

MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR
ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITÉ MOULOUD MAMMERI, TIZI-OUZOU



FACULTÉ DES SCIENCES, DÉPARTEMENT DE MATHÉMATIQUES

MEMOIRE DE FIN D'ETUDE

Spécialité : Mathématiques Appliquées

Option : Mathématiques Appliquées aux Sciences de Gestion

Présenté par :

ARAB Samy

Thème

La modélisation du problème d'optimisation multi-objectif des conditions de coupe pour une opération de tournage

Encadré par :
Mme **LESLOUS Fadila**

Devant le jury d'examen composé de :

Mme OUBAKOUK Lynda	MAA	Présidente
Mme LESLOUS Fadila	MCB	Rapporteur
Mme SLIMI Farida	MCB	Examinatrice

Soutenue : 3 Juillet 2025

Remerciements

Je tiens tout d'abord à exprimer ma profonde gratitude à mon encadrante, **Mme LESLOUS Fadila**, pour son accompagnement, ses conseils pertinents et sa disponibilité tout au long de ce travail.

Je remercie également les membres du jury, **Mme OUBAKOUK Lynda** et **Mme SLIMI Farida**, pour l'honneur qu'elles me font en acceptant d'évaluer ce mémoire.

Je souhaite aussi remercier l'ensemble des enseignants du département mathématique de l'université de MOULOUD MAMMERI pour la qualité de l'enseignement dispensé tout au long de mon parcours.

Et enfin DIEU MERCI pour la patience, le courage, la santé qui m'a donné pour finir ce travail.

Dédicaces

Je dédie ce modeste mémoire à toute ma famille.

À mes parents.

À mes frères et sœurs.

À mes amis et collègues.

À tous ceux qui m'aiment et me respectent.

Table des matières

Table des matières	4
Introduction générale	6
1 Optimisation mono-objectif	9
1.1 Introduction	9
1.2 Rappels et généralités	9
1.2.1 Calcul différentiel	9
1.2.2 Les matrices	10
1.2.3 Convexité	12
1.3 Optimisation sans contraintes	13
1.3.1 Condition nécessaire du premier ordre	13
1.3.2 Condition nécessaire du deuxième ordre	13
1.3.3 Condition suffisantes	14
1.3.4 Exemples	14
1.4 Optimisation avec contraintes	16
1.4.1 Existance et unicité	16
1.4.2 Conditions d'optimalité du premier ordre	16
1.4.3 Conditions d'optimalité du deuxième ordre	19
1.4.4 Exemples	20
2 Optimisation multi-objectif	24
2.1 Introduction	24
2.2 Les structures de préférences	24
2.2.1 Les relations binaires et les ordres	24
2.2.2 Les structures de préférences	25
2.2.3 La structure de préordre complet	25
2.2.4 La structure d'ordre partiel et de préordre partiel	26
2.3 Définition d'un problème multi-objectif (PMO)	26
2.3.1 La multiplicités des solutions	28
2.3.2 La difficulté dans un problème multi-objectif	28

2.3.3	Méthodes de résolution d'un problème multi-objectif	28
2.4	La dominance	29
2.4.1	Propriétés de la relation de dominance	30
2.5	Les points particuliers idéal et nadir	31
2.5.1	Point idéal	31
2.5.2	Point nadir	31
2.5.3	Matrice de gain	31
2.5.4	Exemple d'illustration	32
2.6	Optimalité de Pareto et front de Pareto	35
2.6.1	Efficacité	35
2.6.2	Efficacité faible et efficacité fort	36
2.6.3	Front de Pareto optimal	36
2.7	Exemple	36
3	Méthodes d'optimisation	41
3.1	Introduction	41
3.2	Méthodes d'optimisation mono-objectif sans contrainte	41
3.2.1	Méthode du gradient conjugué linéaire	41
3.2.2	Méthode de Newton	45
3.3	Méthodes d'optimisation mono-objectif avec contrainte	48
3.3.1	Méthode du gradient projetée	48
3.3.2	Méthode du Newton projetée	51
3.4	Méthodes d'optimisation multi-objectif	54
3.4.1	La méthode de pondération des fonctions objectif	54
3.4.2	Méthode de ELECTRE I	58
4	Application en industrie	62
4.1	Introduction	62
4.2	Définition d'une coupe	62
4.3	Les paramètres d'une coupe	62
4.3.1	Mouvement de coupe	63
4.3.2	Mouvement d'avance	63
4.3.3	Le temps de production	63
4.3.4	La consommation en outil	64
4.3.5	Etat de surface	64
4.4	Cas d'application	65
4.4.1	Implémentation	65
4.4.2	Modélisation du problème	69
4.4.3	Optimisation tri-objectif	76

Table des matières	6
<hr/>	
4.5 Conclusion	79
Conclusion générale et perspectives	79
Bibliographie	80

Introduction générale

L'optimisation mathématique constitue un outil fondamental dans l'analyse et la résolution de nombreux problèmes rencontrés aussi bien dans la vie quotidienne que dans les domaines scientifique, industriel et économique [6]. Dans notre routine, nous cherchons instinctivement à optimiser notre emploi du temps, à mieux organiser nos ressources ou à minimiser les distances à parcourir. Dans le monde professionnel, l'optimisation est au cœur de nombreuses décisions stratégiques, telles que la planification de la production, la gestion des stocks, l'allocation des ressources, la conception de produits.

Traditionnellement, l'optimisation se focalise sur un seul objectif à maximiser ou à minimiser. Toutefois, les problèmes réels sont rarement aussi simples. Ils impliquent souvent plusieurs objectifs simultanément, qui sont souvent conflictuels. Par exemple, en ingénierie ou en gestion de production, on cherche généralement à réduire les coûts, tout en améliorant la qualité ou diminuant le temps de réalisation [10]. Dans de tels cas, il n'existe généralement pas une solution unique qui optimise tous les critères à la fois. C'est dans ce contexte que s'inscrit l'optimisation multi-objectifs, une approche qui vise à identifier un ensemble de solutions dites efficaces ou non dominées, parmi lesquelles le décideur pourra choisir en fonction de ses préférences [23].

L'optimisation multi-objectifs est un domaine central de la recherche opérationnelle, car la majorité des problèmes réels impliquent plusieurs objectifs souvent conflictuels. Les premières réflexions sur ce type de problèmes remontent au *XIX^e* siècle avec les travaux de Francis Y Edgeworth [13], puis ont été formalisées par Vilfredo Pareto, à l'origine du concept de dominance de Pareto [26]. Cette approche permet de mieux représenter la complexité des décisions réelles et d'aider les décideurs à identifier des solutions de compromis. Parmi les méthodes de résolution, on distingue notamment les méthodes scalaires, qui transforment le problème multi-objectifs en un problème mono-objectif, comme la pondération ou la méthode des ϵ -contraintes

[23]. Bien que largement utilisées, ces méthodes présentent des limites, notamment le choix difficile des paramètres et leur inefficacité dans les cas non convexes [6].

L'objectif de ce mémoire est d'étudier les concepts fondamentaux de l'optimisation mono et multi-objectif, et vise à modéliser et résoudre un problème d'optimisation des conditions de coupe en tournage, en prenant en compte plusieurs critères techniques et économiques.

Ce mémoire est structuré en quatre chapitres :

- Le chapitre 1 est consacré pour **l'optimisation mono-objectif**. Il présente les fondements théorique, les conditions d'optimalité dans un contexte à critère unique.
- Le chapitre 2 introduit **l'optimisation multi-objectifs**. Il en détaille les concepts clés tels que la dominance de Pareto, les frontières de Pareto, les préférences du décideur.
- Le chapitre 3 est dédié aux **méthodes de résolution** des problèmes mono et multi-objectif. Dans ce chapitre on a présenté quelques algorithmes et méthodes de résolution comme Newton et gradient conjugué pour les problèmes mono-objectif et la méthode de pondération et ELECTRE 1 pour les problèmes multi-objectif.
- Le chapitre 4 sur **l'application en industrie** des conditions et paramètres d'une coupe lors d'une opération de tournage. Dans ce chapitre on va illustrer l'utilité de l'optimisation multi-objectifs dans un contexte industriel réel.

Chapitre 1

Optimisation mono-objectif

1.1 Introduction

L'optimisation mono-objectif consiste à chercher le minimum ou le maximum à un problème en optimisant un seul critère (une seule fonction objectif).

Ce type d'optimisation peut se faire avec ou sans contraintes. Pour l'aborder, on utilise diverses méthodes, allant des approches mathématiques traditionnelles aux algorithmes spécifique pour atteindre l'optimum. Ce type d'optimisation trouve des applications dans de nombreux domaines, tels que l'ingénierie, la finance...

1.2 Rappels et généralités

1.2.1 Calcul différentiel

Considérons \mathbb{R}^n comme l'espace vectoriel réel de dimension n , avec $n \in \mathbb{N}$. Les vecteurs sont systématiquement considérés comme des vecteurs colonnes, et la transposée d'un vecteur x est notée x^T .

Définition 1.1. (Le Gradient) [16] Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ une fonction différentiable. Le gradient de f , noté ∇f , est une application $\nabla f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ définie par :

$$\nabla f(x) = \left(\frac{\partial f}{\partial x_1}(x), \dots, \frac{\partial f}{\partial x_n}(x) \right)$$

Définition 1.2. (La Hessienne) [16]

Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ une fonction deux fois différentiable.

On appelle hessienne de f la fonction $\nabla^2 f : \mathbb{R}^n \rightarrow \mathbb{R}^{n \times n}$, définie comme suit :

$$\nabla^2 f(x) = \left(\frac{\partial^2 f}{\partial x_i \partial x_j}(x) \right), \quad i = 1, \dots, n, \quad j = 1, \dots, n.$$

Définition 1.3. (Point critique) [15] Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ une fonction différentiable. Tout point $x \in \mathbb{R}^n$ qui vérifie $\nabla f(x) = 0$ est appelé un **point critique** de f .

1.2.2 Les matrices

Pour vérifier la nature d'une matrice carrée donnée, il existe deux techniques classiques à savoir l'utilisation des valeurs propres et la technique des mineurs principaux du critère de Sylvestre.

Définition 1.4. [12]

Soit M une matrice carrée et symétrique à éléments réels d'ordre n .

- M est dite **définie positive** si et seulement si :

$$x^T M x > 0, \quad \forall x \in \mathbb{R}^n, \quad \|x\| \neq 0.$$

- M est dite **définie négative** si et seulement si son opposée $-M$ est définie positive, c'est-à-dire :

$$x^T M x < 0, \quad \forall x \in \mathbb{R}^n, \quad \|x\| \neq 0.$$

- M est dite **semi-définie positive** si :

$$x^T M x \geq 0, \quad \forall x \in \mathbb{R}^n.$$

- M est dite **semi-définie négative** si :

$$x^T M x \leq 0, \quad \forall x \in \mathbb{R}^n.$$

— M est dite **indéfinie** (ou non définie) si elle n'est ni semi-définie positive ni semi-définie négative, c'est-à-dire s'il existe $x_1, x_2 \in \mathbb{R}^n$ tels que :

$$x_1^T M x_1 > 0 \quad \text{et} \quad x_2^T M x_2 < 0.$$

Valeurs propres

Définition 1.5. [12] Soit $M \in \mathbb{R}^{n \times n}$ une matrice carrée et symétrique, et soient $\lambda \in \mathbb{R}$ et $v \in \mathbb{R}^n \setminus \{0\}$ tels que l'équation suivante soit vérifiée :

$$Mv = \lambda v. \tag{1.1}$$

Si cette équation est satisfaite, alors λ est appelé valeur propre de la matrice M et v est son vecteur propre associé.

Pour déterminer les valeurs propres de M , nous résolvons son polynôme caractéristique, défini par l'équation suivante :

$$\det(M - \lambda I) = 0. \quad (1.2)$$

Les valeurs propres de M sont les racines de ce polynôme caractéristique.

Théorème. [12] Soit $M \in \mathbb{R}^{n \times n}$ une matrice carrée et symétrique, et soit A l'ensemble des valeurs propres de M , qui sont réelles.

- M est **définie positive** si et seulement si toutes ses valeurs propres $\lambda \in \mathbb{R}$ sont strictement positives : $\lambda > 0$.
- M est **semi-définie positive** si et seulement si toutes ses valeurs propres sont positives ou nulles : $\lambda \geq 0$.
- M est **définie négative** si et seulement si toutes ses valeurs propres sont strictement négatives : $\lambda < 0$.
- M est **semi-définie négative** si et seulement si toutes ses valeurs propres sont négatives ou nulles : $\lambda \leq 0$.
- M est **indéfinie** si et seulement si elle possède à la fois des valeurs propres positives et négatives.

Critère de Sylvestre

Définition 1.6. (Mineurs principaux) [12]

Soit $M \in \mathbb{R}^{n \times n}$ une matrice carrée et symétrique. On note A_k la sous-matrice de taille $k \times k$ extraite du coin supérieur gauche de M , avec :

$$A_k = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1k} \\ a_{21} & a_{22} & \dots & a_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ a_{k1} & a_{k2} & \dots & a_{kk} \end{bmatrix}$$

En particulier, pour $k = 1$, on a $A_1 = [a_{11}]$, et pour $k = n$, on retrouve la matrice entière $A_n = M$.

Le déterminant de A_k est appelé un mineur principal d'ordre k de la matrice M et est noté :

$$\Delta_k = \det(A_k).$$

Théorème. [12]

Soit $M \in \mathbb{R}^{n \times n}$ une matrice symétrique.

1. M est semi-définie positive si et seulement si $\Delta_k \geq 0$, pour tous les mineurs principaux.
2. M est définie positive si et seulement si $\Delta_k > 0$, pour tous les mineurs principaux.
3. M est définie négative si et seulement si Δ_k a le même signe que $(-1)^k$, pour tout $k = 1, 2, \dots, n$.

1.2.3 Convexité

La convexité occupe une place essentielle en optimisation. Voici quelques définitions.

Définition 1.7. [5]

Un ensemble C est dit convexe si, pour tous points x et y de C , le segment $[x, y]$ est inclus dans C , c'est-à-dire que :

$$\forall t \in [0, 1], \quad tx + (1 - t)y \in C.$$

Une fonction f définie sur un ensemble convexe C est dite convexe si :

$$\forall (x, y) \in C \times C, \quad \forall t \in [0, 1], \quad f(tx + (1 - t)y) \leq tf(x) + (1 - t)f(y).$$

La fonction est dite strictement convexe si :

$$\forall (x, y) \in C \times C, \quad x \neq y, \quad \forall t \in]0, 1[, \quad f(tx + (1 - t)y) < tf(x) + (1 - t)f(y).$$

Proposition 1.1. Soit f une fonction différentiable sur un convexe C de \mathbb{R}^n ; f est convexe si et seulement si :

$$\forall (x, y) \in C \times C, \quad \nabla f(x) \cdot (y - x) \leq f(y) - f(x).$$

Remarque 1.1. Pour une fonction strictement convexe, il suffit de remplacer l'égalité large par une égalité strict.

Proposition 1.2. Soit f une fonction convexe définie sur un convexe C .

Tout minimum local f sur C est aussi un minimum global.

Si f est strictement convexe, alors il admet au plus un unique minimum global.

Combinaison convexe [22]

Soient x_1, \dots, x_k un nombre fini de points de \mathbb{R}^n et $\lambda_1, \dots, \lambda_k$ des réels tels que

$$\lambda_j \geq 0, \quad \forall j = 1, \dots, k, \quad \text{et} \quad \sum_{j=1}^k \lambda_j = 1.$$

On dit que

$$x = \sum_{j=1}^k \lambda_j x_j$$

est une combinaison convexe des points x_1, \dots, x_k .

Plus généralement, si $S \subset \mathbb{R}^n$ est un sous-ensemble quelconque, on dit qu'un point $x \in \mathbb{R}^n$ est une combinaison convexe de points de S s'il existe un nombre fini de points de S dont x soit une combinaison convexe.

1.3 Optimisation sans contraintes

L'optimisation sans contrainte consiste à rechercher le minimum (maximum) d'une fonction f de n variables à valeurs réelles sans imposer de contraintes spécifiques sur ces variables.

Un problème d'optimisation sans contrainte est défini par :

$$\min_{x \in \mathbb{R}^n} f(x) \tag{1.3}$$

Où f est appelé une fonction objectif.

Dans ce cas, on cherche un point x^* tel que $f(x^*) \leq f(x)$, où x^* est un minimum local.

1.3.1 Condition nécessaire du premier ordre

Théorème. [16]

Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ une fonction différentiable. Si $x^* \in \mathbb{R}^n$ réalise un minimum ou un maximum local de f , alors :

$$\nabla f(x^*) = 0.$$

Le point x^* est un point critique .

1.3.2 Condition nécessaire du deuxième ordre

Théorème. [15] Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ une fonction de classe C^2 . Si x^* est un minimum local, alors :

- $\nabla f(x^*) = 0$,

- La matrice hessienne $H_f(x^*)$ est semi-définie positive.

Remarque 1.2. Lorsque x^* est un maximum local; $H_{f(x^*)}$ sera semi définie négatif.

1.3.3 Condition suffisantes

Théorème. [12] f une fonction différentiable et de classe C^2 .

Si $\nabla f(x^*) = 0$ et $\nabla^2 f(x^*)$ est définie positif alors x^* est un minimum local.

Remarque 1.3. Si $\nabla^2 f(x^*)$ est définie négatif alors x^* est un maximum local.

1.3.4 Exemples

Exemple 1 [22]

Soit le problème suivant :

$$f(x_1, x_2) = x_1 x_2 + (3 - x_1 - x_2)(x_1 + x_2)$$

Condition nécessaire :

$$\nabla f(x_1, x_2) = \begin{cases} -2x_1 - x_2 + 3 = 0 \\ -2x_2 - x_1 + 3 = 0 \end{cases}$$

La solution de ce système nous donne :

$$\begin{cases} x_1 = 1 \\ x_2 = 1 \end{cases}$$

Donc $x^*=(1,1)$ est un point critique.

Condition suffisante :

$$H_{f(x^*)} = \begin{pmatrix} -2 & -1 \\ -1 & -2 \end{pmatrix} = - \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$$

Il est clair que

$$\begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$$

est définie positive, donc

$$- \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$$

est définie négative.

Conclusion : $x^* = (1, 1)$ est un maximum local.

Exemple 2 [11]

Un fabricant de téléviseurs produit q postes par semaine. Le **coût total** de production est donné par :

$$C(q) = 6q^2 + 80q + 5000.$$

Le **prix de vente unitaire** est donné par :

$$P(q) = 1080 - 4q.$$

Le revenu est donné par :

$$R(q) = P(q) \cdot q = (1080 - 4q)q = 1080q - 4q^2.$$

Le bénéfice s'écrit :

$$B(q) = R(q) - C(q) = 1080q - 4q^2 - (6q^2 + 80q + 5000) = -10q^2 + 1000q - 5000.$$

Condition nécessaire du premier ordre :

$$\nabla B(q) = \frac{dB}{dq} = -20q + 1000.$$

$$\nabla B(q) = 0 \quad \Rightarrow \quad -20q + 1000 = 0 \quad \Rightarrow \quad q = 50.$$

Condition nécessaire du second ordre :

$$\nabla^2 B(q) = \frac{d^2B}{dq^2} = -20 < 0.$$

Donc, $q = 50$ est un maximum local.

Bénéfice maximal :

$$B(50) = -10(50)^2 + 1000(50) - 5000 = -25000 + 50000 - 5000 = 20000.$$

Prix de vente unitaire :

$$P(50) = 1080 - 4 \cdot 50 = 880.$$

1.4 Optimisation avec contraintes

L'optimisation avec contraintes consiste à trouver le maximum ou le minimum d'une fonction tout en respectant certaines conditions exprimées sous forme d'égalités ou d'inégalités. Ces contraintes reflètent souvent des limites réelles, comme des ressources limitées ou des règles à suivre.

Un problème d'optimisation mono-objectif avec contrainte peut être présenté comme suit :

$$P \quad \begin{array}{l} \min_{x \in X} f(x) \\ \text{s.c.} \quad \begin{cases} h_i(x) = 0, & \forall i \in \{1, \dots, p\}, \\ g_j(x) \leq 0, & \forall j \in \{1, \dots, q\}. \end{cases} \end{array}$$

Soit $C \subset \mathbb{R}^n$ un sous-ensemble non vide défini par des contraintes d'égalités et d'inégalités. On note :

$$C = \{x \in \mathbb{R}^n : h(x) = 0, g(x) \leq 0\},$$

où $h : \mathbb{R}^n \rightarrow \mathbb{R}^p$ et $g : \mathbb{R}^n \rightarrow \mathbb{R}^q$ sont des fonctions continues.

L'ensemble C est appelé ensemble des points admissibles. Tout point $x \in \mathbb{R}^n$ vérifiant $x \in C$ est appelé un point admissible (ou point réalisable) du problème P .

Définition 1.8. [15] Une solution optimale du problème P est une solution réalisable x^* qui minimise $f(x)$ sur l'ensemble des solutions admissibles.

1.4.1 Existence et unicité

Théorème. [22]

Soit f une fonction continue et C un sous-ensemble fermé non vide de \mathbb{R}^n . Si l'une des conditions suivantes est satisfaite :

1. C est borné.
2. f est coercive ($\lim_{\|x\| \rightarrow +\infty} f(x) = +\infty$).

Alors le problème P admet au moins une solution.

1.4.2 Conditions d'optimalité du premier ordre

Tout comme le cas d'optimisation sans contrainte, on a besoin des conditions d'optimalité du premier et du second ordre.

Théorème. (*Condition nécessaire du premier ordre*)[22]

Soit f une fonction différentiable, et soit $C \subset \mathbb{R}^n$ un ensemble convexe fermé. Alors, toute solution x^* du problème (P) vérifie la condition nécessaire d'optimalité du premier ordre :

$$\forall x \in C, \quad \langle \nabla f(x^*), x - x^* \rangle \geq 0.$$

Remarque 1.4. Dans le cas convexe (f est convexe et C un convexe fermé) la condition précédente est nécessaire et suffisante pour que x^* soit une solution de P.

Conditions d'optimalité du premier ordre en cas de contraintes d'égalité

Avec uniquement des contraintes d'égalité, le problème P s'écrit comme suit :

$$P' \quad \begin{array}{l} \min_{x \in C} f(x) \\ \text{s.c} \quad \left\{ \begin{array}{l} g_i(x) = 0, \quad \forall i = 1, \dots, p. \end{array} \right. \end{array}$$

On pose

$$G(x) = (g_1(x), g_2(x), \dots, g_m(x))$$

$$G : \mathbb{R}^n \rightarrow \mathbb{R}^m, \quad x \mapsto G(x) = (g_1(x), g_2(x), \dots, g_m(x)).$$

La matrice jacobienne de G est donné par :

$$J_G(x)_{(m \times n)} = \begin{bmatrix} \frac{\partial g_1}{\partial x_1} & \frac{\partial g_1}{\partial x_2} & \dots & \frac{\partial g_1}{\partial x_n} \\ \frac{\partial g_2}{\partial x_1} & \frac{\partial g_2}{\partial x_2} & \dots & \frac{\partial g_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial g_m}{\partial x_1} & \frac{\partial g_m}{\partial x_2} & \dots & \frac{\partial g_m}{\partial x_n} \end{bmatrix}$$

Définition 1.9. [22] On dit qu'un point x^* est régulier (ou qu'il vérifie la qualification des contraintes) si le rang de la matrice jacobienne $J_G(x^*) = m$.

Remarque 1.5. Un point qui n'est pas régulier est dit singulier.

Si tout les points admissibles sont réguliers, on dira que la qualification des contraintes est vérifié par tout.

Théorème. [22] Considérons le problème P' et supposons que les fonctions f et g_i , pour $i = 1, \dots, p$, sont de classe C^1 .

Si x^* est une solution du problème P' , alors il existe μ_1^*, \dots, μ_p^* , appelés **multiplicateurs de Lagrange**, tels que :

$$\nabla f(x^*) + \sum_{i=1}^p \mu_i^* \nabla g_i(x^*) = 0$$

Définition 1.10. [22] On appelle **Lagrangien** du problème (P') la fonction $L : \mathbb{R}^n \times \mathbb{R}^p \rightarrow \mathbb{R}$, définie par :

$$L(x, \mu) = f(x) + \sum_{i=1}^p \mu_i g_i(x)$$

Conditions d'optimalité du premier ordre, en cas de contraintes inégalités

On définit un problème d'optimisation avec contrainte inégalité :

$$P'' \quad \begin{array}{l} \min_{x \in C} f(x) \\ \text{s.c.} \quad \left\{ g_i(x) \leq 0, \quad \forall i = 1, \dots, q. \right. \end{array}$$

On note

$$X = \{x \in \mathbb{R}^n \mid g_i(x) \leq 0, \forall i = 1, \dots, q\}.$$

L'ensemble des points réalisables.

Définition 1.11. [22] La contrainte $g_i(x) \leq 0$ au point $x^0 \in X$ est dite **active (saturée)** si $g_i(x^0) = 0$ et **inactive** si $g_i(x^0) < 0$.

Conditions d'optimalité du premier ordre, cas égalité et inégalité

Reprenons le problème P :

$$P \quad \begin{array}{l} \min_{x \in X} f(x) \\ \text{s.c.} \quad \left\{ \begin{array}{l} h_i(x) = 0, \quad \forall i \in \{1, \dots, p\}, \\ g_j(x) \leq 0, \quad \forall j \in \{1, \dots, q\}. \end{array} \right. \end{array}$$

$$X = \{x \in \mathbb{R}^n \mid h_i(x) = 0, \forall i = 1, \dots, p, \text{ et } g_j(x) \leq 0, \forall j = 1, \dots, q\}$$

Définition 1.12. [22]

On dit qu'un point admissible x^0 est **régulier** (c'est-à-dire qu'il vérifie la qualification des contraintes) pour P si :

- Il est réalisable : $h(x^0) = 0$ et $g(x^0) \leq 0$.
- Les gradients $\nabla h_i(x^0)$ et $\nabla g_j(x^0)$ associés aux contraintes actives sont linéairement indépendants.

Théorème. (Condition de Karush-Kuhn-Tucker) [22]

On suppose que les fonctions f , h_i pour $i = 1, \dots, p$, et g_j pour $j = 1, \dots, q$ sont de classe

C^1 . Soit x^* une solution du problème P . Alors, il existe $\lambda^* = (\lambda_1^*, \lambda_2^*, \dots, \lambda_p^*) \in \mathbb{R}^p$, $\mu^* = (\mu_1^*, \mu_2^*, \dots, \mu_q^*) \in \mathbb{R}_+^q$ et $\mu_0^* \in \mathbb{R}_+$ tels que :

$$\begin{cases} \forall j \in \{0, 1, \dots, q\}, & \mu_j^* \geq 0 \\ h_i(x^*) = 0 & \text{pour } i = 1, \dots, p \text{ et } g_j(x^*) \leq 0 \text{ pour } j = 1, \dots, q \\ \forall j \in \{1, \dots, q\}, & \mu_j^* g_j(x^*) = 0 \\ \mu_0^* \nabla f(x^*) + \sum_{i=1}^p \lambda_i^* \nabla h_i(x^*) + \sum_{j=1}^q \mu_j^* \nabla g_j(x^*) = 0 \end{cases}$$

Ces équations sont appelées les conditions de KKT.

Théorème. (Lagrange-KKT) [22] Supposons que les fonctions f , h_i pour $i = 1, \dots, p$, et g_j pour $j = 1, \dots, q$ sont de classe C^1 .

Soit x^* un minimum local régulier.

Alors, il existe $\lambda^* = (\lambda_1^*, \lambda_2^*, \dots, \lambda_p^*) \in \mathbb{R}^p$, $\mu^* = (\mu_1^*, \mu_2^*, \dots, \mu_q^*) \in \mathbb{R}_+^q$ et $\mu_0^* \in \mathbb{R}_+$ tels que :

$$\begin{cases} \nabla f(x^*) + \sum_{i=1}^p \lambda_i^* \nabla h_i(x^*) + \sum_{j=1}^q \mu_j^* \nabla g_j(x^*) = 0, \\ \mu_j^* g_j(x^*) = 0, \quad \forall j = 1, \dots, q. \end{cases} \quad (1.4)$$

Remarque 1.6. Dans le cas d'un problème convexe (f et g sont convexes et h est linéaire), la **condition nécessaire du 1^{er} ordre de Lagrange-KKT** devient suffisante, et le minimum local est aussi un minimum global.

1.4.3 Conditions d'optimalité du deuxième ordre

Les conditions d'optimalité du premier ordre permettent d'identifier les candidats potentiels à la solution de P . Ensuite, les conditions d'optimalité du second ordre permettent de réduire davantage ce nombre de candidats.

Théorème. [22] Supposons que f, h et g sont de classe C^2 et x^* est un minimum local, et les conditions de la qualifications des contraintes sont vérifiées.

Alors, il existe $\lambda^* = (\lambda_1^*, \lambda_2^*, \dots, \lambda_p^*) \in \mathbb{R}^p$ et $\mu^* = (\mu_1^*, \mu_2^*, \dots, \mu_q^*) \in \mathbb{R}_+^q$ tels que :

— Les relations de K.K.T. sont satisfaites.

— Pour toute direction $d \in \mathbb{R}^n \setminus \{0\}$ vérifiant :

$$\begin{cases} \langle \nabla h_i(x^*), d \rangle = 0, & \text{pour } i = 1, \dots, p, \\ \langle \nabla g_j(x^*), d \rangle = 0, & \text{pour } j \in I^+(x^*), \\ \langle \nabla g_j(x^*), d \rangle \leq 0, & \text{pour } j \in I(x^*) \setminus I^+(x^*). \end{cases}$$

où $I^+(x^*) = \{j \in \{1, 2, \dots, q\} \mid g_j(x^*) = 0 \text{ et } \mu_j^* > 0\}$, on a :

$$\langle H_{xx}L(x^*, \lambda^*, \mu^*)d, d \rangle \geq 0, \quad (2.11)$$

avec

$$H_{xx}L(x^*, \lambda^*, \mu^*) = Hf(x^*) + \sum_{i=1}^p \lambda_i^* Hh_i(x^*) + \sum_{j=1}^q \mu_j^* Hg_j(x^*) \quad (1.5)$$

désignant la seconde dérivée de L au point (x^*, λ^*, μ^*) .

Définition 1.13. [22] L'ensemble $I^+(x^*)$ correspond aux contraintes fortement actives. Lorsque $I^+(x^*)$ coïncide avec $I(x^*)$, c'est-à-dire lorsque $g_j(x^*) = 0$ et $\mu_j^* > 0$, on parle de stricte complémentarité.

Théorème. (Conditions suffisante du deuxième ordre) [22]

Supposons que f , h et g sont de classe C^2 . Soit $x^* \in \mathbb{R}^n$ un point vérifiant les conditions de KKT avec les multiplicateurs (λ^*, μ^*) .

Si la matrice hessienne du lagrangien au point (x^*, λ^*, μ^*) , notée $\nabla_{xx}^2 L(x^*, \lambda^*, \mu^*)$, est définie positive sur le sous-espace :

$$D = \{d \in \mathbb{R}^n \setminus \{0\} \mid \langle \nabla h_i(x^*), d \rangle = 0, \forall i = 1, \dots, p, \text{ et } \langle \nabla g_j(x^*), d \rangle = 0, \forall j \in I^+(x^*)\}$$

alors x^* est un minimum strict de f sur C .

1.4.4 Exemples

Exemple 1 [4]

Nous cherchons à résoudre le problème suivant :

$$\begin{aligned} \min f(x, y) &= xy \\ \text{s.c } \begin{cases} h(x, y) = 4x^2 + y^2 - 4 = 0 \\ (x, y) \in \mathbb{R}^2 \end{cases} \end{aligned}$$

Le Lagrangien associé est défini par :

$$L(x, y, \lambda) = f(x, y) + \lambda h(x, y), \quad \forall \lambda \in \mathbb{R}$$

$$L(x, y, \lambda) = xy + \lambda(4x^2 + y^2 - 4).$$

Les points critiques sont obtenus en résolvant :

$$\nabla L(x, y, \lambda) = 0$$

Ce qui donne le système suivant :

$$\begin{cases} \frac{\partial L}{\partial x} = y + 8\lambda x = 0 & (1) \\ \frac{\partial L}{\partial y} = x + 2\lambda y = 0 & (2) \\ \frac{\partial L}{\partial \lambda} = 4x^2 + y^2 - 4 = 0 & (3) \end{cases}$$

Des équations (1) et (2) on a :

$$\begin{cases} y = -8\lambda x \\ x = -2\lambda y \end{cases}$$

Substitutions x dans l'équation de y :

$$y = -8\lambda(-2\lambda y) = 16\lambda^2 y$$

Ce qui donne deux cas :

Si $y = 0$, alors $x = 0$, mais cela ne satisfait pas l'équation (3), donc il n'y a pas de solution dans ce cas.

Si $\lambda = \frac{1}{4}$ ou $\lambda = -\frac{1}{4}$, on obtient :

$$x = -2\lambda y = \begin{cases} -\frac{1}{2}y & \text{si } \lambda = \frac{1}{4}, \\ \frac{1}{2}y & \text{si } \lambda = -\frac{1}{4}. \end{cases}$$

En remplaçant dans l'équation (3) on obtient :

$$2y^2 = 4$$

Donc :

$$y^2 = 2 \quad \Rightarrow \quad y = \pm\sqrt{2}.$$

Finalement, les solutions candidates sont :

$$\left(\frac{\sqrt{2}}{2}, \sqrt{2}\right), \quad \left(\frac{\sqrt{2}}{2}, -\sqrt{2}\right), \quad \left(-\frac{\sqrt{2}}{2}, \sqrt{2}\right), \quad \left(-\frac{\sqrt{2}}{2}, -\sqrt{2}\right).$$

Pour déterminer la nature des points critiques, évaluons $f(x_0, y_0)$:

(x_0, y_0)	$\left(\frac{\sqrt{2}}{2}, \sqrt{2}\right)$	$\left(\frac{\sqrt{2}}{2}, -\sqrt{2}\right)$	$\left(-\frac{\sqrt{2}}{2}, \sqrt{2}\right)$	$\left(-\frac{\sqrt{2}}{2}, -\sqrt{2}\right)$
$f(x_0, y_0)$	1	-1	-1	1

Les deux points critiques $\left(\frac{\sqrt{2}}{2}, -\sqrt{2}\right)$ et $\left(-\frac{\sqrt{2}}{2}, \sqrt{2}\right)$ sont des points minimaux.

Exemple 2 [15]

$$\min f(x_1, x_2) = x_2 + x_3$$

$$s.c \begin{cases} x_1 + x_2 + x_3 = 1 \\ x_1^2 + x_2^2 + x_3^2 \leq 1 \\ (x_1, x_2, x_3) \in \mathbb{R}^3 \end{cases}$$

Le **Lagrangien** associé est donné par :

$$L(x, \lambda, \mu) = x_2 + x_3 + \lambda(x_1 + x_2 + x_3 - 1) + \mu(x_1^2 + x_2^2 + x_3^2 - 1)$$

Les **conditions KKT** s'écrivent :

$$\left\{ \begin{array}{l} \frac{\partial L}{\partial x_1} = \lambda + 2\mu x_1 = 0 \quad (1) \\ \frac{\partial L}{\partial x_2} = 1 + \lambda + 2\mu x_2 = 0 \quad (2) \\ \frac{\partial L}{\partial x_3} = 1 + \lambda + 2\mu x_3 = 0 \quad (3) \\ x_1 + x_2 + x_3 - 1 = 0 \quad (4) \\ x_1^2 + x_2^2 + x_3^2 - 1 \leq 0 \quad (5) \\ \mu(x_1^2 + x_2^2 + x_3^2 - 1) = 0 \quad (6) \\ \mu \geq 0 \quad (7) \end{array} \right.$$

Cas 1 : $\mu = 0$

$$\left\{ \begin{array}{l} (1) \Rightarrow \lambda = 0 \\ (2) \Rightarrow \lambda = -1 \\ (3) \Rightarrow \lambda = -1 \end{array} \right. \quad \text{Contradiction donc c'est un cas rejeté.}$$

Cas 2 : **contrainte active**, i.e. $x_1^2 + x_2^2 + x_3^2 = 1$

À partir des équations (1), (2) et (3), on a :

$$\left\{ \begin{array}{l} x_1 = -\frac{\lambda}{2\mu} \\ x_2 = x_3 = -\frac{\lambda + 1}{2\mu} \end{array} \right. \quad (8)$$

Substituons dans la contrainte (4) :

$$x_1 + x_2 + x_3 = -\frac{\lambda}{2\mu} - 2 \cdot \frac{\lambda + 1}{2\mu} = -\frac{1}{\mu} - \frac{3\lambda}{2\mu}$$

On impose :

$$x_1 + x_2 + x_3 = 1 \Rightarrow -\frac{1}{\mu} - \frac{3\lambda}{2\mu} = 1 \Rightarrow \lambda = -\frac{2}{3}\mu\left(1 + \frac{1}{\mu}\right) \quad (9)$$

Remplaçons (9) dans (8) :

$$\lambda = -\frac{2}{3}(\mu + 1) \Rightarrow x_1 = \frac{1}{3} + \frac{1}{3\mu}, \quad x_2 = x_3 = \frac{1}{3} - \frac{1}{6\mu}$$

Substituons dans la contrainte (5) rendue active :

$$\left(\frac{1}{3} + \frac{1}{3\mu}\right)^2 + 2\left(\frac{1}{3} - \frac{1}{6\mu}\right)^2 = 1$$

Posons $y = \frac{1}{3\mu}$. L'équation devient :

$$\left(y + \frac{1}{3}\right)^2 + 2\left(\frac{1}{3} - \frac{y}{2}\right)^2 = 1$$

Développement :

$$\begin{aligned} y^2 + \frac{2}{3}y + \frac{1}{9} + 2\left(\frac{1}{9} - \frac{y}{3} + \frac{y^2}{4}\right) &= y^2 + \frac{2}{3}y + \frac{1}{9} + \frac{2}{9} - \frac{2y}{3} + \frac{y^2}{2} \\ &= \frac{3}{2}y^2 + \frac{1}{3} = 1 \quad \Rightarrow \quad y^2 = \frac{4}{9} \quad \Rightarrow \quad y = \pm\frac{2}{3} \quad \Rightarrow \quad \mu = \pm\frac{1}{2} \end{aligned}$$

La solution $\mu = -\frac{1}{2}$ est rejetée car elle viole la condition (7).

Solution finale

$$\mu = \frac{1}{2}, \quad x_1 = 1, \quad x_2 = x_3 = 0, \quad \lambda = -1$$

La solution satisfait toutes les conditions KKT.

$$\Rightarrow f(1, 0, 0) = 0$$

Conclusion : Le minimum global est atteint en $(1, 0, 0)$, avec une valeur minimale de $f = 0$.

Chapitre 2

Optimisation multi-objectif

2.1 Introduction

Un problème multi-objectif consiste à trouver un équilibre entre plusieurs objectifs qui peuvent être en conflit. Contrairement aux problèmes où l'on optimise un seul objectif, ici, on doit prendre en compte plusieurs critères en même temps. Cela rend la résolution plus compliquée, car il n'existe pas toujours une seule meilleure solution. Une option peut être avantageuse pour un objectif mais moins bonne pour un autre. Ainsi, il faut chercher un compromis plutôt qu'une solution parfaite pour tous les objectifs à la fois.

2.2 Les structures de préférences

2.2.1 Les relations binaires et les ordres

Définition 2.1. [24] Une relation binaire R sur un ensemble X est un ensemble de paires ordonnées (x, y) avec $x \in X$ et $y \in X$.

Définition 2.2. [24] Une relation binaire R sur un ensemble X est dite :

- **Réflexive** si xRx pour tout $x \in X$.
- **Irréflexive** si $x \not R x$ pour tout $x \in X$ (autrement dit aucun élément n'est en relation avec lui-même).
- **Symétrique** si xRy implique yRx pour tous $x, y \in X$.
- **Asymétrique** si xRy implique $y \not R x$ pour tous $x, y \in X$.
- **Antisymétrique** si xRy et yRx impliquent $x = y$ pour tous $x, y \in X$.
- **Transitive** si xRy et yRz impliquent xRz pour tous $x, y, z \in X$.
- **Négativement transitive** si $x \not R y$ et $y \not R z$ impliquent $x \not R z$ pour tous $x, y, z \in X$.
- **Connexe** si pour tous $x, y \in X$, on a xRy ou yRx .

- **Faiblement connexe** si pour tous $x, y \in X$ avec $x \neq y$, on a xRy ou yRx .
- **Acyclique** s'il n'existe pas de suite $x_1, x_2, \dots, x_m \in X$ telle que :

$$x_1Rx_2, \quad x_2Rx_3, \quad \dots, \quad x_mRx_1.$$

Définition 2.3. [24] Un ordre partiel est une relation binaire asymétrique et transitive.

Définition 2.4. [24] Un ordre faible est une relation binaire asymétrique et négativement transitive.

Définition 2.5. [24] Un ordre strict est une relation binaire asymétrique, négativement transitive et faiblement connexe.

2.2.2 Les structures de préférences

Définition 2.6. [24] Une **structure de préférence** R est définie sur un ensemble X par :

- une relation binaire P , **asymétrique** ;
- une relation binaire I , **réflexive et symétrique**.

Les relations P et I sont deux sous-ensembles disjoints de $X \times X$. On définit R par :

$$xRy \iff xPy \text{ ou } xIy, \quad \text{soit } R = P \cup I.$$

- Si xPy , alors on dit que " x est préférable à y ", et on note :

$$x \succ y \quad \text{ou} \quad y \prec x.$$

- Si xIy , alors on dit que " x est indifférent à y ", et on note :

$$x \sim y.$$

La relation N est donc interprétée comme une relation de **préférence ou d'indifférence**. Ainsi, xNy signifie que " x est préféré ou indifférent à y ", et on note :

$$x \succeq y \quad \text{ou} \quad y \preceq x.$$

2.2.3 La structure de préordre complet

Cette structure est utilisée dans la plupart des modèles classiques d'économie. Dans ce contexte :

- La relation de préférence P est un **ordre faible**.
- La relation d'indifférence I correspond aux paires qui n'appartiennent pas à P , soit :

$$I = \{(x, y) \in X \times X \setminus (x \succ y) \text{ et } (y \succ x)\}.$$

Si l'ensemble des indifférences est vide ($I = \emptyset$), alors la relation R est un **ordre strict**.

2.2.4 La structure d'ordre partiel et de préordre partiel

La structure d'ordre partiel est utilisée lorsqu'on range du meilleur au moins bon, sans ex aequo, les éléments de certains sous-ensembles d'alternatives. Cette structure autorise les situations d'incomparabilité entre deux alternatives.

La structure de préordre partiel, quant à elle, est utilisée lorsqu'on range du meilleur au moins bon, avec d'éventuels ex aequo, les éléments de certains sous-ensembles d'alternatives. Cette structure autorise les situations d'incomparabilité et d'indifférence entre deux alternatives.

2.3 Définition d'un problème multi-objectif (PMO)

Définition 2.7. [7] Un problème multi-objectif (multicritère) peut être définie comme un problème qui consiste à trouver l'action qui satisfait un ensemble de contraintes et optimise un vecteurs de fonctions objectifs.

Un PMO est présenté comme suit :

$$(PMO) \quad \begin{cases} \min_{x \in X} F(x) = [f_1(x), f_2(x) \dots f_k(x)] \\ h_j(x) = 0, \quad \forall j \in P = \{1, \dots, p\}, \\ g_i(x) \leq 0, \quad \forall i \in M = \{1, \dots, m\}. \end{cases}$$

Une action (ou un vecteur de décisions) sera notée :

$$x = (x_1, x_2, \dots, x_n)$$

avec x_i les variables du problème et n le nombre de variables.

Les contraintes seront notées :

$$g_i(x) \quad \text{pour } i = 1, \dots, m$$

avec m le nombre de contraintes.

Le vecteur de fonction objectif sera noté f :

$$f(x) = (f_1(x), f_2(x), \dots, f_k(x))$$

avec f_i les objectifs ou critères de décision et k le nombre d'objectifs.

Un problème d'optimisation recherche l'action x^* telle que les contraintes soient satisfaites :

$$g_i(x^*) \leq 0, \quad \forall i = 1, \dots, m$$

et qui optimise la fonction objectif :

$$f(x^*) = (f_1(x^*), f_2(x^*), \dots, f_k(x^*)).$$

L'union des domaines de définition de chaque variable et des contraintes forment un ensemble E que nous appelons **l'ensemble des actions réalisables** et nous appellerons F **l'ensemble des objectifs réalisables**.

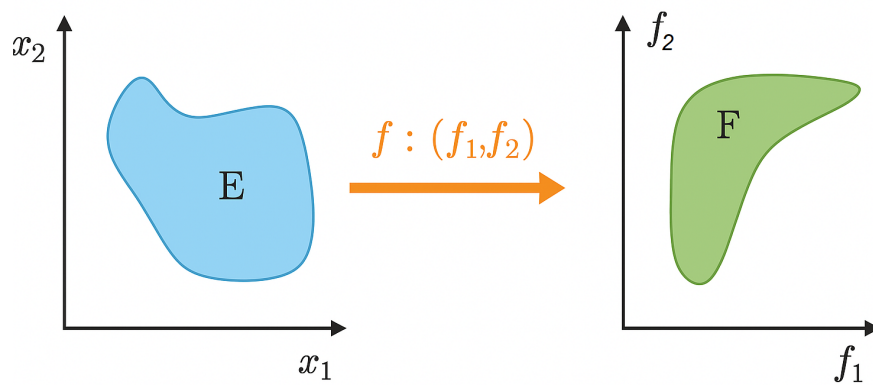


FIGURE 2.1 – Définition de E, F, f .

2.3.1 La multiplicités des solutions

Quand on cherche la meilleure solution à un problème avec plusieurs objectifs, on espère souvent en trouver une seule. Mais en réalité, il y en a souvent plusieurs, car certains objectifs se contredisent.

En général, on obtient donc un grand nombre de solutions. Ces solutions ne seront pas parfaites, car elles ne peuvent pas optimiser tous les objectifs en même temps.

Un concept important ici est l'équilibre. Les solutions trouvées sont un équilibre : elles améliorent certains objectifs mais en dégradent d'autres.

2.3.2 La difficulté dans un problème multi-objectif

La difficulté principale dans un problème d'optimisation multi-objectif est qu'il n'existe pas de définition de la solution optimal. Le décideur peut simplement exprimer le fait qu'une solution est préférable à une autre mais il n'existe pas une solution meilleure que les autres. Résoudre un problème à plusieurs objectifs ne signifie pas trouver une solution unique parfaite, mais plutôt identifier plusieurs solutions acceptables parmi lesquelles on ne peut pas établir de classement absolu. Ces méthodes aident à la décision, mais le choix final revient toujours au décideur.

La communauté scientifique a adopté deux approches pour traiter ces problèmes :

1. Transformer le problème multi-objectif en un problème à un seul objectif, même si cela peut simplifier excessivement la situation.
2. Prendre en compte tous les critères et proposer plusieurs solutions, laissant le décideur choisir ensuite.

2.3.3 Méthodes de résolution d'un problème multi-objectif

Il existe de nombreuses méthodes en optimisation multi-objectif, regroupées en cinq grandes catégories :

- les méthodes scalaires,
- les méthodes interactives,
- les méthodes floues,
- les méthodes basées sur des métaheuristiques,
- les méthodes d'aide à la décision.

Ces méthodes peuvent également être classées selon la manière dont les préférences de l'utilisateur sont prises en compte, en trois grandes familles :

1. **Méthodes à préférence *a priori*** : l'utilisateur exprime ses préférences avant le processus d'optimisation. On retrouve ici notamment les méthodes d'agrégation, qui fusionnent les fonctions objectif en une seule.
2. **Méthodes à préférence *progressive*** : l'utilisateur affine ses préférences au cours de l'optimisation. Cela correspond aux méthodes interactives.
3. **Méthodes à préférence *a posteriori*** : l'utilisateur choisit une solution parmi toutes celles générées par l'optimisation, après coup. Ces méthodes fournissent une surface de compromis à la fin du processus.

Certaines méthodes peuvent combiner plusieurs approches. Par exemple, une méthode à préférence *a priori* utilisant des préférences aléatoires peut générer un grand nombre de solutions, parmi lesquelles l'utilisateur sélectionnera ensuite la plus appropriée. On obtient alors une méthode à préférence *a posteriori*.

2.4 La dominance

Définition 2.8. (la relation de dominance) [8] On dit que le vecteur x domine le vecteur y si :

- x est au moins aussi bon que y dans tous les objectifs.
- x est strictement meilleur que y dans au moins un objectif.

Les solutions qui dominent les autres mais ne se dominent pas entre elles sont appelées solutions optimales au sens de Pareto (ou solutions non dominées).

En pratique on trouve que la dominance au sens de Pareto est la plus connue et utilisée. A l'aide de cette relation, les vecteurs de décisions x et y peuvent être classés de la manière suivante :

- $x \prec y$ (x domine y) si $\forall i \in I, x_i < y_i$
- $x \preceq y$ (x domine faiblement y) si $\forall i \in I, x_i \leq y_i$
- $x \sim y$ (x est équivalent à y) si $x \not\prec y$ et $y \not\prec x$

Définition 2.9. [25] Un point x est dit **faiblement non dominé** s'il n'existe pas de point $y \in E$ tel que :

$$f_i(y) < f_i(x), \forall i = 1, \dots, k.$$

Un point x est dit **fortement non dominé** s'il n'existe pas de point $y \in E$ tel que :

$$f_i(y) \leq f_i(x), \forall i = 1, \dots, k \quad \text{avec au moins un } i \text{ tel que } f_i(y) < f_i(x).$$

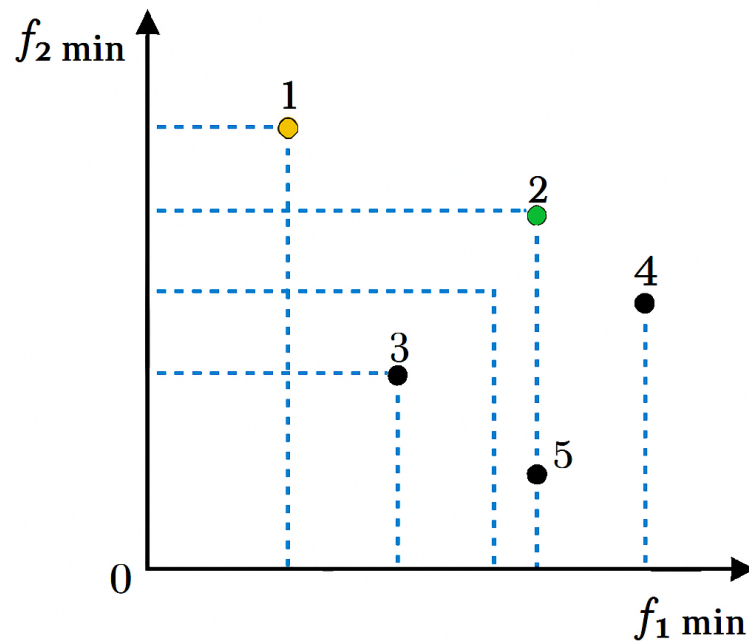


FIGURE 2.2 – Exemple de dominance.

Dans cette exemple aucun point ne domine les points 1, 3 et 5. Alors que le point 2 est dominé par le point 1 et le point 4 est dominé par 3 et 5.

2.4.1 Propriétés de la relation de dominance

La relation binaire de dominance, telle qu'elle est définie ci-dessous :

- Elle n'est pas réflexive, car une solution (x, y) ne se domine pas elle-même.
- Elle n'est pas symétrique, car on n'a jamais $(x, y) \succ (y, x)$ et $(y, x) \succ (x, y)$.
- Elle n'est pas antisymétrique du fait de l'existence des solutions Pareto optimales.
- Elle est transitive, car si $(x, y) \succ (y, z)$ et $(y, z) \succ (z, x)$, alors $(x, z) \succ (z, x)$.

La relation de dominance est donc une relation d'ordre partiel strict sur l'espace des décisions.

2.5 Les points particuliers idéal et nadir

2.5.1 Point idéal

Définition 2.10. [24] Le point idéal est dans \mathbb{R}^n , le point de coordonnées (z_1^*, \dots, z_p^*) , avec

$$z_i^* = \max\{f_i(x) \mid x \in X, i = 1, \dots, p\}$$

2.5.2 Point nadir

Définition 2.11. [24] Le point nadir est dans \mathbb{R}^n , le point de coordonnées $(z_1^{\text{nad}}, \dots, z_p^{\text{nad}})$, avec

$$z^N = \min\{f_i(x) \mid x \in X, i = 1, \dots, p\}$$

Autrement dit les coordonnées de ce point correspondent aux pires valeurs obtenues par chaque fonction objectif lorsque l'on restreint l'espace des solutions à la surface de compromis.

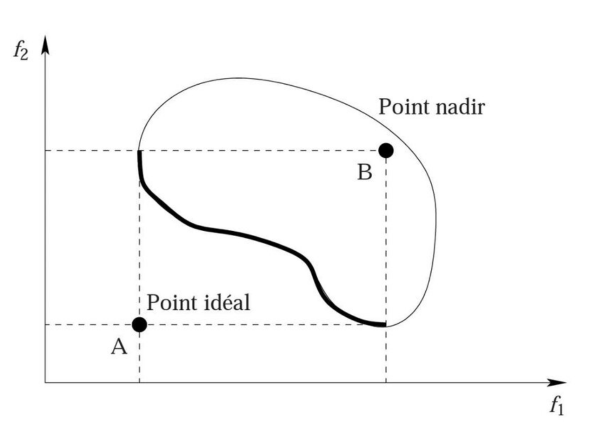


FIGURE 2.3 – Représentation du point idéal et nadir.

2.5.3 Matrice de gain [25]

Pour calculer cette matrice :

1. On optimise séparément chaque fonction-objectif sur l'ensemble D des solutions admissibles.

On obtient p solutions : x_1^*, \dots, x_p^* .

2. On calcule, en chaque point x_j^* ($j = 1, 2, \dots, p$), la valeur de chacune des p fonctions-objectifs.

On obtient une matrice de gain dont la forme générale est donnée par :

$$G = \begin{pmatrix} g_{11} & g_{12} & \cdots & g_{1p} \\ g_{21} & g_{22} & \cdots & g_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ g_{p1} & g_{p2} & \cdots & g_{pp} \end{pmatrix}$$

Avec $g_{ij} = f_i(x_j^*)$, pour tout couple $(i, j) \in \{1, 2, \dots, p\}^2$.

Sur la diagonale de la matrice des gains, on reconnaît les p composantes du vecteur idéal I (i.e. $g_{kk} = I_k, \forall k = 1, 2, \dots, p$).

La $i^{\text{ème}}$ composante du point nadir est identifiée comme la plus petite des valeurs parmi celles de la $i^{\text{ème}}$ ligne de la matrice G :

$$N_i = \min_{1 \leq j \leq p} (g_{ij}) \quad , \quad \forall i = 1, 2, \dots, p$$

Évidemment, cette matrice n'est pas toujours univoquement déterminée puisqu'une solution x_j^* n'est pas toujours unique.

2.5.4 Exemple d'illustration

Pour bien illustrer tous ces points, nous introduisons le problème suivant :

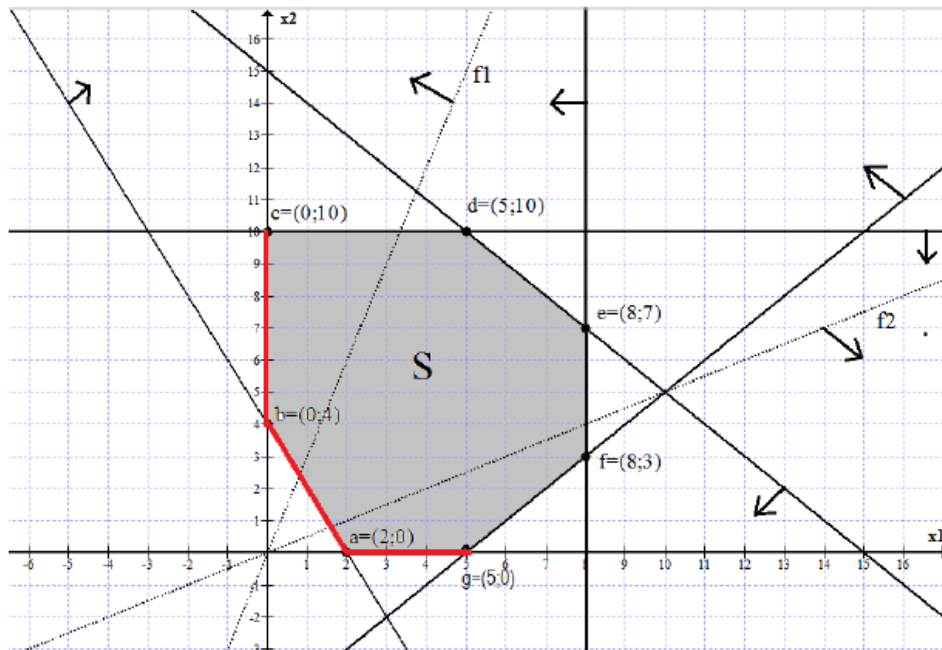
$$\begin{array}{l} \text{(PMO)} \quad \left\{ \begin{array}{l} \text{maximiser } f_1(x) = -3x_1 + x_2 \\ \text{maximiser } f_2(x) = x_1 - 2x_2 \end{array} \right. \\ \text{sous les contraintes} \quad \left\{ \begin{array}{l} x_1 + x_2 \leq 15 \\ 2x_1 + x_2 \geq 4 \\ x_1 - x_2 \leq 5 \\ x_1 \leq 8 \\ x_2 \leq 10 \\ x_1, x_2 \geq 0 \end{array} \right. \end{array}$$

1. Représentation graphique de l'espace de décision S

La figure (Fig. 2.4) montre l'ensemble réalisable S dans l'espace de décision. L'ensemble S est un polyèdre convexe délimité par les 7 points extrêmes :

$$a = (2, 0), \quad b = (0, 4), \quad c = (0, 10), \quad d = (5, 10), \quad e = (8, 7), \quad f = (8, 3), \quad g = (5, 0).$$

On pose : $X_{\text{ex}} = \{a, b, c, d, e, f, g\}$ l'ensemble des points extrêmes de S .

FIGURE 2.4 – L'espace de décision S .

2. Représentation graphique de l'espace des critères Y

Dans le cas linéaire, pour obtenir l'ensemble réalisable Y dans l'espace des critères, il suffit de calculer l'image de chaque point extrême de l'ensemble de décision S afin d'obtenir l'ensemble Y .

La figure (Fig. 2.6) montre l'ensemble Y , qui est un polyèdre convexe délimité par les 7 points extrêmes suivants :

$$F(a) = (-6, 2),$$

$$F(b) = (4, 8),$$

$$F(c) = (10, -20),$$

$$F(d) = (-5, -15),$$

$$F(e) = (-17, -6),$$

$$F(f) = (-21, 2),$$

$$F(g) = (-15, 5).$$

3. Calculons le point idéal, le point nadir et la matrice des gains.

Le point idéal $z^* = (z_1^*, z_2^*)$ est le point qui maximise séparément les deux critères, c'est-à-dire :

$$z_1^* = \max_{x \in S} f_1(x) \quad \text{et} \quad z_2^* = \max_{x \in S} f_2(x).$$

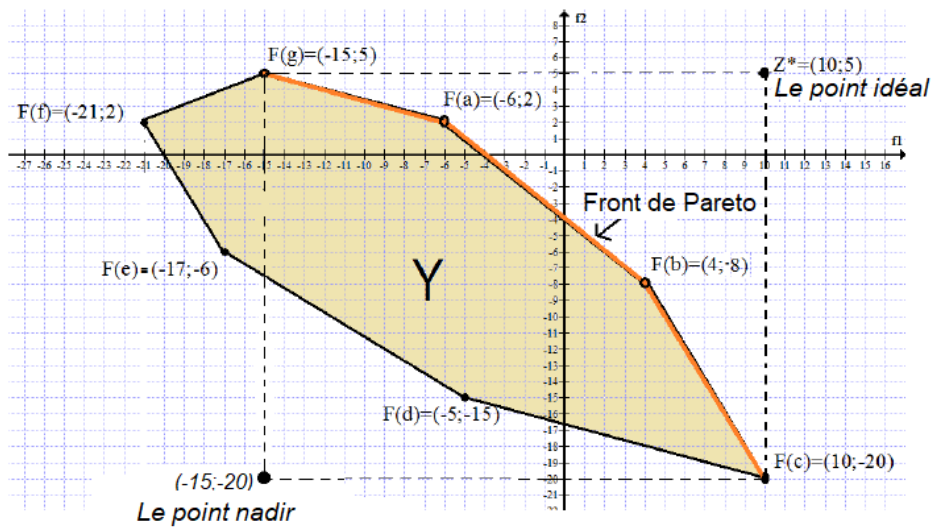


FIGURE 2.5 – L’espace des critères Y.

Il s’agit donc de résoudre séparément les deux programmes linéaires suivants :

(P1)

$$\text{Maximiser } f_1(x) = -3x_1 + x_2$$

$$\text{sous les contraintes } \begin{cases} x_1 + x_2 \leq 15 \\ 2x_1 + x_2 \geq 4 \\ x_1 - x_2 \leq 5 \\ x_1 \leq 8 \\ x_2 \leq 10 \\ x_1, x_2 \geq 0 \end{cases}$$

(P2)

$$\text{Maximiser } f_2(x) = x_1 - 2x_2$$

$$\text{sous les contraintes } \begin{cases} x_1 + x_2 \leq 15 \\ 2x_1 + x_2 \geq 4 \\ x_1 - x_2 \leq 5 \\ x_1 \leq 8 \\ x_2 \leq 10 \\ x_1, x_2 \geq 0 \end{cases}$$

Graphiquement, la figure (Fig. 2.4) montre que la solution de (P1) est le point

$$c = (0, 10)$$

et son image dans l'espace des critères est

$$F(c) = (10, -20).$$

La solution de (P_2) est le point

$$g = (5, 0)$$

et son image est

$$F(g) = (-15, 5).$$

En effet, le point idéal est :

$$z^* = (10, 5).$$

On voit clairement sur la figure (Fig. 2.5) que les coordonnées du point de nadir sont :

$$z^{\text{nad}} = (-15, -20),$$

elles correspondent aux pires valeurs des fonctions objectif sur le front de Pareto.

La matrice des gains est :

$$M = \begin{pmatrix} 10 & -15 \\ -20 & 5 \end{pmatrix}$$

4. Le **front de Pareto**, ou *espace de compromis*, est représenté par la partie en rouge dans l'espace des critères (voir la figure Fig. 2.5).

2.6 Optimalité de Pareto et front de Pareto

2.6.1 Efficacité

Définition 2.12. [14, 24]

Une solution x est dite efficace (ou Pareto-optimale) du problème multi-objectifs (PMO) si

$$x \in X \quad \text{et} \quad \nexists x^* \in X \text{ tel que } f_i(x^*) \leq f_i(x) \quad \forall i \text{ et } \exists j \text{ tel que } f_j(x^*) < f_j(x).$$

Un point est efficace si son image par f est un vecteur critères non dominé. Le terme efficacité est aussi connu sous le nom de Pareto optimalité.

Une autre définition équivalente de l'efficacité est :

Définition 2.13. [24]

Une solution $x^* \in X$ est dite *efficace* pour le PMO si, pour tout $x \in X$, le fait que

$$f_i(x) > f_i(x^*), \quad \forall k \in \{1, \dots, p\}$$

implique qu'il existe au moins un indice $r \in \{1, \dots, p\}$, $r \neq k$, tel que :

$$f_r(x) < f_r(x^*).$$

2.6.2 Efficacité faible et efficacité fort

Définition 2.14. (Efficacité faible) [24] Une solution $x^* \in X$ est dite *faiblement efficace* s'il n'existe aucune solution $x \in X$ telle que : $f(x) < f(x^*)$.

Une solution est faiblement efficace si son vecteur critère n'est pas fortement dominé.

Définition 2.15. (Efficacité forte) [24] Une solution $x^* \in X$ est dite *fortement efficace* s'il n'existe aucune solution $x \in X$ telle que :

$$x \neq x^* \quad \text{et} \quad f(x) \leq f(x^*).$$

Une solution x^* est fortement efficace s'il n'existe pas d'autre solution telle que le vecteur critères, qui lui est associé, soit aussi bon que celui de x . Remarquons que l'efficacité forte implique l'efficacité qui implique à son tour l'efficacité faible.

2.6.3 Front de Pareto optimal

Définition 2.16. [27] Un point $x^* \in X$ est appelé *optimum de Pareto* (également appelé *point efficace global*, *point non dominé* ou *point non inférieur*) pour un problème d'optimisation multiobjectif si et seulement s'il n'existe aucun $x \in X$ tel que :

$$f(x) \prec f(x^*),$$

c'est-à-dire :

$$f_i(x) \leq f_i(x^*) \quad \text{pour tout } i \in \{1, \dots, p\} \quad \text{et} \quad f_j(x) < f_j(x^*) \quad \text{pour au moins un } j.$$

Toutes les solutions du *front de Pareto* sont des solutions situées sur le bord de l'espace des critères faisables. En d'autres termes, la frontière de Pareto est constituée de toutes les solutions x^* qui dominent les autres, mais ne sont dominées par aucune autre.

2.7 Exemple

Une société fabrique un produit T selon deux procédés de fabrication P_1 et P_2 . Chacun de ces procédés permet de fabriquer T mais fait intervenir deux type de matières premières M_1 et M_2 en quantités différentes. En effet, pour fabriquer 1 unités de T selon P_1 on a besoin de 2 unités de M_1 et 3 unités de M_2 alors que pour P_2 on a besoin de 3 unités de M_1 et de 2 unités de M_2 .

Les couts unitaires de production sont 1000 euros si l'on utilise P_1 et 3000 euros si l'on utilise P_2 .

On s'intéresse au plan de production journalier de T sachant que :

- Les disponibilités journalières de M1 sont de 30 unités, et il en est de même pour celles de M2.
- Le procédé P1 ne peut être utilisé qu'à concurrence de 8 unités.
- La société considère deux critères :
 - La quantité produite, qu'elle souhaite maximiser ;
 - Les coûts de production, qu'elle souhaite minimiser.

Ce problème peut s'écrire comme suit :

Considérons les variables de décision suivantes :

x_i : la quantité de produit T fabriquée selon le procédé P_i tel que ($i=1,2$).

$$\max f_1(x) = x_1 + x_2$$

$$\min f_2(x) = x_1 + 3x_2$$

$$(P) \quad \begin{cases} 2x_1 + 3x_2 \leq 30, \\ 3x_1 + 2x_2 \leq 30, \\ x_1 \leq 8, \\ x_1, x_2 \geq 0. \end{cases}$$

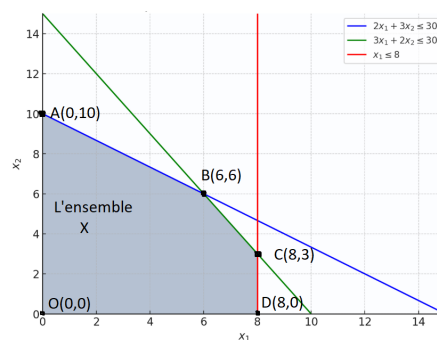


FIGURE 2.6 – Représentation de l'ensemble X des solutions réalisables dans l'espace des décisions.

Afin de représenter graphiquement l'ensemble des solutions dans l'espace des critères, il convient de déterminer l'image de chaque point $x = (x_1, x_2) \in X$ dans cet espace.

On obtient ainsi un ensemble F , défini comme suit :

$$F = \{f = (f_1, f_2) \in \mathbb{R}^2 \mid f_1(x) = x_1 + x_2, f_2(x) = x_1 + 3x_2, \forall x \in X\}$$

Les fonctions objectifs étant linéaire, il suffit de calculer l'image des 5 points extrêmes O, A, B, C et D. Il convient ensuite à relier ses 5 images pour délimiter l'ensemble des solutions dans l'espace des critères.

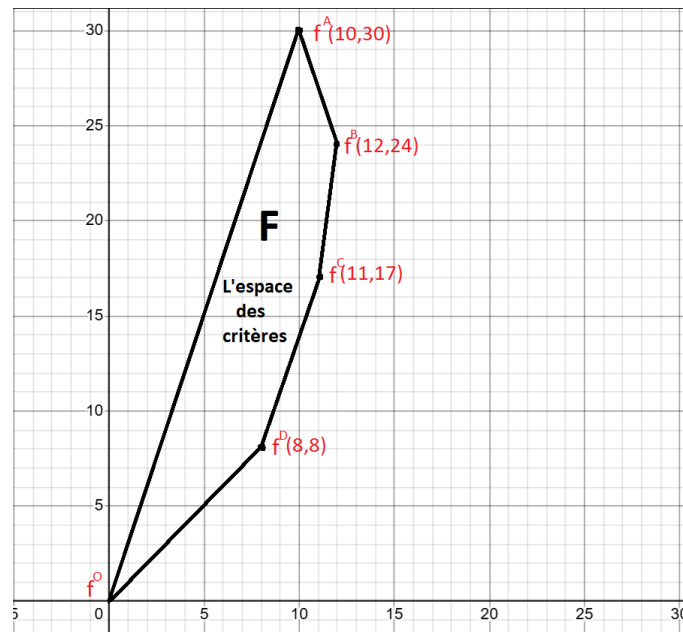


FIGURE 2.7 – Représentation de l'ensemble F des solutions réalisables dans l'espace des critères.

Maintenant on cherche à trouver les points non dominés.

Rappelons la définition d'un point non dominé : étant donné p critères à maximiser, un point $f = (f_1, \dots, f_p) \in F$ est non dominé si et seulement s'il n'existe pas de point $f' \in F$ tel que f' domine f . C'est-à-dire tel que $f'_j \geq f_j (j = 1, \dots, p)$ avec une inégalité stricte au moins. On appelle solution efficace, toute solution $x \in X$ correspondant à un point non dominé.

Dans notre cas, le premier critère est à maximiser alors le second est à minimiser.

Un point $f = (f_1, f_2) \in F$ est non dominé si et seulement s'il n'existe pas de point $f' \in F$ tel que :

$$(f'_1 \geq f_1 \text{ et } f'_2 < f_2) \quad \text{ou} \quad (f'_1 > f_1 \text{ et } f'_2 \leq f_2)$$

Considérons par exemple le point $f^1 = (10, 20)$. L'ensemble Δ^{f^1} dominant f^1 contient clairement des points de F , et en particulier le point f^C . Le point f^1 est donc dominé.

Considérons maintenant le point $f^2 = (6, 6)$, situé sur le segment $[f^O f^D]$. On constate graphiquement que $\Delta^{f^2} \cap F = \emptyset$, ce qui indique que f^2 est non dominé.

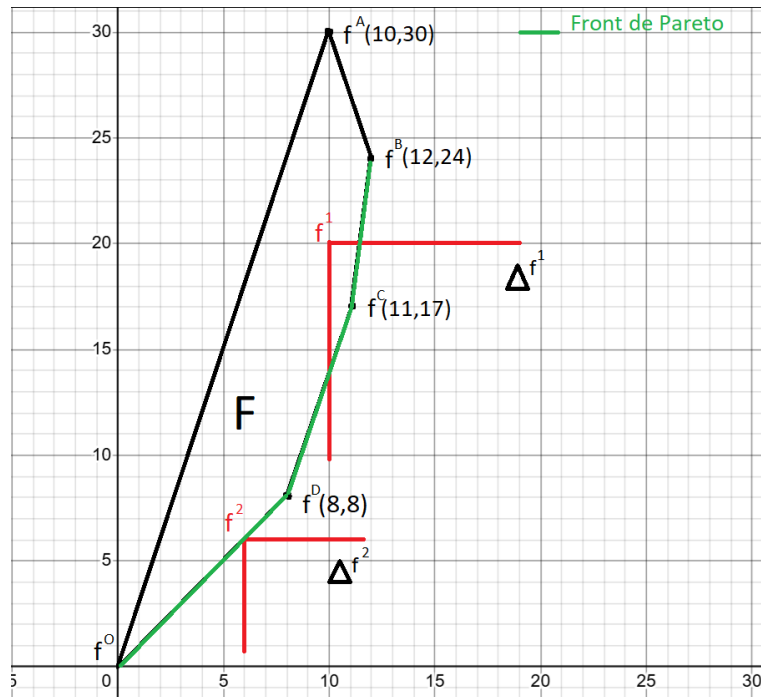


FIGURE 2.8 – Représentation de l'ensemble des points non dominés.

L'ensemble des points non dominés se situe donc sur les segments $[f^B f^C]$, $[f^C f^D]$ et $[f^D f^O]$. L'ensemble des solutions efficaces correspondantes est situé sur les segments $[BC]$, $[CD]$ et $[DO]$. Ces solutions sont caractérisées par :

$$3x_1 + 2x_2 = 30, \quad x_1 \in [6, 8] \quad (\text{segment } BC)$$

$$x_1 = 8, \quad x_2 \in [0, 3] \quad (\text{segment } CD)$$

$$x_2 = 0, \quad x_1 \in [0, 8] \quad (\text{segment } DO)$$

Chaque plan de production correspondant est un candidat potentiellement intéressant si l'on souhaite prendre en compte les deux critères. Le choix d'un plan particulier dépend de l'importance que l'on attribue à chaque critère. En revanche toute solution non efficace peut être écartée.

Chapitre 3

Méthodes d'optimisation

3.1 Introduction

De façon générale, l'optimisation est l'ensemble des méthodes mathématiques et (ou) numériques permettant de résoudre les problèmes de minimisation ou maximisation de fonctions. Un grand nombre de méthodes existent afin de gérer plus de problèmes et situations possibles. L'optimisation sans contrainte consiste à chercher une solution pour un problème donnée (minimum ou maximum) sans imposer des contraintes. L'optimisation avec contrainte, elle, recentre la question autour de la gestion de contraintes, pouvant être exprimées par des fonctions mathématiques simples réduisant directement l'espace des solutions, ou moins directes et plus floues. L'optimisation multi-objectif, s'intéresse à la gestion des compromis entre différents objectifs. Elle est très utilisée dans les domaines de la décision multi-critères afin d'identifier l'ensemble des compromis possibles, appelé front de Pareto.

3.2 Méthodes d'optimisation mono-objectif sans contrainte

3.2.1 Méthode du gradient conjugué linéaire [16]

Considérons le problème de minimisation d'une fonction quadratique $q : \mathbb{R}^n \rightarrow \mathbb{R}$ avec :

$$f(x) = \frac{1}{2}x^T Ax - b^T x \quad (3.1)$$

où $A \in \mathbb{R}^{n \times n}$ est une matrice symétrique définie positive et $b \in \mathbb{R}^n$.

En se référant à la symétrie de A , on obtient :

$$\nabla f(x) = Ax - b \quad \text{et} \quad \nabla^2 f(x) = A.$$

Dans le cas où A est définie positive, la fonction f est strictement convexe et possède un minimum global unique au point x^* tel que :

$$\nabla f(x^*) = 0.$$

En conséquence, la solution du problème de minimisation (3.1) est également solution du problème suivant trouver $x \in \mathbb{R}^n$ satisfaisant,

$$Ax = b \tag{3.2}$$

La méthode du gradient conjugué a été introduite par Hestenes et Stiefel [18] dans les années 1950, comme méthode itérative pour résoudre les systèmes d'équations linéaires $Ax = b$, où $A \in \mathbb{R}^{n \times n}$ est une matrice symétrique définie positive.

Elle peut également être appliquée à des systèmes plus généraux avec A inversible, mais non nécessairement symétrique ou définie positive, après transformation de (3.2) en :

$$A^T Ax = A^T b,$$

où la méthode du gradient conjugué permet de résoudre ce type de problème en au plus n itérations. Cette méthode est souvent appelée « méthode du gradient conjugué linéaire ».

Le nom de la méthode vient du fait qu'elle produit une séquence de directions $\{d_0, \dots, d_{n-1}\}$ telle que les directions sont conjuguées par rapport à la matrice A , c'est-à-dire

$$d_i A d_j = 0 \quad \text{pour tout } i \neq j \tag{3.3}$$

Si l'on note $g_k = \nabla f(x_k)$, l'algorithme de la méthode du gradient conjugué pour les fonctions quadratiques prend la forme suivante :

Cet algorithme consiste à générer une suite itérative $\{x_k\}$ sous la forme :

$$x_{k+1} = x_k + \alpha_k d_k, \tag{3.4}$$

où $d_k \in \mathbb{R}^n$ est une direction de recherche et $\alpha_k \in \mathbb{R}^+$ est le pas de recherche.

À chaque itération k , la direction d_k est obtenue comme une combinaison linéaire du gradient en x_k et de la direction précédente d_{k-1} :

$$d_{k+1} = -g_{k+1} + \beta_k d_k, \tag{3.5}$$

où β_k est une suite de scalaires non nuls, et $g_{k+1} = \nabla f(x_{k+1})$.

Le choix des scalaires β_k assure la propriété de conjugaison (3.3). On en déduit alors :

$$d_{k+1}^T A d_k = 0 \quad (3.6)$$

$$(-g_{k+1} + \beta_k d_k)^T A d_k = 0 \quad (3.7)$$

$$-g_{k+1}^T A d_k + \beta_k d_k^T A d_k = 0 \quad (3.8)$$

$$\beta_k = \frac{g_{k+1}^T A d_k}{d_k^T A d_k}, \quad k = 0, 1, \dots, n-1 \quad (3.9)$$

Dans le cas d'une fonction quadratique f , il est facile de déterminer le pas α_k , comme solution analytique exacte pour le problème de minimisation unidimensionnel :

$$f(x_{k+1}) = \min_{\alpha \in \mathbb{R}} f(x_k + \alpha d_k) \quad (3.10)$$

Ainsi, α_k est une solution optimale de (3.10) si et seulement si α_k vérifie :

$$f'(\alpha_k) = 0.$$

Or

$$f'(\alpha_k) = d_k^T \nabla f(x_{k+1}) = 0$$

c'est-à-dire :

$$d_k^T \nabla f(x_{k+1}) = d_k^T (A x_{k+1} - b) = 0$$

où $d_k = (d_1, \dots, d_n)$ est le vecteur direction, et $\nabla f = \left(\frac{\partial f}{\partial x_i} \right)_{i=1}^n$.

Or, comme $x_{k+1} = x_k + \alpha_k d_k$, on a :

$$d_k^T (A(x_k + \alpha_k d_k) - b) = 0$$

ce qui équivaut à :

$$\alpha_k = - \frac{d_k^T g_k}{d_k^T A d_k}$$

avec $g_k = \nabla f(x_k) = A x_k - b$.

Exemple [16]

Considérons la fonction quadratique suivante :

$$f(x) = \frac{1}{2}x^T Ax - b^T x,$$

avec

$$A = \begin{bmatrix} 4 & 1 \\ 1 & 3 \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \quad x^{(0)} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

La matrice A est symétrique définie positive, donc f admet un unique minimum global.

Itération 0 :

— Gradient initial :

$$g_0 = Ax^{(0)} - b = \begin{bmatrix} 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} -1 \\ -2 \end{bmatrix}$$

— Direction initiale :

$$d_0 = -g_0 = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

— Calcul du pas optimal :

$$\alpha_0 = \frac{g_0^T g_0}{d_0^T A d_0} = \frac{(-1)^2 + (-2)^2}{\begin{bmatrix} 1 & 2 \end{bmatrix} \begin{bmatrix} 6 \\ 7 \end{bmatrix}} = \frac{5}{20} = 0.25$$

— Mise à jour de la solution :

$$x^{(1)} = x^{(0)} + \alpha_0 d_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} + 0.25 \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 0.25 \\ 0.5 \end{bmatrix}$$

Itération 1 :

— Nouveau gradient :

$$g_1 = Ax^{(1)} - b = \begin{bmatrix} 1.5 \\ 2.75 \end{bmatrix} - \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 0.5 \\ 0.75 \end{bmatrix}$$

— Calcul de β_0 :

$$\beta_0 = \frac{g_1^T g_1}{g_0^T g_0} = \frac{0.5^2 + 0.75^2}{5} = \frac{0.8125}{5} = 0.1625$$

— Mise à jour de la direction :

$$d_1 = -g_1 + \beta_0 d_0 = \begin{bmatrix} -0.5 \\ -0.75 \end{bmatrix} + 0.1625 \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} -0.3375 \\ -0.425 \end{bmatrix}$$

— Pas optimal :

$$\alpha_1 = \frac{g_1^T g_1}{d_1^T A d_1} = \frac{0.8125}{0.94375} \approx 0.861$$

— Mise à jour de la solution :

$$x^{(2)} = x^{(1)} + \alpha_1 d_1 \approx \begin{bmatrix} 0.25 \\ 0.5 \end{bmatrix} + 0.861 \begin{bmatrix} -0.3375 \\ -0.425 \end{bmatrix} \approx \begin{bmatrix} -0.041 \\ 0.133 \end{bmatrix}$$

Conclusion :

La méthode converge en au plus n itérations pour une matrice $A \in \mathbb{R}^{n \times n}$. Dans ce cas, la solution exacte est donnée par :

$$x^* = A^{-1}b = \begin{bmatrix} 0.0909 \\ 0.6364 \end{bmatrix}$$

Le point $x^{(2)}$ est très proche de cette solution, confirmant la rapidité de convergence du gradient conjugué.

3.2.2 Méthode de Newton [17]

Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ et (P) le problème d'optimisation sans contraintes :

$$\text{Minimiser } \{f(x) : x \in \mathbb{R}^n\}.$$

La méthode de Newton est attribuée au mathématicien, physicien et astronome anglais Isaac Newton. C'est Raphson qui publiait en 1690 la formule itérative utilisée actuellement. C'est la raison pour laquelle certains auteurs l'appellent méthode de Newton-Raphson. L'algorithme de Newton [20, 21] est la généralisation multi-dimensionnelle de la méthode de Newton-Raphson (voir [28]), appliquée à la recherche des racines de $\nabla f(x)$.

C'est certainement, avec la méthode de la plus forte pente (Cauchy 1847 [17]) l'une des plus anciennes méthodes utilisées pour résoudre le problème (P) .

Le principe de la méthode de Newton [20, 21] consiste à minimiser successivement les approximations du second ordre d'une fonction f , plus précisément si

$$f(x) \simeq f(x^k) + \nabla f(x^k)^T (x - x^k) + \frac{1}{2} (x - x^k)^T H(x^k) (x - x^k) = q(x), \quad \forall x \in V(x^k),$$

$H(x^k)$ étant la matrice hessienne de f au point x^k , alors une condition nécessaire de minimum pour q est que $\nabla q(x) = 0$, où

$$\nabla f(x^k) + H(x^k)(x - x^k) = 0 \tag{3.11}$$

Supposons que H est inversible, alors le successeur de x^k est donné par

$$x^{k+1} = x^k - H^{-1}(x^k)\nabla f(x^k) \quad (3.12)$$

Cette équation donne la forme générale des points générés par l'algorithme de Newton. Supposons que $\nabla f(\bar{x}) = 0$ et que $H(\bar{x})$ est définie positive (\bar{x} un minimum local), alors $H(x^k)$ reste définie positive en tout point voisin de \bar{x} . Ceci assure que le successeur de x^k est bien défini.

Algorithme de Newton

Initialisation : soit ε un paramètre qui détermine le critère d'arrêt. Choisir un point initial x_0 .

Poser $k = 1$ et aller à l'étape principale.

Etape principale : Si $\|\nabla f(x^k)\| < \varepsilon$, stop; sinon, poser $x^{k+1} = x^k - H^{-1}(x^k)\nabla f(x^k)$, remplacer k par $k + 1$ et répéter l'étape principale.

Etude de la convergence

En général la méthode de Newton ne converge pas à cause de la possibilité que $H(x^k)$ soit singulière et donc le point x^{k+1} ne soit pas bien défini et même si $H^{-1}(\bar{x})$ existe, la décroissance de f n'est pas toujours assurée. Cependant ces problèmes peuvent être évités si le point initial est assez voisin du point \bar{x} , tel que $\nabla f(\bar{x}) = 0$ et $H^{-1}(\bar{x})$ existe.

Dans ce cas la méthode de Newton est bien définie et elle converge vers le point \bar{x} .

Théorème. [17] Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ trois fois continûment différentiable. Considérons l'algorithme de Newton défini par l'application $A(x) = x - H^{-1}(x)\nabla f(x)$. Soit \bar{x} tel que $\nabla f(\bar{x}) = 0$ et supposons que $H(\bar{x})^{-1}$ existe. Soit x_1 un point initial assez proche de \bar{x} de sorte que cette proximité implique l'existence de $k_1, k_2 > 0$, avec $k_1 k_2 \|x_1 - \bar{x}\| < 1$ et vérifiant les deux propriétés suivantes :

1. $\|H(x)^{-1}\| \leq k_1$
2. $\|\nabla f(\bar{x}) - \nabla f(x) - H(x)(x - \bar{x})\| \leq k_2 \|\bar{x} - x\|^2$

pour tout x satisfaisant

$$\|x - \bar{x}\| \leq \|x_1 - \bar{x}\|.$$

Alors, l'algorithme converge de façon superlinéaire vers \bar{x} , avec une vitesse de convergence quadratique.

Exemple [17]

Résoudre le problème suivant par la méthode de Newton :

$$\min f(x_1, x_2, x_3, x_4) = (x_1 + 10x_2)^2 + 5(x_3 - x_4)^2 + (x_2 - 2x_3)^4 + 10(x_1 - x_4)^4$$

On pose $g^{(k)} = \nabla f(x^k)$

On utilise comme point de départ $x^{(0)} = [3, -1, 0, 1]^T$. Effectuer trois itérations. Notons que

$$\nabla f(x) = \begin{bmatrix} 2(x_1 + 10x_2) + 40(x_1 - x_4)^3 \\ 20(x_1 + 10x_2) + 4(x_2 - 2x_3)^3 \\ 10(x_3 - x_4) - 8(x_2 - 2x_3)^3 \\ -10(x_3 - x_4) - 40(x_1 - x_4)^3 \end{bmatrix},$$

$$f(x^{(0)}) = 215.$$

La matrice hessienne $H(x)$ est donnée par :

$$H(x) = \begin{bmatrix} 2 + 120(x_1 - x_4)^2 & 20 & 0 & -120(x_1 - x_4)^2 \\ 20 & 200 + 12(x_2 - 2x_3)^2 & -24(x_2 - 2x_3)^2 & 0 \\ 0 & -24(x_2 - 2x_3)^2 & 10 + 48(x_2 - 2x_3)^2 & -10 \\ -120(x_1 - x_4)^2 & 0 & -10 & 10 + 120(x_1 - x_4)^2 \end{bmatrix}.$$

Itération 1 :

$$g^{(0)} = [306, -144, -2, -310]^T,$$

$$H(x^{(0)}) = \begin{bmatrix} 482 & 20 & 0 & -480 \\ 20 & 212 & -24 & 0 \\ 0 & -24 & 58 & -10 \\ -480 & 0 & -10 & 490 \end{bmatrix},$$

$$H(x^{(0)})^{-1} = \begin{bmatrix} 0.1126 & 0.0089 & 0.0154 & 0.1106 \\ -0.0089 & 0.0057 & 0.0008 & -0.0087 \\ 0.0154 & 0.0008 & 0.0203 & 0.0155 \\ 0.1106 & -0.0087 & 0.0155 & 0.1107 \end{bmatrix},$$

$$H(x^{(0)})^{-1}g^{(0)} = [1.4127, -0.8413, -0.2540, 0.7460]^T.$$

Par conséquent,

$$x^{(1)} = x^{(0)} - H(x^{(0)})^{-1}g^{(0)} = [1.5873, -0.1587, 0.2540, 0.2540]^T,$$

$$f(x^{(1)}) = 31.8.$$

Itération 2 :

$$g^{(1)} = [94.81, -1.179, 2.371, -94.81]^T,$$

$$F(x^{(1)}) = \begin{bmatrix} 215.3 & 20 & 0 & -213.3 \\ 20 & 205.3 & -10.67 & 0 \\ 0 & -10.67 & 31.34 & -10 \\ -213.3 & 0 & -10 & 223.3 \end{bmatrix},$$

$$H(x^{(1)})^{-1}g^{(1)} = [0.5291, -0.0529, 0.0846, 0.0846]^T,$$

Par conséquent,

$$x^{(2)} = x^{(1)} - H(x^{(1)})^{-1}g^{(1)} = [1.0582, -0.1058, 0.1694, 0.1694]^T,$$

$$f(x^{(2)}) = 6.28.$$

Itération 3 :

$$g^{(2)} = [28.09, -0.3475, 0.7031, -28.08]^T,$$

$$H(x^{(2)}) = \begin{bmatrix} 96.80 & 20 & 0 & -94.80 \\ 20 & 202.4 & -4.744 & 0 \\ 0 & -4.744 & 19.49 & -10 \\ -94.80 & 0 & -10 & 104.80 \end{bmatrix},$$

$$x^{(3)} = [0.7037, -0.0704, 0.1121, 0.1111]^T, \quad f(x^{(3)}) = 1.24.$$

3.3 Méthodes d'optimisation mono-objectif avec contrainte

3.3.1 Méthode du gradient projetée [4]

Dans l'optimisation sans contrainte, l'algorithme du gradient est donnée comme suit :

$$\begin{cases} x^{(0)} \in \mathbb{R}^n \\ x^{(k+1)} = x^{(k)} + p^{(k)}d^{(k)}, \quad d^{(k)} \in \mathbb{R}^n \setminus \{0\}, \quad p^{(k)} \in \mathbb{R}_+^* \end{cases}$$

Où $p^{(k)}$ et $d^{(k)}$ sont choisit de telle sorte que :

$$f(x^{(k+1)}) \leq f(x^{(k)})$$

Dans le cas avec contraintes, on ne peut pas être sûr que $x^{(k)}$ reste sur l'ensemble des contraintes C . Il est donc nécessaire de se ramener sur C par une projection. L'opérateur associé étant noté Π_C , où C est convexe, fermé et non vide :

$$\Pi_C : \mathbb{R}^n \rightarrow C \quad \text{avec} \quad x \mapsto \Pi_C(x).$$

On obtient ainsi l'algorithme du gradient projeté, qui est identique à celui du gradient à la projection près :

Initialisation :

$$k = 0, \quad \text{choix de } x^{(0)} \text{ et } \epsilon > 0.$$

Itération k : Tant que le critère d'arrêt n'est pas satisfait :

$$\hat{x}^{(k+1)} = x^{(k)} - p^{(k)} \nabla f(x^{(k)}),$$

$$x^{(k+1)} = \Pi_C \hat{x}^{(k+1)},$$

$$k = k + 1.$$

Fin tant que.

Théorème. [4] Soit f une fonction de classe C^1 définie de \mathbb{R}^n dans \mathbb{R} . On suppose que f est elliptique et de dérivée lipschitzienne.

Alors, si on choisit le pas $p^{(k)}$ dans un intervalle $[\alpha_1, \alpha_2]$ tel que

$$0 < \alpha_1 < \alpha_2 < \frac{2}{\beta M^2},$$

où β et M sont respectivement les constantes d'ellipticité et d'inégalité lipschitzienne, avec :

$$(\nabla f(x) - \nabla f(y))^T (x - y) \geq \beta \|x - y\|^2$$

et

$$\|f(x) - f(y)\| \leq M \|x - y\|,$$

la suite $(x_k^{(k)})$ définie par la méthode du gradient projeté converge vers la solution du problème.

Exemple [15]

On veut résoudre le problème suivant par la méthode du gradient projeté :

$$\min_{(x,y) \in \mathbb{R}^2} \frac{1}{2}x^2 + \frac{7}{2}y^2$$

$$\text{sous contrainte : } -x + y = 1$$

Le domaine des contraintes est donné par :

$$X = \{(x, y) \in \mathbb{R}^2 / -x + y = 1\}$$

qui est un ensemble convexe fermé.

Nous savons que le problème sans contrainte admet un minimum global en $(0, 0)$.

Cependant, $(0, 0)$ ne satisfait pas la contrainte, et donc n'est pas un point admissible pour le problème avec contrainte.

Afin de mettre en œuvre l'algorithme du gradient projeté, il faut choisir les méthodes de calcul des pas p_k et α_k :

- On choisit une méthode de gradient pour le calcul du pas p_k .
- En première approche, on choisit un pas fixe $\alpha_k = 1$, ce qui implique :

$$x^{(k+1)} = \Pi_C \left(x^{(k)} - p_k \nabla f(x^{(k)}) \right)$$

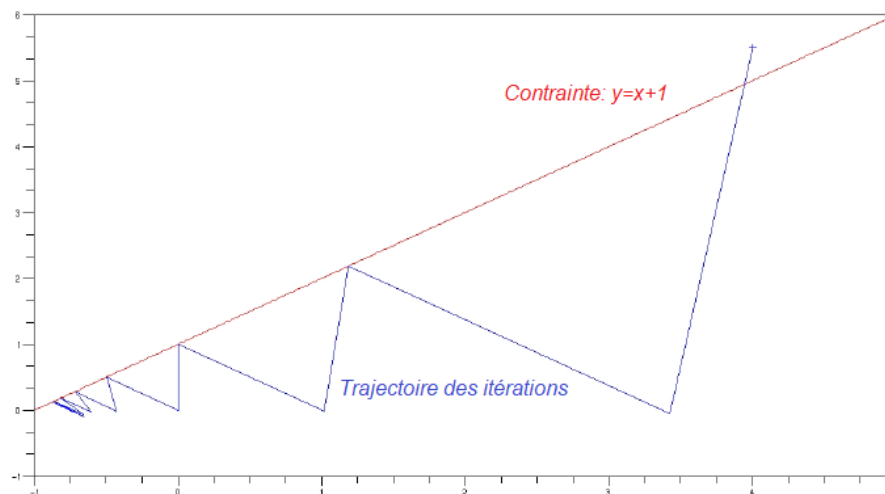


FIGURE 3.1 – Itérations successives (avant et après projection) de l'algorithme du gradient projeté.

Le comportement numérique de la méthode de gradient projeté ainsi définie, est illustré par la figure (3.1) et le tableau suivant des itérations. On observe une convergence de la suite

d'itérés générés à partir du point $x_0=(4, 5.5)$ vers le point $(0.875, 0.125)$.

k	x^k	$x^k - p^k \nabla f(x_k)$	$\ d^k\ = \ \hat{x}^{(k+1)} - x^k\ $
0	[4 5.5]	[3.4232925 - 0.0508095]	4.3472097
1	[1.1862415 2.1862415]	[1.0159064 - 0.0112495]	1.6743058
2	[0.0023285 1.0023285]	[0.0019958 - 9.462e-08]	0.7089885
3	[-0.4990021 0.5009979]	[-0.4264826 - 0.0086691]	0.3091099
4	[-0.7175758 0.2824242]	[-0.6037028 - 0.0313035]	0.1413186
5	[-0.8175032 0.1824968]	[-0.6618980 0.0605186]	0.0618727
6	[-0.8612538 0.1387462]	[-0.6636631 0.0804770]	0.0178399
7	[-0.8738685 0.1261315]	[-0.6570769 0.0920958]	0.0015879
8	[-0.8749913 0.1250087]	[-0.6562565 - 0.0937435]	0.0000123
9	[-0.875 0.125]	[-0.65625 - 0.09375]	7.301e-10

Ce tableau donne les itérations de la méthode du gradient projeté.

On note :

$$\hat{x}^{(k+1)} = x^{(k)} - p^{(k)} \nabla f(x^{(k)}),$$

Le critère d'arrêt $\|d^k\| < \epsilon$ est satisfait en 9 itérations pour une précision $\epsilon = 10^{-5}$

3.3.2 Méthode du Newton projetée [4]

Cas de contrainte de type $x \geq 0$

Soit le problème suivant :

$$(P) \quad \begin{aligned} & \min f(x) \\ & \text{sous contrainte : } x \geq 0 \end{aligned}$$

On suppose que f est de classe C^2 sur \mathbb{R}^n . Si $H^{(k)}$ désigne la matrice hessienne de f en $x^{(k)}$, une itération de la méthode de Newton s'écrit :

$$x^{(k+1)} = x^{(k)} - [H^{(k)}]^{-1} \nabla f(x^{(k)}).$$

On peut affiner cette méthode en introduisant un pas $\beta(k) > 0$, ce qui donne :

$$x^{(k+1)} = x^{(k)} - \beta(k)[H^{(k)}]^{-1} \nabla f(x^{(k)}).$$

Si l'on projette sur l'ensemble des contraintes, on obtient la version projetée :

$$x^{(k+1)} = \Pi_C \left(x^{(k)} - \beta(k)[H^{(k)}]^{-1} \nabla f(x^{(k)}) \right).$$

Où Π_C est l'opérateur de projection sur l'ensemble des contraintes C .

On a le théorème de convergence suivant :

Théorème. [4] On suppose que f est convexe et de classe C^2 et que le problème (P) a une solution unique x^* qui vérifie $\frac{\partial f}{\partial x_i}(x^*) > 0$, $\forall i \in I_a(x^*)$, ($I_a(x^*)$ est l'ensemble des indices actifs). On suppose de plus que l'on peut trouver m_1 et m_2 deux réels strictement positifs tels que :

$$m_1 \|z\|^2 \leq z \nabla^2 f(x) z \leq m_2 \|z\|^2$$

Dans chacun des cas suivants :

- pour tout $z \in \{x \mid f(x) \leq f(x^{(0)})\}$, d'une part ;
- pour x dans une boule centrée en x^* et $z \neq 0$ tel que $z_i = 0$ pour tout $i \in I_a(x^*)$, d'autre part.

Alors la suite $(x^{(k)})$ engendrée par l'algorithme converge vers x^* et le taux de convergence est superlinéaire.

On peut alors généraliser l'algorithme précédent au problème à contraintes bornées.

Cas de contraintes de bornes

On donne le problème suivant :

$$(P') \quad \begin{array}{l} \min f(x) \\ \text{sous contrainte : } a \leq x \leq b \end{array}$$

Nous appliquons les itérés de la méthode de Newton projetée comme dans le cas du problème (P), en utilisant la projection suivante :

$$[x^{(k+1)}] = x^{(k)} - \alpha^{(k)} [H^{(k)}]^{-1} \nabla f(x^{(k)})]^{Proj}$$

où pour tout $z \in \mathbb{R}^n$, $z^{Proj} = \Pi_C(z)$

$$[z]_i^{Proj} = \begin{cases} b_i & \text{si } b_i \leq z_i \\ z_i & \text{si } a_i \leq z_i \leq b_i \\ a_i & \text{si } a_i \leq z_i. \end{cases}$$

Exemple [4]

On cherche à minimiser la fonction suivante :

$$f(x_1, x_2) = (x_1 - 4)^2 + (x_2 - 2)^2$$

sous la contrainte :

$$x_1 \geq 1, \quad x_2 \geq 0$$

À chaque itération, on applique :

$$x^{(k+1)} = \Pi_{\mathcal{C}} \left(x^{(k)} - H^{-1} \nabla f(x^{(k)}) \right)$$

où $\mathcal{C} = \{x \in \mathbb{R}^2 \mid x_1 \geq 1, x_2 \geq 0\}$, et $\Pi_{\mathcal{C}}$ est la projection sur l'ensemble admissible.

Gradient :

$$\nabla f(x) = \begin{bmatrix} 2(x_1 - 4) \\ 2(x_2 - 2) \end{bmatrix}$$

Hessienne (constante) :

$$H = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}, \quad H^{-1} = \frac{1}{2}I$$

Point initial :

Prenons :

$$x^{(0)} = \begin{bmatrix} 0 \\ 3 \end{bmatrix}$$

Gradient :

$$\nabla f(x^{(0)}) = \begin{bmatrix} 2(0 - 4) \\ 2(3 - 2) \end{bmatrix} = \begin{bmatrix} -8 \\ 2 \end{bmatrix}$$

Pas de Newton :

$$d^{(0)} = -H^{-1} \nabla f(x^{(0)}) = -\frac{1}{2} \begin{bmatrix} -8 \\ 2 \end{bmatrix} = \begin{bmatrix} 4 \\ -1 \end{bmatrix}$$

Mise à jour avant projection :

$$x^{(0)} + d^{(0)} = \begin{bmatrix} 0 \\ 3 \end{bmatrix} + \begin{bmatrix} 4 \\ -1 \end{bmatrix} = \begin{bmatrix} 4 \\ 2 \end{bmatrix}$$

Projection sur \mathcal{C} :

$$x^{(1)} = \Pi_{\mathcal{C}}(x^{(0)} + d^{(0)}) = \begin{bmatrix} \max(1, 4) \\ \max(0, 2) \end{bmatrix} = \begin{bmatrix} 4 \\ 2 \end{bmatrix}$$

Vérification :

$$\nabla f(x^{(1)}) = \begin{bmatrix} 2(4 - 4) \\ 2(2 - 2) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \text{Donc convergence atteinte.}$$

Conclusion :

Le minimum est atteint en :

$$x^* = (4, 2), \quad f(x^*) = 0$$

Le point optimal respecte bien les contraintes $x_1 \geq 1, x_2 \geq 0$.

3.4 Méthodes d'optimisation multi-objectif

3.4.1 La méthode de pondération des fonctions objectif [8]

Cette approche de la résolution d'un problème d'optimisation multi-objectif est la plus évidente. Le but ici est de revenir à un problème d'optimisation mono-objectif, dont il existe de nombreuses méthodes de résolution. La manière la plus simple de procéder consiste à prendre chacune des fonctions objectif, à leur appliquer un coefficient de pondération et à faire la somme des fonctions objectif pondérées. On obtient alors une nouvelle fonction objectif.

présentation de la méthode

On part du problème suivant :

$$(PMO) \quad \begin{aligned} \min_{x \in X} f(x) &= [f_1(x), f_2(x), \dots, f_k(x)] \\ \text{sous contraintes} \quad &\begin{cases} h_j(x) = 0, & \forall j \in P = \{1, \dots, p\}, \\ g_i(x) \leq 0, & \forall i \in M = \{1, \dots, m\}. \end{cases} \end{aligned}$$

Le (PMO) se transforme en un autre problème :

$$\begin{aligned} \min_{x \in X} f(x) &= \sum_{i=1}^k w_i f_i(x) \\ \text{sous contraintes} \quad &\begin{cases} h_j(x) = 0, & \forall j \in P = \{1, \dots, p\}, \\ g_i(x) \leq 0, & \forall i \in M = \{1, \dots, m\}. \end{cases} \end{aligned}$$

Soit $x \in \mathbb{R}^n$, $g(x) \in \mathbb{R}^m$ et $h(x) \in \mathbb{R}^p$.

Fréquemment, les coefficients de pondération respectent la relation suivante :

$$w_i \geq 0 \quad \text{pour tout } i \in \{1, \dots, k\} \quad \text{et} \quad \sum_{i=1}^k w_i = 1.$$

On peut représenter graphiquement le fonctionnement de la méthode sur un problème à deux objectifs. Le problème est le suivant :

$$\begin{aligned} &\min f_1(x) \\ &\min f_2(x) \\ \text{sous contraintes} \quad &\begin{cases} h_j(x) = 0, & \text{(contraintes d'égalité)}, \\ g_i(x) \leq 0, & \text{(contraintes d'inégalité)}. \end{cases} \end{aligned}$$

Notre nouvelle fonction objectif sera donnée comme suit :

$$f(x) = w_1 f_1(x) + w_2 f_2(x) \tag{3.13}$$

Ceci est l'expression d'une droite dans le plan f_1, f_2 .

En effet, si l'on cherche à minimiser $f(x)$, on cherche en fait une constante C de l'équation de droite suivante la plus petite possible :

$$f_2(x) = -\frac{w_1}{w_2} \cdot f_1(x) + C \quad (3.14)$$

Cette équation de droite correspond à la courbe des isovaleurs de la fonction objectif équivalente.

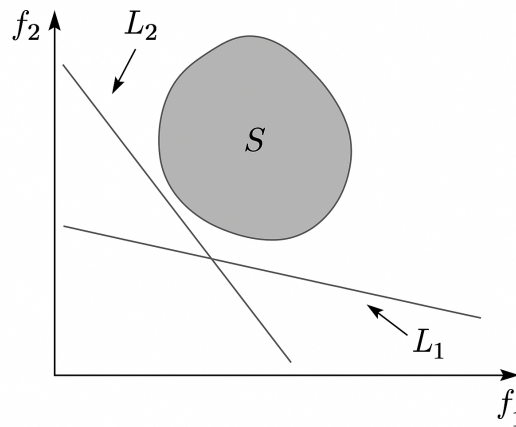


FIGURE 3.2 – La méthode de pondération des fonctions objectif.

Sur la figure (3.2), l'ensemble S correspond à l'ensemble des valeurs du couple (f_1, f_2) respectant les contraintes définies par $g(x)$ et $h(x)$. Les droites L_1 et L_2 correspondent à deux couples de coefficients de pondération (w_1, w_2) différents.

Cette méthode consiste à « faire tangenter » la droite L_1 et la droite L_2 avec l'ensemble S . Le point de tangence est alors la solution recherchée.

Si l'on répète ce processus pour plusieurs valeurs des coefficients de pondération, les différentes solutions trouvées forment la surface de compromis.

Cette méthode n'est applicable qu'à des ensembles S convexes. Dans le cas contraire, elle ne permet pas de trouver la totalité de la surface de compromis.

Exemple (Le problème de Shaffer's F2) [8]

Considérons le problème suivant :

$$\min f_1(x) = x^2$$

$$\min f_2(x) = (x - 2)^2$$

$$\text{sous contrainte : } x \in [0, 2]$$

C'est un problème test appelé Schaffer's F2. La représentation de l'ensemble des solutions de ce problème est différente de celle qui nous a servi à illustrer les différentes définitions de l'optimisation multiobjectif. En effet, si l'on cherche à exprimer f_2 en fonction de f_1 , on obtient une équation d'ellipse :

$$\begin{cases} f_1(x) = x^2 \\ f_2(x) = (x - 2)^2 = x^2 - 4x + 4 = f_1(x) + 4 - 4x \end{cases}$$

On obtient finalement :

$$(f_2(x) - f_1(x) - 4)^2 = (4x)^2$$

Ce qui donne :

$$f_1^2 + f_2^2 - 2f_1f_2 - 8f_1 - 8f_2 + 16 = 0$$

De plus, comme on le verra lors de la résolution du problème d'optimisation, la surface de compromis (qui correspond normalement à une surface) est ici une portion de courbe, et l'ensemble des couples (f_1, f_2) est une courbe (une ellipse dans cet exemple).

La nouvelle fonction objectif s'écrit :

$$f(x) = w_1 \cdot f_1(x) + w_2 \cdot f_2(x) \quad (3.15)$$

Les coefficients de pondération w_1 et w_2 respectent les conditions suivantes :

- $w_1, w_2 \geq 0$
- $w_1 + w_2 = 1$

On cherche le minimum de la fonction objectif $f(x)$, il vérifie les conditions nécessaires de l'existence d'un minimum :

$$\frac{df(x)}{dx} = 0 \quad \text{et} \quad \frac{d^2f(x)}{dx^2} > 0$$

On cherche les points de $f(x)$ qui vérifient :

$$\frac{df(x)}{dx} = 2x(w_1 + w_2) - 4w_2 = 0$$

et

$$\frac{d^2f(x)}{dx^2} = 2(w_1 + w_2) > 0 \quad (\text{puisque l'on a choisi } w_1 + w_2 = 1).$$

Donc le minimum est :

$$x^* = \frac{2w_2}{w_1 + w_2} = 2w_2$$

On va maintenant calculer quatre points de la surface de compromis. On choisit 4 valeurs de w_1 variant de 0,2 à 0,8 par un pas de 0,2. On en déduit les valeurs de w_2 , puis on calcule x^* et $f_1(x^*)$ et $f_2(x^*)$.

w_1	0,2	0,4	0,6	0,8
w_2	0,8	0,6	0,4	0,2
x^*	1,6	1,2	0,8	0,4
$f_1(x^*)$	2,56	1,44	0,64	0,16
$f_2(x^*)$	0,16	0,64	1,44	2,56

On trace maintenant ses points dans le plan f_1 et f_2

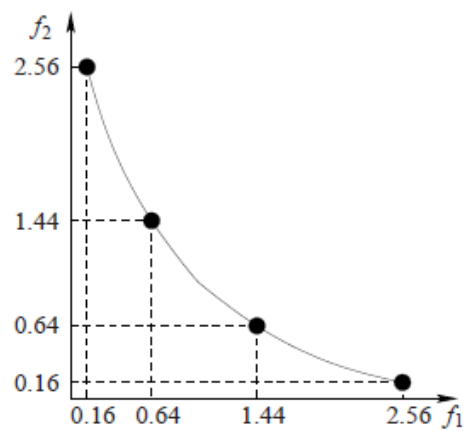


FIGURE 3.3 – La surface de compromis obtenue pour le problème F2 de Schaffer.

On remarque que, dans cet exemple, on trouve une surface de compromis typique dans les problèmes de minimisation sur tous les objectifs. A la figure (3.4), on a représenté l'ensemble des valeurs réalisables du problème test, avec et sans la contrainte $x \in [0, 2]$.

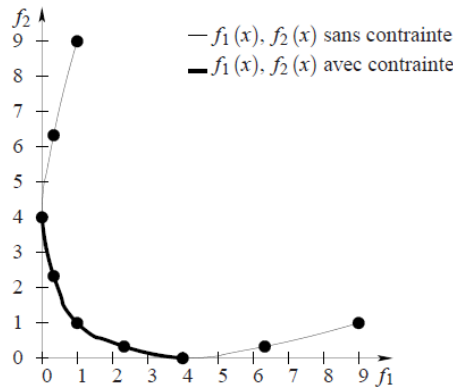


FIGURE 3.4 – La surface de compromis totale.

3.4.2 Méthode de ELECTRE I [8, 19]

Cette méthode permet de choisir la meilleure action suivant un groupe de critères. Elle a été définie par Bernard Roy en 1968.

La manière d'établir le surclassement d'une action par rapport à une autre repose sur :

Une condition de Concordance : on dit que le critère C est un critère concordant avec l'hypothèse de surclassement si l'action a_i est au moins aussi bonne que l'action a_k selon le critère C , c'est-à-dire :

$$C(a_i) \geq C(a_k)$$

Une condition de Non-discordance : la condition de non-discordance permet de refuser une hypothèse de surclassement obtenue après l'application de la relation de concordance lorsqu'il existe une opposition trop forte sur au moins un critère.

La méthode Electre I se présente comme suit :

Étape 01 :

On attribue à chaque critère j , un poids p_j d'autant plus grand que le critère est important.

Étape 02 :

Calculer les indices de concordance à chaque couple d'action (a, b) , on associe l'indice de

concordance suivant :

$$C(a, b) = \frac{\sum_{g_j(a) \geq g_j(b)} p_j}{p} \quad \text{avec} \quad p = \sum_{j=1}^n p_j$$

Étape 03 :

Calculer les indices de discordance à chaque couple d'action (a,b). On associe l'indice de discordance suivant :

$$D(a, b) = \begin{cases} 0 & \text{si } g_j(a) \geq g_j(b) \\ \frac{1}{\delta} \max[g_j(b) - g_j(a)] & \text{sinon} \end{cases} \quad (3.16)$$

Avec : $\delta = \max[g_j(C) - g_j(D)]$

Étape 04 :

Nous avons maintenant toutes les informations nécessaires pour extraire l'action ou les actions les meilleures.

Pour cela, il faut juger les relations de surclassement. Pour qu'une relation de surclassement soit jugée comme fiable, il faut que l'on ait :

$$C(a, b) \geq \tilde{c} \quad \text{et} \quad D(a, b) \leq \tilde{d} \quad \implies \quad a_i S a_k \quad (3.17)$$

où :

- \tilde{c} : seuil de concordance, typiquement $\simeq 0,7$. C'est le seuil au-delà duquel l'hypothèse de départ « a_i surclasse a_k » est considérée comme valable.
- \tilde{d} : seuil de discordance, typiquement $\simeq 0,3$. C'est le seuil au-delà duquel l'hypothèse de départ « a_i surclasse a_k » est rejetée.

Étape 05 :

Exploiter les relations de surclassement :

Cette étape consiste à déterminer le sous-ensemble d'action N appelé noyau tel que toute action qui n'est pas dans N est surclassée par au moins une action de N et les actions de N sont incomparables entre elles.

Exemple [19]

L'exemple traite du choix d'un projet, parmi 6 projets concurrents pour la réalisation d'une raffinerie. Chaque projet est évalué sur la base de 5 critères environnementaux :

- C1 : Nuisance sonore.

- C2 : Séparation du territoire.
- C3 : Pollution de l'air.
- C4 : Impact sur l'aménagement du territoire.
- C5 : Impact sur les activités récréatives.

L'importance de chaque critère dans la prise de décision est traduite par un poids p_j tel que :

Critères	C1	C2	C3	C4	C5
Poids p_j	3	2	3	1	1

TABLE 3.1 – Poids des critères.

Chaque projet est évalué en fonction des critères retenus à l'aide d'une échelle qualitative et des scores. Plus le score est élevé, plus les impacts du projet sur l'environnement sont moindres. Le tableau de performance est donné par la figure 3.2 :

Projets	C1	C2	C3	C4	C5
P1	10	20	5	10	16
P2	0	5	5	16	10
P3	0	10	0	16	7
P4	20	5	10	10	13
P5	20	10	15	10	13
P6	20	10	20	13	13

TABLE 3.2 – Tableau de performances.

La problématique à résoudre est de choisir le sous-ensemble de projets avec le moins d'impacts sur l'environnement. On propose d'utiliser ELECTRE I. Nous présentons un exemple de calcul de l'indice de concordance :

$$C(P_1, P_2) = \frac{3 + 2 + 3 + 0 + 1}{10} = 0.9 \quad , \quad C(P_1, P_3) = \frac{3 + 2 + 3 + 0 + 1}{10} = 0.9$$

$$C(P_2, P_1) = \frac{0 + 0 + 3 + 0 + 1}{10} = 0.4 \quad , \quad C(P_3, P_1) = \frac{0 + 0 + 0 + 1 + 0}{10} = 0.1$$

La matrice des indices de concordance est donnée par :

	P1	P2	P3	P4	P5	P6
P1	-	0.9	0.9	0.7	0.4	0.3
P2	0.4	-	0.8	0.6	0.3	0.1
P3	0.1	0.3	-	0.7	0.5	0.3
P4	0.7	0.9	0.9	-	1.0	0.4
P5	0.4	0.3	0.4	0.5	-	0.4
P6	0.3	0.1	0.3	0.4	0.4	-

TABLE 3.3 – Matrice de concordance.

L'indice de disconcordance est calculé pour une valeur de $\delta = 20 - 0 = 20$:

$$D(R_1, P_2) = \frac{6}{20} = 0.30 \quad , \quad D(R_1, P_3) = \frac{6}{20} = 0.30$$

$$D(P_2, R_1) = \frac{15}{20} = 0.75 \quad , \quad D(P_3, R_1) = \frac{10}{20} = 0.50$$

La matrice de disconcordance est obtenue comme suit :

Projets	P1	P2	P3	P4	P5	P6
P1	-	0.30	0.30	0.50	0.50	0.75
P2	0.75	-	0.25	1.00	1.00	1.00
P3	0.50	0.25	-	1.00	1.00	1.00
P4	0.75	0.30	0.30	-	0.25	0.50
P5	0.50	0.30	0.30	0.00	-	0.25
P6	0.50	0.15	0.15	0.00	0.00	-

TABLE 3.4 – Matrice de disconcordance.

L'intérêt de la méthode ELECTRE I est d'isoler un sous-ensemble de solutions, dans notre cas identifier les projets avec le moins d'impacts sur l'environnement. En considérant $\tilde{c} = 1$ et $\tilde{d} = 0$; P_1, P_2, P_3 et P_6 sont incomparables, par contre $P_6 \succsim P_4, P_6 \succsim P_5$ et $P_5 \succsim P_4$. On obtient le graphe de sur-classement suivant.

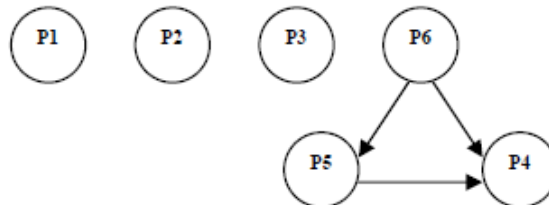


FIGURE 3.5 – Graphe du surclassement.

Chapitre 4

Application en industrie

4.1 Introduction

L'optimisation des conditions de coupe a pour but d'améliorer à la fois la qualité du travail et le coût dans un procédé d'usinage. Pour cela, on suit généralement trois étapes : comprendre et modéliser le problème, trouver une ou plusieurs solutions, puis choisir la plus adaptée. Dans ce travail, nous nous intéressons à l'optimisation des conditions de coupe pour une opération de tournage. Deux modèles sont proposés pour montrer l'importance de cette optimisation, d'un point de vue économique et technologique. La résolution du problème est illustrée à travers deux exemples : le premier basé uniquement sur un critère économique, et le second combinant des aspects techniques et économiques.

4.2 Définition d'une coupe [2]

En usinage, la coupe désigne le processus par lequel une quantité de matière est enlevée d'une pièce brute à l'aide d'un outil coupant, dans le but d'obtenir une géométrie, des dimensions et un état de surface conformes. Cette action résulte du mouvement relatif entre l'outil et la pièce, généralement sous l'effet d'une force mécanique.

Dans une opération de tournage, la pièce est en rotation, tandis que l'outil avance suivant une trajectoire définie.

4.3 Les paramètres d'une coupe [2]

Les paramètres de coupe sont, d'un côté, des valeurs qui décrivent comment l'outil et la pièce bougent pendant l'usinage (on appelle ça les paramètres de coupe liés au mouvement), et de l'autre côté, des valeurs qui concernent la quantité de matière à enlever et les dimensions

de coupe (ce sont les paramètres liés à la forme). Les conditions de coupe dépendent aussi d'autres éléments comme le matériau de la pièce, le matériau et la forme de l'outil, les liquides utilisés pour couper, l'état de la machine, etc.

4.3.1 Mouvement de coupe

Le mouvement de coupe est un mouvement relatif principal entre l'outil et la pièce. Il est caractérisé par **la vitesse de coupe** V_c . Dans le cas du mouvement circulaire, la vitesse de coupe est une fonction de la vitesse de rotation n et de diamètre D de l'élément en rotation. Donc V_c est donnée en mètres par minute par :

$$V_c = \frac{\pi D n}{1000} \quad [m/min]$$

Etant donné que la vitesse de coupe V_c est définie en m/min (mètre/minute) , le diamètre D en mm et la vitesse de rotation n en tr/min (tour/minute) , le chiffre « 1000 » tient place de facteur de conversion d'unité qu'est un nombre utilisé pour transformer une grandeur exprimée dans une unité vers une autre unité équivalente, sans en modifier la valeur physique.

4.3.2 Mouvement d'avance

Au mouvement de coupe, vient s'ajouter un autre mouvement relatif entre l'outil et la pièce, le mouvement d'avance, nécessaire à la génération de la surface de la pièce. Il est caractérisé par l'avance f qui est une vitesse instantanée du mouvement d'avance du point considéré de l'arête de coupe par rapport à la pièce. Elle est exprimée soit en mm/min (millimètre/minute) soit en $mm/tour$. On trouve ainsi que la vitesse d'avance est donnée par la relation suivante :

$$V_f = f.n$$

Où n : la vitesse de rotation.

4.3.3 Le temps de production

Lors d'une opération de tournage, le temps de production est réparti entre trois types d'activités : le montage/démontage de la pièce, la manipulation de l'outil, et la préparation de la machine. Le temps lié au montage et démontage ne dépend pas directement des conditions de coupe ; c'est un temps fixe intégré au temps d'utilisation de la machine. En revanche, les autres tâches dépendent du choix des paramètres de coupe et influencent le temps total de production.

Le temps total de production est la somme des temps suivants :

- Le temps de préparation T_s : il inclut le réglage des conditions de coupe et le montage/démontage de la pièce,
- Le temps de retour de l'outil T_i : il correspond à la manipulation de l'outil.
- Le temps machine : c'est le temps nécessaire à l'usinage, donné par le rapport du volume de matière enlevé V sur le débit de coupe Q (c'est la quantité de matière enlevée par minute lors de l'usinage), soit $\frac{V}{Q}$.
- Le temps de changement d'outil T_c : il correspond au remplacement de l'outil usé.

Ainsi, le temps total de production P_r est donné par :

$$P_r = T_s + T_i + \frac{V}{Q} + T_c \quad (4.1)$$

Le second terme de cette équation, $\frac{V}{Q} + T_c$, représente le temps nécessaire pour enlever la matière et celui consacré aux changements d'outil.

4.3.4 La consommation en outil

C'est le tau d'utilisation de l'outil. Mathématiquement, ce paramètre peut être présenté par le rapport de la durée utilisée d'un outil sur sa durée de vie totale. Sachant que le temps nécessaire pour l'enlèvement d'un volume de matière $V(mm^3)$ est donné par le rapport de ce volume sur le débit de coupe Q , la consommation en outil est formulée comme suit :

$$Ut = \frac{V}{QT} 100\%$$

A partir du modèle de Taylor développé en suite par Gilbert, la durée de vie d'une arête de coupe est donnée par l'équation suivante :

$$T = \frac{k_T}{V_c^{\alpha_1} f^{\alpha_2} a_p^{\alpha_3}}$$

où :

$$Q = 1000.V_c f a_p$$

et

$$V = \pi DLS$$

et a_p est la profondeur de coupe.

4.3.5 Etat de surface

Les paramètres susceptibles de produire un état de surface, selon les critères d'appréciation R_a et R_t révélateurs du degré d'usure de l'arête de coupe des outils, sont le rayon de pointe

(r) de l'outil de coupe et l'avance par tour.

La qualité des surfaces obtenue est évaluée à partir de la rugosité R_a (c'est un paramètre quantitatif couramment utilisé pour évaluer la qualité de surface d'une pièce usinée) donnée par la relation empirique suivante :

$$R_a = k.V_c^{x_1}.f^{x_2}.a_p^{x_3} \quad (4.2)$$

Où : k est le coefficient de la rugosité, x_1, x_2, x_3 sont des exposants empiriques.

4.4 Cas d'application [2]

Sur un tour à commande numérique, nous avons besoin de réaliser une opération de chariotage. La machine a une puissance maximale admissible de 5 kW. Une force maximale de coupe de 1100 N peut être supportée par le système de fixation de la machine. Les paramètres d'usinage utilisés dans l'application numérique sont mentionnés dans le tableau suivant :

Paramètre	Valeur	Paramètre	Valeur	Paramètre	Valeur
V (mm ³)	286513.25	k_T	396135004.71	k	14785
T_s (min)	1.5	α_1	4	x_1	-1.52
T_c (min)	0.5	α_2	1.16	x_2	1.004
T_i (min)	0.13	α_3	1.4	x_3	0.25
C_0 (\$/min)	0.1	C_1 (\$)	0.5	$R_{a_{\max}}$ (μm)	5

TABLE 4.1 – Tableau des paramètres et leurs valeurs.

Où : $\alpha_1, \alpha_2, \alpha_3$ sont les exposants empiriques de la durée de vie de l'outil.

et x_1, x_2, x_3 sont les exposants empiriques de la rugosité.

paramètre	V_c (m/min)	f (mm/tour)	a_p (mm)
valeur max	200	0.762	3
valeur min	30	0.254	1

TABLE 4.2 – Limitations des conditions de coupe.

4.4.1 Implémentation [2]

Dans le processus d'optimisation, nous avons utilisé un essaim de taille $m = 48$ particules. Le nombre maximal d'itérations pour les deux méthodes **DNPSO** et **DNPSO-LNL** est $N_{\text{ite}} = 100$, où "ite" désigne une itération complète.

Pour la mise à jour des déplacements, nous avons utilisé, dans les deux méthodes, les paramètres suivants :

$$c_1=c_2=1$$

$$r_1=r_2=0.4$$

$$W_{max}=0.9 \text{ et } W_{min}=0.4.$$

L'application des méthodes choisies pour la résolution du problème d'optimisation des conditions de coupe (DNPSO et DNPSO-LNL) est présentée par les algorithmes suivants :

PSO à voisinage dynamique (DNPSO)

L'utilisation de cet algorithme consiste en étapes suivantes :

1. Initialisation d'un rang de m particules avec des positions et des déplacements aléatoires dans l'espace de recherche. Ce rang constitue l'essaim initial (première génération).
2. Evaluation de chaque particule du courant essaim, dans l'espace des solutions, en calculant les fonction-objectifs en fonction des variables d'optimisation.
3. Mise à jour du meilleur score local (pbest) de la particule en cours. Si la valeur courante de la particule domine l'évaluation du pbest (c'est-à-dire, la particule a réalisé une nouvelle meilleure position personnelle), alors ; l'évaluation de Pbest sera égale à la l'évaluation courante de la particule et la position de Pbest sera égale à sa position courante. Dans ce cas, le contenu de l'archive illimitée des solutions non dominées (HA) doit être mettre à jour, en sélectionnant le nouveau pbest, si celui-ci n'est dominé par aucun des solutions contenues dans HA.
4. Sélection d'un nouveau leader, pour la particule courante, à partir des éléments du HA. Cette tâche est articulée sur les étapes suivantes :
 - (a) — Calcul des distances entre la particule courante et les particules de l'archive HA dans l'espace des solutions pour le premier objectif (objectif de voisinage).
 - Sélection des n particules voisines de la particule courante (6 particules dans notre application) sur la base des distances calculées dans la première étape. Cet ensemble de solutions constitue le voisinage de la particule courante.
 - Sélection du score local du voisinage en termes de la fonction-objectif du deuxième objectif (objectif d'optimisation). Cet optimum est considéré comme un leader pour la particule courante.

Ce leader constitue, pour la particule, un guide vers le front de Pareto car les éléments de l'archive HA sont considérés comme des solutions Pareto optimales.
 - (b) Mise à jour des positions des particules de l'essaim en utilisant les équations :

$$\vec{x}_i(t) = \vec{x}_i(t-1) + \vec{v}_i(t)$$

$$\vec{v}_i(t) = W\vec{v}_i(t-1) + c_1r_1(pbest_i - \vec{x}_i(t-1)) + c_2r_2(leader_i - \vec{x}_i(t-1))$$

Où :

r_1 et r_2 sont des constantes aléatoires dans l'intervalle $[0,1]$.

$\vec{v}_i(t)$ est le vecteur déplacement .

$\vec{x}_i(t)$ est la position de chaque particule.

pbest : (record local) la position de la meilleure particule du voisinage d'une particule donnée de l'essaim.

leader : une particule utilisée pour guider une autre vers de meilleures régions de l'espace de recherche.

(c) Retour à l'étape 2 si le nombre d'itérations n'arrive pas à N_{ite} .

Cet algorithme peut être résumé comme suit :

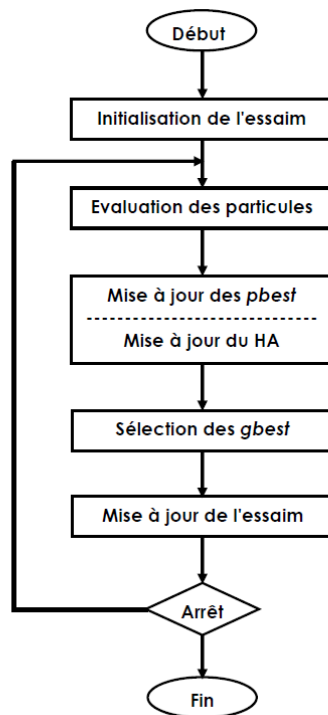


FIGURE 4.1 – Algorithme de DNPSO.

DNPSO à leaders localement non dominés (DNPSO-LNL)

L'algorithme du DNPSO-LNL est présenté par les étapes suivantes :

1. Initialisation d'un rang de m particules avec des positions et des déplacements aléatoires dans l'espace de recherche. Ce rang constitue l'essaim initial (première génération).
2. Initialisation des éléments de l'archive externe (LHA) avec des valeurs infinies pour chaque objectif. Dans notre application, cet archive est de taille 20 (il contient au maximum 20 solutions).
3. Evaluation de chaque particule du courant essaim, dans l'espace des solutions, en calculant les fonction-objectifs en fonction des variables d'optimisation.
4. Sélection d'un nouveau leader, pour la particule courante, à partir des éléments de l'essaim. Cette tâche est articulée sur les étapes suivantes :
 - Calcule des distances géométriques, en termes de tous les objectifs, entre la particule courante et les autres particules de l'essaim dans l'hyperespace des solutions.
 - Sélection des n particules voisines de la particule courante (12 particules dans notre application) sur la base des distances calculées dans la première étape. Cet ensemble de solutions constitue le voisinage de la particule courante.
 - Sélection aléatoire d'un élément, localement non dominé, du voisinage en utilisant la relation de dominance de Pareto. Cet optimum est considéré comme un leader pour la particule courante. Une fois ce nouveau leader est déterminé, le contenu de l'archive des solutions non dominées LHA est mis à jour.
5. Mise à jour du meilleur score personnel (pbest).
6. Mise à jour des positions des particules de l'essaim $\vec{x}_i(t)$ et $\vec{v}_i(t)$.
7. Filtration du LHA en éliminant les solutions dominées.
8. Retour à l'étape 3 si le nombre d'itérations n'arrive pas N_{ite} .

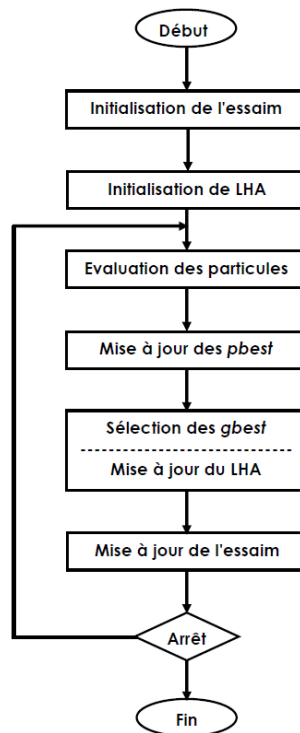


FIGURE 4.2 – Algorithme du DNPSO-LNL.

4.4.2 Modélisation du problème [2]

Dans cet exemple d'application, nous nous intéressons à l'optimisation économique de l'opération de tournage, en prenant en compte deux critères principaux : le temps total de production et la consommation de l'outil. Ainsi, le problème d'optimisation des conditions de coupe se traduit par une double minimisation, à savoir celle du temps d'usinage et de l'usure de l'outil, tout en respectant un ensemble de contraintes techniques et organisationnelles liées à l'ensemble machine-outil-pièce.

En raison des limites imposées par les capacités de la machine, les caractéristiques de l'outil de coupe et les exigences de sécurité de l'opération, les conditions de coupe doivent rester dans une plage définie par les inégalités suivantes :

$$a_p \min \leq a_p \leq a_p \max \quad (4.3)$$

$$f_{\min} \leq f_c \leq f_{\max} \quad (4.4)$$

$$v_c \min \leq v_c \leq v_c \max \quad (4.5)$$

Les fabricants des machine-outil et outils de coupe spécifient des limitations sur les conditions de coupe. Ces limitations concernent les efforts de coupe et la puissance nécessaire à la coupe.

L'effort de coupe, f_c , doit être inférieur à une valeur maximale, f_{max} , obtenue de la condition de stabilité de la machine et l'outil :

$$f_c = 844v_c^{-0.1013} \cdot f^{0.725} \cdot a_p^{0.75} \leq 1100 \quad (4.6)$$

De la même manière, la puissance nécessaire à la coupe, P , doit être inférieure à la valeur maximale, P_{max} , admise par la machine. Cette contrainte est donnée par l'inégalité suivante :

$$P = 14785v_c^{0.91} \cdot f^{0.78} \cdot a_p^{0.75} \leq 5 \quad (4.7)$$

Finalement, la rugosité de surface obtenue, R_a , calculée à l'aide de l'équation 4.2 doit être inférieur à une valeur spécifique $R_{a \max}$ donnée à partir des spécifications techniques. Cette limitation sur la qualité de coupe est présentée par :

$$R_a \leq R_{a \max} \quad (4.8)$$

Les coefficients et les exposants des deux relations sont trouvés expérimentalement par J.S. Agapiou [1]. Le problème d'optimisation des conditions de coupe, basé purement sur un critère économique, peut être représenté par :

$$(P) \quad \begin{aligned} \min \quad Pr &= 1.63 + \frac{286513.25}{Q} \left(1 + \frac{0.5}{T} \right) \\ \min \quad Ut &= \frac{286513.25}{QT} \cdot 100 \\ \text{Sous contraintes} \quad &\begin{cases} 844 \cdot v_c^{-0.1013} \cdot f_c^{0.725} \cdot a_p^{0.75} \leq 1100 \\ 14785 \cdot v_c^{0.91} \cdot f_c^{0.78} \cdot a_p^{0.75} \leq 5 \\ 14785 \cdot v_c^{-1.52} \cdot f_c^{1.04} \cdot a_p^{0.25} \leq 5 \end{cases} \end{aligned}$$

Où : $Q = 1000 \cdot v_c \cdot f \cdot a_p$, $T = \frac{396135004.71}{v_c^4 \cdot f^{1.16} \cdot a_p^{1.4}}$, $1 \leq a_p \leq 3$, $0.254 \leq f \leq 0.762$ et $30 \leq v_c \leq 200$

Ce problème sera résolu par les deux algorithmes DNPSO-LNL et DNPSO.

Les figures 4.3 et 4.4 présentent les résultats graphiques issus des simulations. L'analyse de la distribution des points des archives LHA (Figure 4.3) et HA (Figure 4.4) sur le front de

Pareto met en évidence une bonne diversité ainsi qu'une répartition homogène des solutions, formant des courbes caractéristiques d'un front de Pareto. Ces observations confirment que l'utilisation d'un voisinage dynamique pour la sélection des leaders favorise efficacement la diversification des solutions.

Les deux courbes illustrent la relation entre la consommation d'outil et le temps de production. La figure 4.3 révèle une réduction significative de la consommation d'outil, atteignant 93,31 %, accompagnée d'une augmentation du temps de production de 254,35 %. Il convient de souligner que la réduction de la consommation d'outil reste plus modérée que l'augmentation correspondante du temps de production. De manière analogue, la figure 4.4 montre une augmentation du temps de production de 317,32 %, tandis que la diminution de la consommation d'outil atteint 91,08 %.

Les résultats obtenus par les deux méthodes DNPSO-LNL et DNPSO sont répertoriés respectivement dans les tableaux 4.3 et 4.4. Dans les deux cas, la contrainte limitante identifiée concerne principalement la rugosité de surface obtenue. Ainsi, on trouve que les deux méthodes ont donné des ensembles de solutions non dominées avec des tailles différentes car l'archive externe, des solutions non dominées, utilisé dans le DNPSO-LNL est limité.

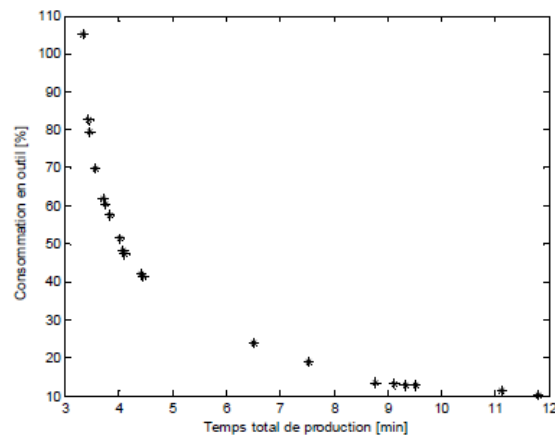


FIGURE 4.3 – Simulation par le DNPSO-LNL du problème P.

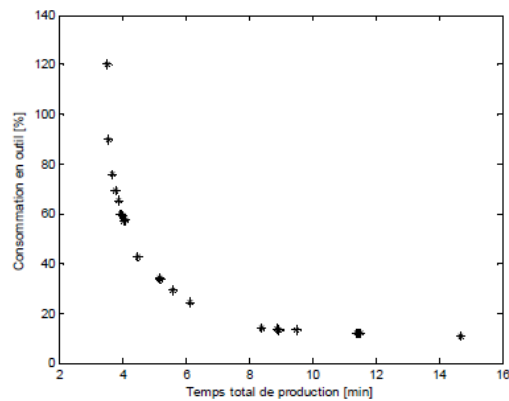


FIGURE 4.4 – Simulation par le DNPSO du problème P.

Par contre, les résultats obtenus à l'aide du DNPSO-LNL sont répartis sur une marge de 3.33 min à 11.8 min pour le temps de production et de 106.2% à 10.19% pour la consommation en outil, tandis que celui du DNPSO sont répartis sur une marge plus au moins proche, de 3.52 min à 14.69 min pour le temps de production et de 120.3% à 10.72% pour la consommation en outil. Cependant, le vecteur nadir correspondant aux résultats trouvés par le DNPSO-LNL (3.33 min , 10.19%) domine celui du DNPSO (3.52 min , 10.72%) Cette comparaison démontre une supériorité relative du DNPSOLNL par rapport au DNPSO à moins, bien sûr, pour la résolution du problème d'usinage. Mais, on voit que cette comparaison n'est pas satisfaisante pour le jugement.

TABLE 4.3 – Solutions obtenues par le DNPSO-LNL pour le problème P.

N°	Pr (min)	Ut (%)	a_p (mm)	f (mm/tr)	v_c (m/min)	F (N)	P (kW)	Ra (μm)
01	3.33	105.20	2.86	0.545	154.77	718.9	5.0	4.91
02	3.44	82.79	2.89	0.493	143.44	679.0	4.3	4.99
03	3.46	79.57	2.99	0.472	141.27	676.0	4.2	4.94
04	3.56	69.90	2.99	0.444	135.72	649.5	3.9	4.94
05	3.70	61.89	2.88	0.428	131.25	616.6	3.6	4.95
06	3.75	60.32	2.83	0.425	130.49	606.0	3.5	4.95
07	3.83	57.48	2.73	0.422	129.05	588.0	3.3	4.96
08	4.02	51.28	2.58	0.413	125.33	555.9	3.0	4.99
09	4.08	48.03	2.68	0.394	122.32	553.9	3.0	4.99
10	4.10	47.33	2.69	0.391	121.72	552.5	2.9	4.99
11	4.42	41.92	2.41	0.386	118.68	505.7	2.6	4.96
12	4.45	41.44	2.39	0.385	118.38	501.8	2.6	4.98
13	6.50	23.95	1.66	0.348	104.09	359.2	1.6	4.99
14	7.53	18.78	1.86	0.275	95.67	334.0	1.3	4.62
15	8.78	13.50	1.82	0.256	86.30	314.7	1.1	4.99
16	9.13	12.26	1.74	0.236	86.04	304.4	1.1	4.96
17	9.34	12.82	1.71	0.255	85.59	299.2	1.1	4.97
18	9.51	12.66	1.66	0.257	85.51	294.6	1.0	4.97
19	11.12	11.26	1.36	0.263	84.83	258.7	0.9	4.99
20	11.80	10.19	1.36	0.254	81.78	252.1	0.8	4.98

Afin de comparer les performances des deux méthodes, d'une manière rationnelle, on utilise la métrique C . cette métrique est définie comme suit :

Soit X' et X'' deux ensembles de vecteurs de décision, la fonction C trace, dans l'intervalle $[0, 1]$, un ordre pour la pair (X', X'') donnée par :

$$C(X', X'') = \frac{|a'' \in X''; \exists a' \in X' : a' \preceq a''|}{|X''|} \quad (4.9)$$

La valeur limite, 1, de C montre que toutes les solutions de l'ensemble X'' sont dominées par ou équivalentes aux solutions de l'ensemble X' , tandis que la valeur 0 montre qu'aucune solution de l'ensemble X'' n'est dominée par une solution de l'ensemble X' .

En utilisant la relation 4.9, l'ordre du pair C (DNPSO, DNPSO-LNL) est d'une valeur nulle et celui du pair C (DNPSO-LNL, DNPSO) est de 0.718. De ce fait, il est clair qu'aucune des solutions obtenues par le DNPSO-LNL n'est dominée par les solutions du DNPSO tandis que la majorité de solutions obtenues par le DNPSO sont dominées par une ou plus d'une solution de l'ensemble trouvé par le DNPSO-LNL.

TABLE 4.4 – Solutions obtenues par le DNPSO du problème P.

N°	Pr (min)	Ut (%)	a_p (mm)	f (mm/tr)	v_c (m/min)	F (N)	P (kW)	Ra (μm)
01	3.52	120.30	2.22	0.597	166.55	631.0	4.7	4.52
02	3.57	89.65	2.35	0.540	150.68	618.5	4.2	4.83
03	3.66	76.10	2.33	0.508	139.33	589.7	3.9	4.88
04	3.77	69.52	2.34	0.486	139.30	575.6	3.6	4.89
05	3.78	69.48	2.34	0.486	139.30	574.7	3.5	4.88
06	3.89	65.41	2.36	0.456	136.84	553.4	3.3	4.72
07	3.94	59.91	2.36	0.451	132.96	550.8	3.2	4.88
08	3.98	59.21	2.36	0.443	132.96	543.4	3.2	4.88
09	4.03	57.09	2.31	0.442	132.78	538.7	3.1	4.76
10	4.05	57.13	2.33	0.436	131.31	533.4	3.1	4.79
11	4.11	57.66	2.14	0.457	132.87	517.7	3.0	4.83
12	4.49	42.79	2.20	0.406	120.67	488.4	2.6	4.99
13	5.19	33.97	1.97	0.376	113.85	427.8	2.1	4.91
14	5.19	33.97	1.97	0.376	113.85	427.7	2.1	4.91
15	5.19	33.96	1.96	0.376	113.85	427.5	2.1	4.91
16	5.19	33.95	1.96	0.376	113.85	427.3	2.1	4.91
17	5.60	29.19	1.94	0.353	108.78	406.7	1.9	4.92
18	6.13	24.40	2.06	0.308	102.39	388.3	1.7	4.79
19	8.39	13.89	1.94	0.254	86.66	327.3	1.2	4.99
20	8.89	13.57	1.79	0.255	86.66	310.0	1.1	4.99
21	8.91	13.47	1.79	0.255	86.46	310.0	1.1	4.95
22	8.93	13.47	1.79	0.254	86.46	309.2	1.1	4.93
23	9.50	13.16	1.66	0.254	86.69	291.7	1.0	4.81
24	11.40	12.11	1.32	0.256	86.89	247.3	0.9	4.56
25	11.40	12.11	1.32	0.256	86.89	246.9	0.9	4.56
26	11.41	12.03	1.32	0.256	86.66	247.4	0.9	4.58
27	11.43	12.02	1.32	0.256	86.66	247.0	0.9	4.57
28	11.45	11.93	1.32	0.256	86.44	247.4	0.9	4.59
29	11.45	11.93	1.32	0.256	86.44	247.4	0.9	4.59
30	11.47	11.92	1.32	0.256	86.44	246.7	0.9	4.59
31	11.51	11.91	1.32	0.255	86.44	246.0	0.9	4.57
32	14.69	10.72	1.00	0.254	86.66	198.8	0.7	4.23

Dans le cas des résultats obtenus avec l'algorithme **DNPSO-LNL**, les meilleures performances en termes de **coût de production** se situent entre les points **9 et 12**. Parmi eux, les **points 11 et 12** présentent les paramètres de coupe les plus adaptés, combinant une **faible consommation d'outil** et un **coût machine raisonnable**. Si l'on considère plutôt le **coût total de production**, les **points 9 et 10** deviennent plus avantageux.

Ce choix n'est cependant pas évident, car il dépend de **préférences subjectives** et souvent **imprécises** des décideurs. Pour faciliter cette décision, on peut recourir à une **fonction d'utilité**, permettant d'évaluer et de comparer la qualité des solutions non dominées, en vue d'identifier un **compromis optimal**.

Ainsi, on a introduit une fonction d'évaluation élémentaire pour présenter la valeur de satisfaction de chaque objectif. Pour chaque fonction-objectif, la valeur de satisfaction élémentaire est présentée par :

$$u_i = \frac{f_i^{\max} - f_i}{f_i^{\max} - f_i^{\min}} \quad (4.10)$$

Où f_i^{\min} est la valeur de la $i^{\text{ème}}$ fonction-objectif considérée complètement satisfaite, et f_i^{\max} la valeur de la même fonction-objectif considérée complètement insatisfaite.

Par conséquent, l'évaluation du $i^{\text{ème}}$ objectif, u_i , représente le degré de satisfaction de la fonction-objectif correspondante à cet objectif comme une valeur entre 0 et 1. La valeur 1 représente le degré d'un objectif complètement satisfait et 0 un objectif complètement insatisfait.

Pour chaque solution non dominée k , la valeur d'évaluation normalisée est calculée par :

$$u^k = \frac{\sum_{i=1}^{N_{\text{obj}}} u_i^k}{\sum_{k=1}^{N_{\text{sol}}} \sum_{i=1}^{N_{\text{obj}}} u_i^k} \quad (4.11)$$

Où N_{sol} est le nombre des solutions non dominées, et N_{obj} est le nombre des objectifs considérés.

La solution qui atteint la valeur maximale de la fonction d'utilité représente le meilleur compromis.

Les éléments des ensembles Pareto optimal obtenus par le DNPSO-LNL et le DNPSO sont évalués en utilisant les équations (4.10) et (4.11). Ensuite, la solution ayant la plus grande valeur d'évaluation, dans chaque ensemble, est sélectionnée comme un meilleur compromis.

Le tableau suivant présente la comparaison entre les meilleurs compromis déterminés et celui obtenu par une méthode agrégée, la méthode Min-max, en utilisant la fonction `fminmax` de Matlab. Il est clair que le coût outil obtenu par la méthode déterministe est le moins élevé par rapport à ceux obtenue par le DNPSO-LNL et DNPSO qui sont proches. Évaluée à, presque, 73.91% cette diminution du coût machine est associée à une importante augmentation dans le coût machine qui arrive à 144.84%. Globalement, on trouve que le moindre coût de production est atteint par le DNPSO-LNL avec une différence de 1.51% par rapport au DNPSO et 43.1% par rapport au Min-max.

TABLE 4.5 – Comparaison des meilleurs compromis du problème P.

Méthode	Pr (min)	Ut (%)	a_p (mm)	f (mm/tr)	v_c (m/min)	Ct (\$)	CI (\$)	PC (\$)
DNPSO-LNL	4,45	41,44	2,39	0,385	118,38	0,207	0,445	0,652
DNPSO	4,49	42,79	2,20	0,406	120,67	0,213	0,449	0,662
Min-max	10,92	10,92	1,47	0,254	82,80	0,054	1,092	1,146

En effet, suivant le coût outil, la solution obtenue par l'approche Min-max peut être sélectionnée, mais dans les situations réelles, le temps de production est l'élément le plus important pour la sélection des conditions de coupe, car il représente plusieurs facteurs de production : un facteur matériel présenté par le temps d'usinage ; un facteur humain présenté par les temps de préparation et de changement d'outil ; un facteur énergétique lié à la consommation de la machine ; et éventuellement un facteur organisationnel correspondant aux temps d'attente et de logistique dans l'atelier.

4.4.3 Optimisation tri-objectif [2]

Dans cette application, l'opération de tournage est étudiée à travers un critère technico-économique. À cette fin, un troisième objectif, relatif à la qualité de la surface obtenue (la rugosité de surface) est introduit. Ainsi, le problème d'optimisation des conditions de coupe consiste à minimiser simultanément le temps de production, l'usure de l'outil et la rugosité de surface, tout en respectant les contraintes et limites technologiques imposées.

Le problème d'optimisation des conditions de coupe, basé sur un critère combiné technico-économique, peut alors être formulé comme suit :

$$(P') \quad \begin{aligned} \min \quad & Ut = \frac{286513.25}{QT} \times 100 \\ \min \quad & Pr = 1.63 + \frac{286513.25}{Q} \left(1 + \frac{0.5}{T}\right) \\ \min \quad & Ra = 14785 \cdot v_c^{-1.52} \cdot f^{1.004} \cdot a_p^{0.25} \end{aligned}$$

$$\text{Sous contraintes} \quad \begin{cases} 844 \cdot v_c^{0.1013} \cdot f^{0.725} \cdot a_p^{0.75} \leq 1100 \\ 14785 \cdot v_c^{0.91} \cdot f^{0.78} \cdot a_p^{0.5} \leq 5 \\ 14785 \cdot v_c^{-1.52} \cdot f^{1.004} \cdot a_p^{0.25} \leq 5 \end{cases}$$

$$\text{Où : } Q = 1000 \cdot v_c \cdot f \cdot a_p, \quad T = \frac{396135004.71}{v_c^4 \cdot f^{1.16} \cdot a_p^{1.4}}, \quad 1 \leq a_p \leq 3, \quad 0.254 \leq f \leq 0.762 \quad \text{et} \quad 30 \leq v_c \leq 200$$

Le traitement d'un problème possédant plus de deux objectifs à l'aide du DNPSO n'est pas défini, donc cette méthode est exclue.

Le tableau présente l'ensemble des solutions non dominées obtenues à l'aide de la méthode de DNPSO-LNL. Ces résultats sont répartis sur une frontière, dite de Pareto, limitée entre 3,69 et 14,60 min pour le temps de production, 10,72 et 212,53% pour la consommation en outil et 1,18 et 4,99 μm pour la rugosité de surface. Une telle diversification offre un avantage considérable, pour les décideurs, lors de la sélection d'une solution satisfaisante. De plus, chacune de ces solutions est un compromis faisable qui satisfait les objectifs considérés, chacun a un tau bien déterminé.

Considérons, par exemple, la rugosité de surface obtenue, les points de 5 à 8 assurent une meilleure qualité de surface (de 2 à 2,5 μm) avec des temps de production et des consommations acceptables. Dans ce sous-ensemble de solutions, on trouve que le point 5 présente un temps de production relativement important. Ainsi, la consommation en outil arrive, dans le point 7, à un pourcentage plus ou moins élevé. Les deux autres points (6 et 8) peuvent être considérés dans le même rang d'importance.

D'autre part, la favorisation du critère économique permet de sélectionner d'autres solutions. Les points 8 et 11 présentent de meilleurs coûts avec des rugosités acceptables, tandis que les points 14, 16, 18 et 20 présentent des coûts minimaux mais de mauvaise qualité de surface. Cependant, on trouve que le point 8 présente un point commun entre les deux considérations du critère économique ou technologique. Par ailleurs, ce même point est sélectionné comme un

TABLE 4.6 – Solutions obtenues par le DNPSO-LNL du problème tri-objectif.

N°	Pr (min)	Ut (%)	Ra (μm)	a_p (mm)	f (mm/tr)	v_c (m/min)	F (N)	P (kW)
01	7,92	131,85	1,18	1,00	0,254	200,00	182,7	1,5
02	5,82	161,74	1,34	1,66	0,254	200,00	268,0	2,3
03	4,97	185,04	1,46	2,33	0,254	200,00	344,9	3,0
04	4,53	204,61	1,56	3,00	0,254	200,00	416,4	3,6
05	9,74	48,53	1,97	1,00	0,254	143,33	188,9	1,1
06	6,64	59,53	2,23	1,66	0,254	143,33	277,2	1,7
07	4,84	133,99	2,26	1,58	0,382	184,95	350,5	2,8
08	5,34	68,11	2,43	2,33	0,254	143,33	356,7	2,2
09	3,98	212,53	2,44	2,73	0,406	200,00	546,0	4,8
10	4,05	153,12	2,56	2,52	0,376	181,91	491,9	3,9
11	4,62	75,31	2,59	3,00	0,254	143,33	430,7	2,6
12	3,88	144,00	2,86	2,66	0,396	176,50	532,6	4,1
13	3,82	137,67	2,97	2,80	0,392	172,73	552,0	4,2
14	5,09	54,09	3,32	2,14	0,314	132,70	394,5	2,2
15	14,60	10,72	4,23	1,00	0,254	86,66	198,0	0,7
16	4,12	60,79	4,23	2,52	0,387	133,49	518,7	3,0
17	9,50	13,16	4,81	1,66	0,254	86,66	291,7	1,3
18	3,69	76,10	4,88	2,33	0,508	143,33	589,7	3,4
19	8,92	13,57	4,90	1,80	0,254	86,66	300,9	1,4
20	4,49	42,79	4,99	2,20	0,406	120,67	488,4	2,6

meilleur compromis en utilisant la fonction d'utilité, présentée dans la section précédente. La comparaison de ce compromis avec celui trouvé par la méthode Min-max est présentée dans le tableau 4.7. Il est clair que le compromis trouvé par le DNPSO-LNL assure, par rapport à celui obtenu par la méthode Min-max, une même qualité de surface en la moitié du temps de production mais avec une consommation en outil doublée. De ce fait, les deux solutions peuvent être considérées équivalentes sans recours au critère coût car le coût de production lié à la première solution est moins de 70% à celui lié à la deuxième.

TABLE 4.7 – Comparaison des meilleurs compromis du problème tri-objectif.

Méthode	Pr (min)	Ut (%)	Ra (μm)	a_p (mm)	f (mm/tr)	v_c (m/min)	Pc (\$)
DNPSO-LNL	5,34	68,11	2,43	2,33	0,254	143,33	0,874
Min-max	10,98	30,34	2,50	1,00	0,254	122,57	1,249

4.5 Conclusion

Dans les deux exemples d'application, le problème d'optimisation des conditions de coupe pour une opération de tournage a été étudié. Une modélisation à deux objectifs, utilisée dans le premier exemple, a permis de comparer la méthode proposée (**DNPSO-LNL**) à la méthode originale (**DNPSO**). Ces deux approches, basées sur l'algorithme *Particle Swarm Optimization* (PSO), ont démontré une efficacité notable dans l'approximation du front de Pareto. Toutefois, la méthode DNPSO-LNL s'est révélée plus performante en termes de qualité des solutions obtenues. Cette supériorité a été confirmée par la comparaison du meilleur compromis issu de l'ensemble des solutions non dominées avec celui obtenu à l'aide de la méthode agrégée *Min-Max*.

La méthode ayant inspiré le développement de DNPSO-LNL est initialement conçue pour résoudre des problèmes à deux objectifs. Or, l'application de cette nouvelle méthode à un problème formulé avec trois objectifs, dans le second exemple, a permis de renforcer l'hypothèse de sa supériorité par rapport à la méthode originale. Ainsi, la comparaison avec la méthode Min-Max a mis en évidence l'intérêt d'adopter une méthode de type Pareto pour la résolution efficace de problèmes d'optimisation multi-objectifs.

Par conséquent, le critère coût de fabrication, qui constitue dans d'autres travaux l'un des objectifs principaux d'optimisation, est utilisé dans cette étude comme critère central d'analyse et de synthèse. Ce choix a permis de mener une discussion approfondie sur l'aspect économique des opérations de tournage, mettant en évidence l'importance de ce paramètre dans la sélection optimale des conditions de coupe.

Conclusion générale et perspectives

Ce mémoire a permis d'explorer en profondeur l'optimisation mono et multi-objectifs, en mettant en évidence leur importance dans la modélisation et la résolution des problèmes. Après avoir étudié les fondements théoriques et les méthodes de résolution associées, une application concrète a été réalisée dans le cadre de l'optimisation des conditions de coupe en usinage.

L'implémentation et la comparaison des deux algorithmes DNPSO et DNPSO-LNL ont montré que les approches basées sur la métaheuristique des essaims particulaires permettent de générer un ensemble de solutions efficaces, offrant une vision globale des compromis possibles entre les différents objectifs industriels (temps, usure, qualité). L'algorithme DNPSO-LNL s'est révélé plus performant en termes de qualité et de diversité des solutions.

Ce travail confirme l'intérêt des méthodes d'optimisation multi-objectifs dans les environnements industriels, où la prise de décision doit concilier des contraintes techniques, économiques et qualitatives.

Bibliographie

- [1] Agapiou, J.S. The Optimization of machining operations based on a combined criterion, Part 1 : The use of combined objectives in single-pass operations. *Journal of Engineering for Industry*, 114, pp.500-507, 1992.
- [2] Ameer, Toufik. Optimisation du processus d'usinage en utilisant la méthode des essaims de particules. Thèse présentée en vue de l'obtention du diplôme de Doctorat en sciences en Génie Mécanique, Université Mohamed Khider – Biskra, 2016.
- [3] Benlahrache, Nadira. Optimisation Multi-Objectif Pour l'Alignement Multiple de Séquences. Mémoire de Magistère, Université Mentouri de Constantine, 2007.
- [4] Boudab, Aissa et Abdelkrim Benhammada. Méthodes d'optimisation avec contraintes. Mémoire de Master, Centre Universitaire Abd elhafid Boussouf Mila, 2022.
- [5] Bouali, Tahar. Convergence des méthodes du gradient conjugué avec la recherche linéaire non monotone. Thèse de Doctorat, Université de Badji Mokhtar Annaba, 2015.
- [6] Boyd, Stephen and Vandenberghe, Lieven. *Convex Optimization*, 2004. Cambridge University Press.
- [7] Bero, Alain. Optimisation multiobjectif et stratégie d'évolution en environnement dynamique. Thèse présentée à l'université des sciences sociales de Toulouse 1, 2001.
- [8] Collette, Yann et Patrick Siarry. *Livre optimisation multiobjectif*. Éditions Eyrolles, Paris, 2005.
- [9] Coello Coello, Carlos A., Lamont, Gary B., et Van Veldhuizen, David A. *Evolutionary Algorithms for Solving Multi-Objective Problems*. 2e éd., Springer, 2007.
- [10] Deb, Kalyanmoy. *Multi-Objective Optimization Using Evolutionary Algorithms*. Wiley, 2001.
- [11] Dodge, Yadolah, avec la collaboration de Sylvie Gonano-Weber et Jean-Pierre Renfer. *Optimisation appliquée*. Springer, Paris, 2004.

-
- [12] Djeddou, Chaima et Bennoui Rebh. La méthode de direction de descente pour l'optimisation multi-objectifs sans contraintes. Mémoire de Master, Université Mohamed El-Bachir El-Ibrahimi de Bordj Bouararidj, 2021.
- [13] Ehrgott, Matthias. *Multicriteria Optimization*. 2e éd., Springer, 2005.
- [14] Fali, Fatima. L'optimisation multiobjectif non linéaire bi-niveau. Université des Sciences et de la Technologie Houari Boumediène, 2022.
- [15] Gournay, Frédéric et Aude Rondepierre. *Introduction à l'Optimisation Numérique*. INSA Toulouse, 2021.
- [16] Hafaidia, Imane. Quelle est la meilleure condition de conjugaison ? Thèse de Doctorat, Université 8 Mai 1945 Guelma, 2022.
- [17] Hebhoub, Fahima. Sur quelques méthodes de résolution des problèmes d'optimisation non linéaire sans contrainte. Mémoire de Magister, Université Badji Mokhtar Annaba, 2001.
- [18] Hestenes, M.R. et Stiefel, E. *Methods of conjugate gradients for solving linear systems*. NBS, Washington DC, 1952.
- [19] Khider, Imane et Marwa Keddar. Outil d'aide à la décision pour la hiérarchisation des projets d'exploration de la Sonatrach. Mémoire de Master, Université M'hamed Bougara de Boumerdès, 2022.
- [20] Leslous, Fadila, Mouloud Goubi et Mohand Ouanes. A new approach for non-convex optimisation problems applied to Hump and benchmark functions. *International Journal of Mathematics in Operational Research*, Vol.25, No.3, 2023.
- [21] Leslous, Fadila. Problème d'optimisation non convexe et optimisation DC. Thèse de Doctorat, spécialité Recherche Opérationnelle et Optimisation, 2023.
- [22] Maitine, Bergounioux. *Optimisation et contrôle des systèmes linéaires. Cours et exercices avec solutions*, Dunod, Sciences Sup, 2001.
- [23] Miettinen, Kaisa. *Nonlinear Multiobjective Optimization*. Springer, 1999.
- [24] Othmani, Imed. *Optimisation multicritère : fondements et concepts. Modélisation et simulation*. Université Joseph-Fourier - Grenoble I, 1998.
- [25] Ouznadj, Said. Optimisation multi-objectif dans des problèmes de filtres. Thèse de Doctorat en mathématique (Recherche Opérationnelle), Université USTHB, 2020.
- [26] Pareto, Vilfredo. *Cours d'économie politique*. F. Rouge, Lausanne, 1896.
- [27] Ramdani, Zoubir. *Cours d'optimisation multiobjectifs*. Université Mohamed Al-Bachir El-Ibrahimi de Bordj-Bouararidj, 2022.

- [28] Vignes, J. Algorithmes numériques : analyse et mise en oeuvre. Vol. 2. Éditions Technip, Paris, 1980.