République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université M'hamed Bougara - Boumerdès

Faculté des Sciences
Département Mathématiques

Domaine : Mathématiques et Informatique
Filière : Mathématiques
Spécialité : **Mathématique Financière**

Mémoire de fin de cycle en vue de l'obtention du Diplôme de Master en Mathématiques Financières

Thème

Algorithme génétique multi-objectif

(Application au portefeuille financier)

Présenté par :

LEGHRIB Nour elhouda et ZIANI Hassina

Devant le jury composé de :

Président : M^{me} BENSAREDJ HASSIBA MCB UMBB

Examinateur : M^{me} LARABI GHENIMA MAA UMBB

Encadreur : M^{me} IKHLEF MASSIKA MAA UMBB

Année universitaire :2022 / 2023

REMERCIEMENTS

En premier lieu, nous remercions Dieu tout puissant de nous avoir accordé santé,

courage et volonté pour accomplir ce travail.

Nous tenons à exprimer nos vifs remerciements à toutes les personnes qui nous ont

soutenues, et qui n'ont pas cessés de nous donner des conseils en signe de reconnaissance et

qui ont contribué de près ou de loin à la réalisation de ce mémoire .

On tient à remercier particulièrement notre promotrice Mme **Massika IKHLEF**

pour son encadrement.

Nous exprimons notre grand respect aux honorables membres de jury qui ont accepté

d'évaluer ce travail.

Notre plus grande reconnaissance est pour nos parents pour leurs présence et leurs soutient.

Table des matières

Liste des figures

Liste de tableaux

Liste de pseudocode

Introduction générale

Les problèmes d'optimisation déterministe sont utilisés pour modéliser et analyser un grand nombre de systèmes pour lesquels on recherche une stratégie optimale (vis à vis d'un certain critère appelé fonction objectif) satisfaisant un certain nombre de contraintes. Dans ces problèmes, les paramètres définissant les contraintes et la fonction objectif sont supposés connus. Cependant, un grand nombre de problèmes d'optimisation sont des problèmes d'optimisation multiobjectif. Par exemple, le problème de gestion de portefeuilles financiers consiste à déterminer les proportions budgétaires attribués à chaque actif constituant le portefeuille de telle sorte à maximiser le rendement et minimiser le risque tout en satisfaisant des contraintes physiques et stratégiques. La complexité du problème vient du fait qu'on doit optimiser une fonction avec plusieurs objectifs et qui sont souvent contradictoires, pour cela il faut recourir à des outils mathématiques extrêmement performants dans le but de sélectionner des solutions optimales en un temps raisonnable.

L'objet de cette thèse est de modéliser et analyser un problème d'optimisation multi objectif et de proposer un algorithme génétique de résolution pour ce problème. Un intérêt tout particulier sera porté sur l'interfaçage entre les statistiques et l'optimisation. Autrement dit, les objets statistiques interviennent afin de résoudre notre problème d'optimisation. Le plan de la thèse est le suivant :

Le premier chapitre est un chapitre introductif dans lequel nous rappelons quelques notions sur l'optimisation multi-objectif, qui consiste à optimiser simultanément plusieurs fonctions. La notion de solution optimale unique dans l'optimisation uni-objectif disparait pour les problèmes d'optimisation multi-objectif au profit de la notion d'ensemble de solutions Pareto optimales.

Le deuxième chapitre porte sur l'imitation du processus biologique de l'évolution pour la résolution des problèmes d'optimisation inventée par John Holland et exposée dans son ouvrage "Adaptation in natural and artificial system" en 1975, connue aujourd'hui sous le nom "Algorithmes Génétiques" et devenue populaire dans le domaine d'optimisation grâce à Goldberg (1989). Les algorithmes génétiques sont basés sur l'évolution naturelle (Darwin, 1859 [8]), le principe de ce processus est la survie du plus adapté. Les éléments concernés par un processus évolutionnaire sont sujets d'une sélection qui ne gardera à la fin que ceux qui

présentent les meilleures facultés d'adaptation. Les organismes biologiques sont un exemple d'adaptation et d'optimisation car ils ont réussi à s'adapter et à survivre l'environnement naturel pendant des millions d'années. Holland dans son ouvrage (1975) a voulu traduire ce processus d'évolution et d'adaptation mathématiquement afin de pouvoir optimiser des problèmes qu'on rencontre dans la majorité des domaines, l'économie, la médecine, l'intelligence artificielle et les réseaux neuronaux … etc.

Ces algorithmes ont été largement utilisés aussi dans la communauté multi objectif, ils sont très appropriés pour résoudre les problèmes de gestion de portefeuilles grâce à l'utilisation d'une population initiale et l'application des opérateurs génétiques (croissement et mutation). Parmi ces algorithmes, on cite en détaille le Non dominated sorting genetic algorithm II proposée par Deb et al. en 2002 et qui basés sur le principe de Pareto.

A la fin de ce travail, on présente une application informatique implémentée sous l'environnement R, distribué gratuitement à partir du site du CRAN (Compréhensive R Archive Network) ! : http ://www.r-project.org/. R est un système d'analyse statistique crée par Ross Ihaka et Robert Gentleman distribué librement sous les termes de la GNU General Public Licence. Cette application a pour objectif de chercher un portefeuille efficient, c'est à dire déterminer les proportions budgétaires attribuées à chaque actif financier constituant le portefeuille. Les données nécessaires pour réaliser cette application sont le vecteur rendement et la matrice de variance covariance de trois marchés financiers Hong Kong Hang Seng (31 actifs), SP 100 américain (98 actifs) et SP 500 américain (276 actifs). Ces données ont été téléchargées sur Beasley's OR Library.

1 . CHAPITRE 01 : OPTIMISATION MULTI OBJECTIF

1.1. Introduction sur les problèmes d'optimisation

De nombreux secteurs de l'industrie sont concernés par les problèmes d'optimisation combinatoire. En effet, que l'on s'intéresse à l'optimisation d'un système de production, au traitement d'images, à la conception de systèmes, au design de réseaux de télécommunication ou à la bio-informatique nous pouvons être confrontés à des problèmes d'optimisation combinatoire. Plusieurs problèmes ont été traités dans différents domaines :

- ✓ Design de systèmes dans les sciences d'ingénieurs (mécanique, aéronautique, chimie, etc.) :ailes d'avions [1], moteurs d'automobiles [2] ;
- ✓ Ordonnancement et affectation : ordonnancement en productique [3], localisation d'usines, planification de trajectoires de robots mobiles [4], etc.
- ✓ Agronomie : programme de production agricole, etc.
- ✓ Transport : gestion de containers [5], design de réseaux de transport [25],tracé autoroutier, etc.
- ✓ Environnement : gestion de la qualité de l'air [6], distribution de l'eau [7], etc.
- ✓ Télécommunications : design d'antennes [8], affectation de fréquences [9 ;10], radiotéléphonie mobile [11], etc.

Un problème d'optimisation est défini par **:**

- ✓ Un espace de recherche (de décision) : ensemble de solutions ou de configurations constituédes différentes valeurs prises par les variables de décision.
- ✓ Une ou plusieurs fonction(s) dite objectif(s), à optimiser (minimiser ou maximiser).
- ✓ Un ensemble de contraintes à respecter.

Dans la plupart des problèmes, l'espace d'état (décision) est fini ou dénombrable. Les variables du problème peuvent être de nature diverse (réelle, entier, booléenne, etc.) et exprimer des données qualitatives ou quantitatives. La fonction objective représente le but à atteindre pour ledécideur. L'ensemble de contrainte définit des conditions sur l'espace d'état que les variables doivent satisfaire. Ces contraintes sont souvent des contraintes

d'inégalité ou d'égalité et permettent en général de limiter l'espace de recherche (solutions réalisables). La résolution optimale du problème consiste à trouver le point ou un ensemble de points de l'espace de recherche qui satisfait au mieux la fonction objective. Le résultat est appelé valeur optimale ou optimum. Néanmoins en raison de la taille des problèmes réels, la résolution optimale s'est souvent montrée impossible dans un temps raisonnable. Cette impossibilité technique impose larésolution approchée du problème, qui consiste à trouver une solution de bonne qualité (la plusproche possible de l'optimum). Il est vital pour déterminer si une solution est meilleure qu'uneautre, que le problème introduise un critère de comparaison (une relation d'ordre). La plupart des problèmes d'optimisations appartiennent à la classe des problèmes NP-difficile classe où iln'existe pas d'algorithme qui fournit la solution optimale en temps polynomial en fonction de lataille du problème et le nombre d'objectifs à optimiser. Dans la littérature il existe desproblèmes académiques utilisés comme des benchmarks : sac à dos, les fonctions de schaffer, voyageur de commerce, *Flowshop,* et des problèmes réels (applications industrielles) : télécommunications, transport, environnement, etc.

Un problème d'optimisation est caractérisé par :

- ✓ Le domaine des variables de décision : soit Continu et on parle alors de problème continu, soit discret et on parle donc de problème combinatoire.
- ✓ La nature de la fonction objective à optimiser : soit linéaire et on parle alors de problème linéaire, soit non linéaire et on parle donc de problème non linéaire.
- ✓ Le nombre de fonctions objectifs à optimiser : soit une fonction scalaire et on parle alors de problème mono-objectif, soit une fonction vectorielle et on parle donc de problème multi objectif .
- ✓ La présence ou non des contraintes : on parle de problème sans contrainte ou avec contrainte.
- ✓ Sa taille : problème de petite ou de grande taille.
- ✓ L'environnement : problème dynamique (la fonction objective change dans le temps).

Face à un problème d'optimisation :

- ✓ Elaborer un modèle (mathématiques) : l'expression de l'objectif à optimiser et les contraintesà respecter.
- ✓ Développer un algorithme de résolution.
- ✓ Evaluer la qualité des solutions produites.

Dans ce chapitre nous allons aborder le premier et le troisième point. Le deuxième point sera sujet du chapitre suivant.

1.2. Problèmes d'optimisation mono-objectifs

Lorsqu'un seul objectif (critère) est donné, le problème d'optimisation est mono-objectif. Dansce cas la solution optimale est clairement définie, c'est celle qui a le coût optimal (minimal, maximal). De manière formelle, à chaque instance d'un tel problème est associé un ensemble Ωdes solutions potentielles respectant certaines contraintes et une fonction d'objectif $f: \Omega \rightarrow \Psi$ qui associe à chaque solution admissible $s \in \Omega$ une valeur $f(s)$. Résoudre l'instance (Ω, f) du problème d'optimisation consiste à trouver la solution optimale $s^* \in \Omega$ qui optimise (minimise ou maximise) la valeur de la fonction objectif f. Pour le cas de la minimisation : le but est de trouver $s^* \in \Omega$ tel que $f(s^*) \leq f(s)$ pour tout élément $s \in \Omega$. Un problème de maximisation peut être défini de manière similaire.

1.2.1. Variables de décision

Les variables de décision sont des quantités numériques pour les quelles des valeurs sont à choisir. Cet ensemble de n variables est appelé vecteur de décision : $(x_1, x_2, \ldots, x_n)$. Les différentes valeurs possibles prises par les variables de décision x_i constituent l'ensemble des solutions potentielles.

1.2.2. Espace décisionnel et espace objectif

Deux espaces Euclidiens sont considérés en optimisation :

- ✓ L'espace décisionnel, de dimension n, n étant le nombre de variables de décision.

Cet espaceest constitué par l'ensemble des valeurs pouvant être prise par le vecteur de décision.

- ✓ L'espace objectif : l'ensemble de définition de la fonction objectif, généralement défini dans R. La valeur dans l'espace objectif d'une solution est appelée coût, ou fitness.

1.2.3. Contraintes

Dans la plupart des problèmes d'optimisation, des restrictions sont imposées par les caractéristiques du problème. Ces restrictions doivent être satisfaites afin de considérer une solution acceptable. Cet ensemble de restrictions, appelées contraintes, décrit les dépendancesentre les variables de décision et les paramètres du problème. On formule usuellement ces contraintes c_j par un ensemble d'inégalités, ou d'égalités.

1.3. Optimisation multi objectif

L'optimisation multi-objectif consiste à trouver un vecteur de décisions qui satisfait les contraintes et optimise le vecteur objectif dont les éléments représentent les fonctions objectives, ces dernières forment une description mathématique du critère de performance, ces fonctions sont généralement en conflit. Le terme « optimiser » veut dire trouver une solution de telle façon que toutes les fonctions objectives renvoient des valeurs acceptables. Un problème de minimisation est décrit comme ci-dessous :

$$minimiser\ y = (f_1(x), \ldots, f_n(x))$$

$$Avec \qquad x = (x_1, \ldots, x_m) \in X$$

$$y = (y_1, \ldots \ldots, y_n) \in Y$$

Où x est appelé vecteur de décision constitué de m variables de décision, et y vecteur objectif de n objectifs. X représente l'espace de décision, et Y l'espace des objectifs. L'optimisation multi-objectif traite plusieurs fonctions objectives en même temps, pour cela, elle utilise la notion de compromis optimaux. Les solutions sont départagées en se basant sur la notion de dominance

au sens de Pareto.

1.3.1. Problèmes d'optimisation multi objectif

Un problème d'optimisation avec objectifs multiples peut être représenté par le programme suivant :

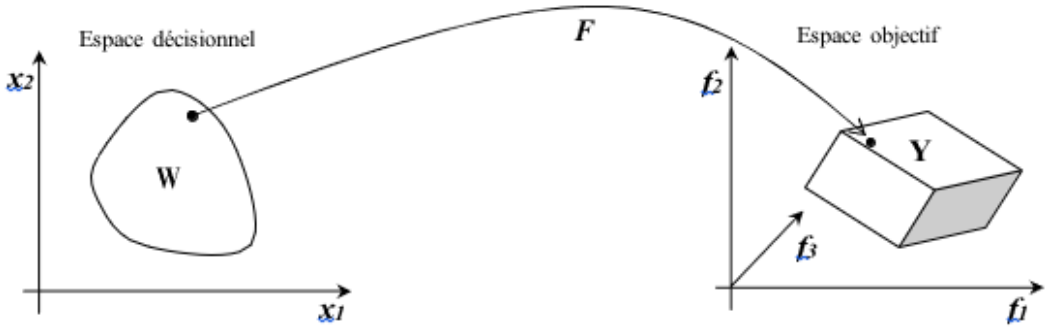

Figure 0.1 *- problème d'optimisation multiobjectif (2 variables de décision et 3 fonctions objectifs).*

Exemple : dans le cas de deux objectifs à minimiser, toute amélioration de l'un des objectifs se fait au détriment de l'autre et que la solution optimale ou proche de l'optimum est un compromis entre les deux. Dans l'achat d'une voiture d'occasion, la voiture idéale est celle qui est peu chère(critère économique) avec peu de kilomètres (critère qualitatif), il n'est pas évident de pouvoir regrouper en un seul objectif ces deux critères non commensurables. Ainsi il n'existe plus une solution optimale unique mais un ensemble de solutions. Nous allons donc devoir identifier lesmeilleurs compromis possibles suivant notre budget.

Les problèmes multi objectifs ont la particularité d'être beaucoup plus difficiles à traiter que leuréquivalent mono-objectif. La difficulté réside dans l'absence d'une relation d'ordre total

entre lessolutions. Une solution peut être meilleure qu'une autre sur certains objectifs et moins bonne surles autres. Donc il n'existe généralement pas une solution unique qui procure simultanément lasolution optimale pour l'ensemble des objectifs. Voilà pourquoi le concept de solution optimaledevient moins pertinent en optimisation multi objectif. Dans ce cas la solution optimale ou de bonne qualité n'est plus une solution unique mais, un ensemble de solutions compromis entre lesdifférents objectifs à optimiser. Il est vital pour identifier ces meilleurs compromis de définir unerelation d'ordre entre ces éléments. La plus célèbre et là plus utilisée est la relation de dominance au sens Pareto. L'ensemble des meilleurs compromis est appelé le front Pareto, lasurface de compromis ou l'ensemble des solutions efficaces. Cet ensemble de solutions constitue un équilibre, dans le sens qu'aucune amélioration ne peut être faite sur un objectif sans dégradation d'au moins un autre objectif. La solution Pareto consiste à obtenir le front de Pareto*PO* ou d'approximer la frontière de Pareto *PO**.

1.3.2. Vocabulaire et définitions

On considère ici le cas de maximisation des objectifs. La minimisation est définie de manière analogue.

Définition 1 « dominance au sens Pareto »

Soient deux vecteurs objectifs Y^1, $Y^2 \in \Psi$ / $Y^1 = F(S^1)$ et $Y^2 = F(S^2)$. On dit que la solution S^1 **domine** S^2 (Y^1 domine Y^2) si et seulement si : $Y^1 \geq Y^2$ et $Y^1 \neq Y^2$ (ie, $y_k^1 \geq y_k^2$ pour tout $k = 1 \ldots p$, et $y_k^1 > y_k^2$ pour au moins un k). On notera alors $S^1 \underline{f} S^2$. Si S^1 est meilleur que S^2 sur tous les objectifs (ie, $y_k^1 > y_k^2$ pour tout $k = 1 \ldots p$) alors on dit que S^1 **domine fortement** S^2 ; On notera alors $S^1 \geq S^2$. Lorsque ni $S^1 \geq S^2$, ni $S^2 \geq S^1$, alors on dit qu'elles sont **incomparables** ou Pareto équivalentes, $S^1 \sim S^2$. La relation de dominance est une relation d'ordre partiel stricte transitive, non réflexive et non antisymétrique [Dupas 2004].

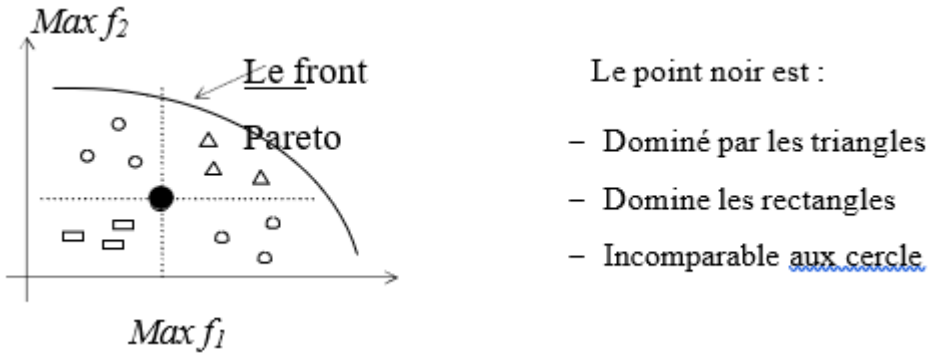

Figure 0.2 - *Relation de dominance (Cas de deux objectifs à maximiser)*

Définition 2 : Solution preto optimale et front de pareto

Une solution est dite Pareto optimale si elle n'est dominée par aucune autre solution réalisable.

L'ensemble Pareto optimal $PO^* = \{S \epsilon \Omega \mid \exists s' \in \Omega, f(s') \geq F(s)\}$

L'image de l'ensemble Pareto optimal *F(PO)* dans l'espace objectif Ψ est appelée **frontière Pareto**, ou surface de compromis. L'allure de cette frontière prend des formes différentes selon que les objectifs doivent être minimisés ou maximisés, (*figure* 1.3) cas de deux objectifs.

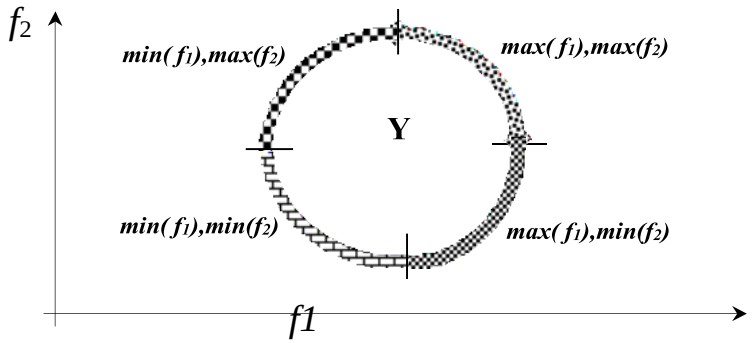

Figure 0.3 - *allure de la frontière Pareto selon l'optimisation (minimisation, maximisation) des différents objectifs.*

L'ensemble Pareto optimal regroupe des solutions dites **supportées** correspondants aux sommets de la fermeture convexe de la frontière et des solutions **non-supportées** n'appartenant pas à cettefermeture convexe.

Définition 3 : convexité

L'ensemble Ψ est dit convexe si tout segment joignant deux points quelconques de Ψ est inclus dans Ψ (*figure* 1.4).

$$y_1 \in \Psi \wedge y_2 \in \Psi \quad \Leftrightarrow \quad segment(y_1, y_2) \subset \Psi$$

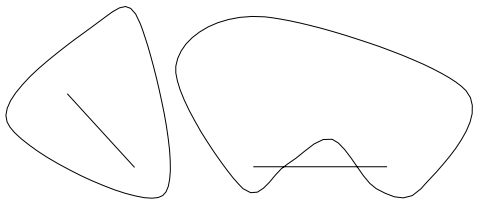

Figure 0.4 - *espace convexe (à gauche) et non convexe (à droite)*

La convexité est le premier indicateur de la difficulté du problème. En effet, certaines méthodes sont dans l'incapacité de résoudre des problèmes non convexes de manière optimale. Mais il existe d'autres indicateurs tout aussi importants, notamment la continuité, la multimodalité, la nature des variables de décision (entières ou réelles), . . .

Définition 4 : point idéal

Les coordonnées du point idéal correspondent aux meilleures valeurs de chaque objectif des points du front Pareto. Les coordonnées de ce point correspondent aussi aux valeurs obtenues enoptimisant chaque fonction objective séparément. Dans Ψ c'est le point de cordonnées (y_1^*,..., y^{p*}), avec $y_k^* = \max f_k(S)$, $S \in \Omega$ et $k = 1 \ldots p$. Ce point ne correspond pas à une solution réalisable car si c'était le cas, cela sous-entendrait que les objectifs ne sont pas contradictoires et qu'une solutionoptimisant un objectif, optimise simultanément tous les autres, ce qui ramènerait le problème àun problème ayant une seule solution Pareto optimale. Une visualisation de l'ensemble de ces définitions est donnée sur la figure 1.5.

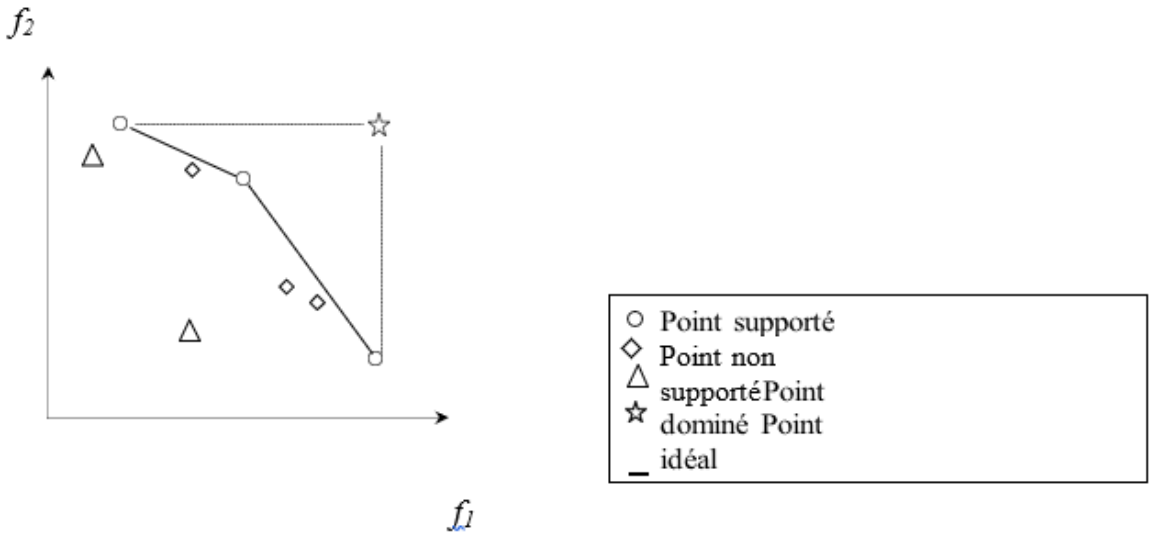

Figure 0.5 *- Points caractéristiques d'un problème de maximisation biobjectif*

L'équilibre souhaité entre l'intensification et la diversification

Deux objectifs doivent être pris en compte dans la résolution d'un problème d'optimisation multi objectif : **l'intensification** et **la diversification**.

- ✓ **Intensification** (exploitation) : converger vers la frontière Pareto.
- ✓ **Diversification** (exploration) : trouver les solutions diversifiées le long de la frontière Pareto.

1.3.3. Approches de résolution multi objectif

La résolution de problèmes multi objectifs relève de deux disciplines assez différentes. En effet,résoudre un problème multi objectif peut être divisé en deux phases :

1. La recherche des solutions de meilleur compromis : C'est la phase d'optimisation multi objectif.

2. Le choix de la solution à retenir : C'est la tâche du décideur qui, parmi l'ensemble des solutions de compromis, doit extraire celle(s) qu'il utilisera. On parle alors ici de décision multi objectif et cela fait appel à la théorie de la décision.

Dans le cadre de ce mémoire nous ne parlerons que de la première phase qui consiste en la recherche des solutions de meilleurs compromis. Dans les différentes

publications, nous rencontrons deux classifications différentes des approches de résolution de problèmes multi objectifs. Le premier classement adopte un point de vue décideur, les approches sont classées en fonction de l'usage que l'on désire en faire. Le deuxième classement adopte un point de vue concepteur, les approches sont triées de leur façon de traiter les fonctions objectives. Ainsi avant de se lancer dans la résolution d'un problème multi objectif, il faut se poser la question du type d'approche de résolution à utiliser.

Classification Point de vue décideur

On distingue à cet égard trois schémas possibles. Soit le décideur intervient dès le début de la définition du problème, en exprimant ses préférences, afin de transformer un problème multi objectif en un problème simple objectif. Soit le décideur effectue son choix dans l'ensemble des solutions proposées par le solveur multi objectif :

Les approches a priori : le décideur intervient en aval du processus d'optimisation, pour définir la fonction d'agrégation modélisant le compromis que l'on désire faire entre les différents objectifs. Dans ce cas le décideur est supposé connaître a priori le poids de chaqueobjectif afin de les mélanger dans une fonction unique. Cela revient à résoudre un problèmemono-objectif. Cependant dans la plupart des cas, le décideur ne peut pas exprimer clairement sa fonction d'utilité, parce que les différents objectifs sont non commensurables (exprimés dans des unités différentes).

Les approches interactives : combinent de manière cyclique et incrémentale les processus dedécision et d'optimisation. Le décideur intervient de manière à modifier certaines variables ou contraintes afin de diriger le processus d'optimisation. Le décideur modifie ainsi interactive- ment le compromis entre ses préférences et les résultats. Cette approche permet donc de bienprendre en compte les préférences du décideur, mais nécessite sa présence tout au long du processus de recherche.

Les approches a posteriori : cherche à fournir au décideur un ensemble de bonnes solutions bien réparties. Il peut ensuite, au regard de l'ensemble des solutions, sélectionner celle qui luisemble la plus appropriée. Ainsi, il n'est plus nécessaire de modéliser les préférences du décideur (ce qui peut s'avérer être très difficile), mais il faut en contrepartie fournir un ensemble de solutions bien réparties, ce qui peut

également être difficile et requérir un tempsde calcul important (mais ne nécessite pas la présence du décideur).

Nous nous placerons dans le cadre de cette troisième famille de méthodes où la modélisation despréférences n'est pas requise et où le procédé d'optimisation doit être puissant afin de fournir une très bonne approximation de la frontière Pareto.

1.3.4. Méthodes de résolution

En présence d'un problème d'optimisation concret, le chercheur est confronté à la principale difficulté du choix d'une méthode efficace (lorsqu'elles existent), capable de produire une solution optimale ; ou d'une méthode dont la qualité de la solution est acceptable, au prix d'un temps de calcul raisonnable. Face à ce souci un grand nombre de méthodes ont été développées pour tenter d'apporter une réponse satisfaisante à ces problèmes. Parmi celles-ci, nous distinguons les méthodes dédiées à un problème et les méthodes plus génériques pouvant s'appliquer à un ensemble de problèmes. Dans ce chapitre, nous essayons de faire un état d'art sur les différentes méthodes d'optimisation, on distingue deux grandes classes à savoir méthodes exactes et méthodes approchées. Les méthodes exactes examinent, souvent de manière implicite, la totalité de l'espace de recherche. Ainsi, elles ont l'avantage de produire une solution optimale lorsqu'aucune contrainte de temps n'est donnée. Néanmoins, le temps de calcul nécessaire pour atteindre une solution optimale peut devenir vite prohibitif. Les méthodes approchées constituent une alternative indispensable et complémentaire. Le but d'une telle méthode n'est plus de fournir une solution optimale au problème donné. Elle cherche avant tout à produire une solution sous-optimale de meilleure qualité possible avec un temps de calcul raisonnable. En général, une méthode approchée examine seulement une partie de l'espace de recherche.

Méthodes exactes

- **Méthode de pondération des fonctions objectif**

Cette approche de la résolution des problèmes d'optimisation multi objectif est la plus évidente. D'ailleurs, on appelle aussi cette méthode l'approche naïve de l'optimisation multi objectif. Le but, ici, est de revenir à un problème d'optimisation mono objectif, pour lequel existent de nombreuses méthodes de résolution. La manière la plus simple de procéder, consiste à prendre

chacune des fonctions objectives et leur appliquer un coefficient de pondération, ensuite faire la somme pondérée des fonctions objectif. On obtient alors une nouvelle fonction objective

$$opt\ F(x) = \sum_{i=1}^{k} w_i f_i$$

$$g(x) \leq 0$$

Les poids w_i sont compris entre 0 et 1 et $\sum_{i=1}^{k} w_i = 1$

- **Méthode de Keeney-Raiffa**

Cette méthode utilise le produit des fonctions objectif pour se ramener à un problème d'optimisation mono objectif. L'approche utilisée ici est semblable à celle utilisée dans la méthode 28 Méthodes de résolution de pondération des fonctions objectif. La fonction objective ainsi obtenue s'appelle la fonction d'utilité de Keeney-Raiffa.

$$opt\ F(x) = \prod_{i=1}^{k} w_i f_i(x)$$

$$g(x) \leq 0$$

Les poids w_i sont compris entre 0 et 1 et $\sum_{i=1}^{k} w_i = 1$

- **Méthode de compromis (L'approche par ε -contrainte)**

Une autre façon de transformer un problème d'optimisation multi objectif en un problème mono objectif, est de convertir m^{-1} des m objectifs du problème en contraintes et d'optimiser séparément l'objectif restant. La démarche est la suivante :

a. On choisit un objectif initial (prioritaire) à optimiser.

b. On choisit un vecteur de contraintes initial.

c. On transforme le problème, en conservant l'objectif prioritaire et transformer les autres objectifs en contraintes d'inégalité.

On appelle aussi cette méthode la méthode de la ε -contrainte. Le problème peut être reformulé de la manière suivante :

$$
\begin{cases}
opt\ f_i(x) \\
f_1(x) \le \varepsilon_1 \\
f_2(x) \le \varepsilon_2 \\
\quad \vdots \\
f_{i-1}(x) \le \varepsilon_{i-1} \\
f_i(x) \le \varepsilon_1 \\
\quad \vdots \\
f_m(x) \le \varepsilon_m \\
g(x) \le 0
\end{cases}
$$

L'approche par ε -contrainte doit aussi être appliquée plusieurs fois en faisant varier le vecteur ε, qui doit être choisit judicieusement, pour trouver un ensemble de points Pareto Optimaux. En transformant des fonctions objectives en contraintes, elle diminue la zone réalisable par paliers. Ensuite, le processus d'optimisation trouve le point optimal sur l'objectif restant.

- **Méthode lexicographique**

Cette méthode est très intuitive. Elle consiste à considérer les fonctions objectives une après l'autre et minimiser un problème d'optimisation mono objectif. Au fur et à mesure, on lui ajoute graduellement des contraintes.

Présentation de la méthode

Nous commençons à partir du problème P. Nous procédons en k étapes (il y a autant d'étapes qu'il y a de fonctions objectif). Commençons par la première fonction objective. Nous résolvons :

$$(p) \left\| \begin{array}{l} min\ f_1(x) \\ \\ g(x) \leq 0 \end{array} \right.$$

Nous notons par f_1^* la solution de ce problème. Après résolution, nous transformerons la première fonction objective en contraintes d'égalité. De ce fait, nous prenons la seconde fonction objective et on résout le problème. Nous répétons cette approche jusqu'à atteindre les k fonctions objectifs. Pour finir, nous obtenons le problème ci-dessous à résoudre :

$$\left\| \begin{array}{l} min\ f_k(x) \\ f_1(x) = f_1^*, f_1^*, \dots, f_1^* \\ g(x) \leq 0 \end{array} \right.$$

La dernière valeur trouvée de x est celle qui réduit au minimum toutes les fonctions objectives.

Méthodes Approchées

Certains problèmes d'optimisation demeurent cependant hors de portée des méthodes exactes. Un certain nombre de caractéristiques peuvent en effet être problématiques, comme l'absence de convexité stricte (multimodalité), l'existence de discontinuités, une fonction non dérivable, présence de bruit, etc. Dans de tels cas, le problème d'optimisation est dit difficile, car aucune méthode exacte n'est capable de le résoudre exactement en un temps raisonnable, on devra alors faire appel à des heuristiques permettant une optimisation approchée. Ces méthodes sont, en général, présentées sous la forme de concept d'inspiration. Plusieurs définitions ont été données pour clarifier le concept de l'heuristique, parmi lesquels on trouve les définitions suivantes :

Définition : (Heuristique selon Reeves)

Une heuristique est une technique trouvant de bonnes solutions (c'est-à-dire proches de l'optimum) pour un coût de calcul raisonnable, sans pouvoir garantir l'admissibilité ou l'optimalité, ou même dans de nombreux cas, préciser la distance à l'optimum d'une solution particulière.

Métaheuristiques

Parmi les heuristiques, certaines sont adaptables à un grand nombre de problèmes différents sans changements majeurs dans l'algorithme, on parle alors de métaheuristiques.

La plupart des heuristiques et des métaheuristiques utilisent des processus aléatoires comme moyens de récolter de l'information et de faire face à des problèmes comme l'explosion combinatoire. En plus de cette base stochastique, les métaheuristiques sont généralement itératives, c'est-à-dire qu'un même schéma de recherche est appliqué plusieurs fois au cours de l'optimisation, et directes, c'est-à-dire qu'elles n'utilisent pas l'information du gradient de la fonction objectif. Elles tirent en particulier leur intérêt de leur capacité à éviter les optimas locaux, soit en acceptant une dégradation de la fonction objective au cours de leur progression, soit en utilisant une population de points comme méthode de recherche (se démarquant ainsi des heuristiques de descente locale).

Définition (Métaheuristique selon Osman et Laporte)

Une métaheuristique est un processus itératif de génération guidant une heuristique subordonnée en combinant intelligemment différents concepts pour explorer et exploiter l'espace de recherche en utilisant des stratégies pour structurer l'information de manière à trouver efficacement des solutions proches de l'optimum.

- ## Algorithmes génétiques

Les AGs ont été largement utilisés dans la communauté multi objectif. Ils sont très appropriés pour résoudre des PMOs grâce à l'utilisation d'une population de solutions. Les AGs peuvent chercher plusieurs solutions Pareto-Optimales dans la même exécution.

Vector Evaluated Genetic Algorithm (VEGA)

Le premier algorithme génétique (AG) proposé dans la littérature pour résoudre des PMOs est l'algorithme VEGA développé par Schaffer [33]. VEGA sélectionne les individus de la population courante suivant chaque objectif séparément.

A chaque génération, la population est divisée en m sous-populations, où m est le nombre d'objectifs à optimiser. Chaque sous-population est sélectionnée suivant son l'objectif. Les m sous-populations sont ensuite mélangés pour construire une nouvelle population, à laquelle sont appliqués les opérateurs génétiques (croisement et mutation).

Dans l'algorithme VEGA la méthode de sélection ne tient compte que d'un seul objectif, elle favorise les meilleurs individus par rapport à cet objectif. Ainsi, ces individus ne seront sélectionnés que lorsque la sélection est effectuée sur cet objectif. Les individus ayant une performance générale acceptable, appelé par Schaffer les individus milieu, vont être éliminés car ils ne seront sélectionnés dans aucune sous-population. Ceci empêche la méthode d'atteindre les solutions compromis.

Les AGs présentés dans la suite utilisent la notion de dominance de Pareto dans la sélection des solutions générées.

- **MultiObjective GA (MOGA)**

MOGA [Fonseca & Fleming, 1993], proposé par Fonseca et Fleming, est le premier algorithme qui utilise directement la notion de dominance de Pareto. L'algorithme utilise une procédure de ranking dans laquelle, pour chaque solution i , le nombre n_i de solutions la dominant est calculé, et le rang $r_i = (1 + n_i)$ lui est associé. Ainsi, le rang de chaque solution non-dominée est égal à 1 et le rang maximal ne peut pas être plus grand que la taille de la population N . La performance (fitness) F_i de chaque individu est basée sur son rang. La performance est calculée de sorte que tous les individus du même rang aient la même performance et que cette performance soit à maximiser.

- **Niched-Pareto Genetic Algorithm (NPGA)**

Horn et al [Horn et al. , 1994] ont proposé en 1994 la méthode NPGA. Cette méthode utilise une nichage (un ensemble d'individus situés dans un espace restreint) mis à jour dynamiquement [Horn et al. , 1994], et une sélection par tournoi. Une paire de solutions i et j est choisie aléatoirement de la population P. Ensuite ces deux solutions sont comparées au sens de la dominance à une sous-population T choisie aussi aléatoirement :

- ✓ Si une des solutions $i$ou $j$domine toutes les solutions de T alors que l'autre solution est dominée par au moins une solution de T, alors la première gagne le tournoi (elle est choisie).
- ✓ Si les deux solutions i et j sont soit dominées par au moins une solution de T soit dominent tout l'ensemble T, alors elles sont mises dans la population des enfants Q et leur compteur de niche (niche count) dans Q est calculé. La solution qui a le compteur de niche le plus petit gagne le tournoi.

- **Le Strength Pareto Evolutionary Algorithm 2 (SPEA2)**

L'algorithme SPEA2, proposé par Zitler et al , a été développé comme une amélioration de SPEA [Zitzler & Thiele, 1999]. SPEA est un algorithme élitiste qui maintient une population externe, appelée aussi archive, contenant le meilleur front de compromis rencontré durant la recherche. Cette population est destinée à contenir un nombre limité de solutions non dominées trouvées depuis le début de l'exécution. A chaque itération de nouvelles solutions non dominées sont comparées aux individus de cette population au sens de dominance et seules les solutions non dominées résultantes restent. Il est à noter que SPEA non seulement préserve l'élite mais aussi la fait participer aux opérations génétiques.

Une procédure de clustering est appliquée à cette population externe pour réduire sa taille si elle dépasse la taille limite. SPEA2 diffère de son prédécesseur en plusieurs aspects. D'abord, la taille de l'archive est fixée, c'est à dire que s'il n'y a pas suffisamment d'individus non dominés, l'archive est complétée par ceux dominés. Notons qu'une telle situation ne peut se produire que durant les premières générations et durant une période relativement courte.

Nous allons noter dans ce qui suit P la population courante de taille N, et P' l'archive de taille N' .

Le calcul de performance est plus raffiné que celui de SPEA, au sens qu'il tient plus compte de la densité des solutions. Pour attribuer une valeur de fitness à un individu p de la population P et P', l'algorithme utilise la règle suivante :

$$fitness(p) = \mathrm{R}(p) + \mathrm{D}(p)$$

Dans cette équation, $\mathrm{R}(p)$est la performance préliminaire (raw fitness) de l'individu p, et $\mathrm{D}(p)$ est sa densité. La performance préliminaire est calculée suivant des valeurs de Strength des solutions. $\mathrm{D}(p)$ est calculée en utilisant une adaptation de la technique du k^{ime} plus proche voisin 33 Méthodes de résolution [34]. Les modifications apportées répondent à certaines critiques soulevées par SPEA.

La procédure de clustering qui contrôlait la taille de l'archive dans SPEA est remplacée dans SPEA2 par une méthode de troncation qui, contrairement au clustering préserve les individus situés aux extrémités du front Pareto. Une autre différence importante est que dans SPEA2 seuls les membres de l'archive participent au processus de reproduction.

o Le recuit simulé

Le premier algorithme multi objectif basé sur le recuit simulé a été proposé par Sérafini en 1992 [Serafini, 1992]. La méthode utilisée le schéma standard du recuit simulé avec une seule solution courante. Le résultat de l'algorithme est un ensemble de solutions Pareto optimales contenant toutes les solutions non dominées générée par l'algorithme.

Serafini considère un certain nombre de d'acceptation définissant la probabilité d'acceptation des nouvelles solutions voisines. Le recuit simulé uni objectif accepte avec une probabilité égale à 1 les nouvelles solutions meilleure ou égale à la solution courante elle acceptée avec une solution inférieure à 1. Dans le cas multi objectif, trois situations sont possibles en comparant une nouvelle solution x'aves la solution courante x:

a. x' domine$x$

b. x' dominée par x

c. x' et x sont mutuellement non dominées

Dans le premier cas, il est évident d'accepter x' avec une probabilité égale à 1. Dans le deuxième cas, la probabilité d'acceptation doit être inferieure à 1. La question se pose pour la troisième situation quelle doit être la probabilité d'acceptation ? Sérafini propose des règles d'acceptations multi objectif qui traite différemment la troisième situation. Ces règles correspondent à certaines fonctions d'agrégation locales.

Ulungu et al [Ulungu et al. , 1999] ont proposé en 1999 une méthode appelée MOSA utilise un nombre de vecteurs poids prédéfinis. Chaque vecteur est associé à un processus de recuit indépendant.

Chaque processus commence d'une solution aléatoire ou d'une solution construite par une heuristique spécialisée. Ensuite la solution réalise des mouvements qui sont acceptées avec une probabilité définie par une règle d'acceptation. Le résultat de l'algorithme est un ensemble de solutions Pareto optimales contenant toutes les solutions non dominées générées par tous les processus de recuit. Ainsi ses processus qui travaillent indépendamment occupèrent dans la génération d'un ensemble commun de solutions Pareto optimales [39].

Czyzak et Jaszkiewicz [Czyzak & Jaszkiewicz, 1998] ont proposés en 1998 la méthode de Pareto Simulated annealing cette méthode utilise les fonctions scalaires basées sur les probabilités pour l'acceptation de nouvelles solutions voisines. Une population de solutions est

utilisée, 34 Méthodes de résolution chacune d'elles explorant l'espace de recherches relativement aux règles de recuit simulé. Les solutions peuvent être traitées comme des agents travaillant indépendamment mais échangeant les informations sur leurs positions [9].

Chaque solution est associée à un vecteur poids séparé. La méthode met à jour les vecteurs poids des solutions en construction en utilisant une règle pour assurer la dispersion des solutions sur toutes les régions de front Pareto.

Suppapitnarm et Parks [Suppapitnarm & Parks, 2001] ont proposé une méthode dans laquelle la probabilité d'acceptation d'une nouvelle solution dépend du fait qu'elle est ajoutée ou non à l'ensemble des solutions Pareto-Optimales potentielles.

Si elle est ajoutée à cet ensemble, ce qui signifie qu'elle n'est dominée par aucune autre solution trouvée, elle est acceptée pour être la solution courante avec une probabilité égale à 1. Sinon, une règle d'acceptation multiobjectif est utilisée.

- **Recherche tabou**

Grandibleux et al [Gandibleux et al. , 1997] ont proposé une version multi objectif de la recherche taboue. La méthode utilise des fonctions pondérées scalaires dont les poids sont changés périodiquement. La modification du vecteur poids dégrade les poids des objectifs qui ont été nettement améliorés. Deux listes taboues sont utilisées. La première est une liste taboue régulière qui empêche de revenir aux solutions déjà visitées. La deuxième contient les vecteurs poids.

Hansen [Hansen, 1998] a proposé une recherche taboue basée sur l'idée de la méthode Pareto simulated annealing. La méthode utilise une population de solutions explorant chacune d'elles différentes régions de l'ensemble Pareto. Le vecteur poids est utilisé dans une fonction scalaire. De plus, à chaque solution est associée une liste taboue. La dispersion des solutions est assurée par la modification de leurs vecteurs poids. Pour chaque solution, la modification du vecteur poids vise à la déplacer des autres solutions proches dans l'espace des objectifs. Hansen a appliqué cette méthode au problème du sac à dos multi objectif.

Ben Abdelaziz et Krichen [Ben Abdelaziz & Krichen, 1997] ont proposé un algorithme de recherche taboue multi-objectif conçu spécialement au problème du sac à dos multi-objectif. La méthode est guidée par des fonctions scalaires pondérées linéaires et par une relation de dominance. La relation de dominance est appliquée non seulement aux solutions mais aussi aux

objets. La recherche locale est basée sur l'idée de l'insertion des objets non-dominés au sac à dos.

o Approches hybrides

Plusieurs approches hybrides ont été proposées dans la littérature essayant de tirer profit des avantages de chacune des méthodes utilisées ou bien pour combler certaines de ses lacunes. Plusieurs travaux ont proposé des méthodes hybrides pour résoudre des PMOs. Jaszkiewicz a proposé l'algorithme Multi-Objective Genetic Local Search (MOGLS) qu'il a appliqué au problème de voyageur de commerce multi-objectif et au problème du sac à dos multi-objectif Cet algorithme utilise une méthode de descente pure comme opérateur de recherche locale. A chaque itération, l'algorithme construit une fonction d'utilité aléatoire ainsi qu'une population temporaire composée d'un nombre de meilleures solutions à partir des solutions générées. Ensuite, une paire de solutions sélectionnée aléatoirement recombinée. La recherche locale est appliquée à chaque solution générée.

Barichard et Hao [Barichard et Hao, 2003] ont proposé l'algorithme GT SMOKP pour résoudre le problème du sac à dos multi-objectif. Cet algorithme combine une procédure génétique avec un opérateur de recherche tabou.

Récemment, plusieurs algorithmes évolutionnaires qui intègre un calcul d'indicateur d'hypervolume ont été proposés dans la littérature.

o La méthode métaheuristique de Monté Carlo

Les méthodes Monté Carlo consistent en des simulations expérimentales ou informatiques des problèmes mathématiques ou physiques, basées sur le tirage des nombres aléatoires. Généralement on utilise en fait des séries de nombres pseudo-aléatoires générées par des algorithmes spécialisés. Les propriétés de ces séries sont très proches de celles d'une véritable suite aléatoire.

Le grand avantage de cette méthode est sa simplicité. Elle permet entre autres de visualiser l'effet de différents paramètres et de donner ainsi des orientations, d'étudier des structures intéressantes qui auraient été a priori écartées et de trouver facilement des structures que l'on n'aurait pas aussi bien optimisées à la main. Les méthodes de types Monté Carlo recherchent

l'optimum d'une fonction en générant une suite aléatoire de nombres en fonction d'une loi uniforme.

o Pareto Ant Colony Optimization (P-ACO)

A été proposé par Doerner [Doerner.2004]. Il a été initialement appliqué au problème de sélection de portefeuille multi-objectif. Il suit le schéma général de l'algorithme Ant System (AS).

Dans la phase initiale de cet algorithme, on commence d'abord par générer Γ fourmis sachant que chaque fourmi a un portefeuille vide $x = (0, \dots, 0)$ qu'elle doit remplir au fur et mesure. La durée de vie Ξ est déterminée aléatoirement dans l'intervalle $[1, \dots, N]$ pour chaque fourmi.

A chaque cycle de l'algorithme, chaque fourmi calcule un ensemble de poids

$P = (P_1, P_2, \dots, P_K)$ et l'utilise pour combiner les traces de phéromone et l'information heuristique afin de construire un portefeuille faisable x, en appliquant une règle basée sur l'information heuristique η_i et sur la structure phéromone τ_i . Une fois le portefeuille est construit, l'algorithme test test la faisabilité et l'efficacité c'est-à-dire : si le portefeuille construit est faisable et efficace il sera stocké dans un ensemble externe.

A la fin, la mise à jour globale de la phéromone est réalisée en utilisant deux fourmis, la meilleure et la deuxième meilleure solution générée dans le cycle courant pour chaque objectif.

2. CHAPITRE 02 : ALGORITHMES GENETIQUES

2.1. Introduction

L'idée de l'imitation du processus biologique de l'évolution pour la résolution des problèmes d'optimisation a été inventée par John Holland et exposée dans son ouvrage "Adaptation in natural and artificial systems," en 1975, connue aujourd'hui sous le nom "Algorithmes Génétiques" et devenue populaire dans le domaine d'optimisation grâce à Goldberg (1989). Les algorithmes génétiques sont basés sur l'évolution naturelle (Darwin, 1859), le principe de ce processus est la survie du plus adapté. Les éléments concernés par un processus évolutionnaire sont sujets d'une sélection qui ne gardera à la fin que ceux qui présentent les meilleures facultés d'adaptation. Les organismes biologiques sont un exemple d'adaptation et d'optimisation car ils ont réussi à s'adapter et à survivre l'environnement naturel pendant des millions d'années. Holland dans son ouvrage (1975) a voulu traduire ce processus d'évolution et d'adaptation mathématiquement afin de pouvoir optimiser des problèmes qu'on rencontre dans la majorité des domaines, l'économie, la médecine, l'intelligence artificielle et les réseaux neuronaux etc.

Dans ce premier chapitre, nous allons porter lumière sur les algorithmes génétiques multi objectifs et expliquer le fonctionnement de base ainsi que les notions relatives à ces algorithmes, un exemple pratique d'un AG est donné dans le chapitre qui suit.

2.2. Algorithmes génétiques

Les AGs [19] font partie des algorithmes méta-heuristiques, ils travaillent sur un espace de recherche sous forme de population de pseudo-solutions en y opérant d'une manière stochastique sans qu'il y ait de contraintes de continuité ni de différentiable de la fonction à optimiser, ce qui est d'ailleurs un des points qui différencie les AGs des autres méthodes classiques. Les caractéristiques qui font l'unicité de cette méthode sont :

1- Les AGs ne travaillent pas directement sur les paramètres, ces derniers sont codés avant l'opération.

2- Ils opèrent sur une population de solutions au lieu d'une seule solution.

3- Les AGs n'utilisent que les valeurs de la fonction à étudier, sa dérivée ou autres connaissances ne sont pas prises en considération.

4- Les AGs se basent sur des données aléatoires, donc ils sont probabilistes.

Les AGs utilisent les éléments suivant pour se définir :

- Individu/chromosome/séquence : une potentielle solution du problème abordé.

- Population : un ensemble de chromosomes de l'espace de l'exploration.
- Environnement : l'espace de l'exploration et de la recherche.
- Fonction d'évaluation (fitness) : la fonction objective qui sert à évaluer la précision de la solution du problème.

2.3. Fonctionnement des algorithmes génétiques

Semblables au processus naturel de l'évolution, les algorithmes génétiques comportent des phases à travers lesquelles une solution potentielle est estimée :

1- **Initialisation** : Un ensemble de N chromosomes est aléatoirement créé formant une population de solutions potentielles.
2- **Evaluation** : Décodage puis évaluation de chaque individus/chromosomes.
3- **Reproduction** : Après l'évaluation, une nouvelle population de N individus est créée à l'aide d'une méthode de sélection adéquate.
4- **Opérations génétiques** : Croisement et mutation de certains individus au sein de la nouvelle population.
5- **Retour** : Tant que la condition d'arrêt n'est pas satisfaite, l'algorithme recommence à partir de la phase d'évaluation**.**

Un AG génère une population en utilisant une représentation chromosomique des individus puis à l'aide de la fonction objectif (fitness), il évalue chaque individu et sélectionne les mieux adaptés pour former une nouvelle population de solutions potentielles à travers les opérateurs génétiques (croisement, mutation) là où se passe la reproduction afin de créer de nouveaux individus au sein de la nouvelle population

À travers ces phases, le mécanisme de la sélection naturelle est simulé, afin d'assurer la survie, les AGs gardent à chaque fois les meilleurs individus en écartant les plus faibles.

Représentation chromosomique (Codage)

Au début, chaque individu (chromosomique) de la population est codé soit en binaire/gray [27], ou en réel [28], représenté par un vecteur dont la longueur dépend de la précision requise, le codage donne à l'algorithme génétique une représentation génotype-phénotype.

Ce processus est illustré dans l'organigramme suivant :

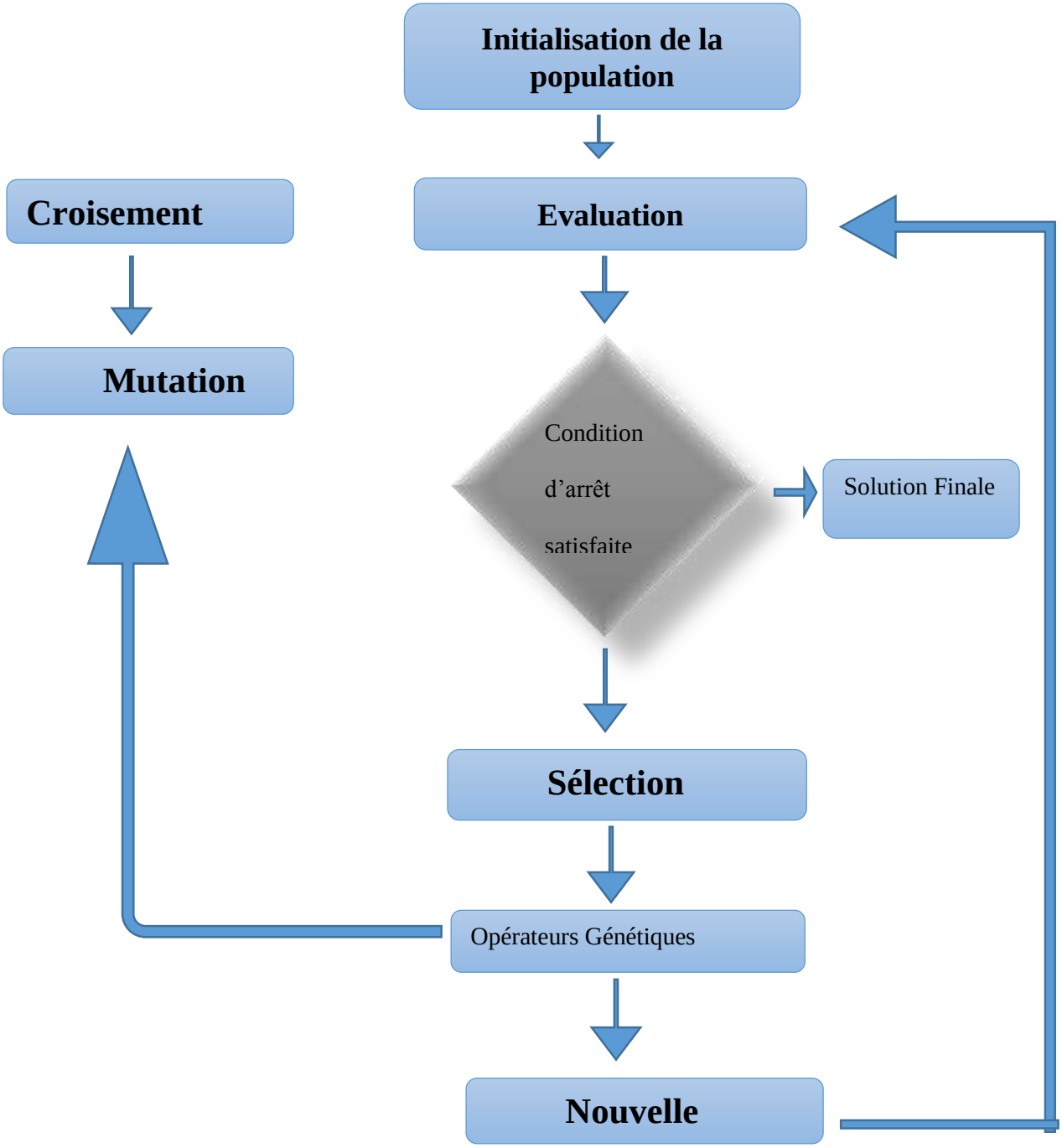

Figure 2.1 *fonctionnement de l' algorithme génétique*

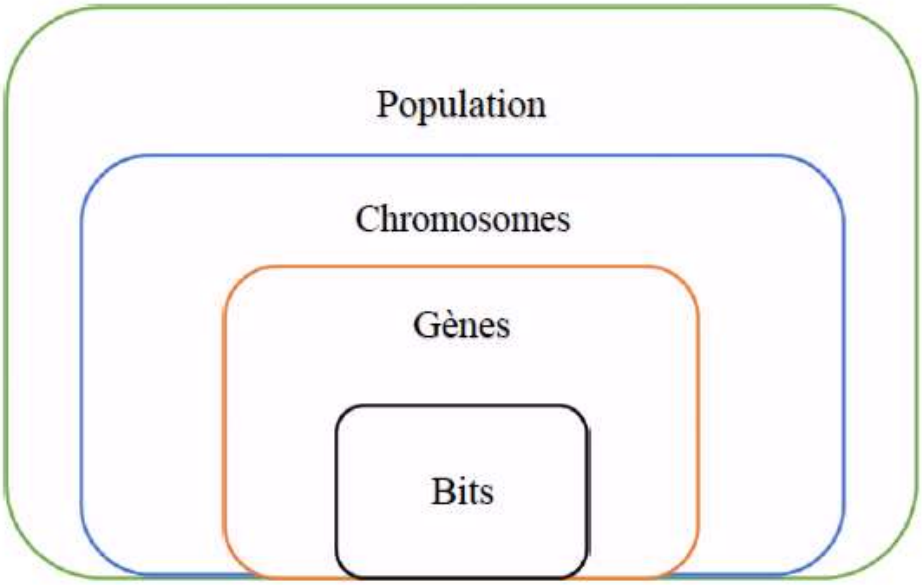

***Figure 2.2** - Représentation Chromosomique*

La rapidité de la convergence d'un AG peut être affectée par le choix initial de la population, car si des renseignements à priori sur le problème sont disponibles, l'orientation du choix de la population peut considérablement améliorer la rapidité du processus de résolution.
Le codage des paramètres d'un AG dépend du problème à optimiser, le codage binaire, gray et réel sont les plus utilisé dans le domaine.

2.3.1.1. Codage binaire

Ce codage est caractérisé par sa simplicité car il ne comporte que deux caractères (0 et 1), chaque individu est représenté par un chromosome et chaque caractéristique de l'individu est un gène. Le chromosome est codé en une suite de bits appelée chaîne binaire, ceci est illustré à travers l'exemple suivant :
Trois variables (203, 13, 45) codées sur 8 bits formants une chaine binaire de taille 24 :

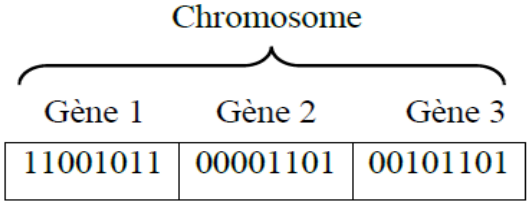

***Figure 2.3** - Codage Binaire de trois variables*

2.3.1.2. Codage réel

L'ensemble des variables est représenté par un vecteur $\bar{x}$= <x_1, x_2,……. x_n> où chaque x_i est un nombre réel. Les opérateurs de croisement et de mutation seront alors soigneusement définis afin de conserver chacune des variables dans son domaine, la plupart de ces opérateurs s'inspirent des opérateurs du codage binaire.

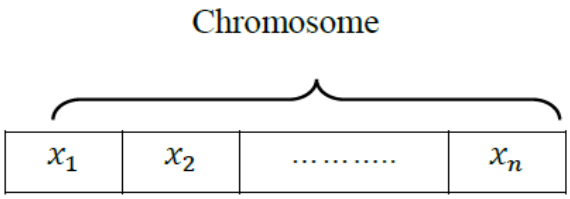

***Figure 2.4** - Codage Réel*

2.3.1.3. Codage gray

Le codage gray à la propriété que deux points adjacents (n et n+1) dans l'espace étudié diffèrent seulement d'un seul bit, autrement dit, une incrémentation d'un pas dans la valeur du paramètre correspond à un changement d'un seul bit dans le code, ce qui est un avantage par rapport au codage binaire qui en termes de distance de Hamming ne code pas forcément deux éléments voisins dans l'espace d'exploration. Le tableau suivant illustre cette différence

Valeur entière	Binaire	Gray
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111

***Tableau 2.1** - Codage Binaire/Gray*

Evaluation

A l'aide d'une fonction objectif (Fitness), l'AG sera en mesure de classer les individus d'une population selon leurs pertinences, elle est une partie inséparable de l'AG car elle représente (sous forme mathématique) le problème à résoudre.

En général, l'algorithme génétique essai de minimiser la fonction de fitness en fonction d'un vecteur ou

d'une valeur issue de la population, la solution est donc jugée selon son image par la fonction de fitness. Il existe des fonctions de test qui permettent de mettre en épreuve l'efficacité de l'algorithme génétique utilisé, certaines de ces fonctions seront utilisées ultérieurement dans ce chapitre.

Sélection

C'est dans cette étape où se décide quels sont les individus de la population qui vont être dupliqués dans la nouvelle population où ils deviendront des parents. Les individus ayants les meilleurs fitness auront donc plus de chance d'être sélectionnés. Il existe essentiellement quartes méthodes de sélection, leur utilisation dépend du problème étudié :

- La méthode de la "loterie biaisée" (roulette wheel) [27] ;
- La méthode de rang [29] ;
- La sélection par tournoi [30] ;
- La sélection universelle stochastique [31].

2.3.1.4. Méthode de la loterie biaisée

C'est la méthode la plus utilisée, elle consiste à dupliquer des individus de la population dont les performances sont relativement bonnes.

Pour un chromosome dont le fitness est (ch_i), la probabilité (ch_i) avec laquelle il sera dupliqué dans la nouvelle population de taille N est :

$$p(ch_i)=\frac{\mathrm{f}(ch_i)}{\sum_{j=1}^{N}\mathrm{f}(ch_j)}$$

La figure suivante illustre le principe de ce type de sélection, en utilisant une population de quatre individus, l'individu 1 avec la probabilité (ch_i)=0.37 a plus de chance d'être sélectionné que les autres.

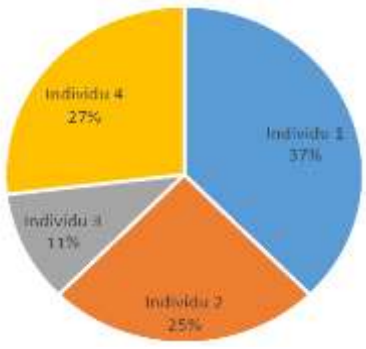

***Figure 2.5** - Méthode de sélection de la loterie biaisée*

2.3.1.5. Méthode de sélection par rang

Le principe de cette méthode est de garder les meilleurs individus (selon leurs fitness) d'une population

P_t pour ensuite les introduire dans la prochaine population p_{t+1}. C'est une façon de protéger des solutions potentielles de la disparition lors de la phase de sélection. C'est une méthode performante, mais dans certains cas, le manque de diversité dans les solutions provoque la convergence prématurée ce qui représente l'inconvénient majeur de cette méthode de sélection.

2.3.1.6. Méthode de sélection par tournoi

Cette méthode est une forme de duel entre deux individus, celui qui domine l'autre fonction de son fitness est sélectionné pour être réintroduit dans la nouvelle population, le perdant est écarté. L'opération est répétée jusqu'à ce que la nouvelle population de n individus soit pleine.

2.3.1.7. Méthode « stochastique du reste »

Cette méthode repose principalement sur la fonction d'évaluation et sa moyenne numérique, chaque individu d'une population est dupliqué nai fois avec nai la valeur de la partie entière de $\left(\frac{f_i}{f_{moy}}\right)$. Chaque individu se fait associer une probabilité de sélection :

$$\boldsymbol{p_s(a_i) = \left(\frac{f_i}{f_{moy}}\right) - E\left(\left(\frac{f_i}{f_{moy}}\right)\right)} \qquad \textbf{(II\_2)}$$

Avec (x) La partie entière de x.4.

Opérateurs génétiques

2.3.1.8. Croisement

Pendant cette opération, la population est divisée en deux sous-populations, ensuite, l'opération de reproduction est appliquée sur chaque individu de la première sous-population avec un autre de la deuxième sous-population, chaque couple de parents forme deux nouveaux chromosomes (enfants) dont les caractéristiques sont issues des deux parents. Le taux de croisement est défini en général entre 0.5 et 0.9, un taux excessivement élevé peut engendrer la perte de bonnes structures, mais aussi un taux faible provoque la stagnation puisque l'exploration est diminuée.

Deux méthodes classiques sont à mentionner, le croisement en un point et le croisement en deux points :

Chaque chromosome de longueur "l" est divisé en segments : deux segments pour le croisement en un seul point [27] et trois segments pour le croisement en deux points [32]. La position de chaque chromosome est tirée aléatoirement ensuite les gênes des individus parents sont échangés selon une probabilité de croisement $p_c$pour former deux nouveau chromosomes.

Les schémas ci-dessous illustrent les deux méthodes :

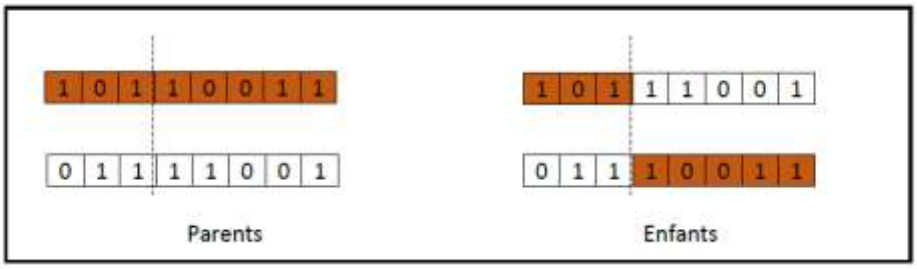

Figure 2.6 *- Croisement en un seul point*

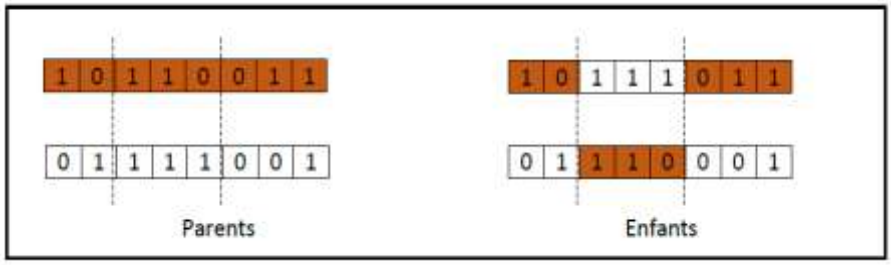

Figure2.7 *- Croisement en deux points*

2.3.4.2. Mutation

Cet opérateur tire au hasard un ou plusieurs gênes d'un individu, puis les modifie aléatoirement à condition que cet individu reste une solution possible au problème.

Chaque bit d'un individu a une probabilité $p_m$de subir une mutation, un réel $r \in \{0,1\}$ est généré, si r < p_m le bit sera donc remplacé par complémentaire [27].

Figure2.8 *- Mutation d'un chromosome*

L'opérateur de mutation rapporte de la diversité à la population, en évitant la convergence prématurée et donc l'AG peut atteindre la propriété $d'ergodicité^2$, c'est-à-dire que l'AG peut atteindre tous les points de l'espace de recherche et donc l'optimum global.

2.4. Non dominated Sorting Genetic Algorithm (NSGA II)

En raison de son efficacité reconnue, nous avons porté notre choix sur la mise en œuvre et l'implémentation de l'algorithme génétique NSGA-II (Non-dominated Sorting Genetic Algorithm) dont le principe de fonctionnement est représenté sur la figure 2.8.

Dans NSGA II, une population initiale Pt de taille N est choisie aléatoirement, les solutions formant cette population sont triées à base dominance, on garde uniquement les solutions équivalentes (les plus performantes) de P_t ensuite on applique les opérateurs génétiques (croisement et mutation) afin de générer une nouvelle population Q_t de taille N.
Les deux populations sont alors combinées ensemble pour former la population R_t de taille 2N. Cette dernière population est à son tour triée à base de dominance et décomposés en différents fronts de telle sorte que le premier front soit constitué uniquement de solutions équivalentes de rang 1, le deuxième front est à son tour composé des solutions équivalentes de rang 2 et ainsi de suite.
A ce stade, l'algorithme sélectionne N solutions à partir de ces fronts tout en commence bien sûr par sélectionner les solutions de rang 1 suivi par les solutions de rang 2 et ainsi de suite, jusqu'à ce qu'on obtienne une population de taille N. Il est à noter que les fronts qui ne constituent pas cette dernière population sont supprimés et les solutions constituant le dernier front utilisé sont toutes triées entre elles à base de la distance de crowding et ce afin de pouvoir sélectionner les nombres de solutions manquant pour avoir une population P_{t+1} de taille N.
Les opérateurs génétiques sont à nouveau appliqués sur P_{t+1} afin de générer une nouvelle population Q_{t+1} .

Le NSGA II est basé sur une classification des individus en plusieurs niveaux. En effet, il utilise une approche élitiste qui permet de sauvegarder les meilleures solutions trouvées lors des générations précédentes (préservation de la diversité). Il ne nécessite pas beaucoup de réglages de paramètres et il utilise un opérateur de comparaison basé sur un calcul de la distance de crowding, l'individu avec la distance de crowding la plus grande est sélectionné.
A l'aide des opérations décrites antérieurement, le NSGA II pourra donc effecteur une recherche rapide, élitiste et moins couteuse.

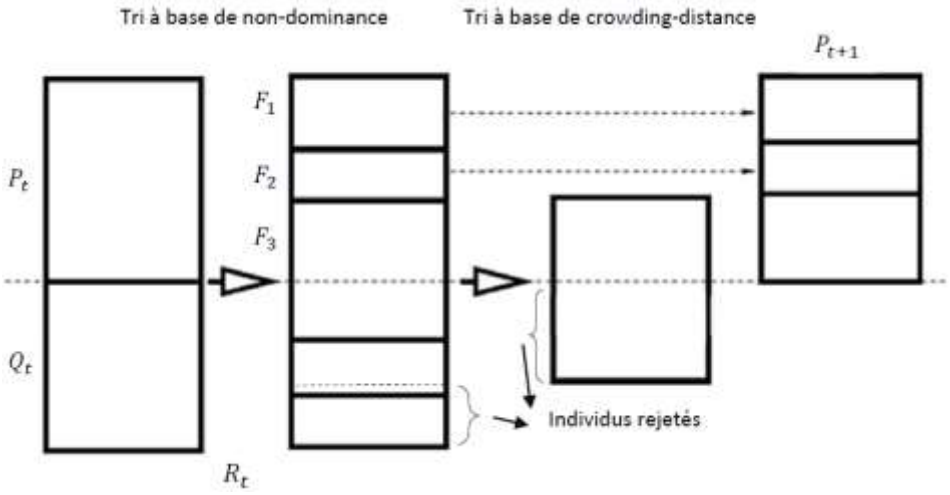

Figure 2.9 - *Fonctionnement du NSGA II*

$NSGA{-}II(N,g)$
$P_0 = initialisation{-}de{-}la{-}population\ (N)$
$Evaluation\ des\ objectifs$
$tri{-}non{-}dominance(P_0)$
$générer\ une\ population\ d'enfants\ Q_0$
$sélection\ par\ tournoi\ binaire$
$mutation\ et\ recombination$
$pour\ t{=}1\ jusqu'à\ g$
$R_t = P_t \cup Q_t$ Combiner les 2 populations
$\{F_1, F_1..F_k\} = tri{-}non{-}dominanc(R_t)$ Trier R_t
$P_{t+1} = \emptyset$ Initialiser P_{t+1}
$i{=}1$ Initialiser le compteur
$tant\ que\ N{-}|P_{t+1}| \geq |F_i|$ Tant que P_{t+1} est incomplète
$P_{t+1} = P_{t+1} \cup F_i$ inclure les fronts F_i dans P_{t+1}

$i = i+1$	incrémenter *i* de 1
$Fin\ tant\ que$	
$si\ N-\|P_{t+1}\|<\|F_i\|$ & $N-\|P_{t+1}\|\neq 0$	Si dernier front F_i est trop large
$assignement-distance-crowdin(Fi)$	attribuer les distances
$tirer\ (F_i,<n)$	opérateur de crowding
$P_{t+1}=P_{t+1}\cup F_i\ [1:N-\|P_{t+1}\|]$	inclure une partie de $F_i \cup P_{t+1}$
$sinon$	Si P_{t+1} est complète
$Q_{t+1}=créer-nouv-pop\ (P_{t+1})$	créer une nouvelle population
$t+1$	Prochaine génération

Pseudocode 1 - NSGA II

L'algorithme suivant permet de calculer et attribuer une distance de crowding à chaque solution dans un ensemble *I*

$assignement-distance-crowding(I)$	
$l=\|I\|$	Le nombre de solutions dans *I*
$pour\ chaque\ i, ettre$ $\mathrm{I[i]_{distance}}=0$	Initialisation de la distance
$pour\ chaque\ objectif\ m$	
$I=tire(I,m)$	Tirer les solutions en fonction de leurs coûts
$\mathrm{I[1]_{distance}}=\mathrm{I[l]_{distance}}=\infty$	Les solutions extrêmes toujours sélectionnées
$pour\ i=2\ jusqu'$à $(l-2)$	Pour toutes les autres solutions
$a=\mathrm{I[i]_{distance}}+(\ I[i+1].m-I[i-1].m)$	
$\mathrm{I[i]_{distance}}=a/(f_m^{max}-f_m^{min})$	calculer les distances

Pseudocode 2 - Attribution des distances de crowding

Le terme "$[i].m$" signifie la m-ième image du i-ème individu par la fonction objectif et f_m^{max} ,f_m^{min}, les valeurs objectifs maximales et minimales de l'ensemble, la complexité de cette procédure peut atteindre (au cas de toutes les solutions contenues dans un seul front) $O(mN\ logN)$.

L'opérateur de comparaison de crowding ($<_n$) comme nous l'avons déjà spécifié contribue au processus de sélection afin d'assurer une uniformité de dispersion des solutions sur le front:

$$i <_n j \ si\ (i_{\text{rank}} < j_{\text{rank}})\ ou\ ((i_{\text{rank}} = j_{\text{rank}})\ et\ (j_{\text{distance}} > j_{\text{distance}}))$$

Pseudocode 3 - Opérateur de comparaison de crowding

Avec i_{rank}le classement de l'individu i et $idistance$ la distance de crowding de ce dernier, si i est mieux classé que j selon la non-dominance, i est choisi, si deux individus sont du même rang, l'individu avec la distance de crowding la plus grande est sélectionné.

A l'aide des opérations décrites antérieurement, le NSGA II pourra donc effecteur une recherche rapide, élitiste et moins couteuse.

3. CHAPITRE 03 : GESTION DE PORTEFEUILLE

ET APPLICATION SOUS R

3.1. Eléments fondamentaux de la gestion de portefeuille.

La gestion de portefeuille, d'abord centrée sur les actions, s'appuie sur des concepts et méthodes issus de la théorie moderne de portefeuille initiée par Markowitz (1952).

3.1.1. Rentabilité et risque

Les concepts clefs sont définis par souci de simplification, sur la base d'un placement en actions. La gestion de portefeuille s'appuie sur deux notions essentielles : la rentabilité et le risque.

3.1.1.1. Le taux de rentabilité d'un actif

Deux méthodes de calcul du taux de rentabilité, sur la base des données historiques existent :

- **Rentabilité classique** : La rentabilité du titre i à l'instant t est égale à :

$$R_{it} = \frac{S_t - S_{t-1} + D_t}{S_{t-1}}$$

Avec :

S_t le cours de l'action en t etD_t le dividende versé en t.

Les cours doivent être tous observés au même moment (les cours de clôture sont le plus souvent retenus) et ajustés des opérations sur titres. Les dividendes sont ici supposés réinvestis et sont donc intégrés au calcul. Dans le cas contraire, leur apport est évalué à part. On détermine alors le rendement en dividende qui est égal à :

$$\frac{D_t}{S_{t-1}}$$

La rentabilité cumulée sur la période n est :

$$R_n = \prod_{t=1}^{n}(1 + R_t) - 1$$

- **Rentabilité logarithmique (ln) :** La rentabilité du titre i à l'instant t est ici égale à :

$$R_{it} = Ln\left(\frac{S_t + D_t}{S_{t-1}}\right)$$

La rentabilité cumulée est égale à : $R_n = \sum_{i=}^{n} R_i$

3.1.1.2. Définition du risque

Voici quelques synonymes et définitions du terme « risque » :
Synonymie : aléa, chance, danger, fortune, fortune de mer, gageure, hasard, inconvénient, menace, péril, responsabilité.[40].

- **Définition 1**

Danger éventuel, plus ou moins prévisible, inhérent à une situation ou à une activité. Risque objectif, subjectif ; comporter des risques ; explorer le risque ; tenir compte des risques ; les risques du métier. Un rebord (...) m'offrit une banquette, d'où je pus à mon aise et sans risque jouir d'un spectacle vraiment neuf.

- **Définition 2**

Éventualité d'un événement futur, incertain ou d'un terme indéterminé, ne dépendant pas exclusivement de la volonté des parties et pouvant causer la perte d'un objet ou tout autre dommage.

- **Définition 3**

Le risque est la réalisation d'un événement futur incertain qui peut impliquer, s'il se réalise, des conséquences négatives pour un acteur ou pour une situation économique donnée.
Dans le domaine financier, le risque se présente sous de multiples formes, les plus célèbres d'entre eux étant le risque de taux d'intérêt, le risque d'évolution des cours (change, actions) et le risque de contrepartie. [Trader Lexique Finance. L :P] mesure du risque d'un actif.

3.1.1.3. Mesure du risque d'un actif

Le risque d'un actif peut être assimilé à la dispersion (ou variabilité) de sa rentabilité autour de sa moyenne. La mesure la plus utilisée est la variance (ou sa racine carrée : l'écart-type). La variabilité d'une action sur une période déterminée est donnée par l'écart-type de la série des taux de rentabilité de cette action sur un certain nombre n de sous-périodes (de jours par exemple). La variance de la rentabilité d'une action sur un an est calculée sur la base des 250 taux de rentabilité quotidiens (nombre approximatif de jours d'ouverture du marché). Le calcul pourra être effectué sur une période plus courte (par exemple 90 jours), de manière à « coller » davantage à l'évolution du marché. La variance d'une série de taux de rentabilité passés est

égale à la somme des carrés des écarts entre ces taux de rentabilité et le taux de rentabilité moyen, divisée par le nombre d'observations moins une.

$$\sigma_i^2 = \frac{1}{n-1}\sum_{t=1}^{n}(R_{it} - \bar{R}_\iota)$$

L'écart-type de la rentabilité du titre permet une comparaison directe avec la moyenne. Il est égal à la racine carrée de la variance.

3.1.1.4. Espérance de rentabilité d'un actif

Les cours futurs des actifs (et les taux de rentabilité) sont des variables aléatoires, dont on suppose connues les distributions de probabilité. La rentabilité anticipée ou « espérance de rentabilité » est mesurée par le concept statistique d'espérance mathématique. Le risque peut être appréhendé par la variance des taux de rentabilité. Celle-ci correspond à la dispersion des taux de rentabilités futurs autour de l'espérance de rentabilité. Cependant, cette mesure du risque est uniquement justifiée dans le cas où les rentabilités sont distribuées selon une loi symétrique, ce qui est le cas de la loi normale supposée suivie par les rentabilités logarithmiques des titres (hypothèse généralement faite en gestion de portefeuille). Dans le cas contraire, la variance constitue une mesure très critiquable du risque. Dans la pratique, les calculs de rentabilité et de risque sont le plus souvent effectués sur la base des données historiques. Ceci revient à considérer que l'évolution passée de la rentabilité permet de prévoir la rentabilité future.

3.1.1.5. Espérance de rentabilité et risque d'un portefeuille

Un portefeuille est caractérisé par les actifs qui le composent et par leurs poids relatifs dans celui-ci. Comme pour un actif, deux caractéristiques majeures sont analysées : la rentabilité et le risque. La rentabilité du portefeuille est égale à la moyenne pondérée de celle des actifs qui le composent. L'espérance de rentabilité d'un portefeuille composé de N titres est égale à :

$$E(R_p) = \sum_{i=1}^{n} w_i \, E(R_i)$$

avec w_i : poids de l'actif i dans le portefeuille Le risque du portefeuille est mesuré par la

variance de la rentabilité du portefeuille (ou son écarttype). La variance du portefeuille est calculée sur la base des actifs pris 2 à 2 (actifs i et j) :

$$VAR(p) = \sigma_P^2 = \sum_{i=1}^{n} \sum_{j=1}^{n} w_i \, w_j \sigma_{ij}$$

avec σ_{ij} la variance si i=j, ou la covariance si i≠j.

La variance du portefeuille fait intervenir la covariance entre les titres. Cela signifie que l'évolution des titres les uns par rapport aux autres a un impact sur le risque du portefeuille. En d'autres termes, la corrélation entre les actifs joue un rôle important. Le coefficient de corrélation entre deux actifs est égal au rapport entre la covariance et le produit des écarts type des deux titres.

$$\rho_{ij} = \frac{\sigma_{ij}}{\sigma_i \, \sigma_j}$$

Ce coefficient varie entre – 1 et +1. Si le coefficient est positif (négatif), cela signifie que les rentabilités des deux actifs sont corrélées positivement (négativement) et que leurs risques sont de même sens (sens opposé). Un coefficient de +1 (–1) signifie que les actifs sont parfaitement corrélés positivement (négativement). Si le coefficient est nul, cela signifie que les actifs sont indépendants.

3.1.2. Diversification et optimisation du portefeuille

La théorie du portefeuille s'appuie sur le critère de décision « espérance mathématique – variance ». Une somme risquée est supposée entièrement décrite par deux caractères de sa distribution de probabilité : son espérance mathématique et sa variance (ou son écart-type). Ce critère constitue une traduction probabiliste d'une attitude fréquente en finance : la recherche du meilleur compromis entre la rentabilité et le risque. Ainsi, pour des placements risqués ayant la même espérance, on choisira celui qui a la variance la plus faible. Le principe de diversification est à la base de la gestion de portefeuille. La diversification repose sur le fait que le risque d'un placement est atténué par la combinaison de différents actifs financiers. La

constitution du portefeuille, et son évolution, visent à réduire le risque global du placement, mesuré par la variance de sa rentabilité, en y combinant des actifs de risque différent. Pour un placement en actions, la diversification peut être sectorielle (secteurs d'activité dans un pays donné), internationale ou les deux à la fois. Plus généralement, la diversification d'un portefeuille s'appuie sur la combinaison de différents types de titres (actions, titres de créances, parts d'OPCVM…). La diversification ne permet cependant de réduire (ou dans le meilleur des cas d'annuler) qu'une partie du risque, le risque spécifique. La limite de diversification correspond ainsi au risque systématique. Un portefeuille parfaitement diversifié sur un marché national conserve un risque positif qui est égal à celui du marché. La diversification internationale des portefeuilles va permettre d'abaisser sensiblement le niveau de risque. La diversification d'un portefeuille s'appuie ainsi sur divers types de facteurs : classes d'actifs, zones géographiques, pays, secteurs d'activité, entreprises… Elle est de plus en plus souvent obtenue grâce à l'utilisation de fonds de fonds (multigestion). Ces derniers gèrent des portefeuilles constitués de fonds partiellement diversifiés (par exemple de fonds sectoriels). Sur la base du principe de diversification, la théorie propose des méthodes d'optimisation de portefeuilles. Celles-ci permettent de déterminer la meilleure composition possible du portefeuille, à un instant t, tel que le risque soit minimum pour un niveau de rentabilité donné (ou réciproquement, tel que la rentabilité soit maximale pour un niveau de risque fixé). Ces méthodes sont utilisées comme aide à la décision pour sélectionner les portefeuilles « efficients » parmi lesquels l'investisseur pourra choisir en fonction de son attitude vis-à-vis du risque. Certains fonds de gestion quantitative s'appuient de manière systématique sur une telle approche.

3.1.3. De la théorie du portefeuille à la gestion de portefeuille

La théorie du portefeuille constitue le cadre conceptuel de la gestion de portefeuille. Sa connaissance est essentielle pour la bonne compréhension des outils et des méthodes utilisés. Cependant, il convient de bien garder à l'esprit qu'elle repose sur deux hypothèses importantes. D'une part, les investisseurs sont supposés totalement rationnels et averses au risque. D'autre part, les marchés sont supposés efficients. D'un point de vue informationnel, cela signifie que les cours des actifs intègrent instantanément toute l'information disponible. En d'autres termes, cela signifie qu'il n'y a pas d'opportunité d'arbitrage sur le marché.

3.1.3.1. Les principes majeurs de la gestion de portefeuille

Compte tenu d'une situation quelque peu différente dans la réalité, la gestion d'un portefeuille d'un client, par un professionnel ou le conseil en gestion de portefeuille, nous semble reposer sur trois principes majeurs :

3.1.3.2. Connaissance du client et définition de ses objectifs d'investissement

- Analyse de sa situation patrimoniale, financière, fiscale, familiale, professionnelle ; – « étude » de sa personnalité, et détection d'éventuels éléments de comportements psychologiques (peur de perdre, effet de mémoire, mimétisme, versatilité…) ;
- Étude de son profil de risque ;
- Évaluation de ses connaissances et expériences des produits financiers ;
- Définition précise des ses objectifs d'investissement (durée, risque accepté…). Depuis le premier novembre 2007, la réglementation (AMF) oblige l'intermédiaire à faire remplir à son client un questionnaire précis reprenant en partie les éléments ci-dessus1.

3.1.3.3. Analyse permanente de l'évolution des marchés et bonne connaissance des produits financiers

- suivi permanent de la situation économique, de l'évolution et des perspectives des différents marchés (financiers, des changes, des matières premières, des biens et services, de l'immobilier…), tant au plan national qu'international ;
- Bonne connaissance des produits financiers proposés et de leurs mécanismes. Ne jamais proposer au client un produit dont on ne maîtrise pas les principes fondamentaux, notamment en matière de risque, nous semble constituer un principe éthique essentiel ;
- Prise en compte de l'existence d'inefficiences de marché et de la psychologie comportementale des investisseurs.

3.1.3.4. Diversification du portefeuille fondée sur l'arbitrage entre le risque et la rentabilité

- Définition de la stratégie d'investissement compte tenu de l'ensemble des points précédents (client, marchés et produits) ;
- Diversification : le maître mot. Envisager les différents facteurs de diversification : classe d'actifs, pays, secteurs, types d'OPCVM, types de gestion… ;
- choix des différents supports d'investissement et de leurs pondérations. Ce choix est fondé sur un arbitrage entre la rentabilité et le risque du portefeuille, compte tenu des objectifs de

l'investisseur et de sa situation (patrimoniale, financière, fiscale…). La rentabilité attendue par l'investisseur, en d'autres termes, sa prime de risque, est directement liée à ses propres caractéristiques et notamment à son attitude vis -à-vis du risque.

3.2. La théorie d'Harry Markowitz

Markowitz (économiste américain, prix Nobel d'économie en 1990) a eu l'idée de mesurer la rentabilité d'un portefeuille par l'espérance de rendement et le risque par sa variance. Il postule dans sa "théorie de portefeuille" que lorsque l'on cherche à répartir son épargne entre différents placements ou types de placements, deux, et deux seulement, critères de choix des supports d'investissements qu'il faut revoir, sont la rentabilité espérée du placement, et le risque associé, mesuré par la variance (ou écart-type) de cette rentabilité. A partir de ces deux critères, ce qui est appelé le modèle de Markowitz fournit la répartition dite optimale des actifs donc "le portefeuille". Les deux étapes du raisonnement sont :

a. les préférences des individus admettent une représentation dans le plan espérance-variance.

b. le modèle de Markowitz permet de construire l'ensemble des portefeuilles optimaux selon ces préférences.

Le choix de ces deux critères n'a rien d'évident comme le reconnaît Markowitz lui-même et repose sur des hypothèses assez restrictives. Malgré son fondement assez fragile, l'analyse de Markowitz a connu un grand succès car elle est intuitivement compréhensible, techniquement réalisable et donne lieu à une profusion d'application en finance de marché et en théorie financière.

3.2.1. But du modèle de Markowitz

Le modèle de Markowitz vise à l'aide d'une méthode mathématique à construire un portefeuille assurant :

a. soit le meilleur rendement à risque donné.

b. soit le plus petit risque à rendement donné.

3.2.2. Notation sur modèle de Markowitz

Considérons un investisseur désirant répartir une somme initial W(0) entre un ensemble d'actifs financiers. Nous supposons ses préférences représentées par l'espérance d'une fonction d'utilité

u croissante, strictement concave et différentiable sur R. Chaque actif du portefeuille peut être acquis à la date t = 0 en quantité illimitée au prix unitaire Vi(0). Chaque actif est caractérisé par deux variables aléatoires Vi(t) et Ri(t) définies sur l'ensemble Ω(t) des mondes possibles la date t telles que : 48 Modèles d'optimisation de portefeuille – Vi(t) est la valeur du i ime actif à l'instant t. – Ri(t) est la rentabilité du i ime actif à l'instant t avec : Vi(t) = (1+Ri(t)Vi(0)) et donc : Ri(t) = −1+ Vi(t) Vi(0) Principe du modèle de Markowitz Entre deux portefeuilles par leurs rendements (supposés aléatoires), on retient : – à risque identique celui qui a l'espérance de rendement la plus élevée. – à espérance de rendement identique, celui qui présente le risque le plus faible. Ce principe conduit à éliminer un certain nombre de portefeuilles, moins efficients que d'autres. La courbe qui relie l'ensemble des portefeuilles efficients s'appelle la frontière efficiente. En dessous de cette courbe, tous les portefeuilles rejetés sont dits dominés.

3.2.3. Principe du modèle de Markowitz

Entre deux portefeuilles par leurs rendements (supposés aléatoires), on retient :

- À risque identique celui qui a l'espérance de rendement la plus élevée.
- À espérance de rendement identique, celui qui présente le risque le plus faible.

Ce principe conduit à éliminer un certain nombre de portefeuilles, moins efficients que d'autres. La courbe qui relie l'ensemble des portefeuilles efficients s'appelle la frontière efficiente. En dessous de cette courbe, tous les portefeuilles rejetés sont dits dominés.

3.2.4. Frontière efficiente

La frontière qui caractérise le polygone ou la courbe des contraintes s'appelle dans cette situation la frontière efficiente de Markowitz et dans le polygone ou la courbe se situent tous les portefeuilles à rejeter dits portefeuilles dominés. Une autre manière de formuler ceci consiste à dire que les combinaisons (rendement, risque) de cette frontière forment un ensemble d'optima, c'est à dire que si l'un des éléments augmente, l'autre doit augmenter aussi.

3.2.5. Estimation des rendements et des risques

Soit Rp le rendement total d'un portefeuille composé de n actifs financiers caractérisés par leurs rendements R_i et leurs variables de decision associées x_i tel que :

$$\begin{cases} 0 \leq x_i \leq 1 \\ \quad \text{Et} \\ \sum_{i=1} x_i = 1 \end{cases}$$

Alors

$$R_p = \sum_{i=1}^{n} x_i R_i$$

avec

$$0 \leq x_i \leq 1$$

Dès lors :

$$E(R_P) = E(\sum_{i=1}^{n} x_i R_i) = \sum_{i=1}^{n} x_i E(R_i)$$

$$\text{Var}(R_P) = \sum_{i=1}^{n} \sum_{j=1}^{n} x_i x_j cov(R_i R_j)$$

Sélectionner un portefeuille d'actifs efficient, revient à choisir celui qui maximise le rendement moyen et minimise la variance et ce durant la période d'investissement à venir, ceci nécessite des estimations à priori des rendements et des risques de chacun des actifs constituant le portefeuille.

3.3. Description du problème

Un portefeuille financier peut être décrit comme étant un sous ensemble de N actifs financiers proposés et modélisés par le vecteur x = (x_1….. x_n), où les variables x_i indiquent la proportion Budgétaire attribuer à chaque actif tel que : $0 \leq x_i \leq 1$ et $\sum_i x_i = 1$

Notre algorithme génétique vise à déterminer un portefeuille d'actifs financiers efficient.
En d'autres termes, c'est de trouver un portefeuille qui, en parallèle, maximise le rendement (modélisé par la moyenne des rendements d'actifs) et minimise le risque (modélisé par la variance) tout en respectant la contrainte Budgétaire. Le problème ci-dessus peut être modélisé par le programme suivant :

Modéliser le rendement par la moyenne et le risque par la variance (var)

$$(P) = \begin{cases} Optimisez(z) = (E(R_p), Var(R_p) \\ \sum_i x_i = 1 \\ 0 \leq x_i \leq 1 \end{cases}$$

C'est-à-dire :

$$(P) = \begin{cases} Optimisez(z) = (\sum_{i=1}^{n} x_i\, E(R_i), \sum_{i=1}^{n} \sum_{j=1}^{n} x_i x_j Cov(R_i R_j) \\ \sum_{i=1} x_i = 1 \\ 0 \leq x_i \leq 1 \end{cases}$$

Où :

$E(R_p)$: le rendement espéré total du portefeuille.

R_i : rendement de l'actif i.

Cov(R_i; R_j) : le risque dû à la détention des actifs i , j .

3.4. Choix du logiciel

Le choix s'est porté sur l'emploi du langage du logiciel R, car il répond aux critères suivants :

- ✓ Les performances du langage : concernent la taille du code généré et l'exploitation efficace des ressources (logique ou physique).
- ✓ La maniabilité du langage : constitué d'un ensemble de possibilités faisant de telle sorte que le programmeur travaille avec aisance, assuré d'une part par la syntaxe du langage et d'autre part par un aspect visuel clair représentatif à la fois du détail et du global.
- ✓
- ✓ Le bagage du langage : il contient une interface graphique puissante ainsi qu'une grande variété de packages scientifiques implémentés.
- ✓ C'est logiciel libre distribué gratuitement sur le site du CRAN.

Le R est distribué gratuitement à partir du site du CRAN (Comprehensive R Archive Network): http ://www.r-project.org/.R est un système d'analyse statistique crée par Ross Ihaka et Robert Gentleman distribué librement sous les termes de la GNU General Public Licence ;

son développement et sa distribution sont assurés par plusieurs statisticiens rassemblés dans le R**.**

Development Core Team. R propose de nombreuses fonctions pour les analyses statistiques : Des plus classiques comme l'analyse en composante principale, la régression, la simulation au plus modernes comme les Support Vector Machines. R propose également un très grand nombre de fonctions graphiques fournissant un outil très puissant de visualisation et d'exploration des données [41].

R est un langage interprété et pas compilé c'est-à-dire que les commandes tapées au clavier sont directement exécutées sans qu'il soit nécessaire de construire un programme complet comme cela est souvent le cas pour la plupart des langages informatiques. Il s'ensuit que la syntaxede R est intuitive, les variables, les données, les fonctions, les résultats, etc. sont stockés dans la mémoire vive de l'ordinateur sous forme d'objets qui ont chacun leur nom. Notons que le nom d'un objet doit commencer par une lettre (majuscule ou minuscule) et peut comporter des lettres des chiffres, et des points. L'utilisateur agira alors sur les objets soit par le biais d'opérateurs soit par le biais défoncions

Le schéma ci-après présente l'interface de R sous Windows 7.

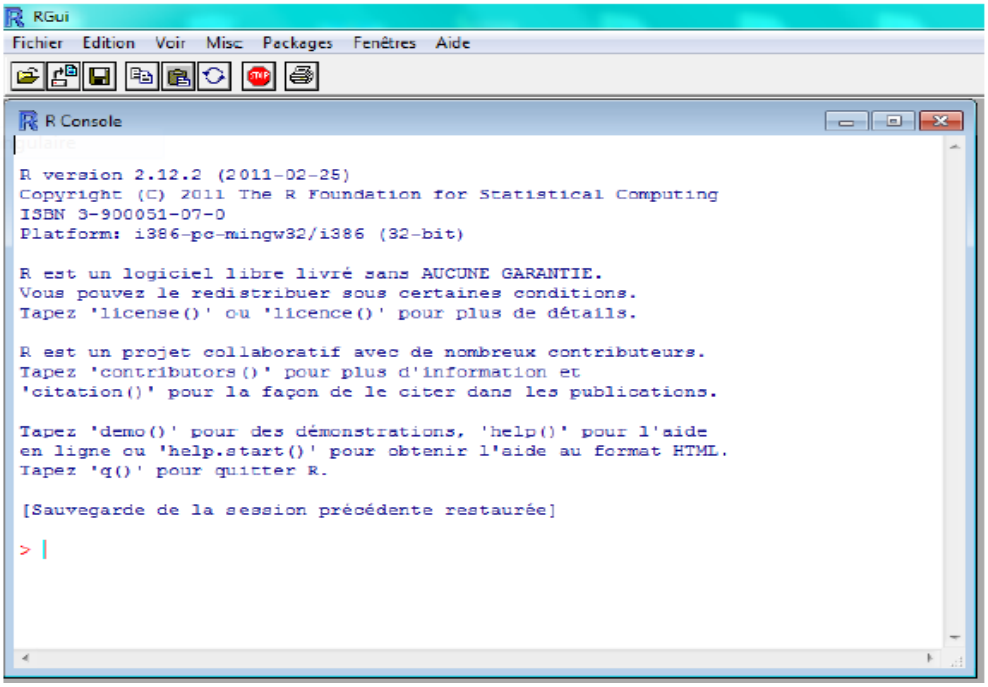

Figure 3.1 *- Interface du logiciel*

3.5. Les packages

Toutes les fonctions de R sont stockées dans une bibliothèque. Cette bibliothèque contient des packages de fonctions. Le package nommer base est le cœur de R et contient les fonctions de base du langage pour la lecture et la manipulation des données, la création de certains types de graphiques et certaines analyses statistiques.

Lors de l'installation de R, un grand nombre de packages sont installés par défaut. On peut accéder à la liste des packages pré-installés grâce à la commande installed.packages().

	Package	Libpath	Version	Priority	Bundle
som	"som"	"/Users/insermu494/Library/R/library"	"0.3-4"	NA	NA
svmpath	"svmpath"	"/Users/insermu494/Library/R/library"	"0.9"	NA	NA
KernSmooth	"KernSmooth"	"/Library/Frameworks/R.framework/Resources/library"	"2.22-14"	"recommended"	NA
MASS	"MASS"	"/Library/Frameworks/R.framework/Resources/library"	"7.2-8"	"recommended"	"VR"
Base	"base"	"/Library/Frameworks/R.framework/Resources/library"	"2.0.0"	"base"	NA
Boot	"boot"	"/Library/Frameworks/R.framework/Resources/library"	"1.2-19"	"recommended"	NA
Class	"class"	"/Library/Frameworks/R.framework/Resources/library"	"7.2-8"	"recommended"	"VR"
cluster	"cluster"	"/Library/Frameworks/R.framework/Resources/library"	"1.9-6"	"recommended"	NA
datasets	"datasets"	"/Library/Frameworks/R.framework/Resources/library"	"2.0.0"	"base"	NA
foreign	"foreign"	"/Library/Frameworks/R.framework/Resources/library"	"0.7"	"recommended"	NA
grDevices	"grDevices"	"/Library/Frameworks/R.framework/Resources/library"	"2.0.0"	"base"	NA
graphics	"graphics"	"/Library/Frameworks/R.framework/Resources/library"	"2.0.0"	"base"	NA
grid	"grid"	"/Library/Frameworks/R.framework/Resources/library"	"2.0.0"	"base"	NA
lattice	"lattice"	"/Library/Frameworks/R.framework/Resources/library"	"0.10-11"	"recommended"	NA
methods	"methods"	"/Library/Frameworks/R.framework/Resources/library"	"2.0.0"	"base"	NA
mgcv	"mgcv"	"/Library/Frameworks/R.framework/Resources/library"	"1.1-5"	"recommended"	NA
nlme	"nlme"	"/Library/Frameworks/R.framework/Resources/library"	"3.1-52"	"recommended"	NA
nnet	"nnet"	"/Library/Frameworks/R.framework/Resources/library"	"7.2-8"	"recommended"	"VR"
rpart	"rpart"	"/Library/Frameworks/R.framework/Resources/library"	"3.1-20"	"recommended"	NA
spatial	"spatial"	"/Library/Frameworks/R.framework/Resources/library"	"7.2-8"	"recommended"	"VR"
splines	"splines"	"/Library/Frameworks/R.framework/Resources/library"	"2.0.0"	"base"	NA
stats	"stats"	"/Library/Frameworks/R.framework/Resources/library"	"2.0.0"	"base"	NA
stats4	"stats4"	"/Library/Frameworks/R.framework/Resources/library"	"2.0.0"	"base"	NA
survival	"survival"	"/Library/Frameworks/R.framework/Resources/library"	"2.13-2"	"recommended"	NA
tcltk	"tcltk"	"/Library/Frameworks/R.framework/Resources/library"	"2.0.0"	"base"	NA
tools	"tools"	"/Library/Frameworks/R.framework/Resources/library"	"2.0.0"	"base"	NA
utils	"utils"	"/Library/Frameworks/R.framework/Resources/library"	"2.0.0"	"base"	NA

Figure 3.2 - *Listes des packages installés*

Les packages installés par défaut sur l'ordinateur ne sont pas forcément utilisable directement. Une étape préliminaire de chargement est parfois obligatoire. Pour connaître les packages chargés par défaut, on regarde la colonne Priority du résultat de la commande installed Packages (). Les packages associés à une priority base sont directement utilisable. D'autres packages (priority recommended ou priority NA) doivent être chargé en mémoire par l'intermédiaire de la fonction library(). Par exemple, on souhaite utiliser les fonctions du package MASS.

La priority étant recommended, on doit charger le package via la commande library(MASS).

Notons que tous les packages téléchargés du CRAN sont library-dépendant ; leur priority étant NA. Par exemple, on souhaite utiliser la fonction som() pour construire des cartes de Kohonen. La fonction som() est dans le package som ; il est indispensable de chargées le packages som.

3.6. Programmation avec R

Il y a deux façons pour écrire des fonctions R :

- ✓ Soit directement dans la fenêtre de commandes ;
- ✓ Soit en utilisant l'éditeur de développement de R, en sauvegardant les programmes dans des fichiers texte avec l'extension "R".

3.7. Implémentation et résultats

Les données choisies pour faire l'application de notre algorithme génétique concerne le rendement et la matrice de variance covariance des actifs de trois marchés financiers : HONG KONG HENG SENG (31 actifs) , SP 100 AMERICAIN (98 actifs) et SP 500 AMERICAIN (287 actifs).

Pour les besoins de la programmation de l'algorithme NSGAII, on a programmé à l'aide des fonctions de base les sous fonctions suivantes :

1. **Téléchargement :** télécharger les donner enregistré dans un fichier texte **(.txt)** à l'aide de la fonction **(read.csv).**
2. **Trier les données :** trier la solution initiale à base de dominance et sélectionnée une population courante.
3. **Opérateur génétique :** application des opérateurs génétique (croisement et mutation) afin de générer une nouvelle population Q_t .
4. **Détermination de la population R_t tel que** : $R_t = P_t \cup Q_t$.
5. **Détermination des front F_i:** détermination à base de dominance.
6. **Sélectionner une population P_{t+1} :**sélectionner P_{t+1} à l'aide de la distance de crowding
7. **Générer une population Q_{t+1} :**appliquer les opérateurs génétiques (OG) sur P_{t+1}

On obtient les résultats suivants

Le model est :

$$(P) = \begin{cases} Optimisez(z) = (\sum_{i=1}^{n} x_i\, E(R_i), \sum_{i=1}^{n}\sum_{j=1}^{n} x_i x_j Cov(R_i R_j) \\ \sum_{i=1} x_i = 1 \\ 0 \le x_i \le 1 \end{cases}$$

Risque	Rend	Risque	Rend
0,00559693	0,3886435	0,00704152	0,4293855
0,00559841	0,3893894	0,00709187	0,4308148
0,00570746	0,3926668	0,00731855	0,4538422
0,00571936	0,3935371	0,00739253	0,4582485
0,00577077	0,3936715	0,00755714	0,4610552
0,00581842	0,3983259	0,00755946	0,463376
0,00613804	0,4002324	0,00768957	0,4714837
0,00633842	0,4027352	0,00788904	0,4796296
0,00638302	0,4027416	0,00804033	0,4990173
0,00641364	0,4059051	0,0081762	0,5073476
0,00644804	0,4065096	0,00828547	0,5316683
0,00664353	0,4116241	0,00844696	0,5513021
0,00669059	0,4130195	0,00849739	0,5538627
0,00682274	0,4131628	0,00855728	0,5661534
0,00685435	0,4143346	0,00923681	0,6735208
0,0068854	0,4185786	0,00944488	0,7242672
0,006975	0,423259	0,00990821	0,8061123

Tableau 1 : *rendement et risque des solutions trouvées pour HANG_SENG*

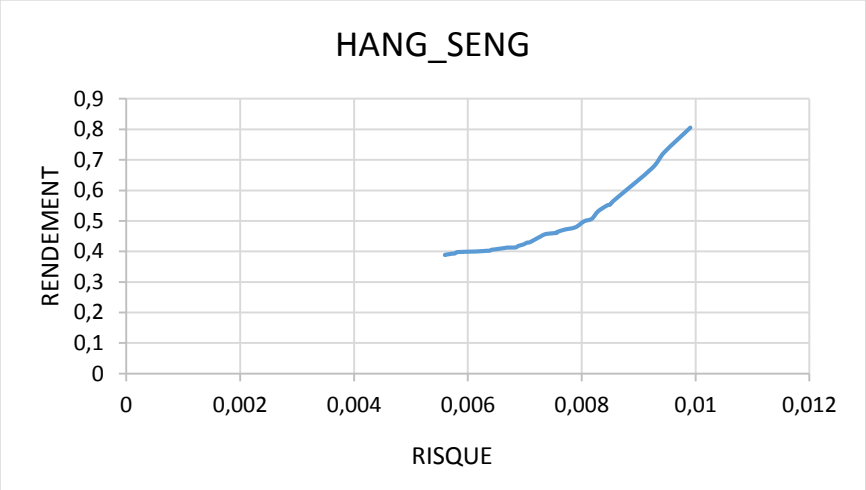

Figure 3.3 - *Frontière efficiente de HANG_SENG.*

Les différents portefeuilles efficients possibles, résultants de modèle HANG_SENG est représenté dans la figure ci dessus, on remarque le plus grand rendement trouvé pour HANG SENG est égale à 0,8061123 pour niveau de risque égale à 0,00990821. Tandi que le plus petit

risque trouvé est égale à 0,00559693 pour un rendement de 0,3886435.

Risque	Rend	Risque	Rend	Risque	Rend
0,1289194	0,00254741	0,1822614	0,00460404	0,2238851	0,00541655
0,1337365	0,00274037	0,1838024	0,00460646	0,225073	0,00558885
0,1345441	0,00276501	0,1841577	0,00462166	0,2304804	0,00559057
0,1369266	0,00297769	0,1887286	0,00476769	0,233114	0,00562261
0,1382192	0,00335064	0,1915952	0,00481267	0,2380853	0,00581156
0,1455042	0,00344548	0,1959754	0,00483222	0,2508707	0,00583995
0,1469577	0,00368767	0,1960831	0,00489077	0,2545641	0,00588308
0,1494809	0,00369087	0,1990961	0,00491291	0,2620893	0,00588531
0,150232	0,00377582	0,2014973	0,00508497	0,2622082	0,00589865
0,1557627	0,00378347	0,2023172	0,00512151	0,2836918	0,00590652
0,1600399	0,00394363	0,2068989	0,00512779	0,2874966	0,00593638
0,1627401	0,00432866	0,2111621	0,00514651	0,3016741	0,0061479
0,1754552	0,00433263	0,2121324	0,00516556	0,3184354	0,0063539
0,1779209	0,00435912	0,2121599	0,00517242	0,3189628	0,00646095
0,1780866	0,00444217	0,2125376	0,0052847	0,4069034	0,00665138
0,1793946	0,00444785	0,220279	0,00539252	0,4680157	0,00688989
0,1817786	0,00459364	0,2216644	0,00541137	0,4813338	0,00691839

***Tableau 2** : rendement et risque des solutions trouvées pour SP 100.*

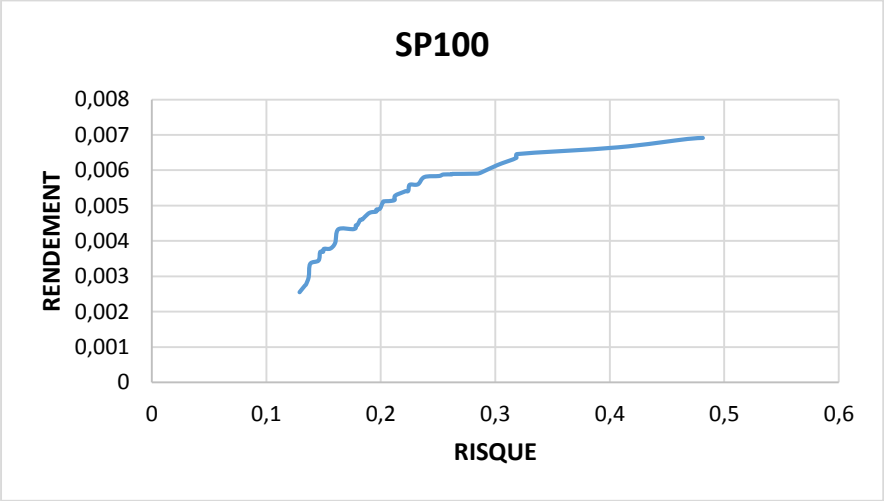

***Figure 3.4 -** Frontière efficiente de SP100.*

Les différents portefeuilles efficients possibles, résultants de modèle SP100 est représenté dans la figure ci dessus, on remarque le plus grand rendement trouvé pour HANG SENG est égale à 0,8061123 pour niveau de risque égale à 0,00691839 avec un risque de 0,4813338 et le rendement minimum est de 0,00254741 avec un risque de 0,1289194..

Risque	Rend	Risque	Rend	Risque	Rend
0,0001359	0,00368966	0,0002121	0,00580332	0,00036639	0,00833261
0,00014142	0,0037607	0,0002122	0,00590489	0,00038061	0,00837468
0,00014194	0,00425278	0,00021833	0,00599219	0,00038861	0,00841632
0,00014815	0,00436966	0,00022418	0,00602417	0,00040316	0,00845653
0,00015334	0,00449731	0,00023469	0,0060433	0,00045633	0,0084897
0,00015741	0,00452256	0,00023701	0,00628699	0,00047758	0,00910519
0,00015977	0,0045249	0,00023795	0,00641291	0,00055607	0,00919438
0,00016213	0,00458839	0,00025811	0,00645067	0,00063401	0,00945098
0,00016216	0,00478609	0,00025962	0,00661187	0,00106401	0,01101382
0,00016299	0,00485024	0,00026352	0,00682127	0,00136904	0,01187763
0,00016883	0,00502842	0,00027047	0,00687285	0,00226964	0,01459262
0,0001692	0,00504598	0,00028931	0,00709923	0,00258265	0,01542128
0,00016955	0,00506894	0,00030516	0,00724291	0,00262349	0,01555473
0,00017825	0,00516343	0,00032333	0,00758055	0,00260346	0,01555966
0,00018081	0,00517042	0,00032462	0,00770987	0,00266178	0,01560993
0,00018242	0,00543687	0,00032819	0,00777433	0,00265784	0,0157332
0,00018701	0,00545101	0,00032899	0,00783058	0,00307522	0,0166241
0,00018923	0,005565	0,00034341	0,00791958	0,00331585	0,01719131
0,000194	0,0057159	0,0003513	0,00802909	0,00339175	0,01731001
0,00020559	0,00571912	0,00036157	0,00823923	0,00354428	0,01764575

Tableau 3 : *rendement et risque des solutions trouvées pour SP 500.*

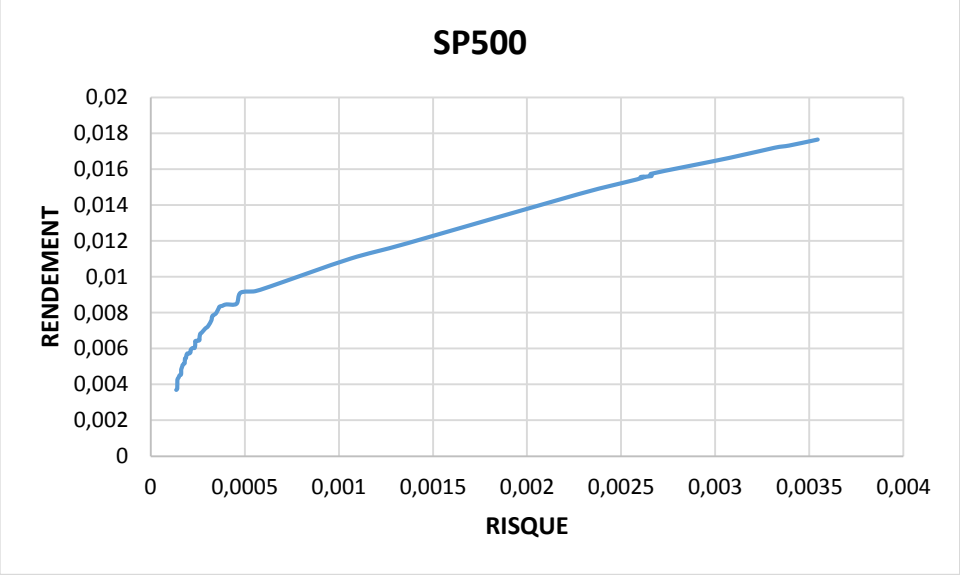

Figure 3.5 - *Frontière efficiente de SP500.*

Les différents portefeuilles efficients possibles, résultants de modèle SP500 est représenté dans la figure ci-dessus, on remarque le plus grand rendement trouvé pour HANG SENG est égale à

0,01764575 pour niveau de risque égale à 0,00354428. Tandi que le plus petit risque trouvé est égale à 0,00368966 pour un rendement de 0,0001359.

Conclusion

Notre objectif à travers cette étude est d'apprécier l'apport de l'optimisation multiobjectif à la résolution des problèmes de gestion de portefeuilles financiers, qui consistent à trouver un portefeuille d'actifs efficient. Afin de résoudre cette problématique, on a utilisé le modèle de Markowitz Mean variance qui consiste à estimer le rendement par le rendement espéré durant la période d'investissement, tandis que le risque est estimé par la variance des rendements. . Par la suite, on a enrichi ce travail en investiguant les capacités de l'optimisation par les algorithmes génétiques plus précisément le non dominated sorting genetic algorithm NSGA II, tout en présentant l'état de l'art de cet algorithme et son fonctionnement présenté dans la littérature qui constitue le cœur de notre travail.

Afin de trouver les résultats demandés, on a programmé et implémenté le non dominated sorting genetic algorithm NSGA II sous langage R tout en utilisant les fonctions de base du langage R. et d'autres fonctions que nous avons-nous même élaborées.

Bibliographie

[1] (Obayashi, 1998) :

Niching and elitist models for multi objective genetic algorithms. S.Obayashi, S.Takahashi, and Y.Takeguchi, (1998). In *Parallel Problem Solving from Nature PPSN'5*, pages 260‑269,Amsterdam. Springer-Verkag.

[2] (Fujita, 1998):

Multi-objective optimal design of automotive engine using genetic algorithm. K.Fujita, N.Hirokawa, S.Akagi, S.Kimatura, and H.Yokohata, (1998). In *Design Engineering Technical Conferences DETC'98*, pages 1–11, Atlanta, Georgia.

[3] (Shaw, 1996) :

Initial study of multi-objective genetic algorithms for scheduling the production of chilled ready meals. K.Shaw, P.Fleming. In *Mendel'96 2nd Int. Conf. on Genetic Algorithms*, Brno, Czech Republic.

[4] (Fujimura, 1996) :

Path planning with multiple objectives. K.Fujimura. *IEEE Robotics and Automation Society Magazine*, 3(1) :33–38

[5] (Todd, 1997) :

A multiple criteria genetic algorithm for container loading. D.Todd, P.Sen.In Back, T., editor, *Seventh Int. Conf. on Genetic Algorithms ICGA'97*, pages 674–681, San Mateo, California. Morgan Kaufmann.

[6] (Loughlin, 1997):

The neighborhood contraint method : A genetic algorithmbased multiobjective optimization technique. D.Loughlin, S.Ranjithan. In Back, T., editor, *Seventh Int. Conf. on Genetic*

[7] (Halhal, 1997):

Water network rehabilitation with a structured messy genetic algorithm. D.Halhal, G.Walters, D.Ouazar, and D.Savic. *Journal ofWater Resources Planning and Management*, 123(3) :137–146.

[8] (Veldhuizen, 1997) :

Finding improved wire-antenna geometries with genetic algorithms. D.V.Veldhuizen, B.Sandlin, R.Marmelstein, G.Lamont, and A.Terzuoli. In

Chawdhry, P., Roy, R., and Pant, P., editors, *Soft Computing in Engineering Design and Manufacturing*, pages 231–240,London. Springer Verlag.

[9] (Dahl,1995) :
A tabu search approach to the channel minimization problem. G.Dahl, K.Jornsten, and A.Lokketangen. In Liu, G., Phua, K.-H.,Ma, J., Xu, J., Gu, F., and He, C., editors, *Optimization Techniques and Applications, ICOTA'95*, volume 1, pages 369–377, Chengdu, China. World Scientific.

[10] (Weinberg, 2001) :
A co-evolutionary meta-heuristic for the assignment frequencies in cellular networks. B.Weinberg. In *EvoCop*.

[11] (Meunier, 2002) :
Algorithmes évolutionnaires parallèles pour l' optimisation multi-objectif de réseaux de télécommunications mobiles. Hervé Meunier 2002 Thèse de doctorat.

[12] (Barichard, 2003) :
Approches hybrides pour les problèmes multiobjectifs. V.Barichard, Thèse de Doctorat 2003, école doctorale d'Angers.

[13] (Ishibuchi, 1998) :
A milti-objective genetic local search algorithm and its application to the flowshop scheduling. H.Ishibuchi and T.Murata, IEEE transactions on systems, Man and Cybernetics - Part C : applications and reviews, 28 :392-403, 1998.

[14] (Schaffer, 1984) :
Multiple objective optimization with vector evaluated genetic algorithms. J.D. Schaffer. PhD thesis, Vanderbilt University, 1984.

[15] (Fourman, 1985) :
Compaction of symbolic layout using genetic algorithm. M.P.Fourman. in procedings of the first international conference on genetic algorithms (ICGA)

[16] (Fonceca , 1995) :
Multiobjective genetic algorithms with applications to control engineering problems. C.Fonseca 1995, PhD thesis, University of Sheffield.

[17] (Barichard, 2003) :
Approches hybrides pour les problèmes multiobjectifs. V.Barichard, Thèse de Doctorat 2003, école doctorale d'Angers.

[18] (Goldberg, 1987) :
Genetic algorithms with sharing for multimodal function optimization. In Genetic Algorithms and their Applications: D.E. Goldberg and J. Richardson. Proceedings of the Second International Conference on Genetic Algorithms, pages 41- 49. Lawrence Erlbaum, 1987.

[19] (Holland, 1975):
Adaptation in natural and artificial systems. J.H.Holland Michigan Press University, Ann Arbor, MI, USA 1975.

[20] (Veldhuizen, 2000) :
On measuring multi-objective evolutionary algorithm performance. D.A.V Veldhuizen and G.B.Lamont. In 2000 Congress of evolutionary computation. Piscataway. New jersey. Volume 1, page 204-211, July 2000.

[21] (Zitzler, 1999) :
Evolutionary algorithms for multiobjective optimization: methods and applications. E. Zitzler. PhD thesis, Swiss Federal Institute of Technology (Zurich), 1999.

[22] (Schott, 1995) :
Fault tolerant design using single and multicriteria genetic algorithm optimization. J.R.Schott.Master's thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, 1995.

[23] (Basseur, 2002) :
Design of Multi-objective Evolutionary Algorithm: Application to the Flow-shop Scheduling Problem. M. Basseur, F.Seynhaeve and EG.Talbi. In Congress of evolutionary computation. CEC'02, pages 1151-1156, Honolulu, Hawaii, USA, May 2002.

[25] (Friesz, 1993):
The multiobjective equilibrium network design problem revisited : A simulated annealing approach. T.Friesz, G.Anandalingam, N.Mehta, K.Nam, S.Shah, and R.Tobin. *EuropeanJournal of Operational Research*, 65 :44‑57.

[26] (Darwin, 1859):
On the origins of species by means of natural selection. *London: Murray*, 247.

[27] (Goldberg 1989):
Genetic Algorithms in Search, Optimization and Machine Learning (1st ed.). *Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.*

[28] (Goldberg 1990) :
Real-coded genetic algorithms, virtual alphabets, and blocking. *Urbana*, *51*, 61801

[29] (Baker, 1985):
Adaptive selection methods for genetic algorithms. In *Proceedings of an International Conference on Genetic Algorithms and their applications* (pp. 101-111).

[30] (Goldberg, Deb, 1991) :
A comparative analysis of selection schemes used in genetic algorithms. *Foundations of genetic algorithms*, *1*, 69-93.

[31] (Baker, 19807) :
Reducing bias and inefficiency in the selection algorithm. In *Proceedings of the second international conference on genetic algorithms* (pp. 14-21).

[32] (Man et al. 1996) :
Man, K. F., Tang, K. S., & Kwong, S. (1996). Genetic algorithms: concepts and applications [in engineering design]. *Industrial Electronics, IEEE Transactions on*, *43*(5), 519-534.

[33] (Srinivas et Deb, 1994) :
Srinivas, N., & Deb, K. (1994). Muiltiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary computation*, *2*(3), 221-248.

[34] (Deb et al. 2002) :
Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. A. M. T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *Evolutionary Computation, IEEE Transactions on*, *6*(2), 182-197.

[35] (Sareni et Krähenbühl, 1998) :

Sareni, B., & Krähenbühl, L. (1998). Fitness sharing and niching methods revisited. *Evolutionary Computation, IEEE Transactions on*, *2*(3), 97-106.

[36] (Deb et Agrawal , 1995) :
Deb, K., & Agrawal, R. B. (1994). Simulated binary crossover for continuous search space. *Complex Systems*, *9*(3), 1-15.

[37] (Deb et Goyal , 1996) :
Deb, K., & Goyal, M. (1996). A combined genetic adaptive search (GeneAS) for engineering design. *Computer Science and Informatics*, *26*, 30-45.

[38] (Black, P. E. (2007)) :
big-O notation. *Dictionary of Algorithms and Data Structures, 2007*.

[39] (Markowitz h.m, 1952) :
Markowitz.H : Portfolio selection, Journal of Finance 7 ,1952, 77-91.

[40] (lexilogos, dictionnaire français)

[41] Christophe Lalanne, Christophe Pallier : Introduction aux Statistiques et a l'utilisation du logiciel R.