

# État de l'art des méthodes de décision multicritère

## 2.1 Introduction

La problématique de décision multicritère est souvent présente dans la vie pratique. Du simple choix d'un achat à la sélection d'une carrière, la question demeure la même : comment faire le bon choix en tenant compte de toutes les contradictions qui existent dans les critères qui participent au processus de décision ? La problématique de décision multicritère se réfère à une prise de décision en présence de plusieurs critères, souvent contradictoires. Par exemple, l'achat d'une nouvelle voiture consiste à trouver le ou les meilleurs modèles qui offrent un bon compromis entre le prix, la consommation de carburant, le confort, etc. On remarquera ici que les critères de confort et de niveau de consommation sont souvent contradictoires : une voiture confortable est supposée être spacieuse, par conséquent lourde, ce qui obligera le constructeur à incorporer un moteur puissant adapté aux dimensions de la voiture. On remarquera aussi que les natures de ces deux critères sont totalement différentes : d'un côté le confort est un critère qualitatif alors que le niveau de consommation est quantitatif. De plus, les critères quantitatifs peuvent être incommensurables, c'est-à-dire qu'il n'existe pas de justification permettant de les agréger en un seul critère. La puissance et le prix d'une voiture donnée sont des exemples de deux critères incommensurables, parce que

ramener une puissance à un coût ne répond pas au besoin de l'utilisateur. Ceci dit, la problématique de décision multicritère est d'autant plus complexe qu'elle a engendré de multiples travaux de recherches, qui ont donné naissance à de multiples méthodes de décision multicritère.

Les systèmes autonomes en général et les réseaux autonomes en particulier sont des systèmes complexes qui visent à se distancier d'un opérateur humain pour leur fonctionnement. Dans un contexte de décision multicritère, un système autonome doit prendre des décisions dans des contextes variés, sans l'aide directe d'un décideur. De plus, les réseaux autonomes fonctionnent dans un environnement dynamique, dans lequel des décisions doivent être prises en temps réel. Cet ensemble de contraintes doit être pris en compte par les méthodes de décision multicritère, afin qu'elles puissent être utilisables dans un contexte autonome.

Dans ce chapitre, nous exposerons les principes fondamentaux de l'optimisation multiobjectif, ainsi que de la décision multicritère. Nous illustrerons en détail le fonctionnement des principales méthodes.

## 2.2 Définitions

### 2.2.1 Alternative

Une alternative ou action désigne un objet sur lequel opérera le processus de décision.

### 2.2.2 Critère

Un critère, objectif ou attribut est une fonction définie sur l'ensemble des alternatives, qui prend ses valeurs dans un ensemble totalement ordonné et qui représente les préférences de l'utilisateur selon le point de vue que le critère modélise [RV89].

### 2.2.3 Décision multicritère

Le domaine de la décision multicritère MCDM (*Multicriteria Decision Making*) se décompose en deux sous-domaines, en l'occurrence la décision multiobjectif MODM (*Multi-Objective Decision Making*) et la décision multi-attribut MADM (*Multi-Attribute Decision Making*) [Kla09] :

- MADM : sélection de la meilleure alternative dans un ensemble fini prédéterminé d'alternatives ;
- MODM : sélection de la meilleure action dans un espace de décisions continu ou discret [Kla09]. L'optimisation multiobjectif est une branche de MODM.

### 2.2.4 Problème d'optimisation multiobjectif

Comme nous l'avons introduit ci-dessus, le problème d'optimisation multiobjectif consiste à sélectionner une ou plusieurs solutions qui présentent un bon compromis entre plusieurs critères contradictoires. Mathématiquement parlant, le problème d'optimisation multiobjectif se définit comme suit, dans le cas d'une minimisation :

$$\begin{cases} \text{Minimiser : } \mathbf{F}(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_k(\mathbf{x})]^T \\ \text{s.t. } g_j(\mathbf{x}) \leq 0; j = 1, 2, \dots, m \end{cases}$$

où  $k$  est le nombre de fonctions objectifs et  $m$  est le nombre de contraintes.  $\mathbf{x} \in E^n$  est un vecteur de variables de décision, et  $\mathbf{F}(\mathbf{x}) \in E^k$  est un vecteur de fonctions objectifs  $f_i(\mathbf{x}) : E^n \rightarrow E^1$ .  $\mathbf{X} = \{\mathbf{x}, g_j(\mathbf{x}) \leq 0, j = 1, \dots, m\}$  est l'espace réalisable des actions, et  $\mathbf{C} = \{\mathbf{F}(\mathbf{x}), \mathbf{x} \in \mathbf{X}\}$  est l'espace réalisable des critères.

Contrairement aux problèmes d'optimisation mono-objectif, la solution à un problème multiobjectif est souvent un ensemble de solutions dites solutions non-dominées. Ces solutions sont meilleures que les solutions dominées, mais elles sont incomparables mutuellement selon la relation de dominance. L'ensemble des solutions non-dominées est défini par une relation de dominance. La dominance de Pareto est fréquemment utilisée dans le domaine de l'optimisation multiobjectif, elle est décrite dans la sous-section 2.2.9.

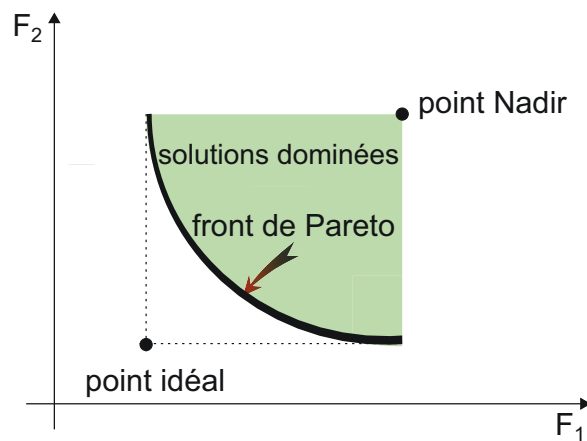


FIGURE 2.1 – Éléments de l'optimisation multiobjectif (cas de deux objectifs)

### 2.2.5 Point idéal

Le point idéal (voir figure 2.1) est un vecteur dont les coordonnées sont les minimums de tous les critères. Il est clair que le point idéal n'est généralement pas atteignable quand les critères sont contradictoires. Le point idéal peut être utilisé comme point de référence que l'on cherche à atteindre.

### 2.2.6 Point Nadir

Les coordonnées du point Nadir correspondent, dans le cas d'une minimisation de tous les critères, aux maxima des critères (voir figure 2.1). Ce point est utile pour normaliser les critères.

### 2.2.7 Décideur

Le décideur est généralement une personne ou un groupe de personnes qui sont supposés connaître le problème de décision multicritère. Le décideur se base sur son expérience et ses connaissances pour exprimer des relations de préférence entre différentes solutions. Dans ce contexte, le décideur est souvent épaulé par un analyste, qui joue le rôle d'interface entre le décideur et l'aspect mathématique du processus de décision multicritère.

### 2.2.8 Fonction d'utilité

Il est souvent supposé qu'un décideur se base sur une certaine fonction pour formuler ses préférences [Mie99]. Cette fonction est appelée fonction d'utilité ou fonction de valeur. Une fonction d'utilité est une fonction  $u : \mathbf{R}^k \mapsto \mathbf{R}$  qui traduit les préférences du décideur comme suit.

Soit  $\mathbf{z}_1$  et  $\mathbf{z}_2$  deux vecteurs de critères :  $\mathbf{z}_1$  est meilleur que  $\mathbf{z}_2$  si  $u(\mathbf{z}_1) > u(\mathbf{z}_2)$ ,  $\mathbf{z}_1$  et  $\mathbf{z}_2$  sont indifférents si  $u(\mathbf{z}_1) = u(\mathbf{z}_2)$ .

Il est à noter que si l'on connaît l'expression mathématique exacte de la fonction d'utilité, nous pouvons traiter le problème décrit dans la section 2.2.4 comme un problème de décision monocritère. Ainsi, nous pouvons appliquer les méthodes d'optimisation classiques pour le résoudre. Cependant, il est *a priori* difficile, voire impossible, de cerner le processus de décision d'un décideur avec une fonction mathématique.

### 2.2.9 Dominance de Pareto

La dominance de Pareto est définie dans l'espace des critères  $\mathbf{C}$ . Soit  $\mathbf{a} = (a_1, \dots, a_k)$  et  $\mathbf{b} = (b_1, \dots, b_k)$  deux vecteurs de critères définis dans l'espace des critères  $\mathbf{C}$ .  $\mathbf{a}$  domine  $\mathbf{b}$  (dans le cas d'une minimisation) si et seulement si :

$$a_i \leq b_i, \forall i \in [1, k] \text{ et } \exists j \in [1, k], a_j < b_j$$

Soit  $\mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_k(\mathbf{x}))$  les fonctions objectifs d'un problème d'optimisation multiobjectif. La solution à ce problème consiste à trouver l'ensemble des vecteurs  $\mathbf{X}^*$  tel que chaque vecteur de critères  $\mathbf{z} = \mathbf{F}(\mathbf{x})$  avec  $\mathbf{x} \in \mathbf{X}^*$  est non dominé. Le front de Pareto  $\mathcal{F}$  est l'ensemble  $\mathcal{F} = \{\mathbf{F}(\mathbf{x}), \mathbf{x} \in \mathbf{X}^*\}$ . La figure 2.1 illustre un front de Pareto dans un espace de critères à deux dimensions.

La problématique de décision multicritère dans ce cas consiste à choisir un point du front de Pareto qui s'apparente le mieux avec les préférences du décideur. Par conséquent, la formalisation des préférences du décideur est un élément crucial du processus de décision multicritère. Cette formalisation est prise en compte par les

différentes méthodes de décision multicritère que nous présentons dans les sections qui suivent.

## 2.3 Classification des méthodes de décision multicritère

Les méthodes de décision multicritère introduites dans la littérature se distinguent les unes par rapport aux autres selon le moment où les préférences du décideur sont utilisées. Dans ce contexte, on distingue trois types de méthodes, à savoir les méthodes à préférence *a priori*, les méthodes à préférence progressive et enfin les méthodes à préférence *a posteriori* [Hor97].

- Les méthodes à préférence *a priori* : dans ce type de méthodes, les préférences du décideur sont spécifiées avant le lancement du processus de décision multicritère. A la fin de l'optimisation, la méthode est supposée fournir une décision qui traduit le mieux les préférences de l'utilisateur. Les méthodes scalaires sont des instances importantes de cette famille. Ces méthodes sont présentées en détail dans la section 2.4.
- Les méthodes à préférence progressive font intervenir le décideur tout au long de l'exécution du processus de décision. En effet, dans ce type de méthodes, le décideur est amené à spécifier ses préférences en fonction de l'évolution du processus de décision. Les méthodes interactives sont des instances de cette famille.
- Les méthodes à préférence *a posteriori* font un travail au préalable, en trouvant les solutions non-dominées qui s'approchent au mieux du front de Pareto. Ensuite, le choix final est laissé au décideur, qui devra sélectionner la meilleure solution non-dominée qui correspond le mieux à ses préférences. Les méthodes évolutionnaires sont des exemples de cette famille (voir section 2.6).

Dans les sections qui suivent, nous illustrons le fonctionnement de quelques méthodes de décision multicritère qui sont largement abordées dans la littérature. Pour une illustration exhaustive, nous invitons le lecteur à consulter les ouvrages [Mie99, CS02, Ehr05, BDM08]. Enfin et afin de ne pas alourdir la présentation, nous considérons que le décideur désire minimiser tous les objectifs/critères du problème traité.

## 2.4 Méthodes scalaires

### 2.4.1 Méthode de la somme pondérée

La méthode de la somme pondérée est à la fois la méthode la plus simple et la plus connue dans le domaine de prise de décision multicritère. Comme son nom l'indique, la méthode de la somme pondérée consiste à pondérer les différents critères du problème de décision multicritère avec des nombres réels appelés poids, qui représentent l'importance de chaque critère dans le processus de décision. Une fois l'importance des différents critères quantifiée, la méthode choisit l'action qui minimise ou maximise la somme pondérée des critères. Mathématiquement parlant, la méthode de la somme pondérée consiste à minimiser une fonction  $u$  définie comme suit :

$$u(\mathbf{x}) = \sum_{i=1}^k w_i f_i(\mathbf{x})$$

Graphiquement, la solution optimale donnée par la méthode de la somme pondérée est le point tangent d'un hyperplan d'équation  $\sum_{i=1}^k w_i f_i(\mathbf{x}) = c$  avec le front de Pareto du problème considéré, tel que  $c$  est minimum et  $\mathbf{x}$  est une action. La figure 2.2, illustre la solution donnée par la méthode de la somme pondérée dans un espace de critères à deux dimensions. La solution donnée par la méthode de la somme pondérée est Pareto optimale si tous les poids utilisés sont strictement positifs [Zad63]. Selon les valeurs des poids  $w_i$ , la méthode peut trouver toutes les solutions appartenant au front de Pareto si l'espace réalisable des critères est convexe [AP96]. En effet, comme le montre la figure 2.3, le point  $\mathbf{c}_1$  qui est situé sur la partie non-convexe du front de Pareto ne minimise pas la fonction  $u$ . Cette situation est problématique quand les points appartenant à

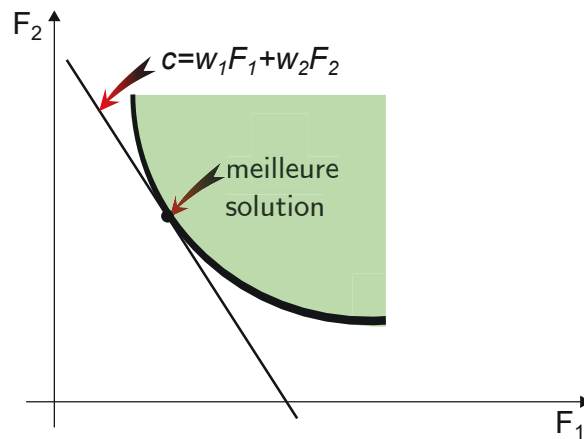
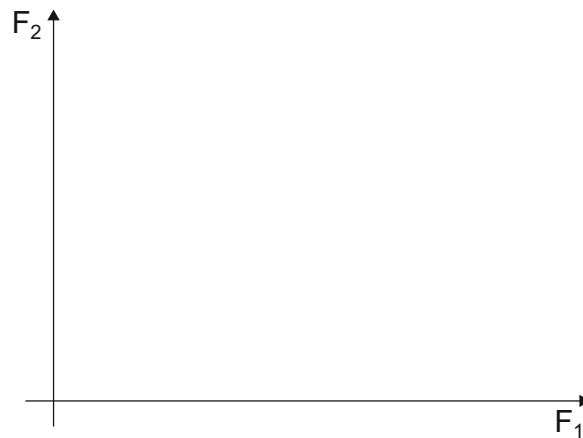


FIGURE 2.2 – Représentation graphique de la méthode de la somme pondérée

FIGURE 2.3 –  $c_1$  est une solution non-dominée non-optimale pour une fonction d'utilité linéaire

la partie non-convexe du front de Pareto sont les solutions qui s'apparentent le mieux aux préférences du décideur.

Bien que la méthode de la somme pondérée soit assez simple et facile à utiliser, elle souffre, néanmoins, d'une difficulté relative au choix des valeurs des poids de pondération. En effet, puisque les poids sont supposés traduire les préférences de l'utilisateur, les performances de toute la méthode en dépendent.

Un problème important de la somme pondérée est la nature linéaire de l'approximation de la fonction d'utilité. En effet, la modélisation des préférences du décideur avec une fonction linéaire peut conduire à des décisions éloignées de ses préférences, en raison d'une modélisation trop grossière. De plus, modéliser les préférences du dé-



cideur par une fonction linéaire suppose que l'importance de chaque critère vis-à-vis des autres est constante, quelles que soient les valeurs prises par les critères. Cependant, cela est rarement le cas dans la pratique. Par exemple, dans le cas de la prise de décision dans des systèmes complexes, l'importance d'un critère donné peut changer en fonction de l'état du système (fonctionnement critique ou normal) [MA10].

### Choix des valeurs des poids

Comme nous l'avons introduit ci-dessus, il est souvent difficile de faire une correspondance directe entre les préférences du décideur et les valeurs des poids de pondération des critères. Néanmoins, des chercheurs ont développé plusieurs méthodes qui ont pour but d'aider le décideur à choisir la valeur des poids. Dans ce contexte, il existe deux classes d'approches, à savoir les approches fondées sur le principe de classement (*ranking methods*) [YH95] et les approches fondées sur le principe de comparaison deux à deux des alternatives (*paired comparison methods*) [SH98].

Dans les approches de classement, le décideur est amené à ordonner les critères selon leur importance, du moins important au plus important. Ensuite, l'approche affecte un poids égal à 1 pour le critère le moins important et des entiers incrémentés successivement pour le reste des poids pondérant les autres critères.

Dans les approches fondées sur le principe de comparaison deux à deux, le décideur est amené à comparer les alternatives deux à deux pour constituer une matrice de comparaison. Dans [SH98], les auteurs proposent d'utiliser le vecteur propre (normalisé à 1) correspondant à la valeur propre maximale de la matrice de comparaison deux à deux comme valeur des poids de pondération des différents critères du problème.

Enfin, dans [RR89], les auteurs introduisent une nouvelle méthode de fixation des poids au moyen de la théorie des ensembles flous.

### 2.4.2 Théorie de l'utilité multi-attribut

La théorie de l'utilité (M.A.U.T, *Multi Attribute Utility Theory*) a fait l'objet de nombreux travaux, dont ceux de Fishburn [Fis70] et de Keeney-Raiffa [KR76]. Les méthodes issues de la théorie de l'utilité sont fondées sur le principe de la fonction

d'utilité. Les méthodes MAUT tentent de modéliser les préférences du décideur par une fonction dite "d'utilité" en utilisant différents outils mathématiques. Ces méthodes sont souvent de type agrégation, dans le sens où la fonction d'utilité résultante est souvent une agrégation de plusieurs "sous-fonctions" qui peuvent être les fonctions objectifs de chaque critère, ou une combinaison de deux ou plusieurs d'entre elles. Mathématiquement parlant, les méthodes MAUT tentent de trouver une fonction  $u$  telle que :

$$u(\mathbf{x}) = f(u_1(f_1(\mathbf{x})), u_2(f_2(\mathbf{x})), \dots, u_k(f_k(\mathbf{x})))$$

où les fonctions  $u_i$  représentent des fonctions d'utilité mono-dimensionnelle de chaque objectif  $f_i$ . Il existe différentes méthodes MAUT, selon le principe de définition des fonctions d'utilité mono-dimensionnelles  $u_i$  ainsi que la fonction d'agrégation  $f$ . Par exemple, dans le cas d'un problème multiobjectif à deux dimensions, nous pouvons citer les fonctions d'utilité suivantes.

$$u(\mathbf{x}) = \alpha_1 u_1(f_1(\mathbf{x})) + \alpha_2 u_2(f_2(\mathbf{x})), \text{ tel que } \alpha_1 > 0, \alpha_2 > 0$$

$$u(\mathbf{x}) = u_1(f_1(\mathbf{x}))^\alpha u_2(f_2(\mathbf{x}))^\beta, \text{ tel que } \alpha > 0, \beta > 0$$

$$u(\mathbf{x}) = \alpha_1 u_1(f_1(\mathbf{x})) + \alpha_2 u_2(f_2(\mathbf{x})) + \alpha_3 (u_1(f_1(\mathbf{x})) - k_1)^\alpha (u_2(f_2(\mathbf{x})) - k_2)^\beta$$

Une fois que la fonction d'utilité est définie, les différentes actions peuvent être comparées au moyen des expressions suivantes.

Soit  $\mathbf{x}$  et  $\mathbf{x}'$  deux actions à comparer :

$$\mathbf{x} \text{ et } \mathbf{x}' \text{ sont "indifférents"} \Leftrightarrow u(\mathbf{x}) = u(\mathbf{x}')$$

$$\mathbf{x} \text{ "est préférable à" } \mathbf{x}' \Leftrightarrow u(\mathbf{x}) > u(\mathbf{x}')$$

En utilisant les expressions ci-dessus, le problème introduit dans la section 2.2.4 est ramené à un problème d'optimisation mono-objectif, défini comme suit :

$$\begin{cases} \text{Minimiser : } u(\mathbf{x}) \\ \text{s.t. } g_j(\mathbf{x}) \leq 0; j = 1, 2, \dots, m \end{cases}$$

### 2.4.3 Méthode des métriques pondérées

La méthode des métriques pondérées (*Method of Weighted Metrics*) sélectionne un vecteur de critères qui minimise une distance à une solution de référence  $\mathbf{R} = (R_1, \dots, R_k)$ . Le vecteur  $\mathbf{R}$  est soit le point idéal, soit une solution de référence fixée par le décideur en fonction de ses préférences [Mie99]. Mathématiquement parlant, la méthode consiste à minimiser une distance  $\Delta_p$  définie comme suit :

$$\Delta_p(\mathbf{x}) = \left( \sum_1^k \omega_i (R_i - f_i(\mathbf{x}))^p \right)^{\frac{1}{p}}$$

Selon la valeur de  $p$ , des cas particuliers de distances bien connues peuvent être utilisées :

1.  $p = 1$  : la distance de Manhattan pondérée

$$\Delta_1(\mathbf{x}) = \sum_1^k \omega_i |R_i - f_i(\mathbf{x})|$$

2.  $p = 2$  : la distance euclidienne pondérée

$$\Delta_2(\mathbf{x}) = \sqrt{\left( \sum_1^k \omega_i (R_i - f_i(\mathbf{x}))^2 \right)}$$

3.  $p = \infty$  : la distance de Tchebychev pondérée

$$\Delta_\infty(\mathbf{x}) = \max_{i \in [1, k]} \omega_i |R_i - f_i(\mathbf{x})|$$

La figure 2.4 donne une représentation graphique des trois distances citées ci-dessus. Du point de vue des relations entre critères, la distance de Manhattan correspond au cas où une compensation entre critères est possible, c'est-à-dire qu'une augmentation de la distance  $\Delta_1$  en raison d'une détérioration d'un critère donné peut être compensée

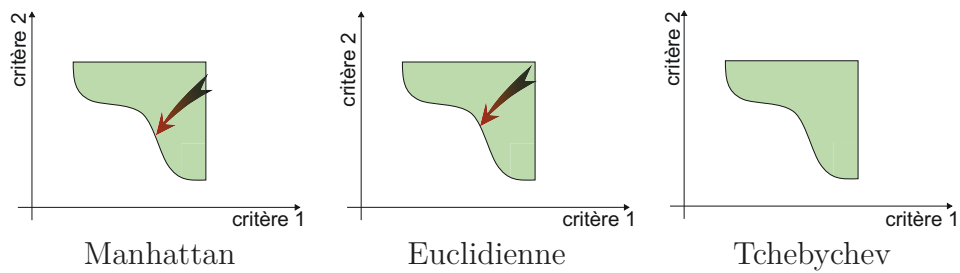


FIGURE 2.4 – Différents types de distances

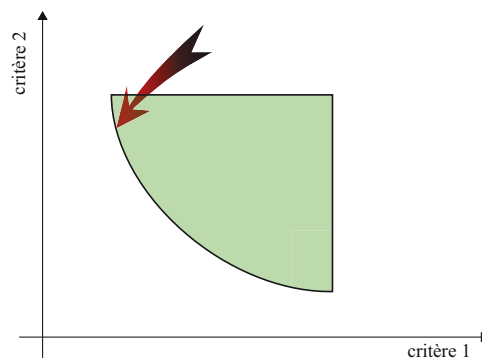


FIGURE 2.5 – Problématique du choix du point de référence

par une amélioration des autres critères. Quant à la distance de Tchebychev, elle correspond à une situation où aucune compensation n'est possible entre critères.

La méthode des métriques pondérées utilise aussi des poids de pondération  $\omega_i$ , ce qui la place au niveau de la méthode de la somme pondérée relativement à la difficulté de traduire les préférences de l'utilisateur par des poids. De plus, la méthode est très sensible à la position du point de référence dans l'espace des critères. Par exemple, la figure 2.5 montre une situation où le point de référence 1 engendre une solution qui minimise uniquement le critère 1 et le point de référence 2 qui minimise uniquement le critère 2. Cela correspond à une optimisation monocritère. Ceci dit, la méthode des métriques pondérées dépend fortement à la fois du choix du point de référence et des valeurs des poids utilisés dans l'expression de la distance. Ce qui rend difficile son utilisation dans un contexte de système autonome.

### 2.4.4 Méthode de programmation par buts

La méthode de programmation par buts (*Goal programming*) [CC61] est une extension de la méthode de la programmation linéaire au cas multiobjectif. Cette méthode se caractérise par la fixation d'un but à atteindre pour chaque critère du problème. La méthode consiste à minimiser les déviations de chaque critère par rapport à son but. Mathématiquement parlant, la méthode de programmation par buts transforme le problème décrit dans la section 2.1 comme suit.

Premièrement, on définit  $k$  valeurs réelles  $F_1, \dots, F_k$  qui représentent les buts à atteindre pour les fonctions objectifs  $f_1, \dots, f_k$  respectivement. Ensuite, on crée  $2 \times k$  variables  $\delta_1^+, \delta_1^-, \dots, \delta_k^+, \delta_k^-$ , appelées variables de déviation, qui représentent le degré de déviation de la solution par rapport aux fonctions objectifs  $f_1, \dots, f_k$  respectivement. Finalement, on obtient le problème suivant :

$$\left\{ \begin{array}{l} \text{Minimiser : } (\delta_1^+ \text{ ou } \delta_1^-, \dots, \delta_k^+ \text{ ou } \delta_k^-) \\ f_1(\mathbf{x}) = F_1 + \delta_1^+ - \delta_1^- \\ \dots \\ f_k(\mathbf{x}) = F_k + \delta_k^+ - \delta_k^- \\ g_j(\mathbf{x}) \leq 0; j = 1, 2, \dots, m \end{array} \right.$$

Les variables de déviation  $\delta_i^+$  et  $\delta_i^-$  doivent vérifier les conditions suivantes :

$$\delta_i^+ \text{ et } \delta_i^- \geq 0, \text{ et } \delta_i^+ \cdot \delta_i^- = 0 \text{ tel que } i \in \{1, \dots, k\}$$

La minimisation des variables  $\delta_i^+$  permet d'atteindre par valeurs supérieures les objectifs  $f_i$ , tandis que minimiser  $\delta_i^-$  permet d'atteindre par valeurs inférieures les objectifs  $i$ .

Dans cette méthode, le problème de décision est ramené à la minimisation d'un vecteur de variables de déviation. Il est par conséquent primordial de définir une fonction d'agrégation des différentes composantes de ce dernier. Par exemple, si l'on désire approcher par valeurs supérieures les objectifs, nous pouvons minimiser la somme pondérée des variables  $\delta_i^+$ . Bien évidemment, on tombe sur le problème difficile du choix des poids que nous avons évoqué dans la section 2.4.1.

### 2.4.5 Méthode AHP

La méthode *AHP* (*Analytic Hierarchy Process*) a été introduite par Saaty en 1980 [Saa80]. Cette méthode tient son avantage de sa similitude au mécanisme de décision de l'être humain, à savoir décomposition, jugement et synthèse. Elle se distingue notamment par sa capacité à gérer différentes classes de critères, à savoir les critères qualitatifs et quantitatifs. De nos jours, elle est très utilisée dans des domaines variés, incluant l'économie, l'écologie, l'industrie, etc. La méthode AHP opère en cinq phases principales :

1. Décomposition hiérarchique du problème de décision initiale : le premier niveau étant l'objectif global du problème, le dernier niveau est formé par les différentes alternatives (actions) offertes au décideur et les niveaux intermédiaires sont occupés par des sous-critères dérivés des critères des niveaux supérieurs.
2. Construction des matrices de comparaison deux à deux : dans cette phase, les sous-critères de chaque niveau, ou alors les alternatives du dernier niveau, sont comparés deux à deux en fonction de leur importance vis-à-vis de l'objectif englobant situé dans le niveau supérieur. Pour ce faire, la méthode AHP utilise un système de comparaison comportant une échelle allant de 1 (importance égale de deux critères) à 9 (un critère est absolument plus important qu'un autre), selon l'importance des sous critères vis-à-vis du critère englobant. Bien évidemment, c'est le décideur qui se charge de cette phase.
3. Établissement des priorités : les priorités de chaque niveau intermédiaire sont représentées par des poids. Elles sont obtenues à partir des principaux vecteurs propres des matrices formées dans la phase 2. Pour ce faire, les valeurs des composantes des principaux vecteurs propres de chaque matrice de comparaison deux à deux sont normalisées à 1. Une valeur proche de 1 signifie que l'élément correspondant est plus prioritaire vis-à-vis du critère englobant.
4. Synthèse des priorités : les priorités finales sont calculées de proche en proche en agrégeant par une somme pondérée les priorités du niveau courant, pondérées par les priorités du niveau supérieur.

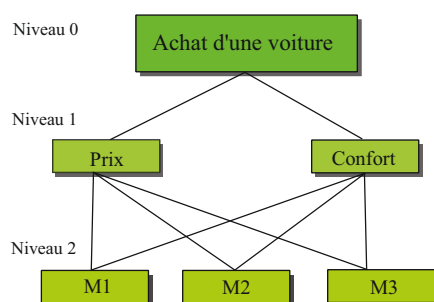


FIGURE 2.6 – Structure hiérarchique d'un problème de décision vu par la méthode AHP

5. Cohérence des jugements : la méthode AHP introduit un paramètre spécial, appelé “ratio de cohérence”, qui est utilisé pour mesurer la cohérence des jugements du décideur (notamment dans la phase de comparaison mutuelle des éléments). En résumé, le ratio de cohérence peut être défini comme la probabilité qu'une matrice de comparaison deux à deux soit complétée aléatoirement.

Pour illustrer le fonctionnement de la méthode, prenons l'exemple de l'achat d'une voiture. L'objectif global du problème est l'achat d'une voiture. Nous considérons deux critères, à savoir le prix (P) et le confort (C). Enfin, nous supposons que le décideur sera amené à choisir entre trois modèles M1, M2 et M3. La figure 2.6 illustre la structure hiérarchique du problème.

Suivant les étapes de la méthode AHP, le décideur est sollicité pour fournir trois matrices de comparaison deux à deux : une matrice  $2 \times 2$  de comparaison des deux critères prix et confort, montrant l'importance des deux vis-à-vis de l'objectif global du problème ; deux matrices  $3 \times 3$  traduisant l'importance des différents modèles M1, M2 et M3 vis-à-vis des deux critères prix et confort pris séparément. Nous supposons que le décideur a fourni les valeurs décrites dans les tableaux 2.1, 2.2 et 2.3. Dans le tableau 2.1, la valeur 5 signifie que le prix est 5 fois plus important que le confort, ce qui justifie la valeur  $\frac{1}{5}$  ligne C colonne P. L'intensité d'importance est donnée par la table établie par Saaty en fonction d'un jugement formulé par le décideur [Saa08].

Comme énoncé ci-dessus, le vecteur des priorités (dernière colonne dans les tableaux 2.1, 2.2 et 2.3) correspond au vecteur propre associé à la valeur propre maximale de chaque matrice de comparaison deux à deux. Par exemple et pour la matrice 2.2, la

	P	C	priorité
P	1	5	0,83
C	$\frac{1}{5}$	1	0,17

TABLE 2.1 – Matrice de comparaison deux à deux du niveau 1

	M1	M1	M3	priorité
M1	1	3	7	0,6586
M2	$\frac{1}{3}$	1	4	0,2628
M3	$\frac{1}{7}$	$\frac{1}{4}$	1	0,0786

TABLE 2.2 – Matrice de comparaison deux à deux du niveau 2 (critère Prix)

valeur propre maximale est  $\lambda_{max} = 3,0324$  et le vecteur propre correspondant est  $W = (0,9232; 0,3683; 0,1102)$ . Enfin, le vecteur des priorités  $W_p = (0,6586; 0,2628; 0,0786)$  est obtenu en divisant chaque coordonnée par la somme des valeurs de toutes les coordonnées.

La consistance des matrices de comparaison deux à deux est déterminée au moyen du “ratio de consistance” CR (*Consistency Ratio*). Le ratio de consistance est le rapport de l’indice de consistance CI (*Concistency Index*) à l’indice de consistance aléatoire RI (*Random Index*).

$$CR = \frac{CI}{RI}$$

tel que,

$$CI = \frac{\lambda_{max} - n}{n - 1}$$

où  $n$  est le nombre de lignes ou colonnes de la matrice de comparaison deux à deux. Concernant le paramètre RI, Saaty a fourni dans son livre [Saa80] des valeurs pour des matrices de comparaison deux à deux de tailles différentes. Par exemple, pour  $n = 3$ ,  $RI = 0,58$ . Par conséquent, le ratio de consistance de la matrice 2.2 est obtenu comme suit :

$$CI = \frac{3,0324 - 3}{3 - 1} = 0,0162$$



	M1	M2	M3	priorité
M1	1	2	4	0,5714
M2	$\frac{1}{2}$	1	2	0,2857
M3	$\frac{1}{4}$	$\frac{1}{2}$	1	0,1429

TABLE 2.3 – Matrice de comparaison deux à deux du niveau 2 (critère Confort)

$$CR = \frac{0,0162}{0,58} = 0,028$$

Selon Saaty [Saa80], une matrice de comparaison deux à deux est consistante si  $RC < 0,1$ . De ce fait, la matrice 2.2 est consistante. Notons que le ratio de consistance doit être calculé pour chaque matrice de comparaison deux à deux afin de vérifier leur consistance. Dans le cas où une inconsistance est détectée, le décideur sera amené à reformuler ses jugements.

Une fois les priorités obtenues, les scores finaux des alternatives sont calculés en faisant une synthèse des scores partiels dans les différents niveaux. Les scores finaux de notre exemple sont donnés comme suit :

$$M1 = 0,83 \times 65,86\% + 0,17 \times 57,14\% = 64,38\%$$

$$M2 = 0,83 \times 26,28\% + 0,17 \times 28,57\% = 26,67\%$$

$$M3 = 0,83 \times 7,86\% + 0,17 \times 14,29\% = 8,95\%$$

Les résultats de cet exemple montrent que le modèle M1 est meilleur que le modèle M2, qui est meilleur que le modèle M3. Ceci est valable pour le décideur qui a fourni les matrices de comparaison deux à deux décrites dans les tableaux 2.1, 2.2 et 2.3.

Notons que cet exemple comprend uniquement deux niveaux. Dans le cas où la structure hiérarchique du problème comprend plusieurs niveaux, les synthèses sont déterminées en séquence, à commencer par le niveau supérieur jusqu'au niveau le plus bas (niveau des actions).

Bien que le processus de décision incarné par la méthode AHP soit intuitif et hiérarchique, il est néanmoins difficile de l'utiliser dans un contexte de système autonome. Ceci est dû principalement à la difficulté de faire intervenir le décideur quand il s'agit

de fournir les matrices de comparaison deux à deux. De plus, il est impossible de faire intervenir le décideur dans des environnements complexes, qui nécessitent des prises de décision en temps réel.

## 2.5 Méthodes de surclassement

Les méthodes de surclassement permettent de prendre en compte l'incomparabilité de certains couples d'actions. La comparaison des couples d'actions permet de construire un graphe des relations entre actions. Ce graphe permet de prendre en compte d'éventuelles relations non transitives, telles que le paradoxe de *Condorcet*. Ce dernier se manifeste, par exemple, quand un ensemble de votants est amené à choisir entre trois actions A, B et C. Dans ce cas, une majorité des votants préfèrent l'action A sur B, une autre majorité préfèrent l'action B sur C et enfin, une autre majorité des votants préfèrent l'action C sur A.

Les méthodes de surclassement permettent de sélectionner une et une seule solution si et seulement si la relation est un ordre total, et un ensemble de solutions dans le cas contraire. Par ailleurs, ce type de méthode est applicable uniquement au cas de problèmes MADM.

Les méthodes de surclassement sont basées sur les comparaisons d'actions deux à deux en vue d'établir une relation de surclassement  $S$  dans un ensemble d'actions ou d'alternatives  $\mathbf{X}$ . Selon B. Roy, une relation de surclassement  $S$  est une relation binaire définie dans l'ensemble des actions  $\mathbf{X}$ , telle que :

$\mathbf{x}S\mathbf{x}'$  si et seulement si il y a "suffisamment d'arguments pour admettre que  $\mathbf{x}$  est au moins aussi bonne que  $\mathbf{x}'$ " et "il n'y a pas de raison importante de prétendre le contraire". De cette définition découlent deux concepts :

- Concordance : pour qu'une action  $\mathbf{x}$  surclasse une autre action  $\mathbf{x}'$  il faut que la majorité (au sens large) des critères dans  $\mathbf{x}$  soient au moins aussi bons que leurs correspondants dans  $\mathbf{x}'$ .
- Discordance : pour qu'une action  $\mathbf{x}$  surclasse une autre action  $\mathbf{x}'$  il faut que  $\mathbf{x}'$  ne présente pas d'aspects, non négligeables, pour lesquels son avantage sur  $\mathbf{x}$  est substantiel.

Dans les sections suivantes, nous présentons les méthodes de surclassement les plus abordées dans la littérature, à savoir les méthodes ELECTRE et PROMETHEE.

### 2.5.1 Méthode ELECTRE

Les méthodes ELECTRE (*EL*imination *Et* *Ch*oix *Traduisant la RE*alité) sont issues des travaux de B. Roy et son équipe de recherche. La première méthode ELECTRE I a été introduite par B. Roy en 1968 [Roy68]. Dans cette section nous présenterons la méthode ELECTRE I en détail, puis nous donnerons un bref aperçu des méthodes ultérieures. Pour un exposé détaillé voir [CS02, RB93, Vin92].

Comme nous l'avons énoncé ci-dessus la méthode ELECTRE se base sur les deux concepts de concordance et discordance afin de classer les actions. Pour ce faire la méthode introduit deux paramètres appelés indice de concordance  $C(\mathbf{x}, \mathbf{x}')$  et indice de non-discordance  $D(\mathbf{x}, \mathbf{x}')$  définis comme suit :

Soit  $\omega_i, i \in [1, k]$  les poids relatifs aux fonctions objectifs  $f_i$  dans le problème de décision. Ces poids sont fournis directement par le décideur.

Soit  $\mathbf{x}$  et  $\mathbf{x}'$  deux actions à comparer :

$$C(\mathbf{x}, \mathbf{x}') = \sum_{i=1}^k \delta_i(\mathbf{x}, \mathbf{x}')$$

Tel que :

$$\delta_i(\mathbf{x}, \mathbf{x}') = \begin{cases} \omega_i & \text{si } f_i(\mathbf{x}) \geq f_i(\mathbf{x}') \\ 0 & \text{sinon} \end{cases}$$

Par construction, comme les poids sont normalisés,  $C(x, x')$  varie entre 0 ( $x$  est dominé par  $x'$ ) et 1 ( $x$  domine  $x'$ ).

$$D(\mathbf{x}, \mathbf{x}') = \begin{cases} \frac{1}{t_i} \max_i (f_i(\mathbf{x}') - f_i(\mathbf{x})) & \text{si } C(\mathbf{x}, \mathbf{x}') \neq 1 \\ 0 & \text{sinon} \end{cases}$$

Tel que :

$t_i$  est l'amplitude de l'échelle associée à la fonction objectif  $f_i$ . De ce fait, on a  $0 \leq D(\mathbf{x}, \mathbf{x}') \leq 1$ .

Une fois les deux indices définis, les deux actions  $\mathbf{x}$  et  $\mathbf{x}'$  peuvent être comparées

comme suit :

$$\mathbf{x}S\mathbf{x}' \iff \begin{cases} C(\mathbf{x}, \mathbf{x}') \geq c \\ D(\mathbf{x}, \mathbf{x}') \leq d \end{cases}$$

où,  $c$  et  $d$  sont les seuils de concordance et de discordance, respectivement. Ce sont des paramètres internes à la méthode. L'expression ci-dessus permet d'extraire de l'ensemble initial des actions un sous-ensemble d'actions qui ne sont surclassées par aucune autre.

Bien que le processus de décision avec la méthode ELECTRE I formalise bien celui du raisonnement humain, la méthode ELECTRE I a l'inconvénient, néanmoins, d'utiliser des poids de quantification de l'importance des différents critères. Par conséquent, malgré l'apport de cette méthode, le décideur est toujours confronté à la tâche difficile de fournir des poids de quantification.

La méthode ELECTRE I est à l'origine de plusieurs autres méthodes, dont le principe fondamental est celui de ELECTRE I, avec quelques extensions.

- ELECTRE II [RB73] : cette méthode remplace la relation de surclassement classique par deux nouvelles relations, à savoir le surclassement fort et le surclassement faible.
- ELECTRE III [Roy78] : cette méthode a introduit la notion de pseudo-critères qui remplacent les critères classiques. Les pseudo-critères sont modélisés par des fonctions dont l'expression est proche des fonctions d'appartenance connues dans le domaine de la logique floue.
- ELECTRE IV [RB93] : cette méthode se distingue par sa capacité à se passer des poids associés à chaque critère. Cependant, cet avantage est tempéré par la nécessité de déterminer un “degré de crédibilité” associé à chaque relation de surclassement utilisée par la méthode.

### 2.5.2 Méthode PROMETHEE

La méthode PROMETHEE (*Preference Ranking Organisation Method for Enrichment Evaluations*) [BMV78] est semblable à la méthode ELECTRE, dans le sens où elle

visé à construire une relation de surclassement dans l'espace des actions ; par contre, elle en diffère par la nature de cette relation. En effet, contrairement à la relation de surclassement construite par la méthode ELECTRE, qui est purement binaire, dans le sens où une relation surclasse ou non une autre, la relation construite par la méthode PROMETHEE est une relation de surclassement évaluée. En d'autres termes, une action surclasse une autre avec une intensité de préférence numérique. De ce fait, la méthode PROMETHEE reprend presque les principes de la théorie de l'utilité multi-attributs, qui associe à une action une valeur d'utilité numérique. La méthode PROMETHEE fonctionne comme suit :

Premièrement, la méthode calcule l'intensité de préférence  $P_i$  d'une action  $\mathbf{x}$  vis-à-vis d'une autre action  $\mathbf{x}'$  sur la fonction objectif  $f_i$  à l'aide de la relation suivante :

$$P_i(\mathbf{x}, \mathbf{x}') = G(d)$$

tel que :

$$d = f_i(\mathbf{x}) - f_i(\mathbf{x}')$$

$G(d)$  est une fonction croissante pour  $d \geq 0$  et nulle pour  $d < 0$ . Par conséquent, une valeur proche de 1 de  $P_i(\mathbf{x}, \mathbf{x}')$  signifie que  $\mathbf{x}$  est nettement préférable à l'action  $\mathbf{x}'$  par rapport à la fonction objectif  $f_i$ . L'inverse est vrai si  $P_i(\mathbf{x}, \mathbf{x}')$  est proche de 0. La fonction  $G$  peut prendre différentes formes et dépendre de certains paramètres au choix du décideur. Le tableau 2.4 montre des formes usuelles de cette fonction. Les paramètres  $p$  et  $q$  introduits dans les figures sont, respectivement, les seuils d'indifférence et de préférence stricte.

Comme dans ELECTRE I, la méthode PROMETHEE utilise aussi des poids  $\omega_i$  pour quantifier l'importance des critères. Les poids sont utilisés pour calculer une moyenne pondérée des intensités de préférence  $P_i(\mathbf{x}, \mathbf{x}')$ , appelée "indicateur de préférence", comme suit :

$$\pi(\mathbf{x}, \mathbf{x}') = \sum_{i=1}^k \omega_i P_i(\mathbf{x}, \mathbf{x}')$$

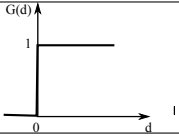
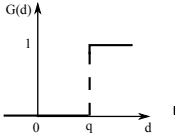
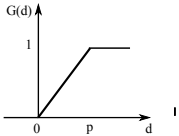
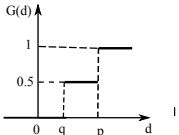
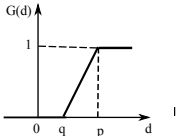
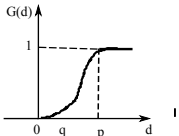
Critère	expression analytique	représentation graphique
Critère usuel	$G(d) = \begin{cases} 0 & \text{si } x = 0 \\ 1 & \text{si } x \neq 0 \end{cases}$	
Quasi-Critère	$G(d) = \begin{cases} 0 & \text{si } 0 \leq x \leq q \\ 1 & \text{si } x > q \end{cases}$	
Critère à préférence linéaire	$G(d) = \begin{cases} \frac{x}{p} & \text{si } 0 \leq x \leq p \\ 1 & \text{si } x > p \end{cases}$	
Critère à niveaux	$G(d) = \begin{cases} 0 & \text{si } 0 \leq d \leq q \\ \frac{1}{2} & \text{si } q < d \leq p \\ 1 & \text{si } d > p \end{cases}$	
Critère linéaire et zone d'indifférence	$G(d) = \begin{cases} 0 & \text{si } 0 \leq d \leq q \\ \frac{d-q}{p-q} & \text{si } q < d \leq p \\ 1 & \text{si } d > p \end{cases}$	
Critère gaussien	$1 - \exp\left(-\frac{d^2}{2\sigma^2}\right)$	

TABLE 2.4 – Différentes formes de critères dans la méthode PROMETHEE

Ainsi,  $\pi(\mathbf{x}, \mathbf{x}') \rightarrow 0$  signifie une faible préférence globale de  $\mathbf{x}$  par rapport à  $\mathbf{x}'$ , et  $\pi(\mathbf{x}, \mathbf{x}') \rightarrow 1$  traduit une préférence globale forte de  $\mathbf{x}$  sur  $\mathbf{x}'$ .

L'ensemble des indicateurs de préférence est représenté par un graphe valué complet appelé "graphe de surclassement". Ce graphe est utilisé pour calculer les flux entrant  $\phi^-$  et sortant  $\phi^+$  comme suit :

$$\phi^-(\mathbf{x}') = \sum_{\mathbf{x} \in \mathbf{X}} \pi(\mathbf{x}, \mathbf{x}')$$

$$\phi^+(\mathbf{x}') = \sum_{\mathbf{x} \in \mathbf{X}} \pi(\mathbf{x}', \mathbf{x})$$

Les flux entrant et sortant représentent les intensités des avantages et des désavantages qu'une action  $\mathbf{x}'$  possède vis-à-vis de l'ensemble des autres actions candidates. Les flux entrant et sortant sont utilisés différemment par les méthodes PROMETHEE I et PROMETHEE II afin de définir des relations de pré-ordre.

### – PROMETHEE I

La méthode PROMETHEE I propose un pré-ordre partiel (permet l'incomparabilité) défini comme suit :

$$\left\{ \begin{array}{l} \mathbf{x}P\mathbf{x}' \text{ si } \left\{ \begin{array}{l} \phi^+(\mathbf{x}) > \phi^+(\mathbf{x}') \text{ et } \phi^-(\mathbf{x}) < \phi^-(\mathbf{x}') \\ \text{ou } \phi^+(\mathbf{x}) > \phi^+(\mathbf{x}') \text{ et } \phi^-(\mathbf{x}) = \phi^-(\mathbf{x}') \\ \text{ou } \phi^+(\mathbf{x}) = \phi^+(\mathbf{x}') \text{ et } \phi^-(\mathbf{x}) < \phi^-(\mathbf{x}') \end{array} \right. \\ \mathbf{x}I\mathbf{x}' \text{ si } \phi^+(\mathbf{x}) = \phi^+(\mathbf{x}') \text{ et } \phi^-(\mathbf{x}) = \phi^-(\mathbf{x}') \\ \mathbf{x}R\mathbf{x}' \text{ sinon} \end{array} \right.$$

où les notations  $P$ ,  $I$  et  $R$  signifient Préférence, Indifférence et Incomparabilité, respectivement.

### – PROMETHEE II

La méthode PROMETHEE II permet de construire un pré-ordre total (exclut l'incomparabilité et réduit considérablement l'indifférence) défini comme suit :

Soit :  $\phi(\mathbf{x}) = \phi^+(\mathbf{x}) - \phi^-(\mathbf{x})$

$$\begin{cases} \mathbf{x}P\mathbf{x}' & \text{si } \phi(\mathbf{x}) > \phi(\mathbf{x}') \\ \mathbf{x}I\mathbf{x}' & \text{si } \phi(\mathbf{x}) = \phi(\mathbf{x}') \end{cases}$$

On remarque que la méthode PROMETHEE II se rapproche nettement des méthodes d'utilité, la mesure  $\phi(\mathbf{x})$  étant l'utilité accordée à l'action  $\mathbf{x}$ . Par contre, la méthode PROMETHEE I garde toujours sa filiation aux méthodes de surclassement, grâce aux flux entrant et sortant qui sont relativement similaires aux notions de concordance et de discordance utilisées dans les méthodes ELECTRE.

## 2.6 Méthodes évolutionnaires

Les méthodes évolutionnaires, principalement les méthodes basées sur les algorithmes génétiques (AG), s'inscrivent dans la famille des méthodes *a posteriori*, dans le sens où le but est de constituer un ensemble de solutions non-dominées qui seront soumises ensuite aux préférences du décideur. Dans cette section, nous présentons brièvement la notion d'algorithme génétique, qui constitue le coeur de ces méthodes, puis nous donnerons un bref aperçu de la version multiobjectif de ce dernier. Pour une présentation détaillée, voir [DPST03].

### 2.6.1 Algorithmes génétiques

Les AGs ont été introduits par Holland en 1975. Ils sont inspirés de la théorie de l'évolution des espèces vivantes. La théorie de l'évolution stipule que les êtres vivants sont constamment soumis au mécanisme de compétition qui sélectionne les individus les plus adaptés à leur environnement. De ce fait, les individus les plus robustes (plus adaptés à leur environnement) auront plus de chance de se reproduire et de transmettre ainsi leurs gènes favorables, qui ont permis leur survie, à leurs enfants. Par conséquent, au fil des générations, les individus qui portent une combinaison de gènes adéquate formeront la majorité au sein de la population. De plus, de nouveaux individus peuvent apparaître au cours de l'évolution, grâce à la variation aléatoire des gènes, ce qui peut



enrichir davantage la population avec de forts individus. Les AGs utilisent les concepts suivants, afin de résoudre des problèmes d'optimisation complexes :

- Individu : représente une solution potentielle au problème d'optimisation. Un individu est composé de plusieurs gènes. Un gène est codé par un bit (0 ou 1) ou un entier ou encore un nombre réel, selon la représentation choisie pour traiter le problème ;
- Génotype : ensemble des caractéristiques héréditaires d'un individu ;
- Population : l'ensemble des individus traités simultanément par l'algorithme génétique ;
- Fonction d'adaptation (*fitness*) : une fonction qui associe à chaque individu une valeur numérique représentant sa qualité ;
- Génération : une itération de l'AG qui enchaîne l'application sur la population des opérateurs de sélection, mutation, croisement et remplacement décrits ci-dessous ;
- Sélection : associe à chaque individu une probabilité de reproduction à chaque génération. La probabilité de reproduction est souvent corrélée avec la qualité de l'individu. Ainsi, les meilleurs individus ont de fortes chances d'être sélectionnés pour se reproduire ;
- Remplacement : sélectionne l'ensemble des individus qui seront conservés au début d'une nouvelle génération ;
- Mutation : c'est un opérateur de variation qui modifie aléatoirement les gènes d'un individu ;
- Croisement : c'est un opérateur de variation qui combine un ou plusieurs individus parents pour former plusieurs individus enfants ;
- Élitisme : c'est un opérateur qui permet de préserver les meilleurs individus de la population à travers les différentes générations ;
- Nichage : opérateur qui permet de diversifier les individus au sein d'une population. En d'autres termes, l'opérateur de nichage permet d'éviter une situation où les copies d'un seul individu composent la majorité au sein de la population.

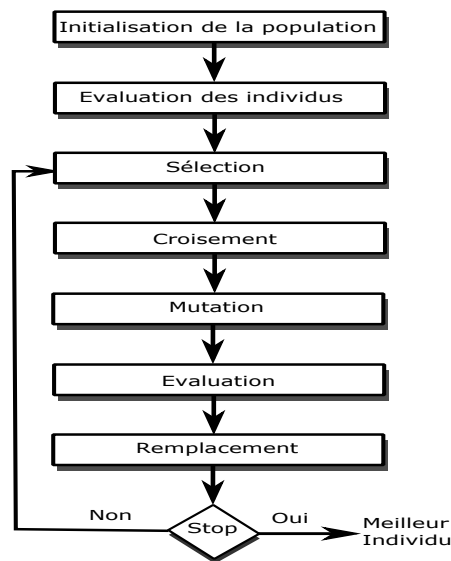


FIGURE 2.7 – Fonctionnement d’un algorithme génétique

La figure 2.7 illustre le schéma général de fonctionnement d’un algorithme génétique. D’abord, un ensemble de solutions ou individus (population initiale) est généré au début de l’optimisation. Généralement, les individus sont choisis par un tirage aléatoire au niveau de chaque gène. Ensuite, les individus sont évalués au moyen de la fonction d’adaptation (*fitness*) afin de déterminer leurs performances. Puis, l’algorithme génétique entre dans une boucle générationnelle (chaque itération étant une génération) dans laquelle les phases de sélection, croisement, mutation, évaluation et reproduction sont exécutées en séquence. Enfin, l’AG s’arrête en fonction d’un critère d’arrêt donné, tel qu’un nombre déterminé d’itérations, un temps donné d’exécution, stabilisation de la population, etc. A la fin de l’optimisation, le ou les meilleurs individus de la dernière génération sont retenus comme solutions du problème traité.

### 2.6.2 Algorithme génétique multiobjectif

Nous avons vu dans la sous-section précédente qu’un AG manipule plusieurs solutions à la fois afin de maximiser la probabilité de tomber sur l’optimum global du problème d’optimisation traité. Cette caractéristique est d’autant plus importante dans un contexte multiobjectif car elle permet à l’AG de bien échantillonner le front de Pareto. De nos jours, les AGs sont très utilisés pour résoudre les problèmes d’optimisation

multiobjectif. Des chercheurs publient constamment des travaux relatifs à ce domaine.

La différence principale entre un AG mono-objectif et un AG multiobjectif réside dans la phase d'évaluation de l'adaptation (*fitness*) des individus, particulièrement la manière de gérer les contradictions qui existent entre les différents objectifs. Dans ce contexte, on distingue deux approches principales, en l'occurrence l'approche élitiste et l'approche non-élitiste [Deb01]. Nous donnons, ci-dessous, un aperçu de ces deux approches.

### Approches non-élitistes

L'algorithme VEGA [Sch85] (*Vector Evaluated Genetic Algorithm*) est la première tentative de concevoir un AG multiobjectif. VEGA exploite individuellement les fonctions objectifs partielles pour déterminer l'efficacité des individus. L'algorithme procède comme suit : premièrement, l'ensemble des individus de la population est partitionné en un nombre de groupes égal au nombre de fonctions objectifs partielles. Ensuite, on associe une fonction objectif partielle à chaque groupe ; par conséquent, l'efficacité d'un individu est déterminée par une seule fonction objectif. Ensuite, tous les individus sont mélangés, et les phases d'un AG classiques sont exécutées. Ce processus est répété tout au long des itérations d'évolution de l'AG.

L'algorithme VEGA souffre d'un biais engendré par les individus les plus forts sur chaque objectif individuel, ce qui concentre la recherche sur des zones particulières du front de Pareto. Pour remédier à ce problème, David E. Goldberg propose dans son livre [Gol89] d'utiliser la notion de dominance dans la fonction d'adaptation pour diriger la recherche vers le front de Pareto, et le mécanisme de nichage qui permet de répartir uniformément les individus sur toute la surface du front de Pareto. Cette contribution a permis le développement de plusieurs approches, qui diffèrent les unes des autres par la façon de formuler la fonction d'adaptation.

Dans [FF<sup>+</sup>93], l'adaptation d'un individu est mesurée par son rang dans la liste de dominance. Le rang d'un individu est associé au nombre d'individus qui le dominent au sens de Pareto. Dans le cas d'une minimisation, on affecte un rang égal à 1 pour les individus non-dominés, puis des rangs supérieurs à 1 aux individus dominés. Ensuite,

les rangs sont ramenés à des valeurs d'efficacité en leur appliquant une interpolation à l'aide d'une fonction décroissante.

Dans [HNG94], les auteurs utilisent la notion de tournoi basée sur la dominance de Pareto pour sélectionner les individus dans la phase de remplacement de l'algorithme *Niched Pareto GA* (NPGA). D'abord, un ensemble d'individus (ensemble de comparaison) est sélectionné à partir de la population courante. Ensuite, une paire d'individus est choisie aléatoirement dans le reste de la population. Ces individus sont comparés avec les individus de l'ensemble de comparaison. Si l'un des individus domine et l'autre est dominé, alors l'individu dominant est sélectionné. Si les deux individus dominent ou sont dominés par les individus de l'ensemble de comparaison, alors l'individu qui est sélectionné est celui qui a le moins de voisins dans une boule de rayon donné dans l'espace des critères.

L'algorithme NSGA (*Non-dominated Sorting Genetic Algorithm*) introduit dans [SD94] calcule la fonction d'adaptation des individus comme suit : d'abord, l'ensemble de la population courante est trié pour trouver le premier sous-ensemble des individus non-dominés. L'adaptation partagée de ces individus est obtenue en appliquant la méthode de nichage dite "du partage" [GR87] avec une adaptation brute prise égale à la taille de la population. La méthode du partage nécessite la détermination d'un rayon de niche, ce qui est un paramètre difficile à déterminer. Ensuite, le second sous-ensemble des individus non-dominés est déterminé en ignorant les individus appartenant au premier sous-ensemble. Les individus du deuxième sous-ensemble auront une adaptation inférieure à l'adaptation partagée du plus mauvais individu du premier sous-ensemble des individus non-dominés. Ce processus est réitéré jusqu'à ce que tous les individus de la population se voient attribuer une valeur d'adaptation.

### Approches élitistes

Le mécanisme d'élitisme permet à un AG de préserver les meilleurs individus rencontrés au fil des générations. Ce mécanisme est efficace pour diriger la recherche vers le front de Pareto. La seconde génération des algorithmes génétiques multiobjectif se base sur ce principe afin d'améliorer leurs performances. Nous décrivons brièvement,

dans cette section, deux approches élitistes très connues, en l'occurrence NSGA-II [DAPM00] et *Strength Pareto EA* (SPEA) [ZT98].

L'algorithme SPEA maintient les individus non-dominés découverts au fil des générations dans un ensemble particulier, appelé archive. Les individus de l'archive participent à l'évolution de la population et permettent, par conséquent, de diriger la recherche vers le front de Pareto. Au début de chaque itération de l'algorithme, les individus de l'archive sont mélangés avec la population courante. Ensuite, les adaptations des individus sont calculées comme suit : les individus non-dominés auront une adaptation égale au nombre d'individus qu'ils dominent, divisée par la taille de la population, plus un. Quant aux individus dominés, ils auront une adaptation égale à la somme des adaptations des individus qui les dominent, plus un. La fonction d'adaptation ainsi définie doit être minimisée. Ceci permet d'orienter la recherche vers les individus non-dominés. Dans [ZLT02], les auteurs proposent une amélioration de l'algorithme SPEA, appelée SPEA2, qui diffère de SPEA au niveau du calcul de la fonction d'adaptation.

NSGA-II est une amélioration de l'algorithme initial NSGA avec le principal objectif d'ajouter l'élitisme, qui améliore beaucoup les performances. De plus, La complexité de NSGA-II est de l'ordre de  $O(mN^2)$ , alors que celle de NSGA est de l'ordre de  $O(mN^3)$ , où  $N$  est la taille de la population et  $m$  est le nombre d'objectifs. Enfin, la méthode ne nécessite plus la détermination d'un rayon de niche.

### 2.6.3 Application aux systèmes autonomes

Les méthodes évolutionnaires sont difficiles à utiliser dans un contexte de systèmes autonomes car le but des AGs est de fournir un ensemble de solutions qui seront soumises au décideur, afin qu'il choisisse la solution appropriée selon ses préférences. Des approches récentes [DS06] permettent de lever une partie de cette problématique en combinant les algorithmes génétiques avec les méthodes de décision multicritère classiques, telles que celles qui utilisent un point de référence [Wie80]. Ces méthodes permettent de réduire l'ensemble des solutions proposées à une seule proposition qui s'accorde le mieux avec les préférences du décideur. Cependant, ces nouvelles approches héritent des problèmes des méthodes de décision multicritère classiques, notamment le

choix du point de référence et la détermination de la valeur des poids de quantification de l'importance des critères dans le processus de décision multicritère.

## 2.7 Conclusion

Il est souvent difficile de résoudre des problèmes de décision mono-critère, car ces problèmes issus du monde réel sont souvent de complexité élevée. Cette difficulté est accrue si plusieurs critères contradictoires sont associés au problème de décision. En effet, en plus de l'explosion combinatoire, il faut gérer les préférences du décideur afin de trouver une solution satisfaisante. Par sa complexité, le domaine de la décision multicritère a engendré une multitude de méthodes diverses selon leur principe d'incorporation des préférences du décideur dans le processus de décision.

Dans ce chapitre, nous avons considéré les éléments essentiels de la décision multicritère. Nous avons exposé les différentes méthodes qui essaient de trouver les solutions non-dominées d'un problème d'optimisation multiobjectif ou de sélectionner une solution qui s'accorde le mieux avec les préférences du décideur. Dans ce contexte, nous avons constaté que la majorité de ces méthodes utilisent des paramètres spécifiques afin de déterminer l'importance de chaque critère dans le processus de décision. Ces paramètres sont soit donnés directement par le décideur, soit obtenus par des matrices de comparaison deux à deux fournies aussi par le décideur. Par ailleurs, les méthodes évolutionnaires permettent d'obtenir un échantillon de solutions les plus proches possibles du front de Pareto. La meilleure solution est sélectionnée par le décideur. Par conséquent, ces procédés sont inadéquats dans un contexte autonome car le système est supposé fonctionner indépendamment d'un décideur.

Le chapitre suivant introduit une approche multicritère conçue pour un contexte autonome.