

## *Etude des méthodes d'optimisation Multicritères*

### **1. Introduction :**

Les ingénieurs se heurtent quotidiennement à des problèmes technologiques de complexité grandissante, qui surgissent dans des secteurs très divers. Le problème à résoudre peut fréquemment être exprimé sous la forme générale d'un problème d'optimisation, dans lequel on définit une fonction objectif, ou fonction Coût, que l'on cherche à minimiser (ou maximiser) par rapport à tous les paramètres concernés.

Il existe deux types de méthodes d'optimisation, la première est l'optimisation mono-objectif, qui se base sur la minimisation (ou la maximisation) d'une seule fonction objectif où le but est de trouver la meilleure solution appelée solution optimale qu'est facilement définie suivant une seule performance du problème étudié (temps de réponse, temps de montée, la robustesse, taux d'erreur,...etc). D'autre part, l'optimisation multi-objectif optimise simultanément plusieurs fonctions objectives qui sont souvent contradictoires, on cherche à trouver la meilleure solution suivant un ensemble de performance du problème (temps de réponse plus la robustesse, temps de réponse plus temps de montée plus la robustesse,...etc). Où le résultat d'un problème d'optimisation multicritère est généralement un assortiment de solutions, qui se distinguent par différents compromis réalisés entre les objectifs. Cet assortiment est connu par *Pareto-optimal*. Donc le but de l'optimisation multicritère, est d'obtenir les solutions de *Pareto*, par conséquent, à connaître l'ensemble des compromis possibles entre les objectifs. La résolution d'un problème d'optimisation multicritère, a conduit les chercheurs à proposer des méthodes de résolution de plus en plus performantes.

Dans ce chapitre, nous présentons les principes de base de l'optimisation multiobjectif, et nous donnerons aussi une présentation de différents méthodes d'optimisation multiobjectif.

## 2. Principaux concepts en optimisation:

Tout d'abord nous définissons les notions communes à n'importe quelle méthode d'optimisation multicritère :

**2.1. Fonction objectif :** Une fonction objectif est une fonction qui modélise le but à atteindre Dans le problème d'optimisation sur l'ensemble des critères. Il s'agit de la fonction qui doit être optimisée. Elle est notée  $F(x)$  de manière générale  $F(x)$  est un vecteur :

$F(x) = [f_1(x), f_2(x), \dots, f_k(x)]$ . Elle est aussi appelée : *critère d'optimisation, fonction coût, fonction d'adaptation, ou encore performance* [3].

**2.2. Paramètres :** Un paramètre du problème d'optimisation, est une variable qui exprime une donnée quantitative ou qualitative sur une dimension du problème: coût, temps, taux d'erreurs,...etc. [3]. Ces paramètres correspondent aux variables de la fonction objectif. Ils sont ajustés pendant le processus d'optimisation, pour obtenir les solutions optimales. On les appelle aussi variables d'optimisation, variables de conception ou de projet.

**2.3. Vecteur de décision :** Un vecteur de décision est un vecteur correspondant à l'ensemble des variables du problème, il est noté :  $\vec{x} = [x_1, x_2, \dots, x_n]^T$  avec :  $n$  le nombre de variables ou dimension du problème et  $x_k$  la variable sur la dimension  $K$ .

**2.4. Critère de décision :** 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.

**2.5. Contraintes :** Une contrainte du problème est une condition que doivent respecter les vecteurs de décision du problème. Une contrainte est notée :  $g_i(\vec{x})$  avec :  $i = 1, \dots, q$   
 $q$  : le nombre des contraintes [4].

**2.6. Espace de recherche :** représentant l'ensemble des valeurs pouvant être prises par les variables.

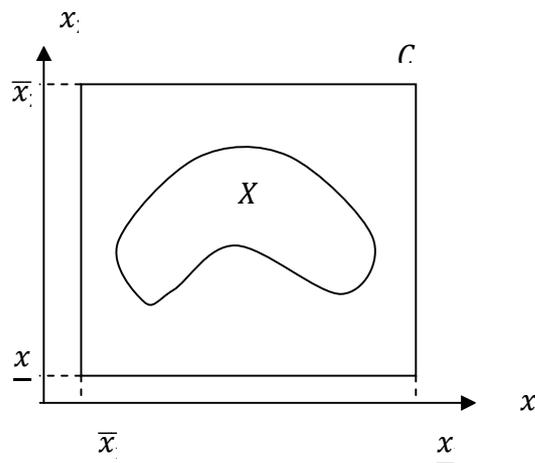
**2.7. Espace réalisable :** représentant l'ensemble des valeurs des variables satisfaisant les contraintes. Dans la figure II.1, on a :

$\underline{x}_i$  et  $\overline{x}_i$ , représentent la borne inférieure et supérieure pour  $x_i$ .

$C$  : représente l'espace de recherche égal au produit cartésien des domaines des variables.

$X$  : représente l'espace réalisable délimité par les contraintes. Cette figure illustre en dimension deux ( $C = \mathbb{R}^2$ ) le fait que :  $X = \{\bar{x} / g_i(\bar{x}) \leq 0, \bar{x} \in C\}$

L'espace réalisable  $X$  (délimité par des contraintes) est un sous-ensemble de  $C$ , l'espace de recherche. [11]



**Figure II.1.** Espace de recherche et espace réalisable.

**2.8. Espace des objectifs :** ensemble image de l'espace de recherche, déterminé par toutes les valeurs possibles des fonctions objectifs [4].

**2.9. Minimum local :** Un point  $x \in X$  est un minimum local du problème si et seulement si :

$$\forall x' \in V(X), f(x) \leq f(x') \text{ où } V(X) \text{ définit un voisinage de } x.$$

**2.10. Minimum global :** Un point  $x \in X$  est un minimum global du problème si et seulement si :

$$\forall x' \in X, f(x) \leq f(x') \text{ (il appartient alors à l'ensemble des minimiseurs) [11].}$$

### 3. Problème d'optimisation multicritère :

Le problème de l'optimisation multicritère a été étudié auparavant par les deux communautés suivantes : la communauté de control (Zadeh, 1963, Gembicki et Haimes 1975 Lin

1976, Giesy 1978, Tabak et al 1979) et la communauté de recherche opérationnelle (Cohon 1978) [4].

Les problèmes réels invoquent souvent de multiples mesures de performance ou objectifs, qui doivent être optimisés simultanément. En pratique, ceci n'est pas toujours possible car les objectifs peuvent être conflictuels, du fait qu'ils mesurent différents aspects de la qualité de la solution. Dans ce cas, la qualité d'un individu est décrite non pas par un scalaire mais par un vecteur. La performance, la fiabilité et le coût sont des exemples d'objectifs conflictuels [12]

Un problème d'optimisation multicritère consiste à trouver le vecteur de décision idéal  $\vec{x}^*$  tel que les contraintes  $g_i(\vec{x})$  et  $h_j(\vec{x})$  soient satisfaites et dont  $\vec{F}(\vec{x}^*)$  est optimal [13].

En conséquence, le problème d'optimisation multicritère peut être défini formellement comme suit :

$$\text{Optimiser : } \vec{F}(\vec{x}) = \{f_1(\vec{x}), f_2(\vec{x}), \dots, f_m(\vec{x})\} \quad (m : \text{fonctions à optimiser}) \quad (\text{II-1})$$

$$\text{tel que : } \vec{x} \in X$$

$$X = \left\{ \vec{x} \left| \begin{array}{l} g_i(\vec{x}) \leq 0 \quad i = 1, \dots, q \\ h_j(\vec{x}) = 0 \quad j = 1, \dots, l \\ \vec{x} = [x_1 \dots x_n]^T \end{array} \right. \right\}$$

$\vec{x}$  : est un vecteur de  $n$  variables de décisions.

$g_i(\vec{x})$ ,  $h_j(\vec{x})$  sont les fonctions des contraintes qui déterminent le domaine  $X$  des solutions.

$X$  : représente le domaine réalisable.

Ces objectifs multiples sont souvent concurrents où l'amélioration de l'un entraîne la détérioration de l'autre ou des autres, Ce conflit entre les objectifs s'explique facilement : de façon générale, des structures de haute performance tendent à avoir un coût élevé, alors que des dispositifs plus simples et usuellement peu coûteux auront des performances moindres. Selon les contraintes du cahier des charges, une solution intermédiaire (performance satisfaisante et coût acceptable) peut être optimale [13]. Dans les problèmes multiobjectifs, l'optimum n'est plus une simple valeur comme pour les problèmes mono-objectif, mais un ensemble de points, appelé l'ensemble des meilleurs compromis ou le front Pareto [12].

Tous les énoncés et définitions seront donnés dans le cadre de problèmes de minimisation. En effet un problème de maximisation peut être aisément transformé en un problème de minimisation en considérant l'équivalence suivante :

$$\text{Maximiser } \vec{F}(\vec{x}) \Leftrightarrow \text{Minimiser } -\vec{F}(\vec{x})$$

#### 4. Méthodes d'optimisation multicritère :

Les approches de résolution des problèmes multiobjectifs peuvent être réparties en classes suivantes :

- ❖ Les méthodes Métaheuristiques.
- ❖ Les méthodes non Pareto.
- ❖ Les méthodes Pareto.
- ❖ Les méthodes hybrides.

##### 4.1. Méthodes Métaheuristiques :

Les métaheuristiques sont un ensemble d'algorithmes d'optimisation visant à résoudre les problèmes d'optimisation difficiles. Elles sont souvent inspirées par des systèmes naturels, qu'ils soient pris en physique (cas du recuit simulé), en biologie de l'évolution (cas des algorithmes génétiques) ou encore en éthologie (cas des algorithmes de colonies de fourmis ou de l'optimisation par essaims particulaires). Dans la section qui suit, on essayera de donner un bref aperçu sur les méthodes métaheuristiques de base les plus populaires [12].

###### 4.1.1. La Méthode de Recuit Simulé :

Cette méthode a été inspirée du processus physique du recuit utilisé en métallurgie. Ce processus consiste en une suite de cycles de refroidissement lent pour obtenir un matériau homogène et de très bonne qualité. En effet, il alterne des cycles de refroidissement lent et de réchauffage (recuit) qui tend à minimiser l'énergie du matériau. Cette méthode s'appuie sur l'algorithme de Metropolis [14] qui permet de décrire l'évolution d'un système thermodynamique. Par analogie avec le processus physique, la fonction  $f(x)$  à minimiser deviendra l'énergie  $E$  du système où  $x$  est un état donné de la matière. Un paramètre fictif est également introduit : la température  $T$  du système. Partant d'une solution donnée nous générons une solution voisine en

utilisant une transformation qui change  $x$  en  $x'$ . Si celle-ci améliore le critère que l'on cherche à optimiser, c'est-à-dire :  $\Delta f = f(x') - f(x) < 0$  on dit alors qu'on a fait baisser l'énergie du système, sinon elle la dégrade. En acceptant une solution améliorant le critère, nous tendons ainsi à chercher l'optimum dans le voisinage de la solution de départ, alors que l'acceptation d'une solution moins bonne, c'est-à-dire  $\Delta f = f(x') - f(x) \geq 0$  permet alors d'explorer une plus grande partie de l'espace de solutions et tend à éviter de s'enfermer trop vite dans la recherche d'un optimum local. La solution voisine  $x'$  est acceptée avec une probabilité  $p$ . Si  $x'$  n'est pas meilleur que  $x$ , alors  $x'$  est accepté avec une probabilité suivante :  $\exp\left[-\frac{\Delta f}{T}\right]$  [15].

#### 4.1.2. Méthode de recherche Tabou :

La recherche Tabou est une métaheuristique développée par Glover. Elle est basée sur des idées simples, mais elle est néanmoins très efficace. Elle combine une procédure de recherche locale avec un certain nombre de règles et de mécanismes pour surmonter l'obstacle des optima locaux. Elle a été appliquée avec succès pour résoudre de nombreux problèmes difficiles d'optimisation combinatoire [16]. Dans une première phase, la méthode de recherche Tabou peut être vue comme une généralisation des méthodes d'amélioration locale. En effet, en partant d'une solution quelconque  $x$  appartenant à l'ensemble de solutions  $X$ , nous nous dirigeons vers une solution  $x'$  appartenant au voisinage  $V(x)$  de  $x$ . L'algorithme explore donc itérativement l'espace de solutions  $X$ . Afin de choisir le meilleur voisin  $x'$  l'algorithme évalue la fonction objectif  $F$  en chaque point  $x'$  et retient le voisin qui améliore la valeur de  $F$ , ou celui qui la dégrade le moins [15].

#### 4.1.3. Les Algorithmes Génétiques (AGs) :

Les algorithmes génétiques (AGs) ont été introduits par Holland comme un modèle de méthode adaptative. Ils ont été efficacement utilisés pour résoudre plusieurs problèmes d'optimisation multicritère.

L'algorithme génétique est une technique d'optimisation basée sur les concepts de la sélection naturelle de Darwin et la procréation selon les règles de Mendel. La sélection naturelle que Darwin appelle l'élément "propulseur" de l'évolution, favorise les individus d'une population qui sont les mieux adaptés à un environnement. Pour évaluer les individus d'une population on

utilise une *fonction d'évaluation* cette fonction est souvent une transformation de la fonction objectif, appelé aussi la fonction *fitness*, le résultat fournit par cette fonction va permettre de sélectionner ou de refuser un individu pour ne garder que les individus ayant le meilleur coût en fonction de la population courante La sélection est suivie de la procréation, réalisée à l'aide de croisements et de mutations au niveau du patrimoine génétique des individus (ou "génotype") constitué d'un ensemble de gènes. Ainsi, deux individus "parents", qui se croisent transmettent une partie de leur patrimoine génétique à leurs descendants. Le génotype de l'enfant fait que celui-ci est plus ou moins adapté à l'environnement. S'il est bien adapté, il a une plus grande chance de procréer dans la génération future. Au fur et à mesure des générations, on sélectionne les individus les mieux adaptés, et l'augmentation du nombre de ces individus fait évoluer la population entière.

Dans les algorithmes génétiques, nous simulons le processus d'évolution d'une population. On part d'une population initiale de  $N$  solutions du problème représentées par des individus judicieusement choisis. Le degré d'adaptation d'un individu à l'environnement est exprimé par la valeur de la fonction coût  $f(x)$ , où  $x$  est la solution que l'individu représente. On dit qu'un individu est d'autant mieux adapté à son environnement, que le coût de la solution qu'il représente est plus faible ou plus important selon le(s) critère(s) d'optimisation choisi(s). Au sein de cette population, intervient alors la sélection au hasard d'un ou deux parents, qui produisent une nouvelle solution, à travers les opérateurs génétiques, tels que le croisement et la mutation. La nouvelle population, obtenue par le choix de  $N$  individus parmi les populations (parents et enfants), est appelée génération suivante. En itérant ce processus, on produit une population plus riche en individus mieux adaptés. [15].

Le chapitre 3 fera l'objet d'une présentation plus approfondie des algorithmes génétiques.

D'autres Métaheuristiques ont été adapté au cas d'optimisation multiobjectif et ont prouvé leur succès :

- ◆ Les colonies de fourmis.
- ◆ L'Optimisation par essaim de particules.
- ◆ Les réseaux de neurones artificiels [16].

## 4.2 Méthodes non Pareto:

Les méthodes non Pareto ne traitent pas le problème comme un véritable problème multiobjectif. Elles cherchent à ramener le problème initial à un ou plusieurs problèmes mono objectifs. Parmi ces approches nous citons :

### 4.2.1. Les méthodes Agrégées:

Cette approche, consiste aux nombreuses méthodes de résolution, parmi les méthodes qui utilisent cette approche, nous pouvons citer les méthodes suivantes :

- Méthodes d'agrégation par pondération.
- Méthode  $\varepsilon$ -contraintes.
- Méthode de but à atteindre.
- Méthode de min-max.
- Méthode de Goal programming.

#### 4.2.1.1. Agrégation par pondération:

Dans cette approche, le but consiste à ramener le problème multicritère à un problème monocritère plus simple à traiter. Cette méthode est la plus simple des méthodes d'optimisation multi-objectif. La transformation que l'on effectue est la suivante :

$$\text{Minimiser } \sum_{i=1}^m w_i \cdot f_i(x) \quad (\text{II-2})$$

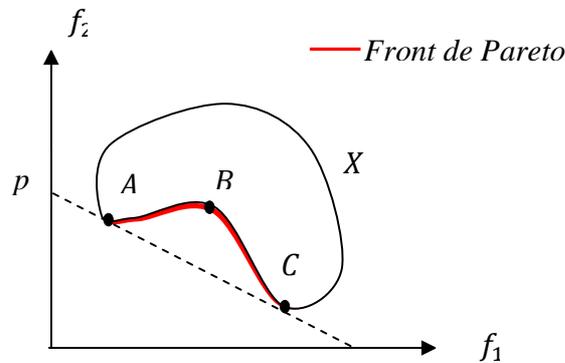
$$\text{Avec : } \vec{x} \in X, w_i \in [0, 1] \text{ et } \sum_{k=1}^m w_k = 1$$

$X$  : représente le domaine réalisable.

$w_i$  : Appelé le poids, est une pondération associée au critère, cette pondération permet d'exprimer des préférences sur les critères de décision [4].

Cette méthode est très efficace du point de vue algorithmique, mais elle ne permet pas de trouver les solutions enfermées dans des concavités. La figure II-2 illustre ce cas en dimension 2. En effet pour un vecteur de poids  $\vec{w} = [w_1 \ w_2]^T$  fixé la valeur optimale atteignable pour la fonction objectif crée est :  $p$ . Les deux points Pareto optimaux trouvés sont A et C. En faisant varier le vecteur  $\vec{w}$  il est possible de trouver d'autres points Pareto optimaux. Seulement tous ces points se

trouveront sur les parties convexes de la surface de compromis. Il n'existe pas de valeur possible pour  $\vec{w}$  permettant de trouver par exemple le point  $B$  [11].



**Figure II-2** Interprétation graphique de la méthode d'agrégation par pondération

Il existe une autre méthode d'agrégation appelée « l'Agrégation par L'intégrale de Choquet » c'est une deuxième technique pour l'agrégation de critères, plus sophistiquée que la première. Son intérêt est qu'elle permet de décrire des situations difficilement modélisables. Cette technique utilise une mesure floue (capacité) sur l'ensemble de critères  $E_c = \{c_1, c_2, \dots, c_m\}$ , qui représente l'importance des sous ensemble de critères dans la prise de décision. Il y a deux notions importantes pour l'analyse sémantique de l'intégrale de Choquet: La notion d'importance globale d'un critère et l'interaction entre les critères [13].

#### 4.2.1.2. Méthode $\varepsilon$ -contrainte :

Elle est aussi dite méthode du compromis. Elle transforme un problème d'optimisation multiobjectif en un problème d'optimisation mono-objectif de la façon suivante :

- ◆ Choisir un objectif à optimiser prioritairement.
- ◆ Choisir un vecteur de contraintes initiales  $\varepsilon$ .
- ◆ transformer le problème en gardant l'objectif prioritaire et en transformant les autres objectifs en contraintes d'inégalités comme suit :

$$\begin{aligned} & \text{Minimiser } f_1(x) && \text{(II-3)} \\ & \text{Tel que : } f_2(x) \leq \varepsilon_2, \dots, f_m(x) \leq \varepsilon_m \\ & x \in X \end{aligned}$$

Cette approche a l'avantage par rapport à la précédente dans les problèmes non convexes, mais présente plusieurs inconvénients à savoir :

- ◆ la formulation des préférences utilisateur est délicate et nécessite une connaissance approfondie du problème de départ.
- ◆ Les contraintes rajoutées compliquent la résolution du problème [16].

#### 4.2.1.3. Méthode Min-Max :

Cette méthode consiste à transformer le problème multiobjectif en un problème à un seul objectif où l'on cherche à minimiser l'écart relatif par rapport à un point de référence appelé but fixé par la méthode ou le décideur. Il existe plusieurs manières de caractériser la distance entre un point de référence (le but) et un autre, notamment à l'aide de normes. Une norme est définie de manière suivante :

$$L_r(F(x)) = \left[ \sum_{i=1}^m |B_i - f_i(x)|^r \right]^{\frac{1}{r}} \quad (\text{II-4})$$

Les principales normes utilisées sont :  $L_1(F(x)) = \sum_{i=1}^m |B_i - f_i(x)|$  la distance classique, et la norme  $L_\infty(F(x)) = \max_{i \in \{1,2,\dots,m\}} (B_i - f_i(x))$ . C'est cette dernière qui est utilisée dans l'approche min-max appelée aussi approche de Tchebychev:

$$\left| \begin{array}{l} \text{Minimiser : } \max(B_i - f_i(x)) \\ \text{tel que : } g_q(x) \leq 0 \\ \text{Avec : } x \in \mathfrak{R}^n, g_q(x) \in \mathfrak{R}^q, F(x) \in \mathfrak{R}^m \end{array} \right. \quad (\text{II-5})$$

Dans cette approche, le point de référence joue un rôle fondamental, s'il est mal choisi la recherche peut s'avérer être très laborieuse. La figure II-3 illustre, en dimension 2, le cas d'une recherche avec un but  $B$  fixé. Il est clair que l'approche permet de traiter les problèmes non convexes à condition que le point de référence soit choisi judicieusement [15].

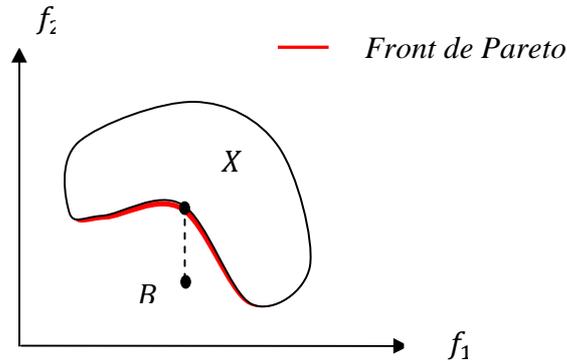


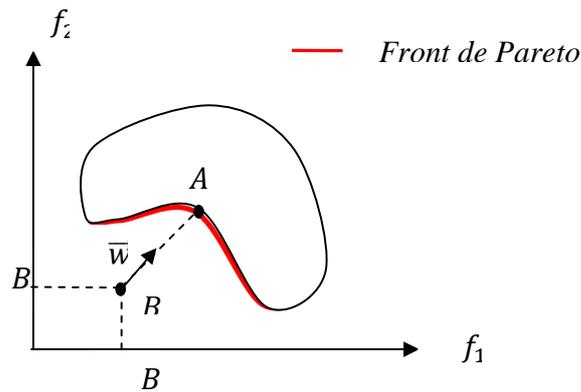
Figure II-3 Interprétation graphique de l'approche Min-Max

**4.2.1.4. Méthode de but à atteindre :**

On définit un ensemble de buts qu'on espère atteindre pour chaque fonction objectif. L'algorithme tente de minimiser l'écart entre la solution courante et ses buts. Cette approche, comme celle de min-max, utilise un point de référence pour guider la recherche. Mais elle introduit aussi une direction de recherche, si bien que le processus de résolution devra suivre cette direction. A la différence de l'approche min-max, qui utilise des normes pour formaliser la distance au point de référence, l'approche du but à atteindre utilise des contraintes, comme de l'approche  $\epsilon$ -contrainte, pour déterminer la position du point de référence. L'écart par rapport à ce but est contrôlé grâce à la variable  $\lambda$  introduite à cet effet :

$$\left\{ \begin{array}{l} \text{Minimiser : } \lambda \\ \text{Tel que :} \\ \quad f_i(x) - w_i \cdot \lambda \leq B_i \\ \text{Avec :} \\ \quad g_q(x) \leq 0 \\ x \in \mathfrak{R}^n, g_q(x) \in \mathfrak{R}^q, F(x) \in \mathfrak{R}^m \end{array} \right. \quad (\text{II-6})$$

Ainsi en minimisant  $\lambda$  et en vérifiant toutes les contraintes, la recherche va s'orienter vers le but  $B$  et s'arrêter sur le point  $A$  faisant partie de la surface de compromis (voir l'illustration en 2 dimensions de la figure II-4). Cette approche permet comme l'approche par  $\epsilon$ -contrainte et l'approche min-max, de trouver les parties non convexes des fronts Pareto.



**Figure II-4** Interprétation graphique de l'approche but à atteindre

Cependant, cette approche comme les précédentes, doit être itérée plusieurs fois dans le but d'obtenir un ensemble de points Pareto optimaux. Les paramètres  $\vec{w}$  et  $B$  doivent être bien choisis par l'utilisateur. Bien que ces paramètres permettent une grande flexibilité de la recherche (orientation et but), s'ils sont mal choisis, ils peuvent, dans certains cas extrêmes, donner des résultats non cohérents [15].

#### 4.2.1.5 Méthode de Goal Programming :

Dans cette méthode le décideur fixe un but  $T_i$  à atteindre pour chaque objectif  $f_i$ . Ces valeurs sont ensuite ajoutées au problème comme des contraintes supplémentaires. La nouvelle fonction objectif est modifiée de façon à minimiser la somme des écarts entre les résultats et les buts à atteindre :

$$\min \sum_{i=1}^m |f_i(\vec{x}) - T_i| \quad (\text{II-7})$$

$T_i$  : représente la valeur à atteindre pour le  $i^{\text{ème}}$  objectif.

Nous pouvons reprendre la critique faite pour la somme pondérée. La méthode est facile à mettre en œuvre mais l'efficacité de la méthode dépend de la définition des poids et des objectifs à atteindre. Cette méthode a l'avantage de fournir un résultat même si un mauvais choix initial a conduit le décideur à donner un ou plusieurs but(s)  $T_i$  non réalisable(s) [17].

### 4.2.2. Méthode lexicographique :

Fourman a proposé une méthode dans laquelle les objectifs sont préalablement rangés par ordre d'importance par le décideur. Ensuite, l'optimum est obtenu en minimisant tout d'abord la fonction objectif la plus importante puis la deuxième et ainsi de suite.

Soient les fonctions objectifs  $f_i$  avec  $i = 1, \dots, k$ , supposons un ordre tel que :  $f_1 \succ f_2 \succ \dots \succ f_k$  Il faut :

$$\left| \begin{array}{l} \text{Minimiser } f_1(x) \\ \text{avec : } g_j(x) \text{ satisfait : } j = 1, \dots, m \end{array} \right. \quad (\text{II-8})$$

Soit  $x_1^*$ , la meilleure solution trouvée avec :  $f_1^* = f_1(x_1^*)$   $f_1$  devient alors une nouvelle contrainte.

L'expression du nouveau problème est donc :

$$\left| \begin{array}{l} \text{Minimiser } f_2(x) \\ \text{avec : } g_j(x) \text{ satisfait : } j = 1, \dots, m \text{ et : } f_1(x) = f_1^* \end{array} \right. \quad (\text{II-9})$$

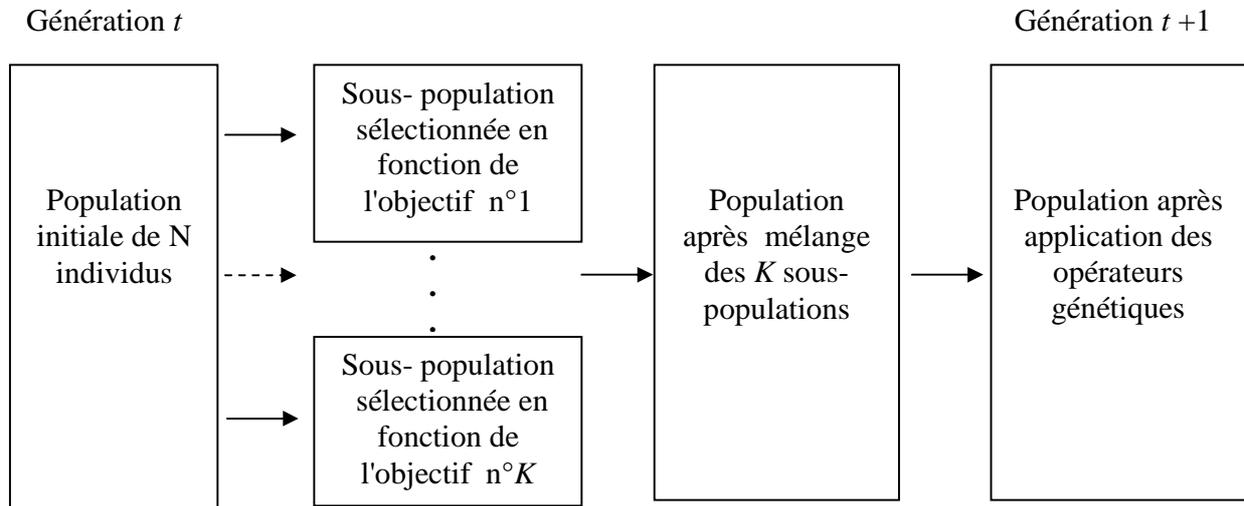
Soit  $x_2^*$  la solution de ce problème. Le  $i^{\text{ème}}$  problème sera le suivant :

$$\left| \begin{array}{l} \text{Minimiser } f_i(x) \\ \text{avec : } g_j(x) \text{ satisfait } j = 1, \dots, m, \text{ et : } f_1(x) = f_1^*, f_2(x) = f_2^*, \dots, f_{i-1}(x) = f_{i-1}^* \end{array} \right. \quad (\text{II-10})$$

La procédure est répétée jusqu'à ce que tous les objectifs soient traités. La solution obtenue à l'étape  $k$  sera la solution du problème. L'inconvénient essentiel de cette méthode est la grande importance attribuée aux objectifs classés en premier. La meilleure solution  $f_1^*$  trouvée pour l'objectif le plus important va faire converger l'algorithme vers une zone restreinte de l'espace d'état et enfermer les points dans une niche (ensemble d'individus situés dans un espace restreint) [17].

### 4.2.3. Vector evaluated genetic algorithm (VEGA):

En 1985 Schaffer propose une extension d'un algorithme génétique simple pour la résolution d'un problème multiobjectif. Cette méthode est appelée Vector Evaluated Genetic Algorithm. La seule différence avec un algorithme génétique simple est la manière dont s'effectue la sélection. L'idée est simple. Si nous avons  $K$  objectifs et une population de  $N$  individus, une sélection de  $N/K$  individus est effectuée pour chaque objectif. Ainsi  $K$  sous-populations vont être créées, chacune d'entre elles contenant les  $N/K$  meilleurs individus pour un objectif particulier. Les  $K$  sous-populations sont ensuite mélangées afin d'obtenir une nouvelle population de taille  $N$ . Le processus se termine par l'application des opérateurs génétiques de modification (croisement et mutation).



**Figure II.5 :** Schéma de fonctionnement de VEGA

La méthode VEGA a tendance à créer des sous-populations dont les meilleurs individus sont spécialisés pour un objectif particulier. L'évolution de la population favorise l'apparition des espèces. En effet, comme la méthode de sélection ne tient compte que d'un seul objectif, elle privilégie les individus qui obtiennent une bonne performance pour cet objectif. Dès lors ces individus ne seront sélectionnés que lorsqu'on effectuera la sélection sur cet objectif. Les individus que Schaffer appelle les individus "milieu", parce qu'ayant une performance générale acceptable mais ne possédant aucun critère fort, vont être éliminés car ils ne seront sélectionnés dans aucune sous-population. Cette disparition entraîne la spécialisation des individus pour

chaque objectif. Ce résultat est contraire au but initial de la méthode qui était de trouver un compromis entre les différents critères [17].

### 4.3. Les approches Pareto :

Les approches Pareto ne transforment pas les objectifs du problème, ceux-ci sont traités sans aucune distinction pendant la résolution.

Dans les paragraphes suivants, nous définissons tout d'abord la notion de dominance au sens de Pareto, la frontière de Pareto, Pareto optimal. Ensuite, nous présentons les techniques évolutionnaires utilisant cette notion.

#### 4.3.1. La dominance au sens de Pareto :

L'idée d'utiliser la dominance au sens de Pareto a été proposée par Goldberg pour résoudre les problèmes multiobjectifs. Il suggère d'utiliser le concept d'optimalité de Pareto pour respecter l'intégralité de chaque critère car il refuse de comparer a priori les valeurs de différents critères. L'utilisation d'une sélection basée sur la notion de dominance de Pareto va faire converger la population vers un ensemble de solutions efficaces. Ce concept ne permet pas de choisir une alternative plutôt qu'une autre mais il apporte une aide précieuse au décideur.

Vilfredo Pareto est un mathématicien italien (Pareto, 1896). Il a posé les bases de la solution d'un problème économique multiobjectif : « Dans un problème multiobjectif, il existe un équilibre tel que l'on ne peut améliorer un critère sans détériorer au moins un des autres ». Cet équilibre est appelé *optimum Pareto*. Donc une solution  $x$  est dite *Pareto optimale* si elle n'est dominée par aucune autre solution appartenant à l'espace réalisable  $X$ . Ces solutions sont appelées solutions *non dominées*, ou *non inférieures*. [17].

**Définition** (*La dominance au sens de Pareto*) : Considérons un problème de minimisation Soient :  $u = [u_1, \dots, u_n]^T$  et  $v = [v_1, \dots, v_n]^T$  deux vecteurs de décision. On dit que le vecteur de décision  $u$  domine le vecteur  $v$  (dénote  $u \leq v$ ), si et seulement si :

$$\forall i \in \{1, 2, \dots, k\}, f_i(u) \leq f_i(v) \wedge \exists i \in \{1, 2, \dots, N\}, f_i(u) < f_i(v) \quad (\text{II-11})$$

Dans le processus d'optimisation multi-objectif, le concept de dominance du Pareto est utilisé afin de comparer et ranger le vecteur de variables des décisions :

$u$  domine  $v$  dans le sens du Pareto, signifie que  $F(u)$  est mieux que  $F(v)$  pour tous les objectifs, et il y a au moins une fonction objectif pour laquelle  $F(u)$  est strictement meilleure que  $F(v)$  [12]

**Définition (Pareto optimal) :** Soit  $x = [x_1, \dots, x_n]^T$  un vecteur de décision avec :  $x \in X$  (L'espace réalisable)  $x$  est dit *Pareto optimal*, s'il n'existe pas une solution  $y$  domine  $x$ . Une solution Pareto optimal appelée aussi : solution efficace, non inférieure ou non dominée solution.

**Définition (front de Pareto):** Le front (frontière) de Pareto est l'ensemble des solutions Pareto optimales qui sont composées des points, ne sont dominés par aucun autre Le front de Pareto appelé aussi surface de compromise ou l'ensemble des solutions efficaces [13].

La figure II.6, représente le Front du Pareto pour un problème de minimisation et maximisation de deux fonctions objective [12]

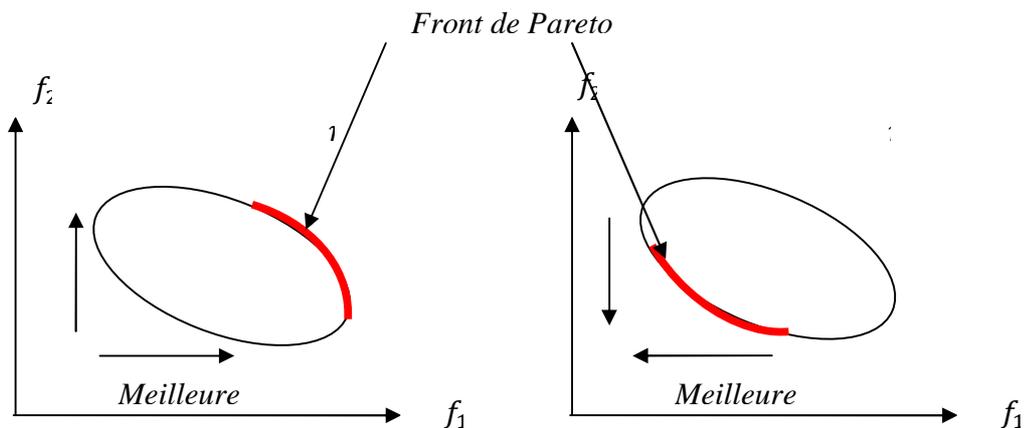
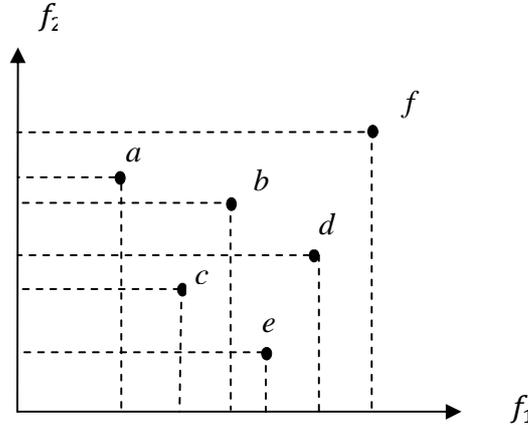


Figure II.6 le front de Pareto.

**4.3.2. Exemple :** Pour mieux expliciter les notions précédemment définies, considérons l'exemple suivant :



**Figure II-7** Exemple de dominance et d'optimalité au sens de Pareto

Soient six vecteurs de décision :

$$a = (a_1, a_2), b = (b_1, b_2), c = (c_1, c_2), d = (d_1, d_2), e = (e_1, e_2), f = (f_1, f_2)$$

Soit la fonction objectif  $F(x) = [f_1(x), f_2(x)]$ . La représentation graphique de l'image de ces six vecteurs par la fonction  $F$  est fournie dans la figure II-3. Dans cet exemple on peut écrire :

- $f$  : est dominé par tous les autres vecteurs.
- $b$  : est dominé uniquement par  $c$ .
- $d$  : est dominé par  $c$  et  $e$ .
- $a, c$  et  $e$  : ne sont dominés par aucun vecteur.

Par conséquent les solutions  $(a, c, e)$  sont *Pareto optimaux*. Donc le *front de Pareto* est l'ensemble :  $\{a, c, e\}$ [4].

### 4.3.3. Multiple Objectives Genetic Algorithm (MOGA) :

Cet algorithme, proposé par Fonseca et Fleming (1993), utilise la notion de dominance pour ranger les individus de la population. Il diffère de l'algorithme génétique standard uniquement dans la manière dont la fitness est assignée pour chaque solution. Pour démarrer l'algorithme, les relations de domination sont d'abord calculées pour chaque solution. Puis, pour une solution  $i$ , un rang égal à un plus le nombre de solutions  $n_i$  qui dominent la solution  $i$  est attribué. Une fitness

est ensuite attribuée à chaque solution en fonction de son rang, les individus avec les rangs les plus faibles ayant les meilleures fitness. Afin de maintenir la diversité entre les solutions non dominées, les auteurs utilisent une fonction de partage (*Sharing*) [18].

#### 4.3.3.1. Maintenir la diversité :

Maintenir un certain degré de diversité dans la population d'un algorithme évolutionnaire consiste à éviter que la population ne converge prématurément vers une petite zone de l'espace de recherche ou de l'espace des objectifs. En effet, s'il n'existe pas de mécanisme de contrôle de la diversité, les opérations de sélection vont privilégier trop vite certains individus meilleurs à cette étape de la recherche. Cette convergence prématurée a comme effet de limiter la recherche à un sous-ensemble plus restreint de l'espace de recherche, qui peut ne contenir aucune solution optimale [11].

#### 4.3.3.2. Le sharing :

Le sharing consiste à ajuster la fitness des individus pour éviter qu'ils se concentrent dans une niche principale (optimum globale). La technique de partage de la fitness (fitness sharing), introduite par Goldberg et Richardson, réduit la fitness de chaque individu d'un facteur correspondant environ au taux d'agrégation de la population autour de son voisinage [18].

Pour déterminer les bornes du domaine ouvert autour de l'individu choisi, on définit une distance maximale, appelée :  $\sigma_{\text{shar}}$  au delà de laquelle les individus ne seront plus considérés comme faisant parti du domaine ouvert.

La distance séparant deux individus  $i$  et  $j$  est calculée grâce à la fonction  $d(i, j)$ .

La valeur d'adaptation  $F(i)$  d'un individu  $i \in P$  (populations) est égale à son cout  $F'(i)$  divisé par sa valeur de niche :

$$F(i) = \frac{F'(i)}{\sum_{j \in P} Sh(d(i, j))} \quad (\text{II-12})$$

Où la fonction  $Sh$  est définie comme suit [11]:

$$Sh(d(i, j)) = \begin{cases} 1 - \left(\frac{d(i, j)}{\sigma_{\text{share}}}\right)^2 & \text{si } d(i, j) < \sigma_{\text{share}} \\ 0 & \text{sinon} \end{cases} \quad (\text{II-13})$$

La figure II.8 montre deux exemples de répartition de populations dans le cas d'une fonction multimodale: le premier sans sharing et le deuxième avec sharing.

La méthode permet d'obtenir des solutions de bonne qualité et s'implante facilement. Toutefois, les performances sont très dépendantes de la valeur du paramètre  $\sigma_{shar}$  utilisé dans le sharing [18].

#### 4.3.4. Non dominated Sorting Genetic Algorithm (NSGA):

Dans l'algorithme NSGA proposé par Srinivas et Deb (1993), le calcul de fitness s'effectue en divisant d'abord la population en plusieurs fronts en fonction du degré de dominance au sens de Pareto de chaque individu. Les individus non dominés de la population courante constituent le premier front de Pareto. On attribue alors à tous les individus de ce front la même valeur de fitness factice. Cette valeur est supposée donner une chance égale de reproduction à tous ces individus. Mais pour maintenir la diversité de la population, il est nécessaire d'appliquer une fonction de partage sur cette valeur. En suite, ce premier groupe d'individus est temporairement supprimé de la population. On recommence cette procédure jusqu'à l'identification des solutions du deuxième front. La valeur factice de fitness attribuée à ce second groupe est inférieure à la plus petite fitness, après application de la fonction de partage sur le premier front. Ce mécanisme est répété jusqu'à ce que l'on ait traité tous les individus. L'algorithme se déroule ensuite comme un algorithme génétique standard. Grâce à sa procédure d'assignement de fitness basée à la fois sur la notion de dominance et la fonction de partage, le NSGA semble le plus approprié à maintenir la diversité de la population et à répartir plus efficacement les solutions sur le front de Pareto. Néanmoins, cet algorithme présente quelques insuffisances en raison de sa complexité de Calcul et de sa sensibilité au choix de la valeur  $\sigma_{shar}$  [18].

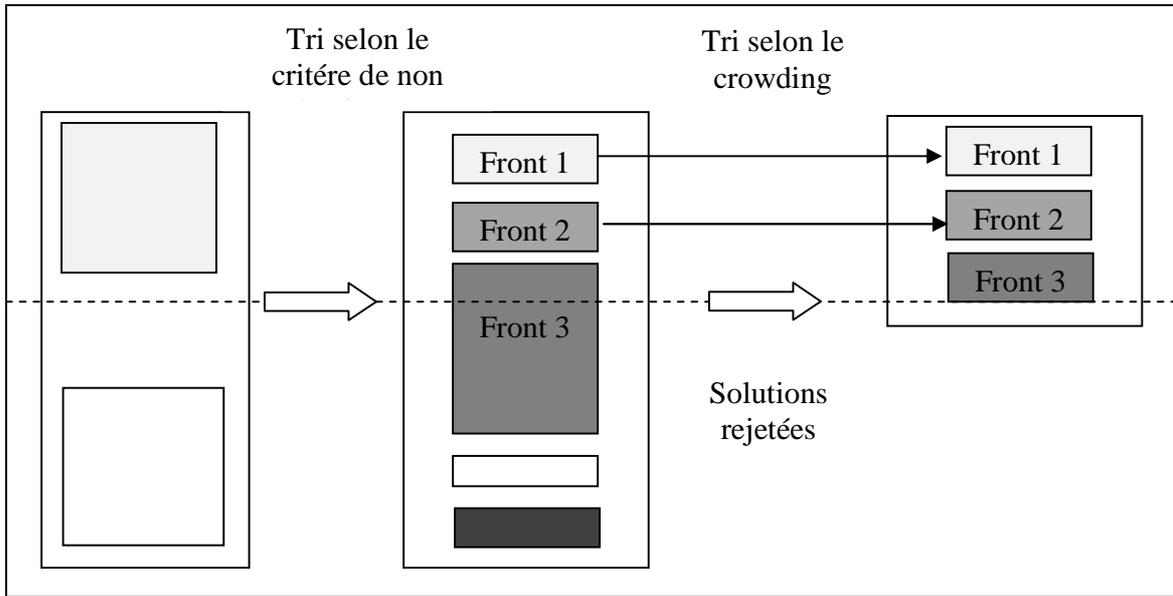
#### 4.3.5 Niche Pareto Genetic Algorithm (NPGA):

Cette méthode proposée par Horn et Nafpliotis (1994) utilise une sélection par tournoi en se basant sur la notion de dominance de Pareto. Le NPGA exécute les mêmes étapes que l'AG standard, la seule chose qui diffère étant la méthode de sélection. A chaque tournoi, deux individus candidats A et B sont pris au hasard dans la population courante. Au lieu de limiter la comparaison aux deux individus (comme c'est le cas pour l'AG standard), une sous population (ou ensemble de comparaison) de taille  $t_{dom}$  est également choisie au hasard. Les deux candidats

sélectionnés sont comparés à chaque individu du sous-groupe. Si l'un des candidats est dominé par l'ensemble de comparaison et le second ne l'est pas, ce dernier est alors positionné dans la population suivante. Dans les autres cas, une fonction de partage est appliquée pour choisir le candidat gagnant. Le paramètre  $t_{dom}$  permet de contrôler la pression de sélection ou de dominance. L'algorithme NPGA est considéré comme étant l'algorithme le plus rapide parmi les approches précédentes car à chaque génération la comparaison n'est appliquée que sur une portion de la population. Le principal inconvénient de cet algorithme est qu'il nécessite, en plus de spécifier le paramètre de sharing  $\sigma_{shar}$ . Un autre paramètre supplémentaire qui est la taille du tournoi  $t_{dom}$  [18].

#### 4.3.6. Non dominated Sorting Genetic Algorithm-II (NSGA-II):

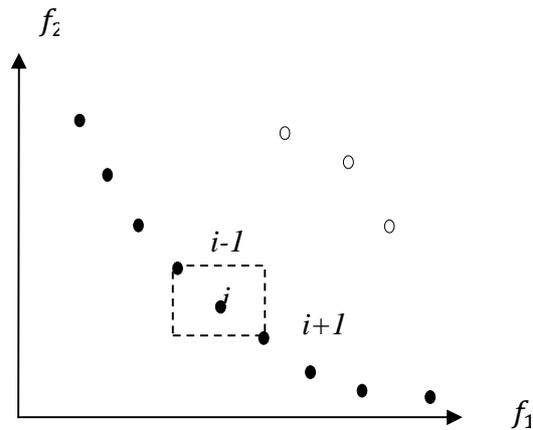
Toutes les méthodes que nous venons de présenter ne conservent pas leurs solutions Pareto-optimales trouvées au cours des générations. Elles sont dites non élitistes. Pour résoudre cette difficulté, de nouvelles techniques ont été appliquées. Nous avons choisi de présenter uniquement le NSGA-II. En proposant le NSGA II, le chercheur Deb (Deb et al. 2002) a tenté de résoudre toutes les critiques faites sur NSGA: non élitiste, complexité de calcul et utilisation de sharing qui implique le réglage d'un ou plusieurs paramètres. Dans cet algorithme, à chaque génération  $t$  une population de parents ( $P_t$ ) de taille  $N$  et une population d'enfants ( $Q_t$ ) de même taille sont assemblées pour former une population ( $R_t$ ) de taille  $2N$ , Cet assemblage permet d'assurer l'élitisme. La population ( $R_t$ ) est ensuite répartie en plusieurs fronts ( $F_1, F_2, \dots$ ) par une procédure de tri, plus rapide que celle proposée dans la première version de NSGA. Une nouvelle population parent ( $P_{t+1}$ ) est formée en ajoutant les fronts au complet (premier front  $F_1$ , second front  $F_2, \dots$  etc) tant que ce ceux-ci ne dépassent pas  $N$ . Si le nombre d'individus présents dans ( $P_{t+1}$ ) est inférieur à  $N$ , une procédure de crowding est appliquée. La figure II.8 illustre le principe de fonctionnement de NSGA-II [18].



**Figure II.8 :** Principe de fonctionnement de NSGA-II.

#### 4.3.6.1. Procédure de Crowding :

Une procédure de crowding, basée sur un calcul de distance (distance de crowding) qui ne nécessite aucun paramétrage et qui est également d'une complexité algorithmique moindre que celle de sharing. La distance de crowding d'une solution particulière  $i$  se calcule en fonction du périmètre de l'hypercube ayant comme sommets les points les plus proches de  $i$  sur chaque objectif. Sur la figure II.9, est représenté l'hypercube en deux dimensions associé au point  $i$ . Le calcul de la distance de crowding nécessite, avant tout, le tri des solutions selon chaque objectif, dans un ordre ascendant. Ensuite, pour chaque objectif, les individus possédant des valeurs limites se voient associés une distance infinie. Pour les autres solutions intermédiaires, on calcule une distance de crowding égale à la différence normalisée des valeurs des fonctions objectifs de deux solutions adjacentes. Ce calcul est réalisé pour chaque objectif. La distance de crowding d'une solution est obtenue en sommant les distances correspondantes à chaque objectif [18]



**Figure II.9:** Distance de Crowding (les points noirs sont des Solutions appartenant au même front)

#### 4.3.7. Strength Pareto Evolutionary Algorithm (SPEA):

En 1998 Zitzler et Thiele proposent une nouvelle méthode d'optimisation multiobjectif qui possède les caractéristiques suivantes:

- ◆ Utilisation du concept de Pareto pour comparer les solutions.
- ◆ Un ensemble de solutions Pareto-optimales est maintenu dans une population externe appelée archive
- ◆ La fitness de chaque individu est calculée par rapport aux solutions stockées dans l'archive
- ◆ Toutes les solutions de l'archive participent 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 paramètres de sharing [17].

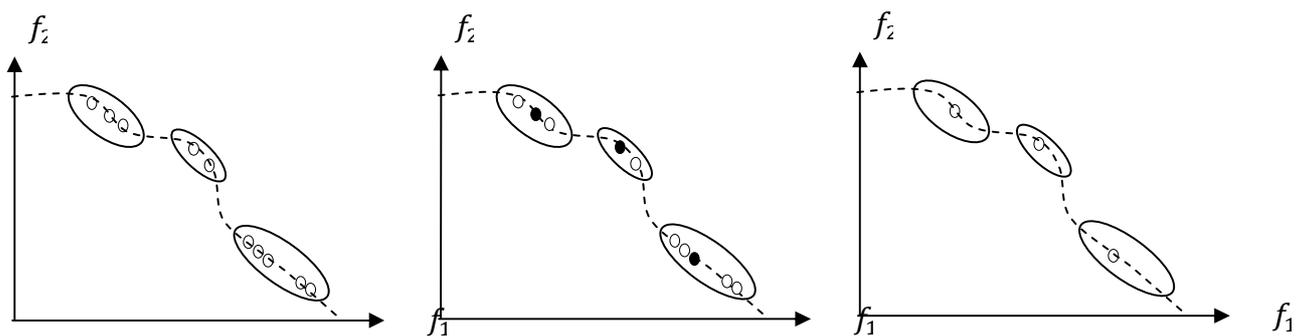
##### 4.3.7.1. Fonctionnement général :

Le passage d'une génération à une autre commence par la mise à jour de l'archive. Tous les individus non dominés sont copiés dans l'archive et les individus dominés déjà présents sont supprimés. Si le nombre d'individus dans l'archive excède un nombre donné, on applique une technique de *clustering* pour réduire l'archive. Ensuite la fitness de chaque individu est mise à jour avant d'effectuer la sélection en utilisant les deux ensembles. Pour terminer, on applique les

opérateurs génétiques de modification. La méthode SPEA implémentée par Zitzler et Thiele est l'illustration même d'un algorithme évolutionnaire élitiste. SPEA maintient un archive externe contenant les meilleurs fronts de compromis rencontré durant la recherche [17].

#### 4.3.7.2. Réduction par clustering :

Lorsque le nombre d'individus de l'archive externe est grand, les performances de l'algorithme peuvent se dégrader significativement. En effet, le nombre d'individus influe sur le calcul de la valeur d'adaptation, celui-ci devient moins fiable, et peut tromper la recherche en la focalisant, par exemple, sur des zones déjà explorées. Pour pallier à ce défaut, une solution consiste à utiliser des techniques de *clustering*. Ces techniques ont été étudiées intensivement dans le contexte des analyses de cluster et ont été appliquées avec succès pour déterminer des partitions d'une collection relativement hétérogène d'éléments. La technique de clustering utilisée par SPEA est assez intuitive. Au début de la procédure, chaque individu constitue son propre groupe, puis on fusionne deux à deux les groupes les plus proches en terme de distance. Cette étape est itérée jusqu'à l'obtention du nombre désiré de groupes. Une fois les groupes identifiés, il ne reste plus qu'à choisir un représentant par groupe. Ce représentant peut être déterminé de plusieurs façons, par exemple en prenant le barycentre du groupe. C'est ce représentant qui sera gardé, les autres éléments étant tout simplement supprimés. Les étapes de la méthode de clustering sont illustrées à la figure II.10 [11].



**Figure II.10:** Illustration du Clustering en dimension 2.

**4.3.8. Pareto Archived Evolution Strategy (PAES):**

Cette méthode a été développée initialement comme une méthode de recherche locale. Les premiers travaux de Knowles et Corne ont montré que cette méthode simple objectif fournissait des résultats supérieurs aux méthodes de recherche basées sur une population. Par conséquent, les auteurs ont adapté cette méthode aux problèmes multiobjectifs. Les particularités de cette méthode sont les suivantes :

- ◆ Elle n'est pas basée sur une population. Elle n'utilise qu'un seul individu à la fois pour la recherche des solutions.
- ◆ Elle utilise une population annexe de taille déterminée permettant de stocker les solutions temporairement Pareto-optimales.
- ◆ L'algorithme utilisé est très simple et inspiré d'une stratégie d'évolution ( $i+1$ ).
- ◆ Elle utilise une technique de crowding basé sur un découpage en hypercubes de l'espace des objectifs [18]

**4.4. Les méthodes hybrides :**

Afin d'améliorer les performances d'un algorithme, on essaye de le combiner avec une autre méthode. Ce principe général appelé hybridation, peut s'appliquer à plusieurs méthodes. Un cas particulier de l'hybridation entre deux méthodes consiste à combiner un algorithme génétique avec une méthode de recherche locale. Dans une telle hybridation on substitue souvent la mutation par une méthode de recherche locale. Dans le cas des problèmes multiobjectifs on peut citer les méthodes hybrides suivantes :

- La méthode MOTS combinant une population et une recherche Tabou,
- La méthode PSA combinant un algorithme génétique et le recuit simulé,
- La méthode M-PAES intégrant un schéma généralisant l'implémentation d'un grand nombre d'algorithmes hybrides pour l'optimisation multiobjectif [16].

**5. Conclusion :**

Nous avons présenté dans ce chapitre les principaux concepts de l'optimisation multiobjectif et les travaux de recherche sur les méthodes d'optimisation de problèmes multiobjectifs. Nous avons vu que cette problématique est divisée en trois approches. La première approche appelée L'approche Métaheuristique, qui utilise des méthodes souvent inspirées par des systèmes naturels comme le recuit simulé et les algorithmes évolutionnaires telle que : les algorithmes génétiques pour apporter une réponse à leurs problèmes. La deuxième approche appelée : L'approches non Pareto, cette approche tente de ramener un problème multiobjectif à un problème simple objectif au risque d'enlever toute signification au problème. La troisième approche est : L'approche Pareto, qui elle adopte un point de vue plus global en prenant en compte l'ensemble des critères et en utilisant la notion de dominance au sens de Pareto. Nous avons vu aussi L'approche hybride qui fait l'hybridation entre deux méthodes différentes.