Ministère de l'Enseignement Supérieur et de la
Recherche Scientifique

UNIVERSITE EL HADJ LAKHDAR BATNA

Faculté des Sciences de l'Ingénieur
Département de Génie Industriel

MEMOIRE

Présenté au
Laboratoire d'Automatique et Productique

En vue de l'obtention du diplôme de
Magister
Spécialité: *Génie Industriel*
Option: *Génie Industriel*

Par

Naima ZERARI
Ingénieur en Informatique
Université de Batna

LES ALGORITHMES GENETIQUES EN MAINTENANCE

Directrice du Mémoire: **Dr. L H. MOUSS**

Soutenu devant le Jury composé de:

M. BATOUCHE	*Président*	Prof. Université de Constantine
L. H. MOUSS	*Rapporteur*	M.C. Université de Batna
H. SMADI	*Examinateur*	M.C. Université de Batna
K. N. MOUSS	*Examinatrice*	M.C. Université de Batna
B. ABDELHADI	*Examinateur*	M.C. Université de Batna
F. ABDESSEMED	*Examinateur*	M.C. Université de Batna
F. DEKHINET	*Invité*	C.C. Université de Batna

Année : 2006

Remerciements

Le travail de recherche présenté a été mené au Laboratoire d'Automatique et de Productique (LAP) au sein de l'équipe Systèmes Sûrs et Performants, dirigé par Madame *L.H.Mouss*, Maître de conférences, qui m'a fait l'honneur de diriger ce travail qui n'aurait pu voir le jour sans le soutien de nombreuses personnes que je tiens à remercier.

La première personne, ma directrice de mémoire, Madame *L.H.Mouss*, Maître de conférences à l'Université de Batna. Ses conseils tout au long du travail m'ont permis d'acquérir une maturité suffisante pour continuer dans le chemin de la recherche et de l'enseignement. Je lui exprime ici ma profonde reconnaissance. J'ai été très touchée par la confiance qu'elle m'a témoignée tout au long de mon travail.

Je remercie chaleureusement Monsieur *B.Abdelhadi*, Maître de conférences à l'Université de Batna pour sa disponibilité, ses compétences, son caractère rigoureux, et ses qualités humaines qui ont permis la bonne conduite de ce travail. J'adresse également mes remerciements à Monsieur *S.Derradji*, Chargé de cours à l'Université de Batna, pour son aide précieuse, sa disponibilité et pour l'ensemble de ces qualités tant scientifiques qu'humaines dont il m'a fait généreusement profiter.

Je ne pourrai jamais remercier suffisamment Monsieur *F.Dekhinet* Chargé de cours à l'Université de Batna et Monsieur *D.Abdelaziz*, Maître assistant à l'Université de Batna pour leur soutien moral et scientifique efficace et constant.

Je suis particulièrement reconnaissante à Monsieur *H.Smadi*, Maître de conférences à l'Université de Batna, qui m'a poussé à un travail intensif.

Je remercie vivement Monsieur *M.Batouche*, Professeur à l'Université de Constantine qui me fait l'honneur de présider le jury.

Je voudrais témoigner ma gratitude et remercier tous ceux qui ont accepté avec bienveillance de participer au jury :
Je remercie alors *N.Mouss*, Maître de conférences à l'Université de Batna, Monsieur *B.Abdelhadi*, Maître de conférences à l'Université de Batna, Monsieur *F.Abdessemed*, Maître de conférences à l'Université de Batna et Monsieur *H.Smadi*, Maître de conférences à l'Université de Batna.

Je tiens à exprimer mes plus sincères remerciements à Mlle *M.Benbrahim*, Maître assistante à l'Université de Batna, qui m'a toujours accueillie dans son bureau et ne m'a pas privé de gentillesse, de conseils et avis pertinents. Mes remerciements vont également à Monsieur *W. Kaddouri*, Maître assistant à l'Université de Batna, Mlle *L. Abdou*, Maître assistante à l'Université de Biskra, Madame *N.Turqui*, Maître assistante à l'Université de Biskra et Monsieur *Y.Bahmani* pour la documentation qu'ils m'ont procuré ainsi que pour les remarques constructives qu'ils ont su me donner.

Je remercie également, l'ensemble du personnel des chercheurs du LAP et tous ceux que j'ai oublié de citer, et qui j'espère, ne m'en tiendront pas rigueur.

Table des Matières

Liste des figures

Liste des tableaux

Liste des algorithmes et des organigrammes

Introduction générale

L'environnement actuel des entreprises est caractérisé par des marchés soumis à une forte concurrence et sur lesquels les exigences et les attentes des clients deviennent de plus en plus forte en termes de qualité, de coût et de délais de mise à disposition. Dans un tel contexte, rester toujours performant, passe obligatoirement par le maintien en état de fonctionnement de l'outil de production, qui reste toujours la préoccupation majeure de tous les gestionnaires dans un monde industriel où les notions de réactivité, de coûts et de qualité ont de plus en plus d'importance, et où il est vital de pouvoir s'appuyer sur un système de production performant et sûr à tout instant.

Les entreprises sont donc de plus en plus sensibilisées à l'importance des coûts induits par les défaillances accidentelles des systèmes de production. Alors que la maintenance, jusqu'à très récemment, était considérée comme un centre de coûts, nous sommes de plus en plus conscients qu'elle peut contribuer d'une manière significative à la performance globale de l'entreprise. La complexité des mécanismes de dégradation des équipements a fait en sorte que la durée de vie de ces derniers a toujours été traitée comme une variable aléatoire. En outre, l'absence de données fiables et d'outils efficaces de traitement de ces données a réduit la fonction maintenance à des tâches de dépannage, et par le fait même, à une fonction dont les coûts ne cessent d'augmenter et dont la contribution à la performance de l'entreprise n'est pas évidente.

Ainsi, les coûts induits par l'arrêt de la production suite à des pannes de machines présentent un axe d'études mettant en évidence l'ampleur de l'effort nécessaire pour

mettre en place un système de gestion des stocks des pièces de rechange palliant dans ce domaine les problèmes des gestionnaires dans le monde industriel. Ces derniers se préoccupent d'une gestion des stocks de pièces de rechange efficace permettant d'éviter les ruptures des stocks et par conséquent de l'arrêt de la production qui entraîne des pertes considérables à l'entreprise. Il s'agit alors d'optimiser le stock des pièces de rechange, en faisant recours à une des méthodes d'optimisation.

Les méthodes d'optimisation se classent en deux catégories bien distinctes. D'une part, les méthodes exactes permettent à certains problèmes de trouver à coup sûr la (ou les) meilleure(s) solution(s) ou solution(s) optimale(s). Mais la plupart des problèmes étudiés en optimisation appartiennent à la classe des problèmes NP-Complets. Cette classe rassemble des problèmes pour lesquels il n'existe pas d'algorithme qui fournisse la solution optimale. D'autre part, les méthodes stochastiques qui se contentent de rechercher une solution " de bonne qualité ". A cet effet, plusieurs approches heuristiques ont été développées. Parmi lesquelles, on peut citer le recuit simulé, la recherche tabou, la programmation génétique, la stratégie d'évolution et les algorithmes génétiques.

Ces algorithmes constituent une famille d'algorithmes par recherche probabiliste, basés sur l'évolution des espèces formulée par Darwin. L'évolution d'une espèce est simplement une suite successive d'amélioration afin qu'elle soit la mieux adaptée au milieu dans lequel elle évolue. Cette adaptation est réalisée grâce à la sélection naturelle et aux mécanismes de la reproduction. Leurs champs d'application sont très vastes. Outre le célèbre problème du voyageur du commerce et les fonctions mathématiques, les algorithmes génétiques peuvent être utilisés pour l'optimisation des problèmes liés à la gestion des stocks gérée par toute entreprise.

Dans ce cadre, les gestionnaires dans le monde industriel s'intéressent à prévoir des investissements aussi réduit que possible tout en assurant une disponibilité des pièces de rechange dans le stock. Ainsi, la gestion des stocks qui est un des aspects opérationnels les plus importants des décisions de production, est aujourd'hui bien connue dans le domaine de la gestion de la production et qui a suscité amplement l'intérêt des

gestionnaires dans le monde industriel. Cet intérêt, nous a conduit à considérer le problème d'optimisation du coût d'exploitation des stocks comme thème de recherche, plus particulièrement, nous nous intéressons à déterminer le niveau du stock optimisant le coût d'exploitation du stock.

L'importance du problème énoncé, faisant partie de la classe des problèmes NP-complets, qui n'ont pas pour l'instant d'algorithme polynomial permettant de répondre à la question qui se pose: " Existe-t-il un stock d'une quantité inférieur à la quantité ' q ' dont le coût d'exploitation est inférieur au coût ' c' ?" nous a amené à proposer la méthode d'optimisation par algorithme génétique.

L'optimisation de la fonction objectif (appelée fitness), définie sur un espace de recherche dépendant du problème de la gestion des stocks, par l'algorithme génétique soumit une population d'individus à une suite de générations (la génération initiale est tirée au hasard dans l'espace de recherche). Une génération commence par la sélection de quelques individus bien adaptés pour la reproduction. Ces individus qui sont codés dans des génotypes composés de gènes correspondant aux valeurs des paramètres du problème de la gestion des stocks, engendrent une descendance en utilisant des opérateurs stochastiques appelés croisement et mutation. Le croisement agit sur plusieurs individus alors que la mutation agit sur un seul. Enfin, quelques-uns des descendants remplacent certains des parents pour terminer le processus de génération. Comme dans l'évolution naturelle, on espère l'émergence progressive d'individus de mieux en mieux adaptés : les meilleurs individus de la population finale (au regard de la fonction objectif) sont des approximations de solutions du problème d'optimisation considéré.

La méthodologie préconise pour résoudre la problématique traitée dans ce mémoire s'échelonne sur quatre chapitres.

Le premier chapitre propose une brève description de la vie artificielle, un des domaines émergent liés à l'intelligence artificielle. En outre, ce chapitre introduit les méthodes d'optimisation qui se divisent en classe déterministe et classe non déterministe. Ensuite, un état de l'art se propose en matière d'algorithmes évolutionnaires.

Dans le second chapitre, nous présentons les concepts de base des algorithmes génétiques qui sont au cœur de ce travail. Nous tentons d'expliquer ce qui en fait leur force tout en exposant l'intervention des différents opérateurs dans les différentes étapes de l'algorithme génétique.

Le troisième chapitre vise la validation de l'algorithme génétique implémenté. L'atteinte de cet objectif a suscité l'usage de quelques fonctions mathématiques dont les solutions analytiques sont connues. Ce chapitre décrit alors la résolution effectuée du problème d'optimisation des fonctions mathématiques par la méthode des algorithmes génétiques. Pour l'implémentation de l'AG, l'environnement Matlab a été utilisé sur la plate-forme Windows.

Les résultats d'application de l'algorithme génétique à chaque fonction mathématique sont détaillés afin d'en tirer des conclusions. Ces résultats ont permis d'envisager l'application de cet algorithme au problème préoccupant les gestionnaires dans le monde industriel, en occurrence la gestion des stocks, objet du chapitre quatre.

Le dernier chapitre passe en revue une définition de ce qui est la gestion des stocks et ses modèles, ensuite il met en exergue la formalisation du problème de la gestion des stocks sous forme de fonction objectif à manipuler par l'algorithme génétique. Il se termine par une représentation des différents résultats obtenus.

Nous terminerons avec une conclusion mettant l'accent sur l'intérêt qu'a apporté les algorithmes génétiques à la gestion des stocks, plus particulièrement à la détermination du coût optimal d'un stock de pièces de rechange.

Chapitre 1

Algorithmes évolutionnaires, principes et méthodes

Résumé : Ce premier chapitre introduit principalement la notion des algorithmes évolutionnaires d'une manière générale. Il s'attache à les positionner par rapport aux domaines de recherche connexe à l'Intelligence Artificielle et à la Vie Artificielle.

La première section est une introduction à la vie artificielle, illustrant ces principes et domaines. La seconde section, présente le processus d'optimisation ainsi que les méthodes les plus connues, classiques et évolutionnaires. La suite du chapitre est consacrée aux Algorithmes évolutionnaires : origines, principes et variantes. Ces algorithmes permettent de s'affranchir une grande partie de la méthode d'essai erreur et sont appelés à se répandre de plus en plus dans l'industrie grâce à la montée en puissance exponentielle du matériel informatique.

1.1 Introduction

Notre monde est intégralement constitué de systèmes vivants ou non-vivants, imbriqués et en interaction. Ainsi les ordinateurs et les machines intelligentes que nous connaissons aujourd'hui sont des applications de la cybernétique. Cette science cherche à expliquer, grâce à des concepts mathématiques les comportements de systèmes naturels ou artificiels et les échanges d'informations entre ces systèmes. La cybernétique a été fondée en 1948 par le mathématicien américain Norbert Wiener, qui projetait la fondation d'une nouvelle discipline scientifique autonome, toutefois ces recherches n'ont abouti qu'à une contribution au développement de divers domaines parmi lesquels on distingue l'intelligence artificielle.

1.2 Intelligence artificielle

L'Intelligence Artificielle (IA) appartient au domaine de l'informatique et concerne les approches informatiques ayant pour but de faire apparaître un comportement intelligent [Sav03]. C'est une science qui s'intéresse à la conception de machines pouvant simuler la cognition humaine. En effet, elle tend à donner à la machine un comportement "humain", capable notamment de s'adapter à des situations nouvelles et d'apprendre en permanence [Sav03]. L'IA est directement issue des concepts cognitivistes, c'est à dire qu'elle envisage le fonctionnement cérébral sous un angle logico-déductif, et considère, de fait que l'acte cognitif s'effectue à travers une manipulation de symboles élémentaires. A travers le cognitivisme, l'IA établit une analogie entre le fonctionnement cérébral et celui de l'ordinateur [Mam03]. Ce qui amène à dire que le but de l'IA est de construire des machines qui réalisent des choses qui requièrent de l'intelligence lorsqu'elles sont faites par des humains. Le terme d'IA est "né" officiellement durant la Dartmouth Conférence de l'été 1956 par Mc Carthy [Bou00b], mais était déjà dans les préoccupations des chercheurs depuis longtemps.

Tout au long de son histoire, deux pôles d'attraction antagonistes ont été constatés et nommés, en particulier par J.Searle, " IA forte " et " IA modérée " (plutôt que " faible ") [Mam03].

L'IA forte est à la base de la discipline : possibilité de créer certaines formes digitales d'intelligence artificielle (et donc non naturelle) qui peuvent véritablement penser, raisonner et éprouver une conscience d'elles-mêmes.

L'IA modérée est en quelque sorte plus raisonnable. Elle considère qu'il est possible de construire des systèmes, des algorithmes capables de raisonner et de résoudre des problèmes. Dans cette hypothèse de l'IA modérée, la machine va agir comme si elle était intelligente.

L'Intelligence Artificielle, cette discipline dont l'objectif classique est de concevoir des « agents rationnels » n'a cessé d'ouvrir ces voies de recherches ce qui a fait apparaître dans les années 80, trois sous domaines, illustrés par la Figure 1.1 [Dro01].

- L'Intelligence Artificielle Distribuée (IAD).
- Le connexionnisme.
- La Vie Artificielle (VA).

C'est sur le dernier sous domaine, *vie artificielle*, que nous mettons l'accent.

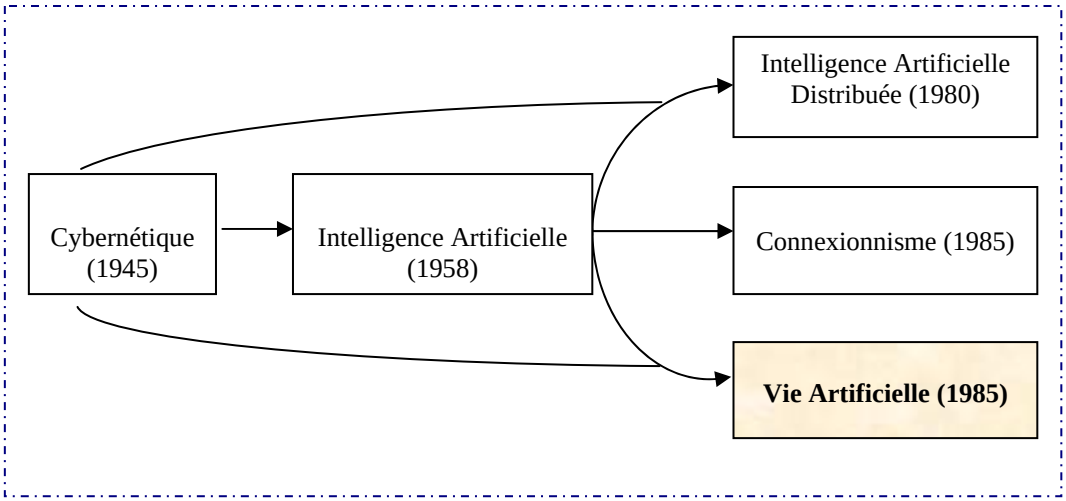


Figure 1. 1- Relation entre la cybernétique, IA et VA

1.3 Vie artificielle

Comme le fait présenter la Figure 1.1, la cybernétique a vu sa concrétisation technologique à travers deux disciplines. Il s'agit de l'intelligence artificielle et de la vie artificielle. Cette discipline plus jeune bénéficie par conséquent de l'expérience de l'intelligence artificielle [Mam03]. D'ailleurs le terme vie artificielle a fait officiellement son apparition en 1987 par le biais du premier séminaire " Artificial life" de Christopher Langton [Lat98] tenu à Santa Fe où il la définie comme étant : " L'étude

des systèmes construits par l'homme qui présentent des comportements caractéristiques des systèmes vivants naturels " [Dam01]. Cependant la définition de la vie artificielle est étroitement dépendante de la définition de la vie. Or cette dernière étant problématique, il est donc difficile de définir la vie artificielle et surtout de la limiter. C'est ainsi que sa définition fût calquée sur celle du vivant, présentée par J. Doyne Farmer et Alleta d'A. Belin, en l'exprimant sous forme d'une liste de propriétés minimales suivantes [Dam01] :

1. L'être humain a contribué au processus d'apparition de tout système de vie artificielle. Évidemment nous ne pouvant pas parler d'un produit artificiel que si nous le trouvons dans la nature.
2. Un système de vie artificielle est autonome.
3. Un système de vie artificielle est en interaction avec son environnement. Il en a une perception qui aura une influence sur les actions de ce système.
4. Il y a émergence de comportements dans un système de vie artificielle. Cela correspond à un comportement résultant d'une coordination spécifique et non prévue à l'intérieur du système d'un ensemble de fonctions basiques.

1.3.1 Principes de la vie artificielle

L'approche de la vie artificielle est essentiellement synthétique, c'est à dire de bas en haut. Son objectif est l'étude de propriétés émergentes au niveau global, en haut, uniquement à partir d'entités et de propriétés définies au niveau local, en bas. Ainsi, les observations sont portées sur des populations d'entités plutôt que sur des individus seuls.

La vie artificielle représente l'étude de l'ensemble des phénomènes liés à l'évolution des populations d'entités artificielles [Lat98].

1.3.2 Domaines de la vie artificielle

Lors de la troisième conférence de Santa Fe, Claus Emmeche évoque les différentes formes de vie artificielle. Il les a classé en deux groupes principaux : le groupe trivial et le groupe non trivial [Mam03].

1. **Le groupe trivial** comprend :
    - Les simulations informatiques fondées sur des modèles mathématiques, conceptuels ou physiques des systèmes biologiques. Elle ne peut aboutir à des formes réelles de vie.

- Les organismes modifiés. Il s'agit d'êtres vivants réels, modifiés par l'homme au moyen de manipulation génétique par exemple.

2. **Le groupe non trivial** correspond :
   - La robotique évolutive qui regroupe les robots autonomes et évolutifs.
   - Les expériences relevant de la biochimie.
   - L'exobiologie.
   - Les systèmes computationnels : monde virtuel, les automates cellulaires et les algorithmes évolutionnaires appartenant à la famille des algorithmes d'optimisation.

Une description de ces algorithmes fait l'objet de la section suivante.

1.4 Introduction aux différentes méthodes d'optimisation

Dans la vie courante, nous sommes fréquemment confrontés à des problèmes d'"optimisation" plus ou moins complexes. Cela peut commencer au moment où l'on tente de ranger son bureau, de placer son mobilier, et aller jusqu'à un processus industriel, par exemple pour la planification des différentes tâches. Ces problèmes peuvent être exprimés sous la forme générale d'un "problème d'optimisation". On définit alors une fonction objectif, que l'on cherche à optimiser (minimiser ou maximiser) par rapport à tous les "paramètres" concernés. Une telle fonction objectif présente généralement un grand nombre de solutions non optimales [Mag98]. Ce qui explique l'importance du problème d'optimisation d'une fonction et ce dans tous les domaines, notamment dans le domaine de la gestion de la production. Cette optimisation cherche à améliorer la robustesse des produits aux aléas (défaillance, erreurs humaines).

L'optimisation est donc une des branches les plus importantes des mathématiques appliquées modernes, et de nombreuses recherches à la fois pratiques et théoriques, lui sont consacrées [Mat02].

La théorie de l'optimisation comprend l'étude quantitative des optimums et les méthodes pour les trouver. L'espace d'état est définit par la modélisation mathématique des paramètres d'entrées du système suivant la nature des variables, cet espace peut être continus ou discrets.

Les méthodes d'optimisation permettent de déterminer plusieurs solutions dans l'espace d'état qui maximisent ou minimisent un critère [Oli98].

1.4.1 Processus d'optimisation

La Figure 1.2, présente le processus d'optimisation en trois étapes : analyse, synthèse et évaluation [Mag98]. Dans une première étape, il convient d'analyser le problème et d'opérer un certain nombre de choix préalables :

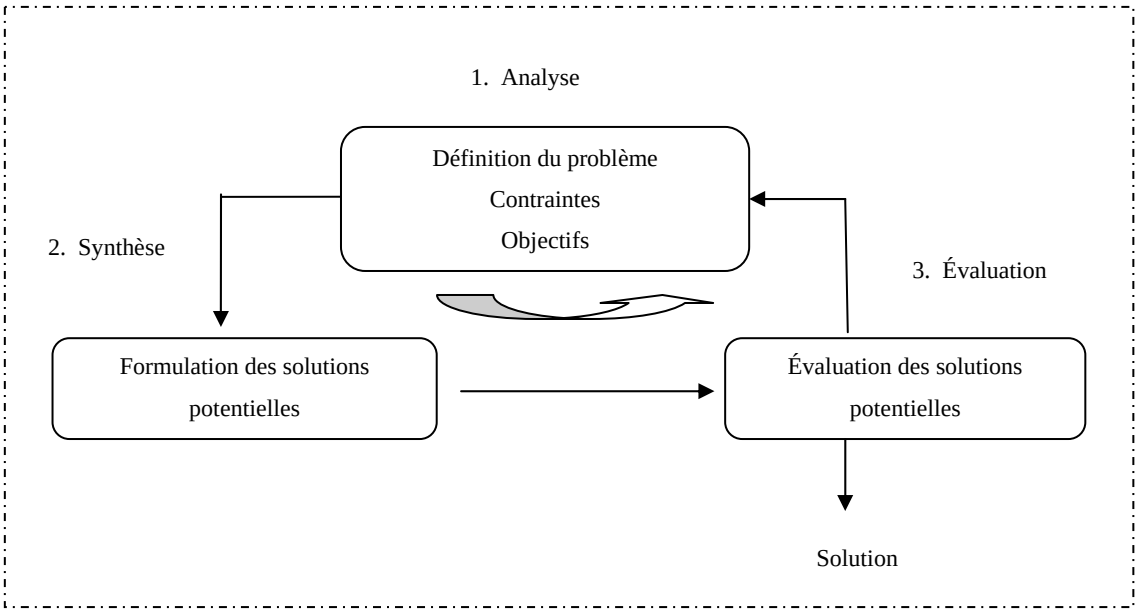


Figure 1. 2- Processus d'optimisation selon Asimow

- **Variables du problème.** Quels sont les paramètres intéressants à faire varier ?
- **Espace de recherche.** Dans quelles limites faire varier ces paramètres ?
- **Fonctions objectifs.** Quels sont les objectifs à atteindre ?
  Comment les exprimer mathématiquement ?
- **Méthode d'optimisation.** Quelle méthode choisir ?

Après la phase d'analyse, la méthode choisie synthétise des solutions potentielles qui sont évaluées, puis éventuellement éliminées jusqu'à obtention d'une solution acceptable. Si nécessaire, le problème peut être alors redéfini à partir des solutions déjà obtenues.

1.4.1.1 Variables du problème

C'est à l'utilisateur de définir les variables du problème. Il peut avoir intérêt à faire varier un grand nombre de paramètres pour augmenter les degrés de liberté de l'algorithme afin de découvrir des solutions nouvelles. Ou bien, s'il a une vue suffisamment précise de ce qu'il veut obtenir, il peut limiter le nombre de variables à l'essentiel [Mag98].

Les variables peuvent être de natures diverses. Par exemple, pour un composant électronique il peut s'agir de sa forme et de ses dimensions géométriques, des matériaux utilisés, et pour un composant en gestion de la production, il peut s'agir des coûts, de la qualité, délai de livraison, quantités économiques de production, stocks tampons entre les postes de travail. En outre, elles peuvent être réelles, complexes ou entières.

1.4.1.2 Espace de recherche

Dans certaines méthodes d'optimisation, tels que les stratégies d'évolution, l'espace de recherche est infini : seule la population initiale est confinée dans un espace fini. Mais dans le cas des algorithmes de type Monte Carlo et génétique, il est généralement nécessaire de définir un espace de recherche fini. Cette limitation de l'espace de recherche n'est généralement pas problématique. En effet, ne serait-ce que pour des raisons technologiques ou informatiques, les intervalles de définition des variables sont en général naturellement limitées. De plus, on a souvent une idée des ordres de grandeur des variables du problème.

1.4.1.3 Fonctions d'adaptation

La ou les grandeurs à optimiser peuvent être par exemple une consommation, un rendement, un facteur de transmission, etc. Un algorithme d'optimisation nécessite généralement la définition d'une fonction rendant compte de la pertinence des solutions potentielles, à partir des grandeurs à optimiser. Cette fonction est nommée fonction d'adaptation (fitness function). L'algorithme convergera vers un optimum de cette fonction, quelle que soit sa définition. Une fois cette fonction définie, il s'agit de choisir une méthode adaptée au problème posé [Mag98].

1.4.2 Méthodes d'optimisation

Cette section n'a pas pour objectif d'étudier comparativement les techniques d'optimisation. Néanmoins, il est important d'identifier les différentes méthodes.
La plupart des problèmes d'optimisation appartiennent à la classe des problèmes NP-difficiles et ne possèdent donc pas de solution algorithmique efficace valable pour toutes les données. Étant donnée l'importance de ces problèmes, de nombreuses méthodes de résolution ont été développées. Ces méthodes peuvent être classées sommairement en deux grandes catégories : les méthodes déterministes et les méthodes non-déterministes [Mag98]. Les méthodes déterministes sont généralement efficaces quand l'évaluation de la fonction est très rapide ou quand la forme de la fonction est connue à priori. Les cas plus complexes (temps de calcul important, nombreux optima locaux, …) sont souvent traités plus efficacement par des méthodes non-déterministes, appelées aussi méthodes stochastiques.

1.4.2.1 Méthodes déterministes

Ce sont des méthodes qui n'utilisent aucun concept stochastique, et exigent des hypothèses sur la fonction à optimiser, telles que la continuité et la dérivabilité en tout point du domaine des solutions. En général, l'utilisation de ces méthodes nécessite comme étape préliminaire la localisation des extrema. Celle-ci peut être faite, par exemple, par une discrétisation fine de l'espace de recherche. La fonction à optimiser est évaluée en chacun des points de discrétisation. La valeur maximale est alors considérée comme une bonne approximation de l'optimum de la fonction. Cette méthode est brutale et le temps de calcul augmentera exponentiellement en fonction du nombre de variables [Mag98].

1.4.2.2 Méthodes non déterministes

Lorsqu'une exploration de type déterministe est difficile à implanter, on fait appel aux techniques à recherche aléatoire. Ces méthodes font appel à des tirages de nombres aléatoires [Mag98]. Elles assurent qu'au bout d'un certain nombre d'itérations, les solutions fournies convergent vers la solution optimale. Ce type de méthodes permet d'explorer l'espace de recherche plus efficacement. Citons entre autres :

- **Monte Carlo** : Ces méthodes consistent en des simulations informatiques de problèmes mathématiques ou physiques, basées sur le tirage de nombres aléatoires. L'utilisation de la méthode de Monte-Carlo est devenue possible grâce à l'amélioration des performances des ordinateurs (il est devenu plus efficace de simuler numériquement le comportement d'un système complexe que de l'observer expérimentalement) car pour obtenir des estimations suffisamment exactes de la grandeur recherchée, il faut réaliser le calcul d'un très grand nombre de cas particuliers et dépouiller ensuite la statistique d'un volume énorme de données. Le grand avantage de cette méthode est sa simplicité. Or son inconvénient est le temps de calcul. Les méthodes Monte Carlo permettent une bonne exploration puisque tout point a une probabilité identique d'être atteint, toutefois, elles ne permettent pas d'exploitation des résultats déjà obtenus [Mag98].
- **Recuit simulé**: Cette classe de méthodes d'optimisation est due aux physiciens Kirkpatrick, Gelatt et Vecchi. Elle s'inspire des méthodes de simulation de Metropolis dans les années 50 en mécanique statistique [Fin02]. Le principe de cette méthode est calqué sur une manipulation sidérurgique. Cette manipulation a pour but, lors de la solidification de certains métaux, d'éviter que les molécules se retrouvent dans une configuration qui pourrait entraîner des faiblesses dans la structure du métal. Pour éviter ceci, le refroidissement est fait de plus en plus lentement au cours de la phase de solidification afin de laisser le temps aux molécules de trouver une position stable, c'est-à-dire celle qui minimise l'énergie. En effet, plus la température est élevée, plus les molécules en phase gazeuse ou liquide se déplacent vite (agitation moléculaire) [Reb99]. Ce procédé fût la source d'inspiration pour la méthode du recuit simulé.

  Il s'agit donc d'un algorithme itératif qui fait évoluer une solution courante. Metropolis et al. utilisent une méthode stochastique pour générer une suite d'états successifs du système en partant d'un état initial donné. Tout nouvel état est obtenu en faisant subir un déplacement aléatoire à un atome quelconque. Le nouvel état est soit accepté pour remplacer l'état courant, soit refusé pour que l'état courant soit maintenu. Après un grand nombre de perturbations, un tel processus fait évoluer le système vers un état d'équilibre. En pratique l'algorithme s'arrête et retourne la

meilleure configuration trouvée lorsque aucune configuration voisine n'a été acceptée pendant un certain nombre d'itérations [Reb99].

Le recuit simulé a comme principal inconvénient d'avoir des performances qui dépendent beaucoup du voisinage défini et du réglage de la fonction de température, notamment pour éviter que l'algorithme ne reste prisonnier d'un minimal local. Par contre, c'est un algorithme simple qui est facile à mettre en œuvre. Cependant, il n'est pas évident de choisir une fonction décroissante pour la température [Reb99].

- **Algorithmes évolutionnaires** : Basés sur une simulation d'évolution de populations de solutions. L'objectif des algorithmes évolutionnaires (AE) est de faire évoluer une population P dans le but de trouver l'optimum. Pour ce faire, à chaque génération t, les individus de la population sont mutés et croisés avec une probabilité et ce sont les plus aptes qui survivent pour la génération suivante. Ce processus est répété pendant un certain nombre de générations, en espérant que les solutions de la fonction fitness apparaissent dans la population.

  La recherche de solutions dans un espace complexe implique souvent un compromis entre deux objectifs apparemment contradictoires : l'exploitation des meilleurs solutions et l'exploration robuste de l'espace des solutions possibles [Ren95].

1.4.3 Exploration et Exploitation

1.4.3.1 Exploration

L'opérateur de mutation permet d'explorer des solutions en échappant aux minima locaux, c'est-à-dire que la meilleure solution n'est pas recherchée que dans la population initiale.

Inconvénient : Si l'exploration est très forte cela correspondra à une recherche aléatoire. Les bons individus ne sont pas conservés, il n'y a pas de convergence.

1.4.3.2 Exploitation

On améliore les performances de la population en multipliant les meilleurs individus et en éliminant les moins bons.

Inconvénient : Si l'exploitation est très forte, la recherche est limitée, on converge vers un minimum local.

À partir des deux définitions ci-dessus, il est clair qu'un compris entre exploration et exploitation s'avère nécessaire.

1.4.4 Compromis exploration et exploitation

Des études ont montré que les Algorithmes Evolutionnaires offrent un bon compromis entre exploration et exploitation [Mag98]. La nécessité d'avoir ce compromis, se justifie par le fait qu'une exploitation élevée entraîne une convergence vers un optimum local, alors qu'une exploration élevée entraîne la non convergence de l'algorithme. Ainsi pour aboutir à un réglage des parts respectives d'exploration et d'exploitation, il faut manipuler les divers paramètres de l'algorithme. Ces derniers seront traités en particulier pour une des catégories des AE.

Les sections suivantes présentent les fondements de ces algorithmes évolutionnaires.

1.5 Algorithmes Évolutionnaires

La naissance des algorithmes évolutionnaires fût marquée dans les années 60 et 70. Cette naissance est due à la recherche qui a abouti aux méthodes approchées. Ces méthodes ont été essayées pour résoudre une catégorie spécifique de problèmes. En effet, il existe une catégorie de problème pour lesquels il est difficile, de trouver une solution en un temps limité. Il est alors utile de trouver une technique permettant la localisation rapide de solutions sous optimales, sachant que l'espace de recherche a une taille et une complexité suffisamment importantes pour éliminer toute garantie d'optimalité. Pour cela, un système capable de s'automodifier au cours du temps, tout en améliorant sa performance dans l'accomplissement des tâches auxquelles il est confronté, semble ouvrir la voie à une recherche intéressante [Spa99].
Il s'agit des algorithmes évolutionnaires comptés parmi les méthodes de l'intelligence computationnelle. La Figure 1.3 en est une description.

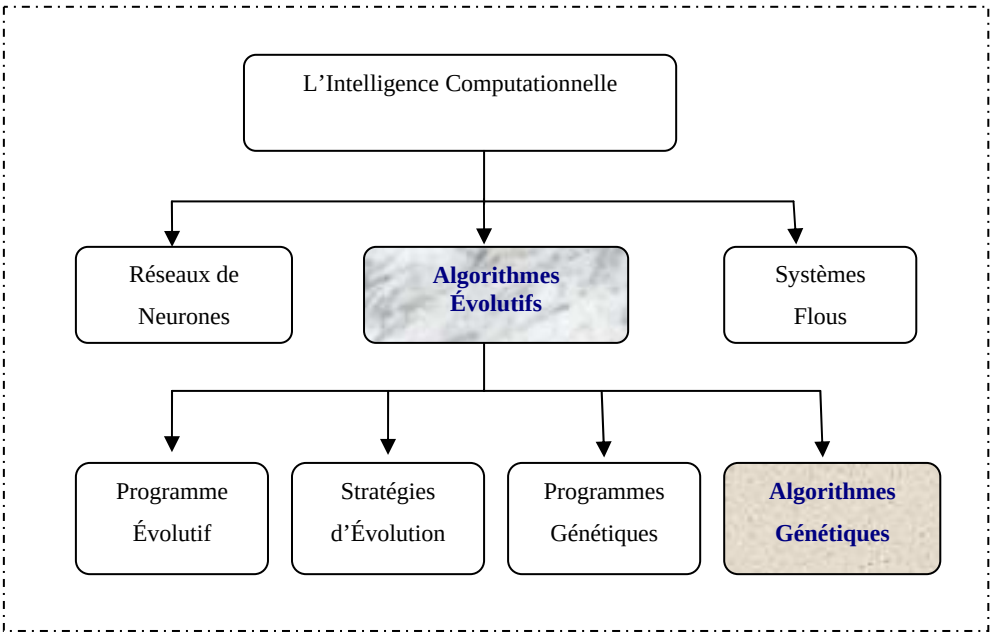


Figure 1. 3- Différentes méthodes de l'intelligence computationnelle

Les AE sont inspirés du concept de *sélection naturelle* élaboré par Charles Darwin. Le vocabulaire employé est directement calqué sur celui de la théorie de l'évolution et de la génétique. Ils sont introduits par Fogel en 1965 [Ben01], Rechenberg en 1973 [Ben01], Holland en 1975 [Ben01] et popularisés par Goldberg en 1989 [Gol94].

1.5.1 La sélection naturelle

S'il n'existe pas de preuve générale de l'efficacité des algorithmes évolutionnaires il est par contre aisé de constater l'efficacité de la sélection naturelle dans le monde vivant. Si l'on adhère à ce paradigme, il est clair que l'évolution a permis l'émergence d'organismes étonnamment adaptés à leur environnement. Les AE sont conçus par analogie avec ce processus d'évolution biologique et tirent leur puissance des mêmes mécanismes [Mag98].

Dans " The Origin of Species (1859) ", Darwin montre que l'apparition d'espèces distinctes se fait par le biais de la sélection naturelle de variations individuelles. Cette sélection naturelle est fondée sur la lutte pour la vie, due à une population tendant naturellement à s'étendre mais disposant d'un espace et de ressources finis. Il en résulte

que les individus les plus adaptés (*the fittest*) tendent à survivre plus longtemps et à se reproduire plus aisément. Le terme “adapté” se réfère à l’environnement, que l’on peut définir comme étant l’ensemble des conditions externes à un individu, ce qui inclut les autres individus [Mag98].

1.5.2 Principes généraux

Les algorithmes évolutionnaires sont basés sur des principes simples. En effet, ils font évoluer une population d’individus, dont l’objectif de trouver la solution optimale, où seulement les mieux adaptés à un environnement donné peuvent survivre et se reproduire. En termes mathématiques, l’environnement est la fonction de performance à optimiser, et elle constitue la seule information nécessaire aux AE, ce qui leur permet de traiter une grande variété de problèmes numériques difficiles à traiter avec des méthodes d’optimisation classiques [Ben01]. Ces algorithmes reposent sur une vision darwinienne relativement simpliste et une optimisation stochastique résumée dans le diagramme de la Figure 1.4 [Sch01].

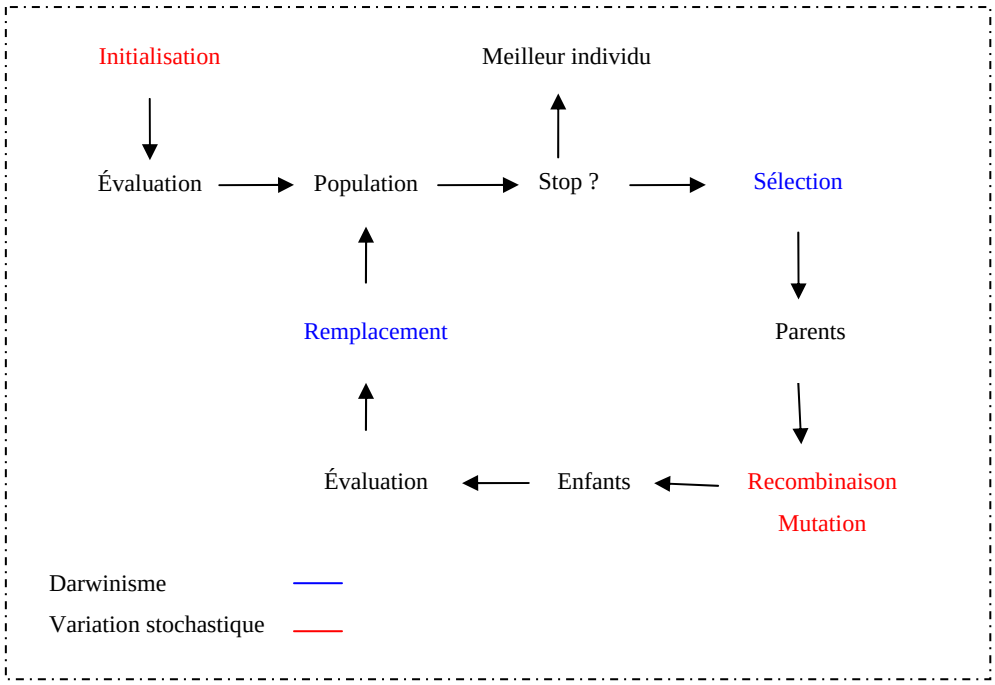


Figure 1. 4- Notions de base de l’algorithmique évolutionnaire

La Figure 1.4 explique alors qu'à chaque population, les individus les plus aptes sont ceux qui survivent pour être croisés et mutés par la suite constituant ainsi une nouvelle population. Ce processus est répété pendant un certain nombre de générations, en espérant que les optima de la fonction apparaissent dans la population.

Quel que soit le type de problème à résoudre, les AE opèrent selon les principes suivants : la population est initialisée de façon dépendante du problème à résoudre (l'environnement), puis évolue de génération en génération à l'aide d'opérateurs de sélection, de croisement et de mutation. Dans cette évolution les générations successives des différentes populations préservent une taille constante (mais ce n'est pas une règle). Pour l'environnement, il a pour charge d'évaluer les individus en leur attribuant une performance (fitness). Cette valeur favorisera la sélection des meilleurs individus, en vue, d'améliorer les performances globales de la population.

1.5.3 Catégories des algorithmes évolutionnaires

Depuis les années soixante, plusieurs tendances d'algorithmes évolutionnaires sont apparues et à peu près simultanément par différents chercheurs. Ces algorithmes se classent en 4 catégories, comme le décrit la Figure 1.5 [Spa99].

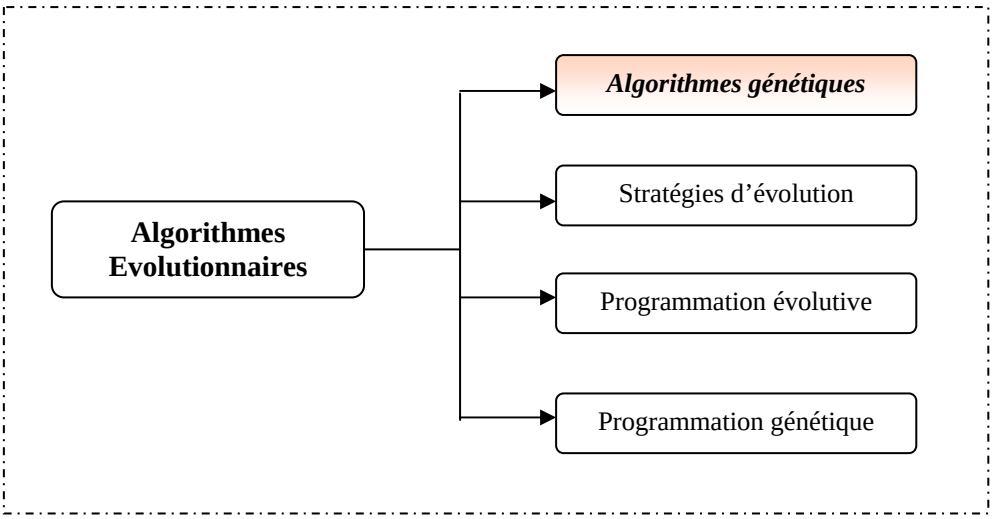


Figure 1. 5- Différentes branches des algorithmes évolutionnaires

Chacun de ces algorithmes manipule une population dont la taille, contrairement à la population naturelle, reste inchangée au long de l'évolution. Ils ont un principe de base

commun, qui caricature le principe de l'évolution biologique établie par Darwin. Toutefois ils ne diffèrent que sur les détails d'implantation des opérations et sur les procédures de sélection et remplacement de la population.

- **Algorithmes génétiques** (AGs), J. Holland, 1975 et D.E. Goldberg, 1989, Michigan, USA

  Les AGs ont été conçus comme outil d'optimisation stochastique. Ils sont inspirés des mécanismes de la génétique et de l'évolution naturelle des être vivants [Abd04]. Ils cherchent la solution globale d'une manière aléatoire. Ces algorithmes sont les plus connus et utilisés des algorithmes évolutionnaires.

- **Stratégies d'évolution** (S*E*), I. Rechenberg et H.P. Schwefel, 1965, Berlin.
  A l'origine, un individu unique produisait au moyen d'une mutation, un descendant unique, une solution s'opérait entre eux deux. Dès leurs origines, une grande originalité des stratégies d'évolution par rapport aux algorithmes génétiques a résidé dans leur utilisation du codage réel [All02].

- **Programmation évolutive** (PE), L.J. Fogel, 1966 et D.B. Fogel, 1991, 1995, Californie, USA.

  Les principales caractéristiques de ces algorithmes par rapport aux autres algorithmes évolutionnaires, étaient, à l'origine, le fait qu'ils n'utilisaient pas de croisement et surtout l'utilisation d'un mode de sélection originale ; la sélection par tournoi [All02].

- **Programmation génétique** (PG), J. Koza, 1990, Californie, USA.
  Le modèle de la PG s'inspire des algorithmes génétiques, a pour but de créer automatiquement un programme pour résoudre un problème. Pour cela on va utiliser non pas des chromosomes codant une solution, mais de petits programmes qui vont évoluer et s'assembler pour fournir un programme plus complet. Ces programmes sont en général des S-expressions LISP, qui ont l'avantage d'être facilement représentés par des arbres syntaxiques, où les nœuds sont des opérateurs, et les feuilles des variables ou des valeurs. L'algorithme utilisé pour la programmation génétique, suit le même schéma qu'un algorithme génétique standard : après une initialisation aléatoire (chaque individu est un programme)

une évaluation des individus s'effectuera suivie de l'application des opérateurs génétiques [Mad02].

1.6 Conclusion

Ce chapitre a entrepris une introduction du concept d'intelligence artificielle et plus spécifiquement une de ces voies de recherche, en l'occurrence la vie artificielle. Celle-ci est définie selon différentes formes. Ce premier chapitre a évoqué alors une de ces formes précisément les algorithmes évolutionnaires. Ces algorithmes utilisés comme outil d'optimisation par recherche probabiliste, sont basés sur le modèle de l'évolution naturelle. Ils sont reconnus avoir passé le stade de la recherche pure à celui de la recherche appliquée.

En outre, les différentes variantes des algorithmes évolutionnaires ont été exposées. En particulier, les algorithmes génétiques seront expliqués en détail au chapitre suivant car il s'agit du sujet de ce mémoire. En effet, nous chercherons à résoudre un problème d'optimisation en utilisant les algorithmes génétiques.

Chapitre 2

Algorithmes génétiques

Résumé : Dans ce second chapitre, nous introduisons les algorithmes génétiques, métaphores biologiques inspirées des mécanismes de l'évolution darwinienne et de la génétique, et utilisés comme outils d'optimisation ou de recherche combinatoire. Nous examinerons leur terminologie de base, leurs principes, les opérateurs participants à l'exploration de l'espace de recherche tout en mettant l'accent sur leur utilité, leur conception et leur mode de fonctionnement. Ces algorithmes sont souvent associés à un processus stochastique, pour éviter qu'un mauvais choix de départ ou des biais systématiques ne les pénalise. Aussi, nous exposerons d'une manière générale les méthodes les plus utilisées pour chaque opérateur génétique à savoir sélection, mutation et croisement. Nous concluons par la difficulté qui réside dans le choix des différents paramètres d'un algorithme génétique.

2.1 Introduction

Malgré l'évolution permanente des calculateurs et les progrès fulgurants de l'informatique, il existe pour plusieurs problèmes d'optimisation une taille critique de l'espace de solutions admissibles. La méthode permettant d'obtenir une solution optimale est bien évidemment celle de l'énumération complète de l'espace de recherche. Cette dernière est dans la plupart des cas prohibitive. Compte tenu de ces difficultés, certains chercheurs, il y a environ une trentaine d'années, se sont interrogés pour savoir comment faire mieux : il est apparu plusieurs similarités entre le monde biologique et le monde informatique. De ce fait, l'approche évolutive fût utilisée. En particulier, les algorithmes génétiques vu qu'ils présentent des qualités intéressantes pour la résolution de divers problèmes. Ils sont basés sur la théorie de l'évolution des espèces dans leur milieu naturel, soit une transposition artificielle des concepts basiques de la génétique et des lois de survie énoncées par Charles Darwin : les individus les plus adaptés survivent et se reproduisent. Selon Darwin, les mécanismes à l'origine de l'évolution naturelle des êtres vivants reposent sur la compétition qui sélectionne les individus les plus adaptés à l'environnement actuel au détriment des autres. L'hypothèse de la théorie de Darwin, compte tenu des connaissances actuelles de la génétique, montre que ces mécanismes ne sont pas toujours justifiés. Ces mêmes mécanismes seront utilisés dans l'implémentation de l'algorithme génétique.

L'application des algorithmes génétiques aux problèmes d'optimisation a été formalisée par Goldberg en 1989 [Fin02]. Ensuite ils se sont vite imposés (comme méthodes d'optimisation globale), en permettant l'optimisation des problèmes très variés. Particulièrement, ces méthodes permettent de traiter des problèmes dont la taille est considérable, ou encore des problèmes non décrits de manière explicite. Leur vaste champ d'action, leur implantation généralement aisée sont certainement à l'origine de leur succès. Ce chapitre présente donc, cette méthode d'optimisation stochastique, qui sera utilisée dans la suite pour le problème de la gestion de stocks. Cette présentation nous semble nécessaire du fait de la relative nouveauté des algorithmes génétiques. Elle se veut générale et sans lien immédiat avec le problème de l'optimisation des stocks.

2.2 Algorithmes génétiques

John Holland, ses collègues et ses étudiants ont développé à l'université de Michigan les Algorithmes Génétiques (AGs) [Gol94], métaphores biologiques inspirées des mécanismes de l'évolution darwinienne (sélection naturelle) et de la génétique. Ces métaphores prennent la forme d'algorithmes de recherche appelés "algorithmes génétiques" [Ren95].

Ces algorithmes font partie de la classe des algorithmes dits stochastiques. En effet une grande partie de leur fonctionnement est basée sur le hasard. Bien qu'utilisant le hasard, les AGs ne sont pas purement aléatoires. Ils exploitent efficacement l'information obtenue précédemment pour spéculer sur la position de nouveaux points à explorer, avec l'espoir d'améliorer la performance [Mad02], [Gol94].

Les algorithmes génétiques permettent à une population de solutions de converger vers les solutions optimales. Pour ce faire, ils vont utiliser un mécanisme de sélection des individus de la population (les solutions potentielles). Les individus sélectionnés vont être croisés entre eux (exploitation), et certains vont être mutés (exploration). Ces mécanismes d'exploitation et d'exploration vont permettre de converger vers les bonnes solutions en évitant, autant que faire se peut, les optima locaux [Mad02].

2.3 Terminologie et éléments de base

Un algorithme génétique recherche les extrêmes d'une fonction définie sur un espace de données appelé *population*. Par analogie avec la génétique, chaque *individu* de cette population est un chromosome et chaque *caractéristique* de l'individu est un gène. Dans un cas simple, un gène sera représenté par un bit (0 ou 1), un chromosome par une chaîne de bits. Chaque gène représente une partie élémentaire du problème, il peut être assimilé à une variable et peut prendre des valeurs différentes appelées ***allèles.*** La position du gène dans le chromosome se nomme *locus* [Har03].

On parle également de *génotype* et de *phénotype*. Le génotype représente l'ensemble des valeurs des gènes du chromosome alors que le phénotype représente la solution réelle après transformation du chromosome. Lors de la génération d'une nouvelle population, des opérateurs génétiques tels que la sélection, le croisement et la mutation sont nécessaires pour la manipulation des chromosomes [Har03].

Le tableau 2.1 présente une récapitulation de la terminologie naturelle et celle utilisée par les algorithmes génétiques [Gol94], [Mad02].

Nature	**Algorithme génétique**
Chromosome	Chaîne
Gène	Trait, caractéristique
Allèle	Valeur de la caractéristique
Locus	Position dans la chaîne
Génotype	Structure Ensemble des valeurs des gènes
Phénotype	Ensemble de paramètres, structure décodée Evaluation d'un génotype

Tableau 2. 1- Comparaison de la terminologie naturelle et celle des algorithmes génétiques

Les AGs utilisent donc un vocabulaire similaire à celui de la génétique. On parlera ainsi d'individus ou chromosomes dans une population. Chaque individu ou chromosome est constitué d'un ensemble d'éléments appelés gènes contenant les caractères héréditaires de l'individu [Ren95]. Ils utilisent un mécanisme de sélection naturelle, basée essentiellement sur la reproduction et sur le codage génétique qui stocke les informations décrivant l'individu sous forme de gènes [Reb99] imitant les systèmes naturels de l'évolution des espèces.

2.4 Évolution des espèces

Dans un environnement quelconque dans lequel vit une population primitive, i.e. peu adaptée à cet environnement. Bien sûr, quoique globalement inadaptée, cette population n'est pas uniforme : certains individus sont mieux armés que d'autres pour profiter des ressources offertes par l'environnement (nourritures, abris, etc.) et pour faire face aux dangers qui y rôdent (prédateurs, intempéries, etc.). Ces individus mieux équipés ont par conséquent une probabilité de survie plus grande que leurs congénères

et auront de fait d'autant plus de chances de pouvoir se reproduire. En se reproduisant entre individus bien adaptés, ils vont transmettre à leurs enfants ces caractéristiques qui faisaient leur excellence. La population qui résultera de cette reproduction sera donc globalement mieux adaptée à l'environnement que la précédente puisque la plupart des individus auront hérité de plusieurs (puisque chacun hérite à la fois de sa mère et de son père) des caractéristiques de l' "élite" de la génération précédente. Et c'est ainsi, en recombinant à chaque génération les caractéristiques élémentaires de bonne adaptation et en saupoudrant-le tout d'un peu de hasard, que la population va évoluer vers une adéquation toujours meilleure avec l'environnement. Par analogie, les AGs joignent le même principe, ils sont basés sur le principe d'" évolution" d'une population d'individus. Dans celle-ci, ce sont en général les plus forts, c'est-à-dire les mieux adaptés au milieu, qui survivent et engendrent des progénitures. À partir des données du problème, on crée (généralement aléatoirement) une " population" de solutions admissibles. Puis on évalue chacune des solutions. On élimine une partie infime de celles qui se sont montrées inutiles, et on recombine les gènes des autres afin d'obtenir de nouveaux individus-solutions. Ainsi, à chaque génération un nouvel ensemble de créatures artificielles (des chaînes de caractères) est crée en utilisant des parties des meilleurs individus de la génération précédente ainsi que des parties innovatrices. Selon la théorie évolutionniste, cette nouvelle génération sera globalement plus adaptée au problème que la précédente. Ce procédé est alors répété jusqu'à la naissance d'une solution que l'on jugera satisfaisante [Har03].

2.5 A quoi sert l'algorithme génétique ?

L'algorithme génétique *résout des problèmes* n'ayant pas de méthode de résolution décrite précisément ou dont la solution exacte, si elle est connue, est trop compliquée pour être calculée en un temps raisonnable. Ceci dit, face à un problème pour lequel il existe pour ainsi dire une infinité de solutions, plutôt que d'essayer naïvement toutes les solutions une à une pour trouver la meilleure, on va explorer l'espace des solutions en se laissant guider par les principes des algorithmes génétiques.

2.6 Conception d'un algorithme génétique

La simplicité de mise en œuvre et l'efficacité constituent deux des caractéristiques les plus attrayantes de l'approche proposée par les AGs. La mise en œuvre d'un algorithme génétique sollicite la disponibilité [Gol94], [Har03] :

- d'une *représentation génétique* du problème, c'est-à-dire un codage approprié des solutions sous la forme de chromosomes. Cette étape associe à chacun des points de l'espace de recherche une structure de données. Elle se place généralement après une phase de modélisation mathématique du problème traité. La qualité du codage des données conditionne le succès des algorithmes génétiques ;
- d'un *mécanisme de génération* de la population initiale. Ce mécanisme doit être capable de produire une population non homogène qui servira de base pour les générations futures. Le choix de la population initiale est important car il peut prendre plus ou moins rapidement la convergence vers l'optimum global. Dans le cas où l'on ne connaît rien sur le problème à résoudre, il est essentiel que la population initiale soit répartie sur tout le domaine de recherche ;
- d'une *fonction d'évaluation* pour mesurer la force de chaque chromosome ;
- d'un *mode de sélection* des chromosomes à reproduire ;
- des *opérateurs* permettant de diversifier la population au cours des générations et d'explorer l'espace de recherche. L'opérateur de croisement recompose les gènes d'individus existant dans la population, l'opérateur de mutation a pour but de garantir l'exploration de l'espace de recherche ;
- des *valeurs* pour les *paramètres* qu'utilise l'algorithme : taille de la population, nombre total de générations ou critère d'arrêt, probabilités de croisement et de mutation.

2.7 Comment fonctionne l'algorithme génétique ?

Un algorithme génétique fonctionne typiquement à travers un cycle simple de quatre étapes [Har03]:

1. création d'une population de chromosomes ;
2. évaluation de chaque chromosome ;
3. sélection des meilleurs chromosomes ;
4. manipulation génétique, pour créer une nouvelle population de chromosomes.

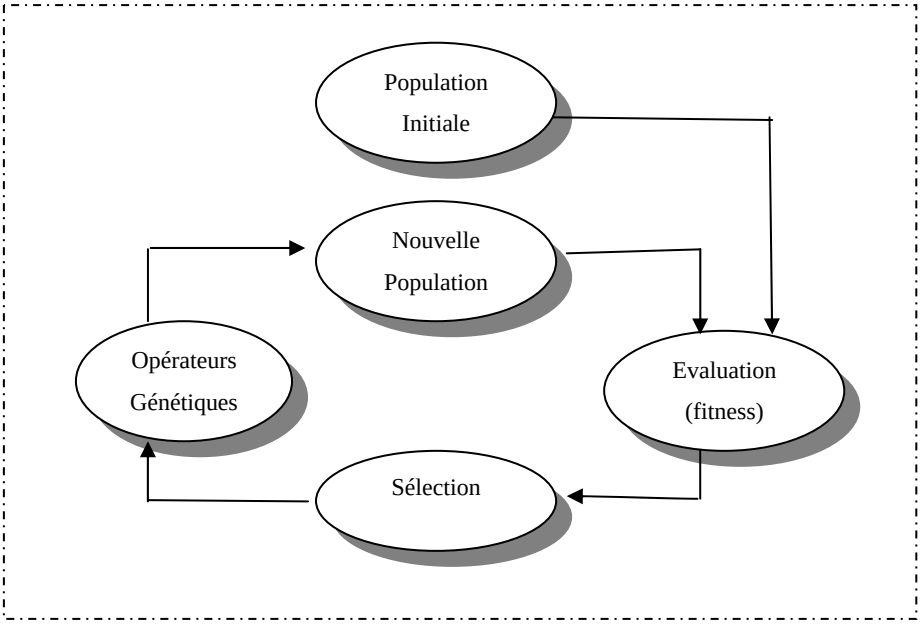


Figure 2. 1- Cycle génétique

Le cycle décrit par la Figure 2.1 est inspiré par la terminologie génétique. Lors de chaque cycle, une nouvelle génération de solutions du problème est obtenue. Initialement, une population initiale est générée où chaque individu-solution de la population est codé sous forme d'une chaîne de caractères (chromosomes). Ensuite, une évaluation de chaque chromosome sera établie. Cette évaluation consiste à évaluer la qualité des chromosomes à l'aide de la fonction d'évaluation : fitness. Ce qui permet de sélectionner les chromosomes les plus adaptés et par conséquent leur appliquer les opérateurs génétiques (croisement et mutation) ce qui crée une nouvelle génération.
A la fin du cycle, une nouvelle population est acquise ouvrant ainsi la voie pour une nouvelle génération et par conséquent un nouveau cycle.

2.8 Variantes

En fait, les algorithmes génétiques sont une famille d'algorithmes, basés autour des mêmes idées. Cependant il existe beaucoup de variantes possibles suivant la représentation choisie, les opérateurs de croisement, de mutation et de sélection [Reb99]. La section suivante présente les choix les plus courants qui définissent les variantes.

2.8.1 Codage

Le codage est une modélisation d'une solution d'un problème donné sous forme d'une séquence de caractères appelée chromosome où chaque caractère, dit aussi gène, représente une variable ou une partie du problème. La tâche principale consiste à choisir le contenu des gènes qui facilite la description du problème et respecte ses contraintes [Har03]. La littérature définit deux types de codage : binaire et réel.

2.8.1.1 Codage binaire

Le codage classique utilise l'alphabet binaire : 0,1. Dans ce cas le chromosome représente simplement une suite de 0 et de 1. Le codage binaire est également indépendant des opérateurs génétiques (croisement et mutation) du moment où ces derniers ne nécessitent aucune spécification. En effet, toute manipulation d'un chromosome donne naissance à un nouveau chromosome valide. Dans la pratique, le codage binaire peut présenter des difficultés. En effet, il est parfois très difficile ou très lourd de coder des solutions de cette manière. En outre, dans certain cas la taille mémoire requise peut devenir prohibitive [Har03].

2.8.1.2 Codage réel

Pour certain problème d'optimisation, il est plus pratique d'utiliser un codage réel des chromosomes. Un gène est ainsi représenté par un nombre réel au lieu d'avoir à coder les réels en binaire puis de les décoder pour les transformer en solutions effectives. Le codage réel permet d'augmenter l'efficacité de l'algorithme génétique et d'éviter des opérations de décodage supplémentaires. En effet, un chromosome codé en réels est plus court que celui codé en binaire.

2.8.2 Évaluation : fitness

L'opérateur d'évaluation n'est pas anodin. Il est utilisé par l'opérateur de sélection pour faire son choix des individus à conserver. Ainsi, pour mesurer les performances de chaque individu qui correspond à une solution donnée du problème à résoudre, on introduit une fonction d'évaluation. Elle permet de quantifier la capacité d'un individu à survivre en lui affectant un poids couramment appelé *fitness*. La force de chaque chromosome de la population est calculée afin que les plus forts soient retenus (étape de

sélection) puis modifiés (croisement et mutation). La complexité de la fonction d'évaluation dépend essentiellement du problème et de ses contraintes [Har03], [Mad02].

Ces deux derniers éléments, codage et évaluation, sont les seuls éléments spécifique au problème à résoudre. Une fois qu'ils sont fixés, l'algorithme génétique que l'on appliquera sera toujours le même [All02].

2.8.3 Population initiale

Une fois le codage choisi, une population initiale formée de solutions admissibles du problème doit être déterminée. Plusieurs mécanismes de génération de la population initiale sont utilisés dans la littérature [Har03]. Le choix de l'initialisation se fera en fonction des connaissances que l'utilisateur a sur le problème. S'il n'a pas d'informations particulières, alors une initialisation aléatoire, la plus uniforme possible afin de favoriser une exploration de l'espace de recherche maximum, sera la plus adaptée. Mais dans d'autres cas, il est possible d'utiliser d'autres mécanismes. Par ailleurs, cette étape présente un problème principal qui est celui du choix de la taille de la population. En effet une population trop grande augmente le temps de calcul et demande un espace mémoire considérable, alors qu'une population trop petite conduit à l'obtention d'un optimum local.

2.8.4 Critère d'arrêt

Déterminer l'arrêt d'un processus génétique est l'une des difficultés majeures de l'approche génétique. En effet, si l'on excepte le cas des problèmes artificiels, on ne sait jamais si l'on a trouvé l'optimum. Dans la pratique, l'utilisateur déclare un nombre de générations maximum. La recherche peut également être stoppée lorsque tous les individus d'une même population sont des copies d'un même individu. On dit alors qu'il y a " perte de diversité génétique " [Seb96].

Les critères d'arrêt se résument alors en :

1. Arrêt après un nombre de générations fixé à priori.
2. Arrêt lorsque la population cesse d'évoluer ou en présence d'une population homogène.

2.8.5 Sélection

L'opérateur de sélection est chargé de " favoriser" les meilleurs individus [Reb99]. Plus formellement, l'opérateur de sélection va générer à partir de la population courante une nouvelle population par copie des individus choisis de la population courante. La copie des chaînes s'effectue en fonction des valeurs de la fonction d'adaptation. Ce procédé permet de donner aux meilleures chaînes, une probabilité élevée de contribuer à la génération suivante. Cet opérateur est bien entendu une version artificielle de la sélection naturelle, la survie darwinienne des chaînes les plus adaptées [Gol94].

Il existe de nombreuses techniques de sélection, les plus courantes seront évoquées dans la section suivante.

- **La sélection par classement** : elle consiste à ranger les individus de la population dans un ordre croissant (ou décroissant selon l'objectif) et à retenir un nombre fixé de génotypes. Ainsi, seuls les individus les plus forts sont conservés. L'inconvénient majeur de cette méthode est la convergence prématurée de l'algorithme génétique. Il est parfois nécessaire de garder quelques individus jugés faibles pour créer la diversité au niveau de la population. Une autre difficulté consiste à fixer une limite à la sélection ce qui empêche parfois de garder des bons candidats pour les futures générations [Har03].

- **La sélection par la roulette :** elle consiste à créer une roue de loterie biaisée pour laquelle chaque individu de la population occupe une section de la roue proportionnelle à sa valeur d'évaluation. Ainsi, même les individus les plus faibles ont une chance de survivre.

  Si la population d'individus est de taille égale à N, alors la probabilité de sélection d'un individu x_i notée $p(x_i)$ est égale à :

$$P(x_i)=\frac{F(x_i)}{\sum_{k=1}^{N} F(x_k)}$$

En pratique, on calcule pour chaque individu x_i sa probabilité cumulée $q_i = \sum_{j=1}^{i} p(x_j)$ et on choisi aléatoirement un nombre r compris entre 0 et 1.

L'individu retenu est x_1 si $q_1 \geq r$ ou $x_i (2 \leq i \leq N)$ si $q_{i-1} < r \leq q_i$. Ce processus est répété N fois. Avec une telle sélection, un individu fort peut être choisi plusieurs fois. Par contre, un individu faible a moins de chance d'être sélectionné [Har03]. C'est cette sélection qui a été exclusivement utilisée dans ce mémoire.

- **La sélection par tournoi** : elle consiste à choisir aléatoirement deux ou plusieurs individus et à sélectionner le plus fort. Ce processus est répété plusieurs fois jusqu'à l'obtention de N individus. L'avantage d'une telle sélection est d'éviter qu'un individu très fort soit sélectionné plusieurs fois [Har03].

2.8.6 Croisement

La naissance d'un nouvel individu, nécessite la prise aléatoire d'une partie des gènes de chacun des deux parents. Ce phénomène, issu de la nature est appelé croisement (crossover). Il s'agit d'un processus essentiel pour explorer l'espace des solutions possibles. Une fois la sélection terminée, les individus sont aléatoirement répartis en couples. Les chromosomes parents sont alors copiés et recombinés afin de produire chacun deux descendants ayant des caractéristiques issues des deux parents. Dans le but de garder quelques individus parents dans la prochaine population, on associe à l'algorithme génétique une probabilité de croisement, qui permet de décider si les parents seront croisés entre eux ou s'ils seront tout simplement recopiés dans la population suivante [Har03], [Ren99].
La littérature définit plusieurs opérateurs de croisement. Ils diffèrent selon le type de codage adapté et la nature du problème traité.

2.8.6.1 Croisement binaire

Ce croisement peut avoir recourt à plusieurs types en occurrence [Har03] :

- **Croisement en 1-point :** c'est le croisement le plus simple et le plus connu dans la littérature. Il consiste à choisir au hasard un point de croisement pour chaque couple de chromosomes. Les sous-chaînes situées après ce point sont par la suite inter-changées pour former les deux fils (Figure 2.2).

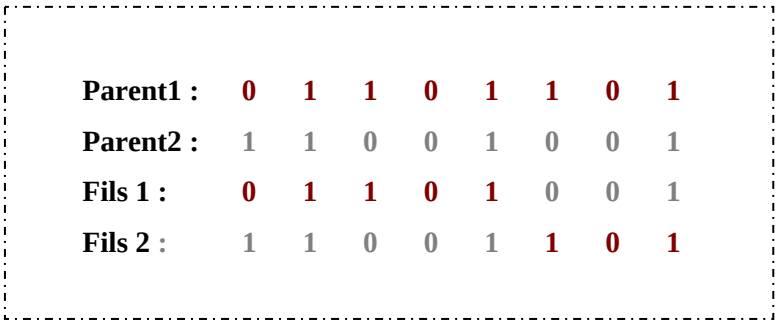


Figure 2. 2- Croisement en un point de deux chromosomes

- **Croisement en n-points :** Ce type de croisement s'énonce par un choix aléatoirement de n-points de coupure pour dissocier chaque parent en n+1 fragments. Pour former un fils, il suffit de concaténer alternativement n+1 sous chaînes à partir des deux parents. Ce croisement cherche à explorer tout l'espace de solutions possibles en créant des descendants ayant des caractéristiques très loin des parents.

- **Croisement en 2-points :** c'est un cas particulier du croisement en n-points. On choisit aléatoirement deux points de coupure pour créer les descendants.

- **Croisement uniforme :** cette technique génère des progénitures gène par gène à partir des deux parents. Il existe des versions distinctes de ce croisement. La plus connue est celle qui utilise un masque. S'il est égal à 1, l'enfant 1 reçoit l'allèle correspondant du parent 1 et l'enfant 2 reçoit celui du parent 2. Sinon, l'échange se fait dans l'autre sens (Figure 2.3).

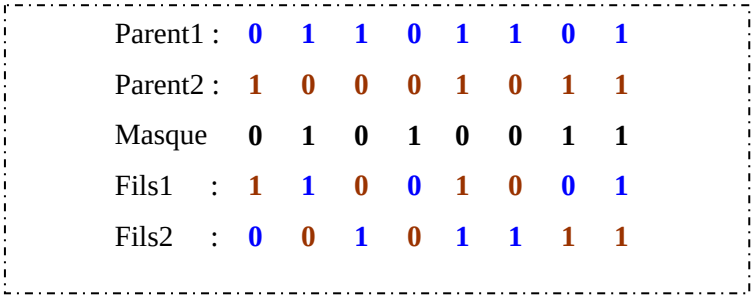


Figure 2. 3- Croisement uniforme

2.8.6.2 Croisement réel

Le codage réel requiert des opérateurs génétiques spécifiques pour la manipulation des chromosomes. Il est de plusieurs types [Har03]:

- **Ordre de base cyclique** : pour créer un fils, il suffit de copier une sous-chaîne d'un parent et de compléter les gènes manquants à partir de l'autre parent, en maintenant l'ordre des gènes. Généralement, une fois deux chromosomes parents sélectionnés pour le croisement, deux points de coupures sont choisis aléatoirement sur chaque parent. Ensuite on place les sous-chaînes entre les points de coupure sur les deux fils dans la même position que les parents. Pour compléter les gènes manquants du fils 1, on commence par insérer les gènes situés à droite du deuxième point de coupure du parent 2 tout en gardant l'ordre des gènes et en ignorant les gènes déjà pris. Le deuxième fils est complété à partir du parent 1 de la même manière que le fils 1. La Figure 2.4 montre sur un exemple des étapes de ce type de croisement.

			1er pt				2ème pt	
Père 1	a	b	c	d	e	f	g	h	i
Père 2	f	b	g	a	e	i	c	h	d
Fils 1	.	.	.	d	e	f	g	.	.
Fils 2	.	.	.	a	e	i	c	.	.
Fils 1	a	i	c	d	e	f	g	h	b
Fils 2	d	f	g	a	e	i	c	h	b

Figure 2. 4- Croisement d'ordre de base cyclique

- **Croisement uniformément continu** : ce type a été suggéré pour produire des chromosomes valides. Un chromosome $X=(x_1,x_2,...,x_n)$ est valide lorsque : $\sum_{i=1}^{n} x_i=1$.

  Étant donné deux chromosomes valides $X=(x_1,x_2,...,x_n)$ et $Y=(y_1,y_2,...,y_n)$ les descendants $X'=(x'_1,x'_2,...,x'_n)$ et $Y'=(y'_1,y'_2,...y'_n)$ sont définis de la façon suivante : $x'_i=sx_i+(1-s)y_i$ et $y'_i=(1-s)x_i+sy_i$. Où s est une constante choisie à chaque itération aléatoirement dans l'intervalle $[-0.5,0.5]$.

- **Croisement d'ordre maximal** : ce type de croisement a pour objectif de garder le maximum possible les positions et l'ordre des gènes. On commence par choisir aléatoirement deux points de coupure. Les sous-chaînes situées au milieu sont inter-changées. Les gènes manquants sont par la suite complétés à partir de chaque père en allant de gauche à droite et en choisissant le premier caractère disponible. A la différence du croisement de base cyclique, le fils 1 est complété à partir du parent 1 et le fils 2 à partir du parent 2. La Figure 2.5 illustre un exemple de croisement d'ordre maximal.

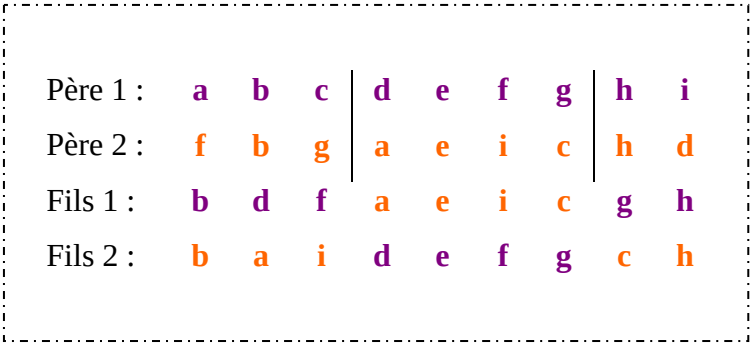


Figure 2. 5- Croisement d'ordre maximal

2.8.7 Mutation

La mutation est définie étant la modification aléatoire d'une partie d'un chromosome. Elle constitue une exploration aléatoire de l'espace des chaînes [Gol94]. C'est un phénomène qui a un rôle théoriquement plus marginal : il est là pour éviter une perte irréparable de la diversité [Ren95]. Différentes manières de mutation d'un chromosome sont aussi définies dans la littérature.

2.8.7.1 Mutation en codage binaire

Dans un algorithme génétique simple, la mutation en codage binaire est la modification aléatoire occasionnelle (de faible probabilité) de la valeur d'un caractère de la chaîne [Gol94].

2.8.7.2 Mutation en codage réel

Pour le codage réel, les opérateurs de mutation les plus connus et les plus utilisés sont les suivants [Har03] :

- L'opérateur **d'inversion simple** : consiste à choisir aléatoirement deux points de coupure et inverser les positions des bits situés au milieu.

- L'opérateur **d'insertion** : consiste à sélectionner au hasard un bit et une position dans le chromosome à muter, puis à insérer le bit en question dans la position choisie.

- L'opérateur **d'échange réciproque** : cet opérateur permet la sélection de deux bits et les inter changés.

L'utilisation de probabilités ne signifie pas que la méthode n'est qu'une exploration aléatoire. Les AGs utilisent des choix aléatoires comme des outils pour guider l'exploration à travers les régions de l'espace de recherche, avec une amélioration probable [Gol94].

2.9 Valeurs des paramètres

Les paramètres qui conditionnent la convergence d'un algorithme génétique sont :

- la taille de la population d'individus ;
- le nombre maximal de générations ;
- la probabilité de croisement ;
- la probabilité de mutation.

Les valeurs de tels paramètres dépendent fortement de la problématique étudiée. Ainsi il n'existe pas de paramètres qui soient adaptés à la résolution de tous les problèmes qui peuvent être posés à un algorithme génétique. Cependant, certaines valeurs sont souvent utilisées (définies dans la littérature) et peuvent être de bons points de départ pour démarrer une recherche de solutions à l'aide d'un AG.

- la probabilité de croisement est choisie dans l'intervalle [0.7, 0.99] ;
- la probabilité de mutation est choisie dans l'intervalle [0.001,0.01].

Trouver de bonnes valeurs à ces paramètres est donc un problème parfois délicat.

2.10 Applications

Ayant été reconnue comme une approche valide des problèmes nécessitant une exploration performante et économique du point de vue calcul, les algorithmes génétiques sont maintenant appliqués plus largement, aux domaines des affaires, à la recherche scientifique en général, ainsi que pour l'industrie. Les raisons de ce nombre grandissant d'applications sont claires. Ces algorithmes sont simples d'un point de vue de calcul, cependant très performants dans leur recherche d'amélioration [Gol94].

2.11 Conclusion

Ce chapitre a établi les fondations nécessaires à la compréhension des algorithmes génétiques, de leurs mécanismes et de leur puissance. Ces algorithmes classés parmi les méthodes stochastiques, s'inspirent de l'évolution génétique des espèces, plus précisément du principe de la sélection naturelle. C'est initialement la quête de robustesse qui a orienté vers ces méthodes ; les systèmes naturels sont robustes, efficaces et performants. En reproduisant sous forme artificielle le principe naturel de l'algorithme de sélection de la meilleure adaptation, les chercheurs visés l'atteinte des mêmes performances.

Le domaine d'application des algorithmes génétiques est assez large. En effet, depuis leur adaptation, ces méthodes connaissent une expansion considérable. Les algorithmes génétiques dans les applications qu'on leur a soumis ont montré leur grande souplesse et leur simplicité d'utilisation. Cette science à part entière a trouvé des applications pratiques que ce soit dans l'industrie ou dans d'autres. Ces algorithmes semblent être une voie prometteuse et laissent espérer une résolution des problèmes de la gestion des stocks. Par conséquent le chapitre quatre traite une application de l'algorithme génétique à ce type de problème. Toutefois, avant d'y arriver au chapitre quatre, on passe par le chapitre trois, chargé de détailler exhaustivement l'application de l'algorithme génétique implémenté à des fonctions mathématiques. Ce passage est jugé indispensable car les résultats de l'optimisation des fonctions mathématiques permettent de valider l'algorithme génétique implémenté.

Chapitre 3

Optimisation des Fonctions Mathématiques par AG

Résumé : Le survol du chapitre deux a donné une idée précise, de ce que sont les algorithmes génétiques et de leur fonctionnement. Ce chapitre s'oriente vers une présentation des résultats d'optimisation de différentes fonctions mathématiques par les AGs. Ce passage par ces fonctions n'a qu'un seul but ; celui de la validation de l'algorithme génétique implémenté. Chaque fonction utilisée en exemple connaît une solution analytique qui sera comparée avec celle obtenue par l'AG. En outre, cette validation va permettre d'envisager son utilisation ultérieure pour le problème considéré. Ce chapitre, commence alors par accentuer l'importance de l'optimisation. Le reste du chapitre sera consacré volontairement à la présentation des résultats obtenus par l'algorithme génétique en définissant comme fonction d'adéquation les différentes fonctions mathématiques. Pour conclure, une synthèse des résultats numériques et graphiques obtenus sera présentée.

3.1 Introduction

Le point de vue classique de l'optimisation est bien présenté par Beightler, Phillips et Wilde [Gol94] : "Le désir humain de perfection trouve son expression dans la théorie de l'optimisation. Elle étudie comment décrire et atteindre ce qui est meilleur, une fois que l'on connaît comment mesurer et modifier ce qui est mauvais… La théorie de l'optimisation comprend l'étude quantitative des optimums et les méthodes pour les trouver".

Ainsi l'optimisation cherche à améliorer une performance en se rapprochant d'un (ou des) point(s) optimum(s). L'ordinateur qui est un outil parfait pour l'optimisation sera utilisé tant que les paramètres mis en jeu peuvent être représentés sous forme informatique. Donc, l'ordinateur va recevoir en entrée des données et fournit en sortie une solution. Est-ce l'unique solution ? Pas dans tous les cas. Est-ce la meilleure solution ? C'est la question la plus difficile. L'optimisation est l'outil mathématique qui nous aide à répondre à ces questions [Hau98].

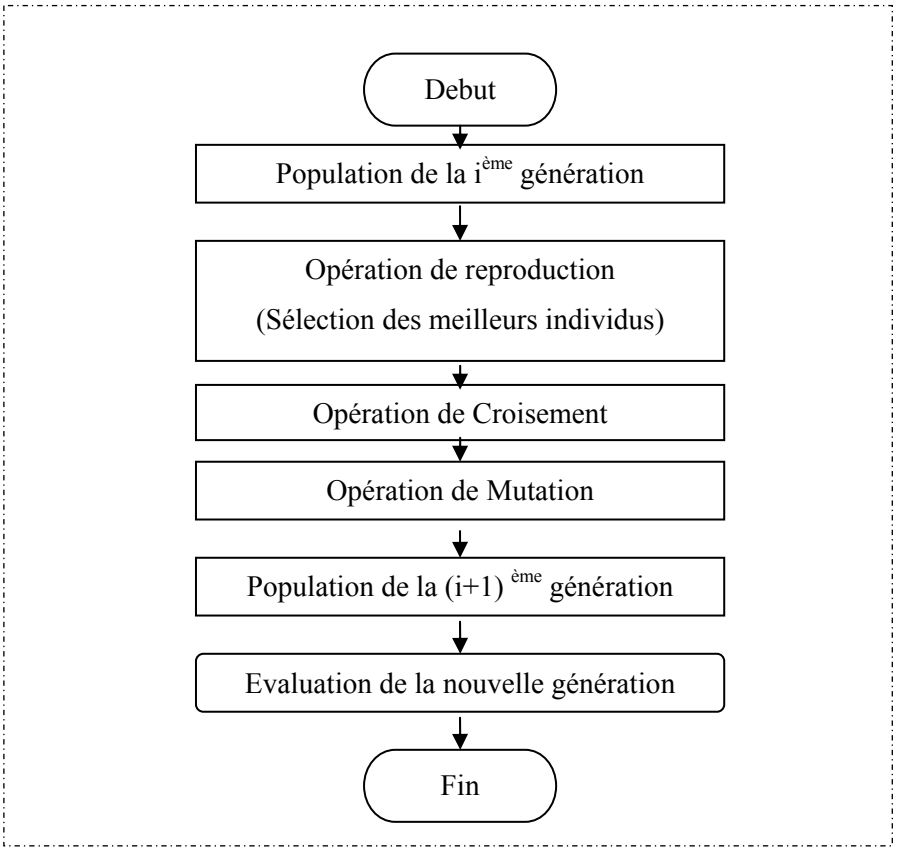
Effectivement, l'optimisation est une technique de grande importance pour le traitement des problèmes de prise de décision. Elle a pris une grande ampleur avec l'évolution substantielle de technologie des systèmes informatiques en terme de capacité et de rapidité des traitements [Abd04].
L'optimisation peut être abordée par des méthodes déterministes ou stochastiques (évoquées dans le deuxième chapitre) dépendant de la nature et du degré de complexité du problème à traiter. Les méthodes déterministes sont connues par leurs rapidités d'aboutissement à la solution mais tendent à subir une récession au fur et à mesure que le nombre de variables à optimiser devient important et que le système devient plus complexe [Abd04]. En revanche, les méthodes stochastiques spécifiquement les algorithmes génétiques appartenant à la famille des algorithmes évolutionnaires, s'appuyant sur des techniques dérivées de la génétique et de l'évolution naturelle, ne requièrent a priori pas de régularité sur les fonctions optimisées [All02].

Ces méthodes à caractère de recherche aléatoire, admettent à la fois une exploration et exploitation de l'espace de recherche d'une part, et d'autre part ne sont pas fondamentalement limitées par des hypothèses contraignantes sur le domaine d'exploration, i.e., des hypothèses relatives à l'estimation du vecteur initial des paramètres, continuité et l'existence des dérivées. Vu l'actualité des algorithmes génétiques et leurs pouvoirs de traiter de nombreuses applications dans différents domaines, notre attention a été focalisée sur leur application à un problème particulier (qui sera abordé au chapitre suivant). Toutefois, on a jugé fondamental, leur application à un problème mathématique spécifiquement à des fonctions mathématiques avant d'aborder l'objectif principal, et ceci dans un seul et unique but : validation de l'algorithme génétique implémenté.

Ce chapitre de transite présente tout d'abord une description formelle des différentes étapes de l'algorithme génétique, propose ensuite une démonstration sur les fonctions mathématiques et s'achève par les résultats obtenus.

3.2 Implantation de l'algorithme génétique

Un algorithme génétique repose fondamentalement sur la recherche d'un ou des extrema d'une fonction d'adaptation (aussi appelée fonction adéquation). Il s'agit donc d'un algorithme itératif de recherche globale dont le but est d'optimiser la fonction d'adéquation. Pour atteindre cet objectif, l'algorithme travaille en parallèle sur une population de chromosomes, distribués dans l'entièreté de l'espace de recherche [Ren95]. L'algorithme démarre alors avec une population initiale cherchant la combinaison optimale des paramètres correspondant à la meilleure solution. A chaque génération, est créée une nouvelle population avec le même nombre de chromosomes. Au fur et à mesure des générations, les chromosomes vont tendre en général vers l'optimum de la fonction d'adéquation. Le cycle d'une génération de cette méthode d'optimisation est illustré par l'organigramme 3.1 [Abd04]. Ce dernier montre bien que le cycle est constitué d'un ensemble d'étapes qui serviront de base à l'implémentation de l'algorithme génétique.



Organigramme 3. 1- Cycle d'une génération de l'algorithme génétique

Il semble primordial de détailler chacune des étapes décrite par le cycle d'une génération de l'algorithme génétique présenté par l'organigramme 3.1, afin d'assurer une compréhension plus profonde.

3.2.1 Instructions de l'algorithme génétique implémenté

Pour chercher le maximum d'une fonction dans l'intervalle [a, b] avec une précision de n chiffres significatifs, on exécute les différentes instructions de l'algorithme génétique 3.2 [Abd04] :

Début

- L'intervalle [a, b] est subdivisé en (b - a) 10^n petits intervalles qui représenteront chacun un chromosome.
- Chaque chromosome parmi les N, taille de la population, est codé en binaire à l'aide de k bits, avec k vérifiant les inéquations suivantes :

$$2^{(k-1)} < (b-a)10^n \leq 2^k \quad \textbf{(3. 1)}$$

- La valeur décimale, x', correspondant au code binaire de chaque chromosome binaire $v_j = (a_{k-1} \ldots a_1 a_0)_j$, est calculée par :

$$x' = \sum_{i=0}^{k-1} a_i 2^i \quad \textbf{(3. 2)}$$

- Le nombre réel, x, correspondant à la valeur binaire est déterminé par :

$$x = a + x'((b-a)/(2^k - 1)) \quad \textbf{(3. 3)}$$

Pour chaque génération les calculs suivants sont effectués :

- Calcul de la fonction d'évaluation $feval(x_j)$ pour chaque chromosome v_j.
- Calcul de l'évaluation totale, F, de la population constituée de N individus :

$$F = \sum_{j=1}^{N} feval(x_j) \quad \textbf{(3. 4)}$$

- Calcul de la probabilité de sélection, p_s, de chaque chromosome :

$$Ps_j = feval(x_j)/F \quad \textbf{(3. 5)}$$

- Calcul de la probabilité cumulative, q, pour chaque chromosome :

$$q_j = p_1 + p_2 + \ldots + p_j \quad \textbf{(3. 6)}$$

- Pour sélectionner à l'aide de la roue de loterie biaisé, on fait tourner la roulette N fois (taille de la population) de la façon suivante : à chaque fois, on génère aléatoirement un nombre r dans l'intervalle [0,1]. Ensuite, on compare ce nombre aux probabilités q_j . Si $r_j < q_1$ alors v_1 est sélectionné, sinon v_j est sélectionné avec j=2:N tel que $q_{j-1} < r_1 < q_j$.
- Pour chaque chromosome de la nouvelle génération, on génère, au hasard, N nombres r dans [0,1] et on les compare à la probabilité de croisement P_c.
  Si $r_i < P_c$ alors le $i^{ème}$ chromosome est sélectionné pour le croisement, sinon il ne l'est pas.
- Les chromosomes ainsi sélectionnés seront croisés deux à deux. Si le nombre de ces chromosomes est impair, on peut soit en élaguer un, soit en reprendre un autre, afin d'introduire une assez grande diversification dans la population. L'algorithme implémenté prend en compte les deux possibilités

- On mute un bit de l'ensemble des gènes des différents chromosomes ; constituant la nouvelle population obtenue après le croisement ; si le nombre généré arbitrairement r est inférieur à la probabilité de mutation P_m.
- Si le critère d'arrêt n'est pas encore atteint, la population obtenue après la mutation sera considérée comme étant la population initiale et le processus sera réitéré.

Fin

Algorithme 3. 2- Algorithme génétique implémenté

Après avoir détaillé les différentes instructions de l'algorithme implémenté, on propose initialement son application à des fonctions mathématiques.

3.2.2 Application des AGs aux fonctions mathématiques

Dans un premier temps, l'approche génétique va être appliquée à des exemples simples, dont la solution analytique est connue, et ceci dans le but de simplifier la compréhension de l'implémentation de cette approche et par conséquent déclarée sa validité.

La méthode d'optimisation par algorithmes génétiques, requiert la définition de plusieurs paramètres d'une importance majeure pour aboutir à de bons résultats. Il s'agit du nombre d'itérations et la dimension de la population d'une part et des probabilités de croisement et de mutation d'autre part.

Il faut donc décider des types de sélection, de croisement et de mutation ainsi que leurs taux. La littérature dans ce domaine donne une idée de l'ordre de grandeur de ces taux (précités dans le chapitre 2). Le type de chaque opérateur ainsi que son taux sera discuté dans le chapitre quatre.

Exemple 1 : Fonction mono variable

Appliquons étape par étape cet algorithme de base à un problème d'optimisation où l'objectif est de maximiser la fonction d'adéquation suivante : $F(x) = -x^2 + 4x$

Analytiquement, cette fonction admet un maximum ***$F_{max}(x_0) = 4$ pour $x_0=2$.***

Cas 1 : Un nombre de chromosomes = 30 et un nombre d'itérations = 70

Dans ce premier cas (avec la fonction mono variable), la recherche de l'extremum par l'algorithme génétique, sur l'intervalle $[0,3]$ attribué arbitrairement, a donné les résultats numériques et graphiques ci-dessous :

Fmax$_{AG}$ = 3.99957922419461

X$_{AG}$ = 1.97948717948718

La Figure 3.1 donne une indication sur l'éparpillement des valeurs de la variable x au voisinage de la valeur obtenue analytiquement, c'est à dire (x=2).

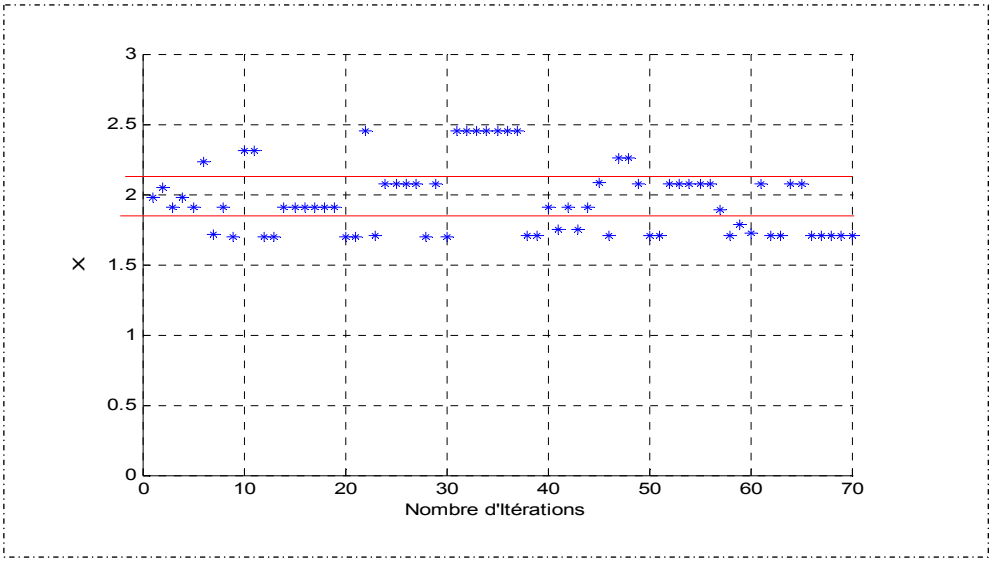


Figure 3. 1- Evolution de x en fonction du nombre d'itérations

Dans le but de voir de plus près la dispersion des valeurs de la variable x, un zoom de la Figure 3.1 a été effectué et représenté par la Figure 3.2.

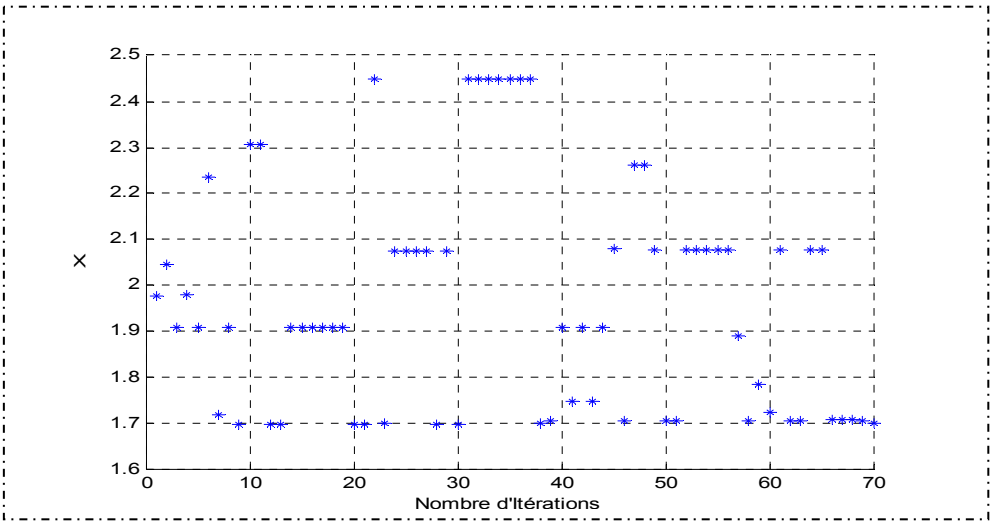


Figure 3. 2- Zoom de la Figure 3.1

Pour cet exemple, l'erreur obtenue est illustrée par la Figure 3.3. Celle-ci identifie son aspect qui est décroissant (mis à part quelques fluctuations) avec l'augmentation du nombre d'itérations. L'expression de l'erreur calculée est (Fmax –Fmoy) / Fmax [Bel98]. Cette erreur est comparée avec epsilon (ε), une très petite constante introduite pour prévenir l'indétermination de la fonction à chaque fois qu'elle prend des valeurs très petites [Bel98].

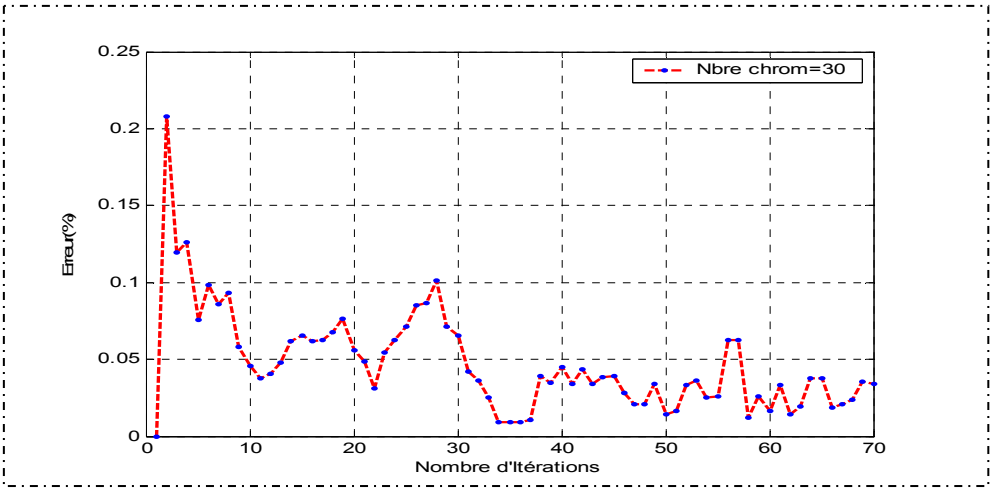


Figure 3. 3- Evolution de l'erreur avec le nombre d'itérations

Cas 2 : Un nombre de chromosomes = 120 et un nombre d'itérations = 70

On a accru le nombre de chromosomes à 120 avec le même nombre d'itérations (70), on a obtenu :

Fmax$_{AG}$ = 3.99999946329617

X$_{AG}$ = 2.00073260073260

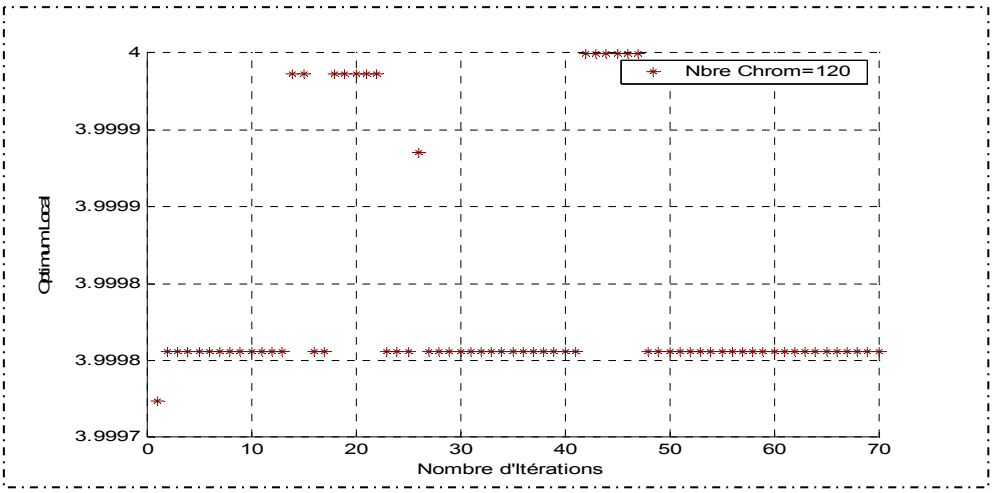


Figure 3. 4- Evolution de l'optimum global avec la variation du nombre d'itérations

Ces résultats traduisent une amélioration par rapport au premier cas. On remarque par le biais de la Figure 3.4 que le résultat se rapproche plus de la valeur exacte. Cette amélioration s'explique par l'augmentation du paramètre "nombre de chromosomes".
De plus, on constate que l'erreur s'affaiblie de plus en plus (avec quelques fluctuations) Figure 3.5.

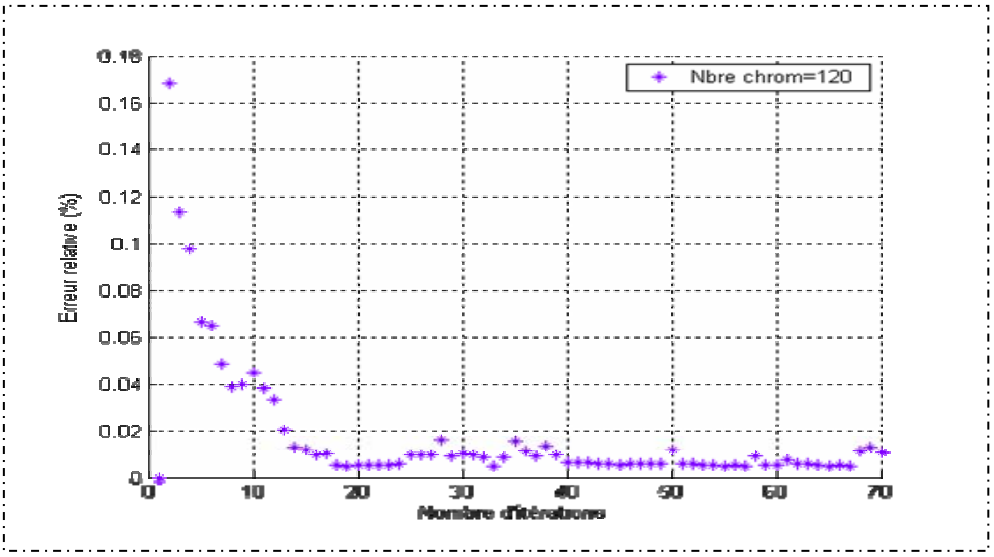


Figure 3. 5- Erreur en fonction du nombre de chromosomes

La Figure 3.6, décrit une comparaison de l'erreur produite pour les deux cas (30 chromosomes et 120 chromosomes) où on constate que l'erreur obtenue dans le 2ème cas est inférieure à celle obtenue dans le cas 1.

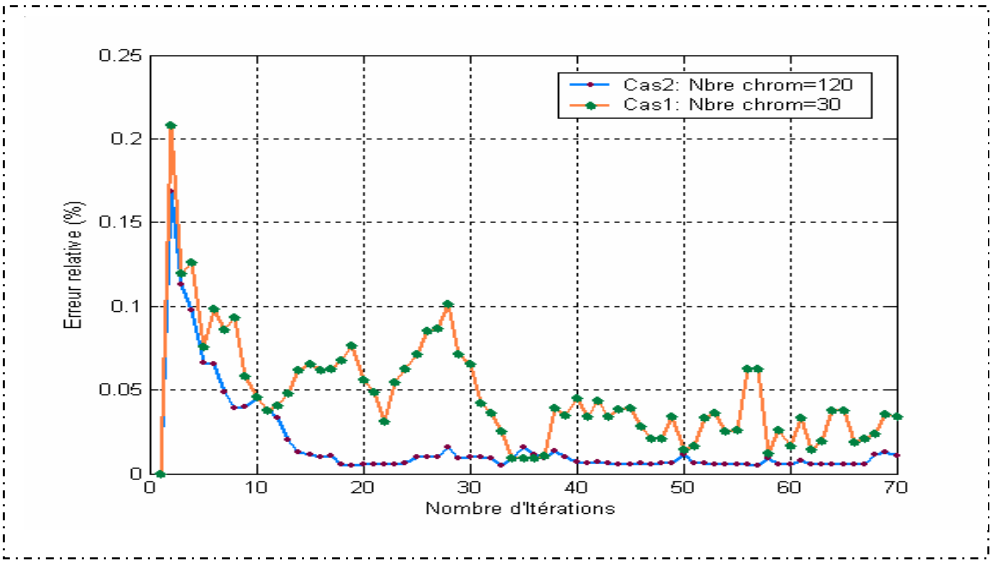


Figure 3. 6- Evolution de l'erreur dans les deux cas considérés

On a présenté deux cas distincts dont les résultats approuvent que le choix de la valeur du paramètre nombre de chromosomes influence à la fois l'optimum et l'erreur.

❖ **Une autre exécution du deuxième cas**

Une autre exécution du programme implémentant l'AG en se servant des mêmes données en entrée, a fourni les résultats graphiques et numériques suivants : $Fmax_{AG}$ = 4 pour X_{AG} = 2.

Rappelons que le résultat analytique de la fonction $F(x) = -x^2 + 4x$ est $Fmax(x_0) = 4$ pour $x_0=2$.

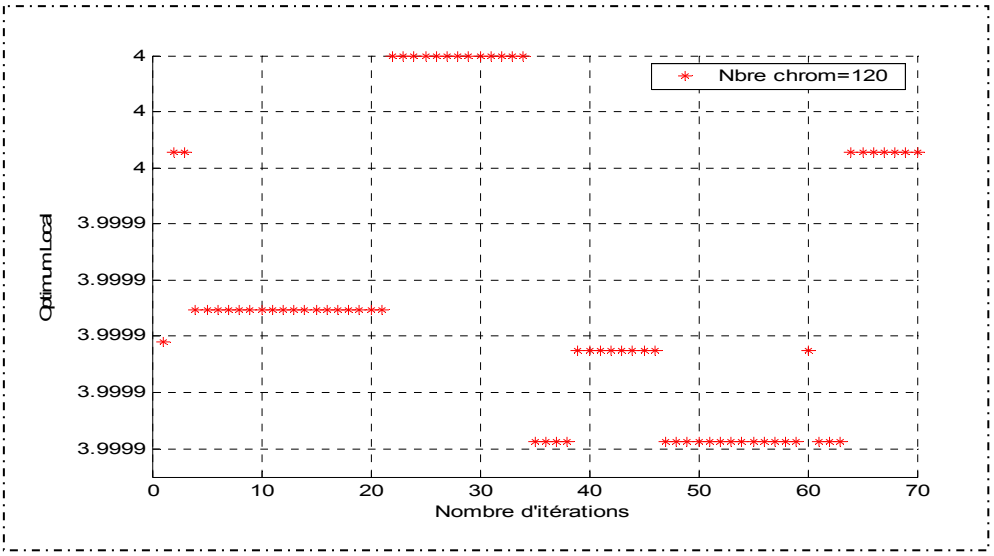


Figure 3. 7- Evolution de l'optimum global en fonction du nombre d'itérations

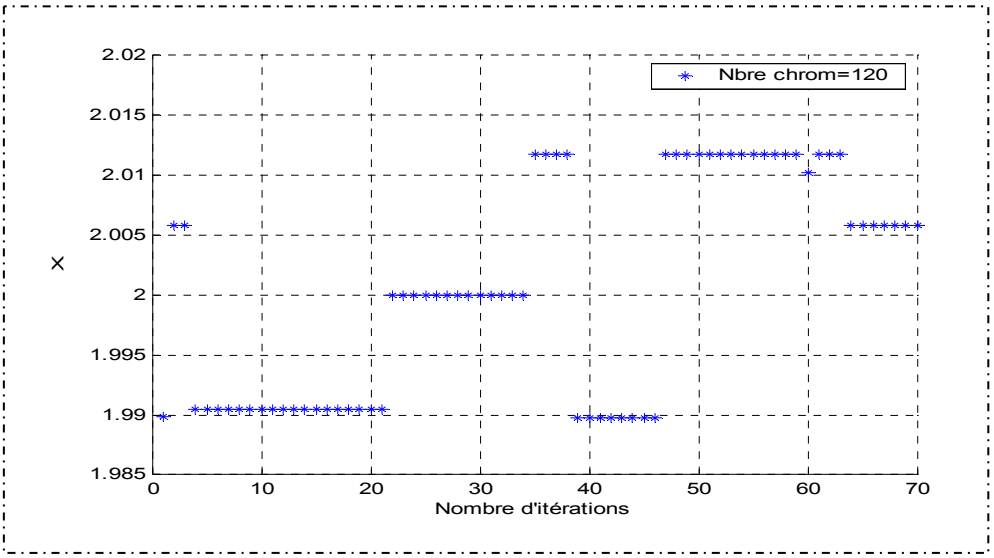


Figure 3. 8- Evolution de x en fonction du nombre d'itérations

Comme le montre la Figure 3.7, parmi les optima identifiés, l'algorithme a donné un optimum de valeur 4. A ce dernier correspond un x de valeur 2, décrit par la Figure 3.8.

Cas 3 : Nombre d'itérations = 50 avec une taille variable de population

Tailles population	Optimums	Valeurs	Erreurs relatives (%)	Temps d'exécution
30	3.99977065583125	2.01514411333659	0.01686075083240	≈ 2 secs
50	3.99998830607256	1.99658036150464	0.01159966136466	≈ 3 secs
100	3.99999976134842	1.99951148021495	0.00853357484875	≈ 4 secs

Tableau 3. 1- Echantillon des résultats du 3ème cas

On constate à partir du Tableau 3.1, que plus la taille de la population est élevée plus on se rapproche du meilleur résultat d'une part, d'autre part l'erreur se réduit. Autrement dit, les performances des populations augmentent avec leurs tailles.

Pour les résultats graphiques de ce 3ème cas, la Figure 3.9 (entre autre le Tableau 3.1) présente l'optimum global, en montrant qu'on se rapproche de plus en plus de l'optimal global par augmentation du nombre de chromosomes.

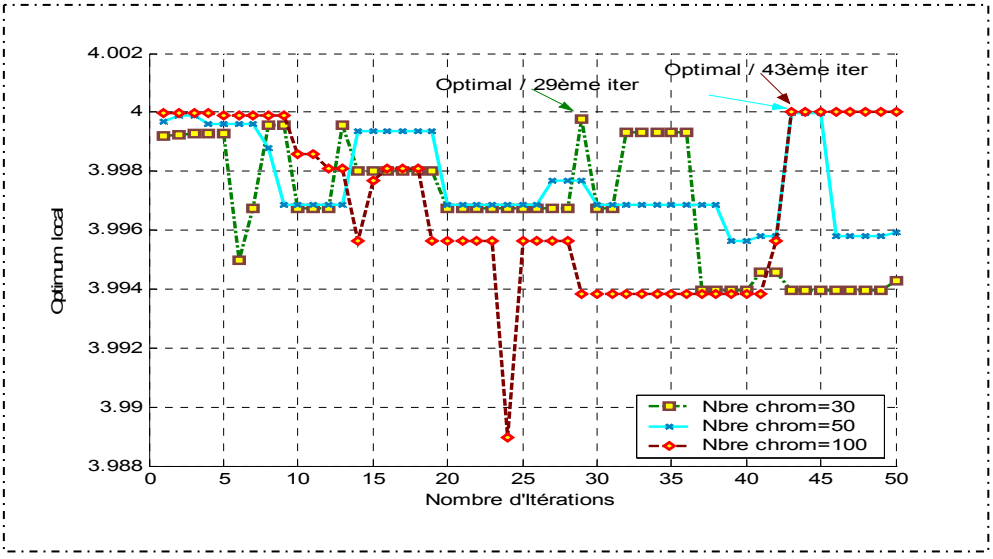


Figure 3. 9- Optimum global en fonction du nombre d'itérations

Aussi, le choix d'une taille de population plus grande permet de diminuer l'erreur de plus en plus. Ce constat est illustré par la Figure 3.10.

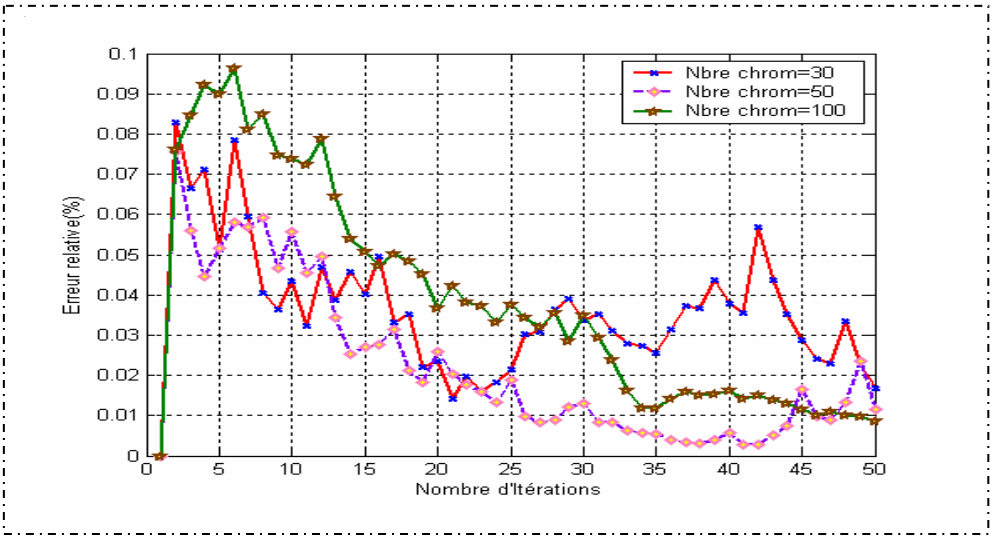


Figure 3. 10- Erreur en fonction du nombre d'évaluation du 3ème cas

Dans cette série d'expériences, l'objectif visé est la recherche de la meilleure manière de faire évoluer une population de réels considérés comme des individus à partir de la méthode d'optimisation par les AGs. En fait, d'après les résultats obtenus par les trois cas (cas1, cas2 et cas3) du premier exemple, on conclue que les paramètres pris en compte (dans ce contexte, on peut varier différents paramètres à savoir : la taille de la population, le nombre d'itérations, les probabilités de croisement et de mutation, le domaine de définition) jouent un rôle primordial au niveau des résultats. En effet le choix des bons paramètres, aboutit aux meilleurs résultats d'optimisation en un temps réduit. Toutefois, ce choix de paramètres n'est pas un standard, il est propre à chaque problème.

Exemple 2 : Fonction à deux variables

Il s'agit cette fois ci d'une fonction à deux variables : L'objectif est de maximiser la fonction d'adéquation suivante :

$$F(x_1,x_2) = \sin \pi x_1 + 2\sin 2\pi x_2$$

Analytiquement, cette fonction admet un maximum $Fmax(x_1,x_2,) = 3$ pour $x_1=1/2$ et $x_2=1/4$.

L'exécution du programme implémentant l'algorithme génétique a pu aboutir aux résultats suivants :

Cas 1 : Un nombre de chromosomes = 80 et un nombre d'itérations = 60

Dans ce second exemple (fonction à deux variables) la recherche de l'extremum sur les intervalles [0 ,1] et [0,0.5] respectivement par l'algorithme génétique a fournie les résultats numériques et graphiques ci-après :

$Fmax_{AG} = 2.99993278765774$

$X_{1AG} = 0.50342130987292$

$X_{2AG} = 0.25048923679061$

La Figure 3.11 représente le maximum de la fonction sinus ($F(x_1,x_2)= \sin \pi x_1 + 2\sin 2\pi x_2$) ainsi que x_1 et x_2 correspondants.

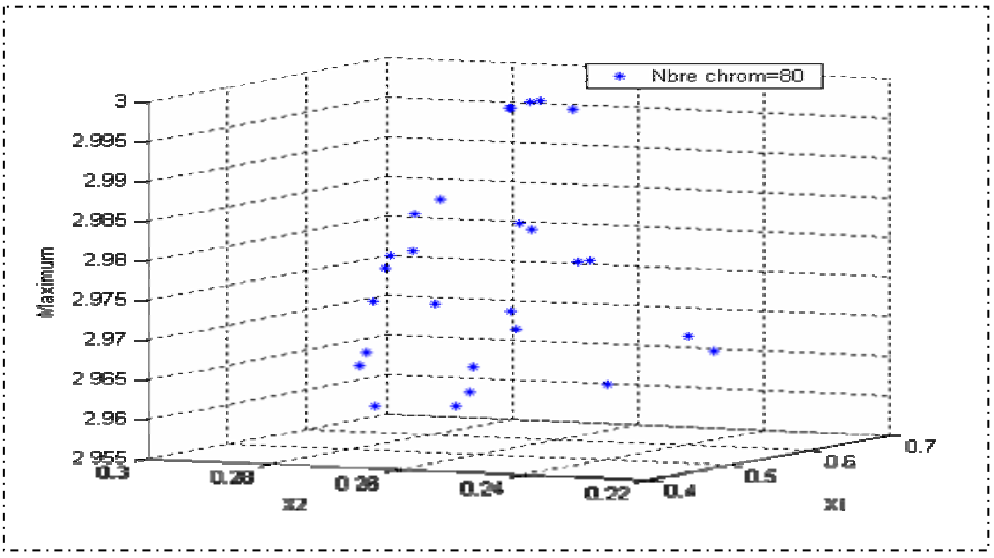


Figure 3. 11- Evolution du maximum en fonction de x_1 & x_2

Cas 2 : **Nombre itérations = 50 et un nombre variable de chromosomes**

Tailles population	Optimums	Valeurs		Erreurs relatives (%)	Temps d'exécution
		X_1	X_2		
40	2.97847240626664	0.44259892525647	0.26160234489497	0.03328799996526	≈ 4 secs
100	2.98797768392128	0.45334636052760	0.25574010747435	0.02899563145116	≈ 8 secs
150	2.99813420528643	0.48070346849047	0.24914509037616	0.01657315613765	≈ 15 secs

Tableau 3. 2- Echantillon des résultats de la fonction $\sin \pi x_1 + 2\sin 2\pi x_2$

Dans ce second cas, on a pris différents nombres de chromosomes (tailles de la population) indiqués sur le tableau 3.2 qui ont servi de paramètres à l'algorithme génétique.

Là aussi les résultats présentés dans le tableau 3.2, montrent qu'on se rapproche de plus en plus du meilleur résultat avec l'augmentation de la taille de la population qui permet aussi d'avoir une erreur de plus en plus petite.

Le résultat graphique de cet exemple relatif à l'erreur est donné par la Figure 3.12.

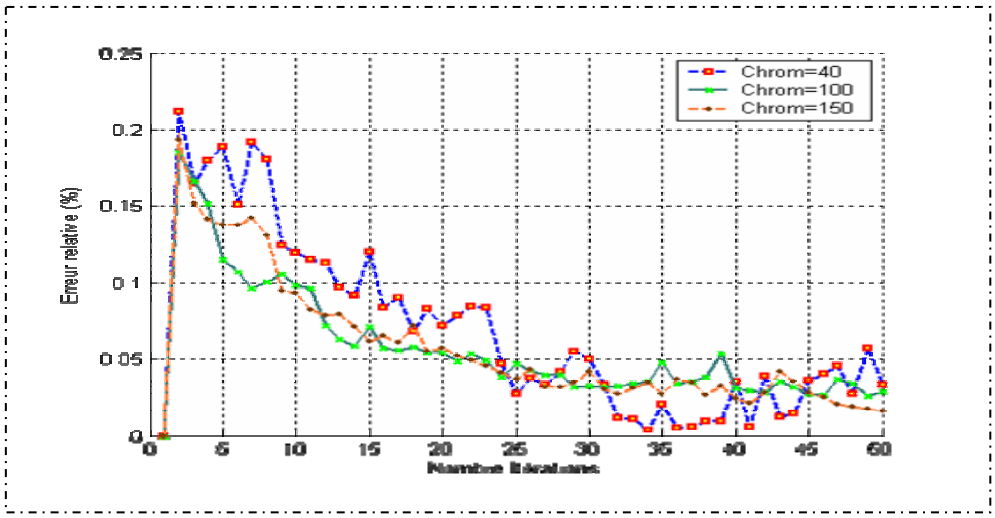


Figure 3. 12- Evolution de l'erreur en fonction du nombre d'itérations pour différentes populations de la fonction sinus

3.3 Résultats

Les résultats présentés par le tableau 3.1 et les Figures de 3.1 à 3.10 sont ceux obtenus par l'algorithme génétique appliqué initialement à une fonction mathématique mono variable. Ces résultats sont vraiment approche du résultat analytique, par conséquent ces résultats sont acceptables. Vu la simplicité de cette fonction et pour confirmer la validité de l'AG implémenté, une deuxième fonction mathématique a été suggérée mais cette fois-ci à deux variables. Les résultats de l'application des AGs à cette dernière sont donnés par le tableau 3.2 et les graphes présentés par les Figures 3.11 et 3.12. Ils ont permis de remarquer que l'augmentation de la taille de la population offre d'une part plus de convergence de la solution et d'autre part réduit le taux de l'erreur. Là aussi, l'algorithme a donc fourni des résultats proches de la solution analytique.

La synthèse des résultats numériques et graphiques obtenus lors de la maximisation des deux fonctions mathématiques présentées en se servant de la méthode d'optimisation par les AGs, a permis de mettre l'accent sur les constats suivants :

- L'algorithme détermine les maximums des fonctions analytiques avec précision.
- La recherche adoptée par les AGs est entièrement aléatoire.
- La recherche de la solution n'est pas affectée par le choix de la solution initiale et admet des domaines de recherche assez larges.
- La précision des résultats dépend des paramètres manipulés par l'algorithme.
- La majorité des solutions potentielles se concentre autour de la valeur exacte.
- Le temps d'exécution est non considérable.

En résumé, les résultats des deux exemples mathématiques ont validé la capacité des AGs à la localisation des optima des fonctions à une seule variable ou à variables multiples. En effet, cette application a permis l'implémentation du programme de la méthode d'optimisation par algorithmes génétiques.

3.4 Conclusion

Dans ce chapitre, on a présenté et expliqué la méthode d'optimisation par algorithmes génétiques à travers deux fonctions mathématiques avant de se pencher et

l'appliquer au problème d'optimisation du coût et ceci dans le but de valider l'algorithme implémenté.

Le comportement de l'algorithme génétique envers ces deux fonctions mathématiques (et pour d'autres tester sans être exposées) s'avère particulièrement efficace. Ceci peut s'expliquer par plusieurs faits. Tout d'abord, l'optimum identifié est trop proche de la réalité (résultat analytique). Ensuite on remarque que le temps d'exécution est loin d'être considérable. Par définition, la qualité d'un algorithme dépend de son temps de calcul ainsi que dans le cas d'une solution approchée, de la qualité de la solution fournie. Comme il est souvent difficile de trouver la meilleure solution possible (solution optimale), on se contente souvent de chercher une solution approchée [Reb99]. Effectivement, les solutions trouvées répondent aux critères d'un algorithme de bonne qualité.

Pour conclure, les résultats obtenus par l'algorithme génétique sont probants et permettent alors d'en prendre appui pour proposer cette méthode d'optimisation au chapitre suivant.
Dans une première étape, ce chapitre explore la voie de recherche de l'approche génétique au type de problème fréquemment rencontré au milieu industriel à savoir *la gestion des stocks*. Celle-ci se rattache à la recherche opérationnelle. Du célèbre modèle de Wilson aux modèles les plus sophistiqués, tous relèvent d'une logique d'optimisation appuyée sur une formalisation mathématique [Cra03].
Dans une deuxième étape, le procédé d'optimisation par AGs pour identifier le niveau du stock conduisant à un coût d'exploitation minimal sera exposé. Cet objectif, étant le centre d'intérêt de la gestion de production moderne, qui quelle qu'elle soit, tend vers des stocks aussi faibles que possible conduisant à un coût de stockage minimum [Dew03], l'a rendu aussi important pour les gestionnaires dans le monde industriel. En effet, les entreprises industrielles, face à la concurrence, sont de plus en plus confrontées aux impératifs de la modernisation de la qualité des produits et de la réduction de leurs coûts.

Chapitre 4

Optimisation du Coût d'un Stock de Pièces de Rechange par AG

Résumé : Ce chapitre introduit des notions relatives à la gestion des stocks et présente l'application de l'algorithme génétique à ce domaine qui préoccupe énormément les gestionnaires dans le monde industriel. En effet, les entreprises industrielles, face à la concurrence, sont de plus en plus confrontées aux impératifs de la modernisation des appareils productifs, de la qualité des produits et de la réduction de leurs coûts. Elles accordent de plus en plus une forte importance à la fiabilité de ces systèmes. Ainsi, un arrêt d'un équipement de production, peut engendrer des coûts énormes à l'entreprise. Il est alors primordial de mettre l'accent sur les problèmes de la fiabilité et de la disponibilité. En particulier, on s'intéresse à la disponibilité d'une quantité de pièces de rechange jugée suffisante en stock : c'est la fonction de la gestion des stocks.

A travers ce chapitre, on présente alors l'application de l'algorithme génétique implémenté pour résoudre le problème de gestion des stocks dans le but d'optimiser le coût d'un stock de pièces de rechange ainsi que les résultats obtenus.

4.1 Introduction

La gestion des stocks est un des aspects opérationnels les plus importants des décisions de production. Elle consiste à assurer à tout moment et aux meilleures conditions la conservation et la mise à disposition de l'utilisateur des matières ou marchandises dont il a besoin [Boy01].

L'académie des sciences commerciales a défini la gestion des stocks comme: "Ensemble des activités se rapportant à la constitution, à la connaissance, à l'entretien et à la liquidation éventuelle des stocks destinés à satisfaire, dans les conditions les plus économiques, les besoins à plus ou moins long terme de la production et de la vente" [Boi87].

Un stock est donc constitué pour satisfaire une demande future. En cas de demande aléatoire, il peut y avoir non coïncidence entre la demande et le stock. Deux cas sont évidemment possibles :

- une demande supérieure au stock : on parle de *rupture de stock* ;
- une demande inférieure au stock : on aura alors un *stock résiduel*.

Le présent travail est limité à l'étude de la gestion des stocks liée à la fonction maintenance. Celle-ci mérite d'être remplie au moindre coût et avec efficacité, vu l'évolution continuelle de la complexité des systèmes qui nécessitent l'utilisation des approches répondant à ces critères. L'intérêt de la fonction maintenance engendre celui des pièces de rechange.

Pour le service de maintenance, la gestion des pièces de rechange est très importante pour les raisons suivantes [Boi87] :

- les sorties sont beaucoup plus aléatoires que pour les consommables ;
- les stocks de chaque pièce sont très faibles donc présentent un risque plus grand de rupture ;
- la rupture de stock est souvent plus lourde de conséquences que celle d'un consommable (arrêt de l'équipement).

Compte tenu de cette importance, cette voie de recherche a été empruntée. Pour ce faire, ce dernier chapitre commence par une revue rapide des concepts de la gestion des stocks, continue par une description du problème et se termine par une présentation et interprétation des résultats.

4.2 Justification des stocks

Les stocks sont des ressources matérielles qui ont une valeur économique et sont inutilisées ou en attente d'utilisation. Les entreprises classent souvent leurs stocks en plusieurs types (matières premières, en-cours de fabrication, produits finis et pièces de rechange). Tous les stocks représentent un investissement dont le but est de faciliter la production et le service des clients. Néanmoins, avoir des stocks consomme des fonds qui ne rapportent pas comme s'ils étaient investis et pourraient être nécessaire d'urgence ailleurs [Mon93].

Pour autant, constituer et conserver un stock entraîne des coûts dont la minimisation doit être un objectif important des gestionnaires : c'est le but des modèles de gestion des stocks [Ala01].

4.3 Les modèles de gestion des stocks

Il existe de nombreux modèles de gestion des stocks qui répondent à la multiplicité des situations rencontrées en entreprise. En effet, ils se présentent selon deux modes [Ala01]:

- Modèles de gestion des stocks en avenir certain : Il s'agit principalement du célèbre modèle de Wilson. Toutefois, ces modèles ont un domaine d'application limité aux cas où la demande est régulière. Cette restriction fait que ces modèles ne sont pas adaptés à une gestion de stock moderne.
- Modèles de gestion des stocks en avenir incertain : ces modèles s'utilisent dans des situations probabilisables.

C'est ce second mode qui nous intéresse. Cet intérêt se justifie par l'occurrence des demandes aléatoires. Ce mode est donc traité dans un contexte d'optimisation.

4.4 Présentation du problème

Les méthodes d'optimisation permettent de déterminer plusieurs solutions dans l'espace d'état qui maximisent (ou minimisent) un critère. Dans notre cas, l'espace d'état est constitué des paramètres des pièces de rechange à stocker à savoir taux de panne et taux de réparation. Il s'agit donc d'un problème de gestion des stocks de pièces de rechange spécifiquement les membranes. Ce qui nous amène à préciser que la durée de réparation d'un matériel, et par conséquent sa disponibilité, dépend fortement de

l'existence d'un stock de pièces de rechange [Pag80]. Un cas trop fréquent est considéré, c'est celui d'un matériel entrant dans une chaîne de production dont la défaillance entraîne l'arrêt de la production.

Les hypothèses sur le modèle sont les suivantes [Pag80]:

- Il y a n exemplaires de ce matériel en exploitation dans n chaînes de production (un exemplaire dans chaque chaîne).
- Les pertes dues à l'arrêt de la production d'une chaîne sont proportionnelles au temps d'arrêt ; soit C_p le coût par unité de temps.
- Le taux de défaillance du matériel varie dans un intervalle de confiance.
- Lorsqu'un matériel tombe en panne, si on a en stock une pièce (matériel) de rechange, on considérera comme négligeable le coût et le temps de réparation (échange).
- Le coût d'achat C_a des pièces.
- Les coûts seront actualisés de manière continue au taux d'actualisation i.
- La durée de vie des chaînes de production T.

4.4.1 Modélisation du problème

L'état du parc à chaque instant sera caractérisé par le nombre de matériels en fonctionnement et le nombre de matériel en stock illustré par la Figure 4.1 [Pag80].

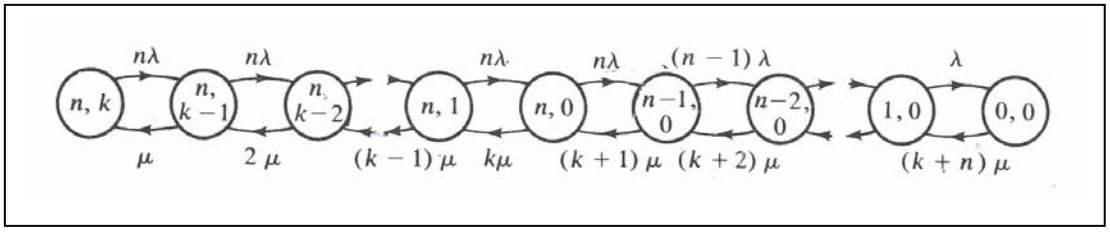


Figure 4. 1- Graphe d'état pour un stock

Où n : le nombre de matériels en fonctionnement ;

k : le nombre de matériels en stock ;

λ : taux de défaillance qui caractérise la défaillance d'un équipement. Il est défini comme étant la probabilité conditionnelle que l'équipement tombe en panne [Kaf01].

μ : taux de réparation d'un équipement. Ce paramètre exprime la probabilité pour qu'une entité, qui a été en panne pendant un temps t, retrouve son aptitude à remplir sa fonction dans l'unité de temps qui suit [Vil88].

Soit $P_{m,l}(t)$ la probabilité de se trouver dans l'état (m,l).

Le coût moyen actualisé d'être dans l'état (n-j, 0) avec j =1:n est :

$$jCp\int_0^T P_{n-j,0}(t)e^{-it}dt \qquad \textbf{(4. 1)}$$

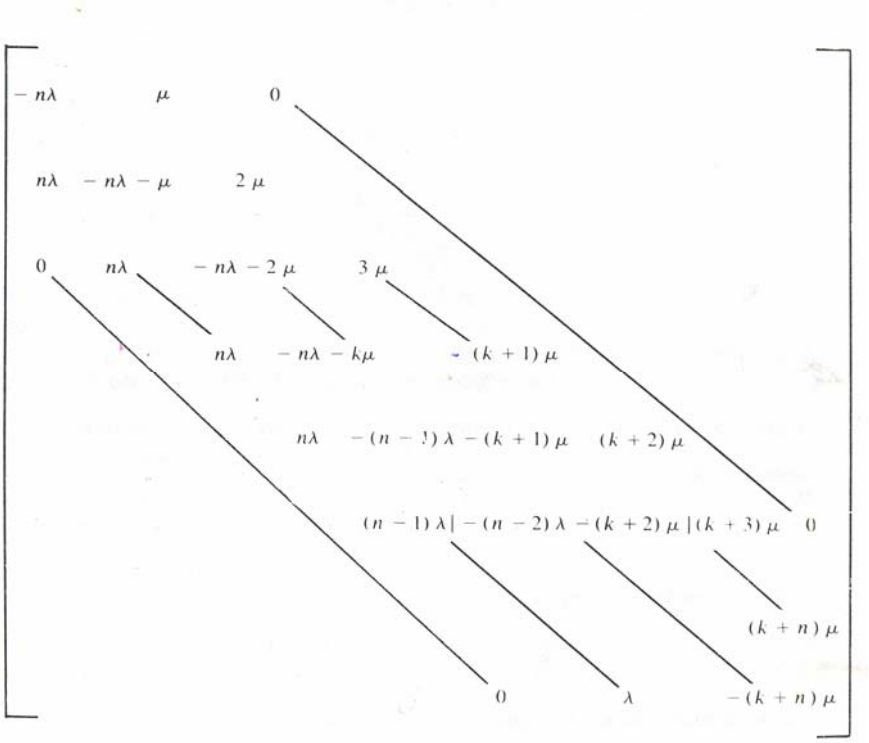
D'où le coût moyen actualisé d'exploitation :

$$C(k) = kCa + Cp\int_0^T \sum_{j=1}^{n} jP_{n-j,0}(t)e^{-it}dt \qquad \textbf{(4. 2)}$$

Le problème est donc ramené au calcul des probabilités, qui sont les solutions du système différentiel suivant [Pag80] :

$$\begin{cases} \dfrac{dP}{dt} = AP \\ P = \begin{bmatrix} P_{n-1,0}(t) \\ P_{n-2,0}(t) \\ . \\ . \\ . \\ P_{1,0}(t) \\ P_{0,0}(t) \end{bmatrix} \\ P_{n,k}(0) = 1 \\ P_{m,j}(0) = 0 \quad \text{si} \quad m+j \neq n+k \end{cases} \qquad \textbf{(4. 3)}$$

Avec

A = 

La résolution du problème passe obligatoirement par deux phases. Il s'agit de résoudre en premier lieu le système d'équations différentielles définit par (4.3). Ensuite, rechercher l'extremum basé sur le modèle (4.2).

4.4.2 Discrétisation réalisée

La solution du système d'équations différentielles décrit par (4.3) requiert en premier lieu l'emploi d'une méthode de discrétisation. Par conséquent la méthode RUNGE KUTTA (R-K) est adoptée. Cette méthode est comptée parmi les méthodes définies par la littérature riche de la discrétisation. L'algorithme R-K d'ordre 4, a été utilisé. Le principe de cette méthode de résolution est détaillé dans l'algorithme 4.1.

Algorithme 4. 1-Méthode de résolution d'équations différentielles : RUNGE KUTTA d'ordre 4

Entrée : w_{ij} une approximation de la $i^{ème}$ composante de la solution du système. Au point t_j , i=1,2,…,m

nc : entier positif choisi (nombre de composantes)

$w_{1,0}$ ……..$w_{m,0}$: conditions initiales

Début

Pour i =1 à nc

$K_{1,i} = hf(t_i , w_i)$;

Pour i =1 à nc

$K_{2,i} = hf(t_i+h/2 , w_i+k_1/2)$;

Pour i =1 à nc

$K_{3,i} = hf(t_i+h/2 , w_i+k_2/2)$;

Pour i =1 à nc

$K_{4,i} = hf(t_{i+1} , w_i+k_3)$

Pour i =1 à nc

$w_{i,,j+1} = w_{i,j} + (K_{1,i} + 2K_{2,i} + 2K_{3,i} + K_{4,i})/6$

Fin

Sortie : le vecteur des approximations w

En second lieu, pour le calcul de l'intégrale, la méthode numérique SIMPSON a été appliquée. Cette technique d'intégration consiste à approcher la valeur de l'intégrale à partir de plusieurs valeurs [Der90]. Ces dernières sont le résultat de la méthode RUNGE KUTTA. Son principe est décrit par la formule suivante :

$$\int_a^b f(x)dx = h/3((y_0 + y_{2m}) + 4\sum_{i=1}^{m} y_{2i-1} + 2\sum_{i=1}^{m-1} y_{2i}) \qquad \textbf{(4. 4)}$$

Dès lors, les résultats obtenus par la méthode RUNGE KUTTA, i.e., les approximations ($w_{i,j}$) seront utilisées par la méthode SIMPSON qui sont formulées par des y (formule 4.4).

Ces deux méthodes (RUNGE KUTTA et SIMPSON) représentent la première phase : la résolution du système d'équations différentielles. En ce qui concerne la seconde phase, son but est la recherche de l'extremum, comme précisé précédemment, basé sur le modèle (4.2). Pour ce faire une méthode d'optimisation est sollicitée.

4.5 Optimisation du coût d'un stock de membranes

Parmi la diversité des applications des algorithmes génétiques qui a connu et connaît encore un grand essor, l'accent est mis en particulier sur le problème de gestion des stocks qui préoccupe les gestionnaires dans le monde industriel depuis longtemps. C'est pratiquement le cas du problème de recherche du niveau du stock conduisant à un coût minimal. Le modèle mathématique a été présenté par le modèle (4.2). En outre, comment avoir un niveau de stock suffisant, des pièces de rechange fréquemment utilisées, telle que les membranes, en conservant un niveau de service suffisant ? La réponse à cette question va dépendre de la nature du stock. Mais dans tous les cas, il faudra agir sur la véritable cause du stock qui peut être [Dew03]:

- mauvaise qualité des prévisions qui entraînent des stocks dormants ou morts ;
- déséquilibre des cadences…

Le niveau du stock dépend naturellement de deux facteurs : les entrées et les sorties. Il sera souvent impossible de jouer sur les sorties (appelées par la production) et la seule façon de réguler le niveau moyen du stock consistera à modifier le mode des entrées [Dew03].

4.5.1 Détermination du coût optimum par algorithme génétique

A travers cette section, la procédure de détermination de l'extremum par algorithme génétique est développée. Elle vise à déterminer aléatoirement les valeurs optimales des paramètres mis en jeu conduisant à l'extremum. La position de cette procédure dans la résolution du problème d'optimisation est fondamentale. A cet effet, elle préconise un intérêt du premier ordre.

4.5.2 Procédure de détermination par algorithme génétique

Il s'agit d'identifier le coût optimal de l'exploitation du stock ainsi que le vecteur optimal des paramètres considérés par le modèle (4.2) à savoir le taux de défaillance, taux de réparation, le niveau du stock. Ce vecteur sera considéré comme étant une solution potentielle, du moment où on traite un problème stochastique qui par définition admet des solutions approchées.

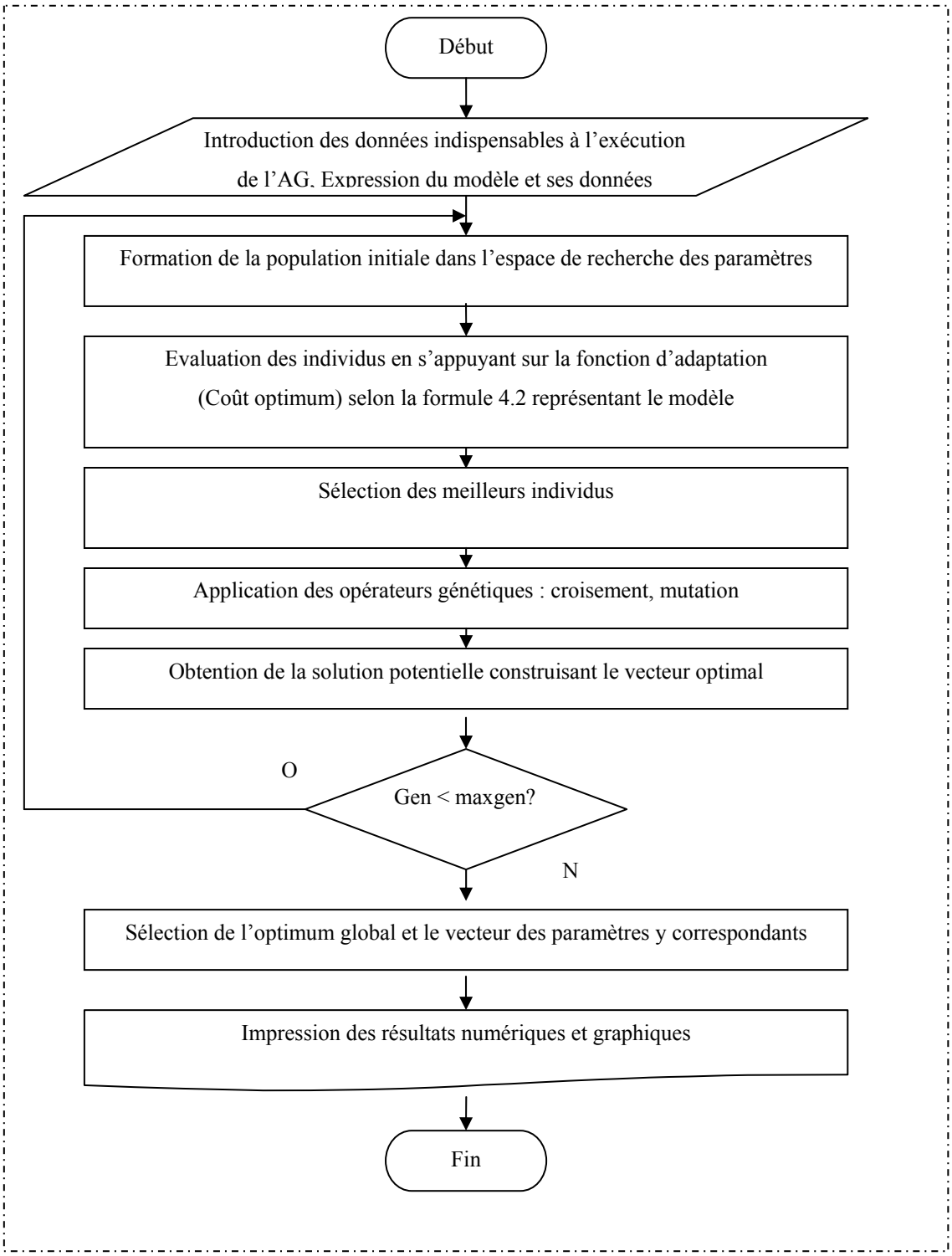
Ainsi, le développement de la procédure d'optimisation exige d'une part le développement d'une procédure de résolution du système d'équations différentielles et d'autre part une procédure de mise en œuvre de l'algorithme génétique durant laquelle les opérateurs de l'évolution naturelle seront principalement appliqués sur les individus suivant la fonction d'adaptation qui permet de les sélectionner. En phase finale, le vecteur optimal des paramètres correspondants à la meilleure solution approchée sera obtenu.

La démarche à suivre est décrite par les étapes résumées par l'organigramme 4.2.

La première étape de cet organigramme consiste à introduire toutes les données vitales à l'exécution. Par la suite un traitement répétitif s'opère de la manière suivante :

- formation de la population initiale sur l'entièreté de l'espace de recherche des paramètres relatifs au problème;
- application des opérateurs de reproduction, de croisement et de mutation afin d'exploiter au mieux les caractéristiques de la population courante et d'obtenir une amélioration en terme de qualité des solutions de l'ensemble de la population ce qui crée une nouvelle population;
- sélection de l'individu de bonne qualité représentant ainsi une solution potentielle et construisant un vecteur des optima locaux ;

Dès lors, ce traitement se répète autant de fois jusqu'à ce que le critère d'arrêt soit satisfait. L'arrêt du processus est essentiel du point de vue pratique. Si l'on a peu ou pas d'information sur la valeur cible de l'optimum recherché, il est délicat de savoir quand arrêter l'évolution. C'est le cas du problème traité. Il paraît alors judicieux d'utiliser la politique la plus courante et la plus simple qui consiste à effectuer un nombre prédéfini d'itérations. A la fin du traitement, la solution finale du problème sera sélectionnée à partir du vecteur des optima locaux ainsi que le vecteur des valeurs y correspondantes.



Organigramme 4. 2- La Recherche du coût optimum par l'AG

4.5.3 Résolution par algorithme génétique

La procédure d'optimisation est implémentée suivant l'organigramme 4.2, ce dernier est tracé à partir de celui présenté par l'organigramme 3.1.

L'organigramme démarre avec l'acquisition des différentes données relatives aux paramètres considérés par le problème qui sont principalement :

- taux de défaillance λ des membranes définit sur l'intervalle $\left[640.10^{-4},900.10^{-4}\right]$;
- taux de réparation μ définit sur l'intervalle $[0.2,0.9]$ [Sma96] ;
- coût d'achat unitaire Ca = 1730 DA;
- coût de perte Cp = MTTR × quantité perdue × prix unitaire

  Où

  - MTTR (Mean Time To Repair) = $\frac{1}{\mu}$ [Pag80].
  - Quantité perdue (pendant l'absence de la production) = 12000 bouteilles.
  - Prix unitaire = 19 DA.

La mise en œuvre d'un algorithme génétique implique la mise au point de plusieurs opérateurs. Dans un premier temps, il convient de choisir un codage par conséquent le codage binaire est adopté. GOLDBERG a proposé d'appliquer autant que faire se peut un alphabet de cardinalité minimale lors de la recherche d'un codage. Il note que l'alphabet binaire a été et restera utile pour une large variété de problèmes à résoudre [Gol94].

Un chromosome est donc représenté par une chaîne de bits identifiant les paramètres du problème. Le procédé suivi est celui décrit dans la section 3.2 du chapitre trois.
Le codage ainsi défini, sollicite une autre étape; celle de la détermination de la population initiale qui doit être la plus diverse possible. Cette population est générée aléatoirement sur tout le domaine de définition de chaque paramètre mis en jeu. Pour le problème traité, une population de 100 individus a paru un bon compromis entre temps de calcul et les résultats obtenus.

L'aspect aléatoire permet de visiter plusieurs zones de l'espace de recherche et par conséquent augmenter la probabilité de trouver la solution optimale du problème.

L'application des opérateurs génétiques (la sélection, le croisement et la mutation), pour l'évolution des populations et la génération de nouvelles solutions, est une autre étape. Pour la mise au point de notre algorithme génétique, la sélection de la Roulette, définie dans le chapitre deux a été adoptée. Cette sélection basée sur le principe de la roue de loterie, favorise les individus forts.

L'opérateur de croisement qui permet l'exploration de l'espace de recherche en créant de nouveaux individus est essentiel pour la convergence d'un algorithme génétique [Har03]. A cet effet le croisement des solutions du problème de gestion des stocks nécessite une adaptation au type de codage utilisé ce qui entraîne l'emploi automatique du croisement binaire et spécifiquement le croisement binaire en 1-point avec une probabilité de 0.75. Ce choix semble judicieux vu le type du codage adopté.

Afin d'introduire des petites modifications à certaines solutions de chaque population, la mutation est considérée dans le processus de recherche génétique avec une faible probabilité soit 0.01. Elle permet d'altérer les gènes du chromosome. Du moment où le codage employé est binaire, la mutation doit l'être aussi.

Pour mettre fin à l'exécution de l'algorithme génétique, un critère d'arrêt est indispensable. Une stratégie couramment employée qui consiste à stopper l'algorithme dès qu'un nombre maximal d'itérations est atteint a été adoptée. Il en résulte alors un vecteur des optima locaux ainsi que celui des valeurs des paramètres qui ont conduit à chaque solution optimale. Une dernière étape doit permettre de sélectionner la solution finale depuis les vecteurs ainsi crées.

4.6 Présentation des résultats

La procédure décrite précédemment a été donc concrétisée et les résultats recherchés ont été identifiés par l'algorithme génétique. L'ensemble des programmes de ce dernier a été développé sous l'environnement Matlab.

Après plusieurs exécutions de l'algorithme avec différentes valeurs, les résultats obtenus sont représentés sur la figure 4.2.

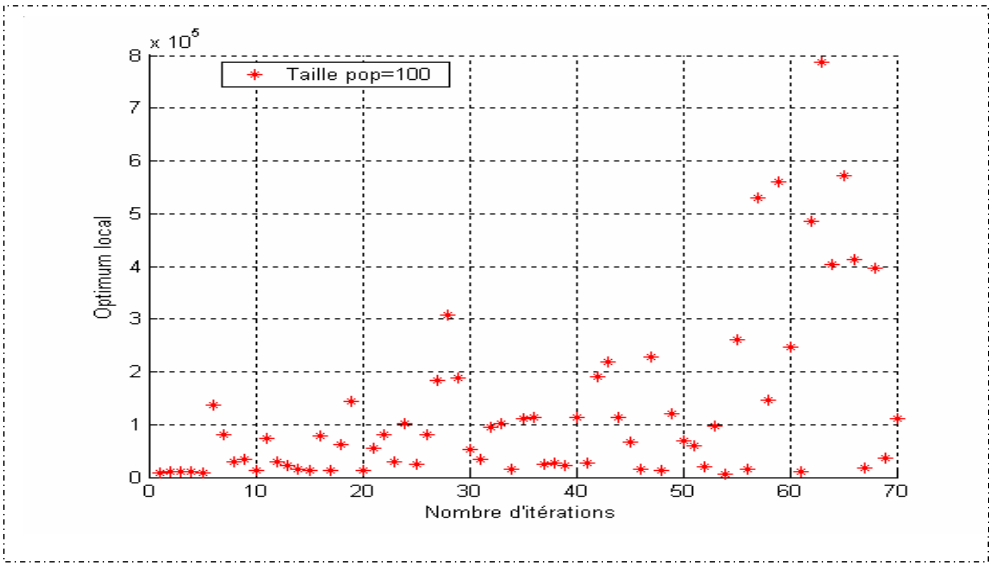


Figure 4. 2- Evolution de l'optimum en fonction du nombre d'itérations

La Figure 4.2 décrit l'évolution de l'optimum en fonction du nombre d'itérations. Celle-ci est tracée à partir du vecteur des optima locaux résultants des différentes générations. Ayant pour but d'améliorer de plus en plus les résultats, une méthode spécifique a été employée. Celle-ci est présentée par la section suivante.

4.6.1 La Méthode élitiste

La méthode élitiste consiste à maintenir une sorte de population archive qui contiendra les meilleurs solutions rencontrées au long de la recherche. Cette population participera aux étapes du cycle de l'algorithme génétique. La population élitiste peut être initialisée comme elle peut être construite lors de la recherche [Har03].
Pour l'AG implémenté, la population élitiste se construit lors de la recherche.

La Figure 4.3 décrit toujours l'évolution de l'optimum en fonction du nombre d'itérations sauf qu'elle est tracée en utilisant la méthode élitiste (conservation du meilleur optimum parmi la génération (i) et celle de (i-1)). Pour calculer la qualité d'un individu de l'espace de recherche, on utilise la fonction d'évaluation. L'évaluation d'un individu ne dépend pas de celle des autres individus, le résultat fournit par la fonction d'évaluation va permettre de sélectionner ou de refuser un individu pour ne garder que les individus ayant la meilleure qualité en fonction de la population courante. Cette

méthode permet de s'assurer que les individus performants seront conservés, alors que les individus peu adaptés seront progressivement éliminés de la population.

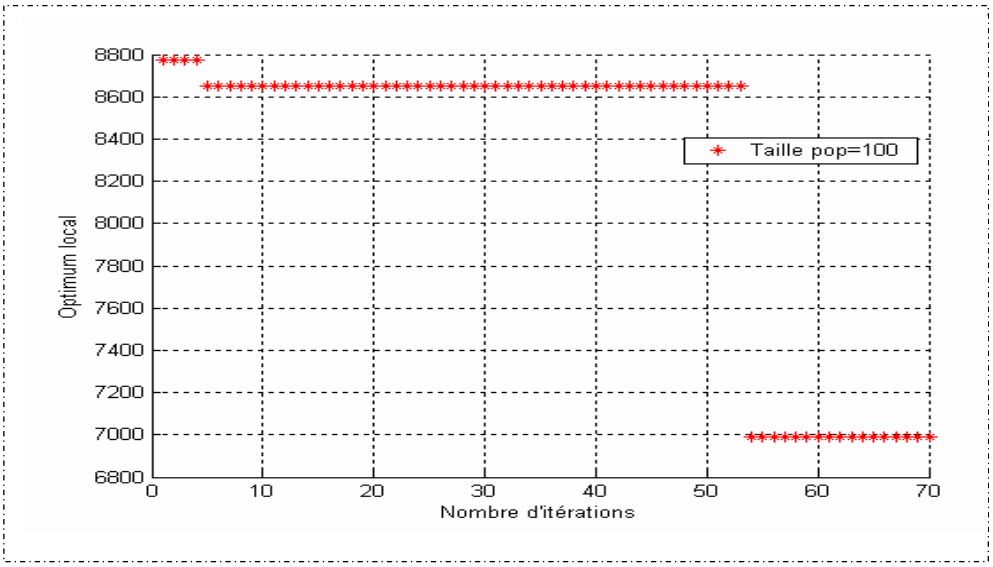


Figure 4. 3- Evolution de l'optimum avec conservation de la meilleure solution

L'évolution de chaque paramètre mis en jeu illustre la signification et l'importance de l'approche génétique en termes de nombre d'itérations, comme le montrent les Figures de 4.4 à 4.9. Donc les Figures suivantes tracent l'évolution des paramètres en fonction du nombre d'itérations. Une première fois, sans conservation ou mémorisation de la meilleure solution et une seconde fois avec mémorisation. Cette stratégie de mémorisation est adaptée dans le but d'aboutir aux meilleurs résultats.

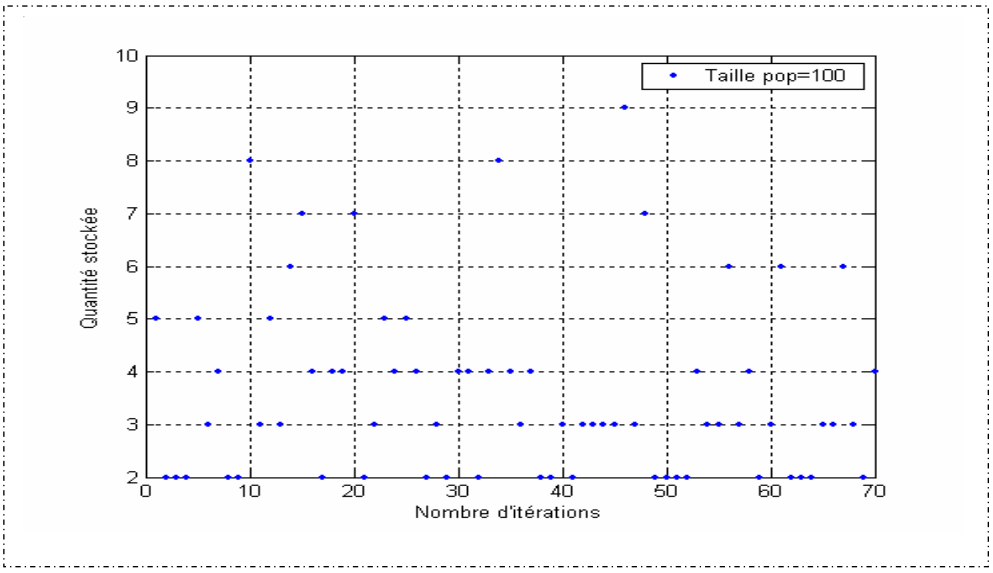


Figure 4. 4- Evolution de la quantité à stocker en fonction du nombre d'itérations

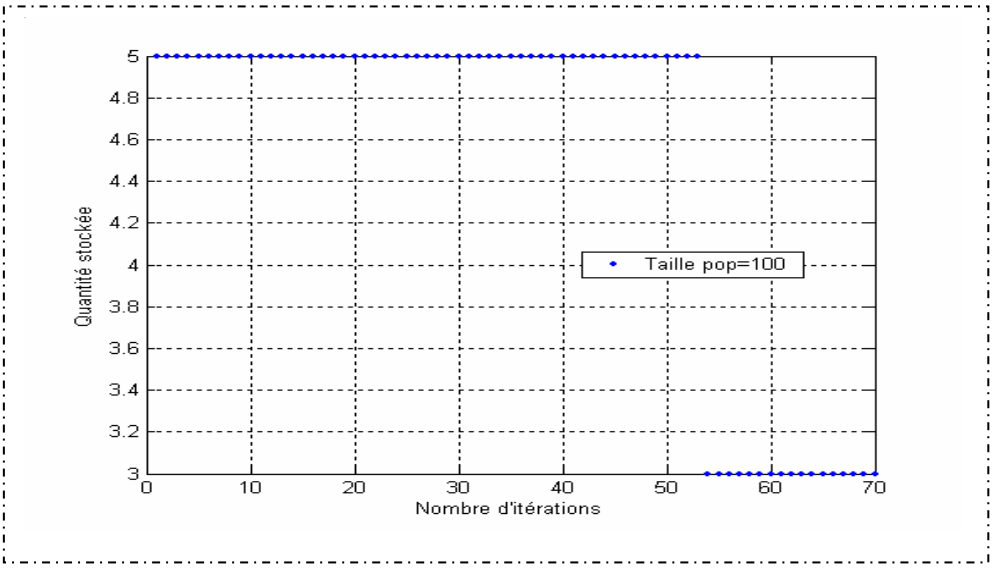


Figure 4. 5- Evolution de la quantité à stocker avec mémorisation de la meilleure solution

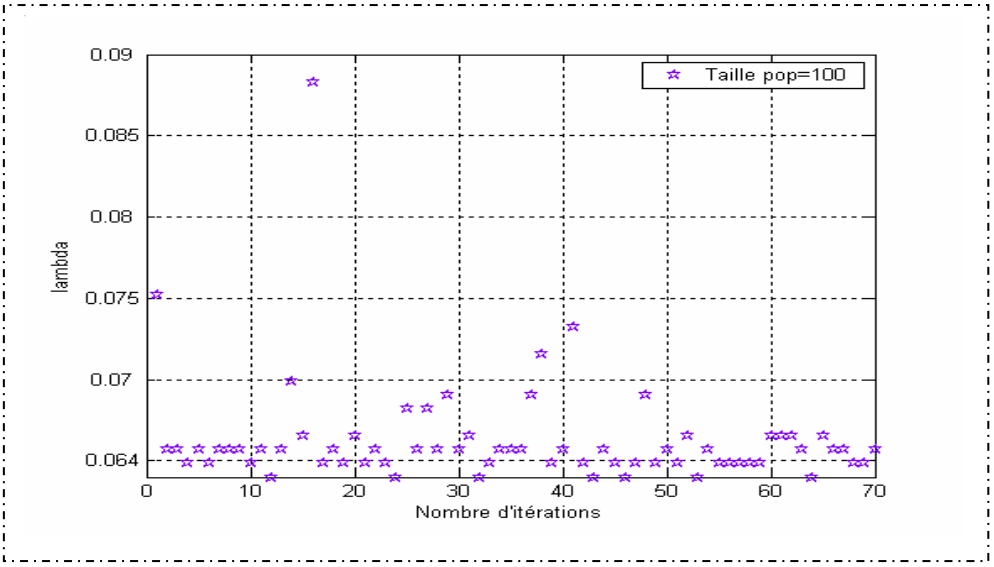


Figure 4. 6- Evolution du taux de défaillance en fonction du nombre d'itérations

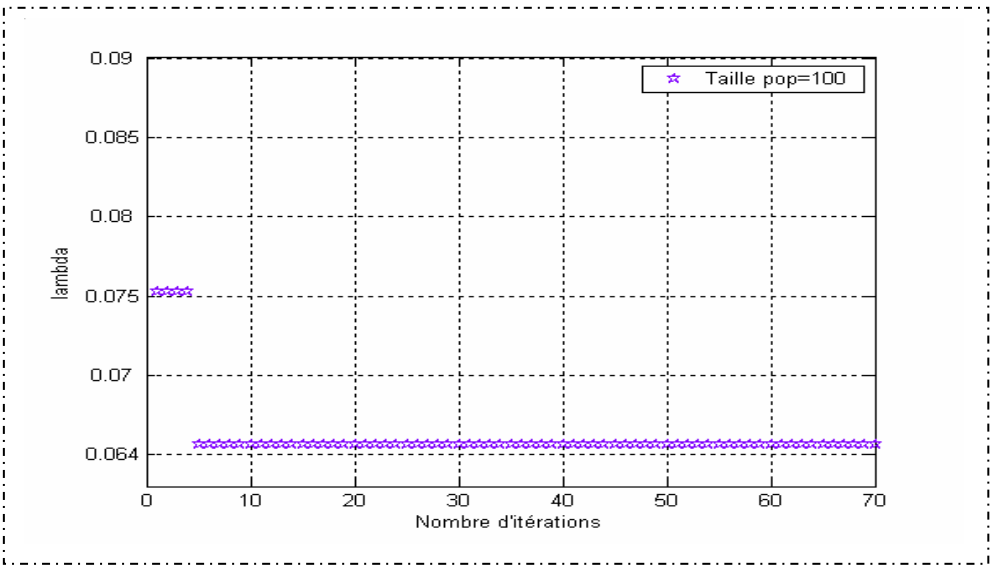


Figure 4. 7- Evolution du taux de défaillance avec mémorisation de la meilleure solution

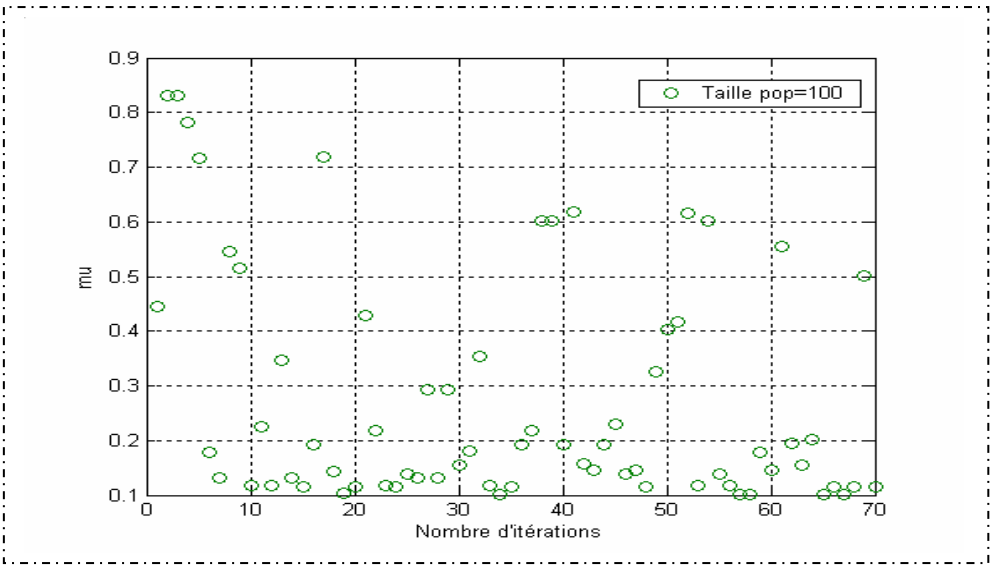


Figure 4. 8- Evolution du taux de réparation en fonction du nombre d'itérations

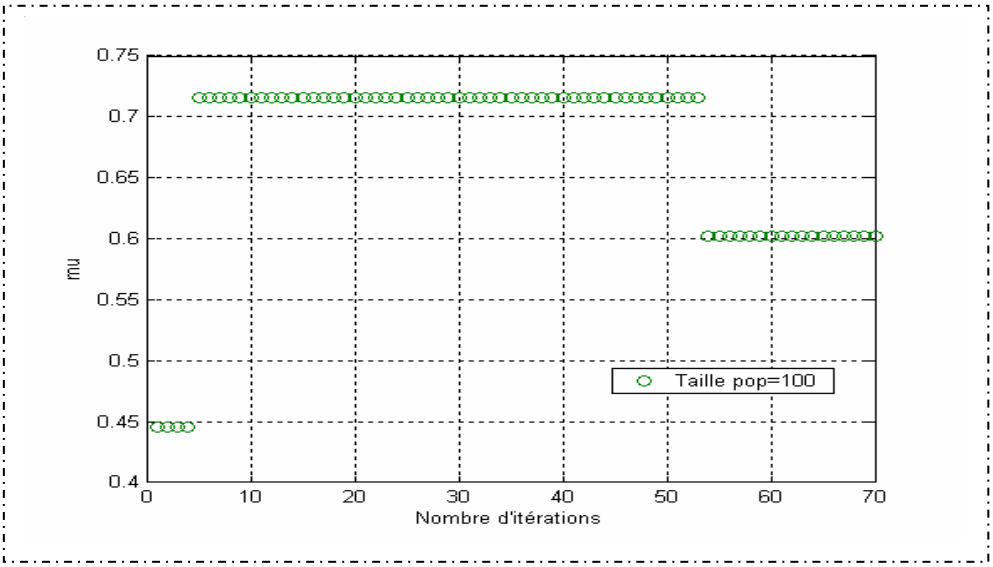


Figure 4. 9- Evolution du taux de réparation avec mémorisation de la meilleure solution

Les résultats graphiques suivants apparaissent sous une représentation en 3D, ils identifient la complicité des paramètres entre eux.

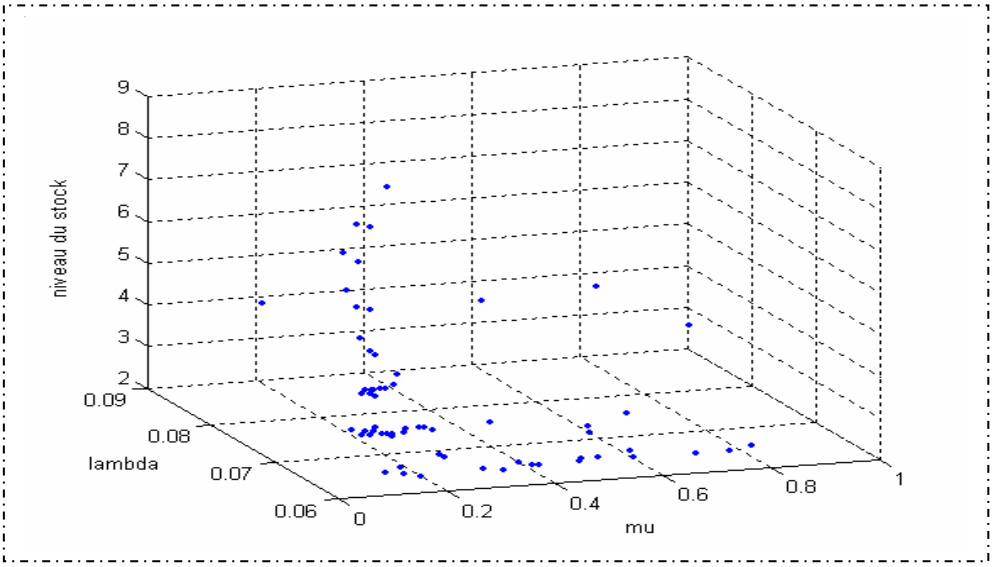


Figure 4. 10- Taux de défaillance et taux de réparation relatifs au niveau du stock

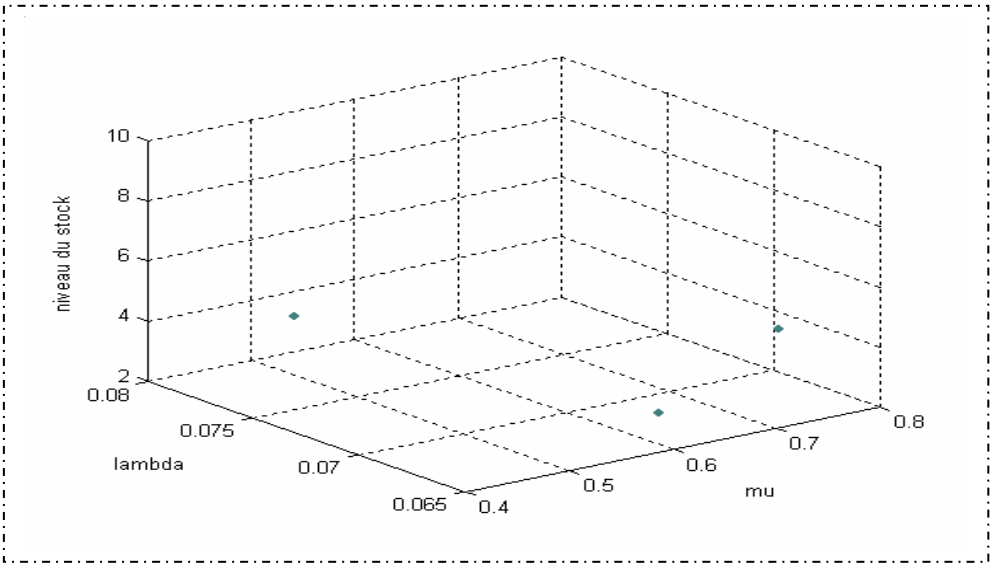


Figure 4. 11- Taux de défaillance et taux de réparation relatifs au niveau du stock avec mémorisation de la meilleure solution

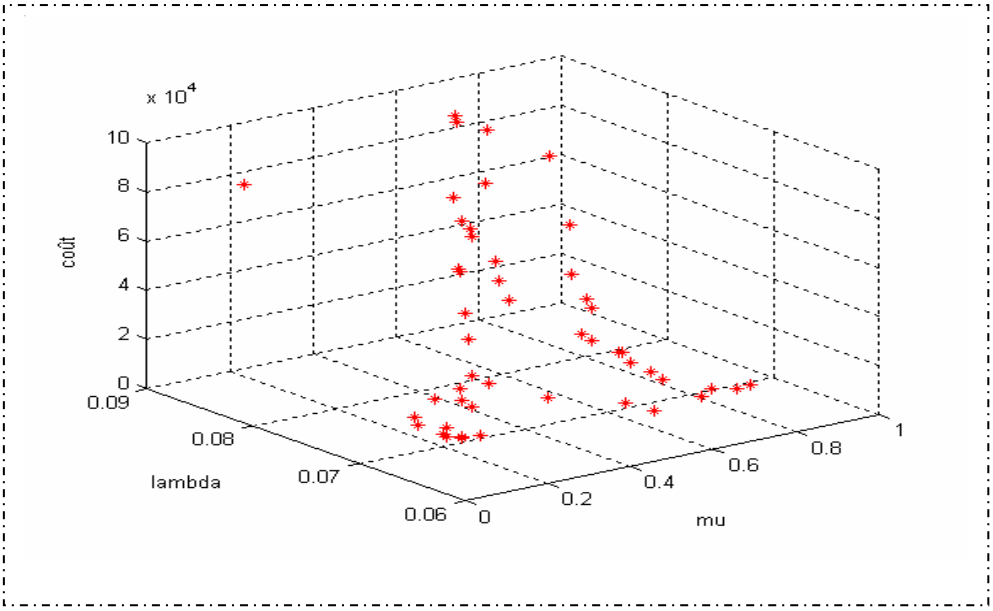


Figure 4. 12- Taux de défaillance et taux de réparation relatifs à l'optimum

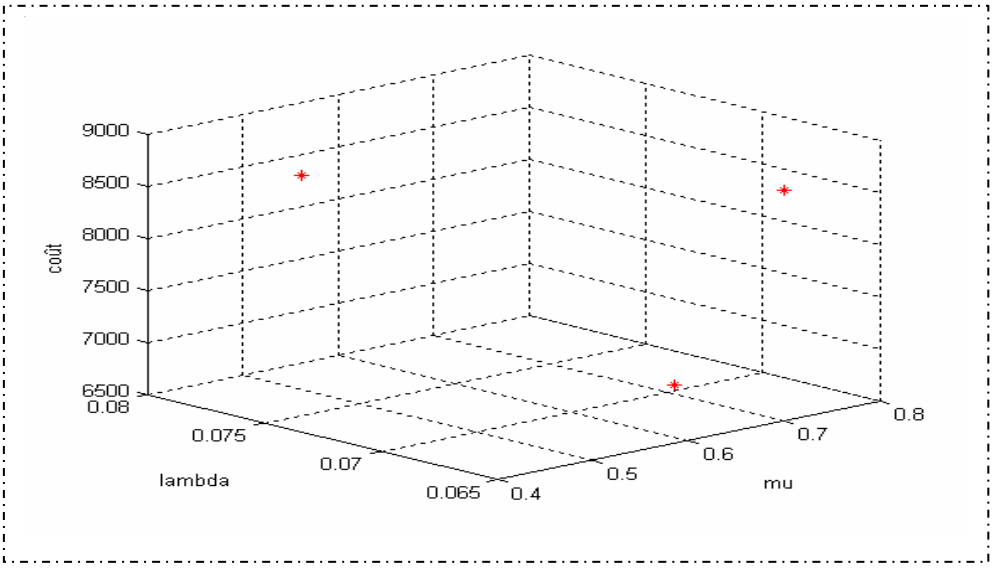


Figure 4. 13- Taux de défaillance et taux de réparation relatifs à l'optimum avec mémorisation de la meilleure solution

4.7 Synthèse et analyse des résultats

Les résultats expérimentaux sur le problème d'optimisation montrent bien l'existence d'une relation entre les résultats trouvés et les paramètres de l'algorithme génétique. Si ce dernier est riche quant à la diversité des solutions générées, il est difficile à paramétrer (taille de la population, nombre de générations, probabilité de croisement, probabilité de mutation). Par ailleurs la détermination de la taille optimale de la population reste encore un sujet de recherche sans réponse. Toutefois, un constat était évident pour le problème d'optimisation du coût du stock des membranes, l'utilisation d'une population de petite taille aboutit rapidement à la solution même avec un nombre important d'évaluations de la fonction d'adaptation, comme l'indique le tableau 4.1. Il semble donc que choisir une population de petite taille n'est pas une voie prometteuse malgré qu'un tel choix ne soit pas coûteux en terme de temps.

Taille population	Nombre générations	Pc	Pm	Optimum	Quantité stockée	Taux panne	Taux réparation
4	10	75%	10%	7050,96	4	0.802	0.088
4	80	75%	10%	8651,94	5	0.684	0.056

Tableau 4. 1- Optimum identifié par des populations de petites tailles

L'algorithme génétique qui a l'avantage d'améliorer au cours de son exécution la meilleure solution potentielle identifiée à la fin de chaque génération a adopté une certaine stratégie. Celle-ci consiste à conserver à chaque fois la meilleure solution entre celle en cours et sa précédente. De cette façon, il ne garde que les bonnes solutions puisqu'il délaisse une solution si l'antécédente est plus meilleure. Ce mécanisme est déclenché à la fin de chaque génération pour en garder la meilleure solution qui sera la base de la prochaine génération. L'objectif de cette stratégie est l'amélioration des solutions dès la première génération jusqu'à la fin du processus descriptif de l'algorithme génétique.

Les résultats obtenus avec les paramètres pris en compte sont résumés par le tableau 4.2. En effet, on retrouve la quantité de pièces de rechange devant être présente dans le stock qui minimise le coût de son exploitation. Dans ces conditions, on retrouve aussi le taux de défaillance qui doit être autant que possible petit (dans son domaine de définition) car on vise toujours de ne pas avoir de panne et le taux de réparation qui doit être autant que possible élevé du moment où on doit réparer rapidement.

Taille population	Nombre générations	Pc	Pm	Optimum	Quantité stockée	Taux panne	Taux réparation
100	70	75%	10%	6989,61	3	0.6020	0.0656

Tableau 4. 2- Récapitulatif des résultats de l'application de l'AG

Le bon choix de la quantité à stocker des pièces de rechange permet de maîtriser le bon fonctionnement des machines ce qui entraîne une productivité continue. En revanche, si un arrêt se produit pour cause de panne de l'équipement de production et que le stock ne

dispose pas de la quantité suffisante, alors des coûts énormes peuvent être engendrés à l'entreprise (pénalités pour les livraisons en retard, perte de clientèle,…); les pannes pénalisent la rentabilité. Une bonne quantité des pièces stockées réduit les risques d'arrêt de productivité et par conséquent les pertes en matière de délai, de panne et de rentabilité seront maîtrisées.

4.8 Conclusion

Dans ce chapitre, nous nous étions fixés pour but d'explorer plus avant le problème de l'optimisation du coût du stock par détermination du niveau du stock conduisant à ce coût optimal. Nous avons donc tout d'abord présenté quelques concepts relatifs à la gestion des stocks. Puis nous nous sommes intéressés directement à l'application de l'algorithme d'optimisation fondé sur la théorie de l'évolution de Darwin à savoir l'algorithme génétique pour optimiser le coût d'exploitation du stock. Cet algorithme présente des caractéristiques intéressantes le rendant populaire, notamment grâce à sa facilité d'emploi. Ses points forts sont une certaine robustesse due à l'utilisation de l'aléatoire et une facilité de mise en œuvre s'appuyant sur les techniques de base présentées dans le second chapitre.

L'implémentation de l'algorithme génétique, a en particulier permis d'apprécier l'importance de la définition de la fonction d'adaptation dans le processus d'optimisation. En outre, la force de cet algorithme par exploration globale de l'espace des paramètres d'autant qu'il ne nécessite aucun calcul des dérivées ou encore sans connaissance explicite du domaine de travail en manipulant simplement des chaînes de bits et en utilisant des opérateurs simples qui ne mettent en jeu que des procédures aussi peu complexes que la génération des nombres aléatoires, la copie des chaînes et les échanges des parties de chaînes.

D'un autre point de vue, la force de l'algorithme génétique est plus considérée avec la montée en puissance exponentielle du matériel informatique qui a diminué la contrainte en termes de temps d'exécution.

Conclusion

L’aspect maintenance industrielle joue aujourd’hui un rôle déterminant dans la conduite de la production industrielle. Elle a acquis dans l’entreprise une position clef. Que ce soit dans le domaine de la gestion de production ou dans celui du pilotage d’un atelier de production. La gestion du stock est un problème difficile qui recouvre souvent des enjeux économiques de première importance. Cette importance devient cruciale lorsqu’elle touche deux fonctions aussi importantes qu’antagonistes au sein de l’entreprise la production et la maintenance. Ces deux fonctions agissent sur les mêmes ressources. En effet, la production oeuvrant quotidiennement sur des équipements que la maintenance doit garder toujours en état de fonctionnement en disposant d’un niveau de stock jugé satisfaisant de pièces de rechange. Pour arriver à cette satisfaction, une méthode stochastique a été adoptée. Il s’agit des algorithmes génétiques.

Ces algorithmes sont des méthodes génériques de résolution approchée des problèmes d’optimisation. Elles permettent avec une même méthode, d’envisager une résolution approchée de nombreux problèmes d’optimisation différents. Parmi lesquels, le problème de gestion des stocks est particulièrement traité.

Les stocks étant des ressources matérielles qui ont une valeur économique et sont inutilisées ou en attente d’utilisation ont nécessité un intérêt particulier et un suivi rigoureux. Par conséquent, la gestion des stocks consiste à assurer à tout moment et aux meilleures conditions la conservation et la mise à disposition de l’utilisateur des matières ou marchandises dont il a besoin.

Dans le cadre de ce mémoire, nous avons tenté de faire un mixage de deux domaines différent l'un de l'autre, les algorithmes évolutionnaires, plus précisément les algorithmes génétiques et la gestion des stocks. Ce mixage s'inscrit dans le souci d'optimiser le coût d'exploitation du stock par algorithme génétique où nous avons considéré l'ensemble des paramètres relatifs à une pièce de rechange stockée (afin d'éviter l'arrêt de la production) comme des individus ayant à s'adapter à leur environnement.

La démarche que nous avons suivie a conduit dans un premier temps à appliquer l'algorithme génétique sur quelques fonctions mathématiques pour pouvoir comparer les résultats qu'il fournit avec ceux connus analytiquement, dans le but de valider l'algorithme génétique implémenté. Dans un second temps, elle nous a conduit à exprimer précisément le problème de gestion des stocks où il nous paraissait intéressant d'imaginer un système recherchant les meilleures données assurant un coût de stockage optimal. Dès lors, il nous a fallu définir une stratégie d'application des algorithmes génétiques au problème de gestion des stocks.

L'algorithme génétique implémenté repose sur l'utilisation des opérateurs génétiques simples à savoir la sélection de la Roulette, le croisement à un–point et la mutation simple. Il n'était pas question de se comparer à des opérateurs spécifiques, l'idée étant de proposer des opérateurs efficaces dans le cadre d'une utilisation simple sans spécialisation. En outre, le codage retenu est très simple dont l'alphabet chromosomique est binaire (donc la cardinalité minimale).
Pour évaluer la valeur des individus, une fonction d'adaptation a été présentée, constituée d'un terme permettant de rechercher l'ensemble des paramètres conduisant à un coût minimal d'un stock de membranes.

La mise en œuvre de l'algorithme génétique et son application, ont permis d'obtenir les résultats recherchés relatifs au problème de détermination du coût optimal du stock se traduisant par le coût optimal (qui est l'objectif), la quantité à être en stock ainsi que les paramètres des pièces de rechange mis en jeu à savoir le taux de panne et le taux de réparation.

Notre démarche consiste alors en une exploitation des algorithmes génétiques qui peuvent être d'une efficacité face à n'importe quel problème notamment au problème de gestion des stocks. Ces algorithmes méritent une attention particulière.

A cet effet, leur application au problème de gestion des stocks a montré leur intérêt à ce type de domaine et ouvre de nombreuses perspectives de recherche prometteuses pour l'avenir telles que :

- Utilisation du codage réel.
- Recherche d'une méthode hybride telle que le Recuit Simulé ou la recherche Tabou avec les Algorithmes Génétiques.
- Intégrer un système d'apprentissage par réseaux de neurones des paramètres de l'Algorithme Génétique.
- Utilisation des Algorithmes Génétiques pour l'ordonnancement du moment où ce dernier représente un autre aspect opérationnel des décisions de la production.

Bibliographie

[Abd04] Bachir Abdelhadi. “Contribution à la conception d’un moteur à induction spécial à rotor externe pour un système de propulsion électrique. Développement d’un algorithme génétique adaptatif pour identification paramétrique”. Thèse de Doctorat en sciences, Université de Batna, 2004.

[Ale97] Frederic Alexandre. “Intelligence neuromimétique”. Mémoire d’Habilitation à Diriger des Recherches, Université Henri Poincaré, Nancy I, 1997.

[Ala01] Claude Alazard et Sabine Separi. “Contrôle de gestion”, Manuel et applications. D E C F. Paris , Dunod, 2001.

[All02] Jean-Marc Alliot, Thomas Schiex, Pascal Brisset, Frédérick Garcia. “Intelligence Artificielle et informatique théorique”. Paris, Cépaduès. 2ème Edition, 2002.

[Ama96] Jean-Louis Amat et Gérard Yahiaoui. “Techniques avancées pour le traitement de l’information. Réseaux de neurones, logique flou, algorithmes génétiques”. Cepadues Editions, 1996.

[Ben01] Sana Ben Hamida. “Algorithmes évolutionnaires, Prise en compte des contraintes et application réelle”, Thèse de Doctorat, Université, 2001.

[Ben95] Boualem Benmazouz. “Recherche opérationnelle de gestion”. Atlas Edition, 1995.

[Boi87] Daniel Boitier et Claude Hazard. “Guide de la maintenance”. Paris, Nathan, 1987.

[Bel98] N. Bellaaj-Mrabet et K.Jelassi. “Comparaison de méthodes d’identifications des paramètres d’une machine asynchrone”. The European Physical Journal, Applied Physics, EDP Sciences, 1998.

[Bou00a] Laurent Bougrain. “Algorithmes Génétiques”, Projet CORTEX, LORIA/INRIA LORRAINE, 2000.

[Bou00b] Laurent Bougrain. “Intelligence artificielle”. Projet CORTEX, LORIA/INRIA LORRAINE, 2000.

[Boy01] André Boyer. “L’essentiel de la gestion”. Termes, Contextes et Bibliographie, Editions d’Organisation, 2001.

[Cou95] Alain Courtois, Chantal Martin Bonnefous et Maurice Pillet. “Gestion de la production. Editions d’Organisation”. 1995.

[Cra03] Yves Crama. “Eléments de gestion de la production”. Ecole d’Administration des Affaires. Université de Liège, 2003.

[Dam01] Jérôme Damelincourt. “Découverte de la vie artificielle”. Futura-Sciences, 2001.

[Der90] Salah Derradji. “Analyse numérique 1”. INESM de Batna département de mécanique, 1990.

[Dew03] Daniel Dewolf. “Gestion de la production”. Première licence sciences de gestion et ingénieurs de gestion, Université de Liège, 2003.

[Dro01] Alexis Drogoul. “Introduction à l'Intelligence Artificielle Distribuée”. Thème OASIS, Equipe MIRIAD, Université Paris 6, LIP6, 2001.

[Dum94] Renaud Dumeur. “Synthèse de comportements animaux individuels et collectifs par Algorithmes Génétiques”. Thèse de Doctorat, Université de Paris-8, 1994.

[Fin02] Sous la direction de Gerd Finke. “Recherche opérationnelle et réseaux. Méthodes d’analyse spatiale”. Paris, Hermès Science, 2002.

[Fra02] Dominique Francisci. “Algorithmes évolutionnaires et optimisation multi-objectifs en Data mining”. Rapport de recherche, Laboratoire I3S, Sophia Antipolis, 2002.

[Gol94] David E.Goldberg. "Algorithmes génétiques Exploitation, optimisation et apprentissage automatique". Addison-Wesley, France, SA, 1994.

[Har03] Youssef Harrat. "Contribution à l'ordonnancement conjoint de la production et de la maintenance : Application au cas d'un job Shop". Thèse de Doctorat, L'U.F.R des Sciences et Techniques, Université de Franche-comté, 2003.

[Hau98] Randy L. Haupt, Sue Ellen Haupt. "Practical genetic algorithms", AWiley-Interscience Publication, New York/ Chichester/ Weinheim/ Brisbane/ Singapore/ Toronto, 1998.

[Kaf01] Hedi Kaffel. "La maintenance Distribuée : Concept, Evaluation et mise en œuvre". Thèse de Doctorat, Université Laval (QUEBEC), 2001.

[Lat98] Claude Lattaud. "Approche adaptative de systèmes multi-agents dans un contexte vie artificielle". Thèse de Doctorat, Université Paris 5- René Descartes, 1998.

[Mad02] Blaise Madeline. "Algorithmes évolutionnaires et résolution de problèmes de satisfaction de contraintes en domaines finis". Thèse de Doctorat en sciences, Université de Nice-Sophia antipolis, 2002.

[Mag98] Vincent Magnin. "Contribution a l'étude et à l'optimisation de composants optoélectroniques". Thèse de Doctorat, Université Des Sciences et Technologies, Lille, 1998.

[Mag99] Jean-Philippe Mague. "Explication des connaissances d'un réseau de neurones artificiels au moyen d'un moniteur génétique". Rapport de stage, Université Joseph Fourier, 1999.

[Mat02] D. Matthieu. "Les algorithmes génétiques". 2002.

[Mam03] Lamia Mamlouk et Fouad Benouirane. "Vie Artificielle". 2003.

[Me94] Ludovic Me. "Audit de sécurité par algorithmes génétiques". Thèse de Doctorat, Université de Rennes 1, 1994.

[Mon00] Nicolas Monmarché. "Algorithmes de fourmis artificielles : applications à la classification et à l'optimisation". Thèse de Doctorat, Université de tours, 2000.

[Mon93] Joseph Monks. "Gestion de la production et des opérations". 1993.

[Oli98] Jean-Luc Olivès. “Optimisation globale d’un système imageur à l’aide de critères de qualité visuelle”. Thèse de Doctorat, Ecole nationale supérieure de l’aéronautique et de l’espace, 1998.

[Pag80] Alain Pages & Michel Gondran. “Fiabilité des systèmes”. Paris, Eyrolles, 1980. (Collection de la Direction des Etudes et Recherche d’Electricité de France.

[Pla02] David Placci. “Développement d’une application de calcul distribué basée sur un protocole peer-to-peer”. Mémoire de postgrade en informatique et organisation, Ecole des Hautes Etudes Commerciales, Université de Lausanne 2002.

[Reb99] Pascal Rebreyend. “Algorithmes génétiques hybrides en optimisation combinatoire”. Thèse de Doctorat, Ecole Normale Supérieure de Lyon, 1999.

[Ren95] Jean-Michel Renders. “Algorithmes génétiques et réseaux de neurones”. Hermes, 1995.

[Sak84] Michel Sakarovitch. “Optimisation combinatoire Méthodes mathématiques et algorithmiques Graphes et programmation linéaire”, 1984.

[Sav03] Marc Savall. “Une architecture d’agents pour la simulation. Le modèle YAMAM et sa plate-forme Phoenix”. Thèse de Doctorat, INSA de Rouen, 2003.

[Sch01] Marc Schoenauer. “Les algorithmes évolutionnaires : état de l’art et enjeux”. Algorithms Seminair 2001-2002. Projet Fractales, INRIA Rocquencourt, 2001.

[Sma96] Hacène Smadi. “Contribution à la mise en place de la maintenance productive totale T.P.M dans une unité industrielle”. Cas. E.MI.B. Mémoire de Magister, Université de Batna, 1996.

[Spa94] Alain Spalanzani. “Précis de gestion industrielle et de production”. Office des Publications Universitaires, 1994.

[Spa99] Anne Spalanzani. “Algorithmes évolutionnaires pour l’étude de la robustesse des systèmes de reconnaissance automatique de la parole”. Thèse de Doctorat, Université Joseph Fourier, Grenoble I, 1999.

[Seb96] Michèle Sebag et Marc Schoenauer. “Contrôle d’un algorithme génétique”. Revue d’intelligence artificielle Vol.10/1996.

[Vil88] Alain Villemeur. “Sûreté de fonctionnement des systèmes industriels, fiabilité- facteurs humains -informatisation”. Paris, Eyrolles, 1988. (Collection de la direction des études et recherches d’électricité de France).

[Val01] Thomas Vallee et Murat Yildizogln. “Présentation des algorithmes génétiques et de leurs applications en économie”. LEN-C3E Université de Nantes, LEA-CIL, Université Montesquieu Bordeaux IV, 2001.

Résumé

Les entreprises sont de plus en plus sensibilisées à l'importance des coûts induits par les défaillances accidentelles des systèmes de production. De ce fait la maintenance a pris une position clef. Pour l'assurer convenablement, les gestionnaires dans le monde industriel se sont intéressés à la gestion des stocks des pièces de rechange, plus particulièrement à l'optimisation des coûts de leur exploitation. Dans ce contexte, les algorithmes génétiques constituent une solution idéale.

Dans un premier temps, ce mémoire expose le fonctionnement de l'optimisation par algorithme génétique qui simule le processus de sélection naturelle. Cet algorithme opère sur des populations composées d'un ensemble d'individus. Ces individus sont codés dans des génotypes composés de gènes correspondant aux valeurs des différentes variables de la fonction mathématique.

Dans un second temps, le même algorithme implémenté est utilisé pour optimiser le coût d'un stock de pièces de rechange, où les paramètres codant les chaînes constituant la population représentent les paramètres des pièces de rechange à stocker.

Comme dans l'évolution naturelle, on espère l'émergence progressive d'individus de mieux en mieux adaptés : les meilleurs individus de la population finale sont des approximations de solutions du problème d'optimisation du coût du stock des pièces de rechange en occurrence le taux de panne, taux de réparation et la quantité à stocker.

Mots clés : Algorithmes génétiques, Gestion des stocks, Coût, Maintenance.

Glossaire

Environnement

Espace de recherche

Évolution

Méthode de recherche de l'individu le plus adapté à partir d'un grand nombre d'individus.

Exobiologie

Le biologiste américain Joshma Lederberg la définie : la recherche et l'étude de la vie en dehors de notre planète. Ce terme englobe désormais tout ce qui concerne l'étude de la vie dans l'univers.

Cognitivisme

Prône le fonctionnement computationnel du cerveau et des mécanismes biologiques en général. Il considère que le vivant, tel un ordinateur, manipule des symboles élémentaires.
C'est un courant de recherche dont l'objectif est l'étude du fonctionnement de l'activité intellectuelle.

Connexionnisme

Discipline concernant les techniques de simulation des processus intelligents par des réseaux de neurones.

Cybernétique

Science de la communication et du contrôle chez l'animal et la machine.

Fonction d'adaptation

La fonction d'adaptation est une fonction mesurant l'adaptation d'un individu d'une population donnée. Elle est la clé de voûte des AGs. Il est très important de définir une bonne fonction d'adaptation pour un problème donné, ce qui est certainement la plus grosse difficulté des Algorithmes Génétiques.

Gène

Les chromosomes se décomposent en gènes qui peuvent prendre des valeurs différentes que l'on appelle allèles. La position d'un gène dans un chromosome est identifiée par son locus. Un gène est une caractéristique génétique d'un individu. En génétique, on parle du gène de la couleur des yeux par exemple.

Dans les AGs, les gènes ont des valeurs appartenant à un alphabet qui dépend du problème à résoudre. On peut ainsi avoir des allèles binaires, décimaux ou encore hexa-décimaux. On peut imaginer d'autres sortes de codage.

Génération

Une génération est une population à un instant t. Les AGs faisant évoluer les populations, on crée une nouvelle génération d'individus plus adaptés. Cette évolution est effectuée par les opérateurs de reproduction, de croisement et de mutation.

Hardware évolutif

Entité matérielle capable de s'adapter à toute situation. C'est une matérialisation des théories sur les algorithmes génétiques.

Heuristique

Méthode conçue pour un problème d'optimisation donné, qui produit une solution non nécessairement optimale lorsqu'on lui fournit une instance de ce problème. Son objectif est trouver une solution de qualité raisonnable en un temps de calcul aussi faible que possible.

Intelligence Computationnelle

Terme populaire utilisé pour référencer les techniques basées sur la puissance de calcul des ordinateurs et sur l'intelligence humaine pour résoudre des problèmes difficiles.

Méthode d'essai et erreur

Méthode de base pour l'optimisation, il s'agit de tester un certain nombre de solutions potentielles jusqu'à l'obtention d'une solution adéquate.

Méthode heuristique

Méthode de calcul pour un problème générique d'optimisation produisant une solution qui n'est pas forcement optimale. La plupart de ces méthodes sont bien spécifiques à un problème donné et s'appliquent en général qu'à des problèmes discrets.

NP-Complets

Un problème appartient à la classe NP si on ne connaît pas d'algorithme polynomial.

Population

Les algorithmes génétiques ne travaillent pas sur un individu, sur une donnée, mais au contraire sur une population de chaînes afin d'effectuer des opérations, des recherches sur un domaine de possibilités plus important. C'est une des grandes forces des AGs. Une population se compose de chaînes ou chromosomes.

Chromosome ou individu

Les chaînes des systèmes génétiques artificiels sont analogues aux chromosomes des systèmes biologiques. Ils portent les informations génétiques d'un individu. Ainsi, un individu se compose de gènes.

Principe Darwinien

Les plus adaptés survivent et se produisent.

Problème de décision

Un problème de décision est un énoncé auquel la réponse peut être uniquement oui ou non. A chaque problème d'optimisation correspond un problème de décision.

Problème d'optimisation

Un problème d'optimisation est un problème pour lequel on doit chercher une solution admissible optimisant au mieux une fonction objectif.

Sélection naturelle

Est le processus selon lequel les individus mieux adaptés à leur environnement tendent à survive en plus grand nombre et à se reproduire plus fréquemment que leurs congénères moins bien lotis.

Sûreté de Fonctionnement

Science des défaillances ; elle inclut aussi leur connaissance, leur évaluation, leur prévision, leur mesure et leur maîtrise. C'est la propriété qui permet aux utilisateurs d'un système de placer une confiance justifiée dans le service qu'il leur délivre.