N° d'ordre : 09/2013–M/MT

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITÉ DES SCIENCES ET TECHNOLOGIE DE HOUARI BOUMEDIENNE
FACULTÉ DES MATHÉMATIQUES

# MÉMOIRE
PRÉSENTÉ POUR L'OBTENTION DU DIPLÔME DE MAGISTER.

EN MATHÉMATIQUES
SPÉCIALITÉ : PROBABILITÉS & STATISTIQUES

**P**résenté Par : Massika IKHLEF

**S**ujet

# Optimisation Stochastique : cas du problème du portefeuille

Soutenu publiquement le 01/12/2013, devant le jury composé de :

| | | | |
|---|---|---|---|
| Mr. BOUKHETALA Kamal | Professeur, | à l'USTHB | Président. |
| Mr. AÏDER Méziane | Professeur, | à l'USTHB | Directeur de thèse. |
| Mme. GUESSOUM Zohra | Maitre de conférence A, | à l'USTHB | Examinatrice . |
| Mr. TATACHAK Abdelkader | Maitre de conférence A, | à l'USTHB | Examinateur . |

# Table des matières

# Table des figures

# Liste des tableaux

# Remerciements.

Premièrement, je remercie Dieu, pour m'avoir donné la volonté et la force pour accomplir ce modeste travail, elhamdou li llah.

Je tiens à exprimer ma profonde gratitude au professeur M. Aïder, mon directeur de thèse, et mon guide, pour la confiance qu'il m'a faite en acceptant de diriger mes recherches, et pour ses précieux conseils et orientations, ainsi que pour l'intérêt particulier qu'il a accordé à ce travail. Je ne le remercierai jamais assez pour la grande contribution et l'aide qu'il m'a apporté pour l'aboutissement de ce travail.

J'adresse mes remerciements au professeur K. Boukhetala, pour l'honneur qu'il m'a fait en acceptant de présider le jury de ce mémoire.

Je remercie Mr A. Tatachak et Mme Z. Guessoum pour avoir accepté d'examiner ce travail.

Je remercie également toute mes parents notamment ma chère mère, mes frères et ma sœur ainsi que toute la famille Chikh.

Je remercie mes amis et mes collègues et tous ceux qui ont contribué de près ou de loin à la réalisation de ce travail : Rafik et Madani.

# Dédicaces.

*Je dédie ce travail :*

*A mes très chers parents notamment ma chère mère.*
*A ma sœurs Naïma.*
*A mes frères : Saïd et Chafik.*
*A la mémoire de ma grande mère.*
*A toute la famille Chikh.*
*A tous mes amis.*

# Résumé


## Résumé

L'objectif de cette thèse est d'étudier les problèmes d'optimisation stochastique, notamment les problèmes de sélection de portefeuilles et d'élaborer une méthode de résolution issue de l'approche $Mean - Variance$ initié par Markowitz $(1959)$ et du critère *VaR* (Value at Risk) dans un deuxième volet, tout en investiguant les outils statistiques et les capacités de l'optimisation par les métaheuristiques, plus précisément la métaheuristique de Pareto Ant Colony Optimisation P-ACO.

Ces deux approches sont implémentées sous l'environnement R, afin de construire un portefeuille efficient sur la base de 15 actifs cotés à la bourse étrangère et comparer par la suite entre les deux approches pour choisir la plus pertinente.



**Mots clés** : problèmes d'optimisation stochastique, sélection de portefeuilles, portefeuille efficient, modèle de Markowitz, value at risk, bootstrap, pareto ant colony optimization, environnement R.



## Abstract

The objective of this thesis is the study of stochastic optimization problems, specially portfolio selection problems and the development of a resolution method issue from Mean-Variance approach, initiated by Markowitz $(1959)$, and from the value at risque criterion in a second handpick. For that, we investigate some statistic objects, we also introduce Pareto Ant Colony Optimization as an especially effective mata-heuristic.

Furthermore, we implement these two approaches under R environment, so we provide a numerical example based on the some 15 strange listed shares and build an efficient portfolio. In the end, we compare between those approaches and adapt the best one.



**Keywords :** stochastic optimization problems, portfolio selection, efficient portfolio, Markowitz model, value at risk, bootstrap, pareto ant colony optimization, R environment.

# 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 stochastique dont les paramètres ne sont pas tous connus avec certitude lorsque la décision doit être prise. Par exemple, le problème de gestion de portefeuilles financiers consiste à choisir dans quels actifs investir pour un certain nombre de dates fixées à l'avance, de façons à maximiser sa richesse tout en satisfaisant des contraintes physiques et stratégiques. Lorsqu'un rebalancement de portefeuille est opéré, le choix des actifs financiers doit se faire sans connaître les rendements exactes des différents actifs sur la période d'investissement à venir. La complexité vient du nombre de combinaisons de portefeuilles réalisables et il faut recourir à des outils mathématiques extrêmement performants dans le but de sélectionner une solution optimale en un temps raisonnable.

L'objet de cette thèse est de modéliser et analyser un problème d'optimisation stochastique et de proposer une méthode 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 de la programmation linéaire, par la suite, nous passerons à l'optimisation stochastique [7] en présentant différentes manières de prendre en compte l'incertitude des paramètres d'un problème d'optimisation stochastique. En effet, on traitera les différents modèles de recours et les modèles de L-Shaped qui consistent à créer deux problèmes complémentaires : le problème maïtre ( master) et les problèmes satellites ou sous problèmes ou encore problèmes esclaves.

Dans la plupart des problèmes du monde réel, il ne s'agit pas d'optimiser seulement un seul critère mais plutôt d'optimiser simultanément plusieurs critères [38] et qui sont généralement conflictuels. L'optimisation multi-objectif, qui fera l'objet du deuxième chapitre, consiste donc 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.

L'utilisation des métaheuristiques pour résoudre des problèmes multi-objectifs a fait l'objet

d'un intérêt de plus en plus croissant. A présent, les métaheuristiques constituent un des champs de recherche les plus actifs dans l'optimisation multi-objectifs. La plupart des travaux existants concernent l'optimisation bi-objectif. Le cas multi-objectif reste difficile à résoudre, non seulement à cause de la complexité de ces problèmes, mais aussi à cause du nombre élevé des solutions Pareto optimales à un problème d'optimisation multi-objectif.

Le chapitre suivant représentera le noyau de ce travail, car nous nous intéressons à l'étude des capacités de la métaheuristique d'optimisation par colonie de fourmis (Ant Colony Optimization-ACO) [14]. Les algorithmes à base de colonies de fourmis ont été initialement proposés dans [13]. Depuis son apparition, l'ACO requiert de plus en plus l'attention de la communauté scientifique vu le succès qu'elle a réalisé. Elle a été appliquée à plusieurs problèmes combinatoires comme le problème du voyageur de commerce [15], affectation quadratique introduite par Gambardella et al [23], routage de véhicules [4], sac à dos multidimensionnel [2].

Cette métaheuristique est basée sur une la stratégie phéromonale qui consiste à un mécanisme d'apprentissage utilisé par les algorithmes ACO pour permettre à la colonie de converger vers les solutions qui est basé sur le dépôt de traces de phéromone. Ces traces sont ensuite utilisées pour biaiser les décisions lors de la construction de solutions. Le choix d'une telle stratégie est à la fois déterminant et délicat en fonction des problèmes.

Initialement, la métaheuristique ACO a été introduite pour résoudre un problème de recherche de chemin hamiltonien dans un graphe, i.e. voyageur de commerce. Dans ce cas, la stratégie phéromonale consiste à déposer la phéromone sur les chemins construits. Pour les problèmes de sélection de sous-ensemble comme par exemple les problèmes de sac à dos, de clique maximale ou de satisfaction de contraintes, il n'est pas évident de choisir une stratégie phéromonale appropriée. En effet, ces problèmes ne peuvent pas être ramenés, généralement, à des problèmes de recherche de chemins, puisque l'ordre dans lequel les composants de solutions sont sélectionnés n'est pas important.

Pour finir, le dernier chapitre est consacré aux modèles spécifiques des problèmes de sélection de portefeuille : le modèle de Markowitz [27], qui a révolutionné l'optimisation de portefeuilles en proposant le modèle moyen-variance. Markowitz a donné le point de départ de la théorie moderne de la gestion de portefeuille. Selon ce modèle, tout investisseur poursuit deux objectifs conflictuels : la maximisation du rendement espéré et la minimisation du risque, mesuré par la variance des taux de rentabilité. Les principaux apports de ce modèle sont sans doute son aspect relativement général, permettant son utilisation dans un grand nombre de situations pratiques, et sa simplicité en termes d'analyse théorique. En effet, une fois déterminer le risque et le rendement d'un actif financier ou d'une catégorie d'actifs financiers, la théorie financière moderne permet d'identifier les allocations d'actifs les plus judicieuses en fonction du degré de risque que l'investisseur est prêt à assumer.

Mais les rendements attendus ne sont pas des constantes physiques existants à l'état naturel comme la vitesse de la lumière par exemple. Dans la plupart des études théoriques comme dans

la pratique, les estimations du taux de rentabilité espéré et du risque d'un portefeuille sont déduites par une méthode historique le bootstrap. Bien que l'étendue des informations quantitatives disponibles concernant les actifs financiers est sans égale en sciences sociales, il est fondamental de rappeler que ces estimations sont faites à partir d'un échantillon d'observations. Ces estimations sont considérées comme les vrais paramètres de la population entière, et non l'estimation de vrais paramètres inconnus.

L'optimisation moyenne-variance de Markowitz [27] est le standard en finance mais son application est néanmoins sujette à caution. En effet, une optimisation moyenne-variance ne sera valable que si les taux de rentabilité suivent une loi normale. De nombreuses études ont pourtant conclu au rejet de la loi normale comme loi de probabilité des taux de rentabilité [16], (Affleck-Graves et MacDonald (1989)) [1]. Alors on fait toujours l'hypothèse de taux de rentabilité normalement distribués mais le problème c'est que la variance ne paraît pas être une mesure du risque adéquate. Dans son calcul, l'écart entre les taux de rentabilité au dessus de la moyenne et la moyenne sont considérés comme une source potentielle de risque, ce qui est contraire à la vision du risque communément admise. Ainsi, des mesures du risque alternatives sont apparues, dans ce document on s'intéressera uniquement au modèle value at risque VaR.

VaR est définie par Jorion [19] comme étant l'espérance de perte maximum d'un investissement sur un horizon déterminé à un certain niveau de confiance. Elle correspond en fait à un quantile d'une distribution de perte.

A la fin de ce travail, on présente une application informatique implémentée sous l'environnement R, 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 ;

Cette application a pour objectif de sélectionner un portefeuille efficient, c'est à dire sélectionner un ensemble d'actifs financiers sur le marché étranger parmi plusieurs, qui maximise le rendement et minimise le risque, tout en appliquant les deux modèles Markowitz et Value-at-Risk. Les cours journaliers d'actifs sont réelles téléchargées sur le site Internet : $http://finance.yahoo.fr$.

Une fois les résultats des deux modèles sont obtenus, on compare entre eux afin de voir lequel est le plus pertinent.

# 1

# Optimisation stochastique

## Sommaire

# Introduction

Le but que l'on se fixe dans un problème d'optimisation est de prendre la meilleure décision vis à vis des situations qui comportent de l'incertitude sans attendre la réalisation de certaines variables aléatoires ; alors on parle de **l'optimisation stochastique**. Le regard vers l'incertitude reste subjectif et varie d'un décideur à l'autre. On peut par exemple souhaiter que dans l'espace de tout évènement possible, la solution adoptée soit telle que son coût soit minimal en moyenne. Ainsi, les approches de l'incertitude comprennent des techniques orientées vers la minimisation (maximisation) de l'espérance mathématique (ou d'autres moments), la minimisation de la déviation par rapport aux cibles (deviation from goals), la minimisation des coûts maximaux, la minimisation de l'écart entre le meilleur et le pire cas, ou même la simple satisfaction des contraintes avec une probabilité donnée d'avance qui représente le taux de probabilité du système.

## 1.1 Introduction à la programmation stochastique

L'incertitude dans les problèmes d'optimisation touche notamment les coûts de production, les prix des marchés, les pénalités en cas des violations des contrats, aussi bien que la demande de clients, les délais de livraison, les temps de traitement, la disponibilité des machines et d'autres coefficients technologiques.

Il se peut que l'on connaisse partiellement certains aspects du phénomène à travers soit des scénarios, soit un historique, soit une loi de probabilité, soit des moments de la variable aléatoire (par exemple espérance mathématique, variance). Des scénarios peuvent être crées à partir des historiques, (comme par exemple les ventes des derniers années, les températures de la dernière décennie, la mortalité d'une certaine population), ou bien à partir de l'opinion d'experts qui peuvent prévoir le comportement du phénomène incertain. Si la loi de probabilité de la variable aléatoire est connue de façon analytique, on peut soit créer des méthodes analytiques (la plupart des fois ceci est un travail non-trivial) qui tiennent compte de cette loi de probabilité, soit générer un ensemble de scénarios suivant cette loi.

Lorsque l'on parle de la programmation stochastique, on entend la programmation mathématique sous incertitude. La programmation stochastique n'est pas une famille de problèmes, de modèles et d'outils différents des autres domaines d'optimisation (programmation linéaire, non-linéaire, dynamique) ; au contraire, elle constitue un complément de ces familles lorsque la notion d'incertitude intervient. On se permet donc de parler de programmation stochastique linéaire, non-linéaire ou bien de programmation stochastique dynamique. Dans le présent travail on s'intéressera uniquement à la programmation stochastique linéaire mais dans ce but, il était nécessaire de faire un feed-back sur la programmation linéaire.

## 1.2 Programmation linéaire

### 1.2.1 définition

Un programme linéaire est un système d'équation ou d'inéquation non strictes appelées contraintes qui sont linéaires (les variables ne sont pas élevées au carré, ne sont pas des exposants, ne sont pas des multipliées entre elles,...) sous lesquelles on doit optimiser une fonction objectif également linéaire.

Les hypothèses de la programmation linéaire sont :

**a-Proportionnalité et additivité (ou linéarité)**

- **Proportionnalité** : signifie que la contribution des variables de décision à la fonction objectif et aux contraintes est proportionnelle à leur valeur par exemple : $3x_1 - x_2$.
- **Additivité** : signifie que la contribution de chaque variable est indépendante de la valeur des autres variables par exemple : $3x_1 - x_2$ mais pas $3x_1x_2$.

Si cette hypothèse est violée alors il s'agit d'une programmation non linéaire.

**b-Divisibilité :**

Signifie que les variables prennent des valeurs fractionnaires.
Si cette hypothèse est violée alors il s'agit d'une programmation en nombre entiers.

**c-Certitude :**

Chaque paramètre est connu précisément. Si cette hypothèse est violée alors il s'agit de programmation stochastique.

Plusieurs problèmes de décisions pratiques se décrivent sous forme de programmes linéaires comme suit :

$$\begin{cases} \min Z = & c_1x_1 & + & c_2x_2 & + & \cdots & + & c_nx_n & & \\ & a_{11}x_1 & + & a_{12}x_2 & + & \cdots & + & a_{1n}x_n & = & b_1 \\ & a_{21}x_1 & + & a_{22}x_2 & + & \cdots & + & a_{2n}x_n & = & b_2 \\ & \cdots & & & & & & & & \\ & a_{m1}x_1 & + & a_{m2}x_2 & + & \cdots & + & a_{mn}x_n & = & b_m \\ & x_1 & , & x_2 & , & \ldots & , & x_n & \geq & 0 \end{cases}$$

Ou encore :

$$\begin{cases} \min\ F = \sum c_j x_j \\ \\ \quad \sum a_{ij} x_j = b_j \\ x_j \geq 0 \\ i = 1 \cdots m \\ j = 1 \cdots n \end{cases}$$

Sous forme matricielle :

$$\begin{cases} \min\ c^T x \\ Ax = b \\ x \geq 0 \end{cases}$$

Avec :

$$A = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix}$$

$$b = \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix}$$

$$x = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$$

Objectif : chercher la combinaison des valeurs optimale qui satisfait les contraintes.

## 1.3 La convexité

### 1.3.1 Quelques définitions

#### Ensemble convexe

Un ensemble C est convexe si pour tout $x_1, x_2 \in C$, et tout $\theta \in [0,1]$

$$\theta x_1 + (1-\theta) x_2 \in C$$

**Fonction convexe**

Une fonction $f : D \longrightarrow R$ est convexe si son domaine de définition D est un ensemble convexe et si pour tout $x_1, x_2 \in C$, et tout $\theta \in [0,1]$

$$f(\theta x_1 + (1-\theta)x_2) \leq \theta f(x_1) + (1-\theta) f(x_2)$$

Si cette dernière inégalité est stricte, pour tout $x_1 \neq x_2$ on parlera alors d'une convexité stricte.

**Fonction concave**

Une fonction $f : D \longrightarrow R$ est concave si $-f$ est convexe, et strictement concave si $-f$ est strictement convexe.

## 1.4 Méthodes de décomposition pour la résolution des PL : génération de colonnes

De nombreux problèmes d'optimisation rencontrés dans la pratique sont complexes et de grande taille, ce qui les place hors des capacités de résolution des logiciels disponibles aujourd'hui, même les plus spécialisés. Pour pouvoir les traiter, des méthodes de décomposition de l'espace des solutions sont utilisées. La décomposition consiste à décomposer le problème en un ou plusieurs sous-problèmes, généralement coordonnés par un problème maître.

Les formulations issues de décompositions comportent le plus souvent un nombre très important de variables, potentiellement exponentiel, ce qui les rend hors des capacités de résolution des logiciels disponibles aujourd'hui et nécessite donc des méthodes de résolution dédiées. La génération de colonnes est une méthode de résolution très adaptée pour la résolution de problèmes de grande taille car elle consiste à résoudre un problème en ne considérant qu'un sous-ensemble de ses variables. De nouvelles variables sont injectées au fur et à mesure des itérations jusqu'à obtenir la base optimale.

### 1.4.1 Principe de la génération de colonnes

Considérons le programme linéaire à n variables et m contraintes :

$$(P) \left\| \begin{array}{lll} \max z = & \sum_{j \in J} c_j x_j & \\ & \sum_{j \in J} A^j x_j & \leq a \\ & x_j \geq 0, & \forall j \in J \end{array} \right.$$

où $J$ est l'ensemble des indices des variables, avec : $(|J| = n), (c_j, A^j) \in (R \times R^m), \forall j \in J$ et $a \in R^m$ .

Soit $B$ une matrice de dimensions $(m \times m)$ associée à une base réalisable de (P) et $x_{(B)}$ une solution de base associée. La solution duale correspondante est donnée par :

$$\pi = c_B B^{-1}$$

La résolution du problème $(P)$ par l'algorithme primal du simplexe consiste à améliorer itérativement la solution de base $x_{(B)}$ en sélectionnant, parmi toutes les variables hors base, une variable de coût réduit améliorant, i.e. positif pour un problème en maximisation. Le coût réduit d'une variable $x_j$ est donné par : $\bar{c}_j = c_j - \pi A^j$ . La colonne qui rentre dans la base peut être obtenue en résolvant le problème d'évaluation :

$$\bar{c}_{j^*} = \max_{j \in J} c_j - \pi A^j \qquad (**)$$

Si $\bar{c}_{j^*} > 0$ alors la solution courante peut être améliorée en introduisant $x_{j^*}$ dans la base. Sinon, la base courante $B$ est optimale pour $(P)$.

Si on est amené à manipuler un nombre démesuré de variables (plusieurs millions), il est alors très difficile de les énumérer explicitement et par conséquent, la résolution de (**) devient ardue, voire impossible. Dans ce cas, les méthodes de décomposition décomposé le problème d'origine en un ou plusieurs sous-problèmes plus simples à résoudre, l'ensemble étant généralement coordonné par un programme linéaire appelé le problème maître.

Le principe de l'algorithme de génération de colonnes est similaire à celui de l'algorithme du simplexe et consiste à résoudre le problème $(P)$ avec un ensemble réduit de ses variables, puis à l'alimenter itérativement avec de nouvelles colonnes jusqu'à atteindre l'optimalité. Ainsi, on construit à une itération $k$ donnée, un problème $(P^k)$ (appelé problème maître restreint) à partir de $(P)$ en utilisant un sous-ensemble de colonnes $J^k \subseteq J$ :

$$(P^k) \left\| \begin{array}{ll} \max & \sum_{j \in J^k} c_j x_j \\ & \sum_{j \in J^k} A^j x_j \quad \leq a \\ & x_j \geq 0, \quad \forall j \in J^k \end{array} \right.$$

Soient $B^k$ une base optimale et réalisable de $(P^k)$, $x^k$ et $\pi^k$ les solutions primales et duales associées. On cherche donc une colonne de coût réduit positif (maximal) qui rentre dans la base en résolvant le problème d'évaluation (**) appelé sous-problème. Deux cas peuvent se présenter :

a. Soit $\bar{c}_{j^*} > 0$ : la colonne $j^*$ est rajoutée à l'ensemble $J^k$ et le problème $(P^k)$,augmenté d'une colonne, est réoptimisé sur l'ensemble $J^{k+1} = J^k \cup \{j^*\}$ .

b. Ou bien $\bar{c}_{j^*} \leq 0$ : aucune variable ne possède un coût réduit positif et par conséquent la base $B^k$ est également optimale pour $(P)$ .

## 1.4.2 Faisabilité et optimalité sous l'incertitude

### Approche de la valeur estimée

Cette approche consiste à remplacer les variables aléatoires du problème stochastique par leurs espérances mathématiques. Considérons le PL stochastique suivant :

$$\left\{ \begin{array}{llllllll} \min z = & -x_2 & & & & & & \\ & x_1 & + & x_2 & + & x_3 & = & 2 \\ & \tilde{a}_{21} & x_1 + & \tilde{a}_{22} & x_2 + & x_4 & = & 2 \\ & -1 & \leq & x_1 & \leq 1 & & & \\ & x_j & \geq 0 & j & = & 2, & 3, & 4 \end{array} \right. \qquad (1.1)$$

Supposant que les coefficients de $x_1$ et de $x_2$ dans (1.1) ne sont pas connus aves certitude, et qu'on connait leur distribution jointe.

$$(\tilde{a}_{21},\tilde{a}_{22}) = \begin{cases} (1 \quad , \quad \frac{3}{4}) & \text{avec une probabilité } 0,5 \\ \\ (-3 \quad , \quad \frac{5}{4}) & \text{avec une probabilité } 0,5 \end{cases}$$

Dans ce cas, En remplaçant les coefficients par leurs valeurs estimées :$E(\tilde{a}_{21}) = -1$ et $E(\tilde{a}_{22}) = 1$, on trouvera la solution faisable : $(x_1,x_2,x_3,x_4) = (0,2,0,0)$ ; maintenant on veut examiner sa faisabilité sous l'incertitude. La contrainte correspondante à (1.1) peut être équitablement :

$$x_1 \quad + \quad \tfrac{3}{4}x_2 \quad + x_4 \quad = 2$$

*ou*

$$-3x_1 \quad + \quad \tfrac{5}{4}x_2 \quad + x_4 \quad = 2$$

La solution $(0,2,0,0)$ ne satisfait aucune des contraintes et elle est de ce fait infaisable sous incertitude.

**Remarque 1.1** Remplacer les variables aléatoires par leurs estimations ne fournit pas toujours une solution faisable à l'égard des variables aléatoires.

### Approche d'analyse de scénarios

Cette approche imite le processus de reporter toutes les décisions jusqu'au dernier moment, où toutes les incertitudes seront levées. Par conséquent, les programmes linéaires associés à toutes les réalisations possibles peuvent être résolus. Cela donne lieu à plusieurs vecteurs de décision, un pour chaque réalisation possible des variables aléatoires.

En travaillant toujours avec l'exemple précédent, on considéra l'existence de deux scenarios pour le vecteur aléatoire $\tilde{a} = (\tilde{a}_{21},\tilde{a}_{22})$ donné comme suit :

$$\text{Scenario 1} : \tilde{a}_1 = (\tilde{a}_{21},\tilde{a}_{22}) = (1,\tfrac{3}{4})$$

$$\text{Scenario 2} : \tilde{a}_1 = (\tilde{a}_{21},\tilde{a}_{22}) = (-3,\tfrac{5}{4})$$

La solution associée au premier scenario est :$(x_1,x_2,x_3,x_4) = (-1,3,0,0.75)$
La solution associée au deuxième scenario est : $(x_1,x_2,x_3,x_4) = (0.11,1.88,0,0)$

**Remarque 1.2** Comme pour l'approche précédente, aucune solution n'est faisable pour la réalisation alternative. Autrement dit, chaque solution a une probabilité de $\frac{1}{2}$ d'échouer à satisfaire la contrainte.

### Conclusion

Comme illustré dans les exemples précédents, un modèle de prise de décisions sous incertitude doit prendre en considération les conséquences des futures infaisabilités d'une manière explicite.

**Modèles de recours**

L'idée est d'optimiser les décisions que l'on doit prendre au moment $t_0$ tout en tenant compte des conséquences de ces décisions pour toute réalisation des aléas qui pourrait se produire aux instants ultérieurs. Le mot « recours » révèle la possibilité dont on dispose de remédier à une décision éventuellement trop optimiste à l'instant $t_0$ en mettant en œuvre des actions correctives, évidemment plus chères, qui satisfont pourtant les contraintes posées aux instants ultérieurs $t_i$. On précise qu'après avoir pris la décision au moment $t_0$ on passe au moment $t_i$ où tous les événements inconnus jusqu'à cet instant se révèlent. Par conséquent, l'étude de tels cas a un intérêt seulement si on dispose de quelques informations sur les réalisations des événements qui nous sont inconnus au moment $t_0$ (information comme par exemple la distribution de probabilité, des scénarios, la moyenne, la variance ... etc).

Le problème peut s'étendre en plusieurs étapes suivant l'horizon d'étude, mais considérons uniquement le cas de deux étapes seulement. Les variables de la première étape sont considérées comme étant des variables structurantes, stratégiques qui ne peuvent plus changer à la seconde étape. La décision à la première étape doit être prise en respectant des contraintes propres à cette étape. La seconde étape comprend des actions (pénalisantes) qui peuvent être entreprises afin de satisfaire des contraintes qui font intervenir des variables de la première étape. Toutes ces idées sont résumées dans le programme suivant [36] :

$$\begin{cases} \min \quad c^T x \quad + \quad E_\xi[Q(x,\xi)] \\ \qquad Ax \quad = \quad b \\ \\ \qquad Q(x,\xi) \quad = \quad \min\limits_{y\in Y} q^T(\xi) y(\xi) \\ \qquad T(\xi)x \quad + \quad W(\xi)y(\xi) \quad = \quad h(\xi) \end{cases} \tag{1.2}$$

où $x$ et $c$ sont des vecteurs de $R^{n_1}, A$ une matrice de dimension $m_1 \times n_1, b \in R^{m_1}$ le vecteur du second membre. De façon similaire $y, q \in R^{n_2}, h \in R^{m_2}$ le vecteur aléatoire qui dépend de la variable aléatoire $\xi \in R^r, E_\xi$ l'espérance mathématique par rapport à la variable aléatoire $\xi$, alors que : $T, W$ sont des matrices qui dépendent de $\xi$ et de dimensions appropriées :$T : m_2 \times n_1$ et $W : m_2 \times n_2$

Les variables $x$ représentent les décisions avant que toute réalisation de la variable aléatoire $\xi$ ne soit connue, alors que les variables $y$ représentent les décisions prises, une fois que $\xi$ sera observée. On cherche à minimiser sur les variables de la première étape $x \in \chi$. Mais en veille en même temps à ce que les coûts au seconde niveau, après, la réalisation de $\xi$, soient minimaux en moyenne.

Il faut bien noter que les variables $x$ de la première étape resteront les mêmes quelle que soit la réalisation de la variable aléatoire $\xi$, alors que les variables $y$ de la seconde étape dépendront de cette réalisation, puisque elles cherchent à minimiser ses conséquences. Le fait que l'on ne se permet pas de faire dépendre $x$ de $\xi$, c'est-à-dire que pour tout $\xi_i$ d'avoir une décision $x_i$ différente, s'appelle la non-anticipativité et c'est la propriété qui couple les variables du premier avec

celles du second niveau. Si on ne tenait pas compte de cette propriété, alors pour toute réalisation de la variable aléatoire $\xi_i$ on changerait à volonté les variables de la première étape, en choisissant chaque fois une valeur $x_i$ telle qu'elle minimise la fonction coût [36].

Dans de nombreuses applications on pourra supposer qu'il existe une mesure de probabilité $P$ telle que $P(\xi = \xi^s) = P_s$ avec un support fini $\Xi$ donné par $\Xi = (\xi^1, \xi^2, \ldots, \xi^S) \subset R^r$. L'ensemble des réalisations possibles $\xi$ est fini, et les réalisations $\Xi$ s'appellent souvent des scénarios, dont $P$ est la probabilité d'occurrence ou bien l'importance que l'on leur accorde. Cette hypothèse va nous permettre de passer de (1.2) au programme suivant (1.3).

$$\begin{cases} \min\limits_{x\in\chi, y^s\in Y^S} & cx+ & p^1q^1y^1+ & p^2q^2y^2+ & \cdots+ & p^Sq^Sy^S \\ & Ax & & & = b & \\ & T^1x & + & W^1y^1 & = h^1 & \\ & T^2x & + & W^2y^2 & = h^2 & \\ & \vdots & & & & \\ & T^Sx & + & W^Sy^S & = h^S & \\ & x\in\chi, & y^1\in Y^1, \ldots, & y^S\in Y^S & & \end{cases}$$

où bien :

$$\begin{cases} \min & c^Tx+ & p^sq_s^Ty^s & & \\ & Ax & & = b & \\ & T^sx & + & W^sy^s & = h^s \\ & \forall s = 1, \ldots, S & & & \end{cases} \qquad (1.3)$$

Où les indices $1, 2, \ldots, S$ font référence aux scénarios correspondants. Par ailleurs dans notre cas, $\chi$ désigne un polyèdre convexe construit par un ensemble de contraintes linéaires.

On s'aperçoit comment un tel programme peut déjà avoir de grandes dimensions. Le nombre de scénarios $S$ peut être grand, puisque si on a $V$ variables aléatoires avec $S_v$ valeurs possibles chacune, on obtient $S = \prod_{v=1}^{V} S_v$ scénarios. Par exemple, pour $V = 10$ variables aléatoires (demande de 10 clients), avec $S_v = 5$ valeurs possibles chacune, on obtient $S = 5^{10}$ scénarios, déjà pour un problème simple.

### Différents cas de recours

En observant la formulation (1.2) on peut distinguer différents cas de recours en fonction des types de $q(\xi), W(\xi), y(\xi)$. Les cas qui méritent une étude plus spéciale et qui sont cités dans la littérature sont le recours fixe, le recours complet et le recours simple.

**Définition 1 (Recours fixe)** Le recours est considérée comme étant fixe ou déterministe si les valeurs $q$ et $W$ d'un programme avec recours comme le (1.2) ne dépendent pas de $\xi$, leurs valeurs sont donc à priori connues.

Si on considère le terme $q$ du programme (1.2) comme des pénalités pour toute action corrective $y$ que l'on doit effectuer à la seconde étape, le recours fixe exige que ces pénalités ne dépendent pas de la réalisation de la variable aléatoire. Ce cas est assez courant d'ailleurs : dans le cas des problèmes de transport, si le nombre de camions (décision à la première étape) n'est pas suffisant pour couvrir la forte demande du lendemain, on fait appel à la location, dont le prix ne dépend pas de la demande de notre clientèle. Il existe toutefois des cas les pénalités dépendent des variables aléatoires, lorsque par exemple on est obligé, à cause d'une forte demande, d'acheter des quantités de gaz, le prix à payer dépendra de la demande.

**Définition 2 (Recours complet)** Le recours fixe est complet si $x \in R^{n_1}$, il existe $y \in Y$ tels que $Q(x,\xi) < +\infty \; \forall \xi \in \Xi$. En d'autres termes le problème de seconde étape est faisable.

**Définition 3 (Recours relativement complet)** Le recours fixe est relativement complet si $x \in \chi \subseteq R^{n_1}$, il existe $y \in Y$ tel que $Q(x,\xi) < +\infty \; \forall \xi \in \Xi$

Notons que la différence entre les deux types de recours porte sur l'ensemble des $x$ pour lesquels il existe une solution au second niveau. Si cet ensemble est le domaine de définition à la première étape le recours est relativement complet, tandis que si c'est $R^{n_1}$, alors le recours est complet.

**Remarque 1.3** il est claire que le recours complet implique le recours relativement complet.

Selon cette définition [36], si pour chaque décision au premier niveau, on peut garantir, pour tout scénario pouvant se produire, l'existence des décisions au second niveau, et par conséquent on arrive toujours à satisfaire les contraintes du second niveau, alors le recours est complet. Dans le cas contraire, le système n'a pas moyen de remédier aux décisions, éventuellement destructives, de la première étape. Considérons à titre d'exemple le problème de transport stochastique. Si on décide à la première étape d'installer des entrepôts qui ne disposent pas d'une capacité suffisante pour couvrir la demande réelle (révélée à la seconde étape) qui peut se produire suivant un scénario (pessimiste), le problème résultant devient non-réalisable, puisque on n'arrivera jamais à satisfaire la demande clientèle. Ceci est un exemple où le recours n'est pas complet. Si, au contraire, on a la possibilité de recourir à d'autres actions, par exemple locations des entrepôts, supplémentaires, alors on arrivera à satisfaire la demande à un coût certes plus élevé mais fini.

Dans les applications, le plus souvent, on choisit la matrice $W$ telle que le recours soit complet et les matrices correspondantes portent le nom de matrices de recours complet. Une question intéressante est de savoir à l'avance si une matrice donnée est une matrice de recours complet. On dit qu'une matrice $W$ de dimension $m \times n$ est une matrice de recours complet si et seulement si :

- son rang est égal à m
- si on considère que les colonnes linéairement indépendantes sont les $m$ premières colonnes $W_1, W_2, \ldots, W_m$ alors les contraintes linéaires (1.3) admettent une solution réalisable.

$$\begin{cases} Wy = 0 \\ y_1 \geq 1 \\ y > 0 \\ i = 1, \ldots, m \end{cases} \qquad (1.4)$$

**Définition 4 (Recours simple)** Le recours simple est un cas particulier du recours complet fixe où $W$ est une matrice identité d'ordre $m$.

### Propriétés de la programmation linéaire avec recours

Deux propriétés de la fonction de recours vont être présentées dans ce paragraphe : la convexité et la linéarité par morceaux.

On s'intéressera d'abord à la convexité de la fonction de recours. Considérons problème de départ (1.4). L'espace des solutions du problème de recours est construit à travers des équations linéaires, il est donc convexe. De plus, si la fonction $E[Q(x,\xi)]$ est convexe en $x$ on a finalement un problème convexe. Si de plus la fonction $Q(x,\xi)$ est convexe pour tout $\xi$ alors par la linéarité de l'espérance, il est évident que la fonction à minimiser est convexe :
$\forall x \in \Xi$, pour $\tilde{x} = \theta x_1 + (1-\theta)x_2$ on a :

$$Q(\tilde{x},\xi) \leq \theta\, Q(x_1,\xi) + (1-\theta)\, Q(x_2,\xi)$$

$$E[Q(\tilde{x},\xi)] \leq \theta\, E[Q(x_1,\xi)] + (1-\theta)\, E[Q(x_2,\xi)]$$

et donc :
$c^T x + E[Q(\tilde{x},\xi)]$ est une fonction à minimiser convexe.

On va maintenant montrer la linéarité par morceaux de la fonction de recours. On se place dans le cas d'un recours fixe ($W$ et $q$ sont déterministes ; $T$ pas nécessairement). On construit le dual :

$$Q(x) = \max\ y^T(h(\bar{\xi}) - Tx)$$

Le polyèdre $y : W^T y \leq q$ est fermé et convexe ; par conséquent, il y a un nombre fini de points extrêmes. La fonction $Q(x)$ peut être réécrite comme suit :

$$Q(x) = \max\ y_i^T(h(\bar{\xi}) - Tx) \quad i = 1,\ldots,N$$

Il est bien connu de la théorie de la programmation linéaire que cette fonction, représentée comme le maximum point à point d'un ensemble de fonctions affines, est une fonction linéaire par morceaux.

### Modèles de L-shaped

En 1962 Benders [J.F.Benders,1962 ] a proposé une méthode de décomposition consistant à partitionner les variables d'un programme linéaire en deux catégories. Son objectif a été de résoudre des problèmes de programmation mixte.

En s'inspirant de cette idée, Wets en 1966 [R.J.B. Wets,1966] a appliqué la décomposition de Benders sur les problèmes de recours en deux étapes. En 1969 Van Slyke et Wets [R. Van Slyke

and R.J.B Wets, 1969] ont développé la première méthode de la programmation stochastique sous le nom « L-Shaped » destinée à résoudre les problèmes représentés en tant que modèles de recours en deux étapes.

L'idée de la méthode L-Shaped consiste à créer deux problèmes complémentaires, le problème maître (master) et les problèmes satellites ou sous-problèmes ou problèmes-esclaves. Les problèmes satellites (un pour chaque scénario) reçoivent comme entrée les valeurs des variables du premier niveau et calculent les variables de recours propres à chaque scénario ainsi que le coût associé. Le problème maître comporte les variables du premier niveau et une description approximée de la fonction de recours dont la forme exacte n'est pas connue. Si une telle fonction était donnée de façon explicite, alors on connaîtrait automatiquement l'impact de chaque solution du premier niveau au second. Le but est de construire cette fonction de recours à partir des solutions proposées au fur et à mesure par le problème maître et en ajoutant des coupes déduites de la résolution des problèmes-satellites. L'algorithme ne démarre qu'avec les contraintes du premier niveau :

$$\begin{cases} \min c^T x + \theta \\ Ax = b \\ \theta \geq -M \\ x \geq 0 \end{cases} \qquad (1.5)$$

$\theta$ est la variable qui donne l'approximation de la fonction réelle de recours. Si on n'y rajoute pas la contrainte supplémentaire $\theta \geq -M$ à la première itération de l'algorithme (où $-M$ est une borne inférieure connue de la fonction de recours (ou un très grand nombre négatif)) ; il est évident que le problème sera non-borné et une solution $x$ ne sera pas obtenue. À chaque itération on injecte la variable $x$ à chaque sous-problème du second niveau. On demande de chaque sous problème deux informations :

- la valeur de sa fonction objectif
- le vecteur des variables duales optimales

Ces informations se regroupent en une coupe, constituant une combinaison linéaire de tous les problèmes satellites.

Au moment où la fonction de recours $Q(x)$ sera **très proche** de son approximation $\theta$, la méthode s'arrête avec la précision souhaitée. Tant que ceci n'est pas le cas, l'algorithme procède en ajoutant des coupes dont l'existence et la validité est assurée par la théorie de la dualité (trouver une direction d'amélioration).

Au départ on résout le problème (1,5), on récupère la solution $(x^*, \theta)$ et on fait passer cette solution à chaque sous-problème :

$$\left\| \begin{array}{c} \textit{primal} \\ \\ \min \ q(\xi)^T y \\ Wy \geq h(\xi) - Tx \\ y \in Y \end{array} \right.$$

$$\left\| \begin{array}{c} \textit{dual} \\ \\ \max \ \lambda^T (h(\xi) - Tx) \\ \lambda W \leq q(\xi) \\ \lambda \geq 0 \end{array} \right.$$

Si lors d'une itération et pour un $x^*$ donné le problème dual n'est pas borné, ce qui signifie que le primaire est non-réalisable, on rajoute dans le problème maître la coupe de réalisabilité suivante :

$$\hat{\lambda}^T T x \geq \hat{\lambda}^T h \qquad (1.6)$$

Cette contrainte est satisfaite également pour la solution $x^*$ qui rendait le problème non réalisable. De cette manière, on arrive à imposer implicitement le recours relativement complet. À noter que $\hat{\lambda}$ est une direction extrême du cône polaire correspondant à la matrice $W$ et elle est calculée pour un point donné $x^*$ à partir du problème suivant :

$$\left\| \begin{array}{c} \max \ \lambda^T (h(\xi) - x^*) \\ \lambda^T W \leq 0 \\ \| \lambda \| \leq 1 \end{array} \right.$$

Si tous les problèmes admettent une solution, on essayera de faire mieux en obtenant une direction d'amélioration de la fonction réelle de recours (ceci pour chaque scénario séparément) et tracer ainsi le plan sécant qui approxime le mieux la fonction de recours au point courant $x^*$. Pour tracer cet hyperplan on a besoin :
De la valeur de la fonction objectif $f_i(x^*)$. Pour chaque sous problème $i$ auquel on a passé la solution $x^*$, ainsi que : le vecteur dual optimal $\lambda_i^*$ du sous-problème $i$.
Si on se restreint au sous problème $i$ (correspondant au scénario $i$), l'hyperplan d'appui au point $x^*$ est donné par la formule suivante :

$$Q_i(x) = f_i(x^*) - \lambda_i^{*T} T_i (x - x^*)$$

Par dualité on aura :

$$Q_i(x) = \lambda_i^{*T} (h_i - T_i x)$$

La fonction de recours est la combinaison linéaire de la fonction de chaque sous problème différent. On a donc :

$$Q^*(x) = \sum_i p_i Q_i(x^*)$$

Où $p_i$ est la probabilité d'occurrence du scénario $i$ qui pondère la partie $Q_i(x^*)$ correspondant au scénario $i$.

La fonction $Q(x)$ étant convexe, le support se trouve géométriquement en dessous du graphe de la fonction. La coupe à rajouter dans le maître, qui regroupe toute information de tout sous

problème est la suivante :

$$\theta \geq \sum_i p_i(f_i(x^*) - \lambda_i^{*T} T_i(x - x^*)) \qquad (1.7)$$

$$\Updownarrow$$

$$\theta \geq \sum_i p_i \lambda_i^{*T} (h_i - T_i x) \qquad (1.8)$$

Les coupes du type (1.6) sont appelées coupes de réalisabilité et celles du type (1.7) ou (1.8) coupes d'optimalité. A chaque itération on rajoute au problème maître soit une coupe de réalisabilité, soit une coupe d'optimalité. Au cours des itérations, $\theta$ sera le maximum point à point d'une famille d'inégalités qui approximeront la fonction de recours. Si on arrive à trouver un $\theta^*$ dont la valeur vaudra $Q(x^*)$, alors on aura trouvé l'optimum, comme $\theta$ représente une minorante de la fonction réelle de recours $Q(x)$. Pratiquement on s'arrête lorsque la différence relative entre $\theta$ et $Q(x)$. est plus petite que $\varepsilon$.

**L-shaped en version multi coupes**

En 1988 Birge et Louveaux [John R. Birge and Fran¸cois V. Louveaux, 1988] ont proposé une variante de la méthode L-Shaped en remplaçant la seule coupe d'optimalité qui regroupe toutes les informations de chaque scénario par plusieurs coupes (une par scénario), introduisant ainsi la version multi-coupes de la méthode L-Shaped. On n'a plus un seul terme $\theta$ qui décrit la fonction de recours, mais plusieurs $\theta_i$ un pour chaque scénario $i$. Les coupes que l'on rajoute au maître à chaque itération sont donc les suivantes :

$$\theta_i \geq \lambda_i^{*T} (h_i - T_i x) \ \forall i$$

La fonction objectif du problème maître ne comporte plus un seul terme $\theta$ mais une combinaison linéaire de. A la k-ième itération, le problème maître ressemble à :

$$\begin{cases} \min c^T x + \sum p_s \theta_s \\ \\ Ax = b \\ \theta_s \geq \lambda_s^{*T} (h_s - T_s x_i^*) \ \ \forall s, i \leq k \\ x \geq 0 \end{cases}$$

La mémoire requise est bien entendu plus importante, mais la méthode exploite bien l'information déjà recueillie des précédentes itérations et donc globalement le temps de calcul est en pratique réduit. Ruszczyiiski [A. Ruszczy´nski and A.Swic¸1997], tanowski cite pourtant qu'il avait auparavant observé qu'une agrégation éventuelle des coupes pourrait ralentir jusqu'à cinq

fois la convergence.
Il suffit de considérer un problème avec des millions de scénarios (réalisations de la variable aléatoire à la seconde étape) et d'imaginer qu'à chaque itération tous les scénarios renverront leurs coupes, une par une, au problème-maître. Il y aura un moment où le temps de calcul pour la résolution du maître sera très lent.

### L'inconvénient de la méthode de L-Shaped

Les coupes à chaque itération que l'on ajoute s'additionnent aux coupes existantes et par conséquent le nombre de coupes dans le problème maître peut augmenter sans pouvoir maîtriser sa taille. Surtout en version multi-coupes, quand le nombre de scénarios devient important, les dimensions du problème-maître en termes de contraintes augmentent très vite.

# 2

# Optimisation multiobjectifs

## Sommaire

# Introduction

L'optimisation est la tâche de trouver une ou plusieurs solutions qui minimisent (ou maximisent) un ou plusieurs objectifs spécifiés et lesquelles satisfont toutes les contraintes (si il y en a). Un problème d'optimisation mono-objectif implique une seule fonction objectif. En revanche, l'optimisation multiobjectif considère plusieurs objectifs souvent incompatibles à optimiser simultanément. Dans un tel cas, il n'y a habituellement pas qu'une seule solution optimale, mais un ensemble d'alternatives avec différentes valeurs de fonctions coût, appelées solutions Pareto optimales, ou solutions non-dominées.

En dépit de l'existence de plusieurs solutions Pareto optimales, en pratique, une seule de ces solutions est choisie. Donc, dans le processus de résolution des problèmes multiobjectif, il y a au moins deux étapes importantes : une première qui consiste à trouver les solutions Pareto optimales (implique une procédure informatisée) et une deuxième étape de la prise de décision pour choisir la meilleure solution parmi celles trouvées dans la première étape. Cette dernière nécessite l'information de préférence du décideur.

Dans ce travail on va traiter la première étape, c'est à dire la recherche de solutions non dominées. Ce chapitre est consacré au rappel de quelques notions de l'optimisation multiobjectif ainsi que les approches de résolution.

## 2.1 Problème d'optimisation multiobjectif

La phrase « Optimisation multiobjectif », est synonyme avec « Optimisation multicritères » ou « Optimisation multi performances », dans [5] l'optimisation multiobjectif est définit comme un problème de recherche d'un vecteur de variables de décision qui satisfait les contraintes et optimise un vecteur ayant comme éléments les fonctions objectifs. Ces fonctions sont souvent en conflit, d'où le terme « Optimiser » veut dire trouver une solution qui puisse donner les valeurs de toutes les fonctions objectifs acceptables au décideur [29].

### Formulation

La formulation générale d'un problème d'optimisation multiobjectif est la suivante :

$$optimiser \quad z(x) = (z_1(x), z_2(x), \ldots, z_p(x)),\ x \in S$$

Où $p \geq 2$ représente le nombre d'objectifs à optimiser, $x$ représente un vecteur de variables de décision, $S = \left\{x \in R^n,\ g_j(x) < 0,\ x > 0\right\}$ est l'ensemble de solutions réalisables associé à des contraintes d'égalité, d'inégalité et des bornes explicites (espace des décisions) et $Z(x)$ est le vecteur des objectifs à optimiser $z^k$ et $g_j$ des fonctions à valeurs réelles du vecteur de décision. $(k = 1, 2, \ldots, p)$ et $(j = 1, 2, \ldots, m)$.

Dans le cadre de l'optimisation multiobjectif, le plus souvent le décideur raisonne plutôt en termes d'évaluation d'une solution sur chaque critère. L'ensemble $Y = Z(S)$ représente les points réalisables dans l'espace des critères, et $z = (z_1, z_2, \ldots, z_p)$ avec $Y_k = z_k(x)$ représente un point de l'espace des critères.

## 2.2 Solutions non dominées et solutions Pareto Optimales

La plupart des algorithmes d'optimisation multiobjectif utilisent le concept de dominance dans leur recherche. Ici, nous définissons le concept de dominance et les termes apparentés et plusieurs techniques pour identifier des solutions non dominées dans une population finie de solutions.

### 2.2.1 Solutions spéciales

Nous définissons en premier quelques solutions spéciales qui sont souvent utilisées dans les algorithmes de l'optimisation multiobjectif.

#### Vecteur Objectif Idéal

Le vecteur idéal $Z^*$ du problème est le vecteur de l'espace des critères dont chaque composante $Z_k$ est la solution du problème d'optimisation de la fonction $Z(x)$ sous les contraintes du problème. Dans le problème de maximisation, le vecteur idéal $Z^*$ est le vecteur qui optimise chacune des fonctions objectifs c'est-à-dire : $Z^* = max(z_k(x), k = 1, \ldots, p)$

**Définition 5** Le vecteur $x_1$ est dit dominé par un autre vecteur $x_2$ si :

a. $x_1$ est au moins aussi bon que $x_2$ dans tous les objectifs, et.

b. $x_1$ est strictement meilleur que $x_2$ dans au moins un objectif.

Mathématiquement, cela est expliqué comme suit :

$$Z_k(x_1) \geq Z_k(x_2),\ \forall k \in 1, \ldots, p$$

$$\exists k^{'} \in \{1, \ldots, p\}\ : Z_{k'}(x_1) \geq Z_{k'}(x_2)$$

Par la suite, si la solution $x_1$ domine la solution $x_2$ on écrira : $x_1 \prec x_2$

**Définition 6** Soit $Z$ et $Z^{'} \in R^p$ on dit que :

a. $Z$ domine fortement $Z^{'}$ si et seulement si $Z_k \prec x_k^{'}\ \forall k \in \{1, \ldots, p\}$ ($Z$ est meilleur que $Z^{'}$ sur tous les critères).

b. $Z$ est faiblement non dominé, s'il n'existe pas $Z^{'} \in R^p$ qui domine fortement $Z$.

### Propriétés de la relation de dominance

La relation binaire de dominance $\prec$ telle qu'elle est définie au dessus ;

a. N'est pas réflexive car une solution ne se domine pas par elle-même.

b. N'est pas symétrique car on a jamais : $x_1 \prec x_2$ et $x_2 \prec x_1$.

c. Elle est transitive, car si $x_1 \prec x_2$ et $x_2 \prec x_3$ implique que $x_1 \prec x_3$.

Notons que pour toute paire de solution $x^1$ et $x^2$ une et seulement une des affirmations suivantes est vraie :

a. $x_1$ domine $x_2$

b. $x_1$ est dominé $x_2$

c. $x_1$ et $x_2$ sont équivalentes au sens de dominance.

Par la suite, les solutions équivalentes au sens de la dominance seront parfois évoquées comme solutions équivalentes au sens de Pareto ou, encore, comme solutions Pareto équivalentes.

### Optimalité de Pareto

Soit $P$ un ensemble de solutions candidats d'un problème d'optimisation multiobjectif. L'ensemble $P^{'} \subseteq P$ composé de tous les éléments de $P$ qui ne sont dominés par aucun élément de $P$, est dit sous ensemble non dominé de l'ensemble de solutions $P$. Les solutions de cet ensemble $P^{'}$ sont dites Pareto-équivalentes et l'ensemble $P^{'}$ est dit Frontière de Pareto.

Un sous ensemble de solutions sont Pareto-équivalentes entre elles même s'il existe d'autres solutions qui dominent des solutions de l'ensemble en question.

### Point idéal et point Nadir

Le vecteur idéal $y^* = (y_1^*, \ldots, y_m^*)$ est obtenu en optimisant séparément chaque fonction objectif $f_i$ c'est-à-dire : $y_i^* = f_i^*(x) : x \in D$ : espace faisable.
Généralement ce vecteur n'appartient pas à l'espace objectif réalisable mais il est dans certain cas utile en tant que référence, par exemple, pour normaliser les valeurs des objectifs.

A la différence du vecteur idéal qui représente les bornes inférieures de chaque objectif dans l'espace faisable, le vecteur de Nadir correspond à leurs bornes supérieures sur la surface de Pareto, et non pas dans tout l'espace faisable. Ce vecteur sert à restreindre l'espace de recherche ; il est utilisé dans certaines méthodes d'optimisation interactives.

## 2.3 Classification des approches multicritères

### 2.3.1 Approches a priori

Les solutions les plus intuitives pour résoudre des problèmes multiobjectif consistent souvent à combiner les différentes fonctions objectifs en une fonction d'utilité suivant les préférences du décideur. Dans ce cas le décideur est supposé connaître a priori le poids de chaque objectif et les mettre dans une fonction unique. Cela revient à résoudre un problème mono-objectif. Cependant dans la plupart des cas, le décideur ne peut pas exprimer clairement sa fonction d'utilité, soit par manque d'expérience ou d'information, soit par ce que les différent objectifs sont non commensurable.

### 2.3.2 Approches progressives ou interactives

Dans ces méthodes, les processus de décision et d'optimisation sont alternés. Par moment, le décideur intervient de manière à modifier certaines variables ou contraintes afin de diriger le processus d'optimisation. Le décideur modifie ainsi interactivement le compromis entre ses préférences et les résultats. Ces méthodes exigent une connaissance approfondie, de la part du décideur, des outils utilisés.

### 2.3.3 Approches a posteriori

Le décideur prend sa décision d'après un ensemble de solutions calculées par un solveur. Dans ce cas la qualité de la décision dépend du choix de la méthode de résolution. Car celle-ci va devoir donner un ensemble plus représentatif de l'espace des objectifs efficaces.

# 3

# Méthodes de résolution

## Sommaire

# Introduction

En présence d'un problème d'optimisation concret, le chercheur est confronté à la principale difficulté du choix d'une méthode efficace (lorsqu'elles existe), 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.

## 3.1 Méthodes Exactes

### 3.1.1 Méthode de pondération des fonctions objectif

Cette approche du résolution des problèmes d'optimisation multiobjectif est la plus évidente. D'ailleurs, on appelle aussi cette méthode l'approche naïve de l'optimisation multiobjectif [20]. 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 objectifs et leur appliquer un coefficient de pondération, ensuite faire la somme pondérée des fonctions objectif. On obtient alors une nouvelle fonction objectif [6].

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

$$g(x) \leq 0$$

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

### 3.1.2 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

de pondération des fonctions objectif. La fonction objectif ainsi obtenue s'appelle la fonction d'utilité de Keeney-Raiffa [26].

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

$$g(x) \leq 0$$

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

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

Une autre façon de transformer un problème d'optimisation multiobjectif en un problème mono objectif, est de convertir $m-1$ des $m$ objectifs du problème en contraintes et d'optimiser séparément l'objectif restant [6]. 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 [28] :

$$\left\| \begin{array}{c} opt\ f_i(x) \\ \\ f_1(x) \leq \varepsilon_1 \\ f_2(x) \leq \varepsilon_2 \\ \vdots \\ f_{i-1}(x) \leq \varepsilon_{i-1} \\ f_i(x) \leq \varepsilon_i \\ \vdots \\ f_m(x) \leq \varepsilon_m \\ g(x) \leq 0 \end{array} \right.$$

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 objectifs en contraintes, elle diminue la zone réalisable par paliers. Ensuite, le processus d'optimisation trouve le point optimal sur l'objectif restant.

### 3.1.4 Méthode lexicographique

Cette méthode est très intuitive. Elle consiste à considérer les fonctions objectifs une après l'autre et minimiser un problème d'optimisation mono objectif. Au fur et à mesure, on lui ajoute graduellement des contraintes [5].

**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 objectif. 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 objectif en contraintes d'égalité. De ce fait, nous prenons la seconde fonction objectif 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 :

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

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

## 3.2 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 les quels on trouve les définitions suivantes :

**Définition 7 (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. [30]

### 3.2.1 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 objectif 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 8 (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.

### 3.2.2 Les Algorithmes génétiques

Les AGs ont été largement utilisés dans la communauté multiobjectif. 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 [32]. 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 $i$ est sélectionnée suivant l'objectif $f_i$. Les $m$ sous-populations sont ensuite mélangées 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 $N$ [17]. 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 [22] :

- 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.

### Non Sorting Dominated Genetic Algorithm II (NSGA-II )

Srinivas et Deb [Srinivas & Deb, 1994] ont implémenté en 1994 la première procédure de ranking qui a été initialement introduite par Goldberg [Goldberg, 1989] nommé le **Non Sorting Genetic Algorithm (NSGA)**. Tous les individus non dominés de la population possèdent le rang 1. Ces individus sont ensuite enlevés de la population et l'ensemble suivant d'individus non-dominés est identifié et on leur attribue le rang 2.

Ce processus est réitéré jusqu'à ce que tous les individus de la population aient un rang. Une nouvelle version élitiste de cet algorithme, nommée **Non Sorting Genetic Algorithm II (NSGA-II )**, a été présentée dans [10]. NSGA-II intègre un opérateur de sélection, basé sur un calcul de la distance de crowding (ou surpeuplement) qui estime la densité de chaque individu dans la population. Comparativement à NSGA , NSGA-II obtient de meilleurs résultats sur toutes les instances ce qui fait de cet algorithme un des plus utilisés aujourd'hui.

Pour gérer l'élitisme, NSGA-II n'utilise pas d'archive externe pour stocker l'élite, comme le font d'habitude les autres AGs. NSGA-II assure qu'à chaque nouvelle génération, les meilleurs

individus rencontrés soient conservés.

A chaque itération $i$, l'algorithme utilise deux populations séparées $P_t$ et $Q_t$ de taille $N$. La population $P_t$ contient les meilleurs individus rencontrés jusqu'à l'itération $t$, et la population $Q_t$ est formée des autres individus, issus des phases précédentes de l'algorithme.

La première étape consiste à créer une population combinée $R_t = P_t \cup Q_t$ et à appliquer une procédure de ranking. Puisque tous les individus précédents et courants de la population sont inclus dans $R_t$, l'élitisme est assuré. La deuxième phase consiste à construire une nouvelle population $P_{t+1}$ contenant les $N$ meilleurs individus de $R_t$. La troisième phase consiste à remplir la population $Q_{t+1}$. Il suffit alors d'utiliser les opérateurs de sélection, croisement et mutation sur les individus de $P_{t+1}$, puis d'insérer les descendants dans $Q_{t+1}$.

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

L'algorithme SPEA2, proposé par Zitler et al dans [40],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 complété 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 :

$$fitness(p) = R(p) + D(p)$$

Dans cette équation, $R(p)$ est la performance préliminaire (raw fitness) de l'individu $p$, et $D(p)$ est sa densité. La performance préliminaire est calculée suivant des valeurs de **Strength** des solutions. $D(p)$ est calculée en utilisant une adaptation de la technique du $k^{ime}$ plus proche voisin

[33]. 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.

### 3.2.3 Le recuit simulé

Le premier algorithme multiobjectif 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 [31] 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 inferieure à 1. Dans le cas multiobjectif, 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 multiobjectif 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,

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 [35].

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.

### 3.2.4 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 [24].

Hansen [Hansen, 1998] a proposé une recherche taboue basée sur l'idée de la méthode **Pareto simulated annealing**[9]. 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 [21].

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 [3].

### 3.2.5 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. Plu-

sieurs 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 [18] et au problème du sac à dos multi-objectif [18]. 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 $GTS^{MOKP}$ 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.

### 3.2.6 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.

### 3.2.7 Colonie de fourmis

#### Introduction

Afin de générer un maximum de profit , les entreprises s'attachent à entreprendre uniquement des projets qui correspondent à la finalité et à la stratégie définie par les dirigeants. Cela implique un processus décisionnel ou cours duquel le décideur a pour objectif de retenir un ensemble de projets, satisfaisants aux contraintes.

Cet ensemble est appelé portefeuille de projets, et sa constitution est une activité aussi déterminante que complexe. La complexité vient du nombre de combinaisons de portefeuilles réalisables dans ce cas, il faut recourir à des outils mathématiques extrêmement performants dans le but de sélectionner une solution optimale en un temps raisonnable.

Mais dans la plupart des problèmes, il ne s'agit pas de maximiser seulement le critère de profit mais plutôt d'optimiser simultanément plusieurs critères et qui sont généralement conflictuels, donc il s'agit d'optimiser simultanément plusieurs fonctions c'est-à-dire : PMO. La notion de solution optimale unique dans l'optimisation uniobjectif disparait pour les problèmes d'optimisation multiobjectif au profit de la notion d'ensemble de solutions Pareto Optimales.

Nous présentons dans cette section, une méthode de résolution heuristique des PMO basée sur les algorithmes des Colonies de Fourmis appelés **MOACO multiobjective Ant Colony Optimization .**

### Historique

La métaheuristique d'optimisation par colonies de fourmis a été initialement introduite par Dorigo, Maniezzo et Colorni [13] et a été inspirée par les études sur le comportement des fourmis réelles effectuées par Deneubourg et al [11].

A l'origine, l'optimisation par colonie de fourmis a été conçue pour résoudre le problème du voyageur de commerce en proposant le premier algorithme ACO : **Ant System (AS)** [13]. Par la suite, un nombre considérable d'applications de ACO a été proposé telles que l'affectation quadratique [23], le routage des véhicules [4] , le problème de satisfaction de contraintes [34], ...

### Analogie biologique

La métaheuristique ACO a été inspirée, essentiellement, par les études sur le comportement des fourmis réelles effectuées par Deneubourg et al [11]. L'un des problèmes étudiés était de comprendre comment des insectes, comme les fourmis, peuvent trouver le chemin le plus court du nid à la source de nourriture et le chemin de retour. Il a été trouvé que le moyen utilisé pour communiquer l'information entre les fourmis qui cherchent des chemins, est le dépôt de traces de phéromone, qui est une substance chimique que les fourmis arrivent à détecter.

En se déplaçant, une fourmi dépose de la phéromone marquant ainsi le chemin par une trace de cette substance. Tandis qu'en absence de traces une fourmi se déplace aléatoirement, une fourmi qui rencontre une trace de phéromone déjà déposée peut la détecter et décider de la suivre avec une probabilité proportionnelle à l'intensité de la trace, et renforce ainsi cette trace avec sa propre phéromone.

Le comportement collectif émerge d'un processus autocatalytique où plus les fourmis suivent une trace, plus cette trace devient attirante : c'est le principe de stigmergie.

Ce processus peut être illustré par l'exemple de la figure ci-après, des fourmis se déplacent dans un réseau qui connecte une source de nourriture (A)à un nid (E). Supposons que les fourmis prennent une unité de temps pour une distance de longueur 1 et qu'à chaque unité de temps, 30

fourmis sortent du nid.
Les premières fourmis qui arrivent à la position D doivent choisir d'aller vers C (avec une longueur totale du chemin de 3) ou vers H (avec une longueur totale du chemin de 4). Supposons qu'à l'instant t=0 , 30 fourmis sont au point D. Comme il n'existe pas de trace de phéromone sur les deux chemins, elles vont choisir une des deux alternatives avec la même probabilité. Donc environ 15 fourmis décident d'aller à C et les 15 autres fourmis décident d'aller à H. A t=1, les 30 prochaines fourmis sortant de B (et de D) peuvent détecter les traces de phéromone (figure 3.1.c). La trace de phéromone sur BCD est deux fois plus intense que celle sur BHD, car 30 fourmis sont passées par BCD (15 de A et 15 de D), tandis que 15 fourmis seulement sont passées par BHD. C'est pourquoi maintenant 20 fourmis prennent BCD et 10 prennent BHD (de même 20 prennent DCB et 10 prennent DHB à partir du point D). Une fois encore, plus de phéromone est déposée sur le plus court chemin. Ce processus est répété à chaque unité de temps et ainsi les traces de phéromone sont renforcées. Grâce à ce processus autocatalytique, toutes les fourmis vont très rapidement choisir le chemin le plus court.

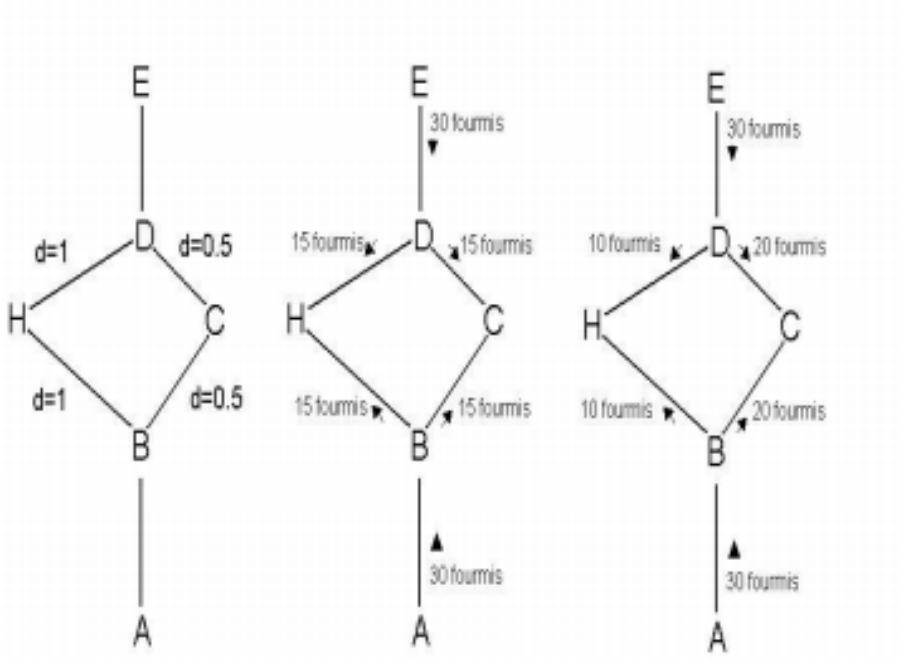



FIGURE 3.1 – Comment les fourmis trouvent le plus court chemin

Pour transposer ce comportement à un algorithme général d'optimisation combinatoire, on fait une analogie entre l'environnement dans lequel les fourmis cherchent de la nourriture et l'ensemble des solutions admissibles du problème (i.e. l'espace de recherche du problème), entre la quantité ou la qualité de la nourriture et la fonction objectif à optimiser et enfin entre les traces et une mémoire adaptative.

Les fourmis artificielles dans les algorithmes ACO se comportent de la même manière. Elles diffèrent des fourmis naturelles dans le fait qu'elles ont une sorte de mémoire, pour assurer la génération de solutions faisables. En plus, elles ont des informations sur leur environnement.

**Algorithme de base : Ant System et problème du voyageur de commerce**

Le problème de voyageur de commerce (Travelling Salesman Problem,TSP) a fait l'objet de la première implémentation d'un algorithme de colonie de fourmis : **le Ant System**[13]. Dans l'algorithme Ant System (AS), chaque fourmi est initialement placée sur une ville choisie aléatoirement, chacune possède une mémoire qui stocke la solution partielle qu'elle a construite auparavant. Initialement, la mémoire contient la ville de départ. Commençant à partir de cette ville, une fourmi se déplace itérativement d'une ville à une autre. Quand elle est à une ville $i$, une fourmi $k$ choisit d'aller à une ville non encore visitée $j$ avec une probabilité donnée par :

$$\begin{cases} p_{ij}^k(t) = \dfrac{[\tau_{ij}(t)]^\alpha [\tau_{ij}]^\beta}{\sum\limits_{y \in N_i^k} [\tau_{iy}(t)]^\alpha [\tau_{ij}]^\beta} & si\ j \in N_i^k \\ 0 & sinon \end{cases}$$

$\tau_{ij}(t)$ : l'intensité de la trace de phéromone dans l'arête $(i, j)$ à l'instant $t$.
$\tau_{ij} = \frac{1}{d_{ij}}$ : information heuristique à priori valable, où $d_{ij}$ est la distance entre la ville $i$ et la ville $j$ ; l'idée étant d'attirer les fourmis vers les villes les plus proches.
$\alpha, \beta$ sont deux paramètres qui déterminent l'influence relative de la trace de phéromone et de l'information heuristique.
$N_i^k$ est le voisinage faisable de la fourmi $k$ c'est à dire l'ensemble des villes non encore visitées par la fourmi $k$.

La construction de solution se termine après que chaque fourmi ait complété un tour. Ensuite, les traces de phéromone sont mises à jour. Dans AS, la mise à jour se fait, d'abord, en réduisant les traces de phéromone avec un facteur constant $\rho$ (c'est l'évaporation de phéromone) et, ensuite, en permettant à chaque fourmi de déposer de la phéromone sur les arêtes qui appartiennent à son tour. Ainsi la formule de mise à jour de phéromone est comme suit :

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \sum_{k=1}^{nbAnts} \Delta\tau_{ij}^k$$

avec $0 \leq \rho \leq 1$ et $nbAnts$ représente le nombre de fourmis.
Le paramètre $\rho$ est ainsi utilisé pour éviter l'accumulation illimitée de phéromone et permet à l'algorithme d'oublier les mauvaises décisions précédemment prises. Sur les arêtes qui n'ont pas été choisis par les fourmis la force associée va décroître rapidement avec le nombre d'itérations.
$\Delta\tau_{ij}^k$ : est la quantité de phéromone que la fourmi $k$ dépose sur l'arête $(i, j)$.

Il est défini par :

$$\begin{cases} \Delta\tau_{ij}^{k} = \frac{Q}{L^k} & si\ (i,j) \in tabou^k \\ \\ 0 & sinon \end{cases}$$

Où : $L^k$ est la longueur du tour généré par la fourmi $k$, $Q$ une constante de l'algorithme et $tabou^k$ est la liste des villes déjà visitées.

Avec cette formule, les arêtes du tour le plus court recevront la plus grande quantité de phéromone. En général, les arêtes qui sont utilisées par plusieurs fourmis et qui appartiennent aux tours les plus courts recevront plus de phéromone et en conséquence seront plus favorisés dans les itérations futures de l'algorithme.

**Extensions de Ant System**

Malgré les résultats encourageants trouvés pour le problème du VC, AS n'était pas compétitif avec les algorithmes de l'état de l'art du PVC. Des recherches sont alors entreprises pour l'étendre et essayer d'améliorer ses performances en contrôlant mieux l'intensification et la diversification de la recherche.

**Rank-based Ant system (AS-rank)**

C'est une extension proposée par Bullnheimer, Hartl, and Strauss [Bullnheimer et al. , 1999]. Cet algorithme trie les fourmis selon les longueurs de tours générés.

Après chaque phase de construction de tours, seulement les $w$ meilleures fourmis sont autorisées à la mise à jour de phéromone. Ainsi, la $r^{ime}$ fourmi contribue à la mise à jour de phéromone avec un poids donné par le $\max\{0, w-r\}$ tandis que la fourmi correspondant au meilleur tour global renforce la trace de phéromone avec un poids $w$. L'objectif de cette version est d'intensifier plus la recherche vers les meilleures solutions.

**Ant Colony System**

L'algorithme **Ant Colony System** (ACS) [Dorigo & Gambardella, 1997] a été introduit par Dorigo et Gambardella pour améliorer les performances du premier algorithme sur des problèmes de grandes tailles.

ACS introduit une règle qui dépend d'un paramètre $q_0 (0 \leq q_0 \leq 1)$, qui définit une balance diversification/intensification. Cette règle permet aux fourmis de favoriser plus, lors de leurs déplacements, l'arête qui contient le maximum de traces suivant $q_0$ , sinon la règle de transition habituelle est utilisée. Ainsi ACS permet d'intensifier plus la recherche vers les zones les plus prometteuses, i.e., qui contiennent plus de traces.

### MAX - MIN Ant Sytem (MMAS)

Stützle et Hoos ont introduit l'algorithme MMAS [Stützle & Hoos, 2000] qui, pour améliorer les performances de ACO, combine une exploitation améliorée des meilleures solutions trouvées durant la recherche avec un mécanisme efficace pour éviter une stagnation prématurée de la recherche. MMAS diffère en trois aspects clé de AS :

- Pour exploiter les meilleures solutions trouvées durant une itération ou durant l'exécution de l'algorithme, après chaque itération, une seule fourmi ajoute de la phéromone. Cette fourmi peut être celle qui a trouvé la meilleure solution depuis le début de l'exécution, ou bien celle durant le dernier cycle
- Pour éviter une stagnation de la recherche, les valeurs possibles de la trace de phéromone sur chaque composant de solution sont limitées à l'intervalle $[\tau_{min}, \tau_{max}]$
- De plus, les traces de phéromone sont initialisées à $\tau_{max}$ pour assurer une exploration élevée de l'espace des solutions au début de l'algorithme.

### Contrôle de l'intensification et de la diversification

Un point critique lors de l'application de l'algorithme ACO, est de trouver un équilibre entre l'exploitation de l'expérience de recherche acquise par les fourmis et l'exploration de nouvelles zones de l'espace de recherche non encore visitées. ACO possède une bonne manière pour accomplir une telle balance, essentiellement par la gestion des traces de phéromone. En effet, les traces de phéromone déterminent dans quelles parties de l'espace de recherche les solutions construites auparavant sont localisées.

Une manière de mettre à jour la phéromone, et pour exploiter l'expérience de recherche acquise par les fourmis, est de déposer une quantité de phéromone fonction de la qualité de la solution trouvée par chaque fourmi. Ainsi, les zones correspondant aux meilleures solutions recevront une quantité de phéromone plus élevée et seront plus sollicitées par les fourmis durant leurs déplacements.

Pour contrôler la balance entre l'exploitation de l'espace de recherche et l'exploration de nouvelles zones, ACO dispose des paramètres $\alpha$ et $\rho$ pour gérer l'importance relatives des traces de phéromone.

$\alpha$ détermine le poids des traces de phéromone dans le calcul des probabilités de transition, et $\rho$ détermine l'évaporation de phéromone. Ces deux paramètres ont une influence sur le comportement exploratoire ; pour favoriser la stratégie d'exploration, on peut diminuer le coefficient $\alpha$ , ainsi les fourmis deviennent moins sensibles aux phéromones lors de leurs déplacements et peuvent visiter des zones non explorées, ou diminuer $\rho$, ainsi la phéromone s'évapore plus lentement et l'intensité des traces de phéromone devient similaire sur les arêtes et donc les fourmis deviennent moins influencées par la phéromone.

Pour favoriser la stratégie d'exploitation, on augmente les valeurs respectives de $\alpha$ et/ou de

ρ, ainsi les fourmis seront plus guidées par la phéromone vers des zones prometteuses de l'espace de recherche. Après un certain nombre d'itérations, toutes les fourmis vont converger vers un ensemble de bonnes solutions. Ainsi, on favorise une convergence plus rapide de l'algorithme.

Toutefois, il faut faire attention et éviter une intensification trop forte dans les zones qui apparaissent prometteuses de l'espace de recherche car cela peut causer une situation de stagnation : une situation dans laquelle toutes les fourmis génèrent la même solution.

Pour éviter une situation pareille de stagnation, une autre solution serait de maintenir un niveau raisonnable d'exploration de l'espace de recherche. Par exemple, dans ACS les fourmis utilisent une règle de mise à jour locale de phéromone durant la construction de solution pour rendre le chemin qu'elles viennent de prendre moins désirable pour les futures fourmis et ainsi, diversifier la recherche. MMAS introduit une limite inférieure pour l'intensité des traces de phéromone pour garantir toujours la présence d'un niveau minimal d'exploration. MMAS utilise aussi une réinitialisation des traces de phéromone, qui est une manière d'enforcer l'exploration de l'espace de recherche.

## 3.3 Pareto Ant Colony Optimization

**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 $\Gamma$ fourmis sachant que chaque fourmi a un portefeuille vide $x = (0,\ldots,0)$ qu'elle doit remplir au fur et mesure. La durée de vie $\Xi$ est déterminée aléatoirement dans l'intervalle $[1,\ldots,N]$ pour chaque fourmi.

A chaque cycle de l'algorithme, chaque fourmi calcule un ensemble de poids $P = (P_1, P_2, \ldots, P_K)$ 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 $\eta_i$ et sur la structure phéromone $\tau_i$. 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 $K$.

L'algorithme de Pareto Ant Colony Optimization est donné comme suit [12] :

**Procedure P-ACO** {

.
**initialisation** de P-ACO ;/*créer $\Gamma$ fourmis,
initialiser le vecteur de la pheromone avec $\tau_0$ */
**répéter** jusqu'a ce que le critère d'arrêt soit vérifié {
**pour** fourmi= 1 **à** $\Gamma$ {
déterminer la durée de vie $\Xi$ pour chaque fourmi aléatoirement dans [1,...,N] ; fixer
$x = (0,\ldots,0)$ ;/* créer un portefeuille vide */
déterminer pour chaque objectif un poids $p_k$ aléatoirement ;
$\xi = \Xi$ ;/* indique le nombre de projets à sélectionner */
**tant que** $\xi \geq 0$ **et** $\exists\, i$ avec $\eta_i(x) > 0$ {
sélectionner un projet $i$ avec la formule (3.1) ensuite le rajouter à $x$ ;
mise à jour locale des pistes de phéromone ;
décrémenter $\xi$ ;
}
tester la faisabilité du portefeuille $x$ ;
**si** le portefeuille est faisable **alors** tester l'efficacité de $x$ ;
**si** le portefeuille $x$ est efficace **alors** il est enregistré et autres sont éliminés ;
} } }
pour chaque objectif $k$ {
déterminer la meilleure et deuxième meilleure solution
pour chaque objectif $k$
mise à jour globale des traces des phéromone en utilisant
uniquement la première et la deuxième meilleure solution : formule (3.2)
} } }

### 3.3.1 La règle de décision

Pour chaque objectif $k$, les structures phéromones sont stockées dans un vecteur de dimension $N$ et pour chaque projet candidat, une valeur agrégée de l'attractivité est calculée.

Soit $\eta_i$ l'attractivité, $\tau_i^k$ la structure phéromone et $\Omega(x)$ l'ensemble de tous les portefeuilles faisables tel que :

$$\Omega(x) = \{i \in N, \eta_i(x) > 0,\ x_i = 0\}$$

Les fourmis artificielles sélectionnent un portefeuille faisable $i \in \Omega(x)$ afin qu'il soit rajouté a l'ensemble des solutions et ce selon la règle suivante :

$$i = \begin{cases} \arg\max\limits_{i \in \Omega(x)} \{ [ \sum\limits_{k=1}^{K} (p_k \tau_i^k) ]^{\alpha} [\eta_i(x)]^{\beta} \} & if\ q \leq q_0 \\ \\ j & ailleurs \end{cases}$$

Où $q$ : variable aléatoire qui suit une loi uniforme sur l'intervalle $[0,1]$. $q_0$ : Un paramètre compris

entre 0 et 1 représente la probabilité que la solution choisie, a les plus grandes valeurs agrégées de l'attractivité et la phéromone. Quant a la variable aléatoire $j$ elle est sélectionnée selon la distribution de probabilité suivante :

$$p_i(x) = \frac{\left[\sum_{k=1}^{K}(p_k\tau_i^k)\right]^{\alpha}\ [\,\eta_i(x)]^{\beta}}{\sum_{h\in\Omega(x)}\left[\left[\sum_{k=1}^{K}(p_k\tau_h^k)\right]^{\alpha}\left[\eta_h(x)^{\beta}\right]\right]} \qquad (3.1)$$

On remarque cette expression est donnée par les $\alpha$ et $\beta$ qui représentent respectivement le poids des traces de phéromone et la visibilité.

### 3.3.2 La mise à jour de phéromone

Une mise à jour locale de phéromone est effectuée une fois une fourmi artificielle ajoute un projet à un portefeuille. Quand une fourmi choisit un projet $i$, la quantité de phéromone sur l'élément $\tau_i^k$ du vecteur phéromone diminue pour chaque $k$ objectif. La règle locale mise à jour de phéromone pour ces éléments est donnée comme suit :

$$\tau_i^k = (1-\rho)\tau_i^k + \rho\tau_0$$

Où $\tau_0$ est la valeur initiale des traces des phéromones et $\rho$ le taux d'évaporation.

Après la construction des solutions par chaque fourmi, La mise à jour globale de phéromone est effectuée et les tests de faisabilité et d'efficacité sont déterminés.

La mise à jour globale de phéromone se fait en utilisant deux fourmis, la meilleure et la deuxième-meilleure solution générée dans le cycle courant pour chaque objectif $k$ comme suit :

$$\tau_i^k = (1-\rho)\ \tau_i^k + \rho\ \Delta\tau_{ij}^k \qquad (3.3)$$

Les tests sur la stratégie de la mise à jour de la phéromone, ont montré que l'utilisation de la meilleure et deuxième meilleure fourmi avec des quantités de phéromone de 10 pour la meilleure et 5 pour la deuxième, donne de bons résultats.

$$\Delta\tau_{ij}^k = \begin{cases} 10 & if\ x_{meil\ sol,i} = 1 \\ 0 & sinon \end{cases}$$

Une fois cette mise à jour est effectuée, tout en utilisant la meilleure solution, une autre mise à jour similaire à la première sera effectuée mais cette fois en utilisant la deuxième meilleure solution et non pas la première et donnée comme suit :

$$\Delta\tau_{ij}^{k} = \begin{cases} 5 & if\ x_{deux\ meil\ sol,i} = 1 \\ 0 & sinon \end{cases}$$

### 3.3.3 L'influence des paramètres $\alpha$ et $\beta$ sur la résolution

Quand on résout un problème d'optimisation combinatoire avec une approche heuristique, il s'agit de trouver un bon compromis entre deux objectifs relativement duaux : d'une part il s'agit d'intensifier la recherche autour des zones de l'espace de recherche les plus prometteuses, qui sont généralement proches des meilleures solutions trouvées ; d'autre part il s'agit de diversifier la recherche et favoriser l'exploration afin de découvrir de nouvelles et si possible meilleures zones de l'espace de recherche. Le comportement des fourmis par rapport à cette dualité entre intensification et diversification peut être influencé en modifiant les valeurs des paramètres.

En particulier, la diversification peut être augmentée soit en diminuant la valeur du poids du facteur phéromone $\alpha$ (de sorte que les fourmis deviennent moins sensibles aux traces phéromonales), soit en diminuant le taux d'évaporation $\rho$ (de sorte que la phéromone s'évapore plus doucement et les écarts d'une trace à l'autre évoluent plus doucement). Lorsque l'on augmente ainsi la capacité exploratoire des fourmis, on trouve généralement de meilleures solutions, mais en contrepartie ces solutions sont plus longues à trouver.

#### Conclusion

Dans ce chapitre, nous avons fait une étude récapitulative est menée sur les méthodes de résolution des problèmes d'optimisation multiobjectif, tout en montrant la manière de définir un problème pareil, ses contraintes, ses objectifs, tout en respectant le concept de compromis et les frontières de Pareto. Nous avons parlé de méthodes utilisées dans le domaine de l'optimisation multiobjectif à savoir exactes et métaheuristiques.

# 4

# Modèles d'optimisation de portefeuille

## Sommaire

# Introduction

Un problème permanent dans la finance est : comment combiner les investissements pour former un portefeuille ? Répondre à cette question, serait faire de la **Sélection de Portefeuille**. En 1952, Harry Markowitz [Markowitz h.m, 1952] a développé sa formulation de la moyenne-variance, puis en 1956, il a introduit un algorithme pour trouver **la Frontière Efficiente**, et s'est basé sur la sélection d'un portefeuille optimal sur la frontière efficiente.

La gestion de portefeuille moyenne-variance fait l'hypothèse que les agents sont rationnels et ont de l'aversion pour le risque. Pour comparer et sélectionner les titres, ils mettent en balance l'intérêt que ces titres procurent avec le risque qu'ils font subir. L'agent essaie d'obtenir un rendement maximum pour un risque minimum. En effet, il est logique qu'un investisseur sacrifie un peu de rentabilité pour diminuer le risque de son portefeuille.
Cependant, plusieurs mesures de risque peuvent être considérées par les modèles d'optimisation de portefeuille, ces types seront détaillés dans les sections qui suivent.

## 4.1 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.[lexilogos, dictionnaire français].

***def 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 (DUSAULX, Voy. Barège, t. 1, 1796, p. 223).*

***def 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 DEBATISSE, Révol. silenc., 1963, p. 166).*

***def 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 ].*

## 4.2 La Variance et la théorie d'Harry Markowitz

### 4.2.1 Le modèle de 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.

#### 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.

#### Notations

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 $V_i(0)$. Chaque actif est caractérisé par deux variables aléatoires $V_i(t)$ et $R_i(t)$ définies sur l'ensemble $\Omega(t)$ des mondes possibles la date $t$ telles que :

– $V_i(t)$ est la valeur du $i^{ime}$ actif à l'instant $t$.
– $R_i(t)$ est la rentabilité du $i^{ime}$ actif à l'instant $t$

avec :

$$V_i(t) = (1 + R_i(t)V_i(0))$$

et donc :

$$R_i(t) = -1 + \frac{V_i(t)}{V_i(0)}$$

**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.

### 4.2.2 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.

### 4.2.3 Estimation des rendements et des risques.

Soit $R_p$ le rendement total d'un portefeuille composé de $n$ actifs financiers caractérisés par leurs rendements $R_i$ et leurs variables binaires associées $x_i$, $i = 1, \ldots, n$ tel que :

$$x_i = \begin{cases} 1 & \text{si l'actif } i \text{ est inclue dans le portefeuille} \\ 0 & \text{sinon} \end{cases}$$

alors

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

avec

$$x_i \in \{0, 1\}$$

dés lors :

$$E(R_p) = E\left(\sum_{i=1}^{n} x_i R_i\right) = \sum_{i=1}^{n} x_i E(R_i)$$

$$Var(R_p) = \sum_{i=1}^{n} \sum_{j=1}^{n} x_i x_j \, Cov(R_i, R_j)$$

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.

**Méthode historique : le Bootstrap**

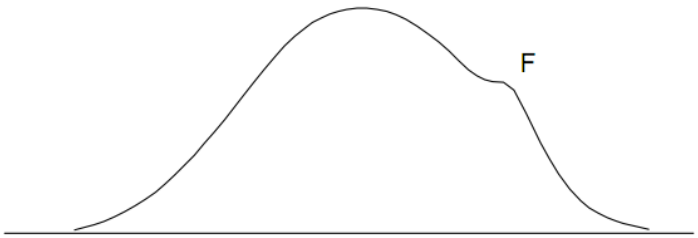
Cette méthode est largement utilisée aujourd'hui dans toutes les institutions de gestion. Elle consiste à utiliser les rendements et les risques historiques. En effet, soit $P$ une population (ensemble d'actifs) de taille $N$ dont on veut estimer un paramètre, par exemple une moyenne $\mu$. Pour cela, on tire un échantillon aléatoire composé de $n$ actifs de cette population et on calcul le rendement et le risque (la caractéristique souhaitée) de chaque actif de l'échantillon. Nous notons $x_i (i = 1, \cdots, n)$ la valeur de cette caractéristique sur l'actif $i$. Une estimation classique de $\mu$ est la moyenne arithmétique

$$\hat{\mu} = \frac{1}{n} \sum_{i=1}^{n} x_i.$$

La question générale qui va nous amener au bootstrap est la suivante : est-ce que cette estimation est bonne ? Une manière d'y répondre est de calculer la variance $\sigma^2(\hat{\mu})$ de l'estimateur $\hat{\mu}$ vu comme variable aléatoire.

Pour calculer la variance de l'estimateur $\hat{\mu}$, il faut étudier sa distribution d'échantillonnage, dont une approximation pourrait se construire de la manière suivante :

– Les individus de $P$, étudiés selon la caractéristique précitée, suivent une certaine distribution $F$




– De cette population, on tire un nombre $k$ d'échantillons de taille $n$ et on calcule alors $k$ estimations $\hat{\mu}_1, \hat{\mu}_2, \cdots, \hat{\mu}_k$ de la moyenne inconnue $\mu$.
– La distribution empirique de ces estimations $\hat{\mu}_1, \hat{\mu}_2, \cdots, \hat{\mu}_k$ est une approximation de la distribution d'échantillonnage de l'estimateur $\hat{\mu}$.

Une estimation de la variance cherchée $\sigma^2(\hat{\mu})$ serait alors :

$$\sigma^2(\hat{\mu}) = \frac{1}{k-1}\sum_{i=1}^{k}(\hat{\mu}_i - \bar{\hat{\mu}})^2$$

où $\bar{\hat{\mu}} = \frac{1}{k}\sum_{i=1}^{k}\hat{\mu}_i$ est la moyenne des $k$ éstimations de $\mu$ trouvées auparavant.

Notons que pour avoir la véritable variance $\sigma^2(\hat{\mu})$, il faudrait avoir tous les échantillons possibles de $n$ éléments pris dans la population $P$, ce qui serait un grand travail ! On aurait alors $\bar{\hat{\mu}} = \mu$ et $\sigma^2(\hat{\mu}) = \frac{1}{k}\sum_{i=1}^{k}(\hat{\mu}_i - \mu)^2$ où $k = C_N^n$ est le nombre d'échantillons de $P$.
Mais le procédé décrit ci-dessus est utopique ; en effet, il exige d'avoir plusieurs échantillons alors que la réalité ne nous en donne qu'un seul. Le bootstrap permet de remédier á ce problème grâce à la simulation d'échantillons.

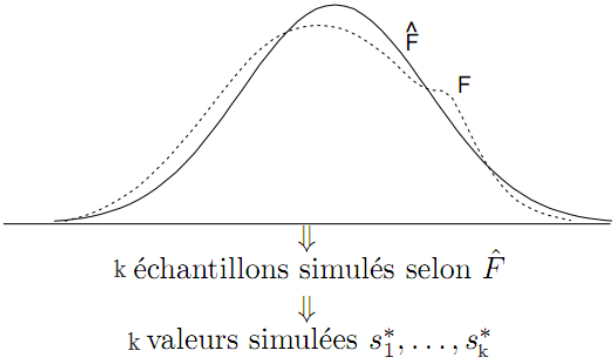
**Le principe du bootstrap**

L'estimateur considéré (plus haut $\hat{\mu}$), ou plus généralement la statistique $S$ d'intérêt, a une distribution d'échantillonnage notée $F_S$ . Cette distribution dépend de la distribution $F$ de la variable aléatoire $X$ dont les valeurs observées sont $x_1, \cdots, x_n$.On écrit $F_S\ (s, F_X)$ où $F_X$(ici $F$) est la distribution de $X$. Comme $F$ est inconnue,on travaille avec une estimation de $F$, que l'on note $\hat{F}$ et qui est :

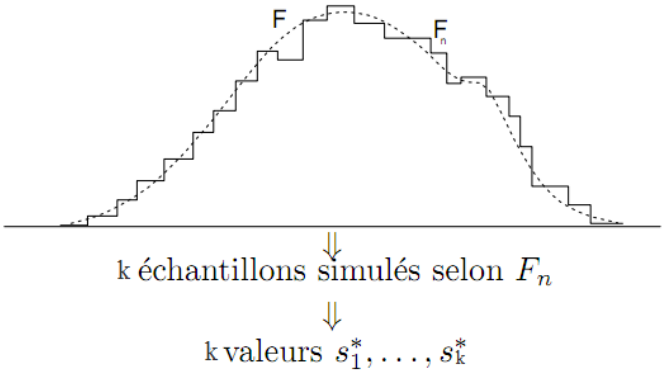
- soit un modèle paramétrique connu (normal, exponentiel, ou autre) qui ajuste assez bien les données.
- soit la distribution empirique $F_n$ des données.

Le fait de remplacer $F$ par l'un des $\hat{F}$ vus ci-dessus va donner une distribution d'échantillonnage $F_S$ également modifiée puisque $F_S$ dépend de $F$. On écrit dans ce cas $F_S\ (s, \hat{F})$ au lieu de $F_S(s, F)$.

Il y a deux formes de bootstrap : le boostrap **paramétrique** et le bootstrap **non paramétrique**. Dans le bootstrap paramétrique, on remplace d'abord $F$ par un modèle paramétrique $\hat{F}$ qui semble bien ajuster les données. On simule ensuite $k$ échantillons ($k = 1000$ par exemple) de taille $n$ , indépendamment les uns des autres, qui proviennent de la distribution $\hat{F}$. Enfin, on calcule la statistique $S$ (moyenne, médiane, ...) pour chacun des $k$ échantillons simulés et l'on obtient les valeurs simulées $s_1^*, \cdots, s_k^*$




Si nous ne pouvons pas attribuer un modèle paramétrique aux données, nous utilisons $\hat{F} = F_n$ comme approximation de $F$ , où $F_n$ est la distribution empirique ; c'est le bootstrap non-paramétrique. On génère alors $k$ échantillons de taille $n$ provenant de $F_n$. De nouveau, on calcule les $k$ valeurs $s_1^*, \cdots, s_k^*$ de $S$ pour ces $k$ échantillons simulées.




Dans les deux cas, la distribution empirique des valeurs simulées $s_1^*, \cdots, s_k^*$ fournit une approximation de la distribution d'échantillonnage de $S$. On appelle cette approximation la distribution bootstrap de $S$. A l'aide de cette distribution on peut alors calculer, par exemple, une approximation de $\sigma^2(S)$ :

$$\sigma^2(S)^* = \frac{1}{k-1}\sum_{i=1}^{k}(s_i^* - \bar{s^*})^2$$

où

$$\bar{s^*} = \frac{1}{k}\sum_{i=1}^{k} s_i^*.$$

**Remarque**

Remplacer $F$ par $\hat{F} = F_n$ et générer un échantillon de taille $n$ selon cette distribution $F_n$ revient au même que de tirer avec remise $n$ éléments de l'ensemble de données originales $x_1, \cdots, x_n$.

**Intervalles de confiance bootstrap**

Grâce au bootstrap, on peut déterminer des intervalles de confiance pour le paramètre inconnu $\mu$ . Il y a plusieurs types d'intervalles de confiance utilisant la simulation bootstrap mais un seul d'entre eux, d'ailleurs simple à déterminer.
Précisons qu'il est utilisable si la distribution bootstrap de l'estimateur est approximativement normale. Si ce n'est pas le cas, d'autres types d'intervalles de confiance sont à utiliser (Efron, 1992).
On appelle intervalle percentile $5\% - 5\%$ pour $\mu$ l'intervalle $[\mu^*_{[0.05.k]}, \mu^*_{[0.95.k]}]$ dont les bornes sont simplement les percentiles 5% et 95% de la distribution bootstrap des $\mu^*_i$

On démontre qu'il s'agit d'un intervalle de confiance dans le sens habituel : si on prenait un grand nombre d'échantillons de même taille que l'échantillon original, dans les mêmes conditions, et que l'on construisait pour chacun d'entre eux un intervalle percentile $5\% - 5\%$, le 90% de ces intervalles contiendrait la valeur inconnue $\mu$. Ce 90% est le coefficient de couverture de l'intervalle.

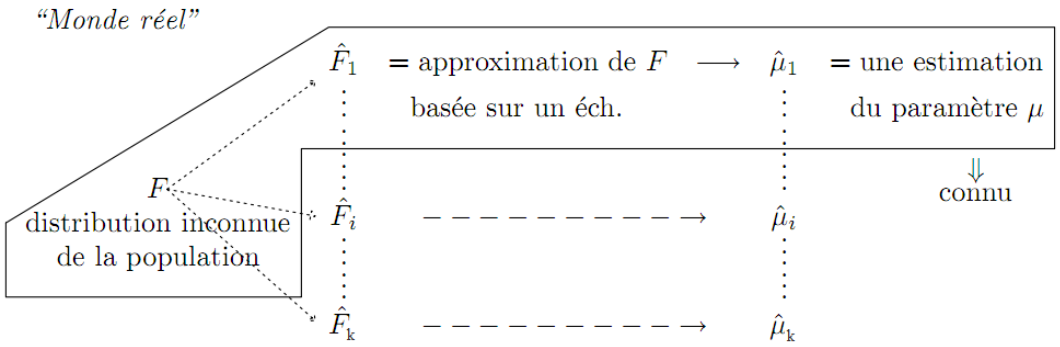
**Remarques**

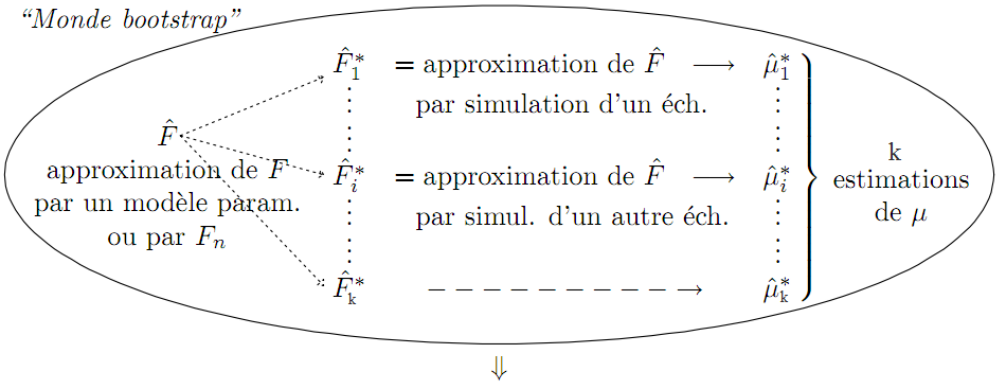
a. On parle souvent de rééchantillonnage dans le cas du bootstrap non paramétrique car on reconstruit un ensemble d'échantillons en partant de l'échantillon de départ.

b. Le bootstrap non paramétrique est approprié lorsqu'il est difficile de trouver un bon modèle pour $F_X$

c. Plus le nombre d'échantillons générés $r$ est grand, meilleures sont les approximations bootstrap, mais plus l'ordinateur prend du temps à calculer aussi. Un $r$ de 100 ou 200 peut suffire à avoir une bonne estimation de $\sigma^2(\hat{\mu})$ mais un $r = 500$ ou 1000 donne une meilleure précision. Pour calculer des intervalles de confiance bootstrap, on choisit en général $1000 \leq r \leq 5000$).

d. Nous présentons ci-dessous un schéma mettant en valeur le principal avantage du bootstrap :




Ici, tout est connu et on peut générer tous les échantillons nécessaires.

### 4.2.4 Appréciation

L'étude de Markowitz reste ce pendant très théorique, car elle prend l'hypothèse que l'on connait le couple espérance de gain et risque de chaque titre. Information qui ne se trouve évidemment pas telle quelle dans la presse économique et qui fait une large part à l'appréciation du gestionnaire. De plus le modèle M-V ne permet pas de sélectionner le portefeuille efficient propre à un investisseur particulier.

Même si les limites de l'approche moyenne - variance sont bien connues, il n'en demeure pas moins vrai que cette analyse est fondamentale pour au moins deux raisons. Pour les praticiens, cette théorie démontre que les portefeuilles efficients au sens moyenne -variance jouent un rôle important dans la gestion des portefeuilles.

Pour les chercheurs, l'analyse M-V est le noyau de plusieurs théories d'évaluation d'actifs et de modèles mathématiques de sélection d'un portefeuille efficient. En effet, quelques algo-

rithmes comme ceux proposés par Sharp [Sharp,1963-1967], Newrocki et Carter [Newrocki & Carter1998], Shing et Nagasawa [Shing & et Nagasawa,1999], ont été mis en œuvre afin d'améliorer le calcule d'efficacité du modèle de Markowitz.

## 4.3 La valeur en risque : *VaR* (Value at Risk)

Le risque est lié à la volatilité du portefeuille d'actifs. Pendant très longtemps, la mesure naturelle du risque a donc été la volatilité. Par exemple, dans le modèle de sélection de portefeuille de Markowitz, l'agent maximise son espérance de gain pour un niveau de risque donné, qui est mesuré par l'écart-type.

Cette vision du risque n'est cohérente que dans un monde gaussien. Cependant, nous savons depuis fort longtemps que l'hypothèse de normalité des rendements des actifs financiers n'est pas vérifiée. Actuellement, la mesure de risque qui est la plus répandue est la valeur en risque (**Value-at-Risk ou** *VaR*).

### 4.3.1 Présentation de *VaR*

Pour expliquer ce qu'est la *VaR*, commençons par prendre un exemple concret. Considérons que nous avons investis $10000EUR$ dans notre portefeuille d'actions. Comment avoir une idée de la perte maximale que le portefeuille peut subir d'ici un mois ?
La réponse la plus logique est que nous pouvons perdre tout notre investissement. Néanmoins, un événement de perte totale est vraiment très peu probable. Une réponse plus réaliste serait par exemple qu'en l'absence d'événement exceptionnel, il y a 5% de chance de perdre $1000EUR$. C'est le type de réponse fournit par la *VaR*. De nombreuses définitions de la *VaR* existent, nous en reprenons deux :

***def 1***

*Selon Calvet [ Calvet, 2000], la VaR d'un portefeuille d'actifs financiers correspond au montant de pertes maximum sur un horizon de temps donné, si l'on exclut un ensemble d'événements défavorables ayant une faible probabilité de se produire.* [25]

***def 2***

*Selon Esch, Kieffer et Lopez [Selon Esch, Kieffer & Lopez , 1997] ainsi que Jorion [Jorion, 2000], la VaR d'un portefeuille ou d'un actif, pour une durée T et un niveau de probabilité p, se définit comme le montant de perte attendu de façon que ce montant, pendant la période* $[0,T]$*, ne devrait pas être plus important que la VaR et ceci avec une probabilité de* $(1-p)$.

**Particularités**

Deux éléments sont communs à toutes ces définitions et il convient de les choisir judicieusement : l'horizon et le niveau de confiance.

L'horizon est influencé par trois facteurs majeurs :
– il doit être adapté à la durée de détention de l'actif ou du portefeuille objet de l'estimation.
– il doit être suffisamment court pour respecter l'hypothèse d'invariance de la composition du portefeuille.
– il doit être suffisamment court pour que la quantité de données disponible (parfois faible) puisse permettre d'estimer une VaR sur cet horizon.
Mais il peut aussi être fixé par des normes réglementaires. Les organismes financiers utilisent une *VaR* à 10 jours mais cet horizon peut atteindre plusieurs mois lorsqu'il s'agit d'obligations.

Le niveau de confiance est quant à lui influencé par deux facteurs :
– il ne doit pas être trop élevé, sinon le risque de réalisation devient suffisamment faible pour être inintéressant en tant qu'indicateur.
– il doit refléter le degré d'aversion des gestionnaires face au risque de réalisation d'événements extrêmes.
Mais tout comme pour l'horizon, le niveau de confiance peut être déterminé par des normes réglementaires. Dans la pratique, la *VaR* est estimée sur la base de niveau de confiance allant de 90 à 99.9%.

Les normes réglementaires imposées par Bâle II exigent que la *VaR* soit calculée par les banques en utilisant un niveau de confiance de 99% et un horizon de 2 semaines soit 10 jours ouvrables.

### 4.3.2 Approche classique de calcul

Après avoir choisi un horizon et un niveau de confiance, il reste une dernière étape à réaliser avant de procéder à l'estimation de la *VaR*. Il faut en effet récupérer les données sur lesquels sera calculée la *VaR* et les retraiter. On peut calculer la VaR à partir des rendements du portefeuille mais aussi à partir de sa distribution de Profit & Loss (P&L) qui correspond aux pertes ou profits journaliers. [25]

Considérons un portefeuille composé d'une ou plusieurs actions dont on connait les cours journaliers. On note $W_t$ la valeur de ce portefeuille à la date $t$. La *VaR* de ce portefeuille pour un horizon d'un jour avec une probabilité $\alpha$ correspond à la perte $\Delta W_{(t+1)} = W_{(t+1)} - W_t$ observée pour le portefeuille avec une probabilité $1-\alpha$ Autrement dit, $VaR_t(\alpha)$ vérifie l'équation :

$$p\left[\Delta w_{t+1} + VaR_t(\alpha) < 0\right] = 1-\alpha$$

$$p\left[\Delta w_{t+1} < -VaR_t(\alpha)\right] = 1-\alpha$$

$$p\left[\Delta w_{t+1} < VaR_t(\alpha)\right] = \alpha$$

En introduisant $F_t$ la fonction de répartition de la variable aléatoire $\Delta W$ pour les valeurs connues à la date $t$ et sa fonction inverse $F_t^{-1}$.
On exprime alors la *VaR* de la façon suivante :

$$VaR_t(\alpha) = F_t^{-1}(\alpha)$$

En général, on essayera de se ramener à une loi de distribution standard qui pourra être assimilée à la loi des rendements. Pour cela on réduit et on centre $\Delta W_{t+1}$ (théorème central limite). Ceci nous donne :

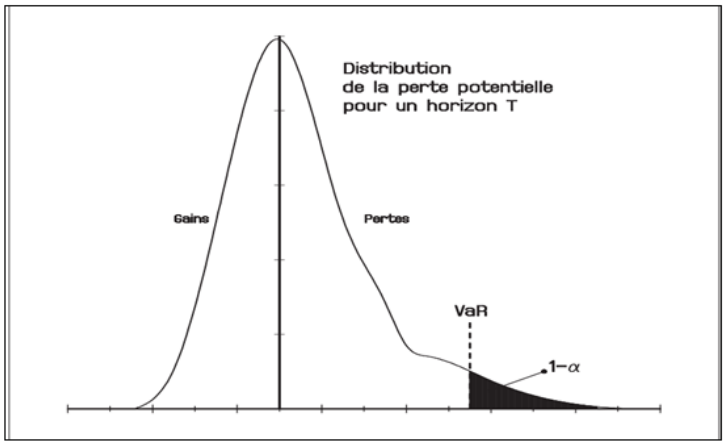
$$\left[\frac{\Delta W_{t+1} - E(\Delta W)}{\sigma(\Delta W)} \leq \frac{VaR_t(\alpha) - E(\Delta W)}{\sigma(\Delta W)}\right] = \alpha$$

On posant : $z_\alpha = \frac{Var_t(\alpha) - E(\Delta W)}{\sigma \Delta W} = \alpha$ où $z_\alpha$ où représente le quantile d'ordre $\alpha$ de la distribution Normale standard, on exprime alors la VaR de la façon suivante :

$$VaR(\alpha) = z_\alpha \, \sigma(\Delta W) + E(\Delta W)$$

La plus part du temps, la loi de distribution des rendements est proche d'une loi Normale. Les tables statistiques donnent des valeurs de $z_\alpha$ pour différentes valeurs de $\alpha$ :

Voici un exemple représentant le positionnement de $z_\alpha$ pour une loi Normale avec $\alpha = 95\%$ :




En réalisant ces calculs sur une distribution journalière des rendements, on exprime une *VaR* à horizon un jour. Pour passer d'un horizon d'un jour à un horizon de $X$ jours, on utilisera la formule suivante :

$$VaR_X = VaR_1 \sqrt{X}$$

### 4.3.3 Le problème des distributions non normales

Dans la réalité, la distribution des rendements d'un portefeuille suit rarement l'hypothèse simplificatrice que les rendements sont normaux. Pour caractériser une loi Normale, seulement deux paramètres sont utilisés : la moyenne et la variance qui représentent les moments d'ordre un et deux. Deux autres paramètres ont une grande importance dans la distribution des rendements : le coefficient de Skewness et le coefficient de Kurtosis.

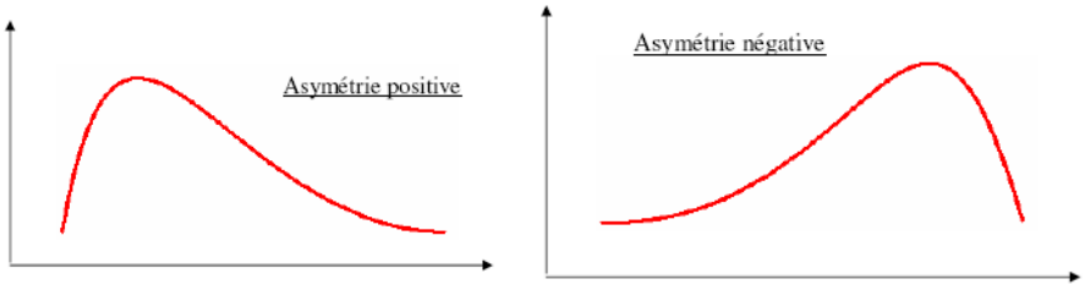
#### Skewness ou coefficient d'assymétrie

Le coefficient de Skewness, que l'on notera $S_k$ caractérise l'asymétrie d'une distribution. Il est associé au moment d'ordre 3 d'une distribution. Voici sa formule de calcul :

$$S_k = \frac{1}{n} \times \sum_{k=1}^{n} \left( \frac{(\Delta W_k - E(\Delta W))^3}{(\sigma(\Delta W)^2)^{\frac{3}{2}}} \right)$$

où n représente le nombre de données disponibles.

On compare le Skewness empirique à celui d'une loi Normale qui vaut 0.
– Si $S_k = 0$ alors la distribution est symétrique et a donc de grandes chances d'être proche d'une loi Normale.
– Si $S_k > 0$ alors la distribution s'étale vers la droite et est dite à asymétrie positive.
– Si $S_k < 0$ alors la distribution s'étale vers la gauche et est dite à asymétrie négative.



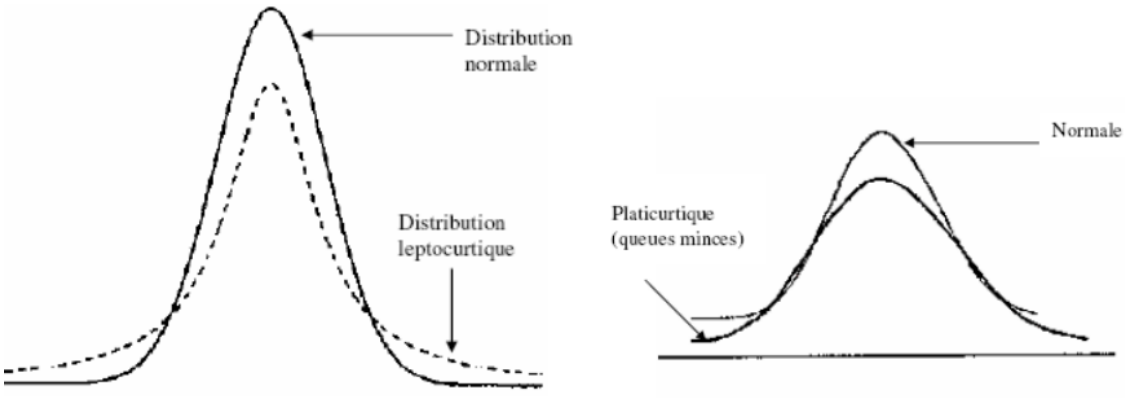

#### Kurtosis ou coefficient d'aplatissement

Le coefficient de Kurtosis, que l'on notera $K$ caractérise l'aplatissement d'une distribution. Il est associé au moment d'ordre 4 d'une distribution. On l'utilise pour les distributions ayant des queues épaisses (fat-tails). Voici sa formule de calcul :

$$K = \frac{1}{n} \times \sum_{k=1}^{n} \left( \frac{(\Delta W_k - E(\Delta W))^4}{(\sigma(\Delta W)^2)^2} \right)$$

où $n$ représente le nombre de données disponibles.

On compare le Kurtosis empirique à celui d'une loi Normale qui vaut 3.
– si $K = 3$ alors la distribution est dite mésocurtique, ses queues sont proches de la loi Normale.
– si $K > 3$ alors la distribution est dite leptocurtique, elle présente des queues épaisses.
– si $K < 3$ alors la distribution est dite platicurtique, elle présente des queues minces.




### 4.3.4 Méthodes de calcul

Nous présentons ici les méthodes de calcul de la *VaR* les plus classiques en ce qui concerne les portefeuilles d'actions. Afin de pouvoir réaliser le calcul de la *VaR*, il faut avoir constitué un historique des cours des actifs constituant le portefeuille et les avoir exprimées dans la même monnaie. Ceci permet de calculer les pertes et profits du portefeuille et de connaître la valeur du portefeuille en fin de période. De plus, on doit s'être fixé un niveau de confiance et un horizon d'observation judicieusement adaptés au contexte d'utilisation de la *VaR* qui sera calculée. [25]

Les méthodes présentées peuvent être utilisées de diverses façon. On peut appliquer directement les estimations sur le portefeuille et ses actifs mais on peut aussi réaliser une décomposition des actifs et du portefeuille selon divers facteurs de risque sur lesquels on appliquera l'une des méthodes citées plus bas.

Notons toutefois que la *VaR* constitue une bonne indication quant à la mesure du risque sous les conditions que le marché fonctionne normalement et de façon stable et qu'il ne subit pas d'évènement très exceptionnel.

#### Méthode historique

La méthode historique est de loin la plus simple à mettre en œuvre. Elle se base sur l'hypothèse que la distribution des rendements du portefeuille dans le futur est la même que dans le passé. Elle a l'avantage de ne pas réaliser d'hypothèse concernant la distribution des rendements

et d'être facile à calculer.

Pour la déterminer, on peut se baser sur la distribution des P&L ou sur la distribution des rendements. Prenons le cas des P&L, on classe toutes les valeurs de la distribution par ordre croissant et on détermine la valeur correspondant à la probabilité que nous nous sommes fixée. Supposons que nous disposons de 500 données, alors dans le cas ou le niveau de confiance est de 99% la $VaR_t\ (99\%)$.

### Méthode paramétrique

Pour utiliser les méthodes paramétriques, on doit faire une hypothèse concernant la distribution des rendements du portefeuille. Généralement, on approxime d'abord leur distribution par un modèle gaussien. Cette hypothèse est certes très réductrice mais elle permet de simplifier les calculs.

Nous allons vous présenter deux méthodes paramétriques de calcul de la *VaR* : l'approche Variance-Covariance et l'approche Risk Metrics de JPMorgan.

### Approche Variance-Covariance

L'approche Variance-Covariance repose sur l'hypothèse'que les rendements du portefeuille et des facteurs de risque suivent des lois Normales et Multi-normales, bien que cette hypothèse soit facilement invalidée empiriquement. La formule utilisée pour calculer cette *VaR* est donnée par la volatilité.

Dans le cas d'un portefeuille d'actions corrélées entre elles on exprimera la volatilité à partir de la matrice de covariance des actifs. La formule de calcul de la *VaR* est la suivante :

$$VaR_t(a,\alpha) = -\acute{a}\mu_t + (\acute{a}\Omega_t a)^{\frac{1}{2}} z_\alpha$$

où :

- $a$ représente le poids des actifs du portefeuille.
- $\mu_t$ représente le vecteur constitué de l'ésperance des rendements des actifs du portefeuille.
- $\Omega_t$ représente la matrice de covariance des actifs du portefeuille.

### Approche RiskMetrics (JPMorgan)

La méthode de RiskMetrics ressemble à la méthode de Variance-Covariance mais la volatilité est calculée différemment. La volatilité est estimée en utilisant ses valeurs passées ainsi que celles des rendements en accordant plus de poids aux valeurs les plus récentes. Ceci permet de pouvoir s'adapter plus facilement aux changements de conditions du marché et de pouvoir mieux tenir compte des évènements extrêmes. [37]

On utilise la formule suivante :

$$VaR_t(\alpha) = E(\Delta W) + z_\alpha \sigma(t)$$

où $\sigma$ est estimée à partir de la formule suivante :

$$[\sigma(t)]^2 = [\sigma(t-1)]^2 \times \lambda + \left[\frac{\Delta W_{t-1}}{W_{t-1}}\right]^2 \times (1-\lambda)$$

avec $\lambda = 0.94$

**Méthode de Monte Carlo**

C'est de loin la méthode la plus difficile à mettre en œuvre car elle nécessite la réalisation d'un nombre conséquent de simulations. On détermine dans un premier temps les lois de distribution des rendements des facteurs de risque décrivant la valeur du portefeuille, ces lois de distributions peuvent entre autre être des modèles stochastiques. On simule ensuite un grand nombre de scénario futurs pour déterminer les trajectoires des facteurs de risque. Les résultats de ces simulations sont ensuite utilisés pour exprimer la distribution des pertes et profits et calculer la *VaR* [25].

**Méthode de bootstrap**

C'est une méthode stochastique alternative. On reconstitue une distribution des pertes et profits du portefeuille en allant piocher aléatoirement avec remise dans l'échantillon historique. On calcule ensuite la *VaR*, selon la méthode désirée, à partir de cette nouvelle distribution.

Maintenant que les bases de la Value at Risk ont été définies, nous allons passer à l'analyse de la sensibilité d'un portefeuille par rapport à la Value at Risk. Le prochain chapitre porte en effet sur la manière de constituer un portefeuille $VaR - efficient$, c'est-à-dire optimal selon le critère de la *VaR*. Il s'agit là d'une méthode originale d'allocation d'actifs.

# 5
# Application sous R

## Sommaire

# Introduction

Afin de mettre en application notre approche d'optimisation **P-ACO**, présentée dans le présent travail, basée sur les algorithmes des Colonies de Fourmis, nous avons sélectionné 15 actifs sur les marchés étrangers (Renault, Ford, Suzuki, Peugeot, Toyota, Bnp Pariba, Careffour, Credit Agricole, Danone, Stmicroelectronics, France Telecom, l'Oreal, Schneider electric, Societe Generale, Technip) dont les cours journaliers, sur une période étalant du 03 Août 2012 au 04 Fevrier 2013, ont été téléchargés sur le site Internet : $http://finance.yahoo.fr$.
et supposés suivre une loi normale. Par la suite, des estimations des rendements et des risques de chacun des actifs, du mois d'Avril, ont été effectuées à l'aide de bootstrap (méthode historique). De ce qui est du risque, cette variable a été modélisée en premier lieu par la variance comme Markowitz l'a défini dans sa théorie (Markowitz.1952) et par la Value at risk en deuxième lieu, dans un second programme.

## 5.1 Description du problème

Un portefeuille financiers peut être décrit comme étant un sous ensemble de $N$ (dans notre cas, $N = 15$ ) actifs financiers proposés et modélisés par le vecteur $x = (x_1, \cdots, x_N)$, où les variables binaires $x_i$ indiquent si l'actif $i$ est inclut dans le portefeuille $(x_i = 1)$ ou pas $(x_i = 0)$. Notre approche d'optimisation **P-ACO** vise à déterminer un portefeuille d'actifs financiers efficient. En d'autres termes, c'est de trouver un ensemble d'actifs financiers qui, en parallèle, maximise le rendement (modélisé par la moyenne des rendements d'actifs) et minimise le risque (modélisé par la variance ou par la Value at risk des rendements) tout en respectant la contrainte budgétaire fixée à l'avance.

Le problème ci-dessus peut être modélisé par le programme suivant :
**Le premier cas** : modéliser le rendement par la moyenne et le risque par la variance (var)

$$(P1)\begin{cases} Optimisez(z) = (E(R_p), Var(R_p)) \\ \sum_{i=1}^{N} p_{it} x_i \leq B_{tot} \\ x \in \{0,1\}, t \in [0,T], i = 1, \cdots, N \end{cases}$$

en remplaçant l'espérance et l'écart-type du portefeuille par leurs formules, le modèle devient :

$$(P1)\begin{cases} Optimisez(z) = \left( \sum_{i=1}^{N} x_i E(R_{it}), \sum_{i,j=1}^{N} x_i x_j Cov(R_{it}, R_{jt}) \right) \\ \sum_{i=1}^{N} p_{it} x_i \leq B_{tot} \\ x \in \{0,1\}, t \in [0,T], i = 1, \cdots, N \end{cases}$$

**Le deuxième cas** : modéliser le rendement par la moyenne et le risque par la value at risk $(VaR)$

$$(P2)\begin{cases} Optimisez(z) = (E(R_p), VaR(R_p)) \\ \sum_{i=1}^{N} p_{it} x_i \leq B_{tot} \\ x \in \{0,1\}, t \in [0,T], i = 1, \cdots, N \end{cases}$$

en remplaçant l'espérance et l'écart-type du portefeuille par leurs formules, le modèle devient :

$$(P2)\begin{cases} Optimisez(z) = \left( \sum_{i=1}^{N} x_i E(R_{it}), (E(\Delta W) * Z_\alpha) + \sigma(\Delta W) \right) \\ \sum_{i=1}^{N} p_{it} x_i \leq B_{tot} \\ x \in \{0,1\}, t \in [0,T], i = 1, \cdots, N \end{cases}$$

où :
$E(R_{it})$ : le rendement espéré de l'actif $i$ à l'instant $t$.
$Cov(R_{it}, R_{jt})$ : le risque dû à la détention des actifs $i, j$ à l'instant $t$.
$p_{it}$ : le prix de l'actif $i$ à l'instant $t$.
$B_{tot}$ : le budget disponible pour l'investissement.
$\Delta W$ : la distribution de la perte du portefeuille.

## 5.2 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.

## 5.3 Généralités sur le Logiciel R

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. [8]

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

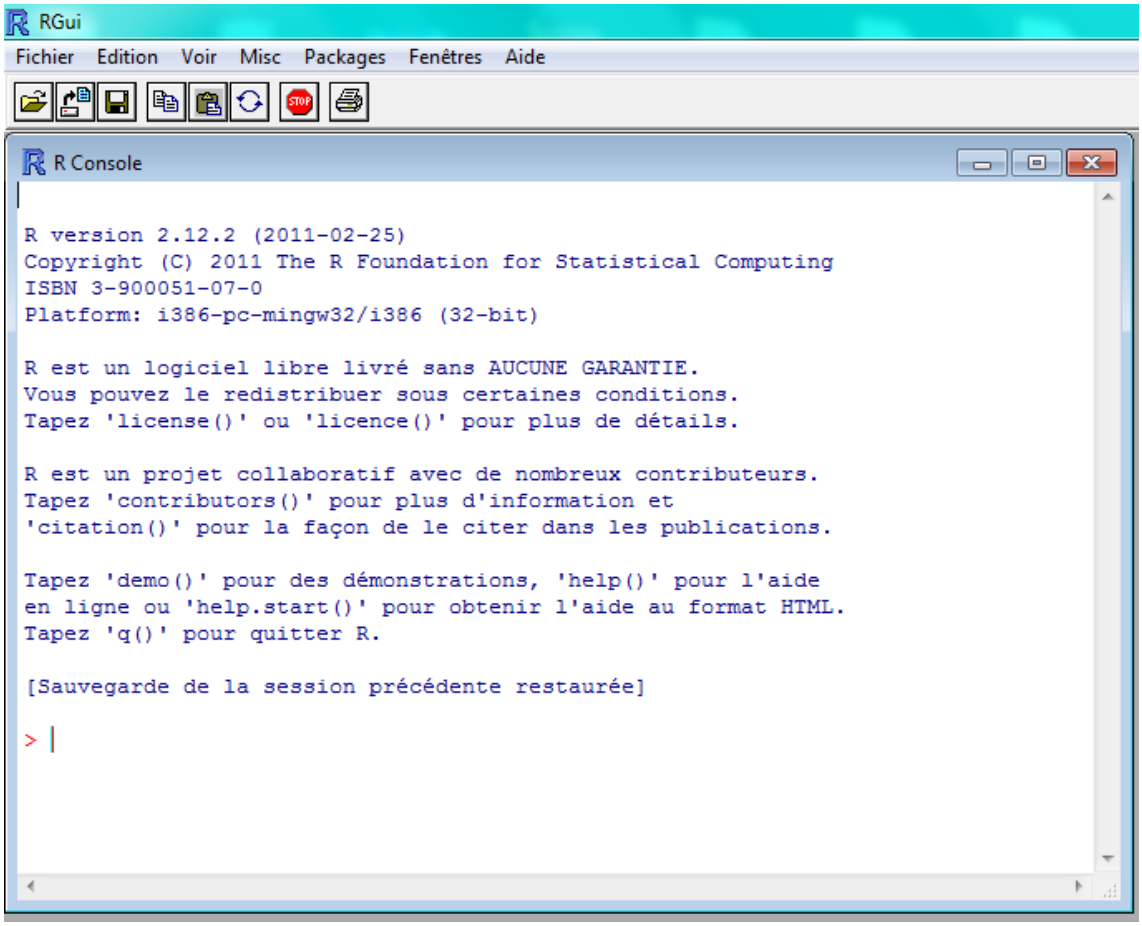



FIGURE 5.1 – Interface du logiciel R

### 5.3.1 Les packages

Toutes les fonctions de R sont stockées dans une bibliothèque. Cette bibliothèque contient des packages de fonctions. Le package nommé **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 5.2 – 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.

### 5.3.2 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"

### 5.3.3 l'aide en ligne : ou help

L'aide en ligne est extrêmement utile dans l'utilisation courante de R. Nous allons illustrer l'utilisation de l'aide de R par un exemple ! : On souhaite faire une régression, deux cas de figure se présentent :

a. On ne connaît pas le nom de la fonction et, a fortiori, on ne sait pas l'utiliser **help.search("regression")**.Cette commande fournit une liste de fonctions reliées d'une manière ou d'une autre au mot **regression**. Voici une copie d'écran du résultat obtenu.

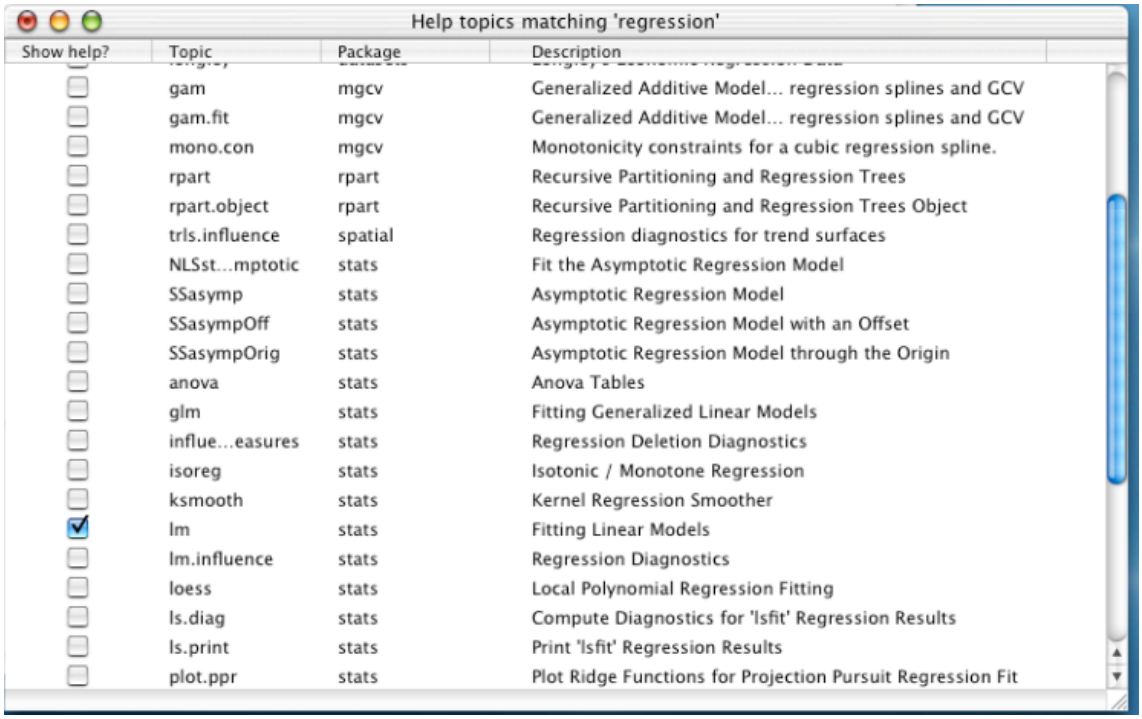



FIGURE 5.3 – Utilisation de l'aide en ligne

On découvre alors grâce au bref descriptif associé à chaque réponse que la fonction *lm*() permet de construire un modèle linéaire (Fitting Linear Models) ; en cliquant sur la fonction associée. On obtient un descriptif détaillé comme l'illustre la capture d'écran suivante :

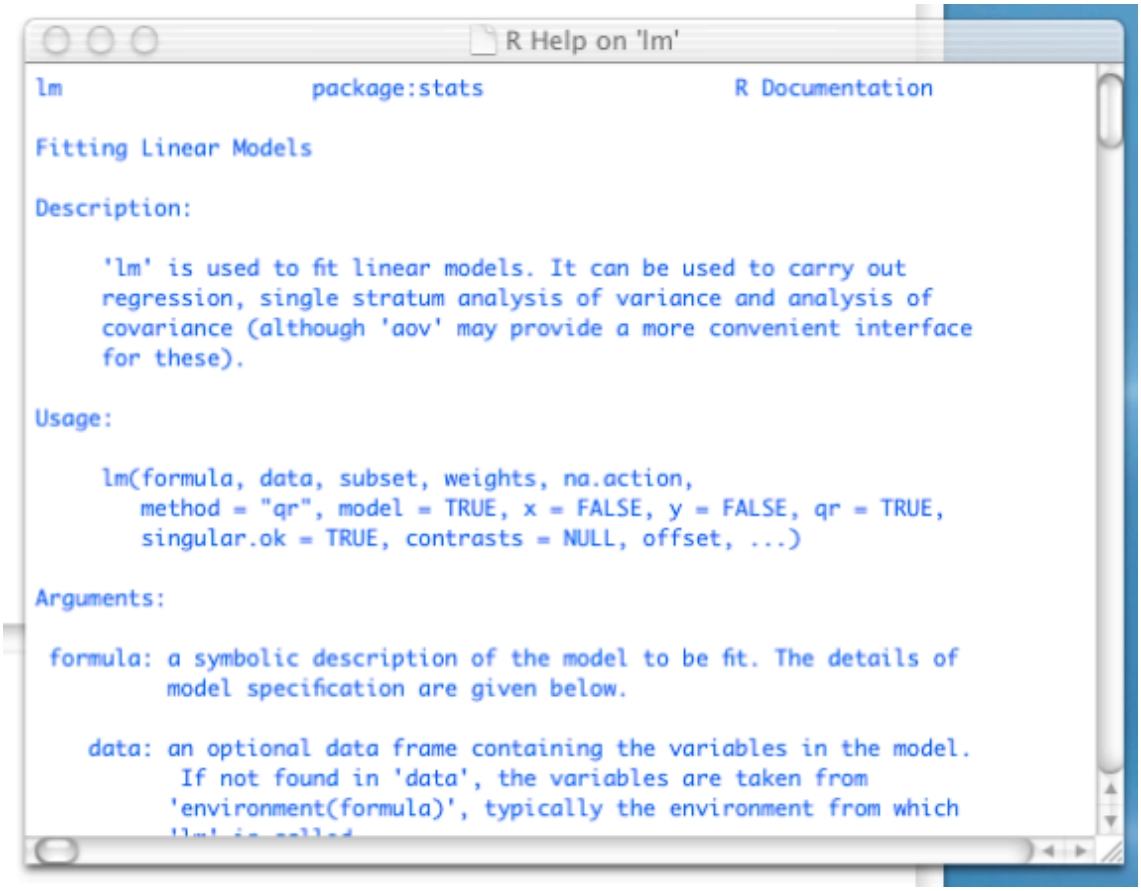



FIGURE 5.4 – Utilisation de l'aide R en ligne

Une manière plus directe d'accéder à la fiche détaillée nécessite la connaissance du nom de la fonction. C'est le deuxième cas de figure.

b. on connaît le nom de la fonction mais on ne sait pas comment l'utiliser *lm ou help* (*lm*) Apparaît alors la fiche détaillée contenant les informations suivantes :

a. Nom du package d'où provient la fonction.

b. Descritpion :brève description.

c. Arguments'r :Pour une fonction détail chacun des arguments.

d. Details :Descritpion détailée.

e. Value : Le type de résultat retourné par la fonction ou l'opérateur

f. See also : Autres rubriques d'aide proche ou similaires à celle documentée

g. Examples : des exemples d'utilisation de la fonction.

## 5.4 Implémentation

Pour les besoins de la programmation de l'approche **P-ACO**, on a utilisé et programmé les sous fonctions suivantes :

**Nombre de projets :** pour déterminer le nombre de projets à sélectionner.

**Portefeuille sélectionné :** pour la détermination des projets à sélectionner (solution initiale).

**Mise à jour locale :** pour effectuer la mise à jour locale des traces de phéromone.

**VaR :** pour le calcul de la Value at risk.

**test de faisabilité :** pour tester la faisabilité de la solution trouvée.

**Test d'efficacité :** pour tester l'efficacité de la solution faisable.

**Mise à jour globale :** pour effectuer la mise à jour globale des traces de phéromone.

## 5.5 résultats

### 5.5.1 Résultats du premier modèle

Le modèle est :

$$(P1)\begin{cases} Optimisez(z) = \left(\sum_{i=1}^{15} x_i E(R_{it}), \sum_{i,j=1}^{15} x_i x_j Cov(R_{it}, R_{jt})\right) \\ 44.7x_1 + 9.47x_2 + 18.92x_3 + \cdots + 80.8x_{15} \leq 200 \\ x_i \in \{0,1\}, t \in [0,T], i = 1, \cdots, 15 \end{cases}$$

**Résultats des estimations des rendements et risque donnés par le bootstrap**

Après une simulation de $s = 600$ échantillons (bootstrap non paramétrique) de notre échantillon d'origine des rendements des cinq actifs, on obtient, pour chaque actif, 600 estimations du rendement moyen et une variance approximée de la moyenne. Pour l'estimation de la matrice des variances covariances des rendements, il suffit de refaire le même travail. Les résultats trouvés sont donnés dans les tableaux ci-après.

| Indice | Rendement moyen | Variance des rendements | Statistique bootstrap | Variance estimée |
|---|---|---|---|---|
| Renault | 0.002123784 | 0.0003168221 | 0.002077639 | 2.461161e-06 |
| Ford | 0.002218014 | 0.0002747171 | 0.002175032 | 1.798519e-06 |
| Suzuki | 0.008723305 | 0.006470278 | 0.008536449 | 4.618037e-05 |
| Peugeot | 0.0001408860 | 0.0009341573 | 0.0001075028 | 7.205594e-06 |
| Toyota | 0.0007933567 | 0.000133964 | 0.0007252773 | 9.509695e-07 |
| Bnp pariba | 0.002388407 | 0.0003177227 | 0.002346357 | 2.173872e-06 |
| Careffour | 0.002290695 | 0.0003648398 | 0.002334333 | 2.817310e-06 |
| Crédit agricole | 0.004944081 | 0.0007086754 | 0.004932067 | 5.486059e-06 |
| Danone | 4.796097e-05 | 0.0001006151 | 5.899838e-05 | 7.592981e-07 |
| STmicroelectric | 0.002605915 | 0.0005692339 | 0.002624199 | 4.262864e-06 |
| France telecom | -0.002151517 | 0.0002512622 | -0.002211196 | 1.674711e-06 |
| L'Oreal | 0.0004973253 | 0.0001524174 | 0.0005846493 | 1.176014e-06 |
| Schneiderelectric | 0.00063083 | 0.0003006303 | 0.0005545822 | 2.332703e-06 |
| Société générale | 0.00393344 | 0.0005237747 | 0.003920474 | 4.036510e-06 |
| Technip | -0.0006763614 | 0.0002270457 | -0.0006745554 | 1.767922e-06 |

TABLE 5.1 – Estimation des rendements des 15 actifs par le bootstrap

| | RENAULT | FORD | SUZUKI | PEUGEOT |
|---|---|---|---|---|
| RENAULT | 3.144077e-04 | -1.670458e-05 | -1.812305e-05 | 5.507846e-05 |
| FORD | -1.670458e-05 | 2.760886e-04 | -1.031277e-05 | 5.608094e-05 |
| SUZUKI | -1.812305e-05 | -1.031277e-05 | 6.453108e-03 | 5.480239e-05 |
| PEUGEOT | 5.507846e-05 | 5.608094e-05 | 5.480239e-05 | 8.792564e-04 |
| TOYOTA | 1.090218e-05 | -1.738805e-05 | -1.343787e-04 | -3.378578e-05 |
| BNP PARIBA | 3.984933e-06 | 5.314647e-05 | 1.921734e-04 | 8.303232e-05 |
| CAREFFOUR | 7.599058e-06 | 7.974914e-06 | 3.096811e-05 | -4.133423e-05 |
| CREDIT AGRICOLE | -4.462647e-05 | 6.341923e-07 | -1.542461e-04 | 1.818132e-05 |
| DANONE | -1.460233e-05 | 2.165477e-05 | 3.726428e-05 | -3.509342e-06 |
| STMICROELECTRONIC | -2.019155e-05 | 4.142648e-05 | 1.627134e-04 | -4.660508e-05 |
| FRANCE TELECOM | 9.714120e-06 | -1.173797e-05 | -7.557765e-05 | -2.063894e-05 |
| L'OREAL | -1.729536e-05 | 1.629538e-05 | -1.037860e-04 | 1.095302e-05 |
| SCHNEIDER ELECTRIC | -8.425923e-06 | 1.584100e-05 | -1.406680e-04 | -1.961635e-05 |
| SOCIETE GENERALE | -3.112952e-05 | -1.711219e-05 | 3.151895e-04 | -8.103038e-05 |
| TECHNIP | -2.967369e-06 | -1.036263e-05 | -9.118809e-05 | 1.494531e-05 |

| | TOYOTA | BNP PARIBA | CAREFFOUR | CREDIT AGRICOLE |
|---|---|---|---|---|
| RENAULT | 1.090218e-05 | 3.984933e-06 | 7.599058e-06 | -4.462647e-05 |
| FORD | -1.738805e-05 | 5.314647e-05 | 7.974914e-06 | 6.341923e-07 |
| SUZUKI | -1.343787e-04 | 1.921734e-04 | 3.096811e-05 | -1.542461e-04 |
| PEUGEOT | -3.378578e-05 | 8.303232e-05 | -4.133423e-05 | 1.818132e-05 |
| TOYOTA | 1.410226e-04 | 6.882378e-07 | 3.569453e-05 | -3.313643e-05 |
| BNP PARIBA | 6.882378e-07 | 3.307808e-04 | -4.245542e-05 | 1.682384e-05 |
| CAREFFOUR | 3.569453e-05 | -4.245542e-05 | 3.836170e-04 | -1.814834e-05 |
| CREDIT AGRICOLE | -3.313643e-05 | 1.682384e-05 | -1.814834e-05 | 6.647076e-04 |
| DANONE | -1.886315e-06 | -6.372454e-06 | 3.348772e-05 | -6.603977e-05 |
| STMICROELECTRONIC | -3.394793e-05 | 1.156618e-04 | -2.836104e-05 | 8.110803e-05 |
| FRANCE TELECOM | 1.430822e-05 | -5.903486e-06 | -3.935126e-05 | -1.240386e-05 |
| L'OREAL | -9.195337e-06 | 1.792963e-05 | 6.861291e-06 | 2.141289e-05 |
| SCHNEIDER ELECTRIC | 5.580201e-06 | -4.164736e-05 | 6.890226e-06 | -8.476708e-05 |
| SOCIETE GENERALE | -1.991224e-05 | 1.357142e-05 | 1.471647e-05 | -3.496885e-05 |
| TECHNIP | 3.244402e-06 | -7.971661e-06 | 8.169847e-06 | -3.677120e-05 |

| | DANONE | STMICROELECTRONIC | FRANCE TELECOM | L'OREAL |
|---|---|---|---|---|
| RENAULT | -1.460233e-05 | -2.019155e-05 | 9.714120e-06 | -1.729536e-05 |
| FORD | 2.165477e-05 | 4.142648e-05 | -1.173797e-05 | 1.629538e-05 |
| SUZUKI | 3.726428e-05 | 1.627134e-04 | -7.557765e-05 | -1.037860e-04 |
| PEUGEOT | -3.509342e-06 | -4.660508e-05 | -2.063894e-05 | 1.095302e-05 |
| TOYOTA | -1.886315e-06 | -3.394793e-05 | 1.430822e-05 | -9.195337e-06 |
| BNP PARIBA | -6.372454e-06 | 1.156618e-04 | -5.903486e-06 | 1.792963e-05 |
| CAREFFOUR | 3.348772e-05 | -2.836104e-05 | -3.935126e-05 | 6.861291e-06 |
| CREDIT AGRICOLE | -6.603977e-05 | 8.110803e-05 | -1.240386e-05 | 2.141289e-05 |
| DANONE | 1.209737e-04 | -1.049471e-05 | 1.931410e-05 | -8.274940e-06 |
| STMICROELECTRONIC | -1.049471e-05 | 5.476626e-04 | -1.814774e-05 | 3.223412e-05 |
| FRANCE TELECOM | 1.931410e-05 | -1.814774e-05 | 2.476558e-04 | -5.603868e-05 |
| L'OREAL | -8.274940e-06 | 3.223412e-05 | -5.603868e-05 | 1.709081e-04 |
| SCHNEIDER ELECTRIC | 3.121245e-05 | 1.132004e-05 | 1.862157e-05 | -1.151288e-05 |
| SOCIETE GENERALE | -2.876705e-05 | 3.361970e-05 | -1.601168e-06 | -3.537035e-06 |
| TECHNIP | 1.144601e-05 | -3.690220e-06 | -3.645890e-06 | 5.733197e-06 |

| | SCHNEIDER ELECTRIC | SOCIETE GENERALE | TECHNIP |
|---|---|---|---|
| RENAULT | -8.425923e-06 | -3.112952e-05 | -2.967369e-06 |
| FORD | 1.584100e-05 | -1.711219e-05 | -1.036263e-05 |
| SUZUKI | -1.406680e-04 | 3.151895e-04 | -9.118809e-05 |
| PEUGEOT | -1.961635e-05 | -8.103038e-05 | 1.494531e-05 |
| TOYOTA | 5.580201e-06 | -1.991224e-05 | 3.244402e-06 |
| BNP PARIBA | -4.164736e-05 | 1.357142e-05 | -7.971661e-06 |
| CAREFFOUR | 6.890226e-06 | 1.471647e-05 | 8.169847e-06 |
| CREDIT AGRICOLE | -8.476708e-05 | -3.496885e-05 | -3.677120e-05 |
| DANONE | 3.121245e-05 | -2.876705e-05 | 1.144601e-05 |
| STMICROELECTRONIC | 1.132004e-05 | 3.361970e-05 | -3.690220e-06 |
| FRANCE TELECOM | 1.862157e-05 | -1.601168e-06 | -3.645890e-06 |
| L'OREAL | -1.151288e-05 | -3.537035e-06 | 5.733197e-06 |
| SCHNEIDER ELECTRIC | 3.166082e-04 | -4.749299e-05 | -2.589689e-06 |
| SOCIETE GENERALE | -4.749299e-05 | 5.029324e-04 | -3.698226e-05 |
| TECHNIP | -2.589689e-06 | -3.698226e-05 | 2.319603e-04 |

TABLE 5.2 – Estimation de la matrice des variances covariances des rendements.

**Exécution de P-ACO**

Après avoir remplaçé les rendements moyens et les covariances par leurs estimations trouvées, on a exécuté le programme P-ACO (100 itérations) sur une machine ayant les caractéristiques suivantes :

**CUP** Core 2 Duo, 2. GHz,

**RAM** 4 GO,

on obtient les résultats suivants :

```
Itération numero: 1
vecteur des ressources
Ant 1  Ant 2  Ant 3  Ant 4  Ant 5  Ant 6  Ant 7  Ant 8
170.75 187.07 184.46  99.94  44.70  36.49 199.00 164.84
la solution faisable:
      A 1  A 2  A 3  A 4  A 5  A 6  A 7  A 8  A 9  A 10  A 11  A 12  A 13
Ant 1     1    1    0    1    0    1    1    1    0     1     0     0     0
Ant 2     0    1    1    1    1    0    1    1    1     1     0     0     0
Ant 3     0    0    0    0    0    0    1    0    0     0     0     1     1
Ant 4     0    0    0    0    0    1    0    0    0     0     0     0     1
Ant 5     1    0    0    0    0    0    0    0    0     0     0     0     0
Ant 6     0    0    0    0    1    0    0    0    0     0     0     0     0
Ant 7     1    0    0    1    0    1    0    1    1     1     1     0     0
Ant 8     0    1    1    1    1    0    0    1    0     1     0     0     0
      A 14  A 15
Ant 1      1     0
Ant 2      1     0
Ant 3      0     0
Ant 4      0     0
Ant 5      0     0
Ant 6      0     0
Ant 7      1     0
Ant 8      0     1
benefice et risque de la solution faisable :
                  Ant 1       Ant 2        Ant 3        Ant 4        Ant 5
benefice_Ant 0.020645222 0.025697654 0.0034188503 0.0030192370 0.0021237840
covariance   0.002260207 0.008859671 0.0005959348 0.0002552119 0.0003144077
                  Ant 6       Ant 7       Ant 8
benefice_Ant 0.0007933567 0.014032957 0.018749196
covariance   0.0001410226 0.002550830 0.007275129
```

```
solution dominante:
[1] NA NA NA NA NA NA NA NA
solution pareto_optimale:
      A 1  A 2  A 3  A 4  A 5  A 6  A 7  A 8  A 9  A 10  A 11  A 12  A 13
Ant 1     1    1    0    1    0    1    1    1    0     1     0     0     0
Ant 2     0    1    1    1    1    0    1    1    1     1     0     0     0
Ant 3     0    0    0    0    0    0    1    0    0     0     0     1     1
Ant 4     0    0    0    0    0    1    0    0    0     0     0     0     1
Ant 6     0    0    0    0    1    0    0    0    0     0     0     0     0
      A 14  A 15
Ant 1      1     0
Ant 2      1     0
Ant 3      0     0
Ant 4      0     0
Ant 6      0     0
benefice et risque de la solution pareto_optimale:
                      Ant 1       Ant 2        Ant 3        Ant 4        Ant 6
benefice_Ant_opt 0.020645222 0.025697654 0.0034188503 0.0030192370 0.0007933567
covariance_opt   0.002260207 0.008859671 0.0005959348 0.0002552119 0.0001410226
```

Au-delà de la 20$^{ime}$ itération l'algorithme n'améliore plus la solution.

```
Itération numero: 100
vecteur des ressources
Ant 1  Ant 2  Ant 3  Ant 4  Ant 5  Ant 6  Ant 7  Ant 8
174.59 186.63 186.63 186.63   9.47 186.63 103.44  38.49
la solution faisable:
      A 1  A 2  A 3  A 4  A 5  A 6  A 7  A 8  A 9  A 10  A 11  A 12  A 13
Ant 1     1    1    0    0    1    1    0    1    0     0     0     0     0
Ant 2     1    1    0    1    1    1    0    1    0     1     0     0     0
Ant 3     1    1    0    1    1    1    0    1    0     1     0     0     0
Ant 4     1    1    0    1    1    1    0    1    0     1     0     0     0
Ant 5     0    1    0    0    0    0    0    0    0     0     0     0     0
Ant 6     1    1    0    1    1    1    0    1    0     1     0     0     0
Ant 7     0    1    0    1    1    1    0    1    0     0     0     0     0
Ant 8     0    0    0    0    0    0    0    0    0     1     0     0     0
      A 14  A 15
Ant 1      1     0
Ant 2      1     0
Ant 3      1     0
Ant 4      1     0
Ant 5      0     0
Ant 6      1     0
Ant 7      0     0
Ant 8      1     0
benefice et risque de la solution faisable :
                  Ant 1       Ant 2       Ant 3       Ant 4        Ant 5
benefice_Ant 0.0164010827 0.01914788 0.01914788 0.01914788 0.0022180140
covariance   0.0008288468 0.00197325 0.00197325 0.00197325 0.0002760886
                  Ant 6       Ant 7         Ant 8
benefice_Ant 0.01914788 0.010484745  6.539355e-03
covariance   0.00197325 0.001857046 -6.042139e-05



solution dominante:
[1] NA NA NA NA NA NA NA NA
solution pareto_optimale:
      A 1  A 2  A 3  A 4  A 5  A 6  A 7  A 8  A 9  A 10  A 11  A 12  A 13
Ant 1     1    1    0    0    1    1    0    1    0     0     0     0     0
Ant 2     1    1    0    1    1    1    0    1    0     1     0     0     0
Ant 3     1    1    0    1    1    1    0    1    0     1     0     0     0
Ant 4     1    1    0    1    1    1    0    1    0     1     0     0     0
Ant 6     1    1    0    1    1    1    0    1    0     1     0     0     0
Ant 8     0    0    0    0    0    0    0    0    0     1     0     0     0
      A 14  A 15
Ant 1      1     0
Ant 2      1     0
Ant 3      1     0
Ant 4      1     0
Ant 6      1     0
Ant 8      1     0
benefice et risque de la solution pareto_optimale:
                      Ant 1       Ant 2       Ant 3       Ant 4       Ant 6
benefice_Ant_opt 0.0164010827 0.01914788 0.01914788 0.01914788 0.01914788
covariance_opt   0.0008288468 0.00197325 0.00197325 0.00197325 0.00197325
                      Ant 8
benefice_Ant_opt  6.539355e-03
covariance_opt   -6.042139e-05
```

### 5.5.2 Résultats du deuxième modèle

Le modèle est :

$$(P2)\begin{cases} Optimisez(z) = \left(\sum_{i=1}^{N} x_i E(R_{it}), (E(\Delta W) * Z_\alpha) + \sigma(\Delta W))\right) \\ 44.41x_1 + 9.43x_2 + 19.33x_3 + 5.74x_4 + 35.42x_5 \leq 90 \\ x_i \in \{0,1\}, t \in [0,T], i = 1, \cdots, 5 \end{cases}$$

**Résultats de P-ACO (rendement estimé par le bootstrap)**

```
   Itération numero: 1

vecteur des ressources
Ant 1  Ant 2  Ant 3  Ant 4  Ant 5  Ant 6  Ant 7  Ant 8
75.69 102.18  55.37 193.32 194.88 187.25 155.60 136.92

la solution faisable:
      p 1  p 2  p 3  p 4  p 5  p 6  p 7  p 8  p 9  p 10  p 11  p 12  p 13
Ant 1    0    1    0    0    0    0    1    1    0     1     0     0     0
Ant 2    1    0    0    1    0    1    0    1    0     0     0     0     0
Ant 3    0    1    1    0    0    0    1    0    0     1     0     0     0
Ant 4    0    0    0    1    1    0    0    1    1     1     0     0     1
Ant 5    0    1    0    0    0    1    0    0    0     0     0     1     0
Ant 6    1    0    1    0    0    1    1    0    1     0     1     0     0
Ant 7    1    0    0    1    1    0    0    1    0     1     0     0     1
Ant 8    0    1    1    0    0    0    1    1    0     0     0     0     0
      p 14  p 15
Ant 1    1     0
Ant 2    0     0
Ant 3    0     0
Ant 4    1     0
Ant 5    1     0
Ant 6    0     0
Ant 7    0     0
Ant 8    0     1

benefice et risque (VaR) de la solution faisable :
                Ant 1        Ant 2       Ant 3       Ant 4        Ant 5
benefice_Ant  0.01599214  0.009597158  0.01583793  0.01309647  0.009037186
VaR          -0.31176058 -0.398019688 -0.32874259 -0.18371383 -0.603201612
                Ant 6       Ant 7       Ant 8
benefice_Ant  0.01342263  0.01123885  0.01749973
VaR          -0.32683257 -0.22718863 -0.09948086
```

```
solution dominante:
[1] NA NA NA NA NA NA NA NA

solution pareto_optimale:
      p 1  p 2  p 3  p 4  p 5  p 6  p 7  p 8  p 9  p 10  p 11  p 12  p 13
Ant 1     0    1    0    0    0    0    1    1    0     1     0     0     0
Ant 2     1    0    0    1    0    1    0    1    0     0     0     0     0
Ant 3     0    1    1    0    0    0    1    0    0     1     0     0     0
Ant 5     0    1    0    0    0    1    0    0    0     0     0     1     0
Ant 8     0    1    1    0    0    0    1    1    0     0     0     0     0
      p 14  p 15
Ant 1      1     0
Ant 2      0     0
Ant 3      0     0
Ant 5      1     0
Ant 8      0     1

benefice et risque de la solution pareto_optimale:
                    Ant 1        Ant 2       Ant 3        Ant 5       Ant 8
benefice_Ant_opt  0.01599214  0.009597158  0.01583793  0.009037186  0.01749973
VaR_opt          -0.31176058 -0.398019688 -0.32874259 -0.603201612 -0.09948086



Itération numero: 100

vecteur des ressources
Ant 1  Ant 2  Ant 3  Ant 4  Ant 5  Ant 6  Ant 7  Ant 8
169.06 144.98  32.12 117.78 189.67 189.67  38.49 189.67

la solution faisable:
      p 1  p 2  p 3  p 4  p 5  p 6  p 7  p 8  p 9  p 10  p 11  p 12  p 13
Ant 1     1    1    1    1    0    1    0    1    0     1     0     0     0
Ant 2     1    1    1    1    0    0    1    1    0     1     0     0     0
Ant 3     0    0    0    0    0    0    0    0    0     0     0     0     0
Ant 4     1    1    1    0    0    1    0    0    0     0     0     0     0
Ant 5     1    1    1    1    0    1    1    1    0     1     0     0     0
Ant 6     1    1    1    1    0    1    1    1    0     1     0     0     0
Ant 7     0    0    0    0    0    0    0    0    0     1     0     0     0
Ant 8     1    1    1    1    0    1    1    1    0     1     0     0     0
      p 14  p 15
Ant 1      1     0
Ant 2      1     0
Ant 3      1     0
Ant 4      0     0
Ant 5      1     0
Ant 6      1     0
Ant 7      1     0
Ant 8      1     0

benefice et risque (VaR) de la solution faisable :
                  Ant 1       Ant 2       Ant 3       Ant 4       Ant 5
benefice_Ant  0.02707783  0.02698012  0.00393344  0.01545351  0.02936853
VaR          -0.35889631 -0.30786554 -1.11923277 -0.60463696 -0.35088316
                  Ant 6       Ant 7       Ant 8
benefice_Ant  0.02936853  0.006539355  0.02936853
VaR          -0.35088316 -0.553775372 -0.35088316
```

```
 solution dominante:
[1] NA NA NA NA NA NA NA NA

 solution pareto_optimale:
      p 1  p 2  p 3  p 4  p 5  p 6  p 7  p 8  p 9  p 10  p 11  p 12  p 13
Ant 1    1    1    1    1    0    1    0    1    0     1     0     0     0
Ant 3    0    0    0    0    0    0    0    0    0     0     0     0     0
Ant 4    1    1    1    0    0    1    0    0    0     0     0     0     0
Ant 5    1    1    1    1    0    1    1    1    0     1     0     0     0
Ant 6    1    1    1    1    0    1    1    1    0     1     0     0     0
Ant 8    1    1    1    1    0    1    1    1    0     1     0     0     0
      p 14  p 15
Ant 1    1     0
Ant 3    1     0
Ant 4    0     0
Ant 5    1     0
Ant 6    1     0
Ant 8    1     0

benefice et risque de la solution pareto_optimale:
                   Ant 1       Ant 3       Ant 4       Ant 5       Ant 6
benefice_Ant_opt  0.02707783  0.00393344  0.01545351  0.02936853  0.02936853
VaR_opt          -0.35889631 -1.11923277 -0.60463696 -0.35088316 -0.35088316
                   Ant 8
benefice_Ant_opt  0.02936853
VaR_opt          -0.35088316
```

### 5.5.3 Discusion

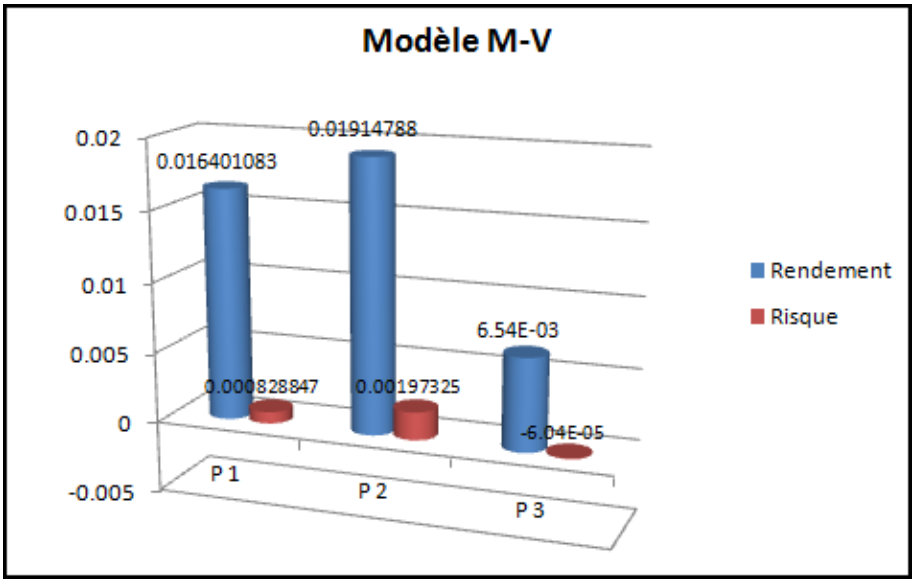



FIGURE 5.5 – Les portefeuilles sélectionnés dans le premier modèle

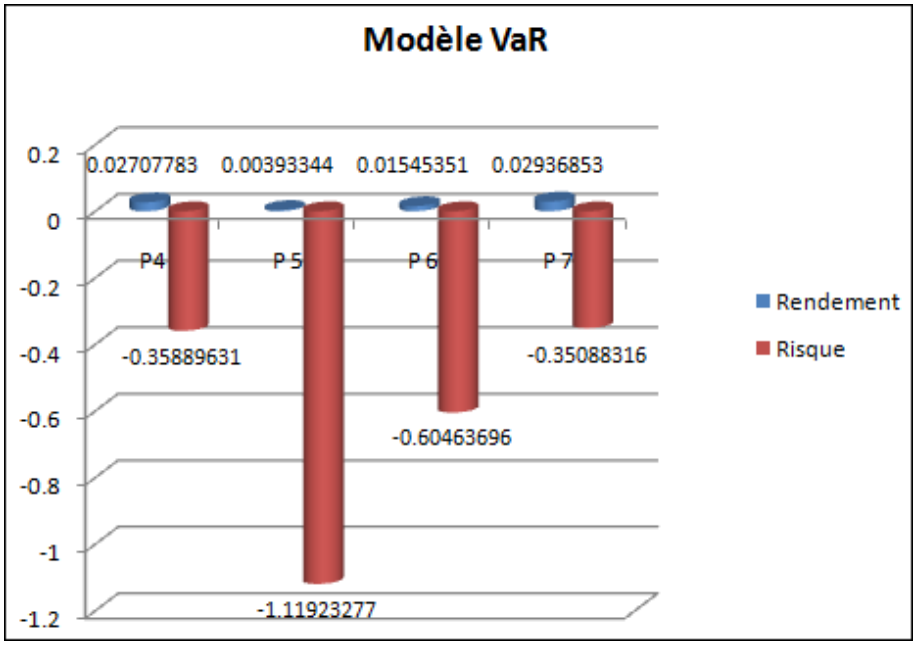



FIGURE 5.6 – Les portefeuilles sélectionnés dans le deuxième modèle

En analysant les caractéristiques des portefeuilles à partir de la figure 5.5, le portefeuille $(P2)$ enregistre une rentabilité relativement accrue $(1.91\%)$ avec un risque de $0.19\%$, les portefeuilles $P1$ et $P3$ avec respectivement $1.64\%$ et $0.65\%$. Si nous nous basons sur le rendement des portefeuilles pour le choix de portefeuille optimal, le portefeuille $(P2)$ représente la stratégie de placement. Pour la figure 5.6, le portefeuille $(P7)$ a la plus grande rentabilité $2.93\%$.

Examinons-nous maintenant la représentation graphique de couple Rendement/Risque et voyons la répartition des portefeuilles selon le rendement et le risque qu'ils présentent.

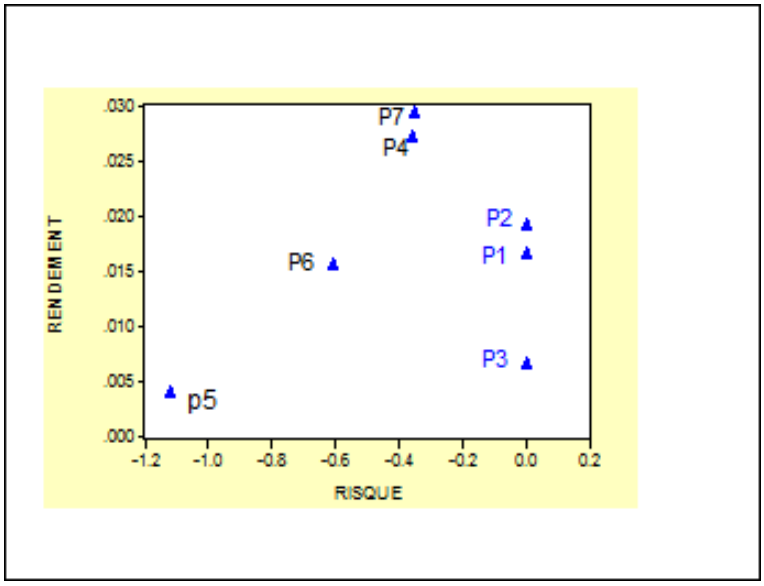



FIGURE 5.7 – Couple rendement risque des portefeuilles

L'ensemble des portefeuilles possibles, résultants des deux modèles M-V et *VaR*, est représenté par un nuage de points dans la figure 5.7.
Pour un niveau du risque donné $(-35.088316\%)$, le portefeuille $P7$ a l'espérance de rendement la plus élevée. Autrement dit, à ce niveau du risque, il n'existe aucun portefeuille ayant une espérance de rendement supérieur : le portefeuille $P7$ domine les portefeuilles $P1$, $P2$ et $P3$. Ces tois portefeuilles sont aussi dominés par le portefeuille $P4$.

Ou encore pour un niveau du rendement donné $(0.39\%)$, le portefeuille $P5$ a le risque le plus petit (à ce niveau du rendement, il n'existe aucun portefeuille ayant un risque plus petit).
En conclusion, les portefeuilles $P4$ et $P7$ résultants du modèle *VaR* dominent tous les portefeuilles $P1$, $P2$ et $P3$ résultants du modèles M-V. En d'autres termes, le modèle *VaR* est meilleur que le modèle M-V.

# Conclusion générale

Notre objectif à travers cette étude est d'apprécier l'apport de l'optimisation stochastique à la résolution des problèmes de gestion de portefeuilles financiers, qui consistent à sélectionner un ensemble d'actif dont le rendement et le risque ne sont pas connus avec certitude durant la période d'investissement.

Afin de résoudre cette problématique, on a transformé le problème stochastique en un problème équivalent déterministe tout en appliquant deux modèles différents : le modèle de Markowitz M-V et le modèle *VaR*. Le premier modèle 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. Ces estimations ont étés établies par une méthode historique le Bootstrap (réechantillonnage).

Par contre, dans le deuxième modèle on estime le risque par la value at risk *VaR*. Cette mesure de risque est plus adoptée par rapport à sa facilité d'approche et à ses caractéristiques : elle est globale et synthétique, car elle permet une évaluation simple et compréhensible pour des néophytes en fournissant un indicateur quantitatif exprimé de façon monétaire.

Par la suite, on a enrichie ce travail en investiguant les capacités de l'optimisation par les métaheuristiques, tout en présentant l'état de l'art de ces approches en proposant une taxonomie des algorithmes fourmis présentés dans la littérature, notamment Pareto Ant Colony Optimisation P-ACO pour la résolution des problèmes de sélection de portefeuilles qui constitue le cœur de notre travail.

# Bibliographie


[1] Affleck-Graves J., MacDonald B : *Non-normalities and tests of asset pricing theories*, Journal of Finance, 1989.

[2] Alaya.I, Solnon.C,Ghédira.K : *Ant algorithm for the multi-dimensional knapsack problem.a* , Proceedings of International Conference on Bioinspired Optimization Methods and their Applications, 63-72,2004

[3] Ben Abdelaziz, F., Krichen, S. *A tabu search heuristic for multiple objective knapsack problems*. Ructor Research Report RR, 1997,28-97.

[4] Bullnheimer, B., Hartl, R.F.,Strauss, C. : *An Improved Ant system Algorithm for the Vehicule Routing Problem.*, Annals of Operations Research, 89, 319-328, 1999.

[5] Coello Coello c., and Becera r. : *Evolutionary multiobjective optimization using acultural algorithm.*, IEEE Swarm Intelligence Sumposium Proceedings, 2003, 6-13.

[6] Collette Y., and Siarry P. *Multiobjective Optimization*. Springer, 2003.

[7] Cornuejols.G,Tütüncü.R : *Optimization Methods in Finance*, Carnegie Mellon University, Pittsburgh, PA 15213 USA.2006.

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

[9] Czyzak, P., Jaszkiewicz, A. : *Pareto simulated annealing a metaheuristic technique for multiple-objective combinatorial optimisation*. Journal of Multi-Criteria Decision Analysis, 7, 1998, 34-47.

[10] Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.*A Fast and Elitist Multiobjective Genetic Algorithm : NSGA-II*. of : IEEE Transactions on Evolutionary Computation, 182-197, 2002.

[11] Deneubourg et al. : *Probabilistic behaviour in ants"*, Journal of Theorical Biology, 105, 259-271.1983.

[12] Doerner, K., Gutjahr, W. J., Hartl, R. F., Strauss,C., Stummer, C. : *Pareto Ant Colony Optimization : A Metaheuristic Approach to Multiobjective Portfolio Selection*,Annals of Operations Research, 2004.

[13] Dorigo.M : *Optimization, Learning and Natural Algorithms*, Ph.D. thesis, Dipartimento di Elettronica, Politecnico di Milano,Italy,1992.

[14] Dorigo.M , G.DI CARO : *The Ant Colony Optimization Meta-Heuristic*, Mc- Graw Hill, UK, 1999.

[15] Dorigo.M , Gambardella.L.M. : *"Ant Colony System : A Cooperative Learning Approach to the Traveling Salesman Problem.*, IEEE Transactions on Evolutionary Computation, 1(1), 53-66, 1997.

[16] Fama E.F : *The behaviour of stock market prices*, Journal of Buisness, vol.38,1965.

[17] Fonseca, C.M., Fleming, P.J. *Genetic algorithms for multi-objective optimization : formulation, discussion and generalization*. The Fifth International Conference on Genetic Algorithms,416-423,1993.

[18] Jaszkiewicz, A . *On the Performance of Multiple-Objective Genetic Local Search on the* $0/1$ *Knapsack Problem-A Comparative Experiment* . IEEE Transactions on Evolutionary Computation, 6(4),2002, 402-412.

[19] Jorion, P. : *Value at risk : the new Benchmark for controlling market Risk.*, JMcGraw-Hill, New-york, 1997.

[20] Haimes Y. Y., H. W. A., and Freedman H. T. *A multiobjective optimization in water resources systems*. Elsevier Scientific, 1975.

[21] Hansen, M.P. *Metaheuristics for multiple objective combinatorial optimization*. Ph.D. Thesis, Technical University of Denmark, Lyngby, 1998.

[22] Horn, J., Nafpliotis, N., Goldberg, D.E. *A niched pareto genetic algorithm for multiobjective optimization.* IEEE Service (ed), First IEEE Conference on Evolutionary Computation, vol. 1. Piscataway : IEEE World Congress on Computational Computation,82-87, 1994.

[23] Gambardella, L., Taillard, E., Dorigo, M. : *Ant Colonies for the Quadratic Assignment Problem.*, Journal of the Operational Research Society,50,1999, 167-176,New-york, 1997.

[24] Gandibleux, X., Mezdaoui, N.,Fréville, A. : *A multiobjective tabu search procedure to solve combinatorial optimization problems* . Advances in Multiple Objective and Goal Programming. Lecture Notes in Economics and Mathematical Systems, 1997, 291-300.

[25] Guillaume Nolain, Yahia Salhi, Serge Werlé : *Optimisation de portefeuille selon le crit'ere de la Value at Risk*,Mai 2007.

[26] Keeney R. L., and Raaiffa H. : *Decision with Multiple Objective : Preference and value Tradoff*. Cambridge University Press, 1993.

[27] Markowitz.H : *Portfolio selection*, Journal of Finance 7 ,1952, 77-91.

[28] Miettienen K. M., and Mäkelä M. M. : *Proper pareto Optimality in Nonconvex Problems-characterization with Tangent and Normal Cones*, Technical report. Jyväskylä University, 1998.

[29] Osyczka a : *Multicritaria optimization for engineering design.*, Design optimization. Academic press, Cambrige,1985, 193-227.

[30] Pirlot M. *Métaheuristiques pour l'optimisation combinatoire*. Hermés Science Publications, Paris, 2002.

[31] Serafini, P. :*Simulated annealing for multiple objective optimization problems.*, Tenth Int. Conf. on Multiple Crit Decision Making, 1992,87-96.

[32] Schaffer, J.D. : *Multiple objective optimization with vector evaluated genetic algorithms*, ICGA International Conference on Genetic Algorithms, 1985,93-100.

[33] Silverman, B. W. :*Density Estimation for Statistics and Data Analysis.*, 1986.

[34] **Solnon, C.** : *Ants can Solve Constraint Satisfaction Problems.*, IEEE Transactions on Evolutionary Computation, 6(4), 2002, 347-357.

[35] *Suppapitnarm, A., Parks, T. Simulated annealing : an alternative approach to true multiobjective optimization* , Genetic and Evolutionary Computation Conference, 2001.

[36] Georges Kolomvos : *Résolution de grands problèmes stochastiques multi-étapes :Application à un problème de dimensionnement de capacités et de gestion de flux et de stocks*, école centrale des arts et manufactures, école centrale Paris, 2007.

[37] *RiskMetrics JPMorgan*, Technical document 4th edition, 1996.

[38] Takahashi.R.H.C, Deb.K, Wanner.E.F, Greco.v : *Evolutionary Multi-Criterion Optimization*, 6th International Conference, EMO 2011,Ouro Preto, Brazil, April 2011,5-8.

[39] Ulungu, E. L., Teghem, J., Frtemps, P., Tuyttens,D. : *MOSA method : A tool for solving multi-objective combinatorial optimization problems*. Tech. rept. Laboratory of Mathematic and Operational Research, Faculté polytechnique de Mons, 1998.

[40] Zitzler, E., Laumanns, M., Thiele, L : *Improving the Strength Pareto Evolutionary Algorithm*,Athens, Greece,2001,12-21.