Université Mohamed el Bachir el Ibrahim Bordj Bou Arreridj

Faculté Mathématiques et Informatique

Département Recherche Opérationnelle

Mémoire

En vue de l'obtention du diplôme de

MASTER

FILIÉRE DES MATHÈMATIQUE APPLIQÈES

Spécialité : Méthodes et outils pour la recherche opérationnelle

---

Strength Pâreto Evolutionary Algorithm for multi-objective knap sack problem.

Réalisé par :

* Zaghdane Nahla

* Zidani Ihsane

Sous la direction de :

Pr. Zaouche Djaafar

Soutenu le Septembre 2021, Devant le jury:

- Président : Pr. Alloui lotfi
- Encadreur : Pr. Zaouche Djaafar
- Examinateur : Pr. Maaza Sofiane

Année Universitaire

2020/2021

Remerciements

Nous tenons à remercier tout premièrement Dieu le tout puissant pour la volonté la santé et la patience, qu'il nous a donné durant toutes ces longues années.

Ainsi, nous tenons également à exprimer nos vifs remerciements à notre encadreur Mr. Zaouche Djaafar pour avoir d'abord proposée ce thème, pour suivi continuel tout le long de la réalisation de ce mémoire et qui n'a pas cessé de nous donner ses conseils.

Nous tenons à remercier vivement toutes personnes qui nous ont aidés à élaborer et réaliser ce mémoire, ainsi à tous ceux qui nous ont aidés de près ou de loin à accomplir ce travail.

Nos remerciements vont aussi à tous les enseignants qui ont contribué à notre formation par ailleurs, Nos remerciements à tous les membres du jury qui ont accepté de juger notre travail. En fin, nous tenons à exprimer notre reconnaissance à tous nos amis et collègues pour le soutien moral et matériel…

DEDICACES

Je dédie ce modeste travail à :

Mes honorable mon père et ma mère puisqu'ils sont la lumière de ma vie.

Tous mes frères et mes soueurs.

A l'ensemble de ma famille : Zeghdane, notamment me grand-mère

Mon encadreur : Dr. Zawach j

Mes collègues en particulier :

Marwa, Somia, khalisa, khawla, Meriem, Youcef, Ouahiba, Manel et Nadia ,

chaima et bouthayna

A tous ceux qui ont été à mes cotés dans les moments difficiles

ZAGHDANE NAHLA

DEDICACES

A ma mère ...
La plus belle créature que Dieu a créée sur terre ...
Cette source de tendresse, de patience et de générosité...
Qui a œuvré pour ma réussite, de par son amour, son
Soutien , tous les sacrifices consentis et ses précieux
Conseils , pour toute son assistance et sa présence dans
ma vie , reçois à travers ce travail aussi modeste soit-il,
l'expression de mes sentiments et de mon éternelle
gratitude.
A mon père ...
Qui peut être fier et trouver ici le résultat de longues
années de sacrifices et de privations pour m'aider à
avancer dans la vie. Puisse Dieu faire en sorte que ce
travail porte son fruit ; Merci pour les valeurs nobles,
l'éducation et le soutient permanent venu de toi.
A ma chère grande mère.
A ma sœur Meryama , Salma et mon frère Brahim ,ahmed.
A tout ma famille.
A mes chères amis Fella, Manale, Nadai
Pour leur soutien et leur assistance.
A tous les enseignantes de département Recharche Oprationnelle de l'université de BBA.

ZIDANI IHSANE

Tableau des matières

III. Le problème de sac à dos

IV. Conception et réalisation

Tableau des Figures

Liste Des Tableaux

Introduction générale

Introduction Générale

Un problème d'optimisation combinatoire consiste à parcourir l'espace de recherche pour extraire une solution optimale parmi un ensemble fini des solutions d'une taille souvent très grande telle que son énumération exhaustive est une tâche fastidieuse parmi les problèmes combinatoires,(le problème du voyageur de commerce et le problème de sac à dos). La résolution d'un problème d'optimisation combinatoire nécessite l'utilisation d'un procédé algorithmique permettant la maximisation ou la minimisation d'une ou de plusieurs fonctions « objectif » en respectant les contraintes posées par le problème. Le nombre de fonctions « objectif » du problème à traiter permet de le classer en problème mono objectif ou problème multi objectifs. En fait, un problème mono objectif consiste à optimiser une seule fonction « objectif », Tandis qu'un problème multi objectifs consiste à optimiser deux ou plusieurs fonctions « objectif » souvent contradictoires. La résolution de ce type des problèmes (les problèmes multi objectifs) consiste à trouver un compromis entre les différents objectifs imposés. En effet, la plupart de ces problèmes appartiennent à la classe des problèmes NP-difficiles et ne possèdent donc pas à ce jour de solution algorithmique efficace valable pour toutes les données.

Le problème du sac à dos multi-objectif à variables bivalentes est une variante du problème du sac à dos (KP) dont la résolution est beaucoup plus difficile. Les objets mis dans le sac à dos doivent maximiser la valeur totale sans dépasser le poids maximum. Le fait qu'on rencontre ce problème dans des domaines d'application aussi différents que l'économie, l'industrie, les transports et le chargement de cargaisons , l'informatique répartie lui confère un grand intérêt pratique. Ce problème d'optimisation combinatoire sous contraintes est NP- complet, les méthodes exactes existantes sont limitées à des petites instances. Aujourd'hui, la résolution des instances très difficiles s'effectue grâce à des approches heuristiques, l'aptitude de ces dernières à fournir des solutions de bonne qualité les rends indispensables dans le domaine pratique et elles s'avèrent aussi très utiles pour le développement des méthodes exactes fondées sur des techniques d'évaluation et de séparation.

Dans ce mémoire, nous avons appliquées l'algorithme évolutionnaire multi-objectif pour la résolution des problèmes de sac-à-dos multi-objectif. Les algorithmes évolutionnaires multi-objectifs sont inspirés par l'évolution génétique et la sélection naturelle dont le rôle de trouver l'ensemble du Pareto optimal.

Ces algorithmes sont invoqués dans plusieurs problèmes réels tels que l'ingénierie, l'économie , la logistique ... etc. Nous pouvons citer : Non-dominated Sorting Genetic Algorithm (NSGA), Strength-Pareto Evolutionary Algorithm (SPEA), Strength Pareto Evolutionary Algorithm2 (SPEA2), Non-dominated Sorting Genetic Algorithm (NSGA-II), Multi-Objective Genetic Algorithm (MOGA) et Pareto Archived Evolution Strategy (PAES).

L'organisation de ce mémoire est la suivante

Chapitre 01 : Aborde la présentation générale de l'optimisation combinatoire.

Chapitre 02 : Porte sur l'algorithme génétique mon objectifs et multi objectifs.

Chapitre 03 : Concerne l'étude générale des problèmes de sac à dos et ses variantes.

Chapitre 04 : Expose l'implémentation d'algorithme SPEA-II pour résoudre le problème de sac à dos multi-objectifs avec un exemple d'application.

Chapitre I :

Probléme d'optimisation Combinatoire

I.1 Introduction

Un problème d’optimisation combinatoire consiste à trouver la meilleure solution dans un ensemble discret de solutions appelé ensemble des solutions réalisables. En général, cet ensemble est fini mais de cardinalité très grande.

Dans ce chapitre, nous allons présenter quelques concepts de base de l’optimisation combinatoire les éléments de base relatifs lesquelles nous allons utiliser dans notre étude. A cela, nous ajoutons quelques exemples de ces problèmes d’optimisation combinatoire avec leurs complexités algorithmiques et les outils de modélisation des problèmes des optimisations combinatoires puis des méthodes de résolution exactes et approchées. en plus de présenter le contexte de l’optimisation multi objectifs et ses principales définitions.

I.2 Problème d'optimisation combinatoire

I.2.1 Définition

Un problème d’optimisation combinatoire est défini par un ensemble d’instances. À chaque instance du problème est associé un ensemble discret de solutions S, un sous-ensemble X de S représentant les solutions admissibles (réalisables) et une fonction de coût f (ou fonction objective) qui assigne à chaque solution x$\in X$. Pour résoudre les problèmes combinatoires, il faut déterminer une solution $s^* \in X$ qui optimise la valeur de la fonction de coût f. Cette solution s^*est appelée la solution optimale ou la solution optimale globale qui se divise en trois types :

- ✓ Les problèmes combinatoires dans un domaine discret.
- ✓ Les problèmes combinatoires continus.
- ✓ Les problèmes aléatoires.

Sa formulation mathématique, dans sa forme mixte, est la suivante :

$$\begin{cases} \boldsymbol{min_x\, f(x)} \\ \text{s.c}\;\; \boldsymbol{g_i(x) \leq 0, i = 1, \dots, m} \\ \boldsymbol{h_j(x) = 0, j = 1, \dots, p} \\ \boldsymbol{x \in S} \end{cases} \tag{I-1}$$

Nous présentons rapidement ici quatre problèmes classiques d’optimisation combinatoire : le problème du sac-à-dos, le problème d’affectation, le problème du voyageur de commerce et le problème d’ordonnancement [**1].**

I.2.2 Principaux concepts en optimisation

Tout d’abord, nous définissons les notions communes à n’importe quelle méthode d’optimisation :

- **Vecteur de décision :** Un vecteur de décision est un vecteur correspondant à l'ensemble des variables du problème, il est noté : $X^* = [x_1, x_2, \dots, x_n]^T$ avec : n le nombre de variables ou dimension du problème et k_x la variable sur la dimension k.
- **Fonction objectif :** est un critère sur lequel sont jugés les vecteurs de décision pour déterminer le meilleur vecteur. Un critère peut être une variable du problème ou une combinaison de variables.
- **Contraintes :** Une contrainte du problème est une condition que doivent respecter les vecteurs de décision du problème. Une contrainte est notée : $g_i(x^*)$ avec : i =1,..., m, avec m est le nombre des contraintes.
- **Minimum local :** Un point x ∈ X est un minimum local du problème si et seulement si : $\forall\, x' \in V(x)$, $f(x) \leq f(x')$ où V (x) définit un voisinage de x.
- **Minimum global :** Un point x ∈ X est un minimum local du problème si et seulement si : $\forall x' \in X, f(x) \leq f(x')$ (il appartient alors à l'ensemble des minimisera). **[2]**

I.3 Résolution d'un problème d'optimisation combinatoire

La résoudre au problème d'optimisation combinatoire nécessite l'étude de trois points particuliers :

- ✓ La définition de l'ensemble des solutions réalisables,
- ✓ L'expression de l'objectif à optimiser,
- ✓ Le choix de la méthode d'optimisation à utiliser,

Les deux premiers points relèvent de la modélisation du problème, le troisième de sa résolution. Afin de définir l'ensemble des solutions réalisables, il est nécessaire d'exprimer l'ensemble des contraintes du problème. Ceci ne peut être fait qu'avec une bonne connaissance du problème sous étude et de son domaine d'application.

I.4 Complexité d'un problème d'optimisation combinatoire

Avant d'aborder à la complexité, les définitions et les classifications des problèmes sont alors comme suit :

- **Classe P :** Un problème de décision est dit appartenir à la classe P, s'il existe un algorithme pouvant le résoudre en temps polynomial **[3].**
- **Classe NP :** La classe NP renferme tous les problèmes de décision dont on peut associer à chacun d'eux un ensemble de solutions potentielles (de cardinal au pire exponentiel) tel qu'on puisse vérifier en un temps polynomial si une solution potentielle satisfait la question posée. De toute évidence, P⊂NP qu'on suppose communément acceptée. De plus cette inclusion est stricte, c'est-à-dire P# NP

- **Classe NP-complet :** On distingue également dans la classe NP, la classe des problèmes NP-complets. La NP-complétude s'appuie sur la notion de réduction polynomiale **[4].**
- **Classe NP-difficile :** Un problème est NP-difficile si savoir le résoudre en temps polynomial impliquerait que l'on sait résoudre en problème (de décision) NP-complet en temps polynomial. Les problèmes NP-difficiles sont donc dans un sens plus dur que les problèmes NP-complet. Ce schéma représente la relation entre les classes des problèmes :

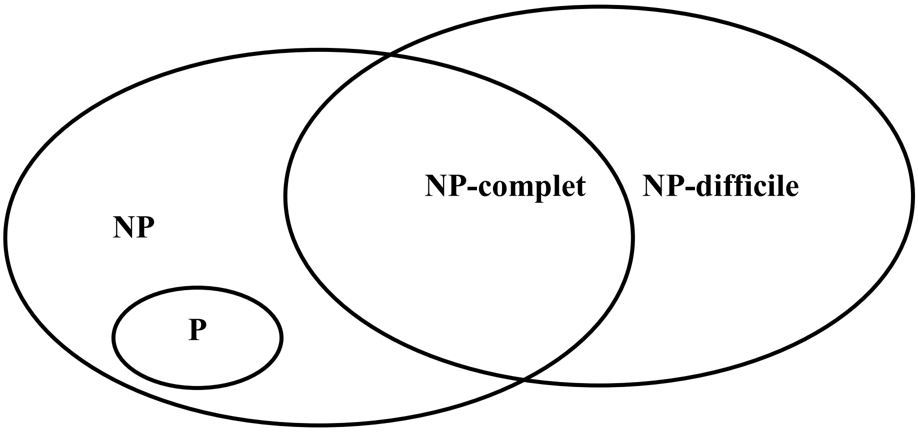


Figure I.1 Relation entre les classes des problèmes.

Exemples :

L'ensemble des problèmes d'optimisation est très vaste, on peut les citer :

- ✓ Le problème de voyageur de commerce (classe NP-difficile).
- ✓ Le problème de connexité d'un graphe (classe P).
- ✓ Le problème de circuit Hamiltonien (classe NP-complet).
- ✓ Le problème du sac à dos (classe NP- difficile).

I.5 Problème d'optimisation multi-objectif

L'optimisation multi-objective est une branche de l'optimisation combinatoire dont la particularité est de chercher à optimiser simultanément plusieurs objectifs d'un même problème. Un problème d'optimisation multi-objectif (MOP) consiste à optimiser (minimiser/maximiser) simultanément plusieurs fonctions objectifs $f_1; f_2; \dots; f_p, p \geq 2$ sur un domaine (ensemble) réalisable S appelle espace de décision définit par des inéquations $g_j(x)$ ≤0, j= 1;2;...;m. Mathématiquement, ce problème peut s'écrire comme suit :

$$(\boldsymbol{MOP})\begin{cases}\textbf{min } \boldsymbol{ou} \textbf{ max}\left(\boldsymbol{f_1(x)}\,;\boldsymbol{f_2(x)}, \dots\,;\boldsymbol{f_p(x)}\right)^{\boldsymbol{T}}, \boldsymbol{p} \geq \mathbf{2} \\ \boldsymbol{g_i(x)} \leq \mathbf{0}, \boldsymbol{j} = \mathbf{1}\,;\mathbf{2}\,;\dots \boldsymbol{m}\end{cases} \qquad \textbf{(I-2)}$$

Où $x = (x_1; x_2; \dots; x_n)^T$ est le vecteur de n variables da décisions. On pose $F(x) = \left(f_1(x); f_2(x), \dots; f_p(x)\right)^T$ est le vecteur de p fonction objectifs à optimiser **[5].**

I.5.1 Notion de dominances et d'optimalite dominance

a) dominance :

On dit que le vecteur de décision $u = [u_1\,, u_2, \dots u_k]^T$ domine le vecteur $v = [v_1, v_2, \dots v_k]^T$, (dénoté : $u < v$), si et seulement si

$\forall i \in \{1,2, \dots K\}, f_i(u) \leq f_i(v) \cap \exists i \in \{1,2, \dots N\}: f_i(u) < f_i(v)$**[6].**

b) dominance faiblement :

Une solution $u = (u_1, \dots, u_k)^T$ domine faiblement une solution $v = (v_1, \dots, v_k)$, si et seulement si : $\forall i \in 1 \dots k\ f_i(u) \leq\ f_i(v)$.

c) Ensemble de Pareto optimal (P^*) :

Pour un problème d'optimisation multi-objectif : $F(x) = [f_1(x), f_2(x), \dots, f_k\,(x)]^T$.
L'ensemble du Pareto optimal (P^*) est défini par l'équation suivant :

$P^* \ = \{x \in \Omega\ |\neg\exists x' \in \Omega: F(x') \leq F(x)\}$.

d) Frontière du Pareto (FP) :

Pour un problème d'optimisation multi-objectif de l'ensemble des fonctions $F(x)$avec P^* l'ensemble du Pareto optimal, la frontière du Pareto FP est définie comme suit :

$FP \ = \left\{u \ = F = \left(f_1(x), \dots, f_k(x)\right)|x \in P^*\right\}$.**[7]**

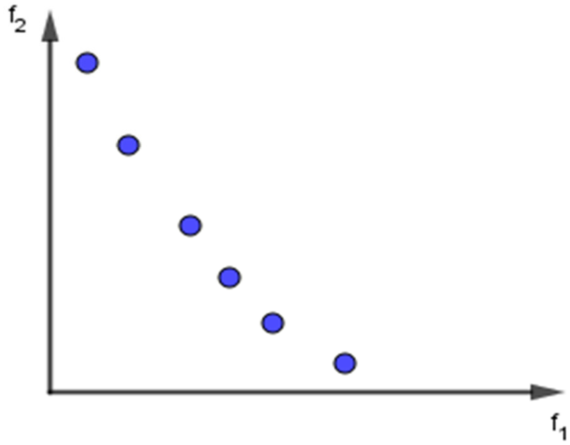


Figure I.2 Exemple de frontière du Pareto.

I.6 Les méthodes de résolution des problèmes multi objectifs

D'optimisation multicritères s'intéressent aux problèmes essayant d'optimiser simultanément plusieurs objectifs. Elles peuvent être classées selon le schéma suivant Les méthodes .

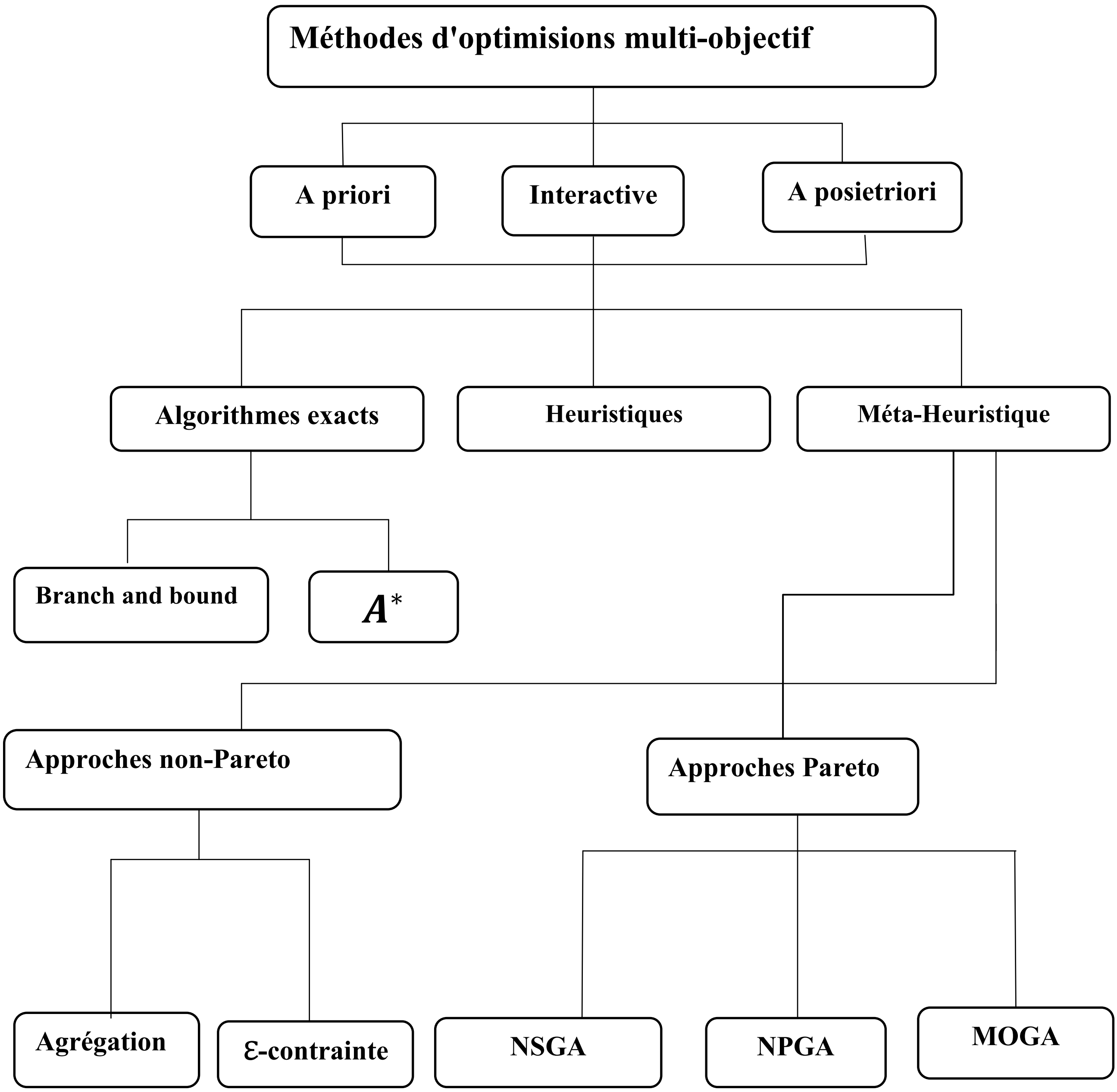


Figure I.3 classification des méthodes d'optimisation multi objectif

Selon le mode de coopération entre le solveur et le décideur, elles peuvent être rangées en trois familles. **[8]**

- **Les méthodes à préférence à priori**

L'utilisateur définit le compromis qu'il désire réaliser avant de lancer la méthode d'optimisation. On retrouve dans cette famille, toutes les méthodes agrégatives.

- **Les méthodes à préférence progressives**

L'utilisateur affine son choix de compromis au fur et à mesure du déroulement de l'optimisation. On retrouve dans cette famille, toutes les méthodes interactives.

- **Les méthodes à préférence à priori**

L'utilisateur choisit une solution de compromis parmi les solutions de la surface de compromis.

Les algorithmes de résolutions des MOP répartissent en trois catégories :

a) Les méthodes exactes

Ces méthodes permettent d'obtenir l'optimum global de manière exacte pour certains types de problèmes. Elles sont efficaces seulement lorsque le PMO à résoudre est bi-objectif et l'espace de recherche est de petite taille.

b) Les heuristiques

Elles sont dédiées à la résolution d'un PMO spécifique et ne fonctionnent que sur ce type de problèmes.

b.1) Méta-heuristiques

Elles sont fondées sur une idée générale. Et peuvent s'adapter à n'importe quel type de problème et donner de bons résultats. On trouve dans cette classe les algorithmes génétiques, optimisation par Essaim particulaire, la recherche tabou, le recuit simulé,…

Les métas-heuristiques résolvant les PMO peuvent être classées en deux approches :

b.2) Approches non Pareto

Ces approches traitent séparément les différents objectifs, elles sont sensibles au paysage du front de Pareto. Elles sont efficaces et faciles à implémenter seulement pour les PMO avec un nombre réduit d'objectifs.

b.2.1) Méthodes d'agrégation par pondération

Les méthodes agrégatives fusionnent les différentes fonctions objectives pour sera mener à un problème d'optimisation mono-objectif. Les coefficients sont généralement choisis en fonction de l'importance relative des objectifs. Soit

$\mathbf{F}_{\boldsymbol{\omega}} = \sum_{m=1}^{M} \boldsymbol{\omega}_{m} \, \mathbf{f}_{m}(\mathbf{x})$, Où les poids $\boldsymbol{\omega}_{m} \geq \mathbf{0}$ sont tels que $\sum_{m=1}^{M} \boldsymbol{\omega}_{m} = \mathbf{1}$.

La résolution du PMO consiste donc à résoudre le problème d'optimisation suivant :

$$\mathbf{minimiser}_{\mathbf{x}\in\mathbf{S}} \mathbf{F}_{\boldsymbol{\omega}}(\mathbf{x})[9].$$

b.2.2) Méthode ε-contrainte

Ce n'est pas une méthode d'agrégation des fonctions objectives. Elle est aussi dite méthode du compromis Elle transforme un PMO en un problème d'optimisation monoobjectif de la façon suivante :

- ✓ Choisir un objectif à optimiser prioritairement,
- ✓ Choisir un vecteur de contraintes initiales $\mathcal{E}^*$
- ✓ transformer le problème en gardant l'objectif prioritaire et en transformant les autres objectifs en contraintes d'inégalités comme suit :

$\left(\begin{smallmatrix}\mathbf{Minimiserf}^*(\mathbf{x}^*) \\ \mathbf{Avecx}^*\in\mathbf{S_y}\end{smallmatrix} \rightarrow\right)$ $\mathbf{minimiser\ f_1^*}(\mathbf{x}^*)$ Avec $\mathbf{f_2^*} \leq \mathcal{E}_\mathbf{2}, \dots, \mathbf{f_n^*}(\mathbf{x}^*) \leq \mathcal{E}_\mathbf{n}, \mathbf{x}^* \in \mathbf{S_y}$

b.3) Approches Pareto

Ces approches utilisent la notion de dominance lors de la sélection des solutions. Elles sont capables de générer des solutions Pareto situées dans des concavités.

Historiquement, on peut considérer l'existence de deux générations des approches Pareto.La première génération est caractérisée par l'utilisation :

- ✓ D'un mécanisme de sélection Pareto. En effet la relation de dominance permet d'affecter des rangs aux individus de la population, ce qui fait apparaître la notion de front. C'est la technique de ranking.
- ✓ Du sharing (niching) qui consiste à pénaliser la valeur d'adaptation en fonction du nombre d'individus au voisinage du regroupement cette technique introduite dans est largement utilisée maintenant.
- ✓ Les algorithmes correspondants sont relativement simples. Parmi les approches les plus représentatives de cette génération on peut citer :
  - ➢ M.O.G.A (Multi-Objective Genetic Algorithm).
  - ➢ N.S.G.A (None dominated Sorting Genetic Algorithm).
  - ➢ N.P.G.A (Niched-Pareto Genetic Algorithm).

Après environs dix années de succès, cette génération devient dépassée. La deuxième génération des approches Pareto est maintenant considérée comme l'état d'art de l'optimisation multi-objectif. Ce sont en général des approches élitistes manipulant une population secondaire externe. Parmi les approches les plus représentatives de cette génération on peut citer :

- ✓ SPEA et SPEA2 (Strength Pareto Evolutionary Algorithm)
- ✓ NSGA-II (Non dominated Sorting Genetic Algorithm)
- ✓ PAES (The Pareto Archived Evolution Strategy)
- ✓ PESA et PESA II (The Pareto Envelope-basedSelectionAlgorithm) **[10].**

I.7 Conclusion

Nous avons présenté dans ce chapitre les concepts principaux de l’optimisation combinatoire indispensables à la compréhension de notre travail. Ces concepts sont centrés sur la complexité algorithmique, les différentes méthodes de résolution des problèmes d’optimisation, en particulier, les méthodes approchées et les méthodes exactes.

Chapitre II :

Les Algorhimes Génétiques

II-1 Introduction

Dans ce chapitre, nous mettrons en évidence les deux parties des algorithmes génétiques à mono-objectifs et multi-objectifs et expliquerons la fonction de base ainsi que les concepts liés à cet algorithme.

II-2 Algorithme génétique

II-2-1 Description de l'algorithme génétique

L'algorithme génétique (GA) est un algorithme évolutif, il est classé parmi les métas heuristiques inspirés de la théorie de l'évolution naturelle. Il est introduit pour la première fois par John Holland en 1975.

II-2-2 Principes généraux des algorithmes génétiques

Pour utiliser un Algorithme Génétique pour un problème particulier, on doit donc disposer des cinq éléments suivants :

- **Le codage des solutions (individus)**

Associe à chacun des points de l'espace d'états une structure de données. Elle vient généralement après une phase de modélisation mathématique du problème traité. La qualité du codage conditionne le succès de l'algorithme.

- **Une méthode de génération de la population initiale**

La population d'individus produite initialement qui servira de base pour les générations futures doit être non homogène.

- **Une fonction à optimiser**

Ou fonction d'évaluation de l'individu. Cette fonction retourne une valeur réelle appelée fitness, qui va permettre de déterminer la probabilité de sélection d'un individu.

- **Les opérateurs génétiques**

Permettent de diversifier la population au cours des générations et d'explorer l'espace d'états. Le croisement recompose les gènes d'individus de la population. La mutation entraîne des altérations minimes sur les individus pour éviter la convergence rapide de la population. La sélection favorise les meilleurs individus.

- **Les paramètres de dimensionnement**

Taille de la population, nombre total de générations ou critère d'arrêt, probabilités d'application des opérateurs de croisement et de mutation **[11].**

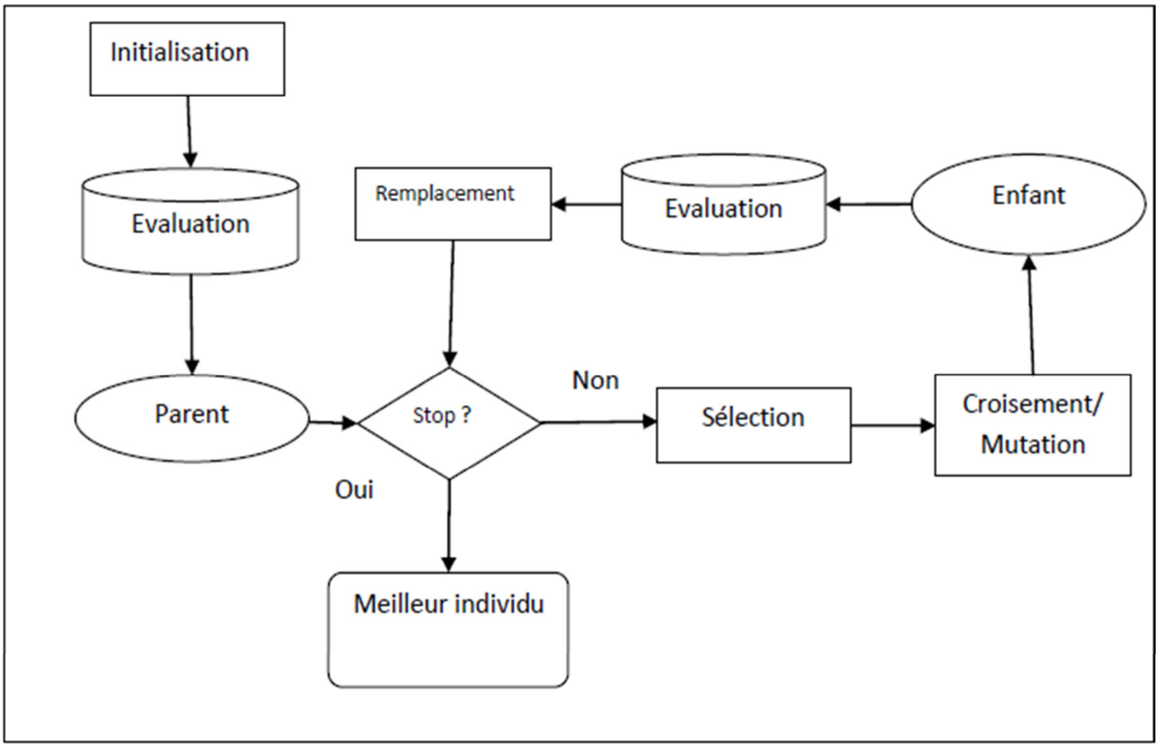


Figure II.1 Les principales étapes d'un algorithme génétique

II.2.3 Le codage des solutions

Les algorithmes génétiques sont appliqués sur une population d'individus, chacun de ces derniers est codé par un chromosome ou génotype. Donc, une population est présentée par un ensemble de chromosomes. Le processus de codage d'individus consiste à trouver une structure de données pour les individus.

On distingue plusieurs types de codage :

- **Codage binaire**

Le codage binaire est un codage élémentaire dont le principe consiste à coder la solution selon une chaîne de bits. Une chaîne de bits est une suite de chiffres, chacun d'entre eux pouvant prendre la valeur 0 ou 1. (Voir la figure II.2). Evidemment, ce codage est le plus répandus en raison de ses nombreux avantages lors de la manipulation des gènes.

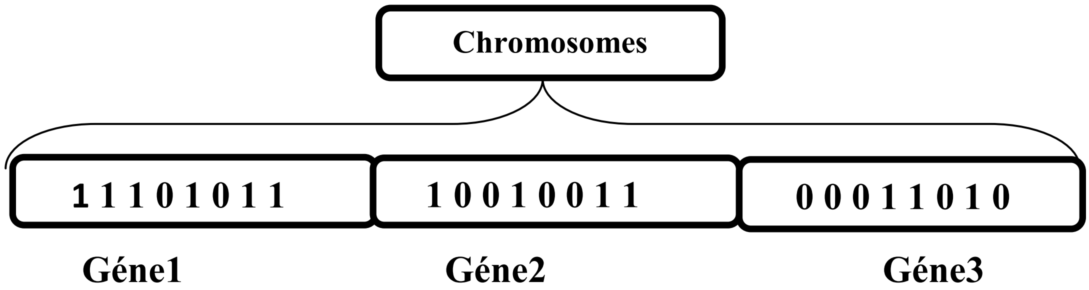


Figure II.2 Chromosome codé en binaire

- **Codage réel**

Le codage binaire est très utile, mais sa représentation montre ses limites lorsque le nombre des paramètres d'une solution est assez grand, ce qu'il rend plus fastidieux et plus difficile de gérer les chromosomes. Une solution possible est l'utilisation des nombres réels pour coder les gènes des chromosomes. **[12]**

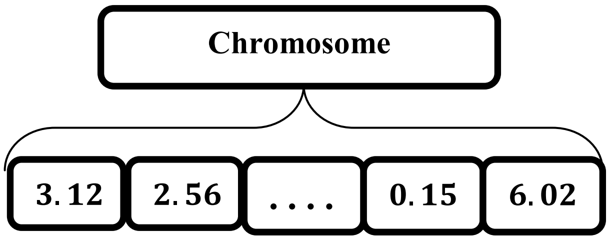


Figure II.3 Chromosome codé réel

- **Codage à caractères multiples**

Par opposition au codage binaire et réel, une autre manière de coder les chromosomes d'un algorithme génétique est le codage à l'aide de caractères multiples. Souvent, ce type de codage est plus naturel que le codage binaire. C'est d'ailleurs celui-ci qui est utilisé dans de nombreux cas poussés d'algorithmes génétiques **[13]**.

Figure II.4 Codage caractères multiples d'un chromosome (Problème bio- informatique).

- **Codage sous forme d'arbre**

Ce codage utilise une structure arborescente ; un arbre est une structure de données munie d'une racine de laquelle peuvent être issus un ou plusieurs fils. Un de leurs avantages est qu'ils peuvent être utilisés dans le cas de problèmes ou les solutions n'ont pas une taille finie. En principe, des arbres de taille quelconque peuvent être formés par le biais de crossing-over et de mutations.

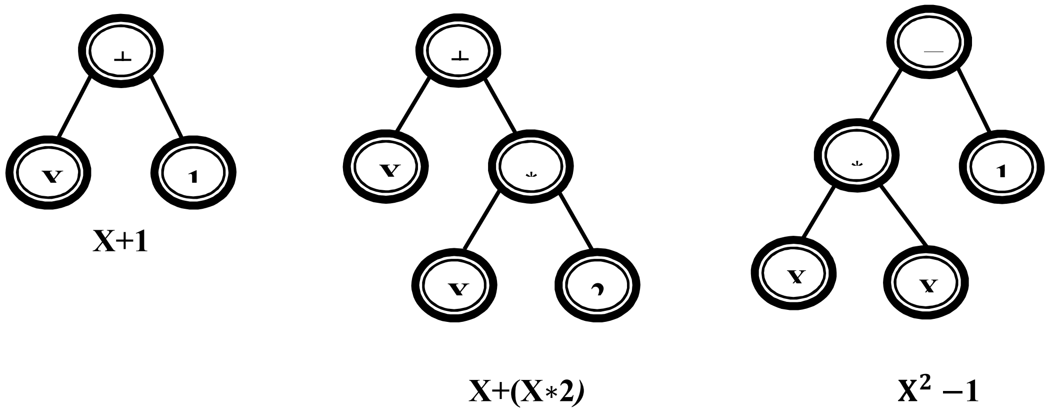


Figure II.5 Codage en arbre : Trois arbres simples de programme de la sorte normalement utilisée dans la programmation génétique.

II.2.4 Création de la Population Initiale

Le choix de la population initiale conditionne fortement la rapidité de convergence de l'algorithme. Si la position de l'optimum dans l'espace d'état est totalement inconnue, il est naturel de générer aléatoirement les individus de la population en faisant des tirages au hasard selon une distribution uniforme. Si des informations a priori sur le problème sont disponibles, il est naturel de générer les individus dans un sous domaine particulier afin d'accélérer la convergence.

La taille de la population doit être choisie de façon à réaliser un bon compromis entre temps de calcul et qualité du résultat. En effet, une taille de population trop grande augmente le temps de calcul et nécessite un espace mémoire considérable, alors qu'une taille de population trop petite conduit à l'obtention d'un optimum local **[14]**.

II.2.5 Fonction d'évaluation (fitness)

L'opérateur d'évaluation n'est pas anodin. Il est utilisé par l'opérateur de sélection pour faire son choix des individus à conserver. Ainsi, pour mesurer les performances de chaque individu qui correspond à une solution donnée du problème à résoudre, on introduit une fonction d'évaluation.

Elle permet de quantifier la capacité d'un individu à survivre en lui affectant un poids couramment appelé fitness. La force de chaque chromosome de la population est calculée afin que les plus forts soient retenus (étape de sélection) puis modifiés (croisement et mutation). La complexité de la fonction d'évaluation dépend essentiellement du problème et de ses contraintes.

Ces deux derniers éléments, codage et évaluation, sont les seuls éléments spécifiques au problème à résoudre. Une fois qu'ils sont fixés, l'algorithme génétique que l'on appliquer sera toujours le même **[15].**

II.2.6 Les opérateurs

a) L'opérateur de sélection

L'opérateur de sélection est chargé de favoriser les meilleurs individus de la population courante pour générer une nouvelle population.

Il existe plusieurs méthodes pour sélectionner ces individus on cite :

- **La loterie biaisée ou roulette wheel**

Elle consiste à créer une roue de loterie biaisée pour laquelle chaque individu de la population occupe une section de la roue proportionnelle à sa valeur d'évaluation. Ainsi, même les individus les plus faibles ont une chance de survivre. Si la population d'individus est de taille égale à *N*, alors la probabilité avec laquelle l'individu *xi* sera réintroduit dans la nouvelle population est notée $p(x_i)$et est égale à :

$$\mathbf{p(x_i) = \frac{f(x_i)}{\sum_{K=1}^{N} f(x_k)}} \qquad \textbf{(II.1)}$$

Pour un problème de minimisation, on utilise une probabilité de sélection pour un individu *xi* notée $p'(x_i)$ égale à :

$$\mathbf{p'(x_i) = \frac{1-p(x_i)}{N-1}} \qquad \textbf{(II.2)}$$

- **La sélection par tournois**

Le principe de cette méthode consiste à choisir une sous population, de taille *M* fixée a priori par l'utilisateur ou bien aléatoirement. L'individu de meilleure qualité par rapport à la sous-population sera sélectionné pour l'application des opérateurs de croisement et mutation. Cette méthode donne plus de chance aux individus de mauvaise qualité de participer à l'amélioration de la qualité de la population **[16]**.

- **La sélection par rang**

La sélection par rang trie d'abord la population par fitness de 1 à *N*. Ainsi le plus mauvais chromosome aura le rang 1, le meilleur chromosome aura le rang *N*. La sélection d'un chromosome est la même que par "roulette", mais les proportions sont en relation avec le rang plutôt qu'avec la valeur de l'évaluation **[16]**.

- **Elitisme**

L'élitisme consiste à conserver à chaque génération un certain nombre des meilleurs chromosomes de la population qui pourraient disparaître par les opérations de mutation, croisement ou sélection. Elle consiste à copier un ou plusieurs des meilleurs chromosomes dans la nouvelle génération. Ensuite, on génère le reste de la population selon l'algorithme de reproduction usuel. Cette méthode améliore considérablement les algorithmes génétiques, car elle permet de ne pas perdre les meilleures solutions **[16]**.

b) L'opérateur de croisement

La naissance d'un nouvel individu nécessite la prise aléatoire d'une partie des gènes de chacun des deux parents. Il s'agit d'un processus essentiel pour explorer l'espace des solutions possibles. Une fois la sélection terminée, les individus sont aléatoirement répartis en couples. Les chromosomes parents sont alors copiés et recombinés afin de produire chacun deux descendants ayant des caractéristiques issues des deux parents.

On peut noter que le nombre de points de croisements ainsi que la probabilité de croisement P_c permettent de décider si les parents seront croisés entre eux ou s'ils seront tout simplement recopiés dans la population suivante.

Il existe plusieurs types de croisement selon le nombre de points de croisement, on cite :

- **croisement n-point**

Ce type de croisement consiste à choisir n-points de coupures (n=1, 2,…), puis échanger les fragments de gènes délimités par les points de coupure choisis. On distingue dans ce type deux autres types de croisement qui sont très utilisés : le croisement 1-point et le croisement 2-points **[17]**.

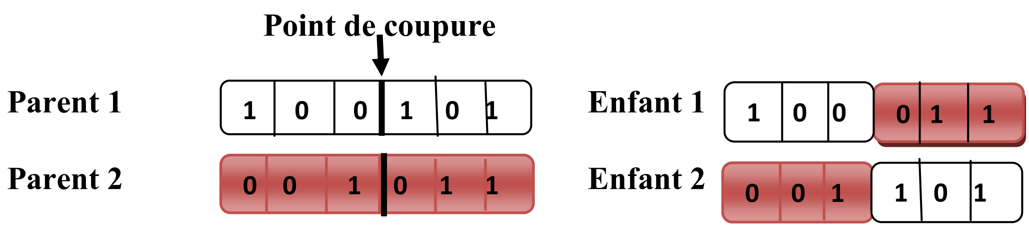


Figure II.6 Croisement 1-point.

- **croisement uniforme**

Ce type de croisement est fondé sur la probabilité. En fait, il permet la génération d'un enfant en échangeant chaque gène des deux parents avec une probabilité égale à 0.5, comme le montre la figure II.7 **[18]**.

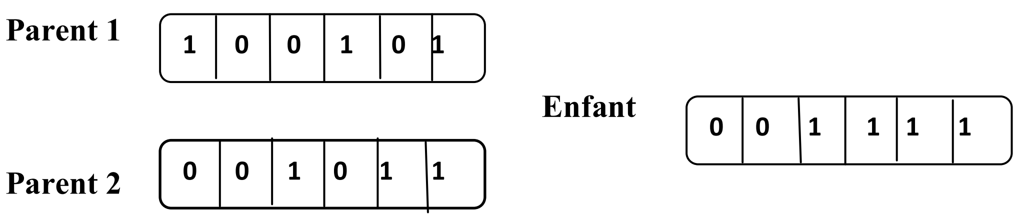


Figure II.7 Croisement uniforme.

c) L'opérateur de mutation

La mutation est un changement aléatoire occasionnel avec une faible probabilité p_m (fixée par l'utilisateur) de la valeur d'un ou plusieurs gènes d'un chromosome. En général, la mutation permet de garder une diversité dans l'évolution des individus et d'éviter les optimums locaux.

Plusieurs opérateurs de mutation existent, les plus utilisés sont **[19]**.

- **Mutation 1-point**

Cette mutation consiste en la transformation aléatoire d'une seule valeur d'un chromosome. La figure II. 8présente un exemple d'une mutation un point **[19]**.

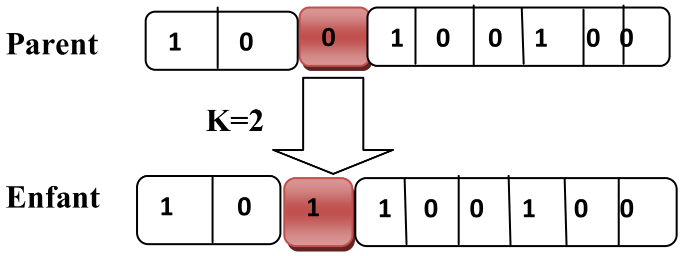


Figure II.8 Exemple d'une mutation uni-point

- **Mutation 2-points**

Cette mutation se fait par altération de plusieurs valeurs sur le chromosome **[19]**.

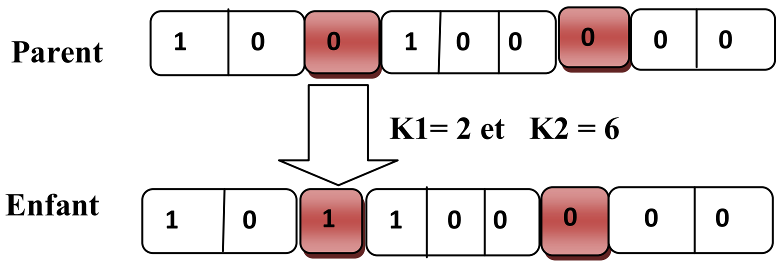


Figure II.9 Exemple d'une mutation bipoints

d) Remplacement

Les nouveaux individus obtenus après l'application des opérateurs de croisement et de mutation, sont réintroduits dans la population en fonction de la valeur de fitness associé aux enfants. Donc la nouvelle population est constituée d'individus sélectionnés pour continuer à participer son amélioration. Une méthode d'insertion est utilisée.

On trouve essentiellement 2 méthodes de remplacement différentes :

- **Remplacement stationnaire**

Un remplacement automatique des parents par les enfants sans prendre en considération leurs performances respectives.

- **Remplacement élitiste**

Le facteur performance est pris en compte dans cette stratégie qui consiste à garder au moins l'individu le plus performant lors du passage d'une génération à la suivante.

f) Convergence de l'algorithme

L'amélioration de la population est très rapide au début (recherche globale) et devient de plus en plus lente à mesure que le temps passe (recherche locale). Le bruit dans la moyenne est essentiellement dû aux mutations.

g) Critères d'arrêt

Généralement, le cycle de génération et remplacement est répétée jusqu'à ce qu'un critère d'arrêt soit satisfait. Ce critère peut être notamment un nombre fixe d'itérations (générations), un temps maximal de calcul, ou/et une solution satisfaisante. L'algorithme évolutionnaire retourne alors la (ou les) meilleure(s) solution(s) qu'il a identifiée(s) de génération en génération **[20].**

II.2.7 Algorithmes génétiques non élitistes et élitistes

Dans le domaine de l'optimisation multi-objectif, l'élitisme concerne l'utilisation d'une population externe (Population secondaire) pour maintenir les individus non dominés trouvés durant le processus d'optimisation.

Ces algorithmes sont des algorithmes évolutionnaires se basant sur la sélection d'un certain nombre d'individus pour évoluer la population, parfois la sélection se base sur les meilleurs individus, on parle des algorithmes élitistes ou de mauvais individus appelés des algorithmes non élitistes.

II.2.7.1 Algorithmes génétiques non Elitistes

Ces approches qui ne conservent pas les individus Pareto-optimaux trouvés au cours du temps, ce qui rend la maintenance de la diversité sur la frontière de Pareto difficile. La convergence des solutions vers la frontière de Pareto est lente pour ces algorithmes **[21].**

a) Non dominate Sorting Genetic Algorithm (NSGA)

Dans la méthode proposée par, le calcul de la de la fitness s'effectue en séparant la population en plusieurs groupes en fonction du degré de domination au sens de Pareto de chaque individu **[21]**.

a.1) Principe de l'Algorithme de (NSGA)

- Dans la population entière, on recherche les individus non dominés. Ces derniers constituent la première frontière de Pareto.
  On leur attribue une fitness factice. Cette valeur est supposée donner une chance égale de reproduction à tous ces individus. Mais pour maintenir la diversité dans la population, il est nécessaire d'appliquer une fonction des haring sur cette valeur.
- Ce premier groupe d'individus est supprimé de la population

- On recommence cette procédure pour déterminer la seconde frontière de Pareto. La valeur factice de fitness sattribuée à ce second groupe est inférieure à la plus petite fitness après application de la fonction de sharing sur le premier groupe. Ce mécanisme est répété jusqu'à ce qu'on ait traité tous les individus de la population.
- L'algorithme se déroule ensuite comme un algorithme génétique classique. Srinivas utilise une sélection basée sur le reste stochastique. Mais sa méthode peut être utilisée avec d'autres heuristiques de sélections (tournoi, roulette, etc.).

b) Multiple Objective Génétique Algorithme (M.O.G.A)

En 1993 Fonseca et Flemming ont proposé une méthode dans laquelle chaque individu de la population est rangé en fonction du nombre d'individus qui le dominent **[22].**Ensuite, ils utilisent une fonction de notation permettant de prendre en compte le rang de l'individu et les nombre d'individus ayant même rang **[22].**

b.1) Principe de L'Algorithme (M.O.G.A)

- Initialisation de la population.
- Evaluation des fonctions objectifs.
- Assignation d'un rang basé sur la dominance.
- Assignation d'une efficacité à partir du rang.

II.2.7.2 Algorithmes génétiques élitistes

Dans le domaine de l'optimisation multi-objectif, la sélection élitiste consiste à maintenir une seconde population appelée archive, contenant les solutions non dominées trouvées au cours des différentes générations de l'algorithme évolutionniste. Les individus de cette population participe avec une certaine probabilité à l'étape de sélection et donc à la reproduction de nouveaux individus. En outre des techniques de regroupements ou clustering sont employées pour limiter la taille de cette archive. Cette technique est utilisée pour résoudre les difficultés des méthodes non-élitistes.

a) Strength Pareto Evolutionary Algorithm (SPEA)

En 1998 Zitzler et Thiele proposent la méthode élitiste SPEA basée sur le concept Pareto. Pour réaliser cet élitisme, SPEA maintient une archive externe contenant le meilleur front de compromis rencontré durant la recherche. Toutes les solutions de l'archive participe à la sélection. Une méthode de clustering est utilisée pour réduire l'ensemble de Pareto sans supprimer ses caractéristiques. Une nouvelle méthode de niche, basée sur Pareto, est utilisée afin de préserver la diversité. L'avantage essentiel est qu'elle n'exige pas de réglage de paramètre des haring **[23].**

a.1) Principe de L'Algorithme SPEA

- Initialiser la population P_0 et créer l'archive externe vide $P^* = \emptyset$.
- Mise à jour de P à partir des individus non dominés de P_0.
- Tant que critère d'arrêt non rencontré faire
- Calcul de la valeur d'adaptation pour tous les individus de $P + P^*$.
- Sélection dans $P_t + P^*$ en fonction de la valeur d'adaptation.
- Croisement.
- Mutation.
- Mise à jour de P^* à partir des individus non dominés de P.

b) Pareto Envelo pebased Selection Algorithm (PESA)

Propose également par Corne et Knowless .Elle reprend approximativement le principe de crowding de PAES et définit un paramètre appelé squeeze_factor qui représente la mesure d'encombrement d'une zone de l'espace. Alors que PAES est basée sur une stratégie d'évolution, PESA est une méthode basée sur les algorithmes génétiques **[24].**

b.1) Principe de l'Algorithme PESA

- Génération aléatoire et évaluation de la populationP_i, et initialisation de $P_E = \emptyset$
- Transfert de tous les individus non dominés de P_I dansP_E.
- Si le critère d'arrêt est atteint
- Alors on retourne PE comme ensemble de solution
- Sinon on suprime tous les individus de P_I et on recrée P_I de la façon suivante :
- On sélectionne 2 parents dans P_E avec la probabilitéP_C,
- On produit un enfant par croisement puis on le mute.
- Avec la probabilité$(1 - P_C)$, on sélectionne un parent et on le mute pour produire un autre enfant.
- On recommence à la 2ème ligne.

c) Non dominate Sorting Genetic Algorithm II (NSGAII)

L'Algorithme Génétique multi-objectif utilisé dans notre travail est le NSGA2 (Non-dominated Sorting Genetic Algorithm 2) présenté et amélioré par Deb et al. **[25]**. C'est I 'un des algorithmes les plus utilisés et les plus cités dans la littérature. Il est très utilisé par plusieurs auteurs non seulement dans le cadre de l'optimisation multi-objectif mais aussi pour comparaison avec d'autres algorithmes.

c.1) Principe de l'algorithme NSGA2

- Deb et al. ont proposé une nouvelle version de l'algorithme NSGA, le NSGA II qui considérée comme étant plus efficace que son prédécesseur car :

- Il utilise une approche élitiste qui permet de sauvegarder les meilleures solutions trouvées lors des générations précédentes.
- Il utilise une procédure de tri basée sur la non-dominance.
- Il ne nécessite aucun réglage de paramètres.
- Il utilise un opérateur de comparaison basé sur le calcul de la distance de Crowding.

d) Strength Pareto Evolutionary Algorithm 2 (SPEA2)

La première version de l’algorithme SPEA utilise une population régulière et une archive. Les étapes suivantes sont ectuées paritération en commençant par une population initiale et une archive vide Premièrement, tous les membres de la population-dominés sont copiés dans l’archive ; des individus ou les redondances (concernant les valeurs des objectifs) dominées sontretirées de l’archive lors de cette opération demise à jour. Si la taille de l’archive mise à jour dépasse une limite prédénie, d’autres membres de l’archive sont supprimés par une technique de clustering qui conserve les caractéristiques de front non-dominé **[26]**.

d.1) Principe de l'algorithme(SPEA2)

- Générer une population initiale P de taille N.
- Evaluer les individus dans P.
- Initialiser l'archive A de taille $\overline{N}$.
- Appliquer les opérateurs génétiques pour créer une population des enfants Q.
- Calculer la fonction d'adaptation F des individus dans O=P∪A.
- Mise à jour de l'archive A.
- Mise à jour de la population P.

II-3 Conclusion

Dans ce chapitre, nous avons présentes les notions de base et le principe de fonctionnement des algorithmes génétiques. L'évaluation des solution par la fonction objectif permet de classer les individus selon leurs performances, la sélection qui à travers ses différentes méthodes, imite le processus de la reproduction naturel .Puis en s'appuyant sur les informations génétiques de chaque paire des chromosomes (individus) sélectionnées, l'AG forme des nouveaux individus enfants qui ensuite seront modifiés pour former une nouvelle population différente et évoluée par rapport à la première.

Chapitre III :

Le Prbléme du sac à dos

III.1 Introduction

Dans ce chapitre, nous présentons le problème de sac à dos dans sa version la plus simple. Nous étendrons sur quelques-unes de ses nombreuses variantes, telles que le sac à dos multidimensionnel et le problème de set picking. Nous détaillerons en particulier la variante du sac à dos multi-objectif unidimensionnel en variables binaires qui sera le problème sur lequel le reste de cette thèse s'appuiera.

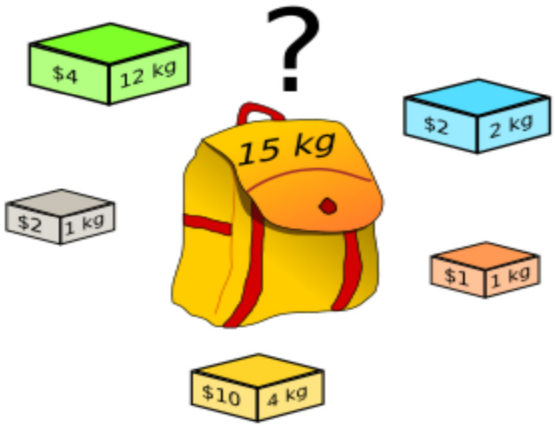


Figure III.1 Problème du sac à dos.

III.2 Historique

Ce problème fait partie des 21 problèmes NP-complets identifiés par Richard Karpen 1972. Ces 21 problèmes sont réputés comme les problèmes les plus difficiles en optimisation combinatoire. Un grand nombre d'autres problèmes NP-complets peuvent se ramené à ces 21 problèmes de base. Nous pouvons retrouver le problème du sac à dos dans de nombreux domaines :

- en cryptographie, où il fut à l'origine du premier algorithme de chiffrement asymétrique en 1976.
- dans les systèmes financiers, où l'idée est la suivante : étant donné un certain montant d'investissement dans des projets, quels projets choisir pour que le tout rapporte le plus d'argent possible.
- pour la découpe de matériaux, afin de minimiser les pertes dues aux chutes.
- dans le chargement de cargaisons (avions, camions, bateaux ...).
- ou encore, dès qu'il s'agit de préparer une valise ou un sac à dos.

III.3 Problème de sac à dos

Le problème de sac à dos est un problème classique d'optimisation combinatoire appartenant à la classe des problèmes NP-complets. L'énoncé de ce problème est simple : étant donné un ensemble de n objets, où chaque objet i est caractérisé par un poids w_i et un profit ci on cherche le sous-ensemble d'objets à charger dans un sac de capacité w afin de maximiser la somme des profits. Dans sa forme la plus simple, dénomée ≪sac à dos unidimensionnel en variables binaires≫, le problème se formule ainsi :

$$\textbf{(01-KP)}\begin{cases}\mathbf{Max\ z(x)} = \sum_{i=1}^{n} c_i x_i \ , c_i \in N^* \\ s.c: \sum_{i=1}^{n} w_i x_i \leq w, w_i \ \in N^* \\ x_i \ \in \{0,1\} \quad \forall\, i \in \{1,\dots,n\}\end{cases} \qquad \textbf{(III.1)}$$

Les poids w_i et les profits c_i ainsi que la capacité w sont des entiers positifs i ∈ $\{1, \dots n\}$.
La variable $x_i$est la variable de décision ; elle prend la valeur 1 si l'objet
i est chargé dans le sac, sinon elle prend la valeur 0 **[27]**.

III.4 Quelques variantes du problème de sac à dos

De nombreuses variantes du problème de sac à dos ont été étudiés en raison de différents intérêts pratiques ou théoriques qu'on peut trouver dans la littérature, par exemple : en économie comme un sous-problème de décision, en conception ou attaque des crypto-systèmes, problème de référence pour le test des algorithmes de résolution des problèmes de classe NP-difficile, etc. Ces variantes peuvent être catégorisées selon :

- Le domaine des variables : sac à dos à variables continues, entières, bornées, bivalentes.
- Structure de l'instance : problème de la somme de sous-ensembles.
- Le nombre de contraintes : sac à dos unidimensionnel, multidimensionnel.
- Le nombre de profits associés à chaque objet ou le nombre de fonctions objectifs à Optimiser : sac à dos mono-objectif, bi-objectifs, multi-objectifs.
- Le nombre de sacs : sac à dos multiple, etc.

Du fait de la quantité des paramètres intervenant dans la formulation, les variantes sont nombreuses. Dans cette section nous présentons quelques variantes.

III.4.1 Problème de la somme de sous-ensembles (SSP)

Étant donné un ensemble de n objets de poids w_j ;$\forall j \in \{1, \dots, n\}$ et un sac à dos de capacité maximale W, on veut sélectionner un sous-ensemble d'objets dont le poids total se rapproche au maximum de W.

$$(SSP)\begin{cases} \qquad max\ Z(x) = \ \sum_{j=1}^{n} w_j \,.\, x_j \\ tq \qquad \sum_{j=1}^{n} w_j . x_j \leq W \\ \quad x_j \in \{0,1\}, \ \forall j \in \{1..n\}\end{cases} \qquad \textbf{(III.2)}$$

Avec le choix de valeurs des variables de décision est interprété comme suit :

$$\mathbf{x_j} = \begin{cases} \mathbf{1\ si\ l'objetj\ est\ pris} \\ \mathbf{0 \qquad\qquad\qquad sinon} \end{cases} \qquad \textbf{(III.3)}$$

Ce problème est lié à la résolution de l'équation suivante :

$$\begin{cases} \sum_{j=1}^{n} \mathbf{w_j}.\mathbf{x_j} \leq \widetilde{\mathbf{W}} \\ \mathbf{x_j} \in \{\mathbf{0}, \mathbf{1}\},\ \forall \mathbf{j} \in \{\mathbf{1}..\mathbf{n}\} \end{cases} \qquad \textbf{(III.4)}$$

Avec $\widetilde{W} \leq W$. Soit $L_{\widetilde{w}}$ l'ensemble de solution de l'équation ci-dessus.

Soit $Z^* = \left\{\max_{\widetilde{W} \leq W} \widetilde{W} | L_{\widetilde{w}} \neq \emptyset\right\}$ l'évaluation optimale du SSP. Dans le cas où cannait Z^*, on peut écrire le problème de la somme de sous-ensemble somme suit :

$$(\mathbf{SSP}) = \begin{cases} \mathbf{Min(Z)} = \left|\mathbf{Z}^* - \sum_{j=1}^{n} \mathbf{w_j x_j}\right| \\ \mathbf{x_j} \in \{\mathbf{0}, \mathbf{1}\},\ \forall \mathbf{j} \in \{\mathbf{1}..\mathbf{n}\} \end{cases} \qquad \textbf{(III.5)}$$

III.4.2 Problème de sac à dos multiple (m-KP)

On considère un ensemble à n objets à charger dans m sacs à dos de capacitéW_i , $i \in \{1, \ldots, m\}$. Chaque article j est caractérisé par son poids w_j son profit c_j et sa variable de décision :

$$\mathbf{x_{ij}} = \begin{cases} \mathbf{1\ si\ l'objet\ j\ est\ chargé\ dans\ le\ sac\ i} \\ \mathbf{0 \qquad\qquad\qquad sinon} \end{cases} \qquad \textbf{(III.6)}$$

Il s'agit alors de trouver m sous-ensembles disjoints de $\{1, \ldots, n\}$, où chaque sous-ensemble correspond au contenu d'un sac, et qui maximisent le profit total formé par la somme des articles sélectionnés **[28]**. Le problème m-KP s'écrit de la manière suivante :

$$(\mathbf{m-KP}) \begin{cases} \max \mathbf{Z(x)} = \sum_{i=1}^{n} \sum_{j=1}^{m} \mathbf{c_j x_{ij}} \\ \sum_{j=1}^{n} \mathbf{w_j x_{ij}} \leq \mathbf{W_i};\ \forall \mathbf{i} = \{\mathbf{1}, \ldots, \mathbf{m}\} \\ \sum_{i=1}^{m} \mathbf{x_{ij}} \leq \mathbf{1};\ \forall \mathbf{j} = \{\mathbf{1}, \ldots, \mathbf{n}\} \\ \mathbf{x_{ij}} \in \{\mathbf{0}, \mathbf{1}\},\ \forall \mathbf{i} \in \{\mathbf{1}, \ldots, \mathbf{n}\}\ \mathbf{et}\ \forall \mathbf{j} \in \{\mathbf{1}..\mathbf{m}\} \end{cases} \qquad \textbf{(III.7)}$$

III.4.3 Problème de sac à dos multidimensionnel (MKP)

Le problème de sac à dos multidimensionnel (MKP), est une variante du problème ayant plusieurs contraintes de capacité. On considère dans ce cas m caractéristiques différentes d'un même sac à dos : volume, poids, etc. Soitw_{ij}la valeur du $i^{ème}$ l'effet engendré par l'objet j sur le sac à dos, et W_i la $i^{ème}$ capacité du sac. Pour cette variante, on doit respecter les m contraintes de capacité **[29]**.

Ainsi, le MKP peut être formulé comme suit :

$$(\boldsymbol{MKP}) \begin{cases} \max \boldsymbol{Z(x)} = \sum_{j=1}^{n} \boldsymbol{c_j x_j} \\ \sum_{j=1}^{n} \boldsymbol{w_{ij} x_j} \leq \boldsymbol{W_i},\quad \forall \boldsymbol{i} \in \{\mathbf{1}, \ldots, \boldsymbol{m}\} \\ \boldsymbol{x_j} \in \{\mathbf{0}, \mathbf{1}\},\ \forall \boldsymbol{j} \in \{\mathbf{1}, \ldots, \boldsymbol{n}\} \end{cases} \qquad \textbf{(III.8)}$$

III.4.4 Problème de sac à dos multi-objectif multidimensionnel (MOMKP)

Ce problème est une combinaison de l'extension multi-objective du problème de sac à dos, i.e., plusieurs fonctions à optimiser, et sa version multidimensionnelle. Le problème MOMKP peut être formulé comme suit : soit n objets ayant p caractéristiques ${w_j}^i$; $j \in \{1, \dots, n\}$ (poids, volume, etc.), et k profits ${c_j}^i j \in \{1, \dots, n\}$, on veut sélectionner des objets de manière à maximiser les différents k profits, tout en respectant les p capacités du sac à dos W_i.

$$(\boldsymbol{MOMKP}) \begin{cases} \max \boldsymbol{Z^i}(\boldsymbol{x}) = \sum_{j=1}^{n} \boldsymbol{c^i}_j \boldsymbol{x_j}, \quad \forall \boldsymbol{i} = \{\mathbf{1}, \dots, \boldsymbol{k}\} \\ \sum_{j=1}^{n} \boldsymbol{w^i}_j \boldsymbol{x_j} \leq \boldsymbol{W_i}, \quad \forall \boldsymbol{i} \in \{\mathbf{1}, \dots, \boldsymbol{p}\} \\ \boldsymbol{x_j} \in \{\mathbf{0}, \mathbf{1}\}, \ \forall \boldsymbol{j} \in \{\mathbf{1}, \dots, \boldsymbol{n}\} \end{cases} \qquad \textbf{(III.9)}$$

Où n est le nombre d'objets, x_j désigne la $j^{ème}$ variable de décision binaire, k représente le nombre d'objectifs, Z^i représente la $i^{ème}$ fonction objectif.

III.5 Conclusion

De nombreux problèmes rencontrés dans la vie quotidienne peuvent être formulés comme des problèmes d'optimisation combinatoire. Dans ce chapitre, nous avons introduit le problème de sac-à-dos et ses variantes.

Chapitre IV :

Conception Et Réalisation

IV.1 Introduction

Nous aboutissons maintenant à l'étape finale, à savoir de choisir un algorithme SPEA-2 pour résolution de notre problème (le problème de sac à dos multi-objectif). Afin de mettre en application la méthode de résolution proposée, nous allons présenter une application concrète à la case de l'utilisation des données du sac à dos multi-objectif.

IV.2 Modélisation du problème

La résolution du problème du sac à dos est souvent difficile. Dans le cas où plusieurs critères sont à optimiser, la problématique se complique davantage car ces critères sont souvent contradictoires. Plusieurs notions d'optimalité ont été utilisées dans la littérature. Mais la plus fréquente est la Pareto optimalité où il s'agit de déterminer l'ensemble des solutions efficaces. Le problème est défini comme suit :

$$(\mathbf{MOKP})\begin{cases}\mathbf{max f^k(x)} = \sum_{\mathbf{j=1}}^{\mathbf{n}} \mathbf{c_j}^{\mathbf{k}}\mathbf{x_j}\ ;\ \mathbf{k = 1, 2, \dots, p} \\ \sum_{\mathbf{j=1}}^{\mathbf{n}} \mathbf{w_j x_j} \leq \mathbf{W} \\ \mathbf{x_j} \in \{\mathbf{0, 1}\};\ \mathbf{j = 1, 2, \dots, n}\end{cases} \qquad \textbf{(IV.1)}$$

Avec tous les paramètres supposés être des entiers non négatifs.

IV.2.1 SPEA-2 le problème de sac à dos

SPEA-2 représente une version améliorée de l'algorithme SPEA proposé par les mêmes auteurs. SPEA-2 repose sur l'utilisation d'une archive représentée par une population externe A (tandis que la population courante est notée P), dite archive, de taille $\overline{N}$ fixé préalablement. Cette archive de taille est destinée à contenir un nombre limité de solutions non dominées trouvées par l'algorithme au cours de l'optimisation. A chaque itération, les nouveaux individus non dominés de la population P sont comparés aux membres de l'archive A en utilisant le critère de dominance. Si le nombre d'individus non dominés n'est pas suffisant, l'archive est complétée par les meilleurs individus dominés. La comparaison des individus dans SPEA-2 se fait comme dans le NSGA-2 en utilisant la relation de dominance ainsi qu'un critère de diversité.

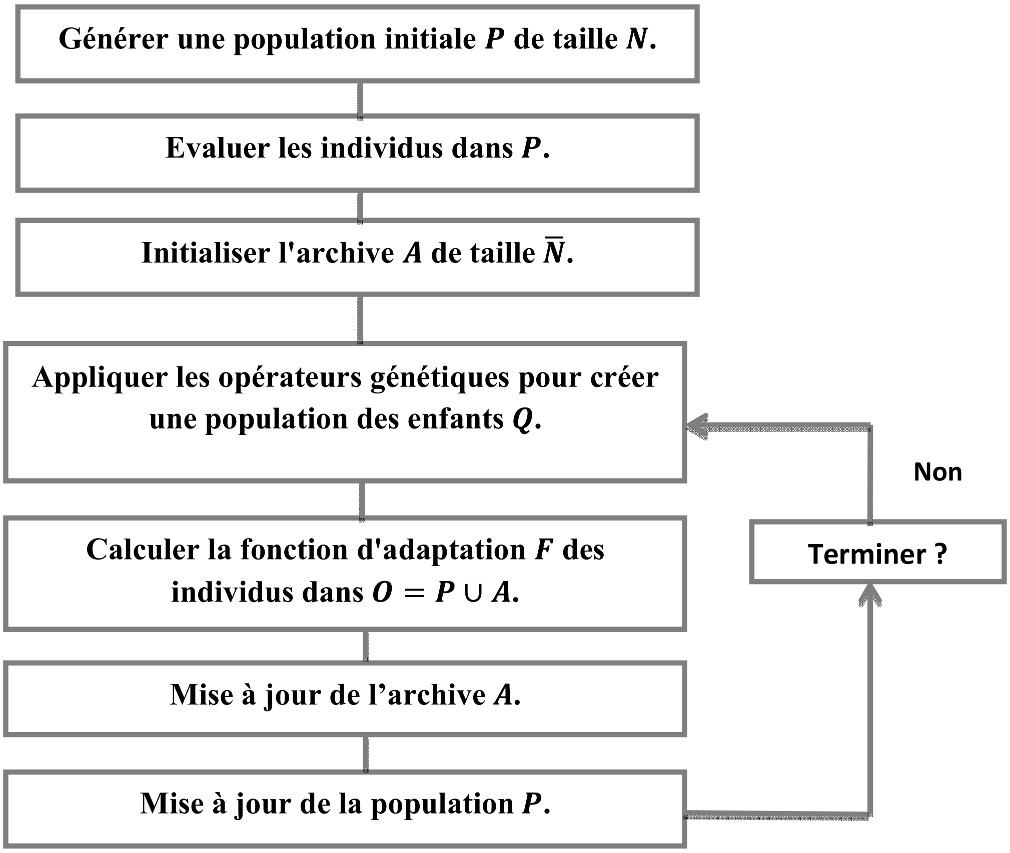


Figure.1 Organigramme de SPEA-2 pour MOKP.

Exemple :

$$(MOKP)\begin{cases} max\ f^1(x) = 7x_1 + 4x_2 + 3x_3 + 3x_4 \\ max\ f^2(x) = 2x_1 + 4x_2 + 5x_3 + 4x_4 \\ 13x_1 + 12x_2 + 8x_3 + 10x_4 \leq 30 \\ x_j \in \{0,1\}; j = 1,2,3,4. \end{cases} \qquad (IV.2)$$

a) Codage d'individu

Le meilleur choix de codage pour notre problème est le codage du type binaire.

0	**1**	**0**	**1**

Figure IV.2 Un exemple de codage de solution

b) Génération initiale

La population initiale est générée N individus aléatoirement.

Pour N=5

Tableau IV.1 Un exemple de population initiale de solution.

	Individu			
p_1	0	0	0	1
p_2	0	1	0	0
p_3	1	0	0	0
p_4	0	1	0	1
p_5	0	0	0	1

c) Évaluation

Nous évaluons les individus de la population initiale par deux fonctions objectives.

La première fonction objectif est: $f^1(x) = \sum_{i=1}^{n} C_i^{\ 1} x_i$.

La deuxième fonction objectif est : $f^2(x) = \sum_{i=1}^{n} C_i^{\ 2} x_i$.

Tableau IV.2 Un exemple d'évaluation initiale de la population de MOKP.

	Individu				f^1	f^2
p_1	0	0	0	1	3	4
p_2	0	1	0	0	4	4
p_3	1	0	0	0	7	2
p_4	0	1	0	1	7	8
p_5	0	0	0	1	3	4

d) Initialisation de la population d'archive

Créer une archive vide $A = \emptyset$.

IV.3 Les opérateurs génétiques

a) Opérateur de sélection

Cet opérateur k prendra des individus (parents) de la population initiale à utilises afin de générer la nouvelle population (enfants). Dans notre problème, nous utilisons la sélection par tournoi qui se base sur la relation de dominance.

Dans cet exemple, nous appliquons la sélection par tournoi à 5 individus pour choisir 2 individus.

Tableau IV.3 L'évaluation initiale de la population

	Individu				f^1	f^2
p_1	**0**	**0**	**0**	**1**	**3**	**4**
p_2	**0**	**1**	**0**	**0**	**4**	**4**
p_3	**1**	**0**	**0**	**0**	**7**	**2**
p_4	**0**	**1**	**0**	**1**	**7**	**8**
p_5	**0**	**0**	**1**	**1**	**6**	**9**

-Pour choisir le premier individu E_1, nous comparons l'individu p_1 et l'individu p_2 en suivant :

-Si $f^j(p_1) \geq f^j(p_2)$et$f^j(p_1) > f^j(p_2)\ \ \forall j = \{1,2\}$, alours nous choisissons p_1.

-Si $f^j(p_2) \geq f^j(p_1)$et$f^j(p_2) > f^j(p_1)\ \ \forall j = \{1,2\}$, alours nous choisissons p_2.

Dans cet exemple, $E_1 \rightarrow p_2$.

-Pour choisir le deuxième individuE_2, nous comparons l'individu p_3 et l'individu p_4 en suivant :

-Si $f^j(p_3) \geq f^j(p_4)$et$f^j(p_3) > f^j(p_4)\ \ \forall j = \{1,2\}$, alours nous choisissons p_3.

-Si $f^j(p_4) \geq f^j(p_3)$et$f^j(p_4) > f^j(p_3)\ \ \forall j = \{1,2\}$, alours nous choisissons p_4.

Dans cet exemple, $E_2 \rightarrow p_4$.

b) Opérateur de croisement

Nous appliquons un opérateur de croisement 1-point aux individus sélectionnés $\{E_1, E_2\}$ avec une probabilité de croisement $p_c = 0.8$.

Nous appliquons un opérateur de croisement1-point aux deux individus E_1etE_2.

E_1	**0**	**1**	0	0
E_2	0	1	0	1

Figure IV.3 Le croisement 1-point aux deux individus E_1etE_2.

Puis échanger les gènes délimités avec les points de coupure choisis.

Q₁	0	1	0	1
Q₂	0	1	0	0

Figure IV.4 Un changements de croisement1-point aux individus E_1etE_2

a) Opérateur de mutation

Nous appliquons un opérateur de mutation 1-point aux individus obtenus d'opérateur de croisement $\{Q_1, Q_2\}$ avec une probabilité de mutation de $p_m = 0.2$.

Nous appliquons un opérateur de mutation 1-point aux individus Q_1.

Q_1	0	1	0	1

Figure IV.5 La mutation 1-point aux individus Q_1.

Ensuite, nous changeons au hasard la valeur du gène de l'individu Q_1.

Q_1	0	1	1	1

Figure IV.6 Changement mutation 1-point aux individus Q_1.

IV.4 Fonction d'adaptation

Les classements des individus se fait selon le taux de dominance (i.e, dominance count), en attribuant à chaque individu i une valeur S_i : $S_i = |\{k | k \in P \cup A \wedge i > k\}|$.

Nous combinons deux populations P et A pour obtenir une nouvelle population $O = P \cup A$.

Tableau IV.4 Les classements des individus des valeurs S_i

	Individu				f^1	f^2	S_i
O_1	0	0	0	1	3	4	0
O_2	0	1	0	0	4	4	1
O_3	1	0	0	0	7	2	0
O_4	0	1	0	1	7	8	3
O_5	0	0	1	1	6	9	2
O_6	0	1	1	1	10	13	5
O_7	1	1	0	0	11	6	3

A partir des valeurs de S_i calculées, un classement entre les individus est effectué sur la base d'une évaluation R_i calculée pour chaque individu i et définie par :

$$\mathbf{R_i = \sum_{k \in O \wedge k > i} S_k} \quad \textbf{(IV.3)}$$

Tableau IV. 5 Les classements entre les individus sont effectués sur la base d'une évaluation R_i.

	Individu				f^1	f^2	S_i	R_i
O_1	0	0	0	1	3	4	0	14
O_2	0	1	0	0	4	4	1	13
O_3	1	0	0	0	7	2	0	11
O_4	0	1	0	1	7	8	3	5
O_5	0	0	1	1	6	9	2	5
O_6	0	1	1	1	10	13	5	0
O_7	1	1	0	0	11	6	3	0

L'évaluation $R_i = 0$ correspond à un individu i non dominé, tandis que une valeur élevée de R_i correspond à un individu dominé par plusieurs individus (i.e., la qualité est inversement proportionnelle à R). Ce critère de comparaison trouve sa limite dans le cas où plusieurs individus ne dominent pas les uns les autres.

Dans ce cas, le critère de densité du voisinage prend la valeur de la distance vers le $j^{ème}$ plus proche voisin comme critère de diversité ; cela afin de départager l'ordre des individus ayant les mêmes valeurs de R (i.e., non dominés entre eux). La densité de chaque solution i est définie par : $\mathbf{DK_i} = \frac{1}{\sigma_i{}^j + 2}$ avec $\mathbf{j} = \sqrt{|\mathbf{O}|}$.

La procédure de l'obtention de la mesure de distance $\sigma_i{}^j$ ce divide en trois étapes :

1-Calculer $\mathbf{D(j)} = \mathbf{d_{ij}} = \sqrt{(\mathbf{f_i}^1 - \mathbf{f_j}^1)^2 + (\mathbf{f_i}^2 - \mathbf{f_j}^2)^2}$; $\forall j \in O$.

Tableau IV.6 la matice de distance d_{ij}.

Distance	O_1	O_2	O_3	O_4	O_5	O_6	O_7
$D(O_1)$	-	1	4.47	5.66	5.83	11.40	8.25
$D(O_2)$	1	-	3.61	5	5.39	10.82	7.28
$D(O_3)$	4.47	3.61	-	6	7.07	11.40	5.66
$D(O_4)$	5.66	5	6	-	1.41	5.83	4.47
$D(O_5)$	5.83	5.39	7.07	1.41	-	5.66	5.83
$D(O_6)$	11.40	10.82	11.40	5.83	5.66	-	7.07
$D(O_7)$	8.25	7.28	5.66	4.47	5.83	7.07	-

2-Trier $(\boldsymbol{D(j)}, <)$**:** la valeur $D(j)$ est classée par ordre croissant**.**

- Pour $j = 1$:

Tableau IV .7 : La valeur D(j)est classée par ordre croissant pour $j = 1$

Distance	$\boldsymbol{O_2}$	$\boldsymbol{O_3}$	$\boldsymbol{O_4}$	$\boldsymbol{O_5}$	$\boldsymbol{O_7}$	$\boldsymbol{O_6}$
$\boldsymbol{D(O_1)}$	**1**	**4.47**	**5.66**	**5.83**	**8.25**	**11.40**

- Pour $j = 2$:

Tableau IV. 8 La valeur D(j)est classée par ordre croissant pour $j = 2$

Distance	$\boldsymbol{O_1}$	$\boldsymbol{O_3}$	$\boldsymbol{O_4}$	$\boldsymbol{O_5}$	$\boldsymbol{O_7}$	$\boldsymbol{O_6}$
$\boldsymbol{D(O_2)}$	**1**	**3.61**	**5**	**5.39**	**7.28**	**10.82**

- Pour $j = 3$:

Tableau IV.9 La valeur D(j) est classée par ordre croissant pour $j = 3$

Distance	$\boldsymbol{O_2}$	$\boldsymbol{O_1}$	$\boldsymbol{O_7}$	$\boldsymbol{O_4}$	$\boldsymbol{O_5}$	$\boldsymbol{O_6}$
$\boldsymbol{D(O_3)}$	**3.61**	**4.47**	**5.66**	**6**	**7.07**	**11.40**

- Pour $j = 4$:

Tableau IV.10 La valeur D(j) est classée par ordre croissant pour $j = 4$

Distance	$\boldsymbol{O_5}$	$\boldsymbol{O_7}$	$\boldsymbol{O_2}$	$\boldsymbol{O_1}$	$\boldsymbol{O_6}$	$\boldsymbol{O_3}$
$\boldsymbol{D(O_4)}$	**1.41**	**4.47**	**5**	**5.66**	**5.83**	**6**

- Pour $j = 5$

Tableau IV.11 La valeur D(x) est classée par ordre croissant pour$j = 5$

Distance	$\boldsymbol{O_4}$	$\boldsymbol{O_2}$	$\boldsymbol{O_6}$	$\boldsymbol{O_1}$	$\boldsymbol{O_7}$	$\boldsymbol{O_3}$
$\boldsymbol{D(O_5)}$	**1.41**	**5.39**	**5.66**	**5.83**	**5.83**	**7.07**

- Pour $j = 6$:

Tableau IV.12 La valeur D(j)est classée par ordre croissant pour j =6

Distance	$\boldsymbol{O_5}$	$\boldsymbol{O_4}$	$\boldsymbol{O_7}$	$\boldsymbol{O_2}$	$\boldsymbol{O_3}$	$\boldsymbol{O_1}$
$\boldsymbol{D(O_6)}$	**5.66**	**5.83**	**7.07**	**10.82**	**11.40**	**11.40**

- Pour $j = 7$:

Tableau IV.13 La valeur D(j) est classée par ordre croissant pour $j = 7$

Distance	$\boldsymbol{O_4}$	$\boldsymbol{O_3}$	$\boldsymbol{O_5}$	$\boldsymbol{O_6}$	$\boldsymbol{O_2}$	$\boldsymbol{O_1}$
$\boldsymbol{D(O_7)}$	**4.47**	**5.66**	**5.83**	**7.07**	**7.28**	**8.25**

3-Initialiser $\boldsymbol{\sigma_i{}^k = D(k)}$où $\boldsymbol{k = \sqrt{|O|}}$

$\boldsymbol{|O| = 7}$;$\boldsymbol{k = \sqrt{7} = 2.}$

Tableau IV.14 les valeurs de σ_i^k=D(k) Pour j =1…7, et k=2

$\boldsymbol{\sigma_1{}^2}$	**Distance(1, 2) = 4.47**
$\boldsymbol{\sigma_2{}^2}$	**Distance(2, 2) = 3.61**
$\boldsymbol{\sigma_3{}^2}$	**Distance(3, 2) =4.47**
$\boldsymbol{\sigma_4{}^2}$	**Distance(4, 2) = 4.47**
$\boldsymbol{\sigma_5{}^2}$	**Distance(5, 2) = 5.39**
$\boldsymbol{\sigma_6{}^2}$	**Distance(6, 2) = 5.83**
$\boldsymbol{\sigma_7{}^2}$	**Distance(7, 2) = 5.66**

Finalement, la fonction d'adaptation F_i attribuée à chaque individu i est donnée par :

$$\boldsymbol{F_i = R_i + DK_i}$$

Tableau IV.15 Les valeurs de la fonction d'adaptation F_I.

	$\boldsymbol{R_i}$	$\boldsymbol{DK_i}$	$\boldsymbol{F_i}$
$\boldsymbol{O_1}$	**14**	**0.15**	**14.15**
$\boldsymbol{O_2}$	**13**	**0.18**	**13.18**
$\boldsymbol{O_3}$	**11**	**0.15**	**11.15**
$\boldsymbol{O_4}$	**5**	**0.15**	**5.15**
$\boldsymbol{O_5}$	**5**	**0.14**	**5.14**
$\boldsymbol{O_6}$	**0**	**0.13**	**0.13**
$\boldsymbol{O_7}$	**0**	**0.13**	**0.13**

IV.5 Mise à jour de l'archive A

Compléter A avec les meilleur individus de O selon la fonction d'adaptation F :

$A \leftarrow A \cup \{x \in O |\ F_x < 1\}$.

Tableau IV.16 Le meilleur individu de O selon la fonction d'adaptation.

	Individu				f^1	f^2	F_i
A_1	0	1	1	1	10	13	0.13
A_2	1	1	0	0	11	6	0.13

Ensuite, les individus dominés sont ordonnés selon leurs évaluations F. Néanmoins, dans le cas où$A > \overline{N}$, la comparaison entre deux i et j est faite selon la relation d'ordre définie comme suit :

$$i \leq_d j \Leftrightarrow \forall k \in \{1, \dots, |A|\}, = {\sigma_j}^k \vee \exists k\{1, \dots, |A|\}\left[\left(\forall l \in \{1, \dots, |A|\}, {\sigma_i}^l = {\sigma_j}^l\right) \wedge {\sigma_i}^k < {\sigma_j}^k\right]$$

Si $\overline{N} = 1$, alors$|A| = 2 > 1$.

Pour A_1 **:**

Tableau IV.17 Mise à jour de l'archive A_1**.**

	A_1	A_2
${\sigma_i}^1$	-	7.07

Pour A_2 :

Tableau IV.18 Mise à jour de l'archiveA_2.

	A_1	A_2
${\sigma_i}^2$	7.07	-

Donc $A \leftarrow A \backslash \{i \in A | \forall j \in A\ \ i \leq_d j\}$.

Tableau IV.19 Le meilleur Individu de Mise à jour de l'archive.

	Individu				f^1	f^2
A_1	0	1	1	1	10	13

IV.6 Mise à jour de la population P

Un remplacement automatique des parents par les enfants sans prendre en considération leurs performances respectives

Nous remplaçons l'individu parent p_2 par individu enfant Q_1 and l'individu parent p_4 par individu enfantQ_2**.**

Tableau IV.20 Le remplaçon l'individu par à rapport à individu enfant.

	Individu				f^1	f^2
p_1	0	0	0	1	3	4
p_2	0	1	1	1	10	13
p_3	1	0	0	0	7	2
p_4	1	1	0	0	11	6
p_5	0	0	1	1	6	9

IV.7 Implémentation de l'algorithme SPEA-2 pour MOKP

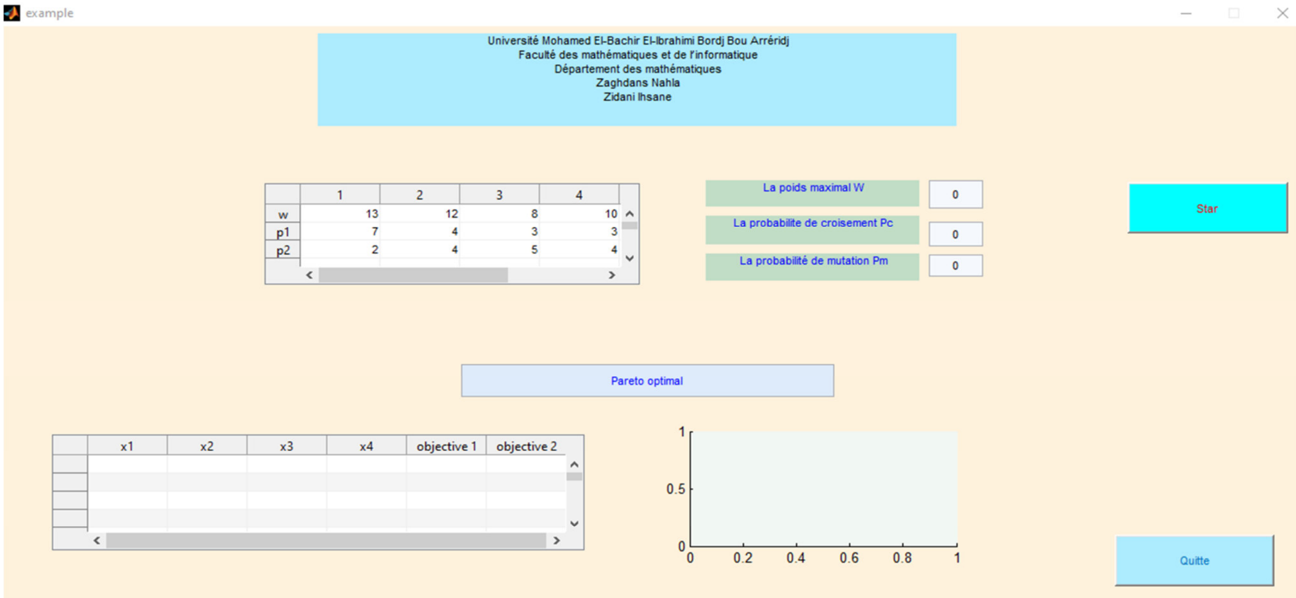
IV.7.1 Environnement du logiciel

MATLAB est un langage de programmation et un environnement d'analyse numérique de quatrième génération qui est utilisé pour les calculs matriciels, le développement et l'exploitation d'algorithmes, la création d'interfaces. Il a été utilisé dans de nombreux domaines par des ingénieurs et des scientifiques. Ce logiciel est commercialisé par la société Math Works .Il a été développe par livermolar, professeur d'informatique à l'Université du Nouveau- Mexique, dans les années soixante-dix. Les applications MATLAB se retrouvent dans de nombreuses disciplines et constituent un outil numérique puissant pour la modélisation de systèmes physiques, simuler des modèles mathématiques, concevoir et valider des applications.

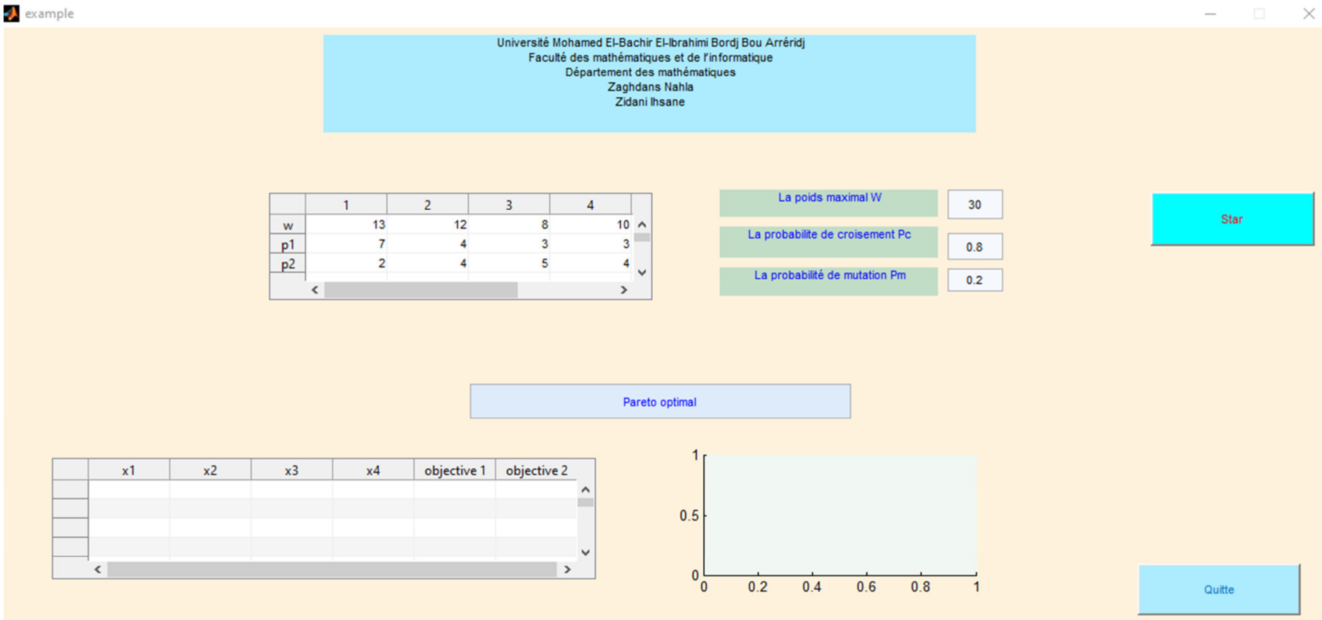
IV.7.2 Présentation de l'application

L'application est construite d'une façon dynamique, elle permet le rajout et la suppression des données qui sont aux guis de l'utilisateur selon ses préférences, ce qui met l'application plus roche de la réalité.

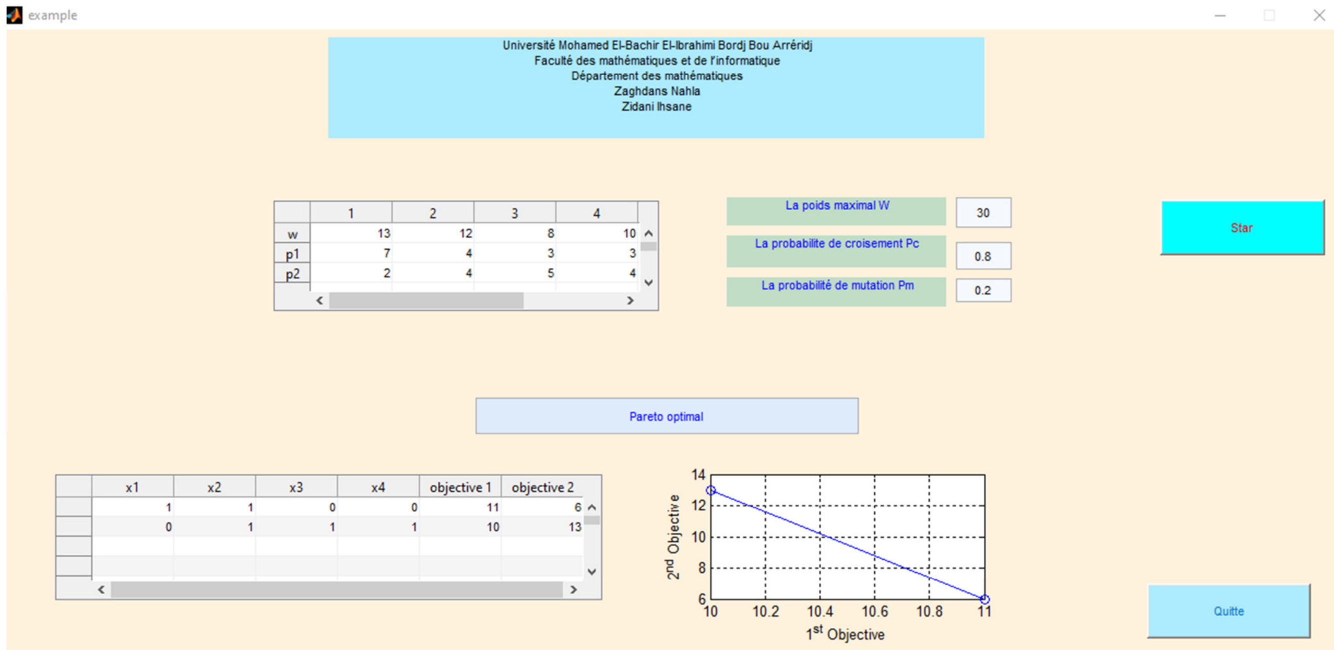
Au lancement de notre application, la fenêtre principale est illustrée dans la figure suivante



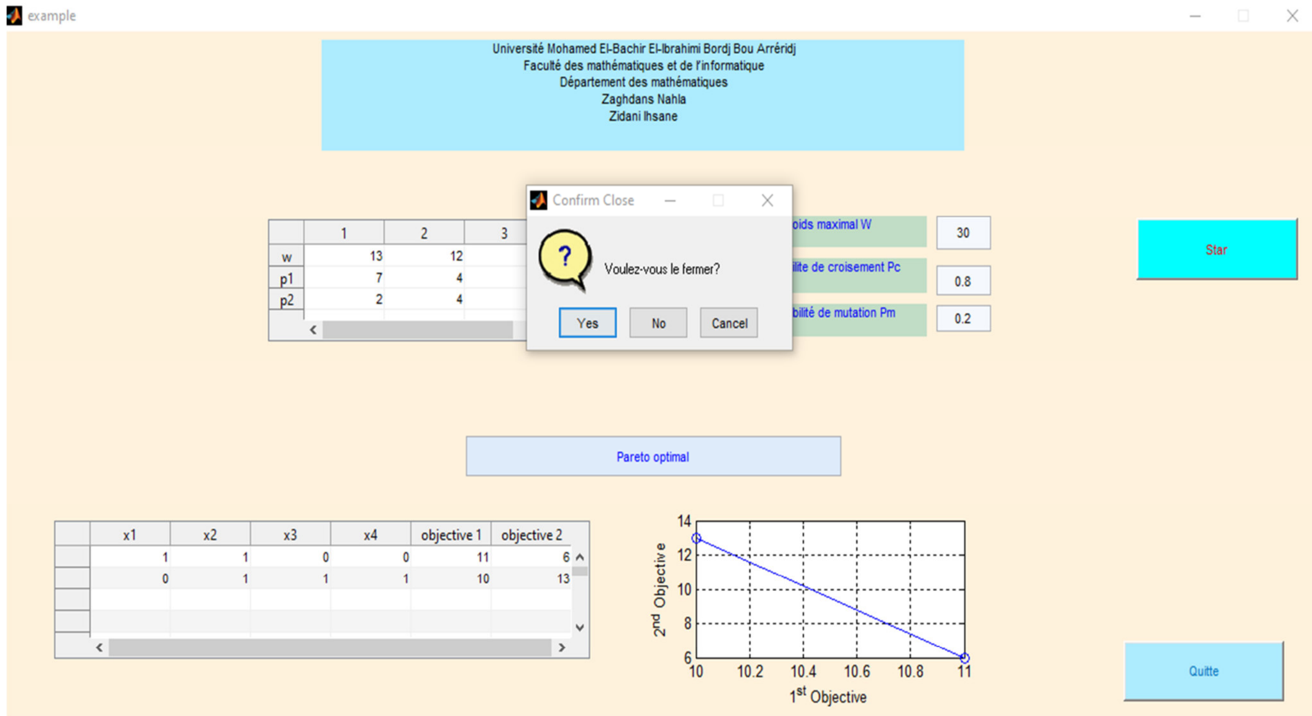
Dans cette fenêtre, nous entrons le poids maximal W, la probabilité de croisement P_c, et la probabilité de mutation P_m du problème que nous voulons traiter.



Une fois les données saisies, la résolution s'entame en cliquant sur « Start » pour afficher le tableau pour l'archive, ainsi que la figure représentant le Pareto optimal.



A la fin, l'application se ferme en cliquant sur le bouton "Quitte", et une petite fenêtre contenant trois boutons (Yes, Non et Cancel) apparaît. Cliquez sur le bouton de sortie "Yes".



IV.8 Conclusion

Dans ce chapitre, nous avons exposé la méthode pour résoudre un problème MOKP en utilisant l'algorithme SPEA-2. Puis nous avons présenté de l'application que nous avons développée, à l'aide du logiciel MATLAB, pour concrétiser notre application de travail.

Conclusion générale

Conclusion générale

Nous avons étudié dans ce travail l'une des méthodes les plus utilisées dans le domaine d'optimisation mono et multi-objectifs qu'est les algorithmes génétique pour résoudre un problème difficile bien connu dans domaine qui est le problème de sac à dos multi-objectifs .D'abord nous avons introduit l'optimisation combinatoire particulièrement ces problème et classifier les différentes méthode de résolution ainsi on a entamé le sujet de l'optimisation multi-objectifs.

Puis, nous avons présenté les algorithmes génétiques (AGS) et ses déférentes opérations, aussi bien que les algorithmes génétiques multi-objectifs avec une manière dé tallé.

Ensuite, nous avons parlé de manière générale sur le problème de sac à dos.

Dans la partie principale de notre travail, nous avons fait une conception pour l'algorithme génétique multi-objectifs est aligné avec le problème de sac à dos multi-objectifs, ensuite nous avons fourni un exemple simple pour illustrer les différentes opérations, enfin nous avons conçu une application (Matlab) permettant de résoudre ce genre de problème et implémenter l'algorithme (SPEA2) au problème de sac à dos multi- objectifs. .

Ce algorithme peut être développer au plus tard , pour étude d'antre problème par exemple un problème de sac à dos multi-objectif multidimensionnel dans lequel l'algorithme (SPEA2) est implémenté

Bibliographie

[1] LAYEB Abdeslam, Utilisation des Approches d'Optimisation Combinatoire pour La Vérification des Applications Temps Réel, Thèse Doctorat en informatique, Université Mentouri de Constantine, 2010.

[2] Laurent Noé. Recherche de similarités dans les séquences d'ADN : modèles et algorithmes pour la conception de graines efficaces.2005.

[3] Laurent Noé. Recherche de similarités dans les séquences d'ADN : modèles et algorithmes pour la conception de graines efficaces.2005.

[4] Cours optimisation chapitre 4 problèmes np-complets.

[5]Fonseca, C. M. and Fleming, P. J. GeneticAlgorithms for MultiobjectiveOptimization: Formulation, Discussion and Generalization. In Proceedings of the FifthInternationalConference on GeneticAlgorithms, pp. 416:423, San Mateo, California 1993.

[6] MAHDI. S. Optimisation Multi-objectif Par Un Nouveau Schéma De Coopération Méta/Exacte. Mémoire de Magister. Université Mentouri, Constantine.

[7] BENLAHRACHE, N. Optimisation Multi-Objectif Pour l'Alignement Multiple de Séquences 2007.

[8] Berro.A(2003). Optimisation multi-objectif et stratégies d'évolution en environnementdynamique .Thèse du Doctorat. Université des Sciences Sociales Toulouse I. France.

[9] Sébastien. A. (Septembre 2011). Documents, Graphes et Optimisation Multi-Objectifs. Rapport de recherche. L'Université de Rouen.

[10] ZERDANI. O. (2013). L'optimisation non Linéaire Multi-objectif. Thèse de Doctorat. UniversitéMOULOUD Mammeri, Tizi-ouzou

[11] Durand, J. M. Alliot, J. Noailles, Algorithmes génétiques, un croisement pour les problèmes partiellement séparables. Journées Evolution Artificielle Francophones JEAF, Toulouse, 1994.

[12] John Henry Holland. Adaptation in natural and artificialsystems : an introductoryanalysiswith applications to biology, control, and artificial intelligence. MIT press,1992.

[13]Z. Michalewicz ,"GeneticAlgorithms + Data Structures = Evolution Programs" ;3^{eme}édition .Springer, 1996

[14] Amira Gherboudj. Méthodes de résolution de problèmes difficiles académiques. Thèse de Doctorat, Université de Constantine2, 2013.

[15] S.Voisin, "Applicationdes algorithmes génétiques à l'estimation de mouvement par modélisation markovienne ", rapport DEA, Université Joseph, France,2004 .

[16] N.Benahmed,"Optimisation de réseaux de neurones pour la reconnaissance de chiffres manuscrits isolés: sélection et pondération des primitives par algorithme génétique".Université du Québec ,2002.

[17] D.Beasly, D .R . Bull, R. R. Matrin"Anoverview of genetic : Part 2, ResearshTopics", Universitycomputing, Vol 15 ,N° 4, 1993, p.170.

[18] Christophe Caux, Henri Pierreval et M-C Portmann. Les algorithmes génétiques et leur application aux problèmes d'ordonnancement.*Automatique-productiqueinformatiqueindustrielle*, 29(4-5):409–443, 1995.

[19] R. L. Haupt,S.E."Practicalgenetic algorithms", 2^{eme}édition, New Jersey,2004.

[20]OSMAN ALP and ERHAN ERKUT "An E¢ cientGeneticAlgorithm for the p-Median Problem". 2003 Kluwer AcademicPublishers. Manufactured in The Netherlands.

[21] Optimisation multiobjectif et stratégie d'évolution en environnement dynamique. AlinBerro 2001 thèse de doctorat.

[22] Carlos M. Fonseca and Peter J. Fleming, « Geneticalgorithmformultiobjectiveoptimization: formulation, discussion andgeneralization », in proceeding of the fifthinternetionalconference on geneticalgorithms, San Mateo, California, p. 416-423, 1993.

[23]An evolutionaryalgorithm for multiobjectiveoptimization: the stength Pareto approach. Technical report, SwissFederal Institute of Technology E. Zitzler and L. Thiele. (Zurich), 1998.

[24] The Pareto Archived Evolution Strategy: A new Baseline Algorithm For Multiobjective Optimisation. J.D.Knowles and D.W.Corne. Congress on Evolutionary Computation, p. 98 105. Washington, July 1999.

[25] F.Glover ; ArtificialIntelligance, heuristicframeworks and tabusearch ; Managerial and decisioneconomics ; volume11, pages 365-375 ;1990.

[26] EckartZitzler,MarcoLaumanns,andLotharThiele, SPEA2:Improvingthestrength paretoevolutionaryalgorithm for multiobjective optimization, EvolutionaryMethods for Design, Optimisation, and Control, CIMNE, Barcelona, Spain, 2002, pp. 95–100.

[27] Pierre Fouilhoux. Optimisation combinatoire : Programmation linéaire et algorithmes. Université Pierre et Marie Curie, 2015

[28] LALAMI, M. E. Contribution a la résolution de problémes d'optimisation combinatoire : methodes heuristiques et paralléles.

[29] HamdaouiFawzia. CONTRIBUTION A LA RESOLUTION DU PROBLEME DUSAC A DOS MULTIDIMENSIONNEL A CHOIX MULTIPLE. PhDthesis, Université Mohamed Boudiaf des sciences et de la technologie d'Oran

[30] Analyse de contraintes de coupe et de substitution pour des solutions de havresac multidimensionnelles améliorées. Annals of Operations Research, 117 :71–93

تلخيص : يهدف العمل الى تنفيذ طرق وأساليب التحليل للوصول الى حلول ذات جودة عالية في وقت معقول للمشاكل المتعددة الأهداف ,مجموعة الخوارزميات المعروضة في هذه المذكرة تم تحريبها على مشكلة حقيبة الظهر متعددة الأهداف. الغرض منها تعظيم دالتين موضوعتين او أكثر مع احترام ومراعاة شرط خطي واحد. بعد عرض المفاهيم المستخدمة في بحوث العمليات من اجل حل المشاكل الاستمثالية. درسنا الخوارزمية الجينية لحل مشكلة حقيبة الظهر ذات الغرض المزدوج، ومن اجل هذا استعملنا في برمجة هذه الخوارزمية MATLAB. ويتيح التطبيق او البرنامج الوصول الى أفضل حلول الممكنة لهذه المشكلة باستخدام SPEA-2.

الكلمات المفتاحية: optimisations multi-objectif ,optimisations ,الخوارزمية الجينية, مشكلة حقيبة الظهر ذات الغرض المزدوج, MATLAB.

Résumé : Ce travail se base sur la mise en couvre de méthodes de résolution pour obtenir des solutions des de bonne qualité en des temps raisonnables pour les problèmes multi-objectifs. L'ensemble algorithmes présentés dans ce mémoire est testé sur le problème du sac à dos multi-objectif. Il consiste à maximiser deux ou ensemble de fonctions objectives en respectant une seule contrainte linéaire. Après une présentation de quelques concepts utilisés en recherche opérationnelle pour résoudre les problèmes d'optimisation. Nous avons étudié l'algorithme génétique pour le problème du sac à dos bi- objectifs et pour cela nous avons implémenté cet algorithme avec un langage de programmations MATLAB. L'application permet l'optimisation des solutions obtenues par la résolution de problème du sac à dos bi-objectif avec l'algorithme génétique et SPEA-2.

MOT CLES : optimisations, optimisations multi-objectif, l'algorithme génétique, sac à dos bi-objectif, MATLAB.

Asbectra : This work focuses on the implementation of resolution methods to obtain good quality solutions in reasonable time for multi-objective problems. All algorithms presented in this paper are teseted on knapsack problem multi-objotive. Ir is to maximize two or set of functions objectives to a single linear constraint. After presenting some concepts used in operations research of solving optimization problens We studied the genetic en algorithm to the prablem of dual objectives knapsack and why we implemented this algorithm with a MATLAB programming language .The application allows optimization solutions obtained by solving the problem of dual-objectf knapsack with the genetic algorithm and SPEA-2

Keywords : optimization, optimizatation multi-objectif, genetic algorithm, knapsack dual-objectif, MATLAB