

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 : Recherche Opérationnelle (RO)

Présenté par :

Diffallah Mohammed

Seghir Elhadi

Thème

**Sur le Contrôle Optimal en Multi-objectifs : de la Modélisation
à la Résolution en Discret**

Encadré par :
Mr **CHEBBAH Mohammed**

Devant le jury d'examen composé de :

AICHE Farid

CHEBBAH Mohammed

HARRACHE Fazia

UMMTO

UMMTO

UMMTO

Président

Rapporteur

Examinatrice

Soutenue : le 03/07/2025

Remerciement

Nous remercions **DIEU** Tout-Puissant de nous avoir guidés vers les portes du savoir, d'avoir illuminé notre chemin et de nous avoir donné le courage et la patience nécessaires à la réalisation de ce travail.

Nous remercions **Mr. CHEBBAH Mohammed**, notre encadrant, pour son accompagnement, son aide et ses encouragements. Nous lui témoignons notre respectueuse gratitude.

Nos sincères remerciements s'adressent également aux membres du jury pour avoir accepté d'examiner notre travail.

Nous tenons également à remercier tous ceux qui ont contribué, de près ou de loin, à la réalisation de ce travail.

Dédicaces

Il m'est un énorme plaisir de dédier ce travail :

À la mémoire de mon cher oncle **Elhadi**, tombé au champ d'honneur, qui a donné sa vie pour la patrie.

À mon grand-père **Omar**, paix à son âme.

À mes chers parents, qui ont sacrifiés leur vie pour me voir réussir.

À mon grand-père et à mes grand-mères.

À mon cher frère **Yacine**.

À toute ma famille.

À tous mes amis et collègues.

Elhadi

Dédicaces

Il m'est d'un immense plaisir de dédier ce travail :

À mes chers parents, qui ont sacrifiés leur vie pour me voir réussir. Leur amour, leur patience et leurs sacrifices sont la clé de mon parcours.

À la mémoire de mes grands-parents, dont les prières et les valeurs continuent de me guider.

À ma chère grand-mère, pour sa tendresse, ses prières, et sa présence réconfortante.
Que Dieu lui accorde longue vie et santé.

À mon cher frère **Nabil** et ma chère soeur **Thanina**.

À toute ma famille.

À tous mes amis et collègues.

Mohammed

Table des matières

Table des matières	5
Introduction générale	8
1 Généralités sur le contrôle optimal	11
1.1 Introduction	11
1.2 Théorie du contrôle et des systèmes de contrôle	11
1.2.1 Objet de la commande	11
1.2.2 Conditions initiales du système	12
1.2.3 Le but de la commande	12
1.2.4 Classes de commandes admissibles	12
1.2.5 Critères de qualité	13
1.3 Position du problème de contrôle optimal	13
1.3.1 Temps optimal	14
1.3.2 Coût optimal	14
1.3.3 Problème de Mayer-Lagrange	14
1.4 Contrôlabilité	15
1.4.1 Contrôlabilité des systèmes linéaires	16
1.4.2 Contrôlabilité des systèmes linéaires autonomes	17
1.4.3 Contrôlabilité des systèmes non linéaires	18
1.5 Observabilité	19
1.5.1 Observabilité des systèmes linéaires	19
1.5.2 Définition et critères d'observabilité	20
1.5.3 Observabilité des systèmes non linéaires	21
1.6 Stabilité	21
1.6.1 Stabilité des systèmes linéaires	22
1.6.2 Stabilisation locale d'un système non linéaire	22
1.7 Principe du maximum de Pontryagin	23
1.8 Méthodes de résolution numérique	26
1.8.1 Méthodes indirectes	26

1.8.2	Méthodes directes	26
1.8.3	Exemple [3]	27
1.9	Conclusion	30
2	Développement et Application de la Méthode Adaptée	31
2.1	Introduction	31
2.2	Position du problème	31
2.3	Accroissement de la fonctionnelle — Critère d’optimalité et de suboptimalité	32
2.4	Itération de l’algorithme	36
2.4.1	Changement de plan	36
2.4.2	Changement du support	37
2.5	Algorithme de la méthode	40
2.6	Convergence de la méthode adaptée	41
2.7	Exemples	41
2.8	Etude comparative entre la méthode adaptée et le simplexe	46
2.9	Conclusion	46
3	Programmation multi-objectif	47
3.1	Introduction	47
3.2	La dominance	50
3.2.1	Dominance forte	51
3.2.2	Dominance faible	51
3.2.3	Non-dominance	51
3.3	Efficacité	51
3.3.1	Pareto optimale	51
3.3.2	Efficacité forte	52
3.3.3	Efficacité faible	52
3.4	Optimalité au sens de Pareto	53
3.5	Caractérisations des solutions efficaces	55
3.5.1	Théorème d’Ecker et Wendell	56
3.5.2	Théorème de Benson	56
3.6	Points spéciaux et matrice des gains	57
3.6.1	Le point idéal	57
3.6.2	Le point anti-idéal	58
3.6.3	Le point de référence	58
3.6.4	Matrice des gains	58
3.6.5	Le point Nadir	58
3.6.6	Front de Pareto	58

3.7	Fonctions scalarisantes	59
3.7.1	Fonctions linéaires	59
3.7.2	Norme L_p pondérée	60
3.7.3	Norme L_1 de Tchebycheff pondérée	60
3.7.4	Norme Tchebycheff pondérée augmentée	60
3.8	Méthodes de résolution d'un programme multi-objectif	60
3.8.1	Méthode du simplexe multi-objectif	60
3.8.2	Méthode de la somme pondérée ou méthode d'agrégation des ob- jectifs	70
3.8.3	Méthode des ε -contraintes	71
3.9	Conclusion	72
4	Contrôle Optimal Linéaire Discret : Méthode Adaptée	73
4.1	Introduction	73
4.2	Le cas discret	73
4.2.1	Position du problème	73
4.2.2	Notion de commandabilité	75
4.2.3	Calcul de l'accroissement de la fonctionnelle	76
4.2.4	Critère d'optimalité	77
4.2.5	Critère de suboptimalité	77
4.2.6	Méthode de résolution	78
5	Le Contrôle Optimal en multi-objectifs	82
5.1	introduction :	82
5.2	Position du problème	82
5.3	Modèle mathématique du problème	83
5.4	MÉTHODE DE RÉOLUTION	85
5.4.1	ALGORITHME DE RÉOLUTION (variante 1)	85
5.4.2	ALGORITHME DE RÉOLUTION (variante 2)	86
5.5	EXEMPLE D'APPLICATION	86
5.6	Conclusion	89
6	Implémentation informatique	90
6.1	Introduction au logiciel LINGO	90
6.1.1	Présentation du logiciel LINGO	90
6.1.2	Interface du logiciel	92
6.1.3	Exemple d'application	96
6.2	Définition du logiciel MATLAB	98
6.2.1	Description de l'interface MATLAB	98
6.2.2	Méthode de travail	100

6.2.3	Création de fichiers de commande et de fonctions utilisateur . . .	101
6.2.4	Exemple d'application	103
6.3	Qu'est-ce que <i>Visual Xpress</i> ?	105
6.3.1	Présentation du logiciel	105
6.3.2	Les étapes du programme	107
6.3.3	Application	109
6.4	Conclusion général	110

Introduction générale

Dans un monde où les systèmes dynamiques deviennent de plus en plus complexes et interconnectés, le besoin de prendre des décisions optimales dans des environnements contraints s'impose avec une acuité croissante. Qu'il s'agisse de piloter un système industriel, de gérer une chaîne logistique ou d'optimiser une stratégie énergétique, la maîtrise des outils mathématiques permettant de modéliser, d'analyser et de résoudre ces problèmes est devenue essentielle. C'est dans ce contexte que s'inscrit la théorie du **contrôle optimal**, un champ de recherche alliant les mathématiques appliquées, l'informatique et l'ingénierie.

Le contrôle optimal vise à déterminer les lois de commande qui pilotent un système dynamique d'une manière optimale, selon un critère de performance donné, tout en respectant les contraintes imposées par le système. À travers les décennies, cette discipline s'est enrichie de méthodes puissantes comme le *principe du maximum de Pontryagin*, la *programmation dynamique*, ou encore des approches numériques adaptées aux cas linéaires, non linéaires, continus ou discrets.

Ce mémoire s'articule autour de plusieurs axes complémentaires. Dans un premier temps, nous introduisons les notions fondamentales du contrôle optimal, en nous intéressant notamment aux concepts de contrôlabilité, d'observabilité, de stabilité, et aux méthodes d'optimisation associées. Ensuite, nous développons une **méthode adaptée** pour résoudre des problèmes linéaires à variables bornées, en mettant en lumière son algorithme, ses conditions de convergence et son efficacité à travers des exemples concrets.

Nous abordons ensuite la problématique plus générale de l'**optimisation multi-objectif**, qui intervient lorsque plusieurs fonctions de coût doivent être optimisées simultanément. Nous explorons alors les notions de dominance, d'efficacité au sens de Pareto, et présentons différentes méthodes de résolution adaptées à ce type de problèmes.

Enfin, une attention particulière est portée à l'**implémentation informatique** des méthodes étudiées. L'utilisation d'outils comme LINGO, MATLAB ou encore Visual Xpress permet non seulement de valider les méthodes théoriques, mais également de préparer leur application dans un cadre professionnel ou industriel.

Afin de structurer notre démarche, ce mémoire est organisé selon le plan suivant :

- **Chapitre 1** : Généralités sur le contrôle optimal. Présentation des fondements de la théorie, des différentes catégories de commandes, des notions de contrôlabilité, observabilité, stabilité, et du principe du maximum de Pontryagin.
- **Chapitre 2** : Développement et application de la méthode adaptée. Formulation mathématique du problème, méthode de résolution et étude de la convergence.
- **Chapitre 3** : Programmation multi-objectif. Notions de dominance, efficacité, et méthodes de scalarisation permettant d'identifier des solutions optimales au sens de Pareto.
- **Chapitre 4** : Contrôle optimal linéaire discret. Adaptation de la méthode au cas discret, étude de la commandabilité et des critères d'optimalité.
- **Chapitre 5** : Contrôle optimal multi-objectif. Fusion des approches précédentes pour traiter les problèmes où plusieurs objectifs sont à optimiser dans un cadre discret.
- **Chapitre 6** : Implémentation informatique. Présentation des logiciels spécialisés (LINGO, MATLAB, Visual Xpress), interfaces, algorithmes et exemples d'application.

Ce travail vise donc à établir un pont entre la **rigueur théorique** et les **applications pratiques**, dans une perspective de résolution efficace des problèmes de contrôle optimal rencontrés dans divers domaines scientifiques et technologiques.

Chapitre 1

Généralités sur le contrôle optimal

1.1 Introduction

Dans ce chapitre, nous allons étudier les notions de base du contrôle optimal (commande admissible, but de la commande, état initial du problème, etc.) ainsi que la contrôlabilité des systèmes linéaires et non linéaires, l'observabilité, la stabilité tout en donnant un aperçu du Principe du Maximum de Pontryagin.

1.2 Théorie du contrôle et des systèmes de contrôle

D'un point de vue mathématique, un système de contrôle est un système dynamique dépendant d'un paramètre dynamique appelé le contrôle. Pour le modéliser, on peut utiliser des équations différentielles, intégrales, fonctionnelles, aux différences finies, aux dérivées partielles, ou encore stochastiques. Pour cette raison, la théorie du contrôle se trouve à l'intersection de nombreux domaines mathématiques. Les contrôles sont des fonctions ou des paramètres, habituellement soumis à des contraintes.

Le problème général de contrôle optimal est constitué des éléments suivants :

1.2.1 Objet de la commande

Le système peut comporter de nombreuses variables ou paramètres. On suppose que n variables sont nécessaires pour décrire son comportement. L'identification de ces variables et la description du système en fonction de celles-ci est une tâche très importante : c'est l'étape de modélisation mathématique.

- **Variables d'état** : Notées x_i , $i = 1 \dots n$, elles décrivent l'évolution du système dans le temps. Ces variables sont donc des fonctions du temps t , soit $x_i(t)$. Elles sont gouvernées par n équations différentielles (ou autres) du premier ordre sur

un intervalle de temps $T = [t_0, t^*]$ de la forme :

$$\dot{x}_i = f_i(t, x_1 \dots x_n, u_1, \dots, u_m), \quad i = 1 \dots n.$$

— **Variables de contrôle** : Notées $u_j(t)$, $j = 1 \dots m$, elles interviennent dans la dynamique du système.

1.2.2 Conditions initiales du système

La condition initiale du système, $x_0 = x(t_0)$, est un vecteur donné dans un espace de phase. En réalité, les composantes de $x(t)$ et de x_0 peuvent représenter des grandeurs physiques telles que la position, la vitesse, la température, etc.

1.2.3 Le but de la commande

Dans un problème de contrôle optimal, l'objectif est d'amener l'état initial $x_0 = x(t_0)$, avec $x_0 \in M_0$, à un autre état $x^* = x(t^*)$, avec $x^* \in M_1$, où M_0 est l'ensemble de départ et M_1 l'ensemble d'arrivée.

1.2.4 Classes de commandes admissibles

L'ensemble U des contrôles admissibles peut être non borné, borné, ou du type bang-bang.

Commande bornée Une commande $u(t)$ est dite bornée si elle est minorée et majorée par des constantes a_j et b_j sous la forme :

$$a_j \leq u_j(t) \leq b_j, \quad j = 1 \dots m, \quad t \in [t_0, t^*].$$

Si de plus $a_j \leq u_j(t) \leq b_j$, on peut exprimer u_j sous la forme :

$$u_j = \frac{1}{2}(a_j + b_j) + \frac{1}{2}(a_j - b_j)v_j,$$

où v_j est une variable intégrable vérifiant $-1 \leq v_j \leq 1$. Ainsi, lorsque U est borné, il est souvent pratique de ramener les commandes dans l'intervalle $[-1, 1]$.

Commande bang-bang Un contrôle $u \in U$ est dit *bang-bang*, si pour chaque instant t et chaque indice $j = 1 \dots m$, on a :

$$|u_j(t)| = 1.$$

1.2.5 Critères de qualité

L'objectif, lors de la formulation d'un problème de contrôle, est de fournir une motivation physique pour la sélection d'une mesure de qualité du système. Il s'agit de définir une expression mathématique qui lorsqu'elle est optimisée, indique que le système atteint un état désirable. Ainsi, le choix d'une mesure de qualité est une traduction mathématique des exigences physiques du système.

Le critère de qualité, également appelé coût ou fonction objectif, est généralement décrit par la formule :

$$J(x, u) = g(t^*, x^*) + \int_{t_0}^{t^*} f_0(t, x, u) dt.$$

Cette fonctionnelle comporte deux parties : $g(t^*, x^*)$ est le coût terminal, une sorte de pénalité liée à la fin de l'évolution du système au temps final t^* . Ce terme est important lorsque t^* est libre ; sinon il est constant.

Le second terme, $\int_{t_0}^{t^*} f_0(t, x, u) dt$, dépend de l'état du système tout au long de la trajectoire de la solution, définie par les variables d'état ($x_i, i = 1 \dots n$) et les variables de contrôle ($u_j(t), j = 1 \dots m$).

— **Équations d'état** : Les n variables d'état sont régies par n équations différentielles du premier ordre :

$$\dot{x} = f(t, x, u), \quad t \in [t_0, t^*].$$

où \dot{x} est le vecteur dérivé par rapport au temps t de toutes les composantes de x .

1.3 Position du problème de contrôle optimal

Dans cette section, nous présentons la formulation générale d'un problème de contrôle optimal. Pour tout contrôle $u \in U$, le coût de la trajectoire associée $x(t)$ sur l'intervalle $[t_0, t^*]$ est défini par :

$$J(x, u) = g(t^*, x^*) + \int_{t_0}^{t^*} f_0(t, x(t), u(t)) dt,$$

avec U l'ensemble des contrôles admissibles sur $[t_0, t^*]$.

La formulation générale du problème de contrôle optimal est la suivante :

$$\left\{ \begin{array}{l} \min_u J(x, u) = g(t^*, x(t^*)) + \int_{t_0}^{t^*} f_0(t, x(t), u(t)) dt, \quad (1) \\ \dot{x}(t) = f(t, x(t), u(t)), \quad (2) \\ x(t_0) = x_0 \in M_0, \quad (3) \\ x(t^*) = x^* \in M_1, \quad (4) \\ u \in U, \quad t \in I = [t_0, t^*]. \quad (5) \end{array} \right. \quad (1.1)$$

Où M_0 (ensemble de départ) et M_1 (ensemble d'arrivée) sont des sous-ensembles de \mathbb{R}^n , I un intervalle de \mathbb{R} , $x_0 = x(t_0)$ est la position initiale du système (2), $x(t^*)$ est sa position terminale. En pratique, la position du système peut représenter la vitesse, la position, la température, etc.

Le problème de contrôle optimal est de déterminer les trajectoires $x(t)$ solutions de $\dot{x}(t) = f(t, x(t), u(t))$ qui minimisent le coût $J(x, u)$ en satisfaisant les contraintes.

1.3.1 Temps optimal

On parle d'un problème en temps optimal lorsque $f_0(t, x, u) = 1$, $g(t^*, x^*) = 0$, et le temps final t^* est libre dans l'expression :

$$\min_u \int_{t_0}^{t^*} 1 dt.$$

1.3.2 Coût optimal

On parle d'un problème en coût optimal lorsque le temps final t^* est fixé dans l'expression :

$$\min_u g(t^*, x^*) + \int_{t_0}^{t^*} f_0(t, x, u) dt.$$

Remarque 1.1. Il existe des problèmes qui combinent les deux critères physiques de qualité, et on parlera dans ce cas d'un problème de contrôle en temps et en coût optimal.

1.3.3 Problème de Mayer-Lagrange

L'objectif du problème de Mayer-Lagrange est de minimiser le coût :

$$J(t^*, u) = g(t^*, x^*) + \int_{t_0}^{t^*} f_0(t, x, u) dt.$$

Lorsque $g = 0$ dans l'expression de la fonctionnelle J , on parlera d'un **problème de Lagrange**.

Lorsque $f_0 = 0$, on parlera d'un **problème de Mayer**.

1.4 Contrôlabilité

La contrôlabilité est l'un des concepts centraux de la théorie du contrôle optimal. Elle a été introduite par Kalman [2] en 1960 pour des systèmes linéaires de la forme :

$$\dot{x}(t) = A(t)x(t) + B(t)u(t).$$

Un système de contrôle est dit **contrôlable** si on peut l'amener en temps fini d'un état initial arbitraire vers un état final prescrit (voir figure 1.1).

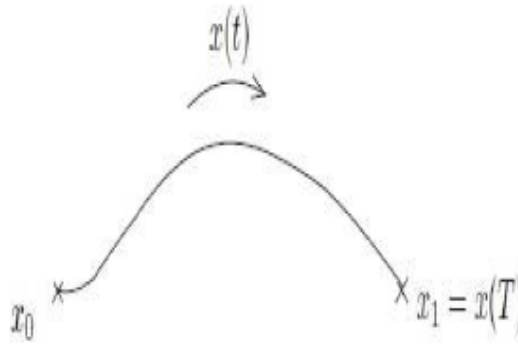


FIGURE 1.1 – Problème de contrôlabilité [3]

Dans cette section, nous allons étudier cette dernière en présentant certaines de ses propriétés.

Définition 1.1. Considérons le système contrôlé :

$$\forall t \in I, \quad \dot{x}(t) = A(t)x(t) + B(t)u(t) + r(t), \quad x(t_0) = x_0. \quad (1.2)$$

L'ensemble des points accessibles à partir de x_0 en un temps $t^* > 0$ est défini par :

$$\text{Acc}(x_0, t^*) = \{x_u(t^*) \mid u \in U\}.$$

où $x_u(\cdot)$ est la solution du système (1.2) associée au contrôle u . Autrement dit, $\text{Acc}(x_0, t^*)$ est l'ensemble des extrémités des solutions de (1.2) au temps t^* , lorsqu'on fait varier le contrôle u (voir figure 1.2).

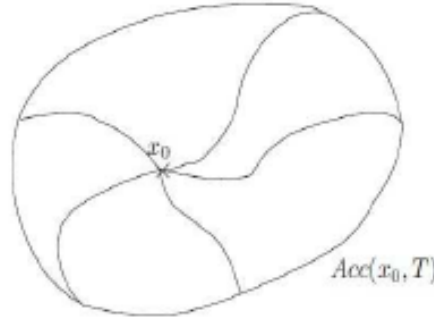


FIGURE 1.2 – Ensemble accessible [3]

1.4.1 Contrôlabilité des systèmes linéaires

La formulation mathématique d'un système de contrôle linéaire est la suivante :

$$\dot{x}(t) = A(t)x(t) + B(t)u(t) + r(t), \quad x(t_0) = x_0, \quad \forall t \in I$$

où I est un intervalle de \mathbb{R} , B , A et r sont trois applications localement intégrables sur I à valeurs respectivement dans $M_n(\mathbb{R})$, $M_{n,n}(\mathbb{R})$ et \mathbb{R}^n .

$M_n(\mathbb{R})$ est l'ensemble des matrices réelles de dimension n , et $M_{n,n}(\mathbb{R})$ est l'ensemble des matrices de n lignes et n colonnes.

L'ensemble des contrôles u considérés est l'ensemble des applications mesurables localement bornées sur I à valeurs dans un sous-ensemble $U \subset \mathbb{R}^m$.

Soit $F(\cdot) : I \rightarrow M_{n,n}(\mathbb{R})$ la résolvante du système linéaire homogène $\dot{x}(t) = A(t)x(t)$, défini par :

$$\{\dot{F}(t) = A(t)F(t), \quad F(t_0) = I_d\}$$

Pour tout contrôle u , le système $\dot{x}(t) = A(t)x(t) + B(t)u(t) + r(t)$, $x(t_0) = x_0$, admet une unique solution $x(\cdot) : I \rightarrow \mathbb{R}^n$ absolument continue, donnée par :

$$x(t) = F(t)x_0 + \int_{t_0}^{t^*} F(t)F(s)^{-1}(B(s)u(s) + r(s)) ds, \quad t \in I$$

Si $r = 0$ et $x_0 = 0$, la solution du système s'écrit :

$$x(t) = F(t) \int_{t_0}^{t^*} F(s)^{-1} B(s) u(s) ds$$

Elle est linéaire en u .

Le théorème suivant donne une condition générale de la contrôlabilité des systèmes linéaires.

Théorème. [1]

Le système $\dot{x}(t) = A(t)x(t) + B(t)u(t) + r(t)$ est contrôlable en temps t^* si et seulement si la matrice :

$$C(t^*) = \int_{t_0}^{t^*} F(t)^{-1} B(t) B'(t) F(t)^{-1} dt$$

appelée **matrice de contrôlabilité**, est inversible.

Remarque 1.2. Cette condition ne dépend pas de x_0 , c'est-à-dire que si un système linéaire est contrôlable en temps t^* depuis x_0 , alors il est contrôlable en temps t^* depuis tout autre point.

1.4.2 Contrôlabilité des systèmes linéaires autonomes

Le système (1.2) est dit **autonome** lorsque les matrices A et B ne dépendent pas de t . Dans ce cas, la matrice est donnée par :

$$F(t) = e^{At}$$

et la solution du système associée au contrôle u s'écrit :

$$\forall t \in I, \quad x(t) = e^{At} \left(x_0 + \int_{t_0}^{t^*} e^{-As} (B(s)u(s) + r(s)) ds \right).$$

Le théorème suivant donne une condition **nécessaire et suffisante** de contrôlabilité dans le cas où le contrôle u n'a pas de contrainte.

Théorème. [2] On suppose que $U = \mathbb{R}^m$. Le système

$$\dot{x}(t) = A(t)x(t) + B(t)u(t) + r(t)$$

est **contrôlable en temps t^*** si et seulement si la matrice :

$$C = (B, AB, \dots, A^{n-1}B)$$

est de **rang** n .

Remarque 1.3. La matrice C est appelée **matrice de Kalman**, et la condition $\text{rang}(C) = n$ est appelée **condition de Kalman**.

Exemple 1.1. *Considérons le système suivant :*

$$\dot{x}(t) = \begin{pmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} u(t) = Ax(t) + Bu(t) + r;$$

avec

$$r = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

Ce système est **contrôlable** car la matrice de Kalman

$$C = (B, AB) = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

est de rang $2 = n$.

Remarque 1.4. Dans le cas où le contrôle u est **contraint**, c'est-à-dire qu'il appartient à un sous-ensemble $U \subset \mathbb{R}^m$, les **propriétés de contrôlabilité globale** sont **liées aux propriétés de stabilité de la matrice A** .

1.4.3 Contrôlabilité des systèmes non linéaires

La **contrôlabilité** est un concept clé pour comprendre les propriétés structurelles et qualitatives d'un système, notamment sa **stabilisation**. L'extension de cette notion aux systèmes **non linéaires**, qu'ils soient de **dimension finie ou infinie**, a fait l'objet de nombreuses recherches ces dernières années. Cependant, ce domaine reste vaste et encore largement exploré.

La plupart des études se sont concentrées sur des généralisations naturelles de l'équation linéaire :

$$\dot{x} = Ax + Bu$$

Le résultat suivant donne une condition pour la **contrôlabilité locale** des **systèmes non linéaires**.

Proposition 1.1. [3]

Considérons le système :

$$\dot{x}(t) = f(t, x(t), u(t)), \quad x(0) = x_0$$

avec la condition $f(x_0, u_0) = 0$.

On définit les matrices :

$$A = \frac{\partial f}{\partial x}(x_0, u_0), \quad B = \frac{\partial f}{\partial u}(x_0, u_0)$$

Si la matrice de contrôle :

$$\text{rang}(B, AB, A^2B, \dots, A^{n-1}B) = n$$

alors le système est **localement contrôlable** en x_0 .

Remarque 1.5. En général, le problème de **contrôlabilité** est complexe. Cependant, certaines **techniques** permettent de déduire la **contrôlabilité locale** en étudiant les **systèmes linéarisés**.

1.5 Observabilité

L'observabilité d'un processus est un concept fondamental dans le domaine de l'estimation de l'état. En effet, pour reconstruire les états inaccessibles d'un système, il est essentiel de savoir à l'avance si les variables d'état sont observables ou non.

L'observabilité d'un système est la propriété qui permet de déterminer si l'état peut être identifié uniquement à partir de la connaissance des signaux d'entrée et de sortie. Dans le cas des systèmes non linéaires, cette notion dépend des entrées et des conditions initiales.

1.5.1 Observabilité des systèmes linéaires

Considérons le système linéaire autonome suivant :

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned} \tag{1.3}$$

où :

— $x(t) \in \mathbb{R}^n$ est le vecteur des états,

- $u(t) \in \mathbb{R}^m$ représente les entrées (commandes),
- $y(t) \in \mathbb{R}^p$ est le vecteur des sorties,
- $A \in \mathbb{M}_n(\mathbb{R})$, $B \in \mathbb{M}_{n,m}(\mathbb{R})$, $C \in \mathbb{M}_{p,n}(\mathbb{R})$, $D \in \mathbb{M}_{p,m}(\mathbb{R})$.

Dans ce qui suit, nous supposons que $D = 0$, car cela ne modifie pas les résultats à venir.

1.5.2 Définition et critères d'observabilité

Soit $(x_u(t, x_0), y_u(t, x_0))$ la solution du système (1.3) telle que $x_u(0, x_0) = x_0$.

Définition 1.2. Le système (1.3) est observable en temps T si :

$$\forall x_1, x_2 \in \mathbb{R}^n, \quad x_1 \neq x_2 \Rightarrow \exists u \in L^\infty([0, T], \mathbb{R}^m) \text{ tel que } y_u(\cdot, x_1) \neq y_u(\cdot, x_2)$$

Dans ce cas, on dit que x_1 et x_2 sont distinguables.

De manière équivalente, on peut dire que :

$$\forall x_1, x_2 \in \mathbb{R}^n, \quad \forall u \in L^\infty([0, T], \mathbb{R}^m), \quad y_u(\cdot, x_1) = y_u(\cdot, x_2) \Rightarrow x_1 = x_2$$

C'est-à-dire que la connaissance de la trajectoire observée permet de déterminer de manière unique l'état initial.

Théorème. Le système (1.3) est observable (pour tout temps T) si et seulement si :

$$\text{rang} \begin{pmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{pmatrix} = n$$

Lemme 1.1. Le système (1.3) est observable en temps T si et seulement si, pour le système réduit $\dot{x} = Ax, y = Cx, x(0) = x_0$, on a :

$$x_0 \neq 0 \Rightarrow y(\cdot) \neq 0 \text{ sur } [0, T]$$

Remarque 1.6. Pour un système linéaire autonome, l'observabilité en temps T implique l'observabilité pour tout temps.

Remarque 1.7. La notion d'observabilité pour un système linéaire autonome ne dépend pas de la matrice B .

Corollaire 1.1. Le système (1.3) est observable en temps T si et seulement si la matrice :

$$O(T) = \int_0^T e^{-sA^T} C^T C e^{-sA} ds$$

est inversible.

1.5.3 Observabilité des systèmes non linéaires

Dans le cas des systèmes non linéaires, étant donné l'espace d'état $X \subset \mathbb{R}^n$ et l'ensemble des entrées U , la notion d'observabilité repose sur la capacité à distinguer deux conditions initiales distinctes. On parle alors de **distinguabilité** d'un couple de conditions initiales.

La définition de l'observabilité pour les systèmes non linéaires est analogue à celle des systèmes linéaires.

1.6 Stabilité

Soit le système :

$$\dot{x} = f(x) \quad (1.4)$$

tel que $f(0) = 0$, admettant $x_e = 0$ comme équilibre (c'est-à-dire que, par un changement de variable, on peut toujours ramener l'équilibre à l'origine).

Définition 1.3. (Système stable)

Soit le système dynamique :

$$\dot{x} = f(x, u, t), \quad x \in M, u \in U. \quad (1.5)$$

Un équilibre $x_e = 0$ du système (1.5) est dit **stable** si, pour tout $\varepsilon > 0$, il existe $\eta > 0$ tel que, pour toute solution $x_s(t)$ du système, on ait :

$$\|x_s(0)\| < \eta \implies \forall t > 0, \|x_s(t)\| < \varepsilon$$

où $\|\cdot\|$ désigne la norme usuelle.

Définition 1.4. (Système attractif)

L'équilibre du système (1.5) est dit **attractif** s'il existe $r > 0$ tel que, pour toute solution $x_s(t)$ du système, on ait :

$$\|x_s(0)\| < r \implies \lim_{t \rightarrow \infty} x_s(t) = 0$$

Définition 1.5. (Système asymptotiquement stable)

L'équilibre $x_e = 0$ du système (1.5) est dit **asymptotiquement stable** s'il est à la fois stable et attractif.

Définition 1.6. (Stabilisation)

Le problème de **stabilisation** (ou régulation) consiste à maintenir un système près d'un équilibre x_e asymptotiquement stable en boucle fermée.

1.6.1 Stabilité des systèmes linéaires

Définition 1.7. (Bouclage d'état linéaire)

On appelle **bouclage d'état linéaire** (ou **régulateur linéaire**) du système

$$\dot{x} = Ax + Bu$$

une loi de commande du type :

$$u = Kx$$

où K est une matrice de gain de dimension $m \times n$.

Une telle loi est dite **stabilisante** si l'origine du système en boucle fermée

$$\dot{x} = (A + BK)x$$

est asymptotiquement stable.

1.6.2 Stabilisation locale d'un système non linéaire

Soit le système commandé :

$$\dot{x} = f(x, u), \quad f(x_0, u_0) = 0 \tag{1.6}$$

On pose $u = R(x)$ telle que le système en boucle fermée

$$\dot{x} = f(x, R(x))$$

admet l'origine comme équilibre asymptotiquement stable.

On considère l'approximation linéaire :

$$\dot{x} = Ax + Bu$$

avec :

$$A = \left. \frac{\partial f}{\partial x} \right|_{(0,0)}, \quad B = \left. \frac{\partial f}{\partial u} \right|_{(0,0)}$$

Théorème. [4]

Le contrôle $u = Kx$ stabilise localement le système (1.6). En effet, le système en boucle fermée s'écrit :

$$\dot{x} = F(x) = f(x, Kx)$$

On a :

$$\left. \frac{\partial F}{\partial x} \right|_0 = \left. \frac{\partial f}{\partial x} \right|_{(0,0)} + \left. \frac{\partial f}{\partial u} \right|_{(0,0)} K = A + BK$$

Ainsi, $x = 0$ est asymptotiquement stable pour le système linéarisé.

1.7 Principe du maximum de Pontryagin

Dans cette section, nous présentons le principe du maximum de **Pontryagin** [6], formulé en 1956 par le mathématicien russe *Lev Semenovitch Pontryagin*. Ce principe généralise les équations d'Euler-Lagrange issues du calcul des variations.

Il fournit une **condition nécessaire d'optimalité**. À l'origine, ce principe a été développé pour déterminer la trajectoire optimale en *temps minimal* d'une fusée se dirigeant vers la Lune.

On considère le problème de contrôle optimal suivant :

$$\left\{ \begin{array}{l} J(t^*, u) = g(t^*, x(t^*)) + \int_{t_0}^{t^*} f_0(t, x(t), u(t)) dt \rightarrow \min_u \quad (1) \\ \dot{x}(t) = f(t, x(t), u(t)) \quad (2) \\ x(t_0) = x_0 \in M_0 \quad (3) \\ x(t^*) = x_1 \in M_1 \quad (4) \\ u \in U, \quad t \in I = [t_0, t^*] \quad (5) \end{array} \right. \quad (1.7)$$

Avant d'énoncer le principe du maximum, introduisons quelques définitions et propriétés fondamentales.

Définition 1.8. Le contrôle u est dit **extrémal** sur l'intervalle $[t_0, t^*]$ si la trajectoire du système (2) associée à u , dans le cadre du problème de contrôle (1.7), vérifie :

$$x(t) \in \partial \text{Acc}(x_0, t^*), \quad \forall t \in [t_0, t^*].$$

Définition 1.9. Un contrôle $u_0(t)$, $t \in [t_0, t^*]$, est dit **optimal** s'il est **extrémal** et vérifie :

$$J(u_0(t)) < J(u(t))$$

pour tout autre contrôle extrémal (linéaire) $u(t)$, $t \in [t_0, t^*]$.

Théorème. [3]

Considérons le système :

$$\forall t \in I, \quad \dot{x}(t) = A(t)x(t) + B(t)u(t), \quad x(t_0) = x_0.$$

On suppose que le domaine de contraintes Ω est compact. Soit $t^* > 0$. Le contrôle u est extrémal sur $I = [t_0, t^*]$ si et seulement s'il existe une solution non triviale $p(t)$ de l'équation :

$$\dot{p}(t) = -p(t)A(t),$$

telle que, pour presque tout $t \in [t_0, t^*]$:

$$p(t)B(t)u(t) = \max_{u \in U} p(t)B(t)u. \quad (1.8)$$

Définition 1.10. Le vecteur $p(t) \in \mathbb{R}^n$ est appelé **vecteur adjoint**.

Définition 1.11. La fonction $\gamma(t) = p(t)B(t)$ est appelée **fonction de commutation**.

Définition 1.12. Le **temps de commutation** t_c est l'instant où le contrôle extrémal $u(t)$, défini sur $[t_0, t^*]$, change de signe.

Théorème. [3]

Ce théorème est l'énoncé général du principe du maximum de Pontryagin.

Considérons le système de contrôle dans \mathbb{R}^n :

$$\dot{x}(t) = f(t, x(t), u(t)) \quad (1.9)$$

où $f : \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$ est une fonction de classe C^1 .

Les contrôles $u(t)$ sont des applications mesurables bornées à valeurs dans un ensemble $U \subset \mathbb{R}^m$.

Soient M_0 et M_1 deux sous-ensembles de \mathbb{R}^n .

Notons par \mathcal{U} l'ensemble des contrôles admissibles u dont les trajectoires associées relient un point initial de M_0 à un point final de M_1 en un temps t .

On définit alors le coût associé à un contrôle u par :

$$J(t^*, u) = \int_{t_0}^{t^*} f_0(t, x(t), u(t)) dt + g(t^*, x(t^*))$$

où $f : \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$ et $g : \mathbb{R} \times \mathbb{R}^n \rightarrow \mathbb{R}$ de classe C^1 , $x(\cdot)$ est la solution de (1.9) associée au contrôle u .

On considère le problème de contrôle optimal suivant : déterminer une trajectoire reliant M_0 à M_1 en minimisant le coût J . Le temps final peut être fixé ou non. Si le contrôle $u \in \mathcal{U}$ associé à la trajectoire $x(\cdot)$ est optimal sur $[t_0, t^]$, alors il existe une application $p(\cdot) : [t_0, t^*] \rightarrow \mathbb{R}^n$ absolument continue, appelée vecteur adjoint, et un réel $p_0 \leq 0$ tel que le couple $(p(\cdot), p_0)$ est non trivial et tels que pour presque tout $t \in [t_0, t^*]$,*

$$\dot{x} = \frac{\partial H}{\partial p}(t, x(t), p(t), p_0, u(t)) \quad (1.10)$$

$$\dot{p} = -\frac{\partial H}{\partial x}(t, x(t), p(t), p_0, u(t)) \quad (1.11)$$

où $H(t, x, p, p_0, u) = p(t)f(t, x, u) + p_0 f_0(t, x, u)$ est le Hamiltonien du système, et on a la condition de maximisation presque partout sur $[t_0, t^*]$

$$H(t, x(t), p(t), p_0, u(t)) = \max_{u \in U} H(t, x(t), p(t), p_0, u) \quad (1.12)$$

Si de plus le temps final pour joindre M_1 n'est pas fixé, on a la condition au temps final t^*

$$\max_{u \in U} H(t^*, x(t^*), p(t^*), p_0, u) = -p_0 \frac{\partial g}{\partial t}(t^*, x(t^*)) \quad (1.13)$$

Si de plus M_0 et M_1 (ou juste l'un des deux ensembles) sont des variétés de \mathbb{R}^n ayant des espaces tangents en $x(t_0) = x_0 \in M_0$ et $x(t^*) = x_1 \in M_1$, alors le vecteur adjoint peut être construit de manière à vérifier les conditions de transversalité aux deux extrémités (ou juste l'une des deux)

$$p(0) \perp t_{x(0)}^* M_0 \quad (1.14)$$

$$p(t^*) - p_0 \frac{\partial g}{\partial x}(t^*, x(t^*)) \perp t_{x(t^*)}^* M_1 \quad (1.15)$$

Remarque 1.8. Si f et f_0 ne dépendent pas du temps t , c'est-à-dire si le système considéré est autonome, alors l'Hamiltonien H ne dépend pas de t , et on a pour tout $t \in [t_0, t^*]$:

$$\max_{u \in U} H(t, x(t), p(t), p_0, u) = \text{constante.}$$

Remarque 1.9. La convention $p_0 \leq 0$ conduit au principe du maximum, tandis que la condition $p_0 \geq 0$ conduira au principe du minimum.

Plusieurs travaux ont été réalisés dans ce sens. Le lecteur est invité à consulter les articles [5][6][7].

Remarque 1.10. Lorsqu'il n'y a pas de contraintes sur le contrôle, la condition de maximisation (1.12) devient :

$$\frac{\partial H}{\partial u} = 0.$$

Définition 1.13. Les conditions (1.14) et (1.15) sont appelées conditions de transversalité sur le vecteur adjoint.

1.8 Méthodes de résolution numérique

Les problèmes de contrôle optimal, en général, n'ont pas toujours de solutions analytiques. Par conséquent, il est nécessaire d'utiliser des méthodes numériques pour pouvoir les résoudre. Il existe différentes méthodes pour résoudre les problèmes de commande optimale, chacune avec ses avantages et ses inconvénients. Le choix de la méthode dépend du problème considéré. Ces méthodes transforment le problème de contrôle original en la résolution d'un système d'équations non linéaires. On distingue deux types de méthodes numériques en contrôle optimal : les méthodes directes et les méthodes indirectes.

1.8.1 Méthodes indirectes

Les méthodes indirectes sont basées sur le **principe du maximum de Pontryagin** [4][8], qui fournit une condition nécessaire d'optimalité. Il est nécessaire de vérifier *a posteriori* l'optimalité de la trajectoire calculée. Ces méthodes offrent une extrême précision numérique, mais elles sont très sensibles au choix de la condition initiale. Contrairement aux méthodes directes, les méthodes indirectes nécessitent une étude théorique préalable ainsi qu'une analyse des variables adjointes. Pour ces méthodes, la structure des commutations doit être connue à l'avance. Elles sont efficaces en toutes dimensions.

Il existe également des approches probabilistes. Ces méthodes consistent à exprimer le problème de commande optimale dans des espaces de mesure, puis à rechercher la commande optimale en tant que mesure d'occupation, qui est approchée par un nombre fini de ses moments. Cette méthode utilise des outils de géométrie algébrique et permet de réduire le problème de contrôle optimal à un problème d'optimisation de dimension finie.

Pour plus d'informations, les lecteurs sont invités à consulter l'article récent d'Emmanuel Trélat [3], *Contrôle optimal : théorie et applications*.

1.8.2 Méthodes directes

Parmi les méthodes directes, on trouve la méthode de résolution par l'approche de la programmation linéaire, qui est la méthode adaptée appelée aussi méthode du support ([9],[1],[10],[11],[12]). Elle permet d'avoir une solution approchée ou une solution exacte. Une autre méthode directe est la méthode de discrétisation du problème initial. Pour

un problème de départ linéaire, on fait une discrétisation de la commande. De là, on obtient un problème de programmation linéaire facile à résoudre. L'inconvénient de cette dernière approche est l'obtention d'une solution approchée. La mise en œuvre des méthodes directes est simple, car elles ne nécessitent pas une étude théorique préalable, on n'a pas à étudier les variables adjointes ou bien à connaître à l'avance la structure des commutations. Ces méthodes sont moins précises.

1.8.3 Exemple [3]

Considérons le problème du temps minimal

$$\dot{x}(t) = y(t); \quad x(0) = 0, \quad \dot{y}(t) = u(t); \quad y(0) = 0. \quad (1.16)$$

Le contrôle $u(t)$, $t \in I = [0; T]$ vérifie $|u(t)| \leq 1$. Résoudre le problème de temps minimal pour atteindre le point final $(0, -1)$, en allant de l'origine $(0, 0)$. Cherchons la solution théoriquement et puis avec les deux méthodes directe et indirecte.

L'Hamiltonien du système (1.16) est

$$H(x(t), y(t), p_x(t), p_y(t), u) = p_x(t)y(t) + p_y(t)u + p_0,$$

où p_x et p_y sont les composantes du vecteur adjoint. Elles sont solutions du système

$$\dot{p}_x(t) = -\frac{\partial H}{\partial x} = 0; \quad \dot{p}_y(t) = -\frac{\partial H}{\partial y} = -p_x(t).$$

Ce système est équivalent à

$$p_x(t) = \text{constante} = \gamma; \quad p_y(t) = -\gamma t + \mu.$$

Reprenons l'expression de l'Hamiltonien

$$H = p_x(t)y(t) + p_y(t)u + p_0.$$

Alors, quelle que soit la valeur de p_0 ,

$$\frac{\partial H}{\partial u} = \frac{\partial}{\partial u} (p_y(t)u + p_x(t)y(t));$$

par ailleurs, la condition de maximum du principe du maximum donne

$$u = \text{sign}(p_y(t)).$$

De là, le contrôle extrémal sera le suivant

$$u(t) = \begin{cases} -1 & \text{si } p_y(t) < 0, \\ +1 & \text{si } p_y(t) > 0, \end{cases} \quad t \in I.$$

Les contrôles extrémaux ont en plus une commutation. Soit t_c , ce temps de commutation. La trajectoire obtenue pour $u(t) = 1$ sur $[0, t_c]$ et pour $u(t) = -1$ sur $[t_c, T]$ est :

- Si $t \in [0, t_c]$, on obtient $y(t) = t$ et $x(t) = \frac{1}{2}t^2$.
- Si $t \in [t_c, T]$, on obtient $y(t) = -t + 2t_c$ et $x(t) = -\frac{1}{2}t^2 + 2t_c t - t_c^2$.

Les trajectoires obtenues en prenant $u = +1$ puis $u = -1$ sont illustrées dans la figure (1.3).

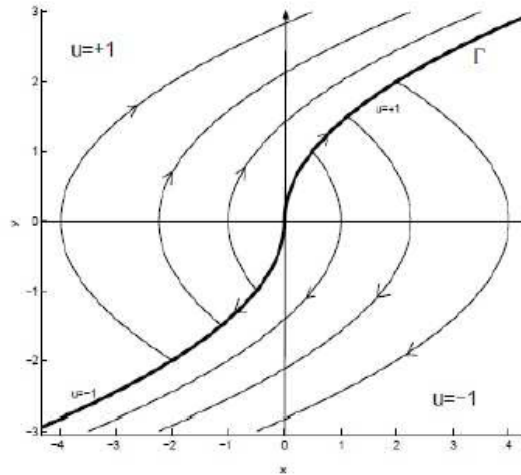


FIGURE 1.3 – Trajectoire optimale

Nous avons mis en œuvre une méthode indirecte, les résultats sont tracés dans la figure (1.4).

Dans le cas d'une méthode directe, on discrétise la commande $u(\cdot)$ et l'état $x(\cdot)$ en utilisant la subdivision :

$$0 = t_0 < t_1 < t_2 < \dots < t_N = T$$

Le problème (1.12) devient le problème suivant : On cherche à minimiser le temps final T , sous les contraintes suivantes :

$$x[i + 1] = x[i] + \frac{T}{N} \cdot y[i]$$

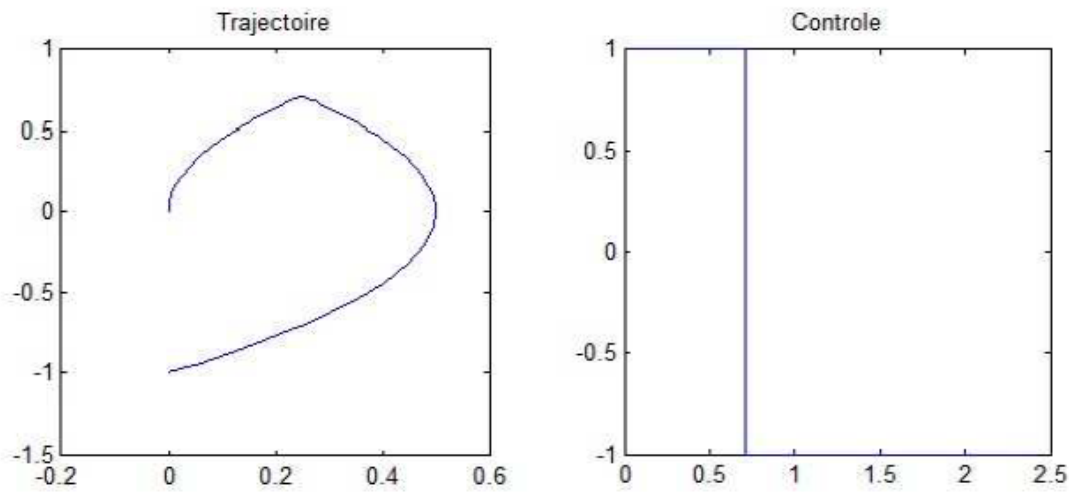


FIGURE 1.4 – Trajectoire et commande optimales par la méthode indirecte.

$$y[i + 1] = y[i] + \frac{T}{N} \cdot u[i]$$

Les résultats d'un tel problème sont tracés dans la figure (1.5).

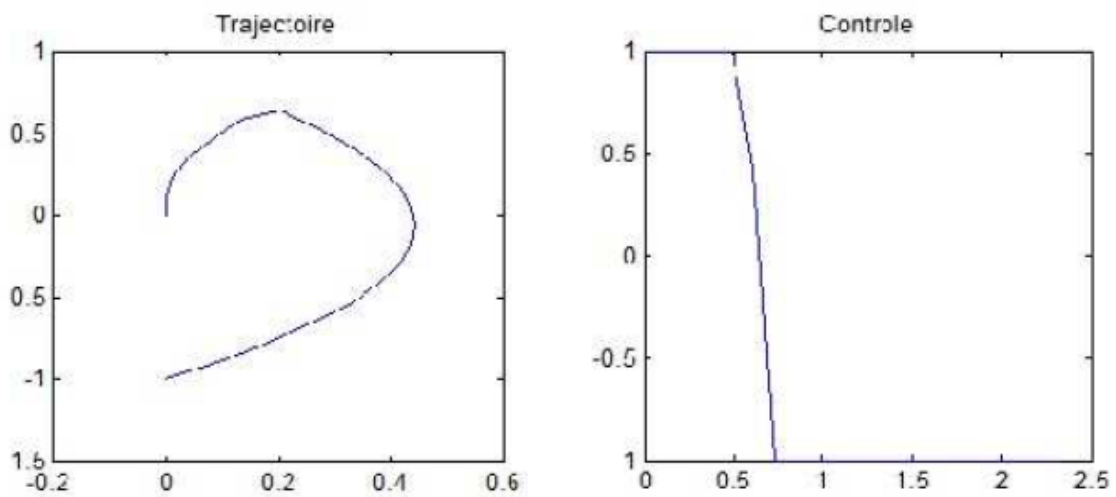


FIGURE 1.5 – Trajectoire et commande optimales par la méthode directe.

1.9 Conclusion

Dans ce chapitre, nous avons étudié la théorie du contrôle optimal en présentant ses principales notions, notamment la contrôlabilité, l'observabilité, la stabilité et le Principe du Maximum de Pontryagin. En nous appuyant sur plusieurs travaux de recherche déjà réalisés dans ce domaine, nous avons clarifié et rappelé certaines définitions, extensions et généralisations, ainsi que leurs paramètres fondamentaux et quelques résultats théoriques.

Chapitre 2

Développement et Application de la Méthode Adaptée

2.1 Introduction

Nous nous proposons de résoudre un problème de programmation linéaire en utilisant la méthode adaptée proposée par R. Gabasov et F. M. Kirillova [13].

Une étude comparative avec la méthode du Simplexe a également été réalisée.

2.2 Position du problème

Considérons le problème linéaire (P1) suivant :

$$(P_1) \quad \begin{cases} \text{Maximiser} & f(x) = c'x & (1) \\ Ax = b & & (2) \\ d_1 \leq x \leq d_2 & & (3) \end{cases}$$

Où x, c, d_1, d_2 sont des n -vecteurs réels, b un m -vecteur réel, $A = A[I, J]$ une matrice $m \times n$;

- $I = \{1, \dots, m\}$: l'ensemble des indices des lignes de A ,
- $J = \{1, \dots, n\}$: l'ensemble des indices des colonnes de A ,
- Rang $A = m$, $m \leq n$, c' est le transposé du vecteur c .

Définition 2.1. Tout vecteur vérifiant les contraintes (2) et (3) est dit plan du problème (P1).

- Un plan x_0 est optimal si $c'x_0 = \max c'x$
- Un plan x_ε est dit ε -optimal si $f(x_0) - f(x_\varepsilon) \leq \varepsilon$, où $\varepsilon > 0$ réel, donné.

Définition 2.2. L'ensemble des m indices $J_B \subseteq J$, $|J_B| = m$ est dit support (ou appui) du problème (P1) et la matrice $A_B = A(I, J_B)$ est la matrice de support (ou matrice d'appui) si $\det(A_B) \neq 0$.

Ainsi, en choisissant un support J_B , tout vecteur $x(J)$ peut s'écrire sous la forme $x(J) = (x(J_B), x(J_H))$, où :

- $x(J_B)$ est l'ensemble des composantes sur les indices du support,
- $x(J_H)$ est l'ensemble des composantes sur les indices hors support.

De la même manière, la matrice A peut être décomposée de la manière suivante :

$$A(I, J) = (A(I, J_B), A(I, J_H))$$

En utilisant cette dernière décomposition, le système $Ax = b$ prend la forme :

$$Ax = (A(I, J_B), A(I, J_H)) \cdot (x(J_B), x(J_H))$$

$$Ax = A(I, J_B) \cdot x(J_B) + A(I, J_H) \cdot x(J_H) = b$$

De là, comme A_B est inversible, on peut calculer les composantes x_B en fonction de x_H .

$$x_B = x(J_B) = A_B^{-1}(b - A_H \cdot x_H), \quad \text{où } A_H = A(I, J_H)$$

Définition 2.3. La paire $\{x, J_B\}$, formée du plan x et du support J_B , est appelée **support-plan** (ou **plan d'appui**) du problème (P1).

Définition 2.4. Un support-plan $\{x, J_B\}$ est dit **non dégénéré** si :

$$d_{1j} < x_j < d_{2j} \quad \text{pour tout } j \in J_B$$

2.3 Accroissement de la fonctionnelle — Critère d'optimalité et de suboptimalité

Soit $\{x, J_B\}$ un support-plan non dégénéré de départ. Construisons les vecteurs suivants :

$$y' = y'(I) = c'_B A_B^{-1}$$

$$\Delta = y' A(I, j) - c_j \quad \text{pour } j \in J$$

$$\Delta' = y' A - c'$$

où y' et Δ' sont appelés respectivement **vecteur des potentiels** et **vecteur des estimations**.

Remarque 2.1. Par construction, les composantes de support du vecteur Δ sont nulles :

$$\Delta_B = \Delta(J_B) = 0$$

Considérons un autre plan $\bar{x} = x + \Delta x$ et calculons la quantité définissant l'accroissement de la fonctionnelle :

$$\begin{aligned} \Delta f(x) &= f(\bar{x}) - f(x) = c'\bar{x} - c'x = c'\Delta x \\ &= c'(J_B)\Delta x(J_B) + c'(J_H)\Delta x(J_H) = c'_B\Delta x_B + c'_H\Delta x_H \end{aligned}$$

Comme $Ax = b$ et $A\bar{x} = b$, alors :

$$\begin{aligned} A\Delta x &= 0 \Rightarrow A_B\Delta x_B + A_H\Delta x_H = 0 \\ &\Rightarrow \Delta x_B = -A_B^{-1}A_H\Delta x_H \end{aligned}$$

En remplaçant Δx_B dans $\Delta f(x)$, on obtient :

$$\Delta f(x) = (-c'_B A_B^{-1} A_H + c'_H) \Delta x_H = -\Delta'_H \Delta x_H = \sum_{j \in J_H} \Delta_j \Delta x_j \quad (4)$$

Comme \bar{x} est un plan admissible, alors l'accroissement Δx vérifie :

$$d_{1j} - x_j \leq \Delta x_j \leq d_{2j} - x_j, \quad \forall j \in J \quad (5)$$

Le maximum de l'accroissement de la fonctionnelle (4) sous les contraintes (5) est atteint pour :

$$\begin{cases} \Delta x_j = d_{1j} - x_j & \text{si } \Delta_j > 0 \\ \Delta x_j = d_{2j} - x_j & \text{si } \Delta_j < 0 \\ d_{1j} - x_j \leq \Delta x_j \leq d_{2j} - x_j & \text{si } \Delta_j = 0 \end{cases} \quad \text{pour tout } j \in J_H$$

et il est égal à :

$$\beta = \beta(x, J_B) = \sum_{j \in J_H^+} \Delta_j (x_j - d_{1j}) + \sum_{j \in J_H^-} \Delta_j (x_j - d_{2j})$$

appelée **valeur de suboptimalité**, où :

$$J_H^+ = \{j \in J_H \mid \Delta_j \geq 0\}, \quad J_H^- = \{j \in J_H \mid \Delta_j \leq 0\}$$

Il en résulte que :

$$\Delta f(x) = f(\bar{x}) - f(x) \leq \beta(x, J_B)$$

et donc pour $\bar{x}=x_0$, on aura :

$$f(x_0) - f(x) \leq \beta(x, J_B)$$

De cette dernière inégalité, on déduit le critère suivant :

Théorème. (*Critère d'optimalité*)

Les relations :

$$\begin{cases} x_j = d_{1j} & \text{si } \Delta_j > 0 \\ x_j = d_{2j} & \text{si } \Delta_j < 0 \\ d_{1j} \leq x_j \leq d_{2j} & \text{si } \Delta_j = 0 \end{cases} \quad \text{pour tout } j \in J_H \quad (6)$$

sont suffisantes, et dans le cas de la non-dégénérescence, elles sont également nécessaires pour l'optimalité du support-plan $\{x, J_B\}$.

Preuve (Condition suffisante) : Si les relations (6) sont vérifiées, alors $\beta(x, J_B) = 0$.
Et comme

$$\Delta f(x) = f(\bar{x}) - f(x) \leq \beta(x, J_B) = 0 \quad \Rightarrow \quad f(\bar{x}) \leq f(x), \quad \forall \bar{x} \Rightarrow x \text{ est optimal}$$

Condition nécessaire

Soit $\{x, J_B\}$ un support plan optimal non dégénéré, et supposons que les relations (6) ne sont pas vérifiées, c'est-à-dire qu'il existe un indice $j_0 \in J_H$ tel que :

$$\Delta_{j_0} > 0 \quad \text{et} \quad x_{j_0} > d_{1j_0}, \quad \text{ou bien} \quad \Delta_{j_0} < 0 \quad \text{et} \quad x_{j_0} < d_{2j_0}.$$

Prenons par exemple le second cas : $\Delta_{j_0} < 0$ et $x_{j_0} < d_{2j_0}$.

Construisons un nouveau plan \bar{x} de la manière suivante :

$$\bar{x} = x + \Delta x = x + \theta.l,$$

où θ est un réel strictement positif, et l est un vecteur (direction).

Il faut trouver l et θ tels que :

$$A\bar{x} = b, \quad d_1 \leq \bar{x} \leq d_2$$

Pour cela, sur J_H , posons :

$$\Delta x = \begin{cases} 0 & \text{si } j \in J_H \setminus \{j_0\}, \\ \theta & \text{si } j = j_0, \end{cases}$$

Avec $\theta > 0$, on a :

$$\Delta x(J_B) = -A_B^{-1} A_H \Delta x(J_H) = -\theta A_B^{-1} a_{j_0},$$

ce qui vérifie bien $Ax = b$.

Pour que \bar{x} vérifie $d_1 \leq \bar{x} \leq d_2$, il faut prendre un θ suffisamment petit, d'autant plus que le support plan $\{x, J_B\}$ est non dégénéré.

En portant $\bar{x} = x + \Delta x = x + \theta l$ dans la formule d'accroissement, on obtient :

$$\Delta f(x) = f(\bar{x}) - f(x) = -\theta \Delta_{j_0} l_{j_0} > 0$$

ce qui contredit l'optimalité du plan $\{x, J_B\}$.

Théorème. (*Critère de suboptimalité*)

Soit $\varepsilon > 0$ donné. Pour que le plan x soit ε -optimal, il suffit de trouver un support J_B tel que la valeur de suboptimalité vérifie l'inégalité suivante :

$$\beta(x, J_B) \leq \varepsilon.$$

Preuve : Condition suffisante. Si $\beta(x, J_B) \leq \varepsilon$, alors :

$$\Delta f(x) \leq \varepsilon \Rightarrow x \text{ est } \varepsilon\text{-optimal},$$

ce qui permet d'obtenir le résultat escompté.

À présent, faisons une décomposition de $\beta(x, J_B)$.

Pour cela, construisons le problème dual de (P1) :

$$\Omega(\lambda) = b'u - d'_1 v + d'_2 w \longrightarrow \min$$

$$A'u - v + w = c, \quad v \geq 0; \quad w \geq 0$$

Il est facile de vérifier que le vecteur $\lambda(u, v, w)$ défini de la manière suivante :

$$u = y; \quad v_j = \begin{cases} \Delta_j, & \text{si } \Delta_j \geq 0, \\ 0, & \text{si } \Delta_j \leq 0, \end{cases} \quad w_j = \begin{cases} 0, & \text{si } \Delta_j \geq 0, \\ -\Delta_j, & \text{si } \Delta_j \leq 0, \end{cases} \quad \text{pour tout } j \in J.$$

Ceci définit un plan dual. On a :

$$\beta = \beta(x, J_B) = \sum_{j \in J} \Delta_f x_j - \sum_{j \in j_H^-} \Delta_f d_{2j} - \sum_{j \in j_H^+} \Delta_f d_{1j}$$

En introduisant le plan dual défini ci-dessus, on obtient :

$$\begin{aligned}\beta(x, J_B) &= \Delta'x - d'_1v + d'_2w = y'Ax - c'x - d'_1v + d'_2w \\ &= b'y - c'x - d'_1v + d'_2w + c'x_0 - c'x_0 = (c'x_0 - c'x) + (b'y - d'_1v + d'_2w - c'x_0) \\ &= (c'x_0 - c'x) + (\Omega(\lambda) - \Omega(\lambda_0))\end{aligned}$$

Donc :

$$\beta(x, J_B) = \beta_x + \beta_B,$$

où :

$$\beta_x = (c'x_0 - c'x) \quad \text{est appelé l'écart de non-optimalité du plan } x,$$

$$\beta_B = (\Omega(\lambda) - \Omega(\lambda_0)) \quad \text{est l'écart de non-optimalité du support } A_B(J_B).$$

Remarque 2.2. À partir de l'expression $\beta(x, J_B) = \beta_x + \beta_B$, on conclut que l'amélioration du support plan $\{x, A_B\}$ peut se faire indépendamment l'un de l'autre.

Si $\beta(x, J_B) > \varepsilon$, alors on procède à un changement du support plan $\{x, J_B\}$.

2.4 Itération de l'algorithme

La méthode de résolution est constituée de deux procédures :

- **Changement de plan** : consiste à augmenter $c'x$,
- **Changement du support** : consiste à diminuer $\Omega(\lambda)$.

2.4.1 Changement de plan

Le nouveau plan \bar{x} sera construit de la manière suivante :

$$\bar{x} = x + \theta l,$$

où l est une direction admissible, et θ (un réel positif) est le pas admissible maximal le long de la direction l , tel que $f(\bar{x}) \geq f(x)$.

Le vecteur de direction

$$l = (l(J_B), l(J_H))$$

est construit de la manière suivante :

— Sur J_H , on pose $\theta = 1$, et :

$$l_j = \begin{cases} d_{1j} - x_j & \text{si } \Delta_j > 0, \\ d_{2j} - x_j & \text{si } \Delta_j < 0, \\ 0 & \text{si } \Delta_j = 0, \end{cases} \quad \text{pour } j \in J_H, \quad (7)$$

Et $l(J_B) = -A_B^{-1}A_H.l(j_H)$ pour avoir $A\bar{x} = b$. Pour que \bar{x} vérifie $d_1 \leq \bar{x} \leq d_2$, il faut calculer :

$$\theta_j = \begin{cases} \frac{d_{1j} - x_j}{l_j} & \text{si } l_j < 0, \\ \frac{d_{2j} - x_j}{l_j} & \text{si } l_j > 0, \\ +\infty & \text{si } l_j = 0, \end{cases} \quad \text{pour } j \in J_B.$$

$$\theta_{j_0} = \min(\theta_j) \quad \text{pour } j \in J_B$$

et le pas maximal sera $\theta_0 = \min(1, \theta_{j_0})$

Dès lors, le nouveau plan est :

$$\bar{x} = x + \theta_0 l,$$

et la valeur de suboptimalité pour ce nouveau plan est :

$$\begin{aligned} \beta(\bar{x}, J_B) &= \sum_{j \in J_H^+} \Delta_j (x_j - d_{1j}) + \sum_{j \in J_H^-} \Delta_j (x_j - d_{2j}) \\ &= \beta(x, J_B) + \theta_0 \sum_{j \in J_H} \Delta_j \cdot l_j \quad (\text{en remplaçant les } l_j \text{ donnés par (7)}) \\ &= \beta(x, J_B) - \theta_0 \beta(x, J_B) \\ &= (1 - \theta_0) \cdot \beta(x, J_B) \end{aligned}$$

De cette dernière expression, on conclut :

- Si $\theta_0 = 1$, alors \bar{x} est optimal.
- Si $\beta(\bar{x}, J_B) < \varepsilon$, alors \bar{x} est ε -optimal.
- Si $\beta(\bar{x}, J_B) > \varepsilon$, on procède à un changement du support : $J_B \rightarrow \bar{J}_B$.

2.4.2 Changement du support

Le changement du support $J_B \rightarrow \bar{J}_B$ consiste à effectuer un changement du coplan Δ vers $\bar{\Delta}$ et du vecteur des potentiels y vers \bar{y} de telle sorte que :

$$\beta(\bar{x}, \bar{J}_B) \leq \beta(\bar{x}, J_B).$$

Pour cela, on pose :

$$\bar{\Delta}(J) = \Delta(J) + \sigma_0 t(J) \in \mathbb{R}^n \quad (8)$$

$$\bar{y}(I) = y(I) + \sigma_0 t(I) \in \mathbb{R}^m \quad (9)$$

Où t est la direction de diminution de la fonction duale, σ_0 le pas maximal le long de cette direction.

Calcul de t et σ_0 :

En utilisant la définition de Δ et y , on obtient :

$$\bar{\Delta} = \bar{y}'A - c' = (y' + \sigma_0 t'(I))A - c' = \Delta' + \sigma_0 t'(I)A$$

De là,

$$t'(J) = t'(I)A(I, J) \Rightarrow t'(J_B) = t'(I)A(I, J_B) \Rightarrow t'(I) = t'(J_B)A_B^{-1}$$

Ce qui donne :

$$t'(J_H) = t'(J_B)A_B^{-1}(I, J_H)$$

Après calcul du plan $\bar{x} = x + \theta_0 l$, le pas θ_0 est donné par :

$$\theta_0 = \min(1, \theta_{j_0}) = \theta_{j_0}, \quad j_0 \in J_B$$

On cherchera un indice $j_1 \in J_H$ qui va entrer dans la base à la place de j_0 . Pour cela, posons :

$$t_j = \begin{cases} -\text{signe}(l_{j_0}) & \text{si } j = j_0 \\ 0 & \text{si } j \in J_B \setminus \{j_0\} \end{cases}$$

$$t'(J_H) = t'(J_B)A_B^{-1}A(I, J_H)$$

Et calculons :

$$\sigma_0 = \sigma_{j_1} = \min_{j \in J_H}(\sigma_j)$$

Avec :

$$\sigma_j = \begin{cases} -\frac{\Delta_j}{t_j} & \text{si } \Delta_j t_j < 0 \\ 0 & \text{si } \Delta_j = 0, x_j \neq d_{1j}, t_j > 0 \text{ ou } \Delta_j = 0, x_j \neq d_{2j}, t_j < 0 \\ \infty & \text{dans les autres cas} \end{cases} \quad \text{pour } j \in J_H$$

$$\bar{\Delta}(j_1) = 0$$

Le nouveau support est :

$$\bar{J}_B = (J_B \setminus \{j_0\}) \cup \{j_1\}$$

On peut facilement remarquer que la quantité $\beta(\bar{x}, \bar{J}_B)$ est égale à :

$$\beta(\bar{x}, \bar{J}_B) = \sum_{j \in J_H^+} \bar{\Delta}_j(\bar{x}_j - d_{1j}) + \sum_{j \in J_H^-} \bar{\Delta}_j(\bar{x}_j - d_{2j})$$

où :

$$J_H^+ = \{j \in J_H \mid \bar{\Delta}_j \geq 0\}, \quad J_H^- = \{j \in J_H \mid \bar{\Delta}_j \leq 0\}$$

En utilisant la relation (8) sur J_B , on a :

$$\beta(\bar{x}, \bar{J}_B) = \sum_{j \in J_H^+} \Delta_j(\bar{x}_j - d_{1j}) + \sum_{j \in J_H^-} \Delta_j(\bar{x}_j - d_{2j}) + \sigma_0 \left(\sum_{j \in J_H^+} t_j(\bar{x}_j - d_{1j}) + \sum_{j \in J_H^-} t_j(\bar{x}_j - d_{2j}) \right)$$

Donc :

$$\beta(\bar{x}, \bar{J}_B) = (1 - \theta_0)\beta(x, J_B) + \sigma_0 \left(\sum_{j \in J_H^+} t_j(\bar{x}_j - d_{1j}) + \sum_{j \in J_H^-} t_j(\bar{x}_j - d_{2j}) \right)$$

Or $tl = 0$ car $Al = 0$ et $t'(J_B) = t'(I)A(I, J_B)$ et $t'(J_H) = t'(J_B)A_B^{-1}A(I, J_H)$.

De plus, par construction, toutes les composantes de $t'(J_B)$ sont nulles sauf à l'indice j_0 .

Posons :

$$\alpha = \alpha_0 = \sum_{j \in J_H^+} t_j(x_j - d_{1j}) + \sum_{j \in J_H^-} t_j(x_j - d_{2j}) = -(1 - \theta_0) \sum_{j \in J_H} t_j l_j$$

$$\alpha_0 = (1 - \theta_0)t_{j_0}l_{j_0}$$

$$\alpha = \alpha_0 = (1 - \theta_0) t_{j_0} l_{j_0} = \begin{cases} x_{j_0} + l_j - d_{1j_0} & \text{si } t_{j_0} = 1 \\ -(x_{j_0} + l_{j_0} - d_{2j_0}) & \text{si } t_{j_0} = -1 \end{cases}$$

Donc :

$$\beta(\bar{x}, \bar{J}_B) = (1 - \theta_0)\beta(x, J_B) - \sigma_0|\alpha_0|$$

Remarque 2.3. L'expression de la vitesse initiale de décroissance de la fonctionnelle Ω (du programme dual du programme initial) est donnée par :

$$\alpha_0 = \lim_{\sigma \rightarrow 0} \frac{\Omega(\bar{\lambda}) - \Omega(\lambda)}{\sigma}$$

avec

$$\bar{\lambda} = \lambda + \sigma \partial \lambda$$

Cette expression montre que la vitesse de décroissance est α_0 .

2.5 Algorithme de la méthode

0. Soit $\{x, J_B\}$ un support plan de départ.

I. Calculer :

- $y' = c'(J_B)A_B^{-1}$
- $\Delta' = y'A - c'$
- β
- Si $\beta = 0$, $\{x, J_B\}$ est optimal, arrêt du processus.
- Si $\beta \leq \varepsilon$, $\{x, J_B\}$ est ε -optimal, arrêt du processus.
- Si $\beta > \varepsilon$, aller à II.

II. — Déterminer le vecteur $l(J)$,

- Déterminer le vecteur $\bar{x}(J)$,
- Calculer $(1 - \theta_0)\beta$,
- Si $(1 - \theta_0)\beta > \varepsilon$, aller à III.
- Si $(1 - \theta_0)\beta < \varepsilon$, $\{\bar{x}, J_B\}$ est ε -optimal, arrêt du processus.
- Si $\theta_0 = 1$, $\{\bar{x}, J_B\}$ est optimal, arrêt du processus.

III. **Changement de support**

- Calculer le vecteur t ,
- Calculer σ_{j_1} tel que $\sigma_{j_1} = \min_{j \in J_H}(\sigma_j)$,

- Le nouvel appui : $\bar{J}_B = (J_B \setminus \{j_0\}) \cup \{j_1\}$,
- Aller à I (on passe à une nouvelle itération avec $\{\bar{x}, \bar{J}_B\}$).

2.6 Convergence de la méthode adaptée

Définition 2.5. On dit qu'un support plan $\{x, J_B\}$ est *très dégénéré* si la valeur de la fonctionnelle pour ce support plan n'augmente pas en appliquant l'algorithme ci-dessus.

Théorème. [13]

L'algorithme de la méthode adaptée pour la résolution du problème (P1) est fini, si à chaque itération on a des supports plans non dégénérés.

Preuve

L'algorithme ci-dessus est **fini** si, à chaque itération, on n'obtient pas un support-plan très dégénéré.

(Se référer à la référence mentionnée précédemment pour une démonstration détaillée.)

2.7 Exemples

Nous allons résoudre le problème suivant (I) à l'aide de la méthode du simplexe et d'une méthode adaptée :

Problème (I)

$$\text{Maximiser } x_1 + 2x_2 + 3x_3 + 4x_4 + 5x_5 + 6x_6$$

sous les contraintes :

$$\begin{array}{ll} 5x_1 + 3x_2 + 2x_3 + 4x_4 + 5x_5 + x_6 = 2 & \text{avec } -2 \leq x_1 \leq 2 \\ -x_2 + 2x_3 + x_4 + 4x_5 + 5x_6 = 2 & -3 \leq x_2 \leq 3 \\ 2x_1 + 4x_2 + 4x_3 + 5x_4 + x_5 + 5x_6 = 3 & -4 \leq x_3 \leq 4 \\ 6x_1 + 5x_2 + 4x_3 + 3x_4 + 2x_5 + x_6 = 4 & -5 \leq x_4 \leq 5 \\ & -6 \leq x_5 \leq 6 \\ & -7 \leq x_6 \leq 7 \end{array}$$

Avec :

$$x' = (x_1, x_2, x_3, x_4, x_5, x_6), \quad d_1 = (-2, -3, -4, -5, -6, -7), \quad d_2 = (2, 3, 4, 5, 6, 7)$$

$$c' = (1, 2, 3, 4, 5, 6), \quad b' = (2, 2, 3, 4)$$

$$A = \begin{pmatrix} 5 & 3 & 2 & 4 & 5 & 1 \\ -1 & 2 & 1 & 4 & 5 & 0 \\ 2 & 4 & 4 & 5 & 1 & 5 \\ 6 & 5 & 4 & 3 & 2 & 1 \end{pmatrix}$$

Posons le changement de variable : $y = x - d_1$, avec $y \geq 0$ et $y \leq d_2 - d_1$.

Résolution par la méthode du simplexe

Réécrivons le problème sous forme canonique en ajoutant des variables d'écart :

$$\text{Maximiser } y_1 + 2y_2 + 3y_3 + 4y_4 + 5y_5 + 6y_6 - 112$$

sous les contraintes :

$$5y_1 + 3y_2 + 2y_3 + 4y_4 + 5y_5 + y_6 = 86$$

$$-y_2 + 2y_3 + y_4 + 4y_5 + 5y_6 = 71$$

$$2y_1 + 4y_2 + 4y_3 + 5y_4 + y_5 + 5y_6 = 101$$

$$6y_1 + 5y_2 + 4y_3 + 3y_4 + 2y_5 + y_6 = 81 \tag{II}$$

Avec les contraintes supplémentaires issues des bornes :

$$y_1 + y_7 = 4$$

$$y_2 + y_8 = 6$$

$$y_3 + y_9 = 8$$

$$y_4 + y_{10} = 10$$

$$y_5 + y_{11} = 12$$

$$y_6 + y_{12} = 14$$

Et $y_i \geq 0$ pour $i = 1, \dots, 12$.

Donc, le problème (II) est équivalent au problème (I), écrit sous forme matricielle :

$$c'\chi + \alpha \rightarrow \max$$

$$A\chi = \gamma, \quad y_i \geq 0 \quad \text{pour } i = 1, \dots, 12$$

$$\chi' = (y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8, y_9, y_{10}, y_{11}, y_{12})$$

On prendra comme solution initiale de départ :

$$\chi' = (2,1662, \underline{0}, 8, 6,5675, 5,5315, 5,2612, 1,8378, 6, \underline{0}, 3,4324, 6,4884, 8,7387)$$

(Les composantes hors-base sont soulignées).

$$\aleph = \begin{pmatrix} 5 & 3 & 2 & 4 & 5 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 2 & 1 & 4 & 5 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 4 & 4 & 5 & 1 & 5 & 0 & 0 & 0 & 0 & 0 & 0 \\ 6 & 5 & 4 & 3 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$c' = [1, 2, 3, 4, 5, 6, 0, 0, 0, 0, 0, 0] \quad \text{et} \quad \gamma' = [86, 71, 101, 81, 4, 6, 8, 10, 12, 14]$$

Initialisation

On note alors que \aleph , c' , γ' deviennent respectivement \aleph_1 , c'_1 , γ'_1 :

$$\aleph_1 = \begin{pmatrix} 1 & 0.7702 & 0 & 0 & 0 & 0 & 0 & 0 & -0.7432 & 0 & 0 & 0 \\ 0 & 7.2164e-16 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0.4459 & 0 & 1 & 0 & 0 & 0 & 0 & 0.1486 & 0 & 0 & 0 \\ 0 & -0.5585 & 0 & 0 & 1 & 0 & 0 & 0 & 0.3693 & 0 & 0 & 0 \\ 0 & 0.1576 & 0 & 0 & 0 & 1 & 0 & 0 & -0.7252 & 0 & 0 & 0 \\ 0 & -0.7702 & 0 & 0 & 0 & 0 & 1 & 0 & 0.7432 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -0.4459 & 0 & 0 & 0 & 0 & 0 & 0 & -0.1486 & 1 & 0 & 0 \\ 0 & 0.5585 & 0 & 0 & 0 & 0 & 0 & 0 & -0.3693 & 0 & 1 & 0 \\ 0 & -0.1576 & 0 & 0 & 0 & 0 & 0 & 0 & 0.7252 & 0 & 0 & 1 \end{pmatrix}$$

$$c'_1 = [0, \underline{1,2927}, 0, 0, 0, 0, 0, 0, \underline{-0,3468}, 0, 0, 0]$$

$$\gamma'1 = [2,1621 \quad 8 \quad 6,5675 \quad 5,5331 \quad 5,2612 \quad 1,8378 \quad 6 \quad 3,4324 \quad 6,4684 \quad 8,7387]$$

1° ITERATION

Après pivotage , \aleph_1 , ζ'_1 , γ'_1 deviennent \aleph_2 , ζ'_2 , γ'_2 respectivement

$$\aleph_2 = \begin{pmatrix} 1.2982 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -0.9649 & 0 & 0 & 0 \\ -9.3687e-16 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ -0.5789 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0.5789 & 0 & 0 & 0 \\ 0.7251 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -0.1695 & 0 & 0 & 0 \\ -0.2046 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -0.5730 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 4.1362e-17 & 0 & 0 & 0 \\ -1.2982 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0.9649 & 0 & 0 & 0 \\ 0.5789 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.5789 & 1 & 0 & 0 \\ -0.7259 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.1695 & 0 & 1 & 0 \\ 0.2046 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5730 & 0 & 0 & 1 \end{pmatrix}$$

$$\zeta'_2 = [\underline{-1,6783} \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad \underline{0,9005} \quad 0 \quad 0 \quad 0]$$

$$\gamma'_2 = [2,870 \quad 8 \quad 5,3157 \quad 7,0994 \quad 4,8187 \quad 4 \quad 3,1929 \quad 4,6842 \quad 4,9005 \quad 9,1812]$$

2° ITERATION

Après pivotage , \aleph_2 , ζ'_2 , γ'_2 deviennent \aleph_3 , ζ'_3 , γ'_3 respectivement

$$\aleph_3 = \begin{pmatrix} 4.8030e-17 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1.3454 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1.0363 & 0 & 0 & 0 & 0 \\ 0.1999 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -0.5999 & 0 & 0 & 0 & 0 \\ 0.4969 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0.1757 & 0 & 0 & 0 & 0 \\ -0.9757 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0.5939 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -4.2866e-17 & 0 & 0 & 0 & 0 \\ -1.3454 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.0363 & 1 & 0 & 0 & 0 \\ -0.2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.6 & 0 & 1 & 0 & 0 \\ -0.4969 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.1757 & 0 & 0 & 1 & 0 \\ 0.9757 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.5939 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\zeta'_3 = [\underline{-0,4666} \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad \underline{-0,9333} \quad 0 \quad 0 \quad 0 \quad 0]$$

$$\gamma'_3 = [6 \quad 4,6909 \quad 3,4 \quad 7,6606 \quad 6,7151 \quad 4 \quad 3,3090 \quad 6,6 \quad 4,3393 \quad 7,2848].$$

le vecteur solution χ (optimal) est :-

$$\chi' = [064,69093,47,66066,7151403,3096,64,33937,2848]$$

et la valeur de la fonctionnelle 6,266 .

Appliquons maintenant la méthode adaptée pour ce même exemple :-

Nous prenons comme solution initiale le vecteur x défini comme suit :-

$$x = \begin{pmatrix} 0,220779 \\ -0,16888 \\ 0,98701 \\ -0,14285 \\ 0 \\ 0 \end{pmatrix} \quad J_B = \{1, 2, 3, 4\}$$

1° Itération :

Δ	x	θ	ℓ	\bar{x}	t
0	0,220779	0,4441	-5	-0,5714	0
0	-0,1688	0,15844	20	3	-1
0	0,98701	0,3562	-14	-1,2311	0
0	0,14285	0,4415	-11	-1,8857	0
-2,5454	0	1	6	0,9506	2,09
-0,8181	0	1	7	1,109	1,064

$$\beta = \beta_1 = 21$$

$$(1 - \theta^0)\beta = \beta_2 = 17,6727$$

$$\theta^0 = \min(\theta_1, \theta_2, \theta_3, \theta_4) = \theta_2 \quad ; \quad \sigma_0 = \min(\sigma_5, \sigma_6) = \sigma_6$$

$$\text{donc } \overline{J_B} = \{1, 3, 4, 6\}$$

2° Itération :

Δ	x	θ	ℓ	\bar{x}	t
0	-0,571	0,14	-10,46	-2	1
-0,7682	3	1	4,44e-16	3	-0,3536
0	-1,231	0,382	13,67	0,69	0
0	-0,1885	3,3885	2,032	-1,6	0
-0,939	0,9506	1	5,04	1,66	2,0121
0	1,1090	0,8179	-9,91	-0,2848	0

$$\beta = \beta_1 = 4,74$$

$$(1 - \theta^0)\beta = \beta_2 = 4,074$$

$$\theta^0 = \min(\theta_1, \theta_3, \theta_4, \theta_6) = \theta_1 \quad ; \quad \sigma_0 = \min(\sigma_2, \sigma_5) = \sigma_5$$

$$\overline{J_B} = \{3, 4, 5, 6\}$$

3° Itération :

Δ	x
0,4666	-2
-0,9333	3
0	0,6909
0	-1,6
0	1,66
0	-0,284

$$J_B = \{ 3, 4, 5, 6 \}; \beta_1 = 0$$

la valeur de la fonctionnelle 6,266

2.8 Etude comparative entre la méthode adaptée et le simplexe

1/ Les deux méthodes possèdent des algorithmes de résolution finis

2/ Au vu de la résolution du problème (P1) par la méthode du simplexe, on a constaté à travers l'exemple que cela occasionne l'ajout de variables supplémentaires et d'équations, ce qui n'est pas le cas avec l'utilisation de la méthode adaptée.

3/ Sachant que les solutions réalisables sont des sommets du polyèdre décrit par les contraintes de plus la recherche de la solution optimale dans le simplexe s'effectue par saut sur les sommets du polyèdre alors que la méthode adaptée prend les points à l'intérieur du polyèdre ce qui fait que cette méthode convergera vraisemblablement plus rapidement vers la solution optimale .

4/ Avec la méthode adaptée, on peut trouver une solution ε -optimale.

2.9 Conclusion

Ce chapitre a permis de poser clairement la structure d'un problème d'optimisation linéaire. Cette base est indispensable pour pouvoir ensuite appliquer une méthode de résolution adaptée.

Chapitre 3

Programmation multi-objectif

3.1 Introduction

La vie réelle regorge de problèmes nécessitant des solutions adaptées. Or, la majorité de ces problèmes ne possèdent pas de solutions universellement satisfaisantes selon un critère bien défini. En effet, de nombreux problèmes d'optimisation réels impliquent plusieurs objectifs ou critères, souvent contradictoires et parfois complémentaires, qu'il convient d'optimiser simultanément. Dans le cas des problèmes à objectif unique, la solution optimale est généralement bien définie. En revanche, dans le cadre de l'optimisation multi-objectif, cette notion doit être élargie pour prendre en compte la diversité des objectifs à atteindre.

Prenons l'exemple d'un individu souhaitant acheter une maison. L'habitation idéale serait à la fois peu coûteuse, spacieuse et bien située. Cependant, une telle maison idyllique est rarement accessible. L'acheteur devra alors rechercher un compromis entre ses différentes attentes, en fonction de ses priorités et de son budget.

D'autres situations illustrent également la nature multi-objectif de nombreux problèmes réels. Par exemple, l'élaboration d'un emploi du temps scolaire représente un problème typique d'optimisation multi-objectif, puisqu'il faut simultanément prendre en compte divers critères : volume horaire à enseigner, disponibilité des salles, charge horaire des enseignants, répartition des matières, etc.

Traditionnellement, les problèmes multi-objectifs étaient transformés en problèmes à objectif unique en combinant plusieurs critères en une seule fonction scalaire. Cependant, cette approche peut entraîner une perte d'informations importantes sur la structure du problème initial. C'est pourquoi, au cours des dernières années, de nombreuses métaheuristiques dédiées à l'optimisation multi-objectif ont vu le jour. Leur

objectif est de générer, en une seule exécution, un ensemble de solutions de compromis (appelé front de Pareto), sans réduire le problème à un modèle mono-objectif.

Ces nouvelles approches ont démontré une efficacité remarquable dans la résolution de problèmes réels à objectifs multiples. Toutefois, il convient de noter que les problèmes multi-objectifs sont généralement plus complexes à résoudre que leurs équivalents à objectif unique, principalement en raison de l'absence d'une relation d'ordre total entre les solutions, rendant la comparaison directe difficile.

Définition 3.1. Un problème d'optimisation multi-objectif (ou *Multiobjective Optimization Problem* (MOP)), également appelé problème multicritère, consiste à maximiser (ou à minimiser) simultanément plusieurs fonctions "objectif". Ce type de problème peut être formulé comme suit :

$$f(x) = F(x) = (f_1(x), f_2(x), \dots, f_k(x)) \rightarrow \max / \min$$

sous les contraintes :

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

$$h_j(x) = c_j, \quad \text{pour } j = 1 \dots q$$

$$x \in \mathbb{R}^n$$

L'ensemble des solutions réalisables, noté D , est défini par :

$$D = \{x \in \mathbb{R}^n \mid g_i(x) \geq b_i \forall i = 1 \dots p \text{ et } h_j(x) = c_j \forall j = 1 \dots q\}$$

Une solution d'un MOP correspond à un vecteur de décision $x = (x_1, x_2, \dots, x_n)'$ permettant de minimiser (ou de maximiser) les composantes du vecteur $F(x)$, tout en respectant les contraintes définissant l'ensemble des solutions admissibles D .

Dans ce travail, on se limite aux problèmes de **minimisation**, car tout problème de **maximisation** peut être aisément transformé en problème de minimisation grâce à la relation suivante :

$$\min F(x) = - \max (-F(x))$$

Définition 3.2. (Problème d'optimisation multi-objectif)

Un **problème d'optimisation multi-objectif** (ou *Multiobjective Optimization Problem* – MOP) se formule généralement de la manière suivante :

$$\text{MOP } F(x) = (f_1(x), \dots, f_k(x)) \rightarrow \max / \min \quad \text{s.c. } x \in D$$

Dans ce mémoire, nous nous intéressons uniquement au cas de la **minimisation** :

$$\text{MOP } F(x) = (f_1(x), \dots, f_k(x)) \rightarrow \min \quad \text{s.c. } x \in D$$

où l'ensemble des solutions réalisables D est défini par :

$$D = \{x \in \mathbb{R}^n \mid g_i(x) \geq b_i, \quad i = 1 \dots p, \quad \text{et } h_j(x) = c_j, \quad j = 1 \dots q\}$$

Définition 3.3. Espace des critères

L'espace des critères est l'image de l'espace réalisable notée $F = f(D)$. Les éléments de F sont appelés (fonctions) vecteurs objectifs ou vecteurs de critère et désignés par $F(x)$, où $F = (f_1, f_2, \dots, f_k)'$ et $F_i = f_i(x)$ pour tout $i = 1..k$, sont des valeurs du vecteur de critère. F est noté :

$$F = f(x), \quad x \in D$$

Un problème d'optimisation multi-objectifs porte sur deux espaces, que sont l'espace de décision D et l'espace fonctionnel (l'espace des fonctions coût) F .

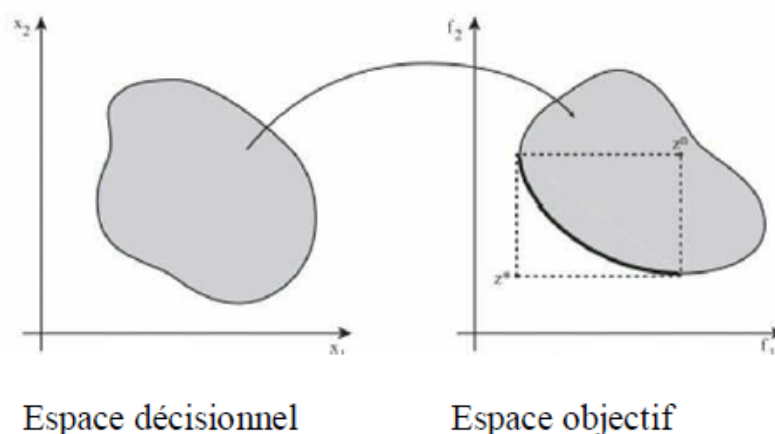


FIGURE 3.1 – Espace décisionnel et espace objectif d'un problème d'optimisation multi-objectif (exemple avec deux variables de décisions et deux fonctions objectifs)

Définition 3.4. (convexité)

Un problème d'optimisation multi-objectif est dit **convexe** si toutes les fonctions objectifs sont convexes et l'ensemble des solutions réalisables est convexe.

L'ensemble D est dit convexe si tout segment joignant deux points quelconques de D est inclus dans D .

$$x \in D \text{ et } y \in D \iff \text{segment } [x, y] \subset D$$

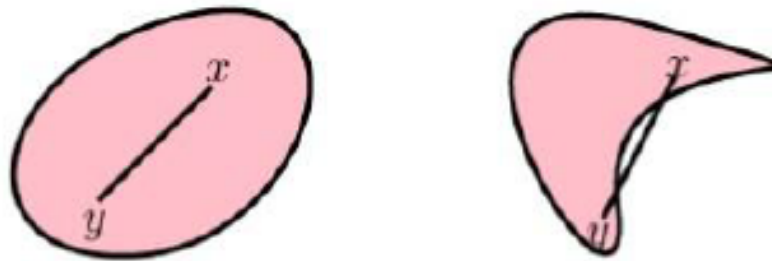


FIGURE 3.2 – Ensemble convexe (à gauche) et ensemble non convexe (à droite)

3.2 La dominance

En raison de la nature conflictuelle des objectifs, il n'y a généralement pas une solution réalisable qui optimise simultanément tous les objectifs. L'optimalité dans un contexte multi-objectif est basée sur la notion de dominance et d'efficacité au sens de Pareto.

Définition 3.5. (dominance)

Soient deux vecteurs critères $w = (w_1, \dots, w_k)$ et $z = (z_1, \dots, z_k)$.

On dit que w domine z si et seulement si :

$$\forall i \in \{1, \dots, k\}, w_i \leq z_i \quad \exists j \in \{1, \dots, k\}, w_j < z_j$$

C'est-à-dire :

Si w domine z , alors w est au moins aussi bon que z sur tous les critères et meilleur que lui sur au moins un des critères.

Propriétés de la relation de dominance

- N'est pas réflexive car une solution ne se domine pas elle-même.
- N'est pas symétrique, car on n'a jamais $y < z \Rightarrow z < y$.
- N'est pas anti-symétrique car on n'a jamais $y < z$ et $z < y$.
- Est transitive, car si $y < z$ et $z < w$ alors $y < w$.

3.2.1 Dominance forte

Soient deux vecteurs critères $w = (w_1, \dots, w_k)$ et $z = (z_1, \dots, z_k)$.

On dit que w domine fortement z si et seulement si :

$$\forall i \in \{1, \dots, k\}, w_i < z_i$$

C'est-à-dire :

Si w domine fortement z , alors w est meilleur que z sur tous les critères.

3.2.2 Dominance faible

Soient deux vecteurs critères $w = (w_1, \dots, w_k)$ et $z = (z_1, \dots, z_k)$.

On dit que w domine faiblement z si et seulement si :

$$\forall i \in \{1, \dots, k\}, w_i \leq z_i$$

3.2.3 Non-dominance

Soit w^* un vecteur critère tel que $w^* \in F$, on dit que w^* est non-dominé si et seulement s'il n'existe aucun autre vecteur critère $w \in F$ tel que :

$$\forall i \in \{1, \dots, k\}, w_i \leq w_i^*, \text{ et } w_i < w_i^* \text{ pour au moins un indice } i.$$

Dans le cas contraire, on dit que w^* est dominé.

3.3 Efficacité

3.3.1 Pareto optimale

Une solution $x^* \in D$ est Pareto optimale si et seulement s'il n'existe pas une solution réalisable $x \in D$ telle que :

$$f_i(x) \leq f_i(x^*) \quad \forall i = 1 \dots k \text{ et } f_j(x) < f_j(x^*) \text{ pour } j \in \{1 \dots k\}.$$

La définition de la Pareto optimalité découle directement de la notion de dominance. Elle signifie qu'il est impossible de trouver une solution qui améliore les performances sur un critère sans que cela entraîne une dégradation des performances sur au moins un autre critère. Elles forment le front Pareto. Les solutions Pareto optimales sont aussi connues sous le nom de solutions efficaces, non dominées ou non inférieures.

3.3.2 Efficacité forte

Une solution $x^* \in D$ est dite fortement efficace, s'il n'existe aucun vecteur $x \in D$ tel que $x \neq x^*$ et $f_i(x) \leq f_i(x^*) \quad \forall i = 1 \dots k$.

Une solution est fortement efficace si son vecteur critère est fortement non dominé.

3.3.3 Efficacité faible

Une solution $x^* \in D$ est dite faiblement efficace, s'il n'existe aucun vecteur $x \in D$ tel que $f_i(x) < f_i(x^*) \quad \forall i = 1 \dots k$.

Une solution est faiblement efficace si son vecteur critère est faiblement non dominé.

Dominance au sens de Geoffrion [14]

Une autre forme de dominance importante dans le monde de l'optimisation multi-objectif est la dominance au sens de Geoffrion. Elle est plus forte que l'autre, les solutions optimales obtenues par ce type de dominance sont appelées solutions optimales propres.

Définition 3.6. Une solution $x^* \in D$ est appelée solution Pareto optimale propre pour le problème multi-objectif noté (MOP) si :

1. x^* est Pareto optimal.
2. S'il existe un nombre $M > 0$ tel que $\forall i = 1, \dots, k, \forall x \in D$ vérifiant $f_i(x) < f_i(x^*)$, alors il existe au moins un $j \in \{1, \dots, k\}$ tel que $f_j(x^*) < f_j(x)$ et $\frac{f_i(x^*) - f_i(x)}{f_j(x) - f_j(x^*)} \leq M$.

Considérons le problème paramétrique mono-objectif suivant :

$$(P_\lambda) \min \sum_{i=1}^k \lambda_i f_i(x) \quad \text{avec } x \in D \quad \text{et} \quad \sum_{i=1}^k \lambda_i = 1.$$

Où λ_i sont des poids relatifs à chaque objectif f_i , $i = 1 \dots k$, tels que $\lambda_i \geq 0$ et $\sum_{i=1}^k \lambda_i = 1$.

Théorème de Geoffrion (propre) [14]

Si x^* est une solution optimale du problème P_λ avec $\lambda_i > 0$ et $\sum_{i=1}^k \lambda_i = 1$, alors x^* est Pareto optimale propre pour le problème (MOP).

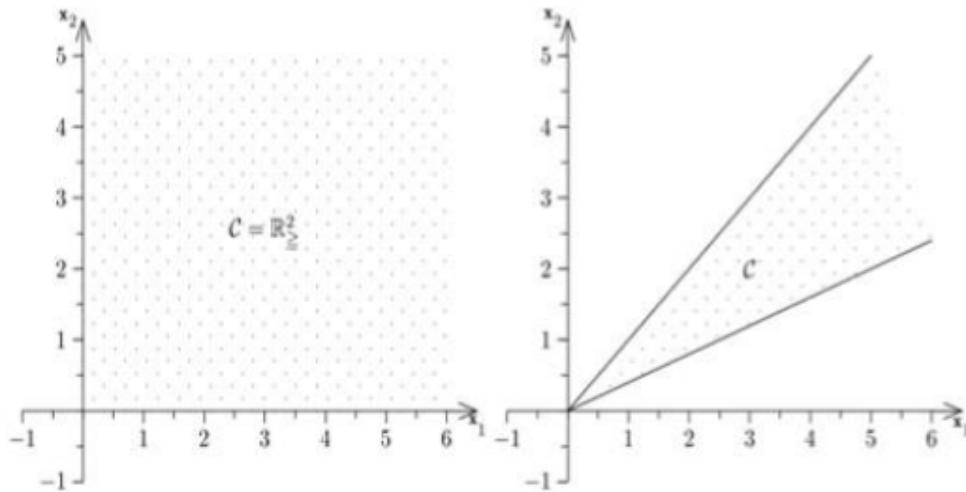
3.4 Optimalité au sens de Pareto

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). L'ensemble des solutions optimales au sens de Pareto est situé sur la limite de l'espace de critère F .

Définition 3.7. (cône)

Un sous-ensemble $K \subseteq \mathbb{R}^n$, $K \neq \emptyset$, est appelé un cône si et seulement si :

$$\forall x \in K, \forall \alpha \in \mathbb{R}_*^+, \alpha x \in K.$$



1. Cône négatif

Il est défini par :

$$C^- = \{x \in \mathbb{R}^n \mid f(x) \in \mathbb{R}^k \text{ avec } f(x) \leq 0\}$$

2. Cône positif

Il est défini par :

$$C^+ = \{x \in \mathbb{R}^n \mid f(x) \in \mathbb{R}^k \text{ avec } f(x) \geq 0\}$$

Remarque 3.1. — Le vecteur $0_{\mathbb{R}^n}$ est contenu dans chaque cône.

— Tous les cônes sont non bornés (sauf le singleton $0_{\mathbb{R}^n}$ qui contient uniquement l'origine).

Théorème du contact

Un vecteur $x \in \mathbb{R}^n$ est optimal au sens de Pareto pour un problème d'optimisation multi-objectif donné si et seulement si :

$$\{C^- + x\} \cap D = \{x\} \quad \text{et} \quad \{C^+ + x\} \cap D = \{x\}$$

Où D désigne l'espace des solutions réalisables.

L'utilisation de ce théorème est illustrée par la figure ci-dessus :

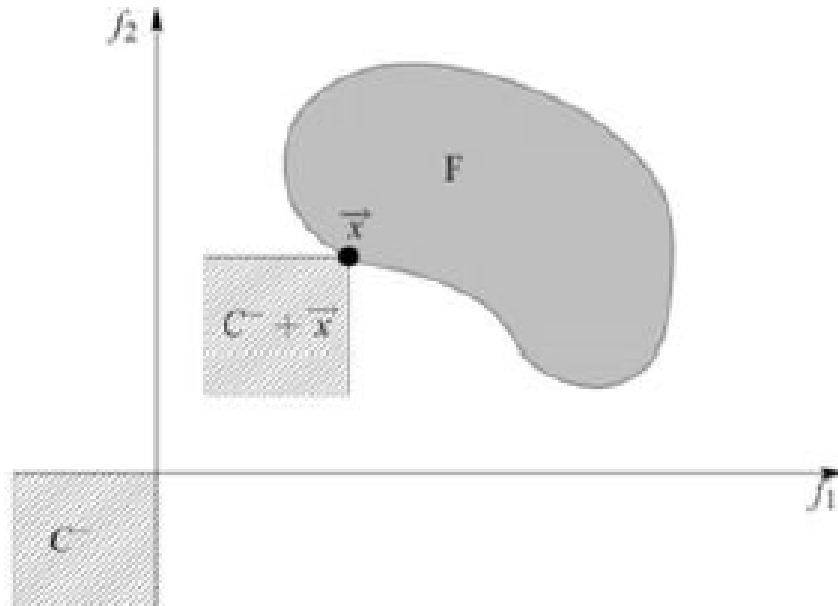


FIGURE 3.3 – Théorème du contact

Définition 3.8. (Optimalité locale au sens de Pareto)

Un vecteur $x \in \mathbb{R}^n$ est dit *optimal localement au sens de Pareto* s'il existe un réel $r > 0$ tel qu'il n'existe aucun vecteur $x' \in \mathbb{R}^n \cap B(x, r)$ qui domine x , où $B(x, r)$ désigne une boule ouverte de centre x et de rayon r .

Autrement dit, le vecteur x est optimal localement au sens de Pareto s'il est optimal (au sens de Pareto) dans une *restriction locale* de l'ensemble D .

Cette définition est illustrée par la figure suivante :

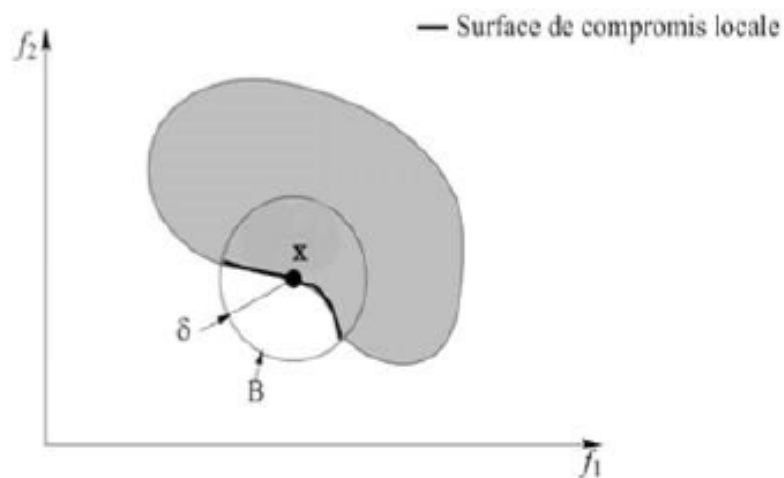


FIGURE 3.4 – Optimalité locale au sens de Pareto

Définition 3.9. (Optimalité globale au sens de Pareto)

Un vecteur x est dit optimal globalement au sens de Pareto (ou simplement optimal au sens de Pareto) s'il n'existe aucun vecteur x' tel que x' domine le vecteur x .

La différence entre cette définition et celle de l'optimalité locale est que l'on ne considère plus une partie restreinte de l'ensemble D , mais l'ensemble entier.

Une version graphique de cette définition utilise le théorème du contact.

3.5 Caractérisations des solutions efficaces

Nous présentons ici quelques caractérisations permettant de vérifier si une solution réalisable est efficace.

3.5.1 Théorème d'Ecker et Wendell

Soit $x^* \in D$ un vecteur de décision donné et \tilde{x} un vecteur quelconque de D . x^* est efficace pour (MOP) si et seulement si x^* est solution optimale du problème auxiliaire suivant :

$$\text{Minimiser } \sum_{i=1}^k f_i(x) \quad \text{sous la contrainte } x \in D, \quad f_i(x) \leq f_i(\tilde{x}) \quad \forall i = 1 \dots k$$

Exemple :

Soit le problème suivant :

$$P : \quad \text{Minimiser } (-3x_1 - x_2; x_1 + 2x_2)$$

$$\text{Sous les contraintes : } 3x_1 - x_2 \leq 6, \quad x_2 \leq 2, \quad x_1, x_2 \in \mathbb{R}_+^2$$

On a, pour $\tilde{x} = (1, 1)$, une solution réalisable.

Formons alors le problème auxiliaire $P_{\tilde{x}}$:

$$P_{\tilde{x}} : \quad \text{Minimiser } \sum_{i=1}^2 f_i(x)$$

$$\text{Sous les contraintes : } 3x_1 - x_2 \leq 6, \quad x_2 \leq 2, \quad x_1, x_2 \in \mathbb{R}_+^2$$

$$f_1(x) \leq f_1(\tilde{x}) = f_1(1, 1) \Rightarrow -3x_1 - x_2 \leq -4$$

$$f_2(x) \leq f_2(\tilde{x}) = f_2(1, 1) \Rightarrow x_1 + 2x_2 \leq 3$$

En résolvant $P_{\tilde{x}}$ par la méthode du simplexe, la solution optimale est $x^* = (2, 0)'$.
Donc, x^* est efficace pour P .

3.5.2 Théorème de Benson

Soit le problème uni-critère suivant :

$$(P_{\varepsilon}) \quad \text{maximiser } \theta = \sum_{i=1}^k \varepsilon_i$$

$$\text{sous : } x \in D, \quad f_i(x) + \varepsilon_i = f_i(x^*) \quad \forall i = 1 \dots k$$

$$\varepsilon_i \geq 0 \quad \forall i = 1 \dots k$$

Soit x^* une solution réalisable donnée, alors :

1. x^* est efficace pour le problème (MOP) si et seulement si la valeur optimale de la fonction objectif est nulle dans $(P_\epsilon x^*)$.
2. Si $\theta_{\text{opt}} \neq 0$ pour la solution x de $P_\epsilon x^*$, alors : x est efficace pour (MOP).

Exemple :

Soit le problème multi-objectif (MOP) suivant :

$$MOP : \quad \text{Minimiser } (x_1 - 2x_2; -x_1 + 4x_2; 2x_1 + x_2)$$

sous les contraintes :

$$-2x_1 + x_2 \leq 0, \quad x_1 \leq 3, \quad x_2 \leq 2, \quad x_1, x_2 \in \mathbb{R}_+^2$$

Vérifions si la solution réalisable $x_1 = 0, x_2 = 0$ est efficace en utilisant le test de Benson.

$$P_\epsilon x_1 : \quad \max \theta = \epsilon_1 + \epsilon_2 + \epsilon_3$$

Sous les contraintes :

$$\begin{aligned} -2x_1 + x_2 \leq 0, \quad x_1 \leq 3, \quad x_2 \leq 2, \quad f_1(x) + \epsilon_1 = f_1(x_1) &\Rightarrow x_1 - 2x_2 + \epsilon_1 = 0 \\ f_2(x) + \epsilon_2 = f_2(x_1) &\Rightarrow -x_1 + 4x_2 + \epsilon_2 = 0 \\ f_3(x) + \epsilon_3 = f_3(x_1) &\Rightarrow 2x_1 + x_2 + \epsilon_3 = 0 \\ x_1, x_2, \epsilon_1, \epsilon_2, \epsilon_3 &\geq 0 \end{aligned}$$

La solution optimale de $P_\epsilon x_1$ est $x_1 = 0, x_2 = 0, \epsilon_1 = 0, \epsilon_2 = 0, \epsilon_3 = 0. \Rightarrow \theta_{\text{opt}} = 0 \Rightarrow x_1 = (0, 0)$ est efficace pour (MOP).

3.6 Points spéciaux et matrice des gains

On pose $Z_i = F_i$.

3.6.1 Le point idéal

Le vecteur idéal $Z^* = (z_1^*, z_2^*, \dots, z_k^*) \in \mathbb{R}^k$ est le vecteur qui minimise chacune des fonctions objectifs F_i séparément :

$$Z_i^* = \min F_i(x), \quad x \in D$$

Il est clair que si le vecteur idéal est réalisable, il est la solution du problème *MOP*. Mais ce n'est pas en général possible à cause des conflits qui existent entre les critères.

3.6.2 Le point anti-idéal

Le vecteur anti-idéal $Z^a = (z_1^a, z_2^a, \dots, z_k^a) \in \mathbb{R}^k$ est le vecteur qui maximise chacune des fonctions objectifs F_i : C'est-à-dire :

$$Z^a = \max F_i(x), \quad x \in D$$

3.6.3 Le point de référence

C'est un vecteur $(\bar{f}_1, \bar{f}_2, \dots, \bar{f}_k)$ dont les coordonnées sont les valeurs souhaitables que l'on doit atteindre.

3.6.4 Matrice des gains

Nous appelons matrice des gains, une matrice qui est formée en utilisant les vecteurs de décision obtenus lors du calcul du vecteur critère idéal. La i -ème ligne de la matrice des gains correspond aux valeurs de toutes les fonctions objectifs calculées au point où f_i atteint sa valeur minimale. Alors $Z_i^* = M_i$ et le point idéal est sur la diagonale principale de la matrice :

$$M = \begin{pmatrix} M_1 & f_1(\bar{x}^2) & \cdots & f_1(\bar{x}^k) \\ f_2(\bar{x}^1) & M_2 & \cdots & f_2(\bar{x}^k) \\ \vdots & \vdots & \ddots & \vdots \\ f_k(\bar{x}^1) & f_k(\bar{x}^2) & \cdots & M_k \end{pmatrix}$$

3.6.5 Le point Nadir

Le point critère nadir est le vecteur dont chaque composante Z_i^{nad} correspond à la valeur maximale de la i -ème ligne de la matrice des gains.

3.6.6 Front de Pareto

C'est l'ensemble des vecteurs de décision qui ne sont pas dominés.

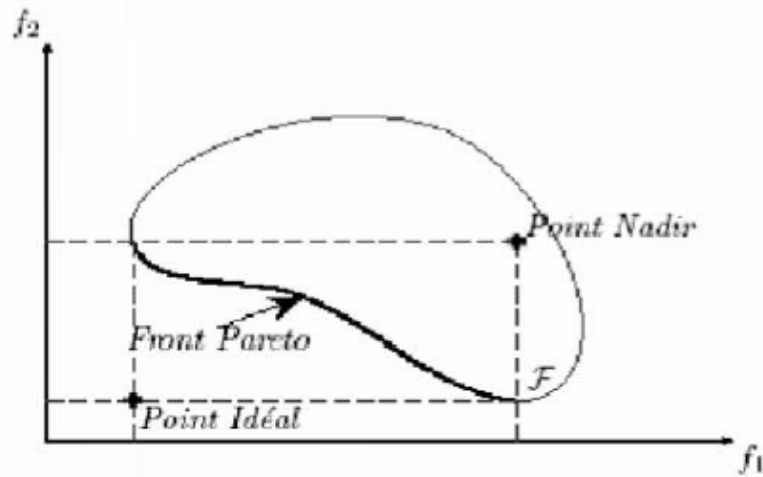


FIGURE 3.5 – point idéal et point nadir

3.7 Fonctions scalarisantes

Le choix d'une solution efficace parmi toutes les solutions réalisables nécessite une certaine connaissance de la structure de préférence. Cette information peut être obtenue directement ou indirectement et peut parfois se traduire en termes de paramètres de préférence. Les plus courantes sont :

- Les poids φ_k , avec $k = 1, 2, \dots, p$, qui reflètent l'importance relative de chaque critère.
- Le point de référence, défini par des niveaux de réservation (valeurs souhaitables) pour chaque critère.
- Le point de réservation, défini par des niveaux (valeurs non souhaitables) pour chaque critère.

La fonction scalarisante est définie par $s(z, \varphi) : \mathbb{R}^p \rightarrow \mathbb{R}$, où φ représente l'ensemble des paramètres. Les fonctions scalarisantes les plus courantes sont les suivantes :

3.7.1 Fonctions linéaires

$$s_1(z, \varphi) = \sum_{k=1}^p \varphi_k z_k \quad \text{avec} \quad \sum_{k=1}^p \varphi_k = 1, \quad \varphi_k > 0, \quad \forall k$$

et

$$s_2(z, \varphi) = \sum_{k=1}^p \varphi_k |z_k - \bar{z}_k|$$

3.7.2 Norme Lp pondérée

$$s_3(z, \varphi) = \left(\sum_{k=1}^p \varphi_k |z_k - \bar{z}_k|^q \right)^{1/q}, \quad q \in \mathbb{Z}^+$$

3.7.3 Norme L1 de Tchebycheff pondérée

$$s_4(z, \varphi) = \max_{1 \leq k \leq p} \varphi_k |z_k - \bar{z}_k|$$

3.7.4 Norme Tchebycheff pondérée augmentée

$$s_5(z, \varphi) = \max_{1 \leq k \leq p} \varphi_k |z_k - \bar{z}_k| + \epsilon \sum_{k=1}^p \varphi_k |z_k - \bar{z}_k|, \quad \epsilon > 0$$

Ici, \bar{z}_k représente une valeur de référence de z_k .

Les fonctions scalarisantes ne peuvent produire que des vecteurs non dominés. Dans ce cas, elles caractérisent complètement l'ensemble des solutions efficaces.

3.8 Méthodes de résolution d'un programme multi-objectif

3.8.1 Méthode du simplexe multi-objectif

[15]

La méthode du simplexe multicritère consiste à générer un premier point efficace à partir d'une solution de base réalisable, puis à énumérer tous les autres points efficaces. Cependant, cette méthode ne teste pas toutes les bases, car certaines sont manifestement dominées.

Considérons le problème suivant :

Maximiser $F_1 = c^1 x$

Maximiser $F_2 = c^2 x$

⋮

Maximiser $F_k = c^k x$

Sous les contraintes : $Ax = b, \quad x \geq 0$

où $D = \{x \in \mathbb{R}^n \mid Ax = b, x \geq 0\}$

Le tableau du simplexe (en considérant séparément chaque objectif) est donné dans le tableau (1).

Dans ce tableau, on suppose, sans perte de généralité, que les m premières variables sont dans la base. On définit alors :

- J_B : l'ensemble des indices des variables de base ;
- $J_H = J \setminus J_B$: l'ensemble des indices des variables hors base.

					C^1	C_1^1	C_m^1	C_{m+1}^1	C_j^1
					C^2	C_1^2	C_m^1	C_{m+1}^2	C_j^2
					\vdots				\vdots		\vdots	
					C^k	C_1^k	C_m^k	C_{m+1}^k	C_j^k
C_B^1	C_B^2	C_B^j	C_B^k	base	B	a_1	a_m	a_{m+1}	a_j
C_1^1	C_1^2	C_1^j	C_1^k	a_1	b_1	1		0	x_{1m+1}	x_{1j}
\vdots	\vdots	\vdots	\vdots	\vdots		0	\vdots	0	\vdots		\vdots	
\vdots	\vdots	\vdots	\vdots	\vdots		0	\vdots	0	\vdots		\vdots	
C_m^1	C_m^2	C_m^j	C_m^k	a_m	b_m	0		1	x_{mm+1}	x_{mj}
					0	0	0		Δ_{m+1}^1		Δ_j^1	
					\vdots	0	0		Δ			
					0	0	\vdots		Δ_{m+1}^i	Δ_j^i
					\vdots	0	\vdots		\vdots			
					0	0	0		Δ_{m+1}^k	Δ_j^k

$$F_0 = \begin{pmatrix} F_0^1 = c^1 b \\ F_0^2 = c^2 b \\ \vdots \\ F_0^k = c^k b \end{pmatrix}$$

Le test d'efficacité :

Théorème. Soit $\bar{x} \in D$, une solution efficace (respectivement faiblement efficace), si et seulement s'il existe :

$$\lambda \in \Lambda = \left\{ \lambda \in \mathbb{R}^k \mid \sum_{i=1}^k \lambda_i = 1, \lambda_i > 0 \right\}$$

(Répectivement, il existe $\lambda \in \Lambda = \left\{ \lambda \in \mathbb{R}^k \mid \sum_{i=1}^k \lambda_i = 1, \lambda_i \geq 0 \right\}$) tel que \bar{x} maximiser le problème des sommes pondérées donné par :

$$P_\lambda : \max \lambda' C' x \quad \text{s.c. } x \in D$$

Théorème. [16]

Si D possède un point efficace, alors au moins un point extrême de D est efficace.

Preuve : Soit \bar{x} un point efficace de D . D'après le théorème 1, il existe $\lambda \in \Lambda$ tel que :

$$\lambda C' \bar{x} = \max_{x \in D} \lambda C' x$$

Comme une fonction linéaire atteint son minimum en un point extrême, alors \bar{x} est un point extrême efficace.

Théorème. Soit $x \in D$ un point extrême associé à une base efficace B , alors x est efficace.

Preuve : Puisqu'il existe un $\lambda \in \Lambda$ pour lequel B est une base optimale (par le théorème [16]), alors x est efficace.

Théorème. Soient B et \bar{B} deux bases efficaces adjacentes obtenues à partir d'un pivot efficace, et soient x et \bar{x} les points extrêmes associés à B et à \bar{B} respectivement. Alors, l'arête (x, \bar{x}) est efficace.

Théorème. [16]

Soit (x, v) une arête efficace infinie de D . Alors, x est un point extrême associé à une base efficace B .

De la théorie du simplexe uni-critère, on a :

$$1. \Delta_j^i = \sum_{r \in J_B} C_r^i x_{rj} - C_j^i, \quad j \in J_H, \quad \forall i = 1 \dots k$$

$$F_0^i = \sum_{r \in J_B} C_r^{i'} b_r \quad \forall i = 1 \dots k$$

Si :

$$\Delta_j^i \geq 0, \quad \forall j \in J_H$$

alors :

$$x^0 = \begin{bmatrix} b \\ 0 \end{bmatrix}, \quad b \in \mathbb{R}_+^m, \quad 0 \in \mathbb{R}^{n-m}$$

est une solution optimale pour le critère i .

2. Si on introduit la $j_i^{\text{ème}}$ variable dans la base, nous obtenons une nouvelle solution x^1 et un vecteur :

$$\hat{F}_0 = F_0 - \theta_j \Delta_j$$

où :

$$\hat{F}_0 = \begin{bmatrix} \hat{F}_0^1 \\ \hat{F}_0^2 \\ \vdots \\ \hat{F}_0^k \end{bmatrix} = \begin{bmatrix} F_0^1 \\ F_0^2 \\ \vdots \\ F_0^k \end{bmatrix} - \theta_j \begin{bmatrix} \Delta_j^1 \\ \Delta_j^2 \\ \vdots \\ \Delta_j^k \end{bmatrix}$$

$$3. \theta_j = \min \left\{ \frac{b_r}{x_{rj}} \mid x_{rj} > 0 \right\}, \quad \forall j \in J_H, \quad r \in J_B$$

Remarque 3.2. 1. Soit x^0 une solution basique réalisable.

a. S'il existe un $j \in J_H$ tel que tous les $\Delta_j^i \leq 0$, avec au moins une inégalité stricte, et si $\theta_j > 0$, alors la solution courante x^0 est dominée. En effet, si l'on introduit la j i-ème variable à la base, on obtient un point extrême adjacent x_1 , pour lequel $\hat{F}_0 \geq F_0$ avec au moins une inégalité stricte, car :

$$\theta_j \Delta_j \leq 0, \quad \text{donc} \quad \hat{F}_0 = F_0 - \theta_j \Delta_j \geq F_0$$

b. S'il existe un $j \in J_H$ tel que $\Delta_j^i \geq 0$, avec au moins une inégalité stricte, et si de plus $\theta_j > 0$, alors l'introduction de la j_i -ème variable à la base mène à une solution dominée. En effet, si on introduit la variable j dans la base, on obtient un nouveau point extrême x_1 pour lequel :

$$\hat{F}_0 = F_0 - \theta_j \Delta_j \leq F_0 \quad \text{avec au moins une inégalité stricte.}$$

2. Soit x^0 une solution basique réalisable. S'il existe $j_1, j_2 \in J_H$ tels que :

$$\theta_{j_1} \Delta_{j_1} \leq \theta_{j_2} \Delta_{j_2} \quad \text{avec au moins une inégalité stricte,}$$

alors l'introduction de la variable d'indice j_2 dans la base conduit à une solution dominée par celle résultant de l'introduction de la variable d'indice j_1 . En effet, puisque $\theta_{j_1} \Delta_{j_1} \leq \theta_{j_2} \Delta_{j_2}$ avec au moins une inégalité stricte, cela implique que :

$$F_0 - \theta_{j_2} \Delta_{j_2} \leq F_0 - \theta_{j_1} \Delta_{j_1} \quad \text{avec au moins une inégalité stricte.}$$

Remarque 3.3. 1. S'il existe un indice $i \in \{1, \dots, k\}$ tel que $\Delta_j^i \geq 0$ pour tout $j \in J_H$, alors le i -ème critère est à son maximum et la solution basique correspondante est non dominée, à condition qu'il n'existe pas de colonne k' avec $\Delta_{k'}^i = 0$.

2. D'après la remarque précédente, seules les colonnes (variables) non comparables à zéro et les variables x_j , et x telles que :

$$\theta_k \Delta_k \text{ incomparable à } \theta_j \Delta_j$$

sont admissibles pour une introduction dans la base. Dans ce cas, on ne peut pas dire si la solution correspondante est dominée ou non. Pour cela, on considère le test dit de non-dominance énoncé par le théorème suivant :

Théorème. (*Test de dominance*)

Soit le problème :

$$\max v = \sum_{i=1}^k \epsilon_i$$

sous les contraintes :

$$C'x - \delta = C'\bar{x}$$

$$x \in D = \{x \in \mathbb{R}^n \mid Ax = b, x \geq 0\}$$

$$\delta \in \mathbb{R}^n, \quad \delta \geq 0, \quad \bar{x} \in D.$$

Alors :

- \bar{x} est efficace si et seulement si : $\max v = 0$
- \bar{x} est dominée si et seulement si : $\max v > 0$

Exemple

Soit le problème multi-critères suivant :

$$(1) \begin{cases} \max Z_1 = 0,4x_1 + 0,3x_2 \\ \max Z_2 = x_1 \\ x_1 + x_2 \leq 400 \\ 2x_1 + x_2 \leq 500 \\ x_1 \geq 0, \quad x_2 \geq 0 \end{cases}$$

$$(1) \Leftrightarrow \begin{cases} \max Z_1 = 0,4x_1 + 0,3x_2 \\ \max Z_2 = x_1 \\ x_1 + x_2 + x_3 = 400 \\ 2x_1 + x_2 + x_4 = 500 \\ x_j \geq 0 \quad \forall j = 1, 4 \end{cases}$$

$x^1 = \begin{pmatrix} 0 \\ 0 \\ 400 \\ 500 \end{pmatrix}$ est une solution de base réalisable associée à la base.

$$(a_3, a_4) = I_2 \quad \text{et} \quad z(x) = \begin{pmatrix} z_1(x) \\ z_2(x) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

Le tableau de simplexe initial est :

			C^1	0.4	0.3	0	0
			C^2	1	0	0	0
C_B^1	C_B^2	Base	B	a_1	a_2	a_3	a_4
0	0	a_3	400	1	1	1	0
0	0	a_4	500	2	1	0	1
			Δ^1	-0.4	-0.3	0	0
			Δ^2	-1	0	0	0

↑

$$J_H = \{1, 2\} \quad \text{et} \quad J_B = \{3, 4\}$$

$\exists i \in \{1, 2\}$ tel que $\Delta_j^i \geq 0$ pour tout $j \in J_H = \{1, 2\}$, donc aucun objectif n'est à son maximum.

On a :

$$\begin{aligned} & \begin{pmatrix} -0.4 \\ -1 \end{pmatrix} < 0, \quad x^1 \text{ est dominé.} \\ \theta_1 &= \min \left(\frac{400}{1}, \frac{500}{2} \right) = 250 \quad (\text{atteint pour } r = 4) \\ \theta_2 &= \min \left(\frac{400}{1}, \frac{500}{1} \right) = 400 \quad (\text{atteint pour } r = 3) \\ \theta_1 \Delta_1 &= 250 \begin{pmatrix} -0.4 \\ -1 \end{pmatrix} = \begin{pmatrix} -100 \\ -250 \end{pmatrix} \\ \theta_2 \Delta_2 &= 400 \begin{pmatrix} -0.3 \\ 0 \end{pmatrix} = \begin{pmatrix} -120 \\ 0 \end{pmatrix} \end{aligned}$$

Donc $\theta_1 \Delta_1$ est incomparable à $\theta_2 \Delta_2$.

Introduisons a_1 dans la base à la place de a_4 , on aura alors le tableau du simplexe suivant :

			C^1	0.4	0.3	0	0
			C^2	1	0	0	0
C_B^1	C_B^2	Base	B	a_1	a_2	a_3	a_4
0	0	a_3	150	0	$1/2$	1	$-1/2$ →
0.4	1	a_1	250	1	$1/2$	0	$1/2$
		$Z_0 = \begin{pmatrix} z_1^1 = 100 \\ z_1^2 = 250 \end{pmatrix}$	Δ^1	0	-0.1	0	0.2
			Δ^2	0	$1/2$	0	$1/2$

↑

$\exists i = 2$ tel que $\Delta_j^i > 0 \forall j \in J_H$, ce qui implique que le deuxième critère est à son maximum.

Le point extrême $x^2 = (250, 0, 150, 0)'$ est non dominé.

$$\theta_2 = \min \left(\frac{150}{0.5}, \frac{250}{0.5} \right) = 300 \quad (\text{atteint pour } r = 3)$$

$$\theta_4 = \min \left(\frac{250}{0.5} \right) = 500 \quad (\text{atteint pour } r = 1)$$

$$\theta_2 \Delta_2 = 300 \begin{pmatrix} -0.1 \\ 0.5 \end{pmatrix} = \begin{pmatrix} -130 \\ 150 \end{pmatrix}$$

$$\theta_4 \Delta_4 = 500 \begin{pmatrix} 0.2 \\ 0.5 \end{pmatrix} = \begin{pmatrix} 100 \\ 250 \end{pmatrix}$$

$$\theta_2 \Delta_2 < \theta_4 \Delta_4.$$

Donc a_2 rentre dans la base et a_3 sort de la base d'après la remarque.

Dressons alors le troisième tableau simplexe :

			C^1	0.4	0.3	0	0
			C^2	1	0	0	0
C_B^1	C_B^2	Base	B	a_1	a_2	a_3	a_4
0.3	1	a_2	300	0	1	2	-1
0.4	1	a_1	100	1	0	-1	1
		$Z_0 = \begin{pmatrix} z_1^1 = 130 \\ z_1^2 = 100 \end{pmatrix}$	Δ^1	0	0	0.2	0.1
			Δ^2	0	0	-1	1

→

↑

Si $\Delta_1 \geq 0$, alors le premier objectif est à son maximum et puisqu'il n'existe pas de colonne k telle que $\Delta_k^1 = 0$ avec $k \in J_H$, alors $x^3 = \begin{pmatrix} 100 \\ 300 \\ 0 \\ 0 \end{pmatrix}$ est non dominée.

$$\theta_3 = \min\left(\frac{300}{2}\right) = 150 \quad (\text{atteint pour } r = 2)$$

$$\theta_4 = \min\left(\frac{100}{1}\right) = 100 \quad (\text{atteint pour } r = 1)$$

$$\theta_3 \Delta_3 = 150 \begin{pmatrix} 0.2 \\ -1 \end{pmatrix} = \begin{pmatrix} 30 \\ -150 \end{pmatrix}$$

$$\theta_4 \Delta_4 = 100 \begin{pmatrix} 0.1 \\ 1 \end{pmatrix} = \begin{pmatrix} 10 \\ 100 \end{pmatrix}$$

On remarque que $\theta_3 \Delta_3$ est incomparable à $\theta_4 \Delta_4$.

Puisque l'introduction de a_3 dans la base conduit à une base déjà explorée, donc on introduira a_4 .

Dressons alors le quatrième tableau du simplexe :

			C^1	0.4	0.3	0	0
			C^2	1	0	0	0
C_B^1	C_B^2	Base	B	a_1	a_2	a_3	a_4
0.3	0	a_2	400	1	1	1	0
0	0	a_4	100	1	0	-1	1
		$Z_0 = \begin{pmatrix} z_1^1 = 120 \\ z_1^2 = 0 \end{pmatrix}$	Δ^1	-0.1	0	0.3	0
			Δ^2	-1	0	0	0

$\nexists i \in \{1, 3\}$ tel que $\Delta_j^i \geq 0 \quad \forall J_H \in \{1, 3\}$, donc aucun objectif n'est à son maximum.

$$\Delta_1 = \begin{pmatrix} -0.1 \\ -1 \end{pmatrix} < 0 \Rightarrow x^4 = \begin{pmatrix} 0 \\ 400 \\ 0 \\ 100 \end{pmatrix} \text{ est dominé.}$$

Finalement :

En considérant le problème initial (1), seuls les points extrêmes

$$x^2 = \begin{pmatrix} 250 \\ 0 \end{pmatrix} \quad \text{et} \quad x^3 = \begin{pmatrix} 100 \\ 300 \end{pmatrix} \text{ sont non dominés, avec :}$$

$$z(x^{2*}) = \begin{pmatrix} 100 \\ 250 \end{pmatrix} \quad \text{et} \quad z(x^{3*}) = \begin{pmatrix} 130 \\ 100 \end{pmatrix}$$

On a recensé deux solutions efficaces.

La méthode du simplexe multicritère nous apporte clarté et précision dans toutes ses étapes, donc facilité d'application. Cependant, en pratique, elle peut être non efficace, car la solution désirée peut se situer sur une arête, et non sur un point extrême efficace.

3.8.2 Méthode de la somme pondérée ou méthode d'agrégation des objectifs

[17]

Cette méthode, probablement la plus largement utilisée dans la pratique, consiste à ramener le problème (MOP) au problème de l'optimisation d'une combinaison linéaire des objectifs initiaux, suivant :

$$(P_\lambda) \quad \min / \max \sum_{i=1}^k \lambda_i f_i(x) \quad \text{avec } x \in D$$

où les poids $\lambda_i \geq 0$, pour $i = 1 \dots k$ et $\sum_{i=1}^k \lambda_i = 1$.

(Les poids λ_i sont choisis en fonction de l'importance relative des objectifs.)

Théorème. *Toute solution optimale du problème (P_λ) est une solution Pareto-optimale pour le problème (MOP) si tous les λ_i sont strictement positifs, pour $i = 1 \dots k$.*

Théorème. *Si le problème (MOP) est convexe (en particulier linéaire), et si x^* est une solution Pareto-optimale de (MOP), alors il existe un vecteur $\lambda = (\lambda_1, \lambda_2, \dots, \lambda_k)$, avec $\lambda_i > 0$ ($\sum \lambda_i = 1$) pour tout $i = 1 \dots k$, tel que x^* est solution optimale de (P_λ) .*

3.8.3 Méthode des ε -contraintes

Une autre façon de transformer le problème (MOP) en un problème à objectif unique consiste à minimiser (ou maximiser) l'objectif f_i jugé prioritaire par le décideur, tout en convertissant les autres objectifs en contraintes. Le problème transformé s'écrit :

$$(P_\varepsilon) \quad \min f_i(x) \quad \text{tel que } x \in D, \quad f_j(x) \leq \varepsilon_j, \quad \forall j \neq i \text{ avec } j = 1 \dots k$$

Les $\varepsilon_j \in \mathbb{R}$ sont appelés *seuils de satisfaction*.

Pour un problème (MOP) de type maximisation :

$$(P_\varepsilon) \quad \max f_i(x) \quad \text{tel que } x \in D, \quad f_j(x) \geq \varepsilon_j, \quad \forall j \neq i \text{ avec } j = 1 \dots k$$

Théorème. *Pour tout vecteur $\varepsilon = (\varepsilon_1, \varepsilon_2, \dots, \varepsilon_{i-1}, \varepsilon_{i+1}, \dots, \varepsilon_k)$ tel que le problème (P_ε) admet une solution optimale, alors cette solution est Pareto-optimale pour le problème (MOP).*

Remarque 3.4. 1. Les paramètres ε_j sont choisis par le décideur.

2. En faisant varier les ε_j dans les deux méthodes, on peut générer un sous-ensemble de l'ensemble Pareto du problème (MOP).

Exemple

Résolution du problème (MOP) suivant avec la méthode des ε -contraintes, en considérant le critère 3 comme prioritaire et $\varepsilon = (\varepsilon_1, \varepsilon_2) = (1, 3)$.

$$(MOP) \quad \min f(x) = \begin{pmatrix} f_1(x) = -3x_1 + x_2 \\ f_2(x) = x_1 + 2x_2 \\ f_3(x) = 2x_1 + x_2 \end{pmatrix}$$

Sous les contraintes :

$$\begin{cases} 3x_1 - x_2 \leq 6 \\ x_2 \leq 2 \\ x = (x_1, x_2) \in \mathbb{R}_+^2 \end{cases}$$

On souhaite résoudre le problème (P_ε) , qui consiste à :

$$(P_\varepsilon) \quad \min f_3(x) = 2x_1 + x_2$$

Sous les contraintes :

$$\begin{cases} 3x_1 - x_2 \leq 6 \\ x_2 \leq 2 \\ f_1(x) \leq \varepsilon_1 \Rightarrow -3x_1 + x_2 \leq 1 \\ f_2(x) \leq \varepsilon_2 \Rightarrow x_1 + 2x_2 \leq 3 \\ x = (x_1, x_2) \in \mathbb{R}_+^2 \end{cases}$$

En résolvant (P_ε) par la méthode du simplexe, on obtient la solution optimale :

$$x^* = (x_1, x_2) = (0, 0)'$$

D'après le théorème 3, $x^* = (0, 0)'$ est une solution Pareto-optimale pour le problème (MOP).

3.9 Conclusion

La programmation linéaire multi-objectifs a connu un fort développement au cours des trente dernières années. Ce progrès est principalement dû aux avancées de la programmation linéaire classique et à la capacité de ce modèle à représenter efficacement de nombreux problèmes réels. Sa mise en œuvre relativement simple et son utilité dans des domaines variés expliquent son succès. Aujourd'hui, elle demeure un domaine de recherche actif, riche en applications pratiques et en perspectives théoriques.

Chapitre 4

Contrôle Optimal Linéaire Discret : Méthode Adaptée

4.1 Introduction

Dans ce chapitre, nous étudions une méthode de résolution d'un problème linéaire de contrôle optimal, appelée méthode adaptée. Nous présentons essentiellement des résultats théoriques ainsi que des exemples d'application pour le cas discret.

4.2 Le cas discret

Dans cette partie, nous étudions une méthode de résolution d'un problème linéaire de contrôle optimal pour le cas discret [13].

4.2.1 Position du problème

Considérons le système dynamique linéaire suivant :

$$J(u) = c'x(t^*) \rightarrow \max_u \quad (3.1)$$

$$\dot{x} = \frac{dx}{dt} = Ax + bu, \quad x(t_0) = x_0 \quad (3.2)$$

$$Hx(t^*) = g \quad (3.3)$$

$$f_* \leq u(t) \leq f^*, \quad t \in T = [t_0, t^*] \quad (3.4)$$

où :

- $x(t)$ est un n -vecteur décrivant la trajectoire du système à l'instant t
- $u(t)$ est la commande d'entrée à l'instant t du système
- $f_*(t)$ et $f^*(t)$ sont des fonctions définies sur $T = [t_0, t^*]$
- A est une matrice $n \times n$ caractérisant l'état du système
- b est un n -vecteur
- x_0 est la position initiale du système à l'instant $t = t_0$
- le système est caractérisé par un signal de sortie à $t = t^*$: $Hx(t^*) = g$
- H est une matrice $m \times n$ avec $\text{rang}(H) = m \leq n$, g est un m -vecteur.
- $J(u) = c'x(t^*) \rightarrow \max_u$ est un critère de qualité.
- $I = \{1, \dots, m\}$ est l'ensemble des indices des lignes.
- $J = \{1, \dots, n\}$ est l'ensemble des indices des colonnes.

Définition 4.1. Une commande u est dite *impulsive* sur l'intervalle $[t_0, t^*]$ si :

$$u(t) = u_i = \text{constante}, \quad t \in [\tau_i, \tau_{i+1}], \quad i = 0, \dots, N;$$

avec

$$\tau_0 = t_0, \quad \tau_{N+1} = t^*, \quad \tau_{i+1} - \tau_i = h > 0,$$

où h est le pas de quantification.

En utilisant la formule de Cauchy, la solution du système dynamique (3.2) s'écrit sous la forme :

$$x(t) = F(t) \left[x_0 + \int_{t_0}^t F^{-1}(\tau) bu(\tau) d\tau \right], \quad (3.5)$$

où $F(t) = \exp(At)$ pour $t \in T$, est la solution du système différentiel suivant :

$$\begin{cases} \dot{F} = AF, \\ F(t_0) = \text{Id}, \quad t \in T. \end{cases}$$

Comme la commande u est impulsive, l'équation (3.5) prend la forme discrète suivante :

$$x(i+1) = x(\tau_{i+1}) = F(h) \left[x(i) + \int_0^h F^{-1}(\tau) bu(i) d\tau \right],$$

où

$$u(i) = u(\tau_i), \quad \tau_i \in T, \quad i = 0, \dots, N, \quad T = \bigcup_{i=0}^N [\tau_i, \tau_{i+1}], \quad \tau_0 = t_0, \quad \tau_{N+1} = t^*, \quad \tau_{i+1} - \tau_i = h > 0.$$

En posant $D = \exp(Ah) = \sum_{n=0}^{\infty} \frac{(Ah)^n}{n!}$ et

$$d = D \cdot \int_0^h F^{-1}(\tau) b d\tau,$$

on aboutit au système suivant :

$$\begin{cases} J(u) = c'x(t^*) \longrightarrow \max_u, \\ x(t+h) = Dx(t) + du(t), \quad x(t_0) = x_0, \\ Hx(t^*) = g, \\ f_*(t) \leq u(t) \leq f^*(t), \quad t \in T = [t_0, t^*]. \end{cases} \quad (3.6)$$

avec $x(t^*)$ vérifiant la propriété suivante :

$$x(t^*) = D^{\frac{t^*}{h}} x_0 + \sum_{t \in T} D^{\frac{t^*-t}{h}-1} du(t).$$

Cela permet d'établir une autre formulation équivalente du problème (3.6), comme suit :

$$\begin{cases} J(u) = \left(c' \sum_{t \in T} D^{\frac{t^*-t}{h}-1} du(t) + c' D^{\frac{t^*}{h}} x_0 \right) \longrightarrow \max_u, \\ H \sum_{t \in T} D^{\frac{t^*-t}{h}-1} du(t) = g - HD^{\frac{t^*}{h}} x_0, \\ f_*(t) \leq u(t) \leq f^*(t), \quad t \in T = [t_0, t^*]. \end{cases} \quad (3.7)$$

où la dynamique du système reste :

$$x(t+h) = Dx(t) + du(t), \quad x(t_0) = x_0.$$

4.2.2 Notion de commandabilité

Un processus de contrôle décrit par l'équation (3.2) est dit **commandable** s'il est possible, pour toute paire de points $x_0, x^* \in \mathbb{R}^n$, de trouver une commande $u(t)$ mesurable et bornée, définie sur un intervalle fini $[t_0, t^*]$, telle que la trajectoire $x(t)$ du système vérifie :

$$x(t_0) = x_0 \quad \text{et} \quad x(t^*) = x^*.$$

Autrement dit, le système peut être piloté depuis une position initiale x_0 jusqu'à une position finale x^* en un temps fini.

Théorème. [13]

Un système linéaire de \mathbb{R}^n décrit par l'équation (3.2) est commandable si et seulement si le rang de la matrice $[b \ Ab \ A^2b \ \dots \ A^{n-1}b]$ est égal à n .

Remarque 4.1. On impose au système linéaire de \mathbb{R}^n décrit par l'équation (3.2) une stabilité (i.e. $\lim A^v = 0$ quand $v \rightarrow \infty$).

Dans toute la suite, on fera l'hypothèse de la stabilité du système linéaire (3.2) pour $v = n$.

Définition 4.2. La commande $u(t)$ et sa trajectoire $x(t)$ sont dites admissibles si elles vérifient les contraintes du problème (3.7).

Une commande admissible u_0 est dite **optimale** si :

$$\max J(u) = J(u_0).$$

Une commande admissible u^ε est dite ε -**optimale** si :

$$J(u_0) - J(u^\varepsilon) < \varepsilon.$$

Définition 4.3. Le support contrôle $\{u, \tau_B\}$ est dit **non dégénéré** si :

$$f_*(t) < u(t) < f^*(t).$$

4.2.3 Calcul de l'accroissement de la fonctionnelle

Considérons un support contrôle non dégénéré $\{u, \tau_B\}$, et une autre commande admissible $\bar{u} = u + \Delta u$, avec $\bar{x} = x + \Delta x$ sa trajectoire correspondante.

L'accroissement de la fonctionnelle est donné par :

$$\Delta J = J(\bar{u}) - J(u) = - \sum_{t \in T} \Delta(t) \cdot \Delta u(t).$$

Le maximum de cet accroissement sous les contraintes :

$$f_*(t) - u(t) \leq \Delta u(t) \leq f^*(t) - u(t), \quad \forall t \in T,$$

est atteint pour :

$$\Delta u(t) = \begin{cases} f^*(t) - u(t), & \text{si } \Delta(t) < 0, \\ f_*(t) - u(t), & \text{si } \Delta(t) > 0, \end{cases} \quad t \in T.$$

La valeur maximale de cet accroissement est égale à :

$$\beta = \beta(u, \tau_B) = \sum_{t \in T_H^+} \Delta(t)(u(t) - f_*(t)) + \sum_{t \in T_H^-} \Delta(t)(u(t) - f^*(t)), \quad (3.8)$$

appelée *valeur de suboptimalité* du support contrôle $\{u, \tau_B\}$,

où :

$$T_H^+ = \{t \in T_H \mid \Delta(t) \geq 0\}, \quad T_H^- = \{t \in T_H \mid \Delta(t) \leq 0\}.$$

Il en résulte que, pour $\bar{u} = u_0$, on a :

$$0 \leq J(u_0) - J(u) \leq \beta(u, \tau_B). \quad (3.9)$$

4.2.4 Critère d'optimalité

Théorème. [13]

Les relations suivantes :

$$\begin{cases} u(t) = f^*(t), & \text{si } \Delta(t) \leq 0, \\ u(t) = f_*(t), & \text{si } \Delta(t) \geq 0, \\ f_*(t) \leq u(t) \leq f^*(t), & \text{si } \Delta(t) = 0, \end{cases} \quad t \in T_H, \quad (3.10)$$

sont suffisantes, et dans le cas de non dégénérescence, elles sont aussi nécessaires pour l'optimalité du support contrôle $\{u, \tau_B\}$.

4.2.5 Critère de suboptimalité

Théorème. [13]

Soit $\varepsilon > 0$ donné. Pour que la commande u soit ε -optimale, il suffit de trouver un support τ_B tel que la valeur de suboptimalité vérifie l'inégalité suivante :

$$\beta(u, \tau_B) \leq \varepsilon.$$

4.2.6 Méthode de résolution

Dans ce cas, il suffit de résoudre le problème (3.7) par la méthode adaptée. Une fois la commande optimale obtenue, on détermine la trajectoire $x(t)$ correspondante à partir de l'expression suivante :

$$x(t+h) = Dx(t) + du(t), \quad x(t_0) = x_0.$$

Algorithme de résolution

1. Tester la commandabilité du système. En cas positif, continuer.
2. Sinon, arrêter le processus : le système n'est pas commandable.
3. Tester la stabilité du système. En cas positif, continuer
4. Sinon, arrêter le processus : le système n'est pas stable.
5. Poser le problème (3.6) à résoudre.
6. Réécrire le problème (3.6) sous la forme (3.7).
7. Résoudre le problème (3.7) par la méthode adaptée.
8. Calculer la trajectoire $x(t)$ grâce à la solution obtenue et à la formule :

$$x(t+h) = Dx(t) + du(t), \quad x(t_0) = x_0.$$

Exemple

Soit à résoudre :

$$\left\{ \begin{array}{l} J(u) = c'x(t^*) \longrightarrow \max_u, \\ \dot{x} = \frac{dx}{dt} = Ax + bu, \quad x(t_0) = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}, \\ Hx(t^*) = g, \\ |u(t)| \leq 2.5, \quad t \in T = [0; 6]. \end{array} \right.$$

avec :

$$c' = (2 \quad -1 \quad 2 \quad 1 \quad 0 \quad 4), \quad b' = (0 \quad 0 \quad 2 \quad 2 \quad 2 \quad 2), \quad g' = (0.1 \quad 0.18 \quad 0.4 \quad 0.5 \quad 0.3),$$

$$H = \begin{pmatrix} -2 & 1 & -2 & 2 & 0 & 0 \\ -1 & -1 & 0 & -1 & 3 & 1 \\ 2 & 1 & -2 & 1 & -2 & -2 \\ -2 & 1 & 3 & -2 & 3 & 1 \\ -4 & 0 & -2 & 2 & -1 & 0 \end{pmatrix}, \quad A = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

On a $\text{rang}(b, Ab, A^2b, \dots, A^{n-1}b) = 6$, donc on continue le processus. De plus $A^n = 0$.

Il en résulte que :

$$F(t) = \begin{pmatrix} 1 & t & \frac{1}{2}t^2 & \frac{1}{6}t^3 & \frac{1}{24}t^4 & \frac{1}{120}t^5 \\ 0 & 1 & t & \frac{1}{2}t^2 & \frac{1}{6}t^3 & \frac{1}{24}t^4 \\ 0 & 0 & 1 & t & \frac{1}{2}t^2 & \frac{1}{6}t^3 \\ 0 & 0 & 0 & 1 & t & \frac{1}{2}t^2 \\ 0 & 0 & 0 & 0 & 1 & t \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \quad F^{-1}(t) = \begin{pmatrix} 1 & -t & \frac{1}{2}t^2 & -\frac{1}{6}t^3 & \frac{1}{24}t^4 & -\frac{1}{120}t^5 \\ 0 & 1 & -t & \frac{1}{2}t^2 & -\frac{1}{6}t^3 & \frac{1}{24}t^4 \\ 0 & 0 & 1 & -t & \frac{1}{2}t^2 & -\frac{1}{6}t^3 \\ 0 & 0 & 0 & 1 & -t & \frac{1}{2}t^2 \\ 0 & 0 & 0 & 0 & 1 & -t \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

Avec un pas de quantification $h = 1$, on a :

$$D = e^{Ah} = \begin{pmatrix} 1 & 1 & 0.5 & 0.166 & 0.0416 & 0.01041 \\ 0 & 1 & 1 & 0.5 & 0.166 & 0.0416 \\ 0 & 0 & 1 & 1 & 0.5 & 0.166 \\ 0 & 0 & 0 & 1 & 1 & 0.5 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

et

$$d = D \int_0^h F^{-1}(s)b ds = \begin{pmatrix} 0.2180 \\ -0.7166 \\ 1.7083 \\ 1.666 \\ 1.5 \\ 1 \end{pmatrix}.$$

Dans ce cas, le problème devient :

$$\begin{aligned} \text{Maximiser} \quad & 574.9943 u(0) + 304.6499 u(1) + 152.8055 u(2) + 72.7944 u(3) \\ & + 33.9499 u(4) + 17.6055 u(5) + 169.5, \end{aligned}$$

sous les contraintes :

$$\begin{aligned} & - 436.9943 u(0) - 202.6499 u(1) - 80.8055 u(2) - 24.7944 u(3) \\ & - 3.9499 u(4) + 0.3944 u(5) = 111.7, \end{aligned}$$

$$\begin{aligned} & - 427.0638 u(0) - 209.5582 u(1) - 88.5527 u(2) - 27.7138 u(3) \\ & - 1.7083 u(4) + 5.7972 u(5) = 117.98, \end{aligned}$$

$$\begin{aligned} & 489.5276 u(0) + 212.5166 u(1) + 73.0055 u(2) + 12.3277 u(3) \\ & - 8.1833 u(4) - 11.1944 u(5) = -115.2, \end{aligned}$$

$$\begin{aligned} & - 73.0777 u(0) + 17.4333 u(1) + 42.4444 u(2) + 38.6222 u(3) \\ & + 26.6333 u(4) + 15.1444 u(5) = -14.9, \end{aligned}$$

$$\begin{aligned} & - 1124.688 u(0) - 544.9998 u(1) - 235.3110 u(2) - 86.2888 u(3) \\ & - 24.5999 u(4) - 4.9110 u(5) = 301.5, \end{aligned}$$

$$\text{avec} \quad - 2.5 \leq u(i) \leq 2.5, \quad \text{pour } i = 0, \dots, 5.$$

La solution de départ est :

$$u' = (-0.25002 \quad 0.7126 \quad -1.9159 \quad 0 \quad 1.9149 \quad -1.0086), \quad J_B = \{1, 2, 3, 4, 6\}.$$

La solution optimale est :

$$\bar{u} = (-0.30208 \quad 0.98806 \quad -2.5 \quad -0.6292 \quad 1.55304 \quad -0.9081), \quad J_B = \{1, 2, 3, 5, 6\}.$$

La trajectoire $x(t)$ est donnée par la formule :

$$x(t+h) = Dx(t) + du(t).$$

ce qui donnera :

	$x(0)$	$x(1)$	$x(2)$	$x(3)$	$x(4)$	$x(5)$
Ligne 1	0	-0.1234	-0.5938	-0.01203	0.1411	0.3622
Ligne 2	0	-0.3913	-0.0619	0.6846	-0.3347	-0.28706
Ligne 3	0	-0.8654	2.1162	-1.2249	-0.9656	1.3985
Ligne 4	0	-0.5069	3.0782	-0.6153	-1.5063	0.5709
Ligne 5	0	0.0937	3.4537	-1.6742	-2.4147	0.8747
Ligne 6	1	0.3958	2.3719	-2.62804	-1.3696	1.7364

Chapitre 5

Le Contrôle Optimal en multi-objectifs

5.1 introduction :

Dans ce chapitre, nous posons le cadre du problème de contrôle optimal multi-objectifs pour un système dynamique linéaire discret soumis à des contraintes sur la commande et l'état final. Nous présentons la dynamique du système, les contraintes associées ainsi que le critère multi-objectif à maximiser. Cette formulation permet de modéliser des systèmes où plusieurs objectifs doivent être pris en compte simultanément tout en respectant les limitations physiques ou opérationnelles.

5.2 Position du problème

Considérons le système dynamique linéaire suivant, défini sur l'intervalle de temps $T = [t_0, t^*]$:

$$\frac{dx(t)}{dt} = A(t)x(t) + b(t)u(t), \quad x(t_0) = x_0, \quad (4.1)$$

où :

- $x(t) \in \mathbb{R}^n$ est le vecteur d'état décrivant la trajectoire du système à l'instant t ,
- $u(t) \in \mathbb{R}$ est la commande d'entrée du système à l'instant t ,
- $A(t) \in \mathbb{R}^{n \times n}$ est une matrice représentant la dynamique du système,
- $b(t) \in \mathbb{R}^n$ est un vecteur colonne,
- x_0 est la position initiale du système à l'instant $t = t_0$,
- La commande est contrainte par :

$$f_*(t) \leq u(t) \leq f^*(t), \quad \forall t \in [t_0, t^*],$$

où $f_*(t)$ et $f^*(t)$ sont deux fonctions définies sur l'intervalle.

À l'instant final $t = t^*$, le système est soumis à une contrainte de sortie :

$$Hx(t^*) = g, \quad (4.2)$$

où :

- $H \in \mathbb{R}^{m \times n}$ est une matrice de rang $m \leq n$,
- $g \in \mathbb{R}^m$ est un vecteur fixé.

Le critère de qualité multi-objectifs à maximiser est donné par :

$$J(u) = \max_u (c'_k x(t^*) + \alpha_k),$$

où :

- $c_k \in \mathbb{R}^n$ sont des vecteurs de pondération,
- $\alpha_k \in \mathbb{R}$ sont des scalaires associés à chaque objectif,
- c'_k désigne la transposée du vecteur c_k ,
- $K = \{1, \dots, k\}$ est l'ensemble des indices correspondant aux différentes fonctions "objectifs".

On introduit également les ensembles d'indices suivants :

- $I = \{1, \dots, m\}$: ensemble des indices des lignes,
- $J = \{1, \dots, n\}$: ensemble des indices des colonnes.

L'objectif est donc de déterminer une commande admissible $u(t)$ qui maximise la fonctionnelle $J(u)$ (au sens de pareto), tout en satisfaisant la dynamique du système, les contraintes sur la commande et la contrainte finale sur la sortie.

5.3 Modèle mathématique du problème

L'objectif est de déterminer une commande admissible $u(t)$ qui maximise les critères de performances au sens de pareto, tout en respectant la dynamique du système.

$$\left\{ \begin{array}{l} J(u) = \max_u (c'_k x(t^*) + \alpha_k) \\ \text{sous les contraintes :} \\ \frac{dx(t)}{dt} = A(t)x(t) + b(t)u(t), \quad x(t_0) = x_0 \quad (\text{C1}) \\ Hx(t^*) = g \\ f_*(t) \leq u(t) \leq f^*(t), \quad \forall t \in [t_0, t^*] \end{array} \right.$$

Le système (4.1), sous l'effet d'une commande u , admet pour solution l'expression donnée par la formule de Cauchy :

$$x(t) = F(t)F^{-1}(\tau)x(\tau) + \int_{\tau}^t F(t)F^{-1}(s)b(s)u(s) ds, \quad \text{pour } t \geq \tau \geq 0, \quad (4.3)$$

où $F(t) = \exp(A(t))$, avec $F(0) = I_n$, et $F(t)$ est la solution matricielle (matrice carrée) de l'équation différentielle suivante :

$$\dot{F}(t) = A(t)F(t), \quad F(0) = I_n.$$

Lorsque la commande u est de type impulsive, l'expression (4.3) devient :

$$x(i+1) = x(\tau_{i+1}) = F(\tau_{i+1})F^{-1}(\tau_i)x(\tau_i) + \int_{\tau_i}^{\tau_{i+1}} F(\tau_{i+1})F^{-1}(s)b(s)u(i) ds,$$

où $u_i = u(\tau_i)$, $\tau_i \in T$, avec $i = 0, \dots, N$, et :

$$T = \bigcup_{i=0}^N [\tau_i, \tau_{i+1}], \quad \tau_0 = t_0, \quad \tau_{N+1} = t^*, \quad \tau_{i+1} - \tau_i = h > 0.$$

En effectuant une translation des instants τ_i et τ_{i+1} , on peut réécrire :

$$x(i+1) = F(h)(x(i) + \int_0^h F^{-1}(s)bu(i)ds).$$

En posant :

$$D = \exp(Ah) = \sum_{n=0}^{\infty} \frac{(Ah)^n}{n!},$$

$$d = D \cdot \int_0^h F^{-1}(s)b(s) ds,$$

on aboutit au système équivalent au système (C1) suivant :

$$\begin{cases} J(u) = c'_k x(t^*) + \alpha_k \longrightarrow \max_{u(t)}, \\ x(t+h) = Dx(t) + d(t)u(t), \quad x(t_0) = x_0, \\ Hx(t^*) = g, \\ f_*(t) \leq u(t) \leq f^*(t), \quad \forall t \in T. \end{cases} \quad (C2)$$

avec $x(t^*)$ vérifiant la propriété suivante :

$$x(t^*) = D^{\frac{t^*}{h}} x_0 + \sum_{t \in T} D^{\frac{t_1 - t - h}{h}} du(t).$$

Cette propriété nous permet d'établir une formulation équivalente du problème (C2), que l'on note (C3), donnée par :

$$\left\{ \begin{array}{l} J(u) = c'_k \left(\sum_{t \in T} D^{\frac{t^* - t - h}{h}} du(t) + D^{\frac{t^*}{h}} x_0 \right) + \alpha_k \longrightarrow \max_u, \\ H \sum_{t \in T} D^{\frac{t^* - t - h}{h}} du(t) = g - HD^{\frac{t^*}{h}} x_0, \\ f_*(t) \leq u(t) \leq f^*(t), \quad \forall t \in T, \end{array} \right. \quad (C3)$$

où $x(t+h) = Dx(t) + du(t)$ avec $x(t_0) = x_0$.

Remarque 5.1. On peut résoudre (C3) en faisant un changement de variables sur la commande u de telle sorte que les bornes inférieures pour la commande soient nulles, sans perte de généralité. En général on prend $t_0 = 0$

5.4 MÉTHODE DE RÉOLUTION

Comme pour le cas du problème mono-objectif, il suffit de résoudre le problème (C3) par la méthode adaptée (ou le simplexe avec ses variantes). Une fois les commandes pareto-optimales obtenues, on cherchera les trajectoires $x(t)$ correspondantes à partir de l'expression suivante :

$$x(t+h) = Dx(t) + du(t) \quad ; \quad x(t_0) = x_0$$

ce qui implique les variantes suivantes d'algorithmes.

5.4.1 ALGORITHME DE RÉOLUTION (variante 1)

- Tester la commandabilité du système, dans le cas positif continuer.
- Sinon, arrêter le processus : le système n'est pas commandable.
- Tester la stabilité du système, dans le cas positif continuer.
- Sinon, arrêter le processus : le système n'est pas stable.
- Poser le problème (C1) à résoudre.
- Écrire (C1) sous forme (C3).
- Reformuler (C3) sous forme standard .
- Résoudre par la méthode du Simplexe multi-objectifs.

- Calcul des trajectoires $x(t)$ grâce aux solutions optimales (pareto) obtenues et la formule :

$$x(t+h) = Dx(t) + du(t) \quad ; \quad x(t_0) = x_0$$

5.4.2 ALGORITHME DE RÉOLUTION (variante 2)

- Tester la commandabilité du système, dans le cas positif continuer.
- Sinon, arrêter le processus : le système n'est pas commandable.
- Tester la stabilité du système, dans le cas positif continuer.
- Sinon, arrêter le processus : le système n'est pas stable.
- Poser le problème (C1) à résoudre.
- Écrire (C1) sous forme (C3).
- Résoudre (C3) par la méthode Adaptée et par le parcours de toutes les bases réalisables.
- Calcul des trajectoires $x(t)$ grâce aux solutions optimales (pareto) obtenues et à la formule :

$$x(t+h) = Dx(t) + du(t) \quad ; \quad x(0) = x_0$$

5.5 EXEMPLE D'APPLICATION

Soit à résoudre :

$$\left\{ \begin{array}{l} J(u) = (c'_k x(t_1) + \alpha_k) \longrightarrow \max_u \\ \frac{dx(t)}{dt} = Ax(t) + bu(t), \quad x(0) = 0 \\ Hx(t_1) = g \\ |u(t)| \leq 1.2, \quad t \in [0, 4] \end{array} \right.$$

avec

$$c'_1 = [0 \quad 2 \quad 3 \quad 0], \quad \alpha'(K) = (0, 0)$$

$$c'_2 = [0 \quad -3 \quad -2 \quad 0],$$

$$g' = (1; 2), \quad b = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}$$

$$H = \begin{pmatrix} 1 & -2 & 1 & 2 \\ 1 & 3 & 1 & 1 \end{pmatrix}$$

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Rang $[b, Ab, A^2b, \dots, A^{n-1}b] = 4 \Rightarrow$ on continue le processus

$A^n = [0] \Rightarrow$ Il en résulte que :

$$F(t) = \begin{pmatrix} 1 & t & \frac{t^2}{2} & \frac{t^3}{6} \\ 0 & 1 & t & \frac{t^2}{2} \\ 0 & 0 & 1 & t \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$F^{-1}(t) = \begin{pmatrix} 1 & -t & \frac{t^2}{2} & \frac{-t^3}{6} \\ 0 & 1 & -t & \frac{t^2}{2} \\ 0 & 0 & 1 & -t \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Avec un pas de quantification $h = 1$, alors :

$$D = \exp(Ah) = \begin{pmatrix} 1 & 1 & 0.5 & 0.166 \\ 0 & 1 & 1 & 0.5 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

et

$$d = D \int_0^h F^{-1}(s)b ds = \begin{pmatrix} 0.20834 \\ -0.66667 \\ 1.5 \\ 1 \end{pmatrix}$$

Dans ce cas le problème devient compte tenu de (C3)

$$\begin{pmatrix} 32,83333333 u(0) + 21,83333333 u(1) + 12,83333333 u(2) + 5,83333333 u(3) \\ -38 u(0) - 24 u(1) - 13 u(2) - 5 u(3) \end{pmatrix} \rightarrow \max_u$$

Avec :

$$-1,2 \leq u(0) \leq 1,2$$

$$-1,2 \leq u(1) \leq 1,2$$

$$-1,2 \leq u(2) \leq 1,2$$

$$-1,2 \leq u(3) \leq 1,2$$

et

$$0.6246 u(0) + 0.0416 u(1) + 0.9583 u(2) + 2.375 u(3) = 1$$

$$47.958 u(0) + 27.375 u(1) + 13.292 u(2) + 4.7084 u(3) = 2$$

La solution de départ pour le premier objectif est : $u = (0 ; -0,442 ; 1.0627 ; 0)$

$$J_B = \{2, 3\}$$

La solution optimale pour le premier objectif est : $\bar{u} = (-0,993 ; 1,2 ; 1,2 ; 0,1771)$

$$J_B = \{1, 4\}$$

La solution de départ pour le deuxième objectif est : $u = (0 ; -0,442 ; 1.0627 ; 0)$

$$J_B = \{2, 3\}$$

La solution optimale pour le deuxième objectif est : $\bar{u} = (-0,9939 ; -1,2 ; -1,2 ; 0,6647)$

$$J_B = \{1, 4\}$$

compte tenu du parcours des bases du problème on aboutit aux solutions suivantes :

au niveau du premier objectif on aboutit aux solutions suivantes :

$$g_1 = \begin{bmatrix} 0 \\ -0.442942378933553 \\ 1.06273716412301 \\ 0 \end{bmatrix} \quad g_2 = \begin{bmatrix} -0.04410058613904 \\ -0.424744056893694 \\ 1.2 \\ -0.04410058613904 \end{bmatrix} \quad g_3 = \begin{bmatrix} -0.993240733727625 \\ 1.2 \\ 1.2 \\ 0.177169169903211 \end{bmatrix}$$

au niveau du deuxième objectif on aboutit aux solutions suivantes :

$$h_1 = \begin{bmatrix} 0 \\ -0.442942378933553 \\ 1.06273716412301 \\ 0 \end{bmatrix} \quad h_2 = \begin{bmatrix} 0.726985090894097 \\ -0.742936323697813 \\ -1.2 \\ 0.726985090894097 \end{bmatrix} \quad h_3 = \begin{bmatrix} 0.993991751630707 \\ -1.2 \\ -1.2 \\ 0.664738752634201 \end{bmatrix}$$

processus de calcul :

$$c'_1 * h_1 = 3.9673$$

$$c'_2 * h_1 = -3.1850$$

$$c'_1 * h_2 = -3.5103$$

$$c'_2 * h_2 = 2.1701$$

$$c'_1 * h_3 = -5.0858$$

$$c'_2 * h_3 = 3.3046$$

$$c'_1 * g_2 = 4.4210$$

$$c'_2 * g_2 = -3.5098$$

$$c'_1 * g_3 = 10.0216$$

$$c'_2 * g_3 = -7.5427$$

5.6 Conclusion

Les solutions suivantes : g_1, g_2, g_3, h_2, h_3 sont des solutions pareto optimales.

Chapitre 6

Implémentation informatique

6.1 Introduction au logiciel LINGO

La programmation informatique regroupe un ensemble d'outils et de techniques visant à résoudre des problèmes mathématiques à l'aide d'un ordinateur. Elle permet, entre autres, de déterminer des solutions optimales à différents types de problèmes.

Les méthodes de résolution mathématique nécessitent souvent un grand nombre de calculs complexes. Leur exécution manuelle est donc fastidieuse, voire impraticable. C'est pourquoi le recours à un logiciel spécialisé comme **LINGO** s'avère judicieux. LINGO est un outil puissant dédié à la résolution de problèmes d'optimisation, qu'ils soient *linéaires*, *non linéaires*, *convexes* ou *non convexes*, entre autres.

Ce logiciel dispose d'une interface conviviale et d'un ensemble riche de commandes permettant une modélisation souple et efficace.

6.1.1 Présentation du logiciel LINGO

LINGO facilite la formulation rapide et intuitive des problèmes d'optimisation, qu'ils soient linéaires, non linéaires ou à variables entières. Grâce à ses outils de modélisation avancés, les modèles peuvent être exprimés de manière claire à l'aide de notations simples telles que les sommes et les variables indicées.

La méthode de modélisation dans LINGO reste proche de celle utilisée à la main (papier et crayon), tout en étant plus flexible, réutilisable et facilement modifiable.

LINGO est équipé de plusieurs solveurs, adaptés à différents types de modèles :

- Solveur direct
- Solveur indirect
- Solveur linéaire

- Solveur non linéaire
- Méthode de séparation et évaluation (pour les problèmes entiers)

Parmi ses principaux atouts, LINGO offre :

- La confirmation de l'optimalité globale d'une solution trouvée
- Une réduction significative du temps de calcul
- Une résolution améliorée de divers types de problèmes
- Une capacité de décomposition pour les modèles complexes avec sous-modèles

LINGO est livré avec un ensemble de solveurs performants couvrant les domaines suivants :

- Optimisation linéaire
- Optimisation non linéaire (convexe et non convexe)
- Optimisation en nombres entiers
- Optimisation sous contraintes

L'un des grands avantages de LINGO réside dans sa capacité à **choisir automatiquement le solveur le plus adapté** en fonction de la nature du problème formulé, sans intervention manuelle de l'utilisateur.

Un modèle d'optimisation dans LINGO comprend généralement trois éléments essentiels :

1. **La fonction objectif** : une formule qui décrit précisément ce que le modèle cherche à optimiser.
2. **Les variables de décision** : les quantités que le modèle ajuste pour atteindre l'objectif.
3. **Les contraintes** : des équations ou inéquations imposant des limites aux valeurs que peuvent prendre les variables.

Fonctions couramment utilisées dans un modèle LINGO : LINGO propose plusieurs fonctions intégrées facilitant la modélisation et la formulation de contraintes ou d'expressions objectives. Parmi les plus utilisées, on retrouve :

- **@FOR** : utilisée pour générer des contraintes en boucle, à l'aide d'ensembles ou d'index.
- **@SUM** : permet le calcul de sommes sur des ensembles d'indices.
- **@MAX** : utilisée pour exprimer la recherche du maximum parmi plusieurs expressions.

- @MIN : utilisée pour exprimer la recherche du minimum parmi plusieurs expressions.

Types de variables dans LINGO : Par défaut, toutes les variables dans un modèle LINGO sont considérées comme **continues** et **non négatives**. Toutefois, LINGO permet de spécifier différents types de variables selon la nature du problème. Les fonctions de typage les plus utilisées sont :

- GIN(x) : force la variable x à prendre des **valeurs entières positives**.
- BIN(x) : impose à la variable x de ne prendre que les valeurs binaires **0 ou 1**.
- FREE(x) : autorise x à prendre **toute valeur réelle**, positive ou négative.
- BND(lower, x , upper) : définit une **borne inférieure et supérieure** pour la variable x , où lower et upper sont les limites.

Syntaxe générale :

- Déclaration simple : GIN(x) ; ou BIN(x) ; ou FREE(x) ;
- Déclaration bornée : BND(1, x , 10) ; signifie que $x \in [1, 10]$

6.1.2 Interface du logiciel

Lors du premier lancement de LINGO après son installation, certaines versions peuvent exiger l'entrée d'une **clé de licence**. Dans ce cas, une fenêtre de saisie s'affiche à l'écran afin de valider l'utilisation du logiciel. Une fois cette étape franchie, l'utilisateur accède à l'interface principale de modélisation, qui permet :

- La saisie et l'édition des modèles ;
- L'exécution des solveurs ;
- L'affichage des résultats et diagnostics.

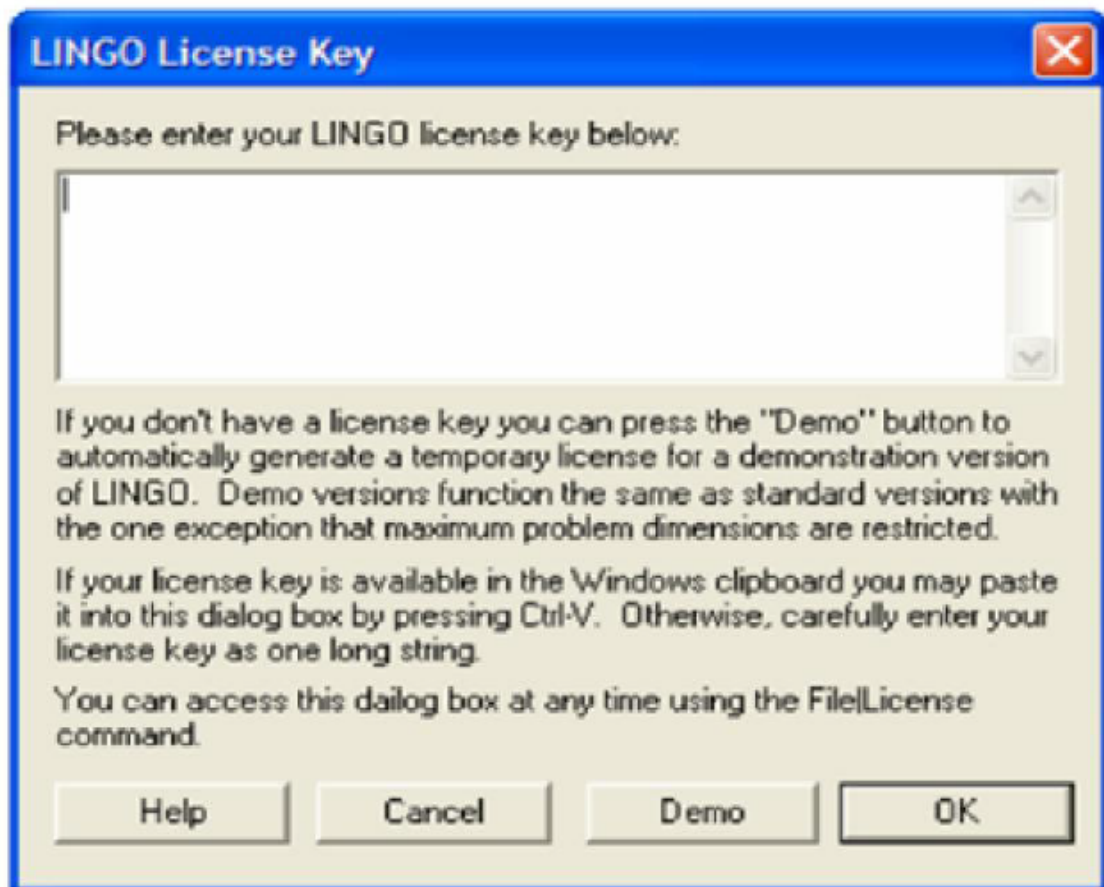


FIGURE 6.1 – Fenêtre de saisie de la licence du logiciel LINGO

Une fois la clé saisie, il suffit de cliquer sur *OK* pour valider l'activation. Si l'utilisateur ne dispose pas d'une clé de licence, il est néanmoins possible d'utiliser LINGO en mode « Demo ». Ce mode permet d'accéder aux fonctionnalités du logiciel de manière limitée et est valable pour une période d'essai de 30 jours.

Juste après la barre des menus de LINGO, une fenêtre dédiée à la saisie du modèle du problème à résoudre apparaît. Cette zone permet à l'utilisateur d'entrer directement les fonctions, les contraintes et les paramètres nécessaires à la modélisation mathématique.

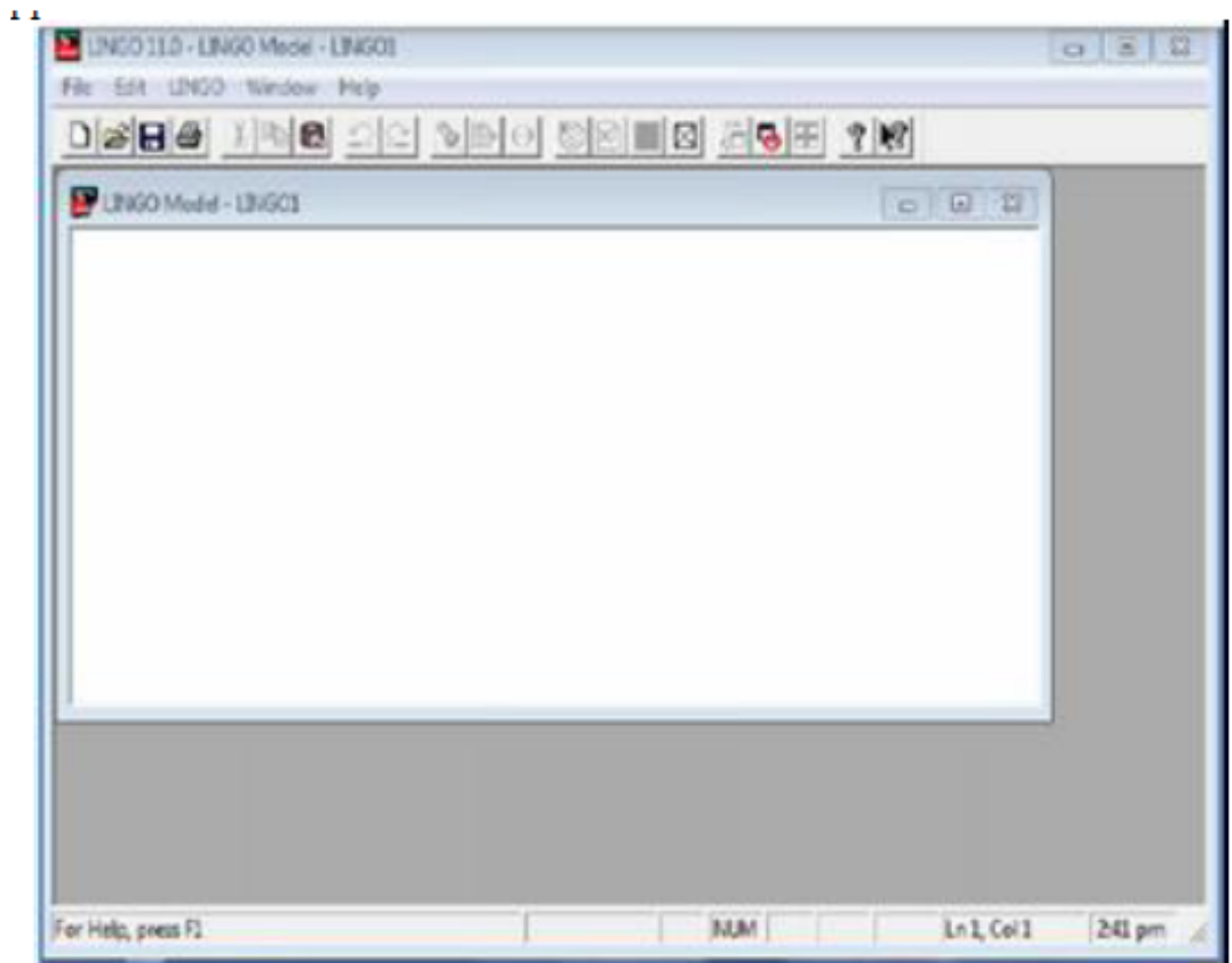
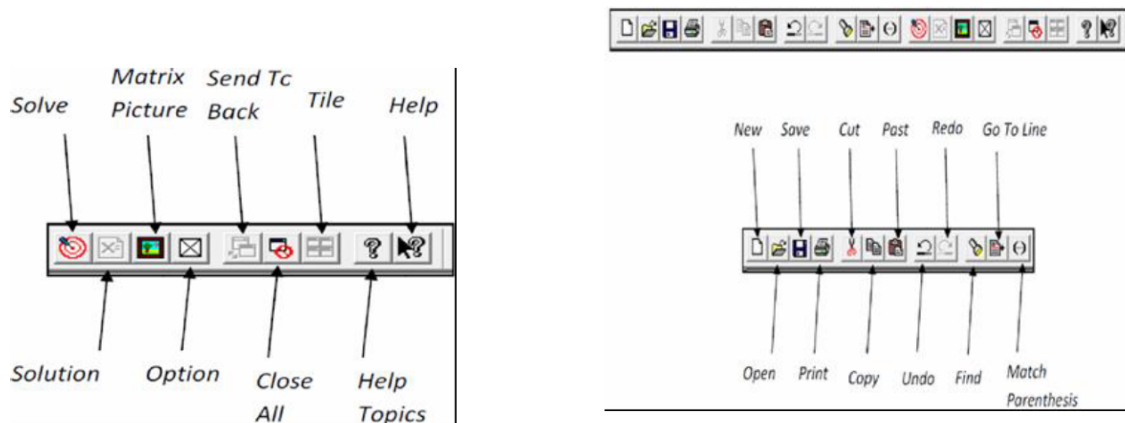


FIGURE 6.2 – La barre des menus du logiciel LINGO

La barre des outils de LINGO se présente comme suit :



Signification :

New : Ouvrir une nouvelle fenêtre.

Open : Ouvrir un fichier enregistré.

Save : Enregistrer le modèle en cours.

Print : Imprimer la fenêtre active.

Cut : Couper le texte sélectionné pour le déplacer.

Copy : Copier le texte sélectionné.

Paste : Coller le contenu précédemment copié ou coupé.

Undo : Annuler la dernière action effectuée.

Redo : Rétablir l'action annulée.

Find : Rechercher un mot ou une expression dans le document.

Go To Line : Déplacer le curseur à une ligne spécifique.

Match Parenthesis : Localiser la parenthèse correspondante à celle sélectionnée.

Solve : Lancer la résolution du modèle.

Solution : Afficher la fenêtre de rapport des résultats du modèle courant.

Options : Accéder aux paramètres du logiciel.

Send To Back : Envoyer la fenêtre active derrière les autres fenêtres ouvertes.

Close All : Fermer toutes les fenêtres ouvertes.

Tile : Organiser les fenêtres ouvertes pour qu'elles occupent la même zone dans l'interface.

Help Topics : Ouvrir le manuel utilisateur de LINGO.

Help : Accéder à l'aide en ligne de LINGO.

6.1.3 Exemple d'application

Soit le problème d'optimisation linéaire (5.1) :

$$\begin{aligned} & \text{Maximiser } Z = 1X_1 + 2X_2 + 3X_3 + 4X_4 + 5X_5 + 6X_6 \\ & \text{sous les contraintes : } 5X_1 + 3X_2 + 2X_3 + 4X_4 + 5X_5 + 1X_6 = 2 \\ & \qquad \qquad \qquad 0X_1 + 1X_2 + 2X_3 + 1X_4 + 4X_5 + 5X_6 = 2 \\ & \qquad \qquad \qquad 2X_1 + 4X_2 + 4X_3 + 5X_4 + 1X_5 + 5X_6 = 3 \\ & \qquad \qquad \qquad 6X_1 + 5X_2 + 4X_3 + 3X_4 + 2X_5 + 1X_6 = 4 \\ & \text{avec les bornes : } -2 \leq X_1 \leq 2 \\ & \qquad \qquad \qquad -3 \leq X_2 \leq 3 \\ & \qquad \qquad \qquad -4 \leq X_3 \leq 4 \\ & \qquad \qquad \qquad -5 \leq X_4 \leq 5 \\ & \qquad \qquad \qquad -6 \leq X_5 \leq 6 \\ & \qquad \qquad \qquad -7 \leq X_6 \leq 7 \end{aligned}$$

solution :

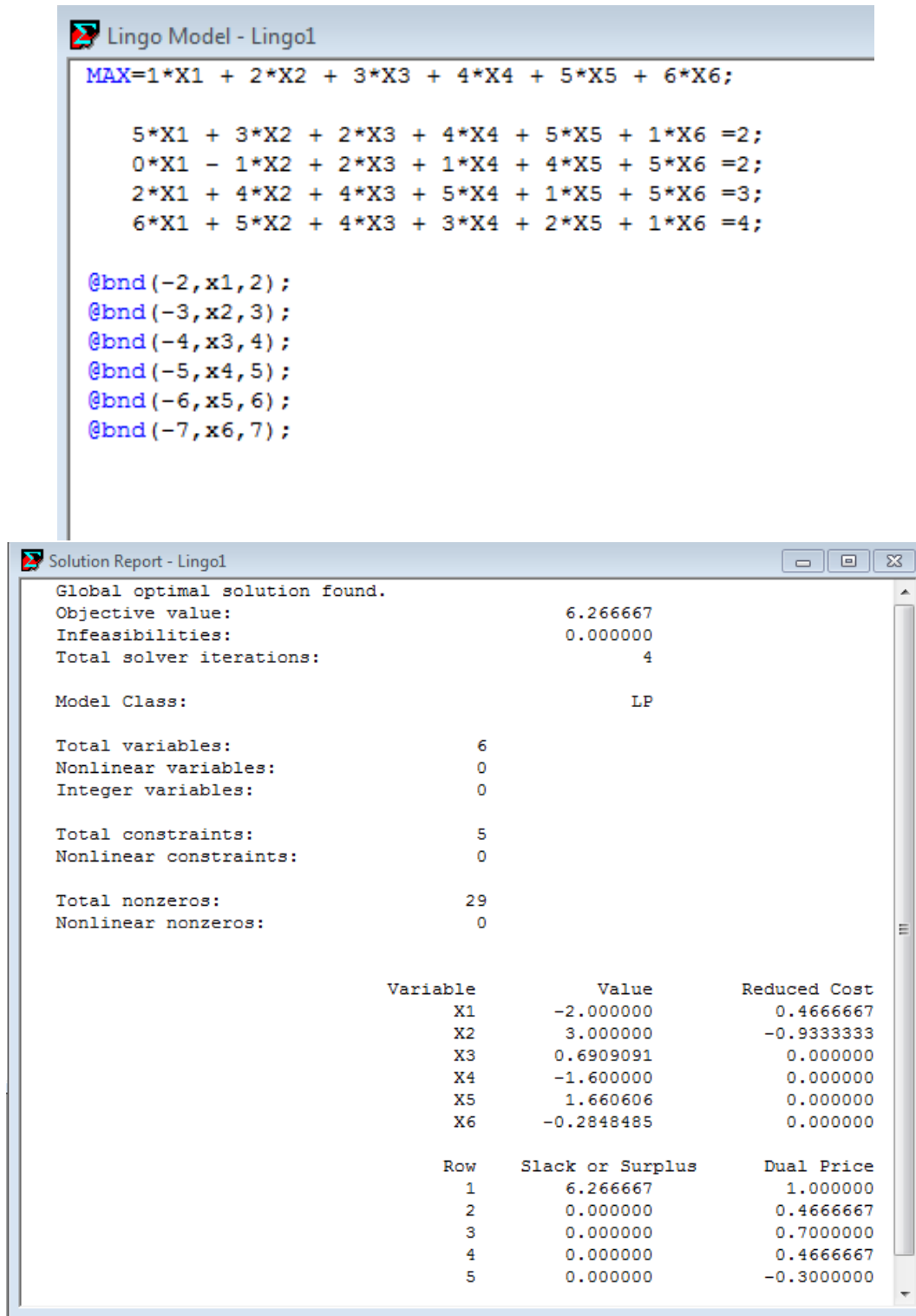


FIGURE 6.4 – Résultats obtenus sous LINGO

6.2 Définition du logiciel MATLAB

Le logiciel MATLAB (MATrix LABoratory) est spécialisé dans le domaine du calcul matriciel numérique. Tous les objets définis dans MATLAB sont donc principalement représentés sous forme de vecteurs, matrices ou tableaux de nombres. Un ensemble important d'opérateurs et de fonctions de base facilitent leur manipulation, notamment des opérations comme le produit matriciel, l'inversion (`inv`), la transposition (`'`), ou encore le calcul des valeurs propres, qui font partie de la bibliothèque standard.

D'autres fonctions servant à la création et à la manipulation de matrices et tableaux, telles que `diag`, `rand`, `ones`, `zeros`, `linspace`, sont également disponibles en grand nombre.

L'environnement MATLAB se présente sous la forme d'un espace de travail (*Workspace*), où un interpréteur de commandes exécute les opérations et fonctions MATLAB. Les sources de ces fonctions sont disponibles, écrites en langage MATLAB, ou parfois en C ou en Fortran. L'utilisateur peut les modifier, mais surtout s'en inspirer pour créer et ajouter ses propres fonctions.

Le langage MATLAB contient un minimum de structures de programmation (structures itératives, conditionnelles, sous-routines), bien qu'il reste assez rudimentaire. L'avantage est qu'il est simple et rapide à programmer, avec une grande tolérance (syntaxe simple, absence de définition stricte des types, etc.), ce qui permet un gain significatif en temps de développement.

Ainsi, l'ingénieur peut être plus efficace dans l'analyse d'un problème en se concentrant sur celui-ci plutôt que sur l'outil de résolution.

6.2.1 Description de l'interface MATLAB

La barre de titre

La fenêtre MATLAB est surmontée par une barre de titre, contenant à sa gauche une icône, et à sa droite les trois boutons suivants : « mise en icône », « minimisation/-maximisation » et « fermeture ».

La barre du menu

La barre du menu contient généralement cinq menus principaux :

- **File (Fichier)** : permet d'accéder à l'éditeur de programmes ;

- **Edit (Édition)** : permet de couper, copier, coller dans la ligne de commande, entre autres fonctions ;
- **Debug** : permet l'exécution d'un programme et diverses opérations de débogage ;
- **Window (Fenêtre)** : permet de passer entre les différentes rubriques ou fenêtres du logiciel ;
- **Help (Aide)** : donne accès au menu d'aide.

La barre d'outils

La barre d'outils propose souvent des raccourcis vers des fonctions contenues dans les menus. De gauche à droite, on trouve notamment :

- Ouvrir un nouveau fichier dans l'éditeur ;
- Rappeler un ancien fichier dans l'éditeur ;
- Couper ;
- Copier ;
- Coller ;
- Annuler ;
- Accéder à l'aide.

La fenêtre de commande

La fenêtre de commande se divise en deux zones distinctes :

- **La zone historique** : zone non modifiable dans laquelle on peut toutefois copier des parties ;
- **La zone de commande éditable** : cette zone permet de taper des commandes, qui seront exécutées lorsqu'on presse la touche `<return>` ou `<entrée>`.

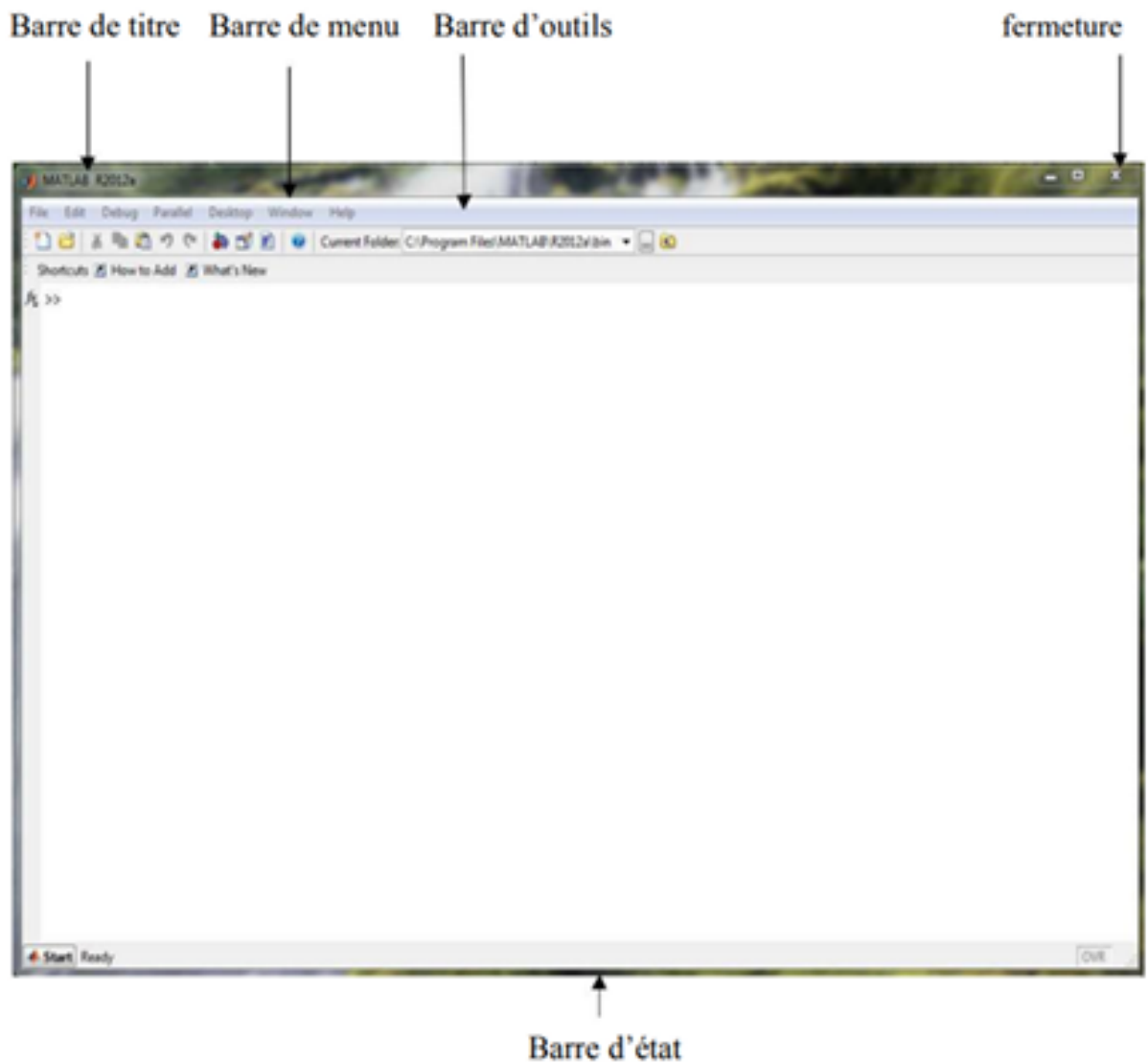


FIGURE 6.5 – La fenêtre principale du logiciel MATLAB

6.2.2 Méthode de travail

Édition et sauvegarde des fichiers MATLAB

Dans un premier temps, il est possible d'entrer les commandes une par une directement dans l'espace de travail, où elles sont interprétées immédiatement. Toutefois, pour un usage plus structuré et réutilisable, il est fortement recommandé de rédiger l'ensemble des instructions dans un éditeur de texte intégré à MATLAB, puis de sauvegarder ce script dans un fichier portant l'extension `.m`.

Ce fichier, appelé *script MATLAB*, pourra ensuite être exécuté en appelant simplement son nom dans la fenêtre de commande, ce qui permet de gagner en efficacité et en reproductibilité.

Aide en ligne

Outre l'aide classique fournie par l'interface Windows, MATLAB propose une aide en ligne très complète. Celle-ci est accessible pour chaque commande via l'instruction suivante :

```
help nom_de_commande
```

Par exemple, pour obtenir des informations sur la commande `plot`, il suffit de taper `help plot` dans la fenêtre de commande.

6.2.3 Création de fichiers de commande et de fonctions utilisateur

Fichiers de commande (scripts)

Un fichier de commande, ou *script*, est un fichier texte au format ASCII, portant l'extension `.m`, qui contient une suite d'instructions MATLAB. Ce type de fichier peut être exécuté directement en tapant simplement son nom (sans l'extension) dans l'espace de travail MATLAB.

Les scripts sont utiles pour automatiser l'exécution d'une série de commandes répétitives, mais toutes les variables qu'ils utilisent ou créent sont accessibles dans l'espace de travail global.

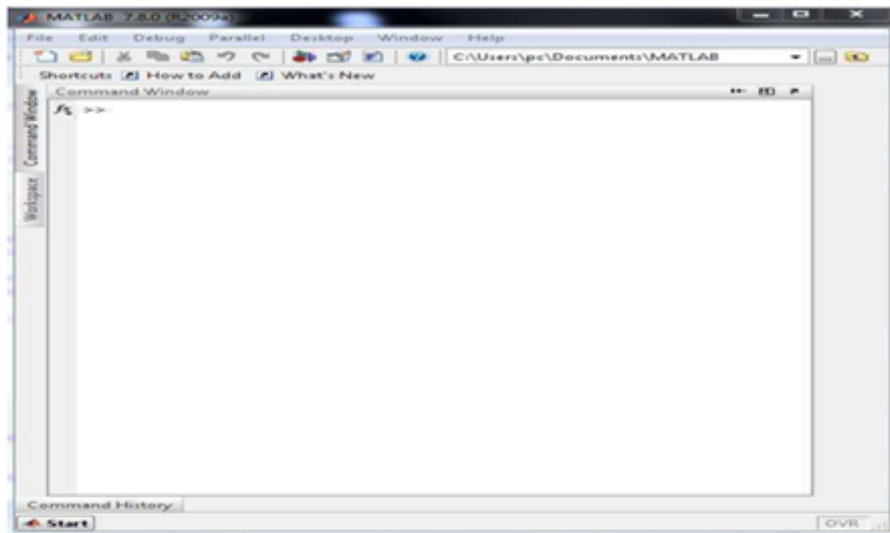
Fonctions utilisateur

L'utilisateur peut également enrichir MATLAB en y ajoutant ses propres fonctions. Pour cela, il suffit de créer un fichier nommé `nom_de_fonction.m`, contenant le code de la fonction. Ce fichier doit commencer par une ligne d'en-tête respectant la syntaxe suivante :

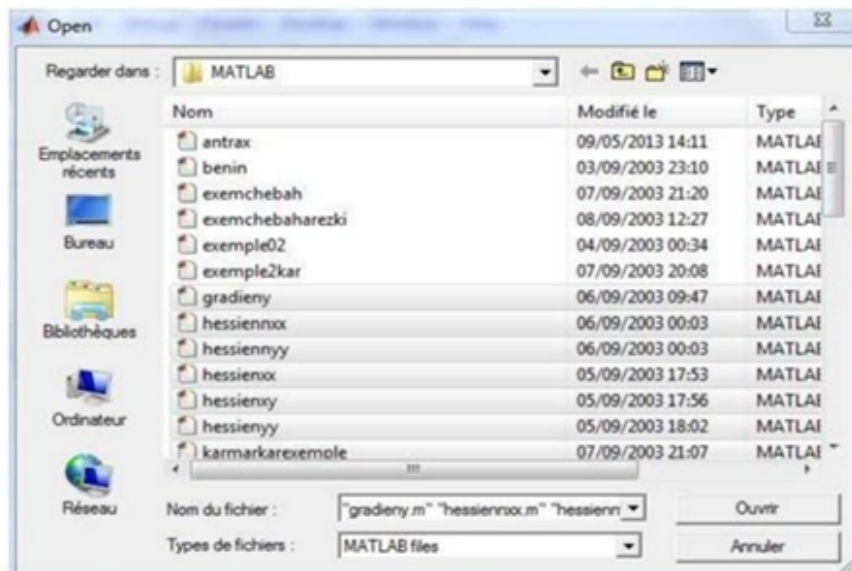
```
function [liste_sorties] = nom_de_fonction(liste_entrees)
```

Contrairement aux scripts, les variables définies à l'intérieur d'une fonction sont locales : elles ne sont pas accessibles depuis l'espace de travail principal, et n'affectent pas celui-ci.

Il est également recommandé d'ajouter une documentation à chaque fonction en insérant des commentaires (précédés du symbole `%`) immédiatement après la ligne d'en-tête. Ces commentaires peuvent être consultés via la commande `help nom_de_fonction`.



(a) La fenêtre d'édition de fichier



(b) La boîte de dialogue d'ouverture de fichiers

FIGURE 6.6 – Interface graphique de l'application : édition et ouverture de fichiers

6.2.4 Exemple d'application

En s'appuyant sur le modèle de la commande `linprog` de Matlab, on considère le problème précédent (5.1).

Remarque 6.1. Pour résoudre un problème de maximisation avec `linprog` (qui minimise), on utilise la transformation :

$$\max f(x) = -\min(-f(x)), \quad x \in D$$

Code Matlab :

```
f = [-1; -2; -3; -4; -5; -6];
```

```
A = [5  3  2  4  5  1;  
     0 -1  2  1  4  5;  
     2  4  4  5  1  5;  
     6  5  4  3  2  1];
```

```
b = [2; 2; 3; 4];
```

```
lb = [-2; -3; -4; -5; -6; -7];
```

```
ub = [2; 3; 4; 5; 6; 7];
```

```
[x, fval, exitflag, output, lambda] = linprog(f, [], [], A, b, lb, ub);
```

résultats Matlab :

```
Command Window
>> f = [ -1; -2; -3; -4; -5; -6];

>> A = [ 5 3 2 4 5 1 ;
0 -1 2 1 4 5 ;
2 4 4 5 1 5 ;
6 5 4 3 2 1 ] ;

>> b = [ 2 ; 2 ; 3 ; 4 ] ;
>> lb = [ -2; -3; -4; -5; -6; -7];
>> ub = [ 2 ; 3 ; 4 ; 5 ; 6 ; 7 ] ;

>> [ x,fval,exitflag,output,lambda ] = linprog(f , [ ] , [ ] , A, b , lb , ub )
Optimization terminated.

x =

    -2.0000
     3.0000
     0.6909
    -1.6000
     1.6606
    -0.2848

fval =

    -6.2667

exitflag =
```

```
exitflag =

     1

output =

    iterations: 5
    algorithm: 'large-scale: interior point'
    cgiterations: 0
    message: 'Optimization terminated.'
```

6.3 Qu'est-ce que *Visual Xpress* ?

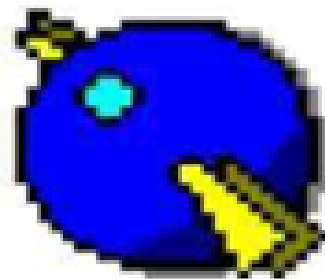
Visual Xpress est un logiciel convivial conçu pour la modélisation et la résolution de problèmes d'optimisation linéaire. Il propose un langage de modélisation symbolique, proche de l'écriture mathématique, facilitant ainsi la formulation des programmes linéaires.

Ce logiciel permet notamment :

- de modifier aisément les données du modèle,
- d'ajouter ou de supprimer des contraintes,
- de comparer deux modèles similaires,
- d'effectuer une analyse de sensibilité des solutions en fonction des données.

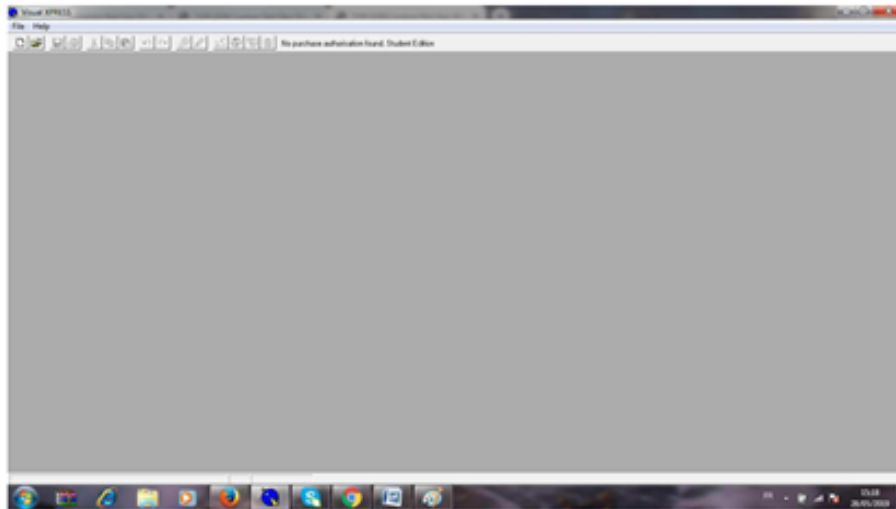
6.3.1 Présentation du logiciel

Avant toute utilisation, il est nécessaire d'installer le logiciel *Visual Xpress*. Une fois installé, le démarrage s'effectue en double-cliquant sur l'icône correspondante (voir Figure 6.7). L'espace de travail standard s'ouvre alors automatiquement, offrant un environnement intuitif pour la modélisation.

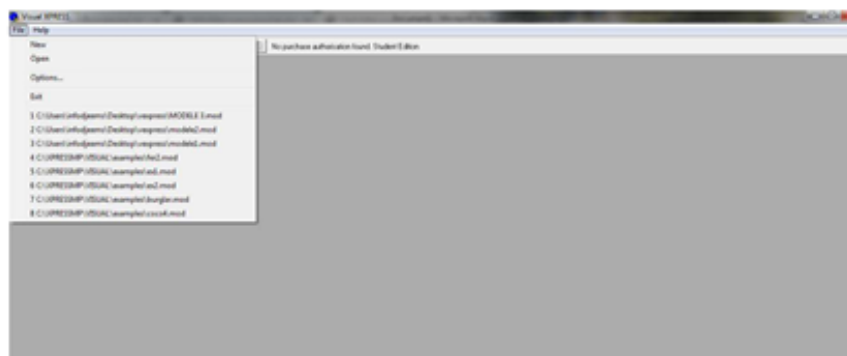


Visual Xpress.lnk

FIGURE 6.7 – L'icône du logiciel *Visual Xpress*

FIGURE 6.8 – Interface principale du logiciel *Visual Xpress*

Nous cliquons sur l'icône **File**, ce qui affiche l'interface suivante :

FIGURE 6.9 – Icône **File** dans l'interface de *Visual Xpress*

Puis, nous cliquons sur *New* et l'écran de l'espace de travail s'ouvre ; nous aurons alors l'interface suivante :

Nous allons résoudre le problème linéaire continu à l'aide de l'icône *Solve LP*, accessible depuis le menu *Run* → *Solve LP*.

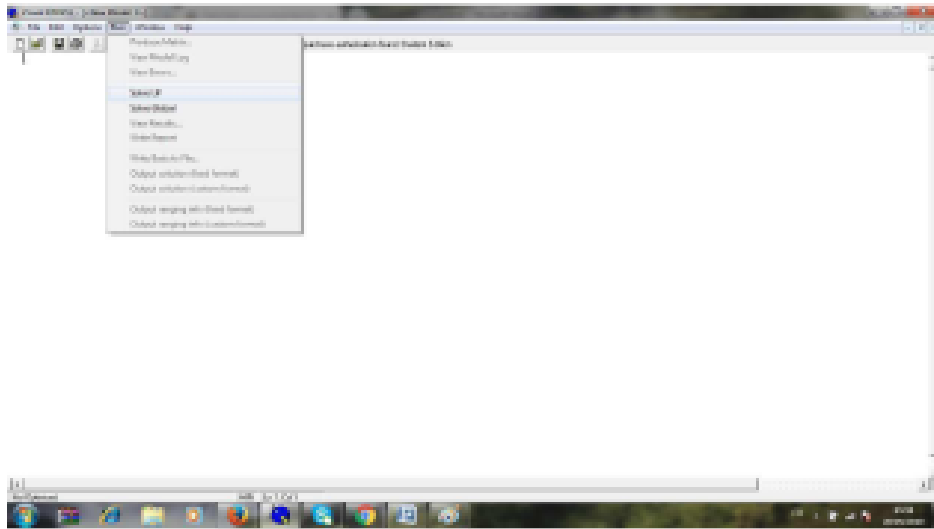


FIGURE 6.12 – Icône *Solve LP* permettant d'exécuter la résolution d'un problème linéaire continu dans le menu *Run* → *Solve LP*

Si le problème linéaire est en nombres entiers, il faut utiliser la commande *Run* → *Solve Global*.

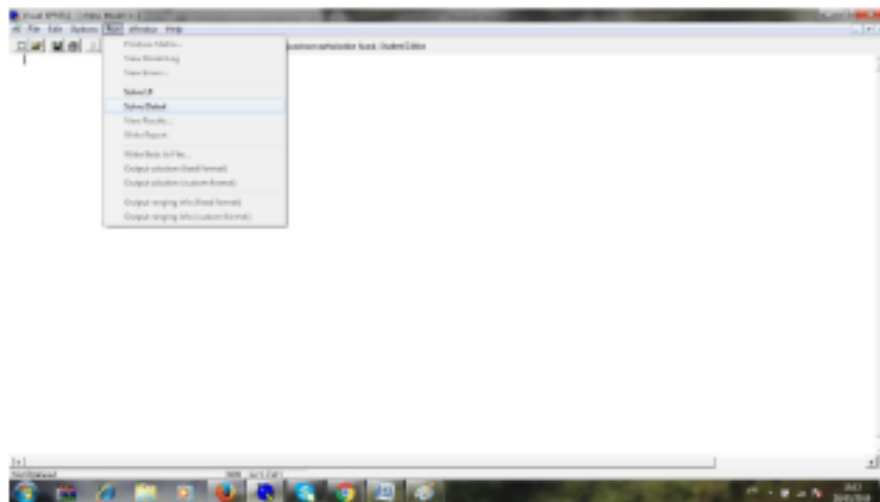
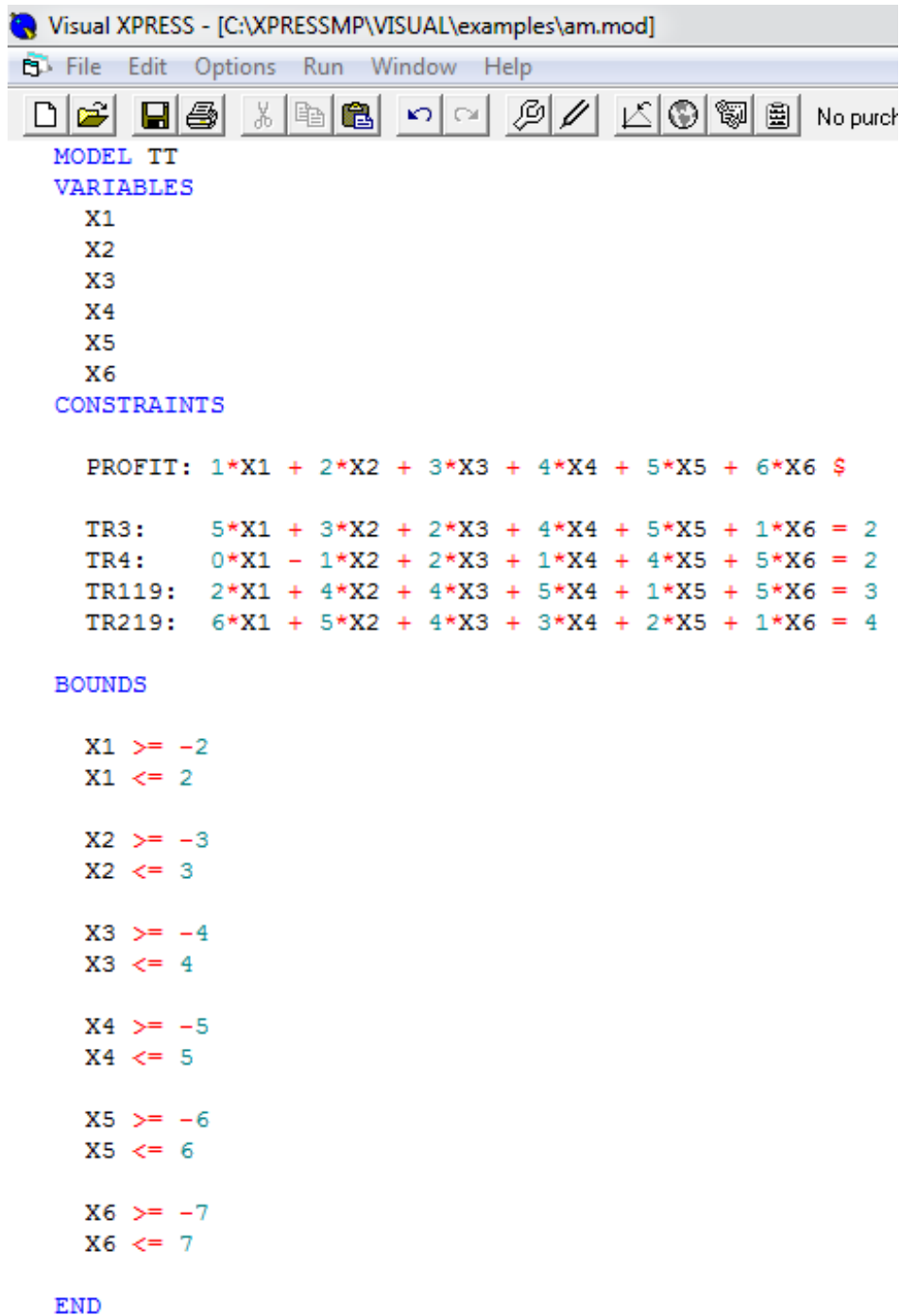


FIGURE 6.13 – Icône *Solve Global* utilisée pour résoudre un problème linéaire en nombres entiers via le menu *Run* → *Solve Global*

6.3.3 Application

soit le problème (1.5) précédent.

résolution :



```

Visual Xpress - [C:\XPRESSMP\VISUAL\examples\am.mod]
File Edit Options Run Window Help
[Icons] No purch

MODEL TT
VARIABLES
  X1
  X2
  X3
  X4
  X5
  X6
CONSTRAINTS

  PROFIT: 1*X1 + 2*X2 + 3*X3 + 4*X4 + 5*X5 + 6*X6 $

  TR3:    5*X1 + 3*X2 + 2*X3 + 4*X4 + 5*X5 + 1*X6 = 2
  TR4:    0*X1 - 1*X2 + 2*X3 + 1*X4 + 4*X5 + 5*X6 = 2
  TR119:  2*X1 + 4*X2 + 4*X3 + 5*X4 + 1*X5 + 5*X6 = 3
  TR219:  6*X1 + 5*X2 + 4*X3 + 3*X4 + 2*X5 + 1*X6 = 4

BOUNDS

  X1 >= -2
  X1 <= 2

  X2 >= -3
  X2 <= 3

  X3 >= -4
  X3 <= 4

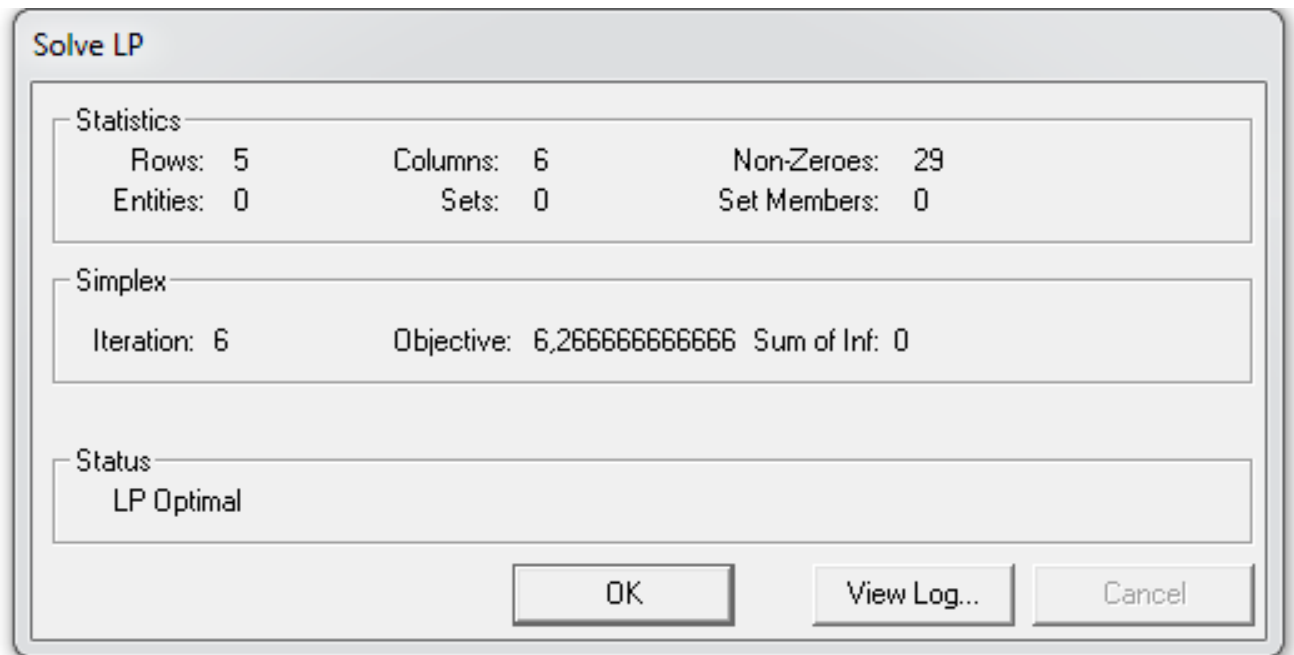
  X4 >= -5
  X4 <= 5

  X5 >= -6
  X5 <= 6

  X6 >= -7
  X6 <= 7

END

```



6.4 Conclusion général

Le présent mémoire a permis d'explorer en profondeur le domaine du **contrôle optimal**, à travers une étude progressive allant des concepts fondamentaux aux méthodes avancées de résolution, jusqu'à l'implémentation informatique.

Dans une première partie, nous avons posé les bases théoriques du contrôle optimal en abordant des notions clés telles que la contrôlabilité, l'observabilité, la stabilité, ainsi que le principe du maximum de Pontryagin. Cette exploration théorique s'est révélée essentielle pour bien comprendre les enjeux du pilotage optimal de systèmes dynamiques.

La seconde partie a été consacrée au développement et à l'application d'une **méthode adaptée**, qui s'est montrée efficace dans la résolution des problèmes de contrôle optimal discret. Nous avons analysé les différentes étapes de l'algorithme proposé, sa convergence, ainsi que son comportement sur des cas concrets.

La troisième partie s'est intéressée à l'aspect **multi-objectif** du contrôle optimal. Nous avons étudié les notions de dominance, d'efficacité au sens de Pareto, et les différentes fonctions scalarisantes. La complexité d'un tel cadre réside dans le compromis entre plusieurs objectifs conflictuels, ce qui reflète bien la réalité des systèmes décisionnels.

Enfin, une attention particulière a été portée sur l'**implémentation informatique** des méthodes développées, en utilisant des outils comme LINGO, MATLAB et Visual Xpress. Cette phase a confirmé la pertinence des approches choisies et a offert un pont concret entre la théorie mathématique et l'application algorithmique.

Ce travail ouvre naturellement plusieurs perspectives. D'une part, l'extension aux systèmes non linéaires ou stochastiques pourrait enrichir l'approche. D'autre part, l'intégration de techniques d'optimisation métaheuristiques ou de l'intelligence artificielle pourrait améliorer la recherche de solutions dans des contextes plus complexes.

En conclusion, ce mémoire illustre la richesse et la puissance de l'optimisation dans le pilotage de systèmes dynamiques, tout en soulignant l'importance d'une approche rigoureuse, progressive et ancrée dans des outils pratiques pour en tirer des bénéfices tangibles.

Bibliographie

- [1] K. Louadj. Résolution de problèmes paramétrés de contrôle optimal. Thèse de doctorat, UMMTO, pp. 25–50, (2012).
- [2] R.E. Kalman. Mathematical description of linear dynamical systems. J. SIAM. control. pp. 152-192, (1963).
- [3] E. Trélat. Contrôle optimal : théorie et applications. Université Pierre et Marie Curie (Paris 6) et Institut Universitaire de France, Laboratoire Jacques-Louis Lions, CNRS, UMR 7598, 4 place Jussieu, BC 187, 75252 Paris cedex 05, (2013)
- [4] L. Pontryagin and al. Mathematical theory of optimal processes. Eds Mir Moscou, pp. 504-1974, (1974).
- [5] F. Jean, Y. Chitour et E. Trélat. Propriétés génériques des trajectoires singulières. C. R. Math. Acad. Sci. Paris, 337, pp. 49–52, (2003).
- [6] F. Jean, Y. Chitour et E. Trélat. Genericity results for singular curves. Journal of Differential Geometry, 73, pp. 45–73, (2006).
- [7] F. Jean, Y. Chitour et E. Trélat. Singular trajectories of control-affine systems. SIAM Journal on Control and Optimization, vol. 2, pp. 1078–1095, (2008).
- [8] R.V. Gamkrelidze, L.S. Pontryagin, V.G. Boltyanski et E.F. Mishchenko. The mathematical theory of optimal processes. Interscience Publishers, New York, (1962).
- [9] I. L. Vorobev M. Aidene and B. Oukacha. Algorithm for solving a linear optimal control problem with minimax. Performance Index Computational Mathematics and Mathematical Physics 45. vol 10, pp. 1691-1700, (2005).
- [10] B. Oukacha. Résolution de problème de contrôle optimal. Thèse de doctorat, Université Mouloud Mammeri de Tizi-Ouzou, (2005).
- [11] M. Chebbah. Résolution et implémentation d'un problème min-max en contrôle optimal. Mémoire de magister en mathématiques appliquées, UMMTO, pp. 63–68, (2006).

-
- [12] R. Gabasov and F. M. Kirillova. Adaptive methods of solving linear programming problems. Preprint Series of University of Karlsruhe. Institute for Statistics and Mathematics, (1994).
- [13] R. Gabasov, F. M. Kirillova, Méthodes de calcul de systèmes de commande optimale, Éditions Mir, Moscou, 1978
- [14] A.M. Geoffrion. Proper efficiency and the theory of vector maximization. Journal of Mathematical Analysis and Applications, vol. 22 (1968), pp. 618–630.
- [15] Yann C., Siarry P., Optimisation multiobjectif, groupe Eyrolles, 2002.
- [16] Klein D. and Hannan E., An algorithm for multiple objective integer linear programming problem, European journal of operational research 9, pp. 152- 159, (1982).
- [17] A. Berro, Optimisation multiobjectif et stratégies d'évolution en environnement dynamique, thèse doctorat en informatique ,Toulouse I,(2001).