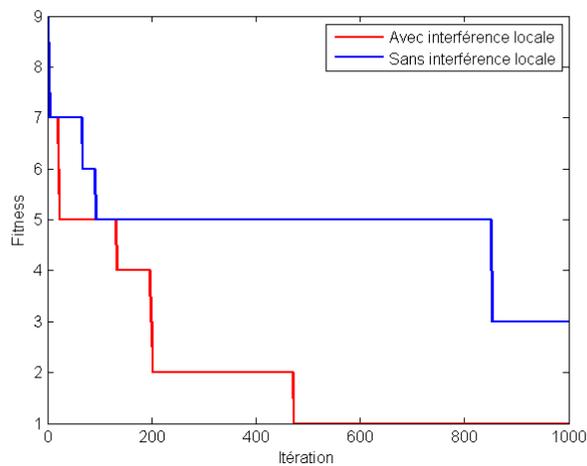


Figure 9.5: Comparaison entre une recherche avec interférence locale (en rouge) et une recherche sans interférence locale (en bleu)

Des, on observe une pseudo plantation de l'algorithme différentiel quantique dans les minimas locaux : l'algorithme passe par des périodes de silence (des fois jusqu'à 600 itérations sans évolution), et donc, l'AED quantique n'a pas résolu le problème de convergence prématurée de l'évolution différentielle classique à 100%. C'est parmi les raisons de notre proposition de l'opérateur de régénération, comme vu dans le chapitre précédent.

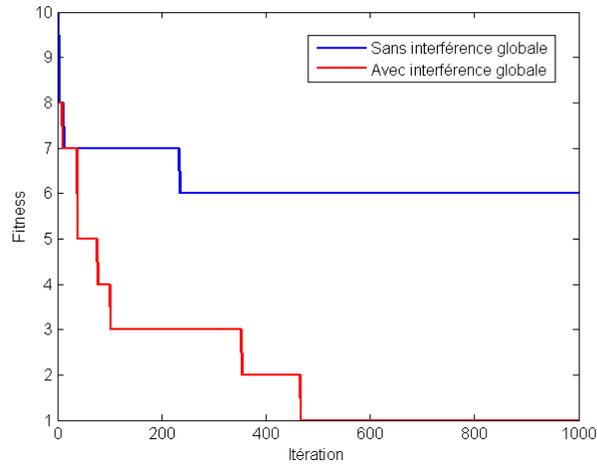


Figure 9.6: Comparaison entre une recherche avec interférence globale (en rouge) et une recherche sans interférence globale (en bleu)

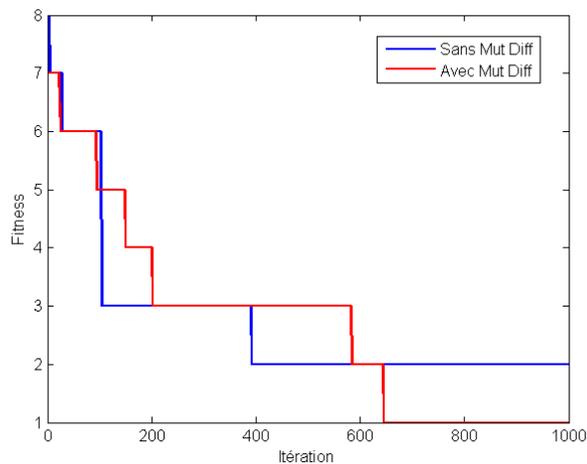


Figure 9.7: Comparaison entre une recherche avec mutation différentielle (en rouge) et une recherche sans mutation différentielle (en bleu)

9.2.3 EDQ avec régénération de la population

L'application de cette nouvelle variante de l'EDQ sur le problème des N reines a prouvé l'utilité de cet opérateur de régénération de la population pour l'obtention de plus de diversité, et donc, une meilleure exploration de l'espace de recherche. Le tableau 9.2 fait une comparaison entre les résultats obtenus de l'ADQR et ceux de l'EDQ.

On voit bien du tableau 9.2 que la version de l'ADQ avec régénération de la population est meilleure (voir les pourcentages marqués en gras). L'efficacité de cet opérateur se voit bien dans les figures 9.8 et 9.9. Dans la première, on compare l'évolution des meilleurs

Table 9.2: Résultats expérimentaux de l'ADQR pour le problème des reines

N	AEDQ (sans régénération)	ADQR (avec régénération)
4	100%	100%
5	100%	100%
6	100%	100%
7	100%	100%
8	100%	100%
9	100%	100%
10	88%	94%
11	39%	74.1%
12	26%	63.25%

locaux (en vert) globaux (en rouge) et de la moyenne des fitness de chaque itération (en bleu) entre le cas d'application de l'opérateur de régénération (à gauche) et sans son application (à droite). La deuxième figure présente une comparaison entre l'évolution du meilleur global dans le cas de régénération (en rouge) et dans le cas de son absence (en bleu).

9.3 EDQ avec régénération pour l'optimisation de Fonctions

En plus de sont application à la résolution du problème des N reines, on a utilisé l'algorithme différentiel quantique avec régénération pour résoudre le problème d'optimisation de fonctions numériques. Les fonctions benchmarks ont été choisies d'être les mêmes utilisées dans le chapitre 6 (Tableau 6.1). Le tableau 9.3 résume les résultats expérimentaux obtenus (en gras) et les compare avec ceux obtenus de l'LCA quantiques décrits dans le chapitre précédent et ceux présentés dans [Huang *et al.*, 2008].

Après une petite analyse des résultats obtenus de l'ADQR et leur comparaison avec ceux obtenus du LCA quantique (Tableau 9.3), on peut dire que l'ADQR et l'LCA quantique offrent des performances de calcul assez proches l'un de l'autre et sont largement meilleures que celles offertes par le PSO quantique de Huang *et al.* [2008] et les algorithmes évolutionnaires classiques.

9.4 Discussion

Dans les sections précédentes, on a présenté l'EDQ et son extension par l'opérateur de régénération de la population, et on a présenté l'application des deux variantes pour

Table 9.3: Résultats expérimentaux de l'ADQR pour l'optimisation de fonctions numériques

Fonction	Dim	ADQR		QLCA		QSEA		CEA	
		Moy.Best	Ecart	Moy.Best	Ecart	Moy.Best	Ecart	Moy.Best	Ecart
Rosenbrock	10	8.0535	0.3445	8.1373	0.2298	10.423	14.479	71.023	174.110
	30	28.3452	0.1995	33.7450	9.0768	59.028	63.485	289.583	478.625
	10	7.9562	987	7.9116	0.3641	8.634	16.589	37.365	57.431
	30	28.8507	0.1117	28.3638	0.1908	51.565	40.843	202.456	289.968
Griewank	10	0.0025	0.007	0.0038	0.0098	0.069	0.051	0.085	0.073
	30	0.0081	0.017	0.0055	0.0229	0.012	0.017	0.013	0.015
	10	1.6E-003	0.0021	2.05E-004	0.0015	0.034	0.033	0.075	0.071
	30	2.3E-007	1.3E-005	1.93E-006	7.44E-006	0.011	0.009	0.013	0.014
Sphere	10	0	0	0	0	8.25E-72	5.82E-71	3.11E-23	8.01E-23
	30	0	0	0	0	1.85E-28	6.71E-28	2.26E-10	5.08E-10
	10	0	0	0	0	3.09E-100	2.09E-99	6.13E-28	2.64E-27
	30	0	0	0	0	1.09E-48	2.66E-48	2.47E-12	7.16E-12

Dim : Dimension de la fonction ; Taille : la taille de l'automate ; Max.It : Nombre maximum de générations ;
 QLCA : Résultats du QLCA ; ADQR : Résultats de l'ADQR ;
 QSEA : Résultats du PSO quantique [Huang *et al.*, 2008] ; CEA : Résultats de l'AE classique ;
 Moy.Best : la moyenne des meilleures solutions obtenues dans 50 exécutions ; Ecart : L'écart type statistique.

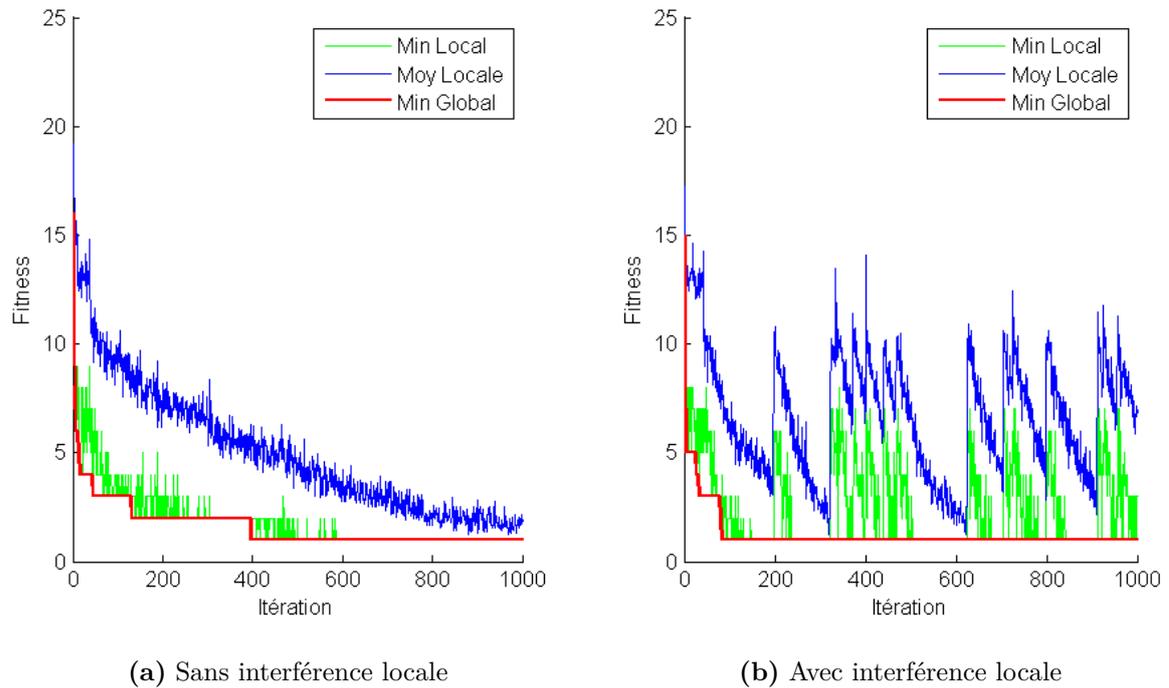


Figure 9.8: Une recherche sans régénération (à gauche) et une recherche avec régénération (à droite)

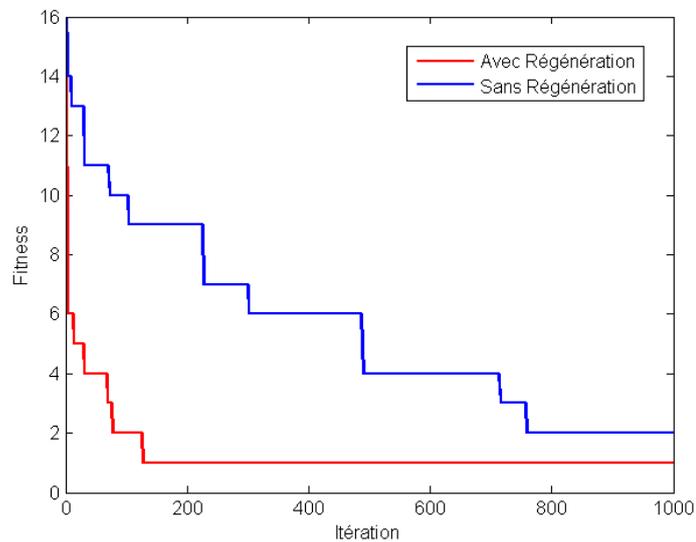


Figure 9.9: Comparaison entre une recherche avec régénération (en rouge) et une recherche sans régénération (en bleu)

résoudre le problème des N reines (en utilisant les deux variantes) et le problème d'optimisation de fonctions numériques (en utilisant la variante avec régénération). Les deux

variantes de l'évolution différentielle quantique ont prouvé d'être meilleures que le PSO quantique et les AEs classiques (selon les résultats présentés par Huang *et al.* [2008]).

Dans cette section, on va procéder à une étude de l'influence du changement des valeurs des différents paramètres de l'algorithme sur la convergence de l'algorithme. Dans ce qui suit, tous les tests seront faits selon la moyenne des résultats de 10 exécutions de l'algorithme pour chaque valeur du paramètre en question d'étude pour la fonction *Rosenbrock* avec 10 variables (Tableau 6.1).

9.4.1 Taille de la population

La taille de la population n'est pas obligée d'être très large dans le cas d'un algorithme inspiré du quantique (grâce à la représentation quantique des individus). Pour vérifier cette propriété dans notre algorithme, on a exécuté l'algorithme 10 fois avec trois configurations différentes de la taille de la population : respectivement, 5×8 , 4×5 et 2×3 individus. Les moyennes des meilleures solution obtenues pour les 10 exécutions ont été 8.5381 pour 40 individus, 8.7850 pour 20 individus et 8.8294 pour 6 individus, respectivement. Selon ces résultats et leur comparaison avec ceux du tableau 9.3, on peut dire que la taille de la population peut être très réduite dans le cas d'un AED quantique, tout en s'assurant qu'il converge et reste toujours meilleur qu'un PSO quantique et qu'un AE classique.

9.4.2 Taux de régénération

L'opérateur de régénération permet de s'échapper des minima locaux. Il est appliqué avec un taux que ne doit pas être très grand (pour éviter une recherche aléatoire) ni très petit (pour garder la diversité). On a exécuté notre algorithme avec 4 sous-populations, 5 individus par chacune en mettant les valeurs de la probabilité de régénération à 0.001, 0.02, 0.10 et 0.2 et on a obtenu les moyennes de fitness de 10 exécutions : 319.9930, 8.5704, 11.2812 et 14.5992, respectivement.

Donc, la régénération ne doit pas être très fréquente, sinon à chaque fois que les individus commencent à trouver un bon chemin vers la solution optimale ils seront remplacés par des nouveaux individus. Ceci est bien montré dans la figure 9.10. Cette figure donne une bonne illustration de la nécessité du bon choix du taux de régénération, la sous-figure 9.10(a) représente le cas d'une pauvre exploration alors que les sous-figures 9.10(c) et 9.10(d) sont équivalentes à une régénération fréquente ce qui donne une recherche presque aléatoire. La sous-figure 9.10(b) représente le cas bien équilibré.

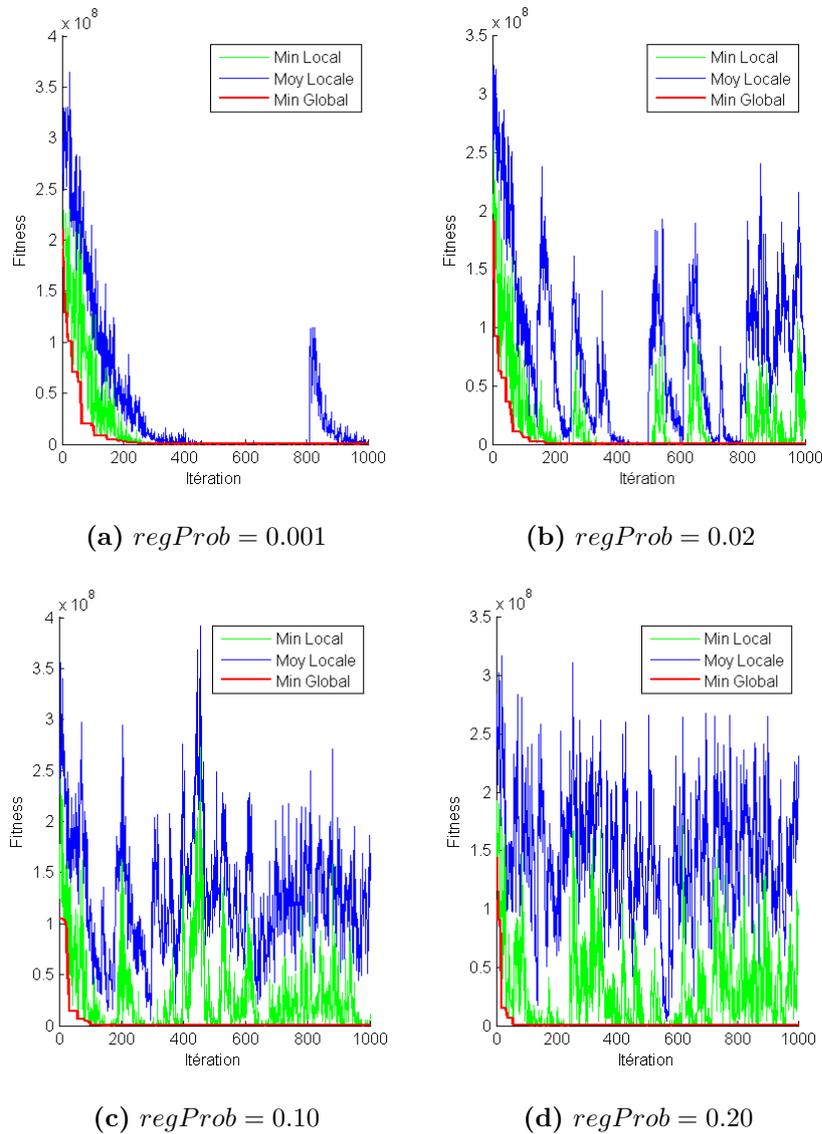


Figure 9.10: Des recherche avec des taux de régénération de 0.1%, 2%, 10% et 20%

9.4.3 Taux de différence de la mutation différentielle

Pour l'étude de l'impact du choix du taux de différence de la mutation différentielle sur la convergence de l'algorithme, on a exécuté l'algorithme 10 fois avec les taux 0.1%, 1% 10% et 100% ce qui a donné respectivement des moyennes de la meilleure solution égales à 11.3165, 8.9152, 12.1042 et 55.0331. Ceci prouve que ce facteur d'amplification ne doit pas être assez grand afin que la diversité ne résulte pas à une recherche aléatoire (Figure 9.11).

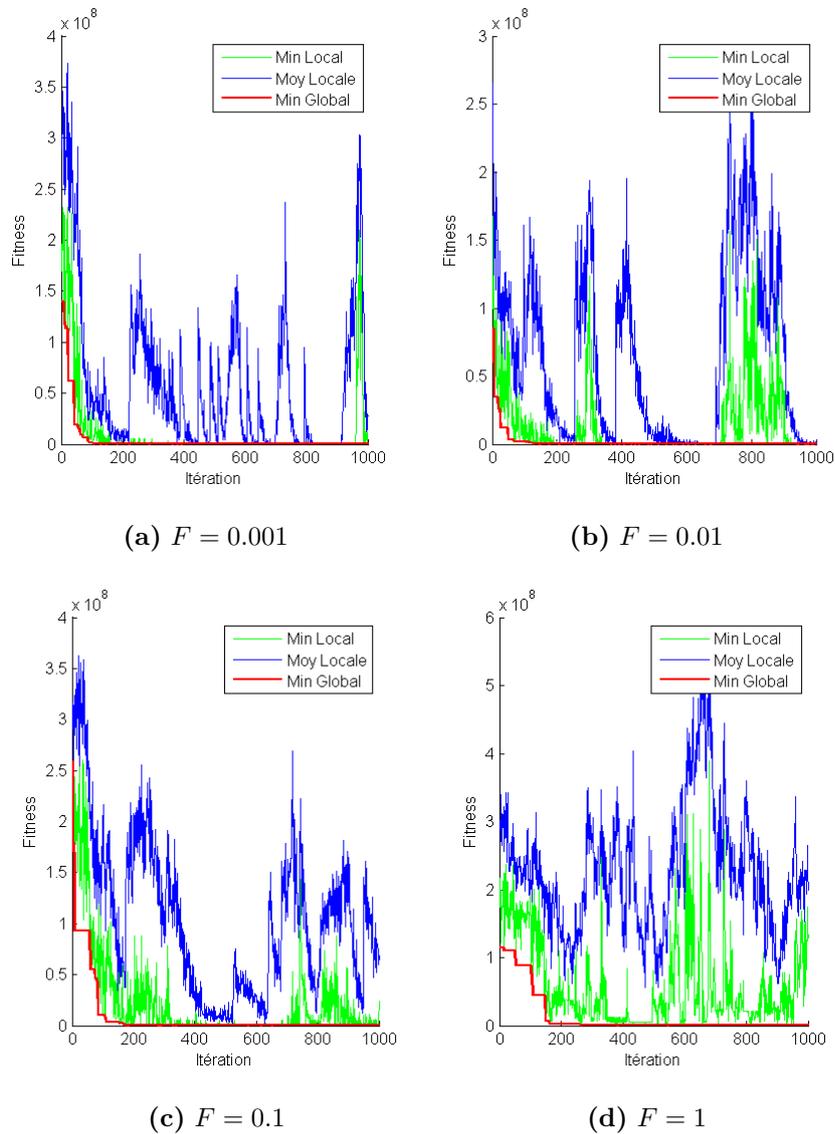


Figure 9.11: Des recherche avec des taux de différence de 0.1%, 1%, 10% et 100%

9.4.4 Taux d'interférences locale et globale

Pour analyser l'impact du choix des taux d'interférence locale et globale sur le comportement de l'algorithme, on a exécuté l'algorithme pour différentes valeurs pour les deux cas en fixant la valeur de l'une et changeant la valeur de l'autre. Les valeurs utilisées ainsi que les résultats obtenus sont résumés dans le tableau 9.4. Selon ces résultats, il est clair que les grands taux d'interférence globale ainsi que les petits taux d'interférence locale vont mener à la convergence prématurée (dans le premier cas) ou à une recherche presque aléatoire (dans le deuxième cas).

Table 9.4: Résultats obtenus selon différents taux d'interférences locale et globale

% Interférence locale	% Interférence globale	Moyenne des fitness
1%	1%	5.9330E+006
5%1	1%	3.6174E+003
10%	1%	8.8978
50%	1%	8.6670
1%	10%	2.1431E+007
1%	50%	2.2491E+007
10%	10%	3.4621E+004

9.4.5 Coefficients d'interférence

Le coefficient d'interférence spécifie à quel degré va le qubit en question contribuer à la constitution du qubit résultat. On a exécuté l'algorithme 10 fois pour différentes configurations des coefficients d'interférence locale (CL) et globale (CG) et on calculé la moyenne des fitness comme le montre le tableau 9.5. De ces résultats, on déduit qu'on doit bien équilibrer ces coefficients pour aboutir à une bonne convergence.

Table 9.5: Résultats obtenus selon différents coefficients d'interférences locale et globale

CL	CG	Moyenne des fitness
30	5	1.7570E+006
100	5	2.1201E+007
5	5	8.9443
5	30	175.5854
5	100	65.6699

9.5 Conclusion

Dan ce chapitre, les résultats expérimentaux obtenus de l'application de l'évolution différentielle quantique pour la résolution du problème des N reines et de l'optimisation de fonctions numériques ont été présentés. Une étude détaillée des paramètres et des opérateurs de l'algorithme a permis la compréhension du comportement de ce dernier et a montré l'utilité de chaque paramètre et son influence sur la convergence globale de l'algorithme.

Conclusion générale

Dans la littérature de la science de complexité, deux modèles artificiels sont les plus utilisés pour la modélisation et la mise en œuvre des systèmes complexes adaptatifs (CASs), à savoir les systèmes multi-agents et les automates cellulaires. Cependant, ces deux paradigmes font face à certaines limites les empêchant de bien ressortir les propriétés clé de ces systèmes et limitant leur exploitation pour la résolution de problèmes. Dans cette optique, nous nous sommes intéressés, dans le cadre de cette thèse, à la proposition de modèles pour la mise en œuvre de systèmes complexes adaptatifs. Deux contributions ont été présentées.

Dans la première contribution, nous avons conçu un automate cellulaire apprenant quantique. Ce modèle est le résultat de l'enrichissement d'un automate cellulaire par la notion de l'environnement d'une part, et par l'utilisation des concepts de l'informatique quantique d'une autre part. Un schéma d'apprentissage par renforcement a été adopté pour la gestion des interactions entre l'automate et son environnement. Des opérations quantiques à base du modèle de boids de Reynolds ont été définies et utilisées comme opérations offertes aux cellules de l'automate.

Le modèle de boids a été choisi pour la raison d'être capable de bien simuler l'intelligence collective : des opérations très basiques des boids (agents du CAS) offrent un comportement global de haute complexité. L'introduction de la notion d'environnement dans ce modèle permet au système de fonctionner et survivre dans un environnement ouvert. De plus, la représentation quantique des cellules de l'automate offre une bonne diversité à ce dernier, et lui permet de couvrir une grande portion de l'espace de recherche.

Pour valider ce modèle, et prouver sa capacité à résoudre des problèmes d'optimisation, nous l'avons utilisé pour la résolution de deux problèmes célèbres dans le domaine d'optimisation : le problème du voyageur de commerce et le problème d'optimisation de fonctions numériques. Les résultats expérimentaux obtenus, qui ont été comparés à ceux du PSO quantique et des algorithmes évolutionnaires classiques, ont montré l'efficacité du modèle pour la résolution des problèmes d'optimisation.

Dans la deuxième contribution, nous avons proposé un algorithme à évolution diffé-

rentielle quantique pour la mise en œuvre des CASs. Une représentation quantique des individus a été utilisée afin d'augmenter la diversité au sein de la population. Par conséquence, des opérations quantiques ont été définies et appliquées. En plus de l'introduction du quantique dans l'algorithme à évolution différentielle, nous avons introduit la notion de voisinage permettant d'exploiter les bonnes solutions locales trouvées au fil du temps. Les opérations d'interférence locale et globale utilisées dans ce deuxième modèle implémentent la notion d'attracteurs, notion clé dans les systèmes complexes adaptatifs. Les meilleures solutions locale et globale ont été utilisées comme un outil d'exploitation, alors que la mutation différentielle quantique a été utilisée comme un outil d'exploration.

Ce deuxième modèle a été également validé par son application à la résolution du problème des N reines et du problème d'optimisation de fonctions numériques, respectivement. Les résultats expérimentaux ont montré la supériorité de l'évolution différentielle quantique par rapport au PSO quantique et aux algorithmes évolutionnaires quantiques. Cependant, nous avons observé, de l'analyse du comportement de ce deuxième modèle, l'occurrence, à un certain degré, du problème de convergence prématurée. Pour surmonter ce problème, une extension de l'évolution différentielle quantique qui utilise un opérateur de régénération de la population a été proposée. Cette variante, l'évolution différentielle quantique avec régénération, a été validée à travers la résolution du problème d'optimisation de fonctions numériques. L'opérateur de régénération garantit une création périodique de nouveaux individus à intégrer dans la population. Ceci a diminué remarquablement le taux de convergence prématurée.

Au cours de la réalisation de ce travail, quelques difficultés ont été rencontrées. Nous citons les suivantes :

- La nature pluridisciplinaire du domaine des systèmes complexes adaptatifs et la non standardisation des définitions des concepts liés aux sciences de la complexité.
- L'orientation de la quasi-totalité des contributions autour des systèmes complexes et des systèmes complexes adaptatifs vers l'application plus que la modélisation. Ceci a rendu difficile la validation des modèles proposés des CASs, seule une validation expérimentale est possible.
- L'absence d'approches d'ingénierie de l'émergence. Ceci est dû à la nature imprévisible du phénomène. Donc, le concepteur d'un modèle pour les CASs doit rester dans une zone très étroite entre l'émergence et l'ingénierie. Un compromis doit avoir lieu pour concevoir des modèles permettant la résolution de problèmes via un comportement émergent.

Enfin, dans une perspective d'amélioration des performances de nos approches, nous proposons :

- D'essayer de diminuer le nombre d'opérateurs dans les deux approches afin de diminuer le nombre de paramètres à régler et de diminuer la complexité algorithmique.
- D'établir un framework plus consistant et plus générique pour la conception de modèles de CASs.
- De travailler pour la mise en place de définitions plus riches et plus liées au domaine informatique des concepts des CASs, notamment l'adaptation, l'apprentissage, l'auto-organisation et l'émergence.

Bibliographie

- ABIN, A. A., FOTOUHI, M. et KASAEI, S. (2008). Skin Segmentation Based on Cellular Learning Automata. *In Proceedings of the 6th International Conference on Advances in Mobile Computing and Multimedia*, MoMM'08, pages 254–259. ACM.
- AKBARI TORKESTANI, J. et MEYBODI, M. R. (2011). A Cellular Learning Automata-Based Algorithm for Solving the Vertex Coloring Problem. *Expert Syst. Appl.*, 38(8): 9237–9247.
- AL-RABADI, A. et LENDARIS, G. (2003). Artificial Neural Network Implementation using Many-Valued Quantum Computing. *In Proceedings of the International Joint Conference on Neural Networks*, pages 3112–3117.
- ALI, S. M. et ZIMMER, R. M. (1997). The Question Concerning Emergence : Implications For Artificiality. *In CASYS'97, Abstract Book, First International Conference on Computing Anticipatory Systems, CHAOS asbl.*
- AMMI, M., DRAA, A. et CHIKHI, S. (2011). A Regenerating Quantum-Inspired Differential Evolution Algorithm for Solving the N-Queens Problem. *In Proceedings of the 7th International Computing Conference in Arabic, ICCA'11.*
- ANDERSON, P. (1999). Perspective : Complexity Theory and Organization Science. *Organization Science*, 10(3):216–232.
- ASHBY, W. R. (1956). *An Introduction to Cybernetics*. New York, J. Wiley,.
- BALMISSE, G. (2002). Les réseaux de neurones. http://www.gillesbalmisse.com/IMG/pdf/GB_RN.pdf.
- BEIGY, H. et MEYBODI, M. R. (2003). A Self-Organizing Channel Assignment Algorithm : A Cellular Learning Automata Approach. *In Proceedings of the 4th International Conference on Intelligent Data Engineering and Automated Learning, IDEAL'03*, pages 119–126.
- BEIGY, H. et MEYBODI, M. R. (2004). A Mathematical Framework for Cellular Learning Automata. *Advances in Complex Systems*, 7(3-4):295–319.
- BEIGY, H. et MEYBODI, M. R. (2005). An Adaptive Call Admission Algorithm for Cellular Networks. *Comput. Electr. Eng.*, 31:132–151.

- BEIGY, H. et MEYBODI, M. R. (2008). Brief Paper : Asynchronous Cellular Learning Automata. *Automatica*, 44:1350–1357.
- BELLAC, M. (2003). Introduction a l'information quantique. Cours donné à l'Ecole Supérieure de Sciences Informatiques (ESSI).
- BENACHBA, K. (2005). *Modèle d'exécution pour l'aide à la résolution du problème MAX-SAT*. Thèse de doctorat, L'institut national d'informatique, Alger.
- BENBOUZID SITAYEB, F. (2005). *Contribution à l'étude de la performance et de la robustesse des ordonnancements conjoints production/maintenance – Cas du flowshop*. Thèse de doctorat, Université de Franche-Comté.
- BERGER, N. (2010). *Modélisation et résolution en programmation par contraintes de problèmes mixtes continu/discret de satisfaction de contraintes et d'optimisation*. Thèse de doctorat, Université de Nantes,.
- BEYER, H. G. et SCHWEFEL, H. P. (2002). Evolution Strategies, A Comprehensive Introduction. 1:3–52.
- BONABEAU, E., DORIGO, M. et THERAULAZ, G. (1998). *From Natural to Artificial Swarm Intelligence*. Oxford University Press. ISBN :0195131584.
- BOUFENAR, C., BENGUERRAICHE, M., TALBI, H., DRAA, A. et BATOUCHE, M. (2005). Une hybridation entre l'informatique quantique et la vie artificielle pour résoudre le problème du voyageur de commerce. *In Proceedings of : Conférence Internationale sur la Productique, CIP'05*. Tlemcen.
- CHANGEUX, J. P. (2009). Réseaux de neurones. http://82.245.102.21/html/reseau_de_neurones.pdf.
- CLERGUE, G. (1997). *L'apprentissage de la complexité*. Hermès-Lavoisier.
- COBB, J. (2011). A Definition of Learning. <http://www.missiontolearn.com/2009/05/definition-of-learning/>.
- CORMAS (2010). Les systèmes multi-agents, présentation générale. <http://cormas.cirad.fr/fr/demarch/sma.htm>.
- CORNUÉJOLS, A. et MICLET, L. (2002). *L'apprentissage artificiel. Concepts et algorithmes*. Eyrolles.
- COUTURE, M. et VALCARTIER, D. (2007). Complexity and Chaos-State-of-the-Art ; Overview of Theoretical Concepts. Rapport technique, Ministry of National Defence, Canada.

- DA CRUZ, A. V., VELLASCO, M. M. B. R. et PACHECO, M. A. (2007). Quantum-Inspired Evolutionary Algorithm for Numerical Optimization. *In* ABRAHAM, A., GROSAN, C. et ISHIBUCHI, H., éditeurs : *Hybrid Evolutionary Algorithms*, volume 75, chapitre 2, pages 19–37. Springer Berlin Heidelberg.
- DA CRUZ, A. V. A., PACHECO, M. A. I. C., VELLASCO, M. B. R. et BARBOSA, C. R. H. (2005). Cultural Operators for a Quantum-Inspired Evolutionary Algorithm Applied to Numerical Optimization Problems. *In* *IWINAC (2)*, pages 1–10.
- DAS, S., ABRAHAM, A., CHAKRABORTY, U. K. et KONAR, A. (2009). Differential Evolution Using a Neighborhood-based Mutation Operator. *Trans. Evol. Comp*, 13:526–553. <http://www.softcomputing.net/tevc2.pdf>.
- DAS, S. et SUGANTHAN, P. N. (2011). Differential Evolution : A Survey of the State-of-the-Art. *IEEE Trans. Evolutionary Computation*, 15(1):4–31.
- DAVISON, A. (2005). *Killer Game Programming in Java*. O’Reilly Media, Inc.
- DE ROMEMON, C. (2010). Quels sont les changements fondamentaux qu’a amené la systématique dans l’architecture? Mémoire de D.E.A., ENSA Paris Val de Seine. <http://architecture-algorithmique.deromemont.com/files/memoire.pdf>.
- DEUTSCH, D. et EKERT, A. (2006). Quantum Computation. *Physics World*, 11(3):47–52.
- DODDER, R. et DARE, R. (2000). Complex Adaptive Systems and Complexity Theory : Inter-related Knowledge Domains. *In* *Proceedings of Research Seminar in Engineering Systems*. Massachusetts Institute of Technology, <http://web.mit.edu/esd.83/www/notebook/ComplexityKD.PDF>.
- DOOLEY, K. (1996). A Nominal Definition of Complex Adaptive Systems. *The Chaos Network*, 8(2).
- DRAA, A. (2004). Une nouvelle approche pour le recalage des images multimodales, basée sur l’informatique quantique et les algorithmes évolutionnaires. Mémoire de Magistère, Université Mentouri, Constantine.
- DRAA, A. et MESHOU, S. (2010). A Quantum Inspired Learning Cellular Automaton for Solving the Travelling Salesman Problem. *In* *Proceedings of the 2010 12th International Conference on Computer Modelling and Simulation, UKSIM’10*, pages 45–50. IEEE Computer Society.
- DRAA, A., MESHOU, S. et CHIKHI, S. (2011). A Fertile Quantum-Inspired Differential Evolution Algorithm for Combinatorial Optimisation. *In* *Proceedings of the 7th International Computing Conference in Arabic, ICCA’11*, pages 993–1010.

-
- DRAA, A., MESHOUL, S., TALBI, H. et BATOUCHE, M. (2010). A Quantum-Inspired Differential Evolution Algorithm for Solving the N-Queens Problem. *Int. Arab J. Inf. Technol.*, 7(1):21–27. <http://www.ccis2k.org/iajit/PDF/vol.7,no.1/4.pdf>.
- DUTECH, A. (2010). Méta-heuristiques pour l’optimisation : Differential Evolution et Particle Swarm Optimisation. Séminaire MAIA, LORIA, Mai 2010. http://www.loria.fr/~dutech/Papier/pres_optim_maia_100503.pdf.
- EBERHART, R. C. et SHI, Y. (2007). Computational Intelligence : Concepts to Implementations.
- ECOFR (2011). Les principes de l’adaptation. <http://www.developpement-durable.gouv.fr/Les-principes-de-l-adaptation,14478.html>.
- EDMONDS, B. (2000). *Syntactic Measures of Complexity*. Thèse de doctorat, University of Manchester.
- ELIE-DIT-COSAQUE, C. (2009). *Studies on Adaptation to Information Systems : Multiple Roles and Coping Strategies*. Thèse de doctorat, Georgia State University.
- ENAMI ERAGHI, A., AKBARI TORKESTANI, J. et MEYBODI, M. R. (2011). Solving the Bandwidth Multicoloring Problem : A Cellular Learning Automata Approach. *International Proceedings of Computer Science and Information Technology*, 3:163–168. IACSIT Press, Singapore.
- ERNEHOLM, C.-O. (2011). Simulation of the Flocking Behavior of Birds with the Boids Algorithm. Bachelor of Science Thesis, KTH Computer Science and Communication, Stockholm, Sweden.
- ESNAASHARI, M. et MEYBODI, M. R. (2008). A Cellular Learning Automata Based Clustering Algorithm for Wireless Sensor Networks. *Sensor Letters*, 6(5):723–735.
- FEI, Q. et HIRONORI, H. (1996). A Parallel Learning Cellular Automata for Combinatorial Optimization Problems. *In Proceedings of the International Conference on Evolutionary Computation*, pages 553–558.
- FEI, Q., ZHAO, Y. et HIRONORI, H. (2001). Learning Cellular Automata for Function Optimization Problems. *T. IEE Japan*, 121-c(261-268):189–195.
- FILLIAT, D. (2004). Robotique mobile. Support de cours, ENSTA, <http://www.dfr.ensta.fr/Cours/docs/C10-2/PolyRobotiqueMobileENSTA.pdf>.
- FOGEL, D. B. (1997). Evolutionary Computation : A New Transactions. *IEEE Trans. Evolutionary Computation*, 1(1):1–2.
- FOGEL, L. J., ANGELINE, P. J. et FOGEL, D. B. (1995). An Evolutionary Programming Approach to Self-Adaptation on Finite State Machines. *In* MCDONNELL, J. R.,
-

-
- REYNOLDS, R. G. et FOGEL, D. B., éditeurs : *Proc. of the 4th Annual Conference on Evolutionary Programming*, pages 355–365. MIT Press.
- FOGEL, L. J., OWENS, A. J. et WALSH, M. J. (1966). *Artificial Intelligence Through Simulated Evolution*. John Wiley, New York, USA.
- FRASER, A. S. (1957). Simulation of Genetic Systems by Automatic Digital Computers. *Australian Journal of Biological Sciences*, 10:484–491.
- FROMM, J. (2004). *The Emergence of Complexity*. Kassel University Press. ISBN :3-89958-069-9.
- FROMM, J. (2005). Types and Forms of Emergence. <http://arxiv.org/ftp/nlin/papers/0506/0506028.pdf>.
- FROMM, J. (2006). On Engineering and Emergence. *Electrical Engineering*. <http://arxiv.org/abs/nlin/0601002>.
- GAGNÉ, C. (2010). Apprentissage par renforcement : Introduction. Notes de cours, Université Laval, <http://wcours.gel.ulaval.ca/2010/a/GIF4101/default/5notes/ar-sem15-renforcement.pdf>.
- GANGULY, N., SIKDAR, B. K., DEUTSCH, A., CANRIGHT, G. et CHAUDHURI, P. P. (2003). A Survey on Cellular Automata. Rapport technique, Centre for High Performance Computing, Dresden University of Technology. <http://www.cs.unibo.it/bison/publications/CAsurvey.pdf>.
- GARCIA, F. (2011). Apprentissage par renforcement : Introduction. Unité de biométrie et d’Intelligence Artificielle, INRA, <http://www.inra.fr/internet/Departements/MIA/T/garcia/Doc/Talks/talkPDM.pdf>.
- GEORGE, J. P. (2003). L’émergence. Rapport technique, IRIT.
- GHALI, K. (2005). *Méthodologie de conception système à base de plateformes reconfigurables et programmables*. Thèse de doctorat, Université Paris XI.
- GHARAJEDAGHI, J. (1999). *Systems Thinking : Managing Chaos and Complexity : A Platform for Designing Business Architecture*. Butterworth-Heinemann.
- GLEIZES, M. P. (2004). *Vers la résolution de problèmes par émergence*. Habilitation à diriger des recherches, Université Paul Sabatier.
- GOLDBERG, D. et CORRUBLE, V. (1994). *Algorithmes génétiques : exploration, optimisation et apprentissage automatique*. Vie artificielle. Addison-Wesley.
- GOLDBERG, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st édition. ISBN :0201157675.
-

-
- GOLDSTEIN, J. (1999). Emergence as a Construct : History and Issues. *Emergence*, 1(1):49–72. http://www.informaworld.com/openurl?genre=article&doi=10.1207/s15327000em0101_4&magic=crossref.
- GRISOGONO, A. M. (2010). Complex Adaptive Systems Concepts for Complex Engineering, JSA Action Group 14 Complex Adaptive Systems for Defence. Rapport technique, The Technical Cooperation Program.
- GROVER, L. K. (1996). A Fast Quantum Mechanical Algorithm for Database Search. *In Proceedings of the twenty-eighth annual ACM symposium on Theory of computing, STOC'96*, pages 212–219. http://arxiv.org/PS_cache/quant-ph/pdf/9605/9605043v3.pdf.
- HAN, K. H. (2003). *Quantum-Inspired Evolutionary Algorithm*. Thèse de doctorat, Korea Adv. Inst. Sci. Technol. (KAIST).
- HAN, K. H. et KIM, J. H. (2000). Genetic Quantum Algorithm and Its Application to Combinatorial Optimization Problem. *In Proceedings of the 2000 IEEE Congress on Evolutionary Computation*, pages 1354–1360. .
- HAN, K. H. et KIM, J. H. (2002). Quantum-inspired Evolutionary Algorithm for a Class of Combinatorial Optimization. *IEEE Trans. Evolutionary Computation*, 6:580–593.
- HAN, K. H. et KIM, J. H. (2004). Quantum-inspired evolutionary algorithms with a new termination criterion, Heps gate, and two-phase scheme. *IEEE Trans. Evolutionary Computation*, 8(2):580–593.
- HARIRI, A., RASTEGAR, R., NAVI, K., ZAMANI, M. S. et MEYBODI, M. R. (2005). Cellular Learning Automata based Evolutionary Computing (CLA-EC) for Intrinsic Hardware Evolution. *In Proceedings of Evolvable Hardware'05*, pages 294–297.
- HARTVIGSEN, G., KINZIG, A. et PETERSON, G. (1998). Use and Analysis of Complex Adaptive Systems in Ecosystem Science : Overview of Special Section. *Ecosystems*, 1:427–430.
- HASSAS, S. (2003). *Systèmes complexes à base de multi-agents situés*. Habilitation à diriger des recherches, Université Claude Bernard-Lyon 1.
- HAYKIN, S. (1994). *Neural Networks : A Comprehensive Foundation*. Macmillan, New York.
- HENON, S. et WEISBUCH, G. (2003). Action Concertée : "Systèmes complexes en SHS". Rapport technique, CNRS, PARIS.
- HEY, T. (1999). Quantum Computing : An Introduction. *Computing and Control Engineering Journal*, 10(3):105–112.
-

- HIGGS, D. (2001). APEC as a Complex Adaptive System : Insights on the Problem of Multilateralism Versus Bilateralism from a New Science. Rapport technique, Australian APEC Study Centre.
- HOLLAND, J. H. (1962). Outline for Logical Theory of Adaptative Systems. *Journal of the association of Computing Machinery*, 3:484–491.
- HOLLAND, J. H. (1975). Adaptation in Natural and Artificial Systems.
- HOLLAND, J. H. (1996). *Hidden Order : How Adaptation Builds Complexity*. Addison Wesley Longman Publishing Co., Inc., New York.
- HUANG, Y., TANG, C. et WANG, S. (2008). Quantum-Inspired Swarm Evolution Algorithm. *Journal of Communication and Computer*, 5(5):208–211.
- IANIK, M. (1995). *Les implications épistémologiques des théories du chaos pour l'hypothèse des anticipations rationnelles*. Thèse de doctorat, Université du Québec.
- ILACHINSKI, A. (2004). *Artificial War : Multi-Agent Based Simulation of Combat*. World Scientific, Singapore.
- JUARRERO, A. (2010). Complex Dynamical Systems Theory. *Cognitive Edge Reference Paper*, pages 1–11. <http://cognitive-edge.com/articledetails.php?articleid=62>.
- KEMPE, J., LAPLANTE, S. et MAGNIEZ, F. (2006). Comment calculer quantique ? *La Recherche*, 398:30–37. <http://www.liafa.univ-paris-diderot.fr/~magniez/PAPIERS/LaRecherche.pdf>.
- KIM, J. (2000). L'émergence, les modèles de réduction et le mental. *Philosophiques*, 27(1):11–26.
- KOZA, J. (1996). *Genetic Programming, On the Programming of Computers by Means of Natural Selection*. A Bradford Book. MIT Press.
- LABURTHER, F. (1998). *Contraintes et algorithmes en optimisation combinatoire*. Thèse de doctorat, PARIS VII.
- LANGTON, C. G. (1986). Studying Artificial Life with Cellular Automata. *Phys. D*, 2:120–149. <http://portal.acm.org/citation.cfm?id=25201.25210>.
- LANGTON, C. G. (1990). Computation at the Edge of Chaos : Phase Transition and Emergent Computation. *Physica*, 42:12–37.
- LAYEB, A., MESHOUL, S. et BATOUCHE, M. (2008). Quantum Genetic Algorithm for Multiple RNA Structural Alignment. *In Asia International Conference on Modelling and Simulation*, pages 873–878.

- LAYEB, A. et SAIDOUNI, D.-E. (2007). Quantum Genetic Algorithm for Binary Decision Diagram Ordering Problem. *International Journal of Computer Science and Network Security*, 7(9):130–135.
- LE MOIGNE, J. L. (1999). *La Modélisation des systèmes complexes*. Dunod, Paris.
- LEARNTHEO (2011). Learning Theory. <http://www.npac.syr.edu/projects/cpsedu/summer98summary/examples/hpf/hpf.html>.
- LIU, J. et LAMPINEN, J. (2005). A Fuzzy Adaptive Differential Evolution Algorithm. *Soft Comput.*, 9:448–462.
- MAGNIEZ, F. (2002). Introduction au calcul quantique. Rapport technique. Centre national de la recherche scientifique (CNRS).
- MANSON, S. M. (2001). Simplifying Complexity : a Review of Complexity Theory. *Geoforum*, 32(3):405–414.
- MARTINEZ, Y., VAN VRECKEM, B., CATTEEUW, D. et NOWE, A. (2010). Application of Learning Automata for Stochastic Online Scheduling. *Recent Advances in Optimization and its Applications in Engineering*, pages 491–498. ISBN :978-3-642-12598-0.
- MATURANA, H. et VARELA, F. (1992). *The Tree of Knowledge*. Shambhala, Boston, MA.
- MCLAUGHLIN, B. P. et VARELA, F. (1980). *Autopoiesis and Cognition : The Realization of Living*. D. Reidel Publishing Company Holland, London.
- MEGHRICHE, S. (2008). *Reconnaissance de forme de signaux biologiques*. Thèse de doctorat, Université de Batna, Batna.
- MEYBODI, M. R. et BEIGY, H. (2002). A Note on Learning Automata-Based Schemes for Adaptation of BP Parameters. *Neurocomputing*, pages 957–974.
- MEYBODI, M. R., BEIGY, H. et TAHERKHANI, M. (2000). Application of Cellular Learning Automata to Image Processing. *In Proceedings of the First Conference in Mathematics and Communication*, pages 23.1–23.10, Tehran, Iran.
- MEYBODI, M. R. et KHOJASTEH, M. R. (2001). Application of Cellular Learning Automata in Modelling of Commerce Networks. *In Proceedings of the 6th Annual International Computer Society of Iran Computer Conference, CSICC-2001*, pages 284–295, Isfahan, Iran.
- MEYBODI, M. R. et MEHDIPOUR, F. (2003). VLSI Placement Using Cellular Learning Automata. *In 8th Annual International Computer Society of Iran Computer Conference, CSICC-200*, pages 195–203, Mashad, Iran.
- MEYBODI, M. R. et TAHERKHANI, M. (2001). Application of Cellular Learning Automata in Modeling of Rumor Diffusion. *In Proceedings of the 9th Conference on Electrical Engineering*, pages 102–110, Tehran, Iran.

- MEZURA-MENTES, E. et PALMEQUE-ORTIZ, A. G. (2009). Self-adaptive and Deterministic Parameter Control in Differential Evolution for Constrained Optimization. *Optimization*, pages 95–120.
- MILGRAM, M. (1993). *Reconnaissance des formes : Méthodes numériques et connexionnistes*. Armand Collin, Paris. ISBN :9780314933911.
- MITCHELL, M. (1998). *An Introduction to Genetic Algorithms*. MIT Press, Cambridge, MA, USA. ISBN :0262631857.
- MOJARADI, B., LUCAS, C. et VARSHOSAZ, M. (2004). Using Learning Cellular Automata for Post Classification Satellite Imagery. *International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences*, 35(4):991–995. <http://www.isprs.org/proceedings/XXXV/congress/comm4/papers/487.pdf>.
- MONTALVO, V. (2005). Modélisation de la couche sociale d’humanoïdes virtuels. Rapport technique, Rennes. <ftp://ftp.irisa.fr/local/caps/DEPOTS/BIBLIO2005/Montalvo.pdf>.
- MORIN, E. (2008). *On Complexity*. Advances in Systems Theory, Complexity, and the Human Sciences. Hampton Press.
- MOUTAIROU, M. M. (2009). *Application de réseaux Mesh dans un milieu confiné*. Thèse de doctorat, Université Laval, Québec.
- NAMATAME, A. et SASAKI, T. (1998). Self-organization of Complex Adaptive Systems as a Society of Rational Agents. *Artificial Life Robotics*, 2(4):189–195.
- NARAYANAN, A. et MOORE, M. (1996). Quantum-Inspired Genetic Algorithms. *In Proceedings of the 1996 IEE International Conference on Evolutionary Computation, ICEC’96*, pages 41–46.
- NECSI (2011). New England Complex Systems Institute. <http://necsi.edu/>.
- NEDJAH, N., COELHO, L. et MOURELLE, L. (2008). *Quantum Inspired Intelligent Systems*. Studies in Computational Intelligence. Springer.
- NEUMANN, J. V. (1966). *Theory of Self-Reproducing Automata*. University of Illinois Press, Champaign, IL, USA.
- NIELSEN, M. A. et CHUANG, I. L. (2000). *Quantum Computation and Quantum Information*. Cambridge University Press.
- NOMAN, N., BOLLEGALA, D. et IBA, H. (2011). An Adaptive Differential Evolution Algorithm. *In 2011 IEEE Congress on Evolutionary Computation, CEC’11*, pages 2229–2236. http://www.iba.t.u-tokyo.ac.jp/~danushka/papers/Noman_CEC_2011.pdf.
- O’BRIEN, K. et HOLLAND, T. D. (1992). The Role of Adaptation in Archeological Explanation. *American Antiquity*, 57:36–69.

- OOMMEN, B. J. et de ST. CROIX, E. V. (1996). Graph Partitioning Using Learning Automata. *IEEE Trans. Comput.*, 45:195–208.
- OOMMEN, B. J. et ROBERTS, T. D. (2000). Continuous Learning Automata Solutions to the Capacity Assignment Problem. *IEEE Trans. Comput.*, 49:608–620.
- PAPADIMITRIOU, G. I., SKLIRA, M. et POMPORTSIS, A. S. (2004). A New Class of ϵ -Optimal Learning Automata. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 34(1):246–254. <http://doi.ieeecomputersociety.org/10.1109/TSMCB.2003.811117>.
- PARROTT, L. (2002). Complexity and the Limits of Ecological Engineering. *Transactions of the ASAE*, 45(5):1697–1702.
- POTGIETER, A. (2004). *The Engineering of Emergence of Complex Adaptive Systems*. Thèse de doctorat, Pretoria. <http://upetd.up.ac.za/thesis/available/etd-09222004-091805/unrestricted/01thesis.pdf>.
- PRESKILL, J. (1998). Lecture Notes for Physics 229 : Quantum Information and Computation. California Institute of Technology, http://lorentz.leidenuniv.nl/quantumcomputers/literature/preskill_1_to_6.pdf.
- PRIGOGINE, I. et STENGERS, I. (1984). *Order Out of Chaos*. Bantam, New York.
- PURCINA, L. A. et SARAMAGO PEREIRA, S. F. (2008). Differential Evolution Applied to the Solution of Large Linear Systems. *In International Conference on Engineering Optimization*, pages 146–156. http://www.engopt.org/nukleo/pdfs/0146_saramago.pdf.
- RASTEGAR, R., MEYBODI, M. R. et BADIE, K. (2004). A New Discrete Binary Particle Swarm Optimization Based on Learning Automata. *In Proceedings of 2004 International Conference on Machine Learning and Applications, ICMLA'04*, pages 456–462.
- RECHENBERG, I. (1965). Cybernetic solution path of an experimental problem. *In Royal Aircraft Establishment Translation No. 1122, B. F. Toms, Trans.* Ministry of Aviation, Royal Aircraft Establishment, Farnborough Hants.
- REUS, B. (2010). Bas Reus' Quest on Self-organization and Online Collaborative Spaces. <http://basreus.nl/tag/complex-adaptive-systems/>.
- REYNOLDS, C. W. (1987). Flocks, Herds and Schools : A Distributed Behavioral Model. *SIGGRAPH Comput. Graph.*, 21:25–34. <http://www.naturewizard.com/papers/flocks%20-%20p25-reynolds.pdf>.
- REZVANIAN, A. et MEYBODI, M. R. (2010). LACAIS : Learning Automata Based Cooperative Artificial Immune System for Function Optimization. *In Proceedings of the 3rd International Conference on Contemporary Computing, IC3 (1)*, pages 64–75.

- RIEFFEL, E. G. et POLAK, W. (2000). An Introduction to Quantum Computing for Non-Physicists. *ACM Comput. Surveys*, 32(3):300–335. <http://quantum3000.narod.ru/papers/edu/qc-intro.pdf>.
- ROBIN, M. (2003). A Note on Complex Adaptive Systems. <https://sites.google.com/a/tcib.org.uk/the-centre-for-international-business/research-papers-1/a-note-on-complex-adaptive-system>.
- ROSNET, E. (1999). *L'adaptation psychologique au stress dans les situations extremes*. Habilitation à diriger des recherches, Université de Reims Champagne Ardenne.
- SANDFORD, R. (2007). Modélisation de réseaux de neurones quantiques. Un modèle pour le fonctionnement du cerveau? Laboratoire de sciences cognitives, Université Victor Segalen Bordeaux 2.
- SEEL, R. (2000). Culture and Complexity : New Insights on Organisational Change. *HEA*, 7(2):2–9.
- SETH, L. (1999). MIT News. *In Complexity in Engineering Conference*. <http://web.mit.edu/newsoffice/1999/complexity.html>.
- SHOR, P. W. (1994). Algorithms for Quantum Computation : Discrete Logarithms and Factoring. *In Proceedings of the 35th Annual Symposium on Foundations of Computer Science*, pages 124–134, Washington, DC, USA. IEEE Computer Society.
- SMIT, B. et WANDEL, J. (2006). Adaptation, Adaptive Capacity and Vulnerability. *Global Environmental Change*, 16(3):282–292.
- STORN, R. et PRICE, K. (1997). Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *J. of Global Optimization*, 11:341–359.
- TALBI, H. (2009). *Algorithmes évolutionnaires quantiques pour le recalage et la segmentation multiobjectif d'images*. Thèse de doctorat, Université Mentouri, Constantine.
- TALBI, H., BATOUCHE, M. et DRAA, A. (2004a). A Quantum-Inspired Genetic Algorithm for Multi-source Affine Image Registration. *In CAMPILHO, A. et KAMEL, M., éditeurs : Image Analysis and Recognition*, volume 3211 de *Lecture Notes in Computer Science*, pages 147–154. Springer Berlin / Heidelberg.
- TALBI, H., BATOUCHE, M. et DRAA, A. (2007). A Quantum-Inspired Evolutionary Algorithm for Multiobjective Image Segmentation. *International Journal of Mathematical, Physical and Engineering Sciences*, 1(2):109–114.
- TALBI, H., DRAA, A. et BATOUCHE, M. (2004b). A New Quantum-Inspired Genetic Algorithm for Solving the Travelling Salesman Problem. *In Proceedings of the 2004 IEEE International Conference on Industrial Technology, IICIT'04*, pages 1192–1197.

- TALBI, H., DRAA, A. et BATOCHE, M. (2006). A Novel Quantum-Inspired Evaluation Algorithm for Multi-Source Affine Image Registration. *Int. Arab J. Inf. Technol.*, 3(1):9–15.
- TAZI, A. et PABLO, A. M. (2008). Estimation d’un arbre de recherche. Rapport technique, Université des Sciences Montpellier 2. <http://www.lirmm.fr/~leclere/enseignements/TER/2008/Rapport/8.pdf>.
- TSPLIB (1995). TSPLIB. Site de référence du problème du voyageur de commerce. <http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/>.
- TURING, A. M. (1950). Computing Machinery and Intelligence. *Mind*, LIX:433–460.
- UNEWISSE, M. et GRISOGONO, A.-M. (2006). Adaptivity Led Networked Force Capability. *In Proceedings of the 12th ICCRTS - Adapting C2 to the 21st Century*. http://www.dodccrp.org/events/12th_ICCRTS/CD/html/papers/200.pdf.
- UNSAI, C. (1998). *Intelligent Navigation of Autonomous Vehicles in an Automated Highway System : Learning Methods and Interacting Vehicles Approach*. Thèse de doctorat, Electrical and Computer Engineering, Virginia Tech.
- VENTURA, D. et MARTINEZ, T. (1997). An Artificial Neuron with Quantum Mechanical Properties. *In Proceedings of the International Conference on Artificial Neural Networks and Genetic Algorithms*, pages 482–485.
- WEISS, G., éditeur (1999). *Multiagent systems : A Modern Approach to Distributed Artificial Intelligence*. MIT Press, Cambridge, MA, USA. ISBN :0-262-23203-0.
- WHITLEY, D. (1994). A Genetic Algorithm Tutorial. *Statistics and Computing*, 4:65–85.
- WILLIAMS, G. P. (2001). *Chaos Theory Tamed*. National Academy Press (Trade). ISBN :0309063515.
- WOLFRAM, S. (1983). Cellular Automata. Los Alamos Science, <http://library.lanl.gov/cgi-bin/getfile?09-01.pdf>.
- WOLFRAM, S. (2002). *A New Kind of Science*. Wolfram Media, 1 édition.
- YANG, Z., TANG, K. et YAO, X. (2007). Differential Evolution for High-Dimensional Function Optimization. *In IEEE Congress on Evolutionary Computation’07*, pages 3523–3530.
- YOON, B., HOLMES, D. J., LANGHOLZ, G. et KANDEL, A. (1994). Efficient Genetic Algorithms for Training Layered Feedback Neural Networks. *Inf. Sci.*, pages 67–85.
- ZHANG, G., JIN, W. et LI, N. (2003). An Improved Quantum Genetic Algorithm and its Application. *In Proceedings of the 9th International Conference on Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing, RSFDGrC’03*, pages 449–452, Berlin, Heidelberg. Springer-Verlag.