

# Une approche multicritère pour les systèmes autonomes

## 3.1 Introduction

Les méthodes de décision multicritère ou d'aide à la décision multicritère permettent de sélectionner un ensemble de solutions alternatives à un problème de décision multicritère. Ce travail doit être complété par le décideur, qui doit choisir la meilleure alternative qui correspond le mieux à ses préférences. Cependant, il est difficile d'estimer les paramètres de ces méthodes qui, souvent, utilisent des poids pour quantifier l'importance de chaque critère dans le processus de décision. De ce fait, quand les alternatives offertes par la méthode ne sont pas conformes aux préférences du décideur, une interaction entre la méthode et ce dernier est indispensable afin d'affiner le processus de décision.

Les systèmes autonomes sont des systèmes autoadaptatifs qui sont capables d'apprendre le comportement de leur environnement afin de s'auto-configurer et de prendre des décisions optimales, sans avoir recours à une aide externe. Par conséquent, l'interaction avec un décideur pour la prise de décision multicritère est impossible pour ce type de systèmes. Les méthodes de décision multicritère classiques ne sont pas adéquates dans ce contexte, car la majorité d'entre elles se basent sur l'interaction avec le décideur afin d'affiner le processus de décision.

Les systèmes dynamiques adaptent constamment leur état, ainsi que leurs paramètres, sous différentes contraintes temporelles. Le front de Pareto de ce type de systèmes peut varier considérablement en fonction du temps, ce qui rend difficile la prise de décision multicritère, particulièrement dans le cas où les contraintes de temps sont en dessous des temps de réaction de l'être humain. Par conséquent, il est difficile, voire impossible, de compter sur l'interaction avec le décideur afin d'adapter constamment les paramètres des méthodes de décision multicritère classiques.

Dans ce chapitre, nous présenterons une nouvelle méthode de décision multicritère adaptée aux systèmes autonomes, ainsi qu'aux systèmes dynamiques. Ce chapitre est organisé comme suit : premièrement, nous introduisons dans la section 3.2 le principe général de la "méthode de la poupée russe". Ensuite, une description détaillée, comportant des définitions formelles, ainsi que l'exposé des différentes variantes de la méthode, est donnée dans la section 3.3. Enfin, la section 3.4 conclut ce chapitre.

## 3.2 Principe général de la méthode

L'objectif de la méthode consiste à construire des fonctions d'utilité en fonction d'un ensemble de paramètres fournis par le concepteur du système autonome. Nous rappelons que le concepteur du système diffère d'un décideur, dans le sens où le concepteur agit au moment de la conception d'un système en fonction de données préétablies, contrairement au décideur, qui agit au moment de la prise de décision.

Il existe plusieurs méthodes qui permettent au concepteur de définir les paramètres de la méthode de décision multicritère. Par exemple, il est commun d'évaluer la qualité d'une application multimédia, telle que la voix sur IP VoIP (*Voice over IP*) au moyen d'un sondage d'opinion appelé MOS (*Mean Opinion Score*) [Uni96]. Le MOS consiste à réunir l'opinion d'un ensemble d'utilisateurs sur plusieurs passages d'une application donnée (des séquences d'une communication VoIP, par exemple). L'opinion consiste, par exemple, à attribuer une note comprise entre zéro et cinq, en fonction de la qualité perçue. Cinq étant la qualité excellente et zéro la plus mauvaise qualité. La figure 3.1, illustre les résultats qui peuvent être obtenus au moyen d'un MOS (notons que cette figure est donnée à titre illustratif seulement, puisqu'elle ne provient d'aucune expé-

rience réelle). Dans cet exemple, la qualité de l'application VoIP est évaluée en fonction de deux métriques, à savoir le délai et le taux de perte. Nous remarquons, à travers la figure, que la qualité de la VoIP varie entre "excellente", "bonne", "acceptable" et "mauvaise". Suivant cette observation, nous pouvons décomposer l'espace des critères à l'aide d'un ensemble de boîtes englobantes, semblables à des poupées russes (voir figure 3.1). La boîte numéro 3 est définie comme la boîte qui couvre l'ensemble de l'espace des critères. Selon les observations illustrées dans la figure 3.1, la boîte 0 englobe les vecteurs de critères traduisant une excellente qualité de la VoIP. Les vecteurs de critères qui sont dans la boîte 1 et n'appartiennent pas à la boîte 0 traduisent une qualité bonne. Les vecteurs de critères qui sont dans la boîte 2 et n'appartiennent pas à la boîte 1 traduisent une qualité acceptable. Enfin, les vecteurs de critères restants traduisent une qualité mauvaise.

La figure 3.2 illustre un front de Pareto dans un espace de recherche formé par les deux critères délai et taux de perte. Cette figure montre que les boîtes 0 et 1 ne contiennent aucune solution appartenant au front de Pareto. Par conséquent, il existe au mieux des solutions *acceptables* pour cet exemple de front de Pareto. De ce fait, la solution choisie est la plus proche par rapport à la boîte de bonne qualité (boîte 1), selon une distance  $\Delta(\mathbf{x})$  ( $\mathbf{x}$  est un vecteur de critères) qui est à définir. Cette distance est utilisée pour définir la fonction d'utilité représentée par la méthode de la poupée russe.

La distance  $\Delta(\mathbf{x})$  est donnée dans l'ensemble des boîtes de qualité avec des valeurs allant de 0 à  $m$ , où  $m$  est l'index de la plus grande boîte qui englobe tout l'espace de critères.  $\Delta(\mathbf{x}) = 0$  signifie que le vecteur  $\mathbf{x}$  est dans la boîte 0 qui est la boîte de la meilleure qualité possible. Cette distance permet de définir une fonction d'utilité  $u$ , telle que  $u(\mathbf{x}) = m - \Delta(\mathbf{x})$ . De ce fait, la fonction d'utilité  $u$  est maximale pour tout  $\mathbf{x}$  appartenant à la boîte 0.

Plusieurs formes de boîtes peuvent être utilisées afin de construire une fonction d'utilité  $u$  qui est capable d'approcher une relation de qualité  $Q(\mathbf{R}^n, \mathbf{R})$ , où  $\mathbf{R}^n$  désigne l'ensemble des vecteurs de critères et  $\mathbf{R}$  est l'ensemble de définition des qualités. Pour notre part, nous avons choisi de représenter une boîte par l'intersection d'un hyper-

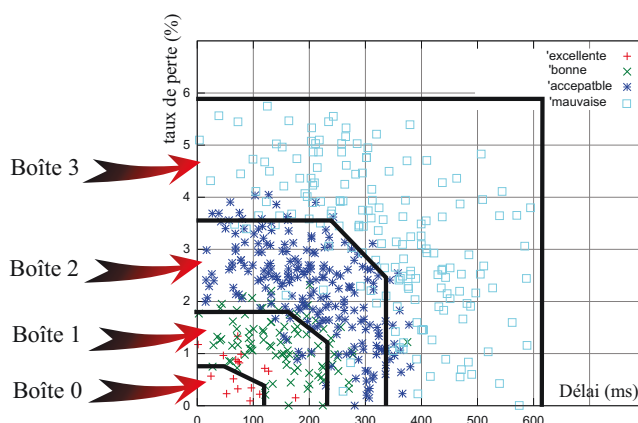


FIGURE 3.1 – Évaluation de la qualité de la VoIP en fonction du délai et du taux de perte

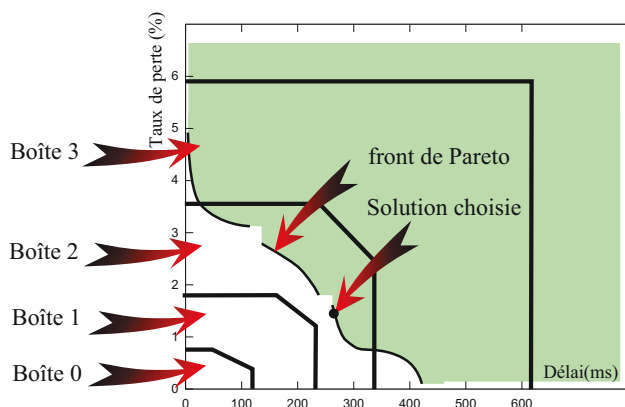


FIGURE 3.2 – Processus de décision via la méthode de la poupée russe

rectangle avec un simplexe. Ce choix est dû à des considérations que nous expliquerons dans la section 3.3.

Dans cette section, nous avons donné une idée générale de la méthode multicritère proposée. Les sections suivantes présentent en détail le fonctionnement de la méthode de la poupée russe.

### 3.3 Définition formelle de la méthode

Soit  $n$  le nombre de critères du problème. Sans perte de généralité, nous considérons, dans ce chapitre, le cas de la minimisation de tous les critères. De plus, nous supposons que tous les critères sont positifs ; dans le cas où un ou plusieurs critères ne le sont pas,

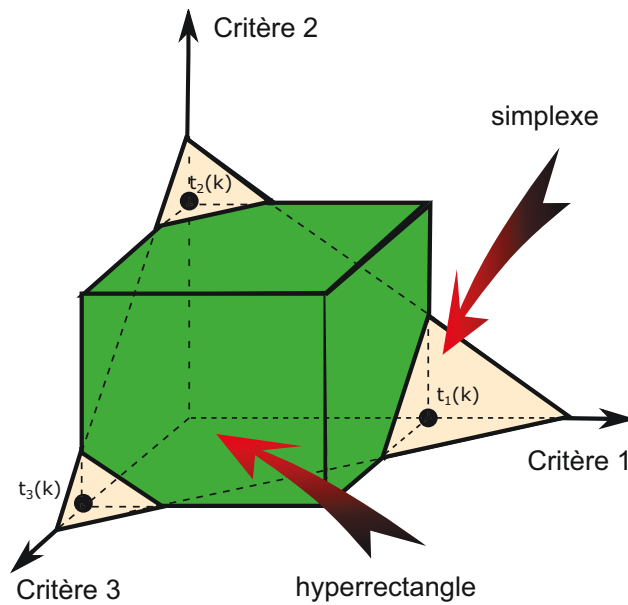


FIGURE 3.3 – Construction d’une boîte de qualité

une transformation affine est appliquée, afin de les ramener au quadrant positif. Un ensemble de  $m + 1$  boîtes indexées de 0 à  $m$  est défini dans l’espace de critères, de telle sorte que chaque boîte, excepté la boîte  $m$ , soit englobée dans une autre, comme une poupée russe (voir la figure 3.1). La boîte  $m$ , qui englobe tout l’espace des critères, est le produit cartésien de  $n$  intervalles  $[0, u_i]$  tel que  $u_i$  est la valeur maximale du critère  $i$ . Chacune des  $m$  autres boîtes restantes est formée par l’intersection de  $2n + 1$  demi-espaces dont les bordures sont des hyperplans. Les hyperplans  $\mathbf{H}_i(k)$ ,  $1 \leq i \leq n$  sont orthogonaux aux axes et l’hyperplan  $\mathbf{H}_{n+1}(k)$  est oblique (voir la figure 3.3). Les hyperplans orthogonaux aux axes définissent un hyperrectangle et l’hyperplan oblique définit un simplexe. Ainsi, nous pouvons aussi définir une boîte par l’intersection d’un hyperrectangle avec un simplexe (voir la figure 3.3). Cette façon de construire une boîte de qualité permet d’obtenir une structure qui est capable de représenter différentes relations entre critères, notamment la compensation et la non-compensation (voir la figure 3.4). Les sous sections 3.3.3 et 3.3.4 présentent en détail ces deux types de relations.

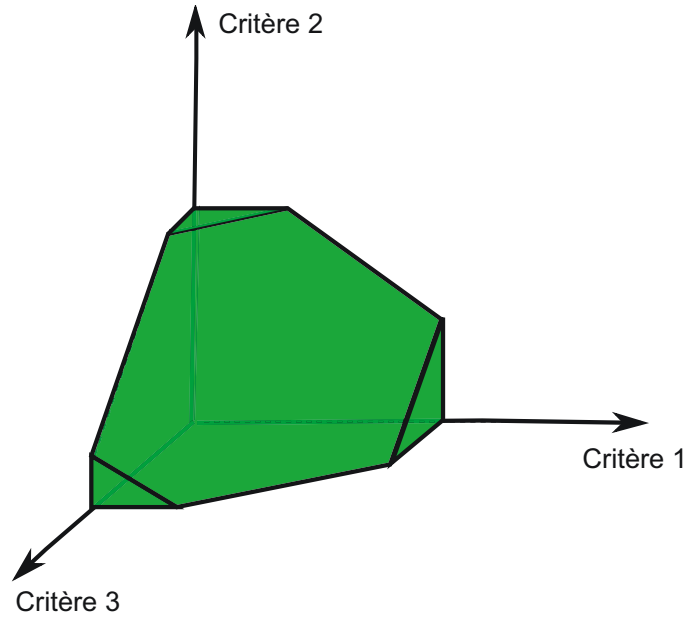


FIGURE 3.4 – Forme finale d’une boîte de qualité

Une boîte  $\mathbf{B}(k)$  est définie par un ensemble de points  $\mathbf{x} = (x_1, \dots, x_n)^T, x_i \geq 0, \forall i \in [1, n]$  tel que :

$$\mathbf{A}(k) \mathbf{x} \leq \mathbf{b}$$

tel que  $\mathbf{A}(k) = (a_{ij}(k))$  est une matrice  $(n+1) \times n$  et  $\mathbf{b}$  un vecteur colonne de taille  $n+1$ . “ $\leq$ ” dans le système d’inéquations ci-dessus signifie que chaque élément de la matrice colonne  $\mathbf{A}(k) \mathbf{x}$  est inférieur ou égal à son vis-à-vis dans  $\mathbf{b}$ . Dans ce qui suit, nous posons  $\mathbf{b} = (1, \dots, 1)$ . Pour les hyperplans orthogonaux aux axes  $\mathbf{H}_i(k), 1 \leq i \leq n : a_{ij}(k) = 0$  si  $i \neq j$  et  $a_{ii} > 0$ . Pour l’hyperplan oblique  $\mathbf{H}_{n+1}(k) : a_{n+1,j}(k) \geq 0$ .

### 3.3.1 Distance globale d’un vecteur de critères à la boîte 0

Nous définissons la distance  $\Delta(\mathbf{c})$  entre un vecteur de critères  $\mathbf{c} = (c_1, \dots, c_n)^T$  et la boîte  $\mathbf{B}(0)$  comme suit :

$$\Delta(\mathbf{c}) = \begin{cases} k + \max_{i \in [1, n+1]} \delta_i(\mathbf{c}) & \text{si } \mathbf{c} \in \mathbf{B}(k+1) \text{ et } \mathbf{c} \notin \mathbf{B}(k) \\ 0 & \text{si } \mathbf{c} \in \mathbf{B}(0) \end{cases} \quad (3.1)$$

où,  $\delta_i(\mathbf{c})$  est la distance normalisée entre le vecteur  $\mathbf{c}$  et l'hyperplan  $\mathbf{H}_i(k)$  représentant une frontière de la boîte  $\mathbf{B}(k)$ . Son expression est donnée ci-dessous.

Soit  $\mathbf{p}$  la projection orthogonale du vecteur  $\mathbf{c}$  sur l'hyperplan  $\mathbf{H}_i(k)$ . Soit  $\mathbf{e}$  le point d'intersection de l'hyperplan  $\mathbf{H}_h(k+1)$  composant la frontière supérieure de la boîte  $\mathbf{B}(k+1)$  avec la normale de l'hyperplan  $\mathbf{H}_i(k)$  de la boîte  $\mathbf{B}(k)$  passant par  $p$  (Figure 3.5). L'hyperplan  $\mathbf{H}_h(k+1)$  est choisi tel que  $\|\mathbf{p} - \mathbf{e}\|$  est minimale, où  $\|\mathbf{x}\|$  désigne le module du vecteur  $\mathbf{x}$ . D'abord, nous déterminons les coordonnées  $p_s$  et  $e_s$ ,  $s \in [1, n]$  en utilisant les expressions 3.2 et 3.3, respectivement, puis nous calculons la distance normalisée  $\delta_i(\mathbf{c})$  en utilisant l'expression 3.4.

$$p_s = c_s + a_{is}(k) \frac{1 - \sum_{j=1}^n a_{ij}(k)c_j}{\sum_{j=1}^n a_{i,j}^2(k)} \quad (3.2)$$

$$e_s = p_s + a_{is}(k) \frac{1 - \sum_{j=1}^n a_{hj}(k+1)p_j}{\sum_{j=1}^n a_{hj}(k+1)a_{ij}(k)} \quad (3.3)$$

$$\delta_i(\mathbf{c}) = \begin{cases} 0 & \text{si } \sum_{j=1}^n a_{ij}(k)c_j \leq 1 \\ 0 & \text{si } \|\mathbf{p} - \mathbf{e}\| = 0 \\ \frac{\|\mathbf{p} - \mathbf{c}\|}{\|\mathbf{p} - \mathbf{e}\|} & \text{sinon} \end{cases} \quad (3.4)$$

où  $\|\mathbf{x} - \mathbf{x}'\|$  est la distance euclidienne entre  $\mathbf{x}$  et  $\mathbf{x}'$ .

Ainsi, chaque point situé sur la frontière supérieure de la boîte  $\mathbf{B}(k+1)$  est à la distance 1 de la boîte englobée  $\mathbf{B}(k)$ , excepté dans le cas où  $\|\mathbf{p}, \mathbf{e}\| = 0$ . Dans ce dernier cas, la distance vaut 0.

### 3.3.2 Processus de décision multicritère

Soient  $\mathbf{u}$  et  $\mathbf{v}$  deux vecteurs de critères à comparer. Le processus de décision multicritère est résumé dans l'algorithme suivant :

En appliquant l'algorithme ci-dessus, il est simple de démontrer que la relation "est meilleur que ou indifférent" est réflexive et transitive. Cependant, cette relation n'est

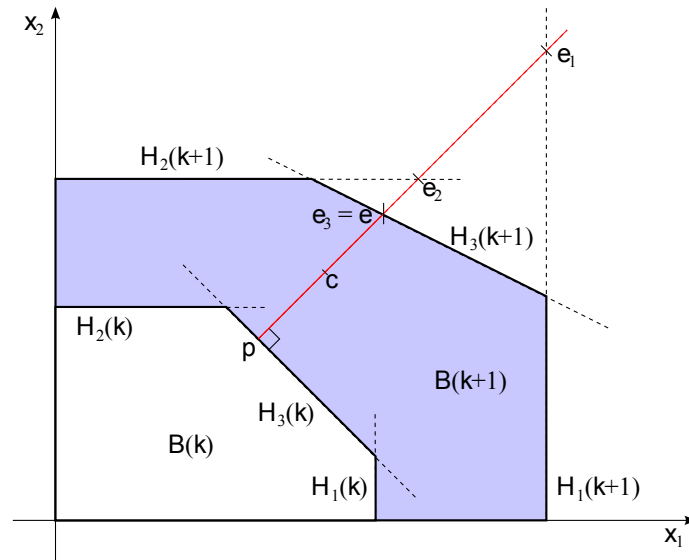


FIGURE 3.5 – Distance normalisée  $\delta_3(\mathbf{c})$  entre l'hyperplan  $\mathbf{H}_3(k)$  et  $\mathbf{c}$

---

**Algorithme 1:** règles de comparaison

---

**Données :** deux vecteurs de critères  $\mathbf{u}$  et  $\mathbf{v}$

**Résultat :** décision sur la préférence

**si**  $\Delta(\mathbf{u}) < \Delta(\mathbf{v})$  **alors**

    |  $\mathbf{u}$  est dit “meilleur que”  $\mathbf{v}$ ;

**sinon si**  $\Delta(\mathbf{u}) > \Delta(\mathbf{v})$  **alors**

    |  $\mathbf{v}$  est dit “meilleur que”  $\mathbf{u}$ ;

**sinon**

    |  $\mathbf{u}$  et  $\mathbf{v}$  sont dit indifférents”;

---



pas antisymétrique, à cause de la règle “indifférent”, et elle n’est pas symétrique, à cause de la règle “meilleur que”.

La méthode de la poupée russe est implémentée en comparant les vecteurs de critères en utilisant les règles de comparaison explicitées ci-dessus. Le meilleur vecteur de critères est celui pour lequel la distance  $\Delta$  est minimale.

### 3.3.3 Boîte de qualité hyper-rectangulaire

Une boîte de qualité donnée est définie par l’intersection d’un hyperrectangle avec un simplexe. Quand l’hyperrectangle est complètement à l’intérieur du simplexe, la boîte de qualité est réduite à un hyperrectangle. Dans ce cas,  $a_{ij} = 0$  si  $i \neq j$ . De ce fait, la boîte de qualité est définie par les inéquations suivantes :  $x_i \geq 0$ ,  $a_{ii}(k)x_i \leq 1$ , tel que :  $1 \leq i \leq n$ .

De plus, si nous considérons que toutes les boîtes qui composent la poupée russe sont des hyperrectangles, alors les équations (3.2) et (3.3) deviennent, respectivement :

$$p_s = \begin{cases} c_i & \text{si } i \neq s \\ 1/a_{ii}(k) & \text{sinon} \end{cases}$$

et

$$e_s = \begin{cases} c_i & \text{si } i \neq s \\ 1/a_{ii}(k+1) & \text{sinon} \end{cases}$$

Selon l’équation (3.4), pour tout vecteur de critères  $\mathbf{x}$  choisi dans l’ensemble défini par l’intersection des boîtes  $\mathbf{B}(k+1)$  et  $\mathbf{B}(k)$ , avec  $a_{ii}(k) \neq a_{ii}(k+1)$  :

$$\Delta(\mathbf{x}) = k + \max_{i \in [1, n]} \frac{x_i - p_i}{e_i - p_i}$$

Plus précisément, si  $k = 0$  et  $\mathbf{x}$  est choisi tel que  $x_i \geq 1/a_{ii}(k)$ , alors  $\Delta(\mathbf{x})$  est une distance de *Chebyshev* pondérée entre  $\mathbf{x}$  et  $\mathbf{p} = (p_1, \dots, p_i, \dots, p_n)$ .

Il est utile d’étudier le modèle d’agrégation utilisé pour résoudre un problème de décision multicritère du point de vue de la compensation ou la non-compensation entre critères. La compensation signifie qu’une diminution de la valeur de la fonction d’utilité en raison d’une détérioration d’un critère donné peut être compensée par une

### 3.3. Définition formelle de la méthode

Qualité	délat	gigue	taux de perte
Bonne	$[0, 150ms]$	$[0, 20ms]$	$[0, 0.5\%]$
Acceptable	$[150ms, 300ms]$	$[20ms, 50ms]$	$[0.5\%, 1.5\%]$
Mauvaise	$> 300ms$	$> 50ms$	$> 1.5\%$

TABLE 3.1 – Seuils de qualité des critères *délat*, *gigue* et *taux de perte* pour une application de vidéoconférence

amélioration des autres critères. La non-compensation entre critères signifie que la diminution de la valeur de la fonction d'utilité en raison d'un critère donné ne peut pas être compensée par une amélioration des autres critères.

La fonction d'utilité  $u(\mathbf{x}) = m - \Delta(\mathbf{x})$  décroît quand  $\Delta(\mathbf{x})$  croît. Cette situation survient lorsque  $x_i$  se détériore, alors que  $(x_i - p_i)/(e_i - p_i)$  est maximal pour le critère  $i$ . Dans ce cas, l'amélioration des autres critères, c'est-à-dire la réduction de leur valeur, ne permet pas d'augmenter  $u(\mathbf{x})$ . Ainsi, une poupée russe composée de boîtes hyperrectangulaires forme un modèle d'agrégation non compensatoire. Un concepteur d'un système de décision multicritère peut vouloir utiliser un modèle non compensatoire lorsque certains critères sont incommensurables.

Nous définissons le seuil de qualité (QT : *Quality Threshold*)  $t_i(k)$  comme la pire valeur du critère  $i$  pour laquelle la valeur de la qualité est  $k$ , sachant que les autres critères sont à leurs valeurs optimales. Typiquement, ce genre d'information devrait être facilement disponible pour les concepteurs du système. Ces paramètres pourraient provenir de normes, spécifications techniques des composants du système ou des expériences. Par exemple, pour une application de type vidéoconférence, ces paramètres peuvent être les données indiquées dans le tableau 3.1 [CSMS04]. Un hyperplan  $\mathbf{H}_i(k)$  est défini par les coefficients  $a_{ii}(k) = 1/t_i(k)$  et  $a_{ij} = 0$  pour  $i \neq j$  pour les autres coefficients. l'hyperplan oblique  $\mathbf{H}_{n+1}(k)$  est inutile : par conséquent, il n'est pas défini.

#### 3.3.4 Boîte de qualité de type simplexe

Quand le simplexe qui compose la boîte  $\mathbf{B}(k)$  est complètement à l'intérieur de l'hyperrectangle, la boîte de qualité est réduite à un simplexe défini par les inéquations

suivantes :

$$\begin{aligned} \sum_{j=1}^n a_{n+1,j}(k) x_j &\leq 1 \\ x_j &\geq 0 \end{aligned}$$

tel que :  $\forall j \in [1, n], a_{n+1,j}(k) > 0$ .

La distance globale à la boîte 0 devient :

$$\Delta(\mathbf{x}) = k + \delta_{n+1}(\mathbf{x})$$

Supposons que toutes les boîtes composant la poupée russe soient des simplexes. Nous voulons caractériser l'ensemble des points  $\mathbf{x}$  qui sont à une distance  $\delta_{n+1}(\mathbf{x}) = \gamma$  de l'hyperplan  $\mathbf{H}_{n+1}(k)$ , où  $\gamma$  est une constante choisie dans l'intervalle  $[0, 1]$ . Le système de coordonnées est transformé par une translation et une rotation afin que l'équation de l'hyperplan  $\sum_{j=1}^n a_{n+1,j}(k) x_j = 1$  dans le système de coordonnées initial devienne  $x'_1 = 0$  dans le nouveau système de coordonnées. L'équation de l'hyperplan  $\mathbf{H}_{n+1}(k+1)$  devient  $\sum_{j=1}^n \alpha_{n+1,j} x'_j = 1$ . Dans ce cas, tout point  $\mathbf{x}'$  appartenant à l'hyperplan dont l'équation est  $\alpha_{n+1,1} x'_1 / \gamma + \sum_{j=2}^n \alpha_{n+1,j} x'_j = 1$  est à une distance  $\delta_{n+1}(\mathbf{x}) = \gamma$  de l'hyperplan  $\mathbf{H}_{n+1}(k)$ . En effet :

Soit,

$$\mathbf{c} = (x'_1, \dots, x'_n) \in \left( \frac{\alpha_{n+1,1} x'_1}{\gamma} + \sum_{j=2}^n \alpha_{n+1,j} x'_j = 1 \right)$$

Donc,

$$\mathbf{c} = \left( \frac{\gamma \left( 1 - \sum_{j=2}^n \alpha_{n+1,j} x'_j \right)}{\alpha_{n+1,1}}, x'_2, \dots, x'_n \right)$$

$\mathbf{p} = (0, x'_2, \dots, x'_n)$  est la projection orthogonale du point  $\mathbf{c}$  sur l'hyperplan dont l'équation est  $x'_1 = 0$  et  $\mathbf{e} = \left( \frac{1 - \sum_{j=2}^n \alpha_{n+1,j} x'_j}{\alpha_{n+1,1}}, x'_2, \dots, x'_n \right)$ .

Donc,

$$\|\mathbf{p} - \mathbf{c}\| = \sqrt{\gamma^2 \left( \frac{\sum_{j=2}^n \alpha_{n+1,j} x'_j - 1}{\alpha_{n+1,1}} \right)^2}$$

Et

$$\|\mathbf{p} - \mathbf{e}\| = \sqrt{\left(\frac{\sum_{j=2}^n \alpha_{n+1,j} x'_j - 1}{\alpha_{n+1,1}}\right)^2}$$

Finalement,

$$\delta_{n+1}(\mathbf{c}) = \frac{\|\mathbf{p} - \mathbf{c}\|}{\|\mathbf{p} - \mathbf{e}\|} = \frac{\sqrt{\gamma^2 \left(\frac{\sum_{j=2}^n \alpha_{n+1,j} x'_j - 1}{\alpha_{n+1,1}}\right)^2}}{\sqrt{\left(\frac{\sum_{j=2}^n \alpha_{n+1,j} x'_j - 1}{\alpha_{n+1,1}}\right)^2}} = \gamma$$

Par conséquent, les hypersurfaces iso-valeur dans une poupée russe de type simplexe sont des hyperplans.

Soit la distance  $\delta_{n+1}(\mathbf{x})$  pour un vecteur de critères donné  $\mathbf{x}$ . Cette distance définit une hypersurface iso-valeur représentée par l'hyperplan  $\mathbf{I}(\mathbf{x})$ . La détérioration de la qualité d'un critère donné ne change pas la valeur de la distance, à condition que les autres critères soient choisis de telle sorte que le nouveau vecteur de critères appartienne aussi à l'hyperplan  $\mathbf{I}(\mathbf{x})$ . De ce fait, pour tout vecteur de critères  $\mathbf{x}$  suffisamment loin des bords du simplexe, il existe un voisinage dans lequel la dégradation d'un critère peut toujours être compensée par l'amélioration des autres critères. En conclusion, la méthode de la poupée russe instanciée avec des boîtes simplexes est un modèle d'agrégation compensatoire.

Par ailleurs, soit  $f(y)$  une fonction croissante de  $y \in \mathbf{R}$  avec  $f(0) = 1$ . Si les coefficients  $a_{n+1,j}(k) = a_{n+1,j}(0) / f(k)$ ,  $\forall j \in [1, n]$  pour toute boîte de qualité  $k$ , alors les hyperplans obliques qui forment les différentes boîtes simplexes sont parallèles. De ce fait, les hypersurfaces iso-valeur à la boîte 0 sont des hyperplans parallèles dont l'équation est  $\sum_{j=1}^n a_{n+1,j}(0) x_j = f(\Delta(\mathbf{x}))$ . Par conséquent, minimiser  $\Delta(\mathbf{x})$  revient à minimiser  $\sum_{j=1}^n a_{n+1,j}(0) x_j$ . Dans ce cas, la méthode de la poupée russe est une méthode de somme pondérée pour les vecteurs de critères n'appartenant pas à la boîte 0. De plus, si la poupée russe est composée d'un ensemble de simplexes dont les hyperplans obliques ne sont pas parallèles, alors la méthode de la poupée russe se comporte comme une méthode de la somme pondérée avec des poids variables selon la valeur des vecteurs de critères qui sont en dehors de la boîte 0.

Les paramètres des boîtes de qualité de type simplexe peuvent être directement

introduits par le concepteur à l'image des boîtes hyperrectangulaires. Dans ce contexte, l'hyperplan oblique  $\mathbf{H}_{n+1}(k)$  est défini au moyen des coefficients  $a_{n+1,i}(k) = 1/t_i(k)$ , tels que  $t_i(k)$  sont des seuils de qualité introduits dans la section précédente.

#### 3.3.5 Définition d'une boîte de qualité au moyen d'un processus d'apprentissage

Une boîte de qualité  $k$  est construite par l'intersection d'un hyperrectangle défini par  $n$  hyperplans  $\mathbf{H}_i(k)$ , avec  $1 \leq i \leq n$  et un simplexe défini par un hyperplan  $\mathbf{H}_{n+1}(k)$ . Ainsi, le modèle d'agrégation peut être compensatoire ou non compensatoire dans un voisinage d'un vecteur de critères  $\mathbf{x}$ . Les relations entre les critères peuvent ensuite être adaptées en fonction du problème de décision multicritère, par un choix adéquat des paramètres des hyperplans. Cependant, il est difficile pour un concepteur d'obtenir ces paramètres à partir des données disponibles, telles que celles fournies par un MOS pour un nombre de critères supérieur à deux. Pour résoudre ce problème, un algorithme de classification automatique, capable de faire face à l'incertitude des données MOS, peut être utilisé. Dans ce contexte, nous utilisons la méthode décrite brièvement ci-dessous.

Globalement, l'algorithme d'apprentissage utilise la méthode de la descente du gradient afin de minimiser la distance entre les hyperplans et des ensembles de points mal classés dans l'espace des critères. Avant d'introduire formellement l'algorithme, les ensembles de points mal classés en dehors de la boîte  $\mathbf{B}(k)$  et l'ensemble des points mal classés à l'intérieur de la boîte  $\mathbf{B}(k)$ , associés à un hyperplan donné  $\mathbf{H}_i(k)$ , doivent être définis. Soit  $l$  une étiquette qui désigne la qualité affectée au point  $\mathbf{x}$  dans l'espace des critères.  $l = 5 - l_{mos}$ , tel que  $l_{mos}$  est la qualité affectée au point  $\mathbf{x}$  dans l'expérience MOS. Nous rappelons que, lors d'une expérience de type MOS, les utilisateurs attribuent des notes allant de 5 (excellente qualité) à 1 (mauvaise qualité) aux échantillons de la VoIP, en fonction de la qualité perçue.

- **Ensemble des points mal classés en dehors d'une boîte** :  $\mathbf{x}$  est dit "mal classé" si  $\mathbf{x}$  est en dehors de la boîte  $\mathbf{B}(k)$  et  $l \leq k$ .

L'ensemble des points mal classés situés en dehors de la boîte  $\mathbf{B}(k)$  par rapport

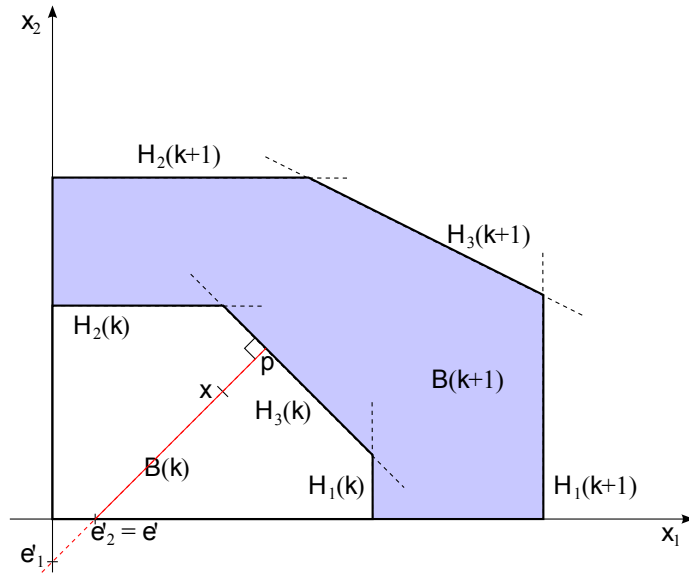


FIGURE 3.6 –  $\frac{\|\mathbf{p}-\mathbf{x}\|}{\|\mathbf{p}-\mathbf{e}'\|}$  est la distance normalisée entre le vecteur de critères  $\mathbf{x}$  et la boîte  $\mathbf{B}(k)$

à l'hyperplan  $\mathbf{H}_i(k)$  contient les points mal classés  $\mathbf{x}$  dont la distance  $\delta_i(\mathbf{x})$  à l'hyperplan  $\mathbf{H}_i(k)$  est maximale parmi tous les hyperplans composant la boîte  $\mathbf{B}(k)$ . L'hyperplan  $\mathbf{H}_i(k)$  est appelé “facette active” du point  $\mathbf{x}$ .

- **Ensemble des points mal classés à l'intérieur d'une boîte** :  $\mathbf{x}$  est dit “mal classé” si  $\mathbf{x}$  est à l'intérieur de la boîte  $\mathbf{B}(k)$  et  $l > k$ .

L'ensemble des points mal classés à l'intérieur d'une boîte  $\mathbf{B}(k)$  par rapport à l'hyperplan  $\mathbf{H}_i(k)$  contient les points mal classés  $\mathbf{x}$  dont la distance  $\delta'_i(\mathbf{x})$  à l'hyperplan  $\mathbf{H}_i(k)$  est minimale parmi tous les hyperplans composant la boîte  $\mathbf{B}(k)$ .  $\delta'_i(\mathbf{x})$  est définie d'une façon similaire à la distance  $\delta_i(\mathbf{x})$  (équation 3.4).

$$\delta'_i(\mathbf{x}) = \frac{\|\mathbf{p} - \mathbf{x}\|}{\|\mathbf{p} - \mathbf{e}'\|} \quad (3.5)$$

tel que  $\mathbf{p}$  est la projection de  $\mathbf{x}$  sur  $\mathbf{H}_i(k)$  et  $\mathbf{e}'_j$  est l'intersection de l'hyperplan  $x_j = 0$  avec la normale de  $\mathbf{H}_i(k)$  passant par  $\mathbf{p}$ .  $\mathbf{e}' = \mathbf{e}'_j$  en choisissant  $j$  tel que  $\|\mathbf{p} - \mathbf{e}'\|$  est minimale (voir la figure 3.6). L'hyperplan  $\mathbf{H}_i(k)$  est appelé “facette active” de  $\mathbf{x}$ .

Soit  $\mathbf{M}_i(k)$  l'ensemble des vecteurs de critères mal classés associés à l'hyperplan  $\mathbf{H}_i(k)$ . Cet ensemble est l'union des deux sous-ensembles des vecteurs de critères mal

classés en dehors et à l'intérieur de la boîte  $\mathbf{B}(k)$ . Le processus d'apprentissage minimise l'erreur quadratique  $e(\mathbf{a}_i)$  entre  $\mathbf{H}_i(k)$  et les éléments de l'ensemble  $\mathbf{M}_i(k)$ , en ajustant les coefficients  $a_{ij}(k)$  au moyen de la méthode de la descente du gradient.

$$e(\mathbf{a}_i) = \frac{1}{s\mathbf{a}_i^2} \sum_{m=1}^s (\mathbf{a}_i \mathbf{x}_m - 1)^2 \quad (3.6)$$

tel que  $\mathbf{a}_i$  est le vecteur contenant les coefficients de  $\mathbf{H}_i(k)$ ,  $\mathbf{x}_m \in \mathbf{M}_i(k)$  et  $s$  est le nombre d'éléments dans  $\mathbf{M}_i(k)$ . L'équation du gradient est donnée comme suit :

$$\nabla e(\mathbf{a}_i) = \frac{2}{s(\mathbf{a}_i^2)^2} \sum_{m=1}^s (\mathbf{a}_i \mathbf{x}_m - 1) (\mathbf{a}_i^2 \mathbf{x}_m - (\mathbf{a}_i \mathbf{x}_m - 1)\mathbf{a}_i) \quad (3.7)$$

Particulièrement, pour un hyperplan  $\mathbf{H}_i(k)$  de l'hyper-rectangle composant la boîte, cette équation devient :

$$\frac{\partial e}{\partial a_{ii}} = \frac{2}{a_{ii}^2} \left( \frac{\sum_{m=1}^s x_{mi}}{s} - \frac{1}{a_{ii}} \right) \quad (3.8)$$

Le gradient est nul pour :

$$a_{ii} = \frac{s}{\sum_{m=1}^s x_{mi}} \quad (3.9)$$

L'algorithme d'apprentissage est composé d'une boucle principale, qui optimise d'une façon itérative les paramètres d'une boîte. Dans ce contexte, les coefficients des différents hyperplans composant l'hyperrectangle sont mis à jour au moyen de la fonction "HyperRectangleUpdate", tandis que ceux de l'hyperplan oblique sont mis à jour au moyen de la fonction "ObliqueUpdate". La fonction "isMisclassified" retourne "vrai" si un vecteur de critère  $\mathbf{x}$  associé à une qualité donnée  $l$  est mal classé par rapport à la boîte de qualité  $k$ . Par ailleurs, la fonction "GetQuadraticCost" est utilisée pour calculer la somme des erreurs quadratiques données par l'équation (3.6) pour tout hyperplan d'une boîte de qualité donnée. Enfin, le processus d'apprentissage se termine quand la variation de l'erreur quadratique est inférieure à un seuil déterminé  $\varepsilon$ . L'algorithme utilise les notations suivantes :

- $\mathbf{T}$  : est l'ensemble d'apprentissage composé par des couples  $(\mathbf{x}, l)$  où  $l$  est un indicateur de qualité subjective associée au vecteur de critères  $\mathbf{x}$ .  $\mathbf{T}$  peut être le

résultat d'une expérience de type MOS.

- $\alpha$  : est le pas utilisé dans la méthode de la descente du gradient.  $\alpha$  doit être positif et suffisamment petit pour garantir la convergence de l'algorithme.

L'algorithme d'apprentissage présenté ci-dessous doit être appliqué à chaque boîte de la poupée russe, en commençant par la plus grande boîte jusqu'à la plus petite (boîte 0). Ainsi, la distance  $\delta_i$  est toujours correctement définie.

---

**Algorithme 2:** Algorithme d'apprentissage

---

**Données :**

- $\mathbf{T}$  : l'ensemble d'apprentissage,
- $k$  : index d'une boîte,
- $n$  : nombre de critères dans le système,
- $\alpha$  : pas de la méthode du gradient,
- $\varepsilon$  : critère d'arrêt de l'algorithme.

**Résultat :**

$\mathbf{a}$  : matrice  $(a_{ij})$  où  $a_{ij}$  est le  $j$ -ième coefficient de  $\mathbf{H}_i(k)$

$\mathbf{a} \leftarrow \text{RandomInit}(n)$

$e \leftarrow 0$

**répéter**

HyperRectangleUpdate( $\mathbf{a}, n, k, \mathbf{T}$ )

ObliqueUpdate( $\mathbf{a}, n, k, \mathbf{T}, \alpha$ )

$p \leftarrow e$

$e \leftarrow \text{GetQuadraticCost}(\mathbf{a}, n, k, \mathbf{T})$

**jusqu'à**  $|e - p| < \varepsilon$ ;

---



---

**Procédure** HyperRectangleUpdate( $\mathbf{a}, n, k, \mathbf{T}$ )

---

**pour chaque**  $i \in [1, n]$  **faire**

$s_i \leftarrow 0$

$c_i \leftarrow 0$

**pour chaque**  $(\mathbf{x}, l) \in \mathbf{T}$  **faire**

**si**  $\text{isMisclassified}((\mathbf{x}, l), \mathbf{a}, k)$  **alors**

$i \leftarrow \text{GetActiveHyperplaneIndex}(\mathbf{x}, \mathbf{a})$

**si**  $i \leq n$  **alors**

$s_i \leftarrow s_i + 1$

$c_i \leftarrow c_i + x_i$  //  $x_i$  est la  $i$ -ème coordonnée de  $\mathbf{x}$

**pour chaque**  $i \in [1, n]$  **faire**

**si**  $s_i > 0$  **alors**

$a_{ii} \leftarrow s_i/c_i$  // voir l'équation (3.9)

---



---

**Procédure** ObliqueUpdate( $\mathbf{a}$ ,  $n$ ,  $k$ ,  $\mathbf{T}$ ,  $\alpha$ )

---

```

 $s \leftarrow 0$ 
 $\mathbf{g} \leftarrow 0$ 
pour chaque  $(\mathbf{x}, l) \in \mathbf{T}$  faire
    si  $isMisclassified((\mathbf{x}, l), \mathbf{a}, k)$  alors
         $i \leftarrow \text{GetActiveHyperplaneIndex}(\mathbf{x}, \mathbf{a})$ 
        si  $i = n + 1$  alors
             $s \leftarrow s + 1$ 
             $\mathbf{g} \leftarrow \mathbf{g} + (\mathbf{a}_{n+1}\mathbf{x} - 1) (\mathbf{a}_{n+1}^2\mathbf{x} - (\mathbf{a}_{n+1}\mathbf{x} - 1)\mathbf{a}_{n+1})$ 
            // voir l'équation (3.7)
 $\mathbf{a}_{n+1} \leftarrow \mathbf{a}_{n+1} - \alpha\mathbf{g}/s$ 

```

---

### 3.4 Conclusion

La plupart des méthodes de décision multicritère classiques utilisent des paramètres qui sont déduits des préférences du décideur. Ces paramètres sont souvent des poids qui quantifient l'importance de chaque critère dans le processus de décision multicritère. Dans ce chapitre, nous nous sommes intéressés au cas de la prise de décision multicritère dans les systèmes autonomes et dynamiques dont le front de Pareto varie en fonction du temps. Bien évidemment, il est impossible de prévoir une interaction avec le décideur dans ce type de systèmes.

Dans ce contexte, nous avons présenté une méthode de décision multicritère qui s'inspire du principe d'empilement d'une poupée russe pour construire une fonction d'utilité dans l'espace des critères. Cette fonction d'utilité utilise des paramètres déterminés lors de la phase de conception. En effet, les paramètres de la méthode sont soit directement déduits des normes et spécifications techniques des systèmes, soit éventuellement obtenus par l'intermédiaire d'une méthode de classification automatique exécutée sur un MOS. Plus précisément, la méthode de la poupée russe permet trois types d'adaptation distincts :

- Elle peut adapter l'importance des critères selon leurs valeurs, grâce aux multiples boîtes de qualité englobantes,
- Elle offre la possibilité de prendre en compte les différentes relations qui existent entre critères, en l'occurrence la compensation (boîtes de type simplexe), la non-compensation (boîtes de type hyperrectangulaire) et une combinaison de ces

dernières (intersection d'un simplexe et d'un hyperrectangle),

- Elle offre la possibilité d'apprendre ses paramètres à partir de données dont l'acquisition peut être automatisée, ce qui lui permet d'adapter sa forme si cela est utile à la maximisation des performances du système.

Dans le chapitre suivant, nous allons évaluer les performances de la méthode dans un cas de prise de décision multicritère, pour un système autonome et dynamique, en l'occurrence le routage multicritère dans les réseaux ad hoc sans fil.

# Chapitre 4

## Application au routage multicritère dans les réseaux ad hoc sans fil

### 4.1 Introduction

Les réseaux ad hoc sans fil sont des réseaux auto-adaptatifs formés par un ensemble de terminaux sans fil qui établissent une topologie afin de communiquer. L'objectif de la tâche du routage dans ces réseaux est de trouver un chemin efficace (au sens large) entre un nœud source donné et un nœud destinataire. Comme la portée des nœuds est relativement petite, les chemins utilisés pour la communication sont souvent composés de plusieurs nœuds intermédiaires. De plus, la tâche du routage doit optimiser un certain nombre de critères relatifs à la qualité de service QoS (*Quality of Service*), en l'occurrence le délai, le taux de perte, la gigue, etc. Dans ce contexte, la majorité des protocoles de routage introduits dans la littérature prennent uniquement le nombre de sauts comme métrique à optimiser et négligent, par conséquent, les exigences des applications en termes de différents critères de QoS. Cependant, le problème de routage dans les réseaux ad hoc sans fil est typiquement une problématique de décision multicritère, dans ce sens que plusieurs critères, souvent contradictoires, sont impliqués dans le processus de décision. Dans ce chapitre, nous comparerons la méthode de la poupée russe à la méthode de la somme pondérée, dans le cas du routage dans les réseaux sans fil. Ce problème consiste à choisir le meilleur chemin en fonction des exigences