

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE  
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA  
RECHERCHE SCIENTIFIQUE  
UNIVERSITÉ MOULOUD MAMMERI, TIZI-OUZOU  
FACULTÉ DES SCIENCES  
DÉPARTEMENT DE MATHÉMATIQUES



**MÉMOIRE DE MASTER**

en

RECHERCHE OPÉRATIONNELLE

OPTION : MÉTHODES ET MODÈLES DE DÉCISION

Sujet :

**Les algorithmes de l'optimisation globale et leurs applications**

Présenté par :

***M<sup>r</sup>* AHMED ALI Abdelkader**

Encadré par :

***M<sup>me</sup>* F.LESLOUS M.A.A**

**Promotion : 2016/2017**

# Table des matières

<b>Introduction générale</b>	<b>6</b>
<b>1 Généralités</b>	<b>10</b>
1.1 Convexité . . . . .	10
1.1.1 Ensemble convexe : . . . . .	11
1.1.2 Fonction Convexe : . . . . .	11
1.2 Problème d'optimisation . . . . .	12
1.3 Signe d'une matrice : . . . . .	16
1.4 Différentiabilité . . . . .	17
1.4.1 Gradient . . . . .	18
1.4.2 Hessienne . . . . .	18
<b>2 Optimisation continue sans Contraintes</b>	<b>19</b>
2.1 Introduction : . . . . .	19
2.1.1 Les méthodes d'optimisation locale . . . . .	19
2.1.2 Optimisation déterministe . . . . .	20
2.1.3 Les méthodes d'optimisation globale . . . . .	24
<b>3 Optimisation continue avec Contraintes</b>	<b>31</b>
3.1 Introduction . . . . .	31
3.2 Méthodes d'optimisation par transformation . . . . .	31
3.3 Méthodes de pénalités intérieures et extérieures . . . . .	33
3.3.1 Les méthodes de pénalité intérieurs . . . . .	33
3.3.2 Les méthodes de pénalités extérieures . . . . .	33
3.4 La pénalisation radicale des solutions . . . . .	33
3.5 Choix d'une méthode d'optimisation . . . . .	34
3.6 Exploration et exploitation des algorithmes d'optimisation . . . . .	35
<b>4 Applications en Optimisation</b>	<b>37</b>
4.1 Introduction . . . . .	37
4.2 Équipements électronique . . . . .	37

4.2.1	Circuit imprimé . . . . .	38
4.2.2	Composants électroniques . . . . .	39
4.3	Optimisation des cartes électroniques . . . . .	40
4.3.1	introduction : . . . . .	40
4.4	Optimisation de la position de la vis d'une carte électronique .	40
	<b>Conclusion générale</b>	<b>46</b>
	<b>Bibliographie</b>	<b>48</b>

# Table des figures

1.1	Ensemble non Convexe . . . . .	10
1.2	Ensemble Convexe . . . . .	11
1.3	Fonction Convexe . . . . .	12
1.4	Fonction non Convexe . . . . .	12
1.5	Optimum global et Optimum local . . . . .	16
2.1	colonies de fourmis . . . . .	28
2.2	Algorithmes génétique . . . . .	30
4.1	Schéma d'une carte électronique avec un composant. . . . .	38
4.2	Schéma des couches du PCB. . . . .	38
4.3	FR4 . . . . .	39
4.4	Schéma d'une carte électronique avec un composant . . . . .	40
4.5	Modèle géométrique . . . . .	42
4.6	Maillage . . . . .	43
4.7	Déplacement maximal . . . . .	45

# Liste des tableaux

4.1	Propriétés des matériaux . . . . .	44
4.2	Résultats déterministes . . . . .	46

# Remerciements

«Celui qui ne remercie pas les gens n'a pas remercié Allah»

**Hadith charif**

En tout premier lieu, je remercie le bon Dieu, tout-puissant, de m'avoir donné la force pour survivre, ainsi que l'audace pour dépasser toutes les difficultés.

Je tiens d'abord à adresser mes plus vifs remerciements et ma profonde reconnaissance à ma promotrice *M<sup>me</sup>* **LESLOUS Fadila** pour son encadrement et sa patience afin que je puisse réaliser ce modeste travail.

Que Monsieur le président et Messieurs les membres de jury trouvent ici l'expression de ma gratitude et respect de m'avoir fait l'honneur d'examiner et juger mon travail.

Pour tous mes amis qui m'ont apporté leur soutien moral pendant cette année d'études, je les en remercie sincèrement.

Enfin, un très grand MERCI à toute ma famille qui m'a gratifié de son amour et fourni les motivations. Je leur adresse toute ma gratitude du fond du coeur.

# Dédicaces

Au nom du dieu le clément et le miséricordieux louange à **ALLAH** le tout puissant.

Je dédie ce modeste travail en signe de respect, reconnaissance et de remerciement :

A mes chers parents, qui m'ont aidé de près et de loin.

A mes chers Soeurs et Frères.

A tous mes chers amis.

A tout ceux qui ont participé à l'élaboration de ce modeste travail et tous ceux qui nous sont chers.

**AHMED ALI Abdelkader**

# Introduction générale

La recherche opérationnelle appelée aide à la décision peut être définie comme l'ensemble des méthodes et des techniques rationnelles orientées vers la recherche de la meilleure façon d'opérer des choix en vue d'aboutir au résultat visé au meilleur résultat possible, elle fait partie de l'aide à la décision elle propose des modèles conceptuels en vue d'analyser et de maîtriser des situations pour permettre aux décideurs de comprendre et dévaluer ou de faire les choix les plus efficaces.

Au fil de l'histoire la collaboration entre les scientifiques et les militaires est fréquente à fin de prendre les meilleures décisions dans les batailles et d'essayer acquérir la victoire. C'est pour cela que beaucoup d'experts dans le domaine considèrent le début de la recherche Opérationnelle dans le troisième siècle av. J. -C, pendant les guerres puniques, avec l'analyse et la solution qu'Archimède a proposée pour la défense de la ville de Syracuse, assiégée par les Romains. Entre ses inventions on trouve la catapulte et un système de miroirs qui enflammait les embarcations ennemies en faisant usage des rayons de soleil.

En 1503, Léonard de Vinci a participé comme ingénieur dans la guerre contre Pise puisqu'il connaissait les techniques pour bombarder, construire des bateaux, des véhicules cuirassés, des canons, des catapultes et d'autres machines de guerre.

Un autre précédent c'est l'usage de la recherche Opérationnelle pendant la Première Guerre mondiale en Angleterre, grâce à l'étude mathématique de Frederick William Lanchester sur la puissance balistique des forces opposantes.

Néanmoins, la recherche opérationnelle n'était pas considérée comme une science que jusqu'à la Seconde Guerre mondiale, pendant la bataille d'An-



gleterre.

Dans notre travail on s'est intéressé aux Algorithmes en optimisation globale et leurs Application.

L'optimisation est devenue une discipline incontournable du monde moderne, car celui-ci est sujet à une compétition internationale, excessive et croissante. Dès lors, il devient nécessaire, pour les entreprises comme pour les gouvernements, de maximiser ou de minimiser toutes sortes de choses ; par exemple, maximiser les profits tout en minimisant les pertes. On peut voir de façon intuitive, un problème d'optimisation comme un problème de recherche qui consiste à explorer un espace contenant l'ensemble de toutes les solutions potentielles réalisables, dans le but de trouver la solution optimale, sinon la plus proche possible de l'optimum, permettant de minimiser ou de maximiser une fonction dite fonction objectif.

On distingue classiquement deux types d'optimisation :

**L'optimisation locale** : recherche une solution qui est la meilleure localement, c'est-à-dire que dans son voisinage aucune solution n'est meilleure qu'elle. Cette solution est appelée un optimum local.

**L'optimisation globale** : recherche quant à elle la meilleure solution du domaine en entier, c'est-à-dire que dans tout le domaine il n'existe aucune solution qui lui soit meilleure tout en respectant les contraintes. Cette solution est appelée optimum global.

L'intérêt de l'optimisation globale par rapport à l'optimisation locale est patent. Elle garantit en effet que personne ne peut avoir une solution meilleure que celle trouvée. Or, pour une entreprise, cette information a son importance, car la différence entre la solution globale et une solution locale est bien souvent significative. Mais l'intérêt n'est pas que compétitif. Dans de nombreux problèmes, l'optimum global est la seule solution mathématique correspondant à une réalité physique. C'est ce qu'illustre par exemple la recherche de la quantité de chaque élément présent dans un mélange chimique à l'équilibre.

Le travail accompli est présenté à travers les différents chapitres. Dans le premier chapitre, nous allons donner des rappels sur quelques notions théo-

riques de base.

Dans le deuxième chapitre, nous présentons les différentes méthodes d'optimisation continue sans contraintes.

Dans le troisième chapitre, nous présentons l'une des méthodes d'optimisation continue avec contraintes.

le quatrième chapitre, nous l'avons consacré à l'application qui a pour but d'optimiser la position de la vis d'une carte électronique en utilisant l'une des méthodes de l'optimisation globale.

# Chapitre 1

## Généralités

### 1.1 Convexité

**Définition 1.1.1.** [5]

*La convexité est à la base une propriété géométrique, assez intuitive d'ailleurs, qui permet de caractériser certains objets. On voit assez bien ce qu'est un objet convexe dans un espace à deux ou trois dimensions. Nous allons maintenant montrer comment cette propriété peut aussi s'appliquer aux fonctions de  $\mathbb{R}^n$  dans  $\mathbb{R}$ .*

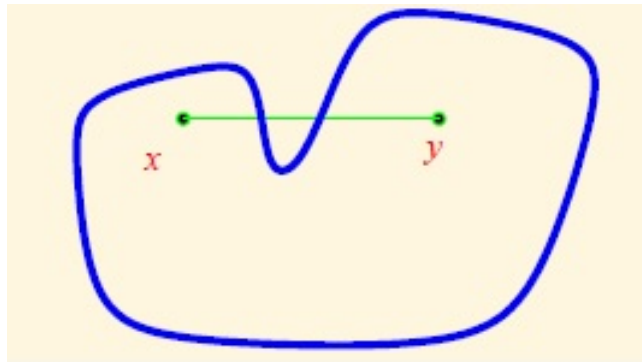


FIGURE 1.1 – Ensemble non Convexe

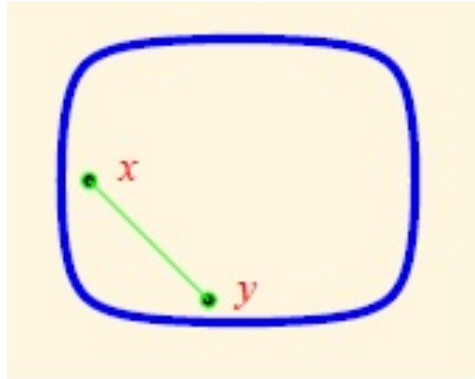


FIGURE 1.2 – Ensemble Convexe

### 1.1.1 Ensemble convexe :

**Théorème 1.1.1.** [5]

On dit qu'un ensemble  $K \subset \mathbb{R}^n$  est dit convexe si pour tout couple  $(x, y) \in K^2$  et  $\forall \lambda \in [0, 1]$  on a :

$$\lambda x + (1 - \lambda)y \in K.$$

Cette définition peut s'interpréter en disant que le segment reliant  $x$  et  $y$  doit être dans  $K$ . On dira qu'un vecteur  $y$  est une combinaison convexe des points  $(x^1, \dots, x^p)$  si on a :

$$y = \sum_{i=1}^p \lambda_i x^i$$

avec  $\lambda_i \geq 0, \sum_{i=1}^p \lambda_i = 1$  et  $y \in K$ .

### 1.1.2 Fonction Convexe :

**Définition 1.1.2.** [5]

On dit qu'une fonction  $f : K \rightarrow \mathbb{R}$ , définie sur un ensemble convexe  $K$ , est convexe si elle vérifie :

$$\forall (x, y) \in K^2, \forall \lambda \in [0, 1], f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y).$$

On dira que  $f$  est **strictement** convexe si

$$\forall (x, y) \in k^2, x \neq y, \forall \lambda \in ]0, 1[, f(\lambda x + (1 - \lambda)y) < \lambda f(x) + (1 - \lambda)f(y).$$

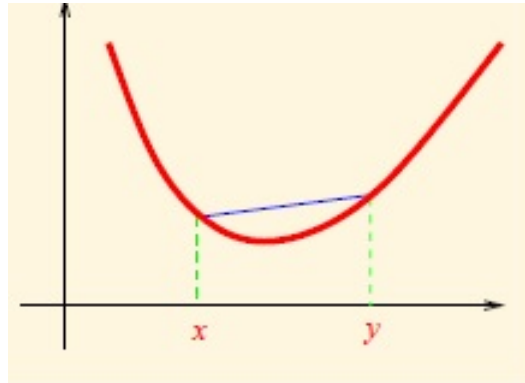


FIGURE 1.3 – Fonction Convexe

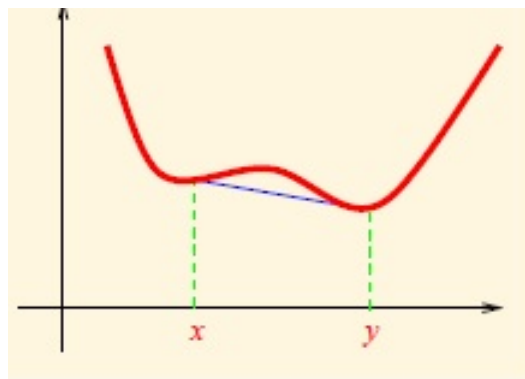


FIGURE 1.4 – Fonction non Convexe

## 1.2 Problème d'optimisation

La formulation des problèmes d'optimisation est comme suit :

**Un problème d'optimisation mono-objectif** est défini par un ensemble de variables, une fonction objectif à minimiser ou à maximiser et un ensemble de contraintes.

**Un problème d'optimisation multiobjectif** est défini par un ensemble de variables, un ensemble de fonctions objectif à minimiser ou à maxi-

minimiser et un ensemble de contraintes.

**L'espace d'état**, appelé aussi domaine de recherche, est l'ensemble des domaines de définition des différentes variables du problème.

**Les variables du problème** dite aussi variable de conception ou de décision peuvent être de nature diverse (réelle, entière, booléenne. etc.) et exprimer des données qualitatives ou quantitatives.

**La fonction objectif** ou encore (**fonction de coût**) définit le but à atteindre, on cherche à **minimiser** ou à **maximiser** celle-ci.

**Une fonction multimodale** présente plusieurs minima et maxima (locaux et globaux). Tandis qu'une fonction **unimodale** n'a qu'un minimum, le minimum global.

**L'ensemble des contraintes** est en général un ensemble entre autre d'égalités ou d'inégalités que les variables de l'espace d'état doivent satisfaire. Ces contraintes limitent l'espace de recherche.

Les méthodes d'optimisation recherchent un point ou un ensemble de points dans l'espace de recherche qui satisfont l'ensemble des contraintes, et qui maximisent ou minimisent la fonction objectif.

**Problème d'optimisation mono-objectif** Mathématiquement, dans le cas d'une minimisation, un problème d'optimisation(PO) mono-objectif se présente sous la forme suivante :

$$PO \left\{ \begin{array}{l} \text{Minimiser } f(x) \\ \text{Sous les contraintes} \\ h_i(x) = 0, \quad i = 1, \dots, n \\ g_j(x) \leq 0, \quad j = 1, \dots, n \\ x_L \leq x \leq x_U \end{array} \right.$$

avec  $x \in R^n$  est le vecteur des variables de décision,  $f$  la fonction objectif,  $h_i(x)$  et  $g_i(x)$  sont respectivement les contraintes d'égalités et d'inégalité et  $x_L, x_U$  sont respectivement les bornes inférieures et supérieures du domaine de recherche des variables. Cet ensemble définit l'espace d'état, tandis que l'ensemble de points de l'espace des états possibles qui satisfait au mieux les contraintes est donné par :

$$C = \{x \in R^n / h_i(x) = 0, g_j(x) \leq 0, x_L \leq x \leq x_U\}$$

Il est possible de passer d'un problème de maximisation à un problème de minimisation grâce à la propriété suivante :

$$\max_{x \in C} f(x) = - \min_{x \in C} (-f(x))$$

Suivant la nature de  $f(x)$ , des contraintes  $g_j(x)$  et des variables  $x$ , le problème d'optimisation correspondant porte des noms divers. On distingue plus particulièrement les cas suivants :

### La programmation Linéaire :

Comme le nom l'indique la fonction objectif et les contraintes sont toutes linéaires :

$$\begin{cases} \min & f(x) = \sum_{i=1}^n a_i x_i \\ S.C & g_j(x) = \sum_{i=1}^n c_{ji} x_i, \quad x_i \geq 0, \quad j = 1, \dots, n \end{cases}$$

### La Programmation Quadratique :

Dans ce cas, on a :

$$\begin{cases} \min & f(x) = \sum_{i=1}^n a_i x_i + \sum_{i,j=1}^n a_{ij} x_i x_j \\ S.C & g_j(x) = \sum_{i=1}^n c_{ji} x_i, \quad j = 1, \dots, n \quad x_i \geq 0, \quad i = 1, \dots, n \end{cases}$$

C'est-à-dire  $f(x)$  quadratique et  $g_j(x)$  linéaire.

### La Programmation Convexe :

La fonction objectif et les contraintes sont toutes convexes ( $f(x)$ ,  $g_j(x)$  et  $h_j(x)$  sont convexes).

### La programmation Discrète :

où les variables prennent des valeurs entières.

### Les Problèmes de la Programmation continue :

où les variables prennent des valeurs réelles, on a  $C \subset R^n$ , avec  $C$  l'ensemble des solutions admissible du problème.

## La Programmation Non Linéaire :

La fonction objectif et les limitations sont non linéaires. Pratiquement ils recouvrent toutes les catégories, ce qui fait apparaître les autres comme des cas particuliers.

Derrière toutes ces catégories, il y a bien entendu des méthodes mathématiques de résolution qui en général portent le même nom.

Soit un problème d'optimisation mono-objectif PO et C l'ensemble des solutions admissibles du problème.

- si l'on peut prouver que  $\forall x \in C, f(x_g) < f(x)$ , alors on dira que  $x_g$  est l'optimum (minimum) global du problème PO ;

-s'il existe un ensemble  $V \subset C$ , contenant  $x_{loc}$ (x local), tel que  $\forall x \in V, f(x_{loc}) < f(x)$ , avec  $x \neq x_{loc}$ , alors on dira que  $x_{loc}$ (x local) est un optimum (minimum) local du problème PO.



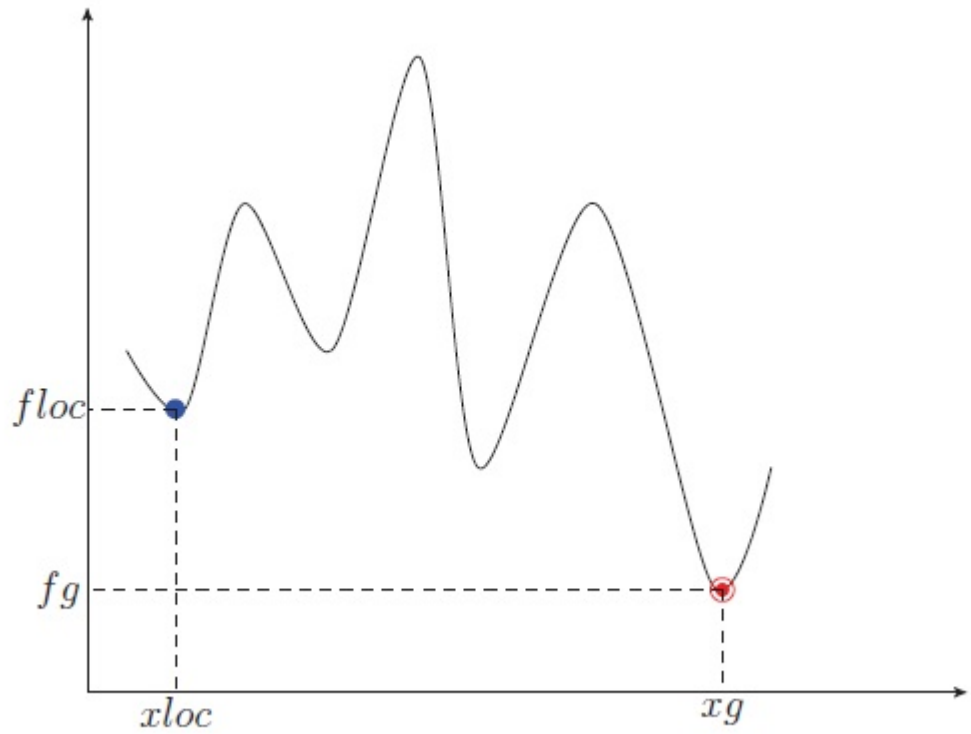


FIGURE 1.5 – Optimum global et Optimum local

### 1.3 Signe d'une matrice :

On définit une matrice carrée  $A(n \times n)$  d'éléments réels par un tableau :

$$A = \begin{pmatrix} a_{11} & \dots & a_{1n} \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ a_{n1} & \dots & a_{nn} \end{pmatrix}$$

**Théorème 1.3.1.** *Soit  $A$  une matrice carrée symétrique  $(n \times n)$ .*

1.  $A$  est définie positive  $\Leftrightarrow \det(A_k) > 0$ ,  $\forall k = 1, \dots, n$

2.  $A$  est définie négative  $\Leftrightarrow (-1)^k \cdot \det(A_k) > 0, \forall k = 1, \dots, n$

Où

$$A_k = \begin{pmatrix} a_{11} & \dots & a_{1k} \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ a_{k1} & \dots & a_{kk} \end{pmatrix}$$

**Théorème 1.3.2. (Critère qui utilise les valeurs propre)**

Soit  $A$  une matrice carrée symétrique ( $n \times n$ ),  $\beta_i$  les valeurs propres de  $A$ .

1. Si  $\beta_i > 0 \Leftrightarrow A$  est définie positive.
2. Si  $\beta_i \geq 0 \Leftrightarrow A$  est semi-définie positive.
3. Si  $\beta_i < 0 \Leftrightarrow A$  est définie négative.
4. Si  $\beta_i \leq 0 \Leftrightarrow A$  est semi-définie négative.

## 1.4 Différentiabilité

**Théorème 1.4.1. (Différentiabilité)**

Soit  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  une fonction définie par un voisinage de  $x \in A$ , on dit que  $f$  est différentiable en  $x^0$ , si :

$$\lim_{x \rightarrow x^0} \frac{f(x) - f(x^0) - \nabla f(x^0)(x - x^0)}{\|x - x^0\|} = 0$$

**Remarque 1 :**  $f$  est différentiable sur  $\mathbb{R}$  si elle est différentiable en chaque point de  $\mathbb{R}$ .

**Remarque 2 :**  $f$  est dite continument différentiable sur  $\mathbb{R}$  si les dérivées partielles de  $f$  sont continuent sur  $\mathbb{R}$ , et on dit que  $f$  est de classe  $C^1$ .

**Théorème 1.4.2. (Dérivées partielles)**

Soit  $f : A \rightarrow \mathbb{R}$  une fonction définie par un voisinage de  $x^0 \in A$ .

Si la

$$\lim_{h \rightarrow 0} \frac{f(x_1, x_2, \dots, x_i + h, x_{i+1}, \dots, x_n) - f(x^0)}{h}$$

existe et finie alors elle est appelée la  $i^{\text{me}}$  dérivée partielle, elle est notée  $\frac{\partial f(x^0)}{\partial x_i}$  ou  $\nabla_i f(x^0)$ .

### 1.4.1 Gradient

**Théorème 1.4.3.** Si  $f \in C^1(\mathbb{R}^n, \mathbb{R})$ , on appelle gradient de  $f$  en  $x$  le vecteur de  $\mathbb{R}^n$  :

$$\nabla f(x) = \left( \frac{\partial f(x)}{\partial x_1}, \frac{\partial f(x)}{\partial x_2}, \dots, \frac{\partial f(x)}{\partial x_n} \right)^T$$

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

### 1.4.2 Hessienne

**Théorème 1.4.4.** Soit  $f$  de classe  $C^2$ , on définit la hessienne de  $f$  par :

$$Hf(x) = J(\nabla f)^T = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_1 \partial x_2} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \cdots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{pmatrix}$$

**Remarque :**  $Hf(x)$  est une matrice carrée ( $n \times n$ ) symétrique.

# Chapitre 2

## Optimisation continue sans Contraintes

### 2.1 Introduction :

L'objet de ce chapitre est la présentation de techniques permettant de résoudre le problème d'optimisation. Pour cela, nous nous intéresserons à des sous-problèmes sans contraintes, où on effectue la minimisation de la fonction  $J : \mathbb{R}^n \rightarrow \mathbb{R}$  sur tout l'espace.

Nous considérons donc le problème de la manière suivante :

$$(P) \begin{cases} \min & J(x) \\ x \in & \mathbb{R}^n \end{cases}$$

où  $J$  une fonction de  $\mathbb{R}^n$  vers  $\mathbb{R} \cup \{+\infty\}$ .

#### 2.1.1 Les méthodes d'optimisation locale

la recherche locale est une méthode utilisée pour résoudre des problèmes d'optimisation. La recherche locale peut être utilisée sur des problèmes de recherche d'une solution minimisant un critère parmi un ensemble de solutions candidates. Les algorithmes de recherche locale passent d'une solution à une autre dans l'espace des solutions candidates (l'espace de recherche) jusqu'à ce qu'une solution considérée comme optimale soit trouvée ou que le temps imparti soit dépassé. En revanche optimiser localement, c'est chercher une solution à un problème qui soit proche d'une solution de départ (optimisation locale), mais qui soit meilleure en terme de coût (fonction objectif).

Pour cela, nous recherchons une meilleure solution par itérations successives, cette classe de méthodes peut être déterministe ou non-déterministe[6].

### 2.1.2 Optimisation déterministe

Lorsque l'évolution de la méthode de résolution est prévisible et ne laisse aucune place au hasard, celle-ci est qualifiée de déterministe. Bien que l'optimisation locale ne soit pas le thème essentiel de notre travail. Nous en présentons les grandes lignes. En effet, comme nous le verrons plus loin beaucoup d'algorithmes d'optimisation globale utilisent des codes d'optimisation locale comme sous-programmes. Donc l'efficacité de ces derniers affecte celle des premiers[6].

#### **l'optimisation locale déterministe :**

Soit  $f : IR^n, \rightarrow IR$ , une fonction continue, la formulation du problème d'optimisation locale (Ploc) s'écrit[6] :

$$(Ploc) \left\{ \begin{array}{l} \text{Trouver } (xloc, floc) \in C \times IR, \text{ où } floc = f(xloc), \\ \text{telque :} \\ \\ \forall x \in V, f(xloc) \leq f(x) \\ \text{où } V \text{ est un voisinage de } xloc. \end{array} \right.$$

Nous nous intéressons ici à l'aspect numérique de la résolution de Ploc. Pour ce faire, on va construire itérativement une suite  $\{ x^1, x^2, \dots, x^n \}$ . de façon à trouver un point xloc solution de (Ploc).

La valeur xloc qui minimise  $f(x)$  peut ainsi être caractérisée de 2 manières :

$$i) \forall x \neq xloc \quad f(xloc) \leq f(x)$$

$$ii) \nabla f(xloc) = 0 \text{ et } H(f(x)|_{x=xloc}) \text{ est une matrice semi définie positive.}$$

**La condition (i)** a donné naissance à la méthode de **recherche directe** :

### Les méthodes de recherche directe (direct search) :

Dans cette section sont illustrées les méthodes de recherche directe. Elles sont appelées ainsi car elles ne supposent aucune condition sur la fonction objectif et les contraintes comme la convexité, la continuité, ou la différentiabilité, par exemple. Ces méthodes considèrent que les dérivées sont inaccessibles ou inestimables et elle utilisent uniquement les valeurs de la fonction pour conduire l'optimisation. Elles reposent sur la génération de directions qui permettent d'explorer l'espace de recherche[6].

Cette famille de méthodes ne tente pas d'approcher le gradients ni de construire un modèle de  $f$ , des approximations de gradients sont utilisées pour ordonner les directions de recherche, elle se base principalement sur le calcul de la fonction objectif. Parmi ces méthodes se trouvent la méthode simplexe proposée par DANTZIG[3]. L'algorithme général des méthodes de recherche directe est le suivant :

---

**Algorithme** général des méthodes de recherche directe

---

**Étape 1 : Initialisation**

$x_0$

**Étape 2 : Affectation**

$x_{opt} \leftarrow x_0$

**Étape 3 : Itération**

**Tant que** test d'arrêt n'est pas vérifié **faire**

Mise à jour de  $x_k$

**Si**  $f(x_k) < f(x_{opt})$

$x_{opt} \leftarrow x_k$

**FinSi**

**Fin du Tant que**

---

En principe la mise à jour se fait selon une direction de recherche  $d_k$  par

le processus itératif suivant

$$x_{k+1} = x_k + \lambda_k d_k$$

avec  $\lambda_k$  est un pas de déplacement et  $x_{k+1}$  est l'approximation de  $x$  loc à l'itération  $k + 1$ .

Parmi les méthodes directes, ils existent des méthodes dont les algorithmes n'obéissent pas vraiment au schéma de l'algorithme général, c'est à- dire dont les directions de recherche ne sont pas choisies de la façon définie ci-dessus.

**La condition (ii)** donne naissance à une famille de méthodes dite de type gradients :

### Les méthodes de type gradients :

Elles utilisent les gradients et/ou les Hessiens de  $f$  (ou leurs approximations) pour choisir des  $d_k$  et  $\lambda_k$  appropriés, de façon que la direction choisie soit une direction de descente, elles sont appelé aussi **méthodes géométriques**[6].

### Direction de Descente

Soit  $f : \mathbb{R}^n \rightarrow \mathbb{R}$ , une fonction continûment différentiable, et  $x$  un vecteur de  $\mathbb{R}^n$ . Le vecteur  $d \in \mathbb{R}^n$  est appelé **direction de descente** de  $f$  en  $x$  ssi  $\nabla f(x)^T d < 0$ . Le principe général est le suivant :  $x_1$  donné, nous construisons un processus  $x_k$  satisfaisant

$$f(x_{k+1}) < f(x_k) \text{ avec } x_{k+1} = x_k + \lambda_k d_k$$

où  $d_k$  est une direction de descente ( $\lambda_k \in \mathbb{R}_+$ ) qui minimise  $f(x_{k+1})$  dans la direction  $d_k$  ( $d_k \in \mathbb{R}^n$ )[6].

---

### Algorithme du gradient

---

#### Initialisations

$$x_0, \lambda_0, d_0$$

## Itération

**Tant que** test d'arrêt n'est pas vérifié **faire**

$$x_{k+1} = x_k + \lambda_k d_k$$

**Fin du tant que**

---

Pour calculer  $d_k$  et  $\lambda_k$ , l'algorithme peut utiliser des informations sur les valeurs du gradient (voire du Hessien) de  $f$  au point  $x_k$  courant. Le choix de la direction de descente conduit à attacher à l'algorithme général des noms particuliers, tels qu'on le verra juste après.

Les méthodes de descente sont celles où les  $\lambda_k$  et  $d_k$  sont supposés être ceux qui donnent une direction de descente et un pas optimal. Comme il est signalé juste en haut le choix de la direction de descente a conduit au développement de plusieurs algorithmes, chacun avec ses propres caractéristiques, parmi ces algorithmes on trouve :[6]

## L'algorithme du gradient à pas fixe

Elle consiste à construire le processus itératif suivant :

$x_1$  donnée

$$x_{k+1} = x_k - \lambda \nabla f(x_k)$$

où  $\lambda$  est appelé pas de descente, est une constante positive donnée, ne dépendant pas de  $k$ .

La direction de descente est celle opposée à la direction du gradient.

La convergence de la méthode n'est assurée que pour de bonnes valeurs de  $\lambda$ , c'est-à-dire ni trop grandes ni trop petites. Quand cette convergence existe, elle est généralement lente, voire très lente.

Ces considérations de lenteur ne doivent cependant pas faire rejeter a



priori la méthode car dans certaines situations elle est la seule à pouvoir être facilement et rapidement mise en œuvre. En prenant de petites valeurs de  $\lambda$ , en effet la descente doit être vérifiée, c'est-à-dire que  $f(x_{k+1}) < f(x_k)$ .

### 2.1.3 Les méthodes d'optimisation globale

Durant ces dernières années, l'optimisation globale a fait l'objet de plusieurs études grâce à des résultats théoriques nouveaux, une forte demande dans plusieurs domaines incluant des applications industrielles, et au développement des moyens de calcul. Contrairement à ce que l'on pourrait être tenté de croire, l'optimisation globale numérique n'a pas hérité (du moins pas systématiquement) des facilités des techniques numériques d'optimisation locale. En effet, ces dernières utilisent pour la plupart des directions de descentes (conditionnées par des calculs de gradients ou de leurs approximations) comme nous avons vu, ce qui permet de converger naturellement vers un point de minimum local. Or, justement, l'optimisation globale évite de rester en de tels points. Il lui faudrait, au contraire, en échapper. C'est pourquoi beaucoup d'approches ont été utilisées dans la tentative de résoudre le problème. Il s'agit, de trouver un état de minimum et de ne s'arrêter que s'il est le meilleur (l'optimum global). Pour illustration de cette diversité, nous citons dans la suite les méthodes les plus utilisées d'entre elles ainsi que les différents principes sur lesquels elles se basent. En dépit de l'abondance des méthodes proposées, nous pouvons leur trouver des traits caractéristiques suivant leurs approches. Ainsi nous donnons la classification suivante : On distingue deux types d'approches selon qu'elles incluent ou non des processus probabilistes : Les méthodes déterministes et les méthodes non-déterministes[6].

#### 1. Les méthodes déterministes :

Ces méthodes ont tout d'abord été introduites pour résoudre de manière exacte des problèmes particuliers comme par exemple les problèmes continus et linéaires sous contraintes linéaires (algorithme du simplexe de Dantzig) ; ces méthodes ont aussi été élargies aux cas discrets et mixtes mais uniquement dans le cas linéaire.

Ces méthodes permettent de bien gérer les contraintes, Elles garantissent l'obtention de la solution globale du problème.

Ces méthodes permettent d'obtenir, à la convergence, la solution exacte du problème d'optimisation considéré avec une garantie absolue. Aucune erreur numérique insidieuse ne peut écarter de tels algorithmes de la solution optimale, il sera dans le pire des cas seulement ralenti. Ces algorithmes sont appelés : algorithmes de **Branch-and-Bound** par (Séparation et Évaluation) qui est une méthode de résolution d'une classe de problème d'optimisation globale.

c'est une méthode itérative qui divise un ensemble **H** donné en plusieurs sous ensembles de plus en plus petits.

A chaque sous ensemble de **H**, on construit une borne inférieure de la fonction objectif dans le but d'éliminer les parties qui ne contiennent pas l'optimum globale et de sélectionner le sous ensemble qu'on doit diviser.

Cette méthode est très utilisée pour résolution d'un grand nombre de problème mathématiques avec leurs différentes structures[6].

## **2. Les méthodes Evolutionnaires et les méthodes non déterministes :**

Ces méthodes non-déterministes font appel à des tirages de nombres aléatoires. Elles permettent d'explorer l'espace de recherche plus efficacement.

Dans la suite on s'intéressera plus particulièrement aux métaheuristiques.

Le mot **métaheuristique** est dérivé de la composition de deux mots grecs :

- **heuristique** qui vient du verbe heuriskein (euriskein) et qui signifie trouver ;
- **méta** qui est un suffixe signifiant au-delà, dans un niveau supérieur.

Les premières métaheuristiques datent des années 1980. Elles sont utilisées généralement quand les méthodes classiques échouent. Le terme métaheuristique est utilisé par opposition aux heuristiques. En effet, les métaheuristiques peuvent être utilisées pour plusieurs types de problèmes, tandis qu'une heuristique est adaptée à un problème donné. Les métaheuristiques ont comme caractéristiques communes de part leurs caractères stochastiques, c.à.d. qu'une partie de la recherche est conduite de façon aléatoire, elles sont inspirées d'analogies avec la réalité : physique (recuit simulé,...), biologie (algorithmes évolutionnaires,...) ou éthologie (colonies de fourmis,...). En plus de cette base stochastique, les métaheuristiques sont généralement itératives,

c-à-d qu'un même schéma de recherche est appliqué plusieurs fois au cours de l'optimisation, et directes, c-à-d qu'elles n'utilisent pas l'information du gradient de la fonction objectif. Elles tirent en particulier leur intérêt de leur capacité à éviter les optima locaux, soit en acceptant une dégradation de la fonction objectif au cours de leur progression, soit en utilisant une population de points comme méthode de recherche. Du fait du foisonnement de la recherche dans ce domaine, un grand nombre de méthodes de ce type existent. Dans la suite, nous verrons comment plusieurs concepts théoriques, notamment d'inspiration biologique, peuvent aider à la conception de nouvelles métaheuristiques[6].

#### **a) Les méthodes inspirées des principes physiques :**

##### **Le recuit simulé :**

L'origine de cette méthode vient de l'analogie avec la métallurgie, elle consiste à monter la température du solide à des valeurs élevées pour atteindre des états de basse énergie d'un solide. Lorsque le solide est à une forte température, chaque particule possède une très grande énergie et peut effectuer de grands déplacements aléatoires dans la matière. Au fur et à mesure que la température est abaissée, chaque particule perd de l'énergie et sa capacité de déplacement se réduit. Les différents états transitoires de refroidissement permettent d'obtenir des matériaux très homogènes et de bonne qualité. Ce processus est appelé **recuit**.

Le recuit simulé a été établi indépendamment par [11]. L'idée de base est la suivante : à des paliers de températures décroissantes, l'algorithme utilise la procédure itérative, pour atteindre un état de quasiéquilibre thermodynamique. Cette procédure permet de sortir des minima locaux avec une probabilité d'autant plus grande que la température est élevée. Quand l'algorithme atteint les très basses températures, les états les plus probables constituent en principe d'excellentes solutions au problème d'optimisation.

#### **b) Les méthodes inspirées par des comportements biologiques :**

Les méthodes évolutionnaires ont un point commun puisque leurs itérations tendent à l'amélioration d'une solution unique. En effet, à partir d'une première solution, les approches décrites tentent de l'améliorer en fonction des contraintes du problème. Les méthodes auxquelles nous nous intéressons désormais considèrent la solution comme étant une population formée de plusieurs individus capables de répondre au problème[6].

## L'algorithme de colonie de fourmis :

Le premier algorithme d'optimisation par colonies de fourmis (Ant colony optimization, ACO) a été proposé par Dorigo [4], [2]. vers le début des années quatre vingt dix ; . aussi appelé Ant System (AS), a été développé spécialement pour résoudre le problème du voyageur de commerce. Depuis, cette approche a connu des variantes importantes et le nombre de travaux publiés augmente d'une année à l'autre.

Le principe de la méthode provient analogiquement avec les comportements collectifs des insectes, les algorithmes de colonies de fourmis sont nés d'une constatation simple : les insectes sociaux, et en particulier les fourmis, résolvent naturellement des problèmes complexes. Un tel comportement est possible car les fourmis communiquent entre elles de manière indirecte par le dépôt de substances chimiques, appelées phéromones, sur le sol et le suivi de pistes observés dans les colonies de fourmis et construisent ainsi une solution à un problème en tenant compte de leur expérience collective. Au fait, elles adoptent pour la recherche de la solution la notion du **plus court chemin**. D'une manière simplifiée, les fourmis commencent par se déplacer au hasard. Puis, lorsqu'elles trouvent de la nourriture, elles retournent vers leur colonie, en marquant leur chemin à l'aide de phéromone. Si d'autres fourmis rencontrent ce chemin, il y a de fortes chances qu'elles arrêtent leurs déplacements aléatoires et qu'elles rejoignent le chemin marqué, en renforçant le marquage à leur retour, s'il mène bien vers la nourriture. Par conséquent, le chemin le plus court sera davantage parcouru, et donc plus renforcé et plus attractif. Par conséquent, le nombre de fourmis suivant cette trajectoire augmente. Au fil du temps, la quantité de phéromones déposée sur le plus long chemin diminue et finit par disparaître. Toutes les fourmis suivent alors le chemin le plus court.

L'algorithme de colonies de fourmis a été principalement utilisé pour produire des solutions quasi-optimales au problème du voyageur de commerce, puis, plus généralement, aux problèmes d'optimisation combinatoire. Récemment, son emploi se généralise à plusieurs domaines, depuis l'optimisation continue jusqu'à la classification, ou encore le traitement d'image.

---

### Algorithme de colonies de fourmis

---

#### Étape 1 : Initialisation

Initialiser les pistes de phéromone

### Étape 2 : Construction de la solution

Pour chaque fourmi répéter

Construction de la solution en utilisant les pistes de phéromone

### Étape 3 : Mise à jour des pistes de phéromone

Jusqu'à atteindre la condition d'arrêt

---

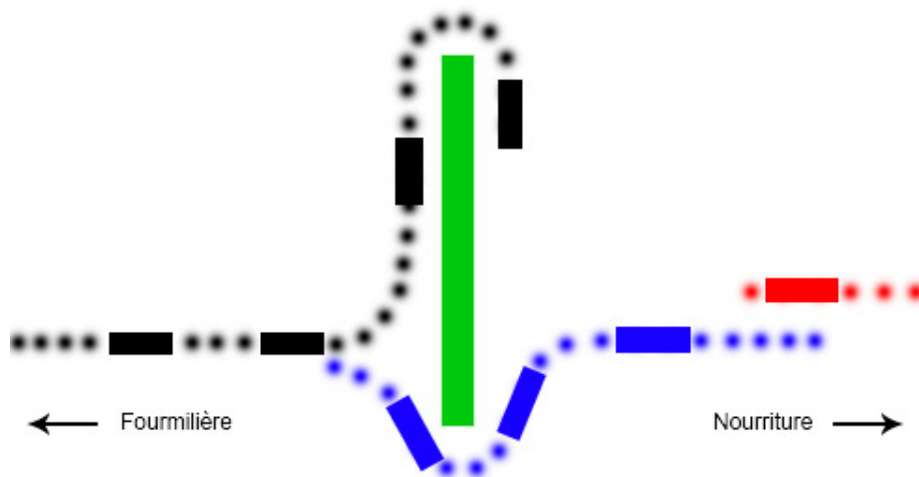


FIGURE 2.1 – colonies de fourmis

### Les algorithmes génétiques

Le concept d'**algorithme génétique** (de façon plus générale : algorithmes d'évolution) a été proposé par HOLLAND en 1975 pour décrire les systèmes adaptatifs.

Les algorithmes génétiques, également appelés **algorithmes évolutionnaires**, sont inspirés du concept de sélection naturelle proposée par CHARLES DARWIN. Le vocabulaire employé est directement calqué sur celui de la théorie de l'évolution et de la génétique. Nous parlerons donc ici d'individus (solutions potentielles), de populations, de gènes (variables), de chromosomes, de parents, de descendants, de reproductions, etc. Les points de l'espace de

recherche sont alors les individus d'une population et la fonction à optimiser correspond à leur adaptation. Ces algorithmes font évoluer une population de manière itérative. Certains individus se reproduisent, d'autres mutent ou encore disparaissent et seuls les individus les mieux adaptés sont supposés survivre. L'héritage génétique des générations doit permettre à la population d'être de mieux en mieux adapté et donc de mieux répondre au critère d'optimisation.

Les principales étapes d'un algorithme génétique sont :

---

**Algorithme : Algorithme génétique**

---

**Étape 1 : Génération de la population initiale**

**Étape 2 : Constitution d'une nouvelle population**

- Mesure de l'adaptation de chacun des individus
- Reproduction des individus en fonction de leur adaptation.
- Les plus performantes se reproduisent en priorité.
- Croisement de paires de séquences choisies aléatoirement
- Mutation de séquences tirées de manière aléatoire.

**Étape 3 : Critère d'arrêt**

---

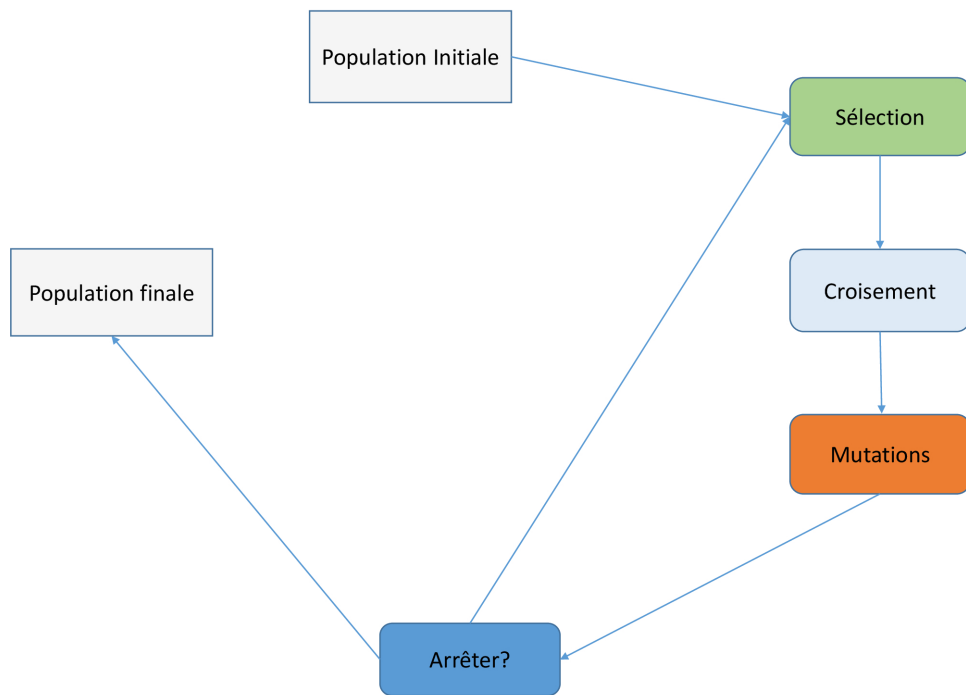


FIGURE 2.2 – Algorithmes génétique

# Chapitre 3

## Optimisation continue avec Contraintes

### 3.1 Introduction

Toutes les méthodes décrites dans la section précédente s'appliquent à des problèmes d'optimisation sans contraintes. Or, en réalité il est quasiment impossible de trouver ce genre de problème, il est habituel de poser des contraintes sur les variables de conception ou encore des contraintes imposées par le cahier des charges. Donc, la prise en considération de ces contraintes lors de la résolution d'un problème d'optimisation doit avoir lieu. Autrement dit, certaines méthodes d'optimisation avec contraintes doivent être appliquées. L'idée est de substituer à la fonction à minimiser une autre fonction incluant les contraintes. On obtient alors l'optimum en cherchant les minima d'une suite de fonctions sans contraintes. Dans la suite on va donner une brève description des méthodes d'optimisation avec contraintes, les plus répandues.

### 3.2 Méthodes d'optimisation par transformation

Les méthodes d'optimisation par transformation consistent à introduire les contraintes de conception dans la fonction que l'on cherche à optimiser.

Parmi ces méthodes on évoque la pénalisation qui est un concept simple qui permet de transformer un problème d'optimisation avec contraintes en un problème ou en une suite de problèmes d'optimisation sans contrainte.



C'est un concept qui a une utilité à la fois théorique et numérique. Ainsi le problème avec les contraintes de type inégalité :

$$\left\{ \begin{array}{l} \text{Minimiser } f(x) \\ \text{Sous les contraintes} \\ h_i(x) = 0, \quad i = 1, \dots, n \\ g_j(x) \leq 0, \quad j = 1, \dots, n \end{array} \right.$$

peut être transformé en un problème équivalent sans contraintes, formulé par :

$$\min \phi = f(x) + r \left( \sum_{j=1}^n P_j(g_j(x)) + \sum_{j=1}^n P_j(h_j(x)) \right)$$

où  $r$  est un paramètre (ou coefficient) de pénalité. La fonction de pénalité  $P_j$  est construite de manière à favoriser la sélection des points admissibles par rapport aux points non admissibles, par la méthode d'optimisation sans contrainte qui sera utilisée. Les méthodes de pénalité sont divisées en deux classes selon la nature de la fonction de pénalité.

En analyse, l'approche par pénalisation est parfois utilisée pour étudier un problème d'optimisation dont les contraintes sont difficiles à prendre en compte, alors que le problème pénalisé a des propriétés mieux comprises ou plus simples à mettre en évidence. Si on a de la chance ou si la pénalisation est bien choisie, des passages à la limite parfois délicats permettent d'obtenir des propriétés du problème original (l'existence de solution par exemple). D'autre part, comme nous allons le montrer ci-dessous, la pénalisation est un outil permettant d'étudier les problèmes d'optimisation avec et sans contrainte dans un même formalisme.

D'un point de vue numérique, cette transformation en problèmes sans contrainte permet d'utiliser des algorithmes d'optimisation sans contrainte pour obtenir la solution de problèmes dont l'ensemble admissible peut avoir une structure complexe. Cela semble merveilleux, inespéré, de voir que l'on puisse ainsi utiliser des algorithmes qui ne cherchent qu'à minimiser une fonction pour trouver des points qui, en plus d'être optimaux, sont admissibles. Cette approche est de ce fait très souvent utilisée. Elle permet d'obtenir une solution de qualité suffisante rapidement sans avoir à entrer dans l'algorithmique sophistiquée de l'optimisation avec contraintes. Ce n'est cependant pas une technique universelle, car elle a ses propres inconvénients : non-différentiabilité ou nécessité de minimiser une suite de fonctions de plus en plus mal conditionnées. On peut dire qu'il s'agit de la méthode du pauvre de l'optimisation avec contraintes.

### 3.3 Méthodes de pénalités intérieures et extérieures

#### 3.3.1 Les méthodes de pénalité intérieurs

Les méthodes de pénalité intérieures utilisent une fonction de pénalité construite de telle façon que la possibilité de réalisation soit garantie dans l'ensemble du processus d'optimisation. De telles méthodes sont également appelées méthodes à barrière car la fonction de pénalité forme une barrière infinie le long de la frontière du domaine réalisable.

$$\text{La fonction inverse : } P_j(g_j(x)) = \frac{-1}{g_j(x)}$$

$$\text{La fonction logarithmique : } P_j(g_j(x)) = -\log(-g_j(x))$$

L'utilisation des méthodes à barrière impose que les paramètres soient situés dans l'espace réalisable.

#### 3.3.2 Les méthodes de pénalités extérieures

ne présentent pas d'inconvénient car elles sont valables aussi bien à l'intérieur qu'à l'extérieur de l'espace réalisable. Elles augmentent la pénalisation des solutions à mesure que l'on s'éloigne de l'espace réalisable. Les fonctions de pénalité extérieures les plus populaires sont les suivantes :

$$P_j(g_j(x)) = \max[0, g_j(x)] \quad j = 1, \dots, n$$

$$P_j(g_j(x)) = \max[0, g_j(x)]^2 \quad j = 1, \dots, n$$

### 3.4 La pénalisation radicale des solutions

La méthode de pénalisation radicale est un processus de pénalisation très populaire dans le domaine de l'optimisation évolutionniste [12]. Il s'agit d'écartier les solutions non réalisables en attribuant à la fonction de transformation une valeur très élevée en cas de minimisation ( $\phi \rightarrow \text{inf}$ ) ou une

valeur nulle en cas de maximisation ( $\phi = 0$ ). La probabilité de survie de ces solutions, déterminée par les mécanismes de sélection, est alors quasi-nulle. Cette méthode est séduisante en raison de sa grande simplicité. Elle peut être appliquée avec succès lorsque l'espace de recherche est convexe et qu'il constitue une part importante de l'espace complet. Dans le cas contraire, cette approche a de sérieuses limitations, les solutions situées dans l'espace irréalisable ne pouvant être améliorées en raison de l'absence de directions données par la méthode de pénalisation.

### 3.5 Choix d'une méthode d'optimisation

On peut se demander à quoi sert une classification des types de problèmes et de méthodes. A notre avis quand un utilisateur est confronté à un problème d'optimisation globale, la première chose à faire est de bien cerner le problème, à savoir :

- Hypothèses sur  $f$  (différentiabilité, convexité...),
- hypothèses sur le domaine de recherche,
- existence ou non de contraintes, quels types de contraintes,
- coût d'évaluation de la fonction (temps, nombre de sous programmes),
- facilité de l'évaluation (accès, formule explicite de  $f$ ),
- précision dont on dispose sur les calculs,
- type de matériel utilisé,
- temps dont on dispose pour résoudre le problème.

Donc, on voit bien, que connaître une classification des problèmes et des méthodes, peut faciliter la tâche de l'utilisateur et le guider dans son choix ; ce qui lui permet de fixer ses objectifs en conséquence. Car il est préférable d'avoir une solution approchée (avec une précision insuffisante) dans un délai raisonnable qu'une solution exacte (avec la précision souhaitée).

Malheureusement, aucune méthode d'optimisation n'est capable de traiter efficacement tous les cas. En effet, [9] ont démontré que si l'on considère l'ensemble de tous les problèmes d'optimisation possibles, alors aucun algorithme n'est meilleur qu'un autre. Le choix d'un algorithme ne peut donc pas se faire par comparaison sur des cas tests généraux, mais doit aussi étudier des problèmes représentatifs des applications envisagées. La comparaison d'algorithmes d'optimisation locales ou globales ne peut avoir lieu qu'une fois on a précisé le problème traité, autrement dit la fonction objectif.

l'optimisation est non seulement une théorie mathématique mais aussi une sorte de cuisine algorithmique où c'est principalement l'expérience qui guide l'utilisateur dans le choix de l'algorithme à implanter. Pour choisir la méthode la plus adaptée à un problème bien précis, les caractéristiques principales prises en compte sont :

***1-La capacité à éviter les minima locaux :***

C'est selon la complexité du problème qu'on peut choisir une ou l'autre des méthodes présentées en haut. Par exemple si la fonction objectif est convexe il est sans doute recommander de travailler avec une méthode locale qui permet d'atteindre l'optimum rapidement par rapport aux méthodes d'optimisation globale. Ceci dit, si la fonction objectif est multimodale ce n'est sans doute pas la peine d'appliquer une méthode local et le recourt à des méthodes globales devient nécessaire. Quoique là encore plusieurs méthodes existent.

***2-La robustesse d'un optimum :***

est aussi une notion déterminante pour les concepteurs, et qui ne doit pas être négligée lors du choix de l'algorithme d'optimisation. La forme à optimiser est modélisée de manière déterministe ; cette forme est donc optimisée en négligeant les incertitudes liées par exemple aux procédés de fabrication ou aux simplifications du problème. Pour diminuer la probabilité que les performances de la forme optimisée ne soient pas vérifiées en réalité, on doit chercher un optimum robuste. Un optimum robuste est une solution peu sensible aux incertitudes.

## **3.6 Exploration et exploitation des algorithmes d'optimisation**

Pour un algorithme d'optimisation, l'exploration est sa capacité à explorer le domaine des variables pour rechercher la meilleure vallée, c'est à dire celle qui contient l'optimum global. A l'inverse, l'exploitation est sa capacité à converger rapidement vers le minimum d'une vallée donnée à partir d'un point de départ.

Le succès et l'efficacité d'une technique de résolution dépendent la plupart du temps d'un compromis entre l'exploration et l'exploitation.

Certaines méthodes toutefois n'utilisent qu'un seul de ces opérateurs pour

parvenir à l'optimum. Ainsi, les méthodes déterministes, exploitant les dérivées de la fonction objectif et des contraintes pour atteindre rapidement et précisément le minimum local le plus proche du point de départ, privilégient l'exploitation au détriment de l'exploration.

Tout algorithme d'optimisation doit utiliser ces deux stratégies pour trouver l'optimum global : l'exploration pour la recherche de régions inexplorées de l'espace de recherche et l'exploitation pour exploiter la connaissance acquise aux points déjà visités et ainsi trouver des points meilleurs.

# Chapitre 4

## Applications en Optimisation

### 4.1 Introduction

le champ d'application de l'optimisation structurale s'étend aujourd'hui à de nouveaux défis toujours plus intéressants[7]. La détermination de formes adéquates des composants structuraux est un problème de première importance pour l'ingénieur. Dans tous les domaines de la mécanique des structures, l'impact de la bonne conception d'une pièce est très important sur sa résistance, sa durée de vie et son utilisation en service. Ce défi est quotidien dans les secteurs de pointe tels que la recherche spatiale, électronique, l'aéronautique, l'automobile, la construction navale, la mécanique de précision et les ouvrages d'art en génie civil[8]. Le développement de l'art de l'ingénieur requiert des efforts considérables pour améliorer sans cesse les techniques de conception des structures. L'optimisation intervient de façon primordiale dans l'augmentation des performances soit de masse, de position ou de coût. Ce chapitre est consacré à un problème mono-objectif dans le domaine de la mécatronique. Ce travail aborde le problème des cartes électroniques.

### 4.2 Équipements électronique

Un équipement électronique est constitué d'un ensemble de plusieurs cartes électroniques assemblées, inter-connectées entre elles afin de réaliser les fonctions souhaitées. Chaque carte électronique est composée par trois sous-éléments ; la carte électronique nue, les composants électroniques et les joints de brasage qui permettent la liaison mécanique et électrique entre les composants et la carte[10].

### 4.2.1 Circuit imprimé

Le circuit imprimé ou PCB (Printed Circuit Board) (cf. figure 4. 1) est une carte électronique qui assure deux rôles principaux : le support mécanique des composants et leur interconnexion électrique. Le circuit imprimé est multicouche, il est constitué des couches conductrices en cuivre CU alternées avec des couches diélectriques de FR4(Frame Resistant 4), de fibre de verre ou d'autres matériaux isolants (cf. figure 4. 2)[10].

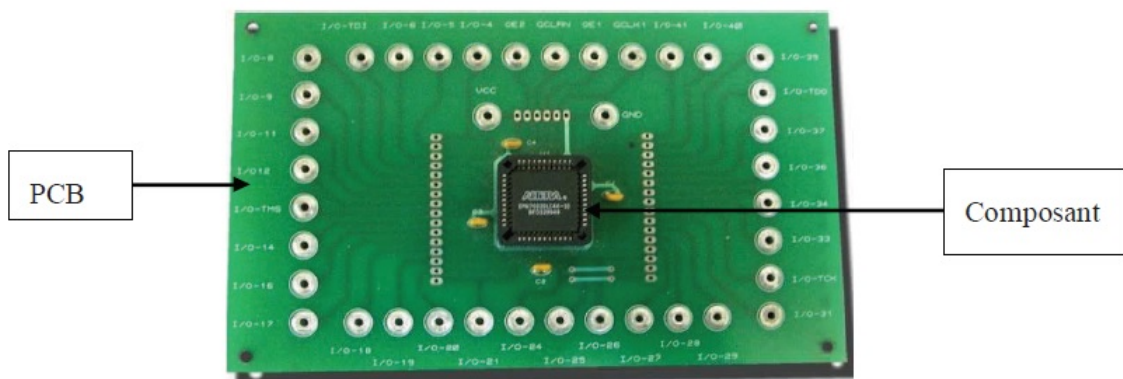


FIGURE 4.1 – Schéma d'une carte électronique avec un composant.

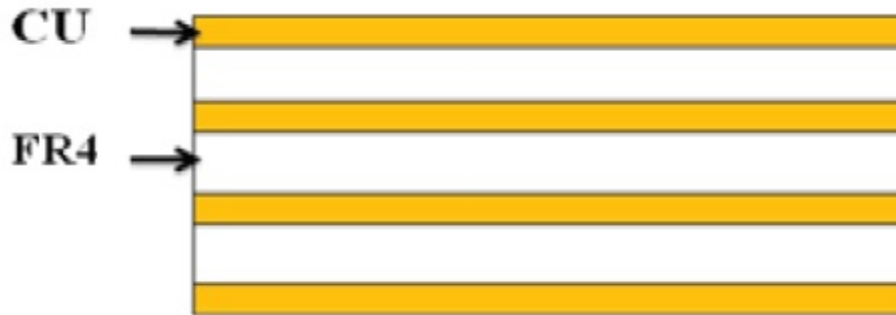


FIGURE 4.2 – Schéma des couches du PCB.

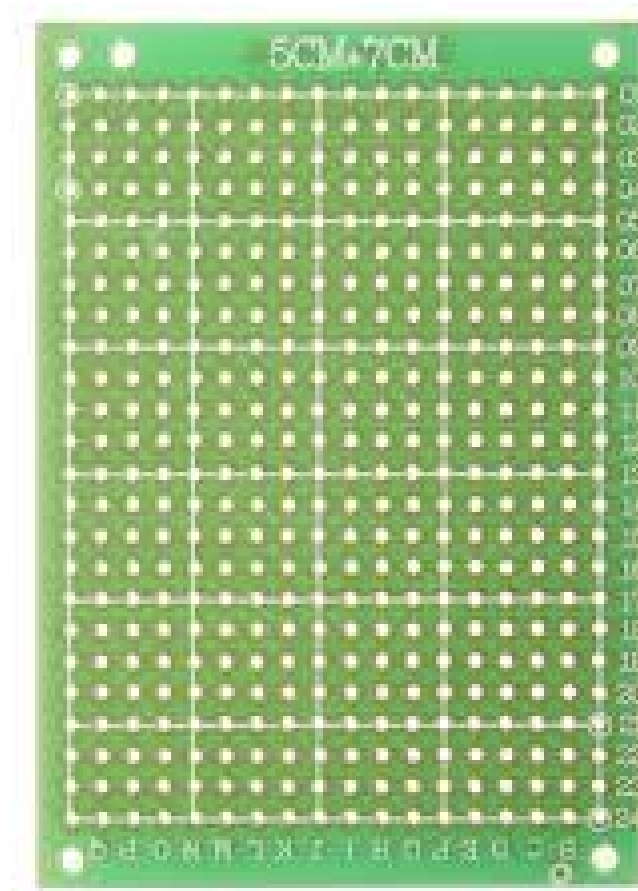

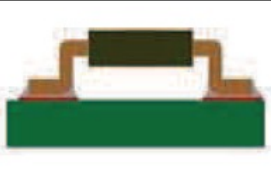
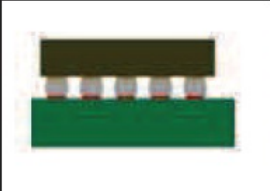



FIGURE 4.3 – FR4

## 4.2.2 Composants électroniques

Les composants électroniques sont nombreux et disponibles en plusieurs types. Ils peuvent être classés en fonction du mode d'interconnexion entre le composant et le circuit imprimé. Les modes d'interconnexion peuvent être à broches traversantes, à pattes, à billes ou à assemblage direct (cf. FIGURE 4.4)[10].



Modes d'interconnexion	composant à braches	composant à pattes	composant à billes	Assemblage direct
				






-  Composant
-  Circuit imprimé
-  Plage de report côté circuit imprimé
-  Alliage d'assemblage
-  Broche, patte, ou plage côté composant

FIGURE 4.4 – Schéma d'une carte électronique avec un composant

## 4.3 Optimisation des cartes électroniques

### 4.3.1 introduction :

Notre objectif est l'étude de la fiabilité des cartes électroniques. Ces cartes sont utilisées dans plusieurs domaines, tels que l'industrie automobile, l'aéronautique, les télécommunications, le secteur médical, ..., etc. Elles assurent toutes les fonctions nécessaires au bon fonctionnement d'un système électronique. Les cartes électroniques subissent diverses sollicitations (mécaniques, électriques et thermiques) durant la manipulation et la mise en service. Ces sollicitations sont dues aux chutes, aux vibrations et aux variations de température. Elles peuvent causer la rupture des joints de brasure des composants électroniques. Cette rupture entraîne la défaillance du système électronique complet.

## 4.4 Optimisation de la position de la vis d'une carte électronique

Les systèmes électroniques embarqués jouent un rôle très important dans plusieurs domaines, tels que l'industrie automobile, l'aéronautique, les télécommunications, le secteur médical... etc. Afin d'assurer leurs fonctionnements, les systèmes électroniques doivent être fiables. Toutefois, l'un des plus

grands problèmes des systèmes mécatroniques embarqués concerne l'évaluation de leur fiabilité. Les systèmes mécatroniques sont sujets à une dégradation liée à des modes de défaillance multiples selon les technologies utilisées et les conditions d'utilisation. L'utilisation simultanée de plusieurs technologies augmente les risques de dysfonctionnement des systèmes mécatroniques. Afin d'obtenir des systèmes dans lesquels les utilisateurs placent une grande confiance, des études de sûreté de fonctionnement, et en particulier de fiabilité, doivent être menées tout au long du cycle de développement ou de vie du système : de la spécification jusqu'à la validation et à la mise en exploitation.

## Modèle de simulation

La défaillance du circuit imprimé (PCB) est un point crucial pour assurer la fiabilité des cartes électroniques. Dans cette étude, la modélisation mécanique du PCB est réalisée sous Matlab. Ce travail vise d'une part l'analyse de la réponse du PCB pour une meilleure compréhension du comportement thermomécanique mis en jeu, et d'autre part la détermination de la position optimale de la vis qui minimise la déformation de la carte électronique[10].

## Modélisation géométrique

Le modèle géométrique, Figure 4.5, est une carte électronique embarquée sous forme d'une plaque rectangulaire avec quatre couches de cuivre (CU) et trois couches de composite FR4. Elle est encastrée par cinq vis comme le montre la figure 4.5. Elle est soumise à une température uniforme  $T$ . Un modèle en éléments finis 2D est développé.

Les propriétés de ces matériaux sont données au Tableau.

Les conditions aux limites sont introduites dans la modélisation[10].

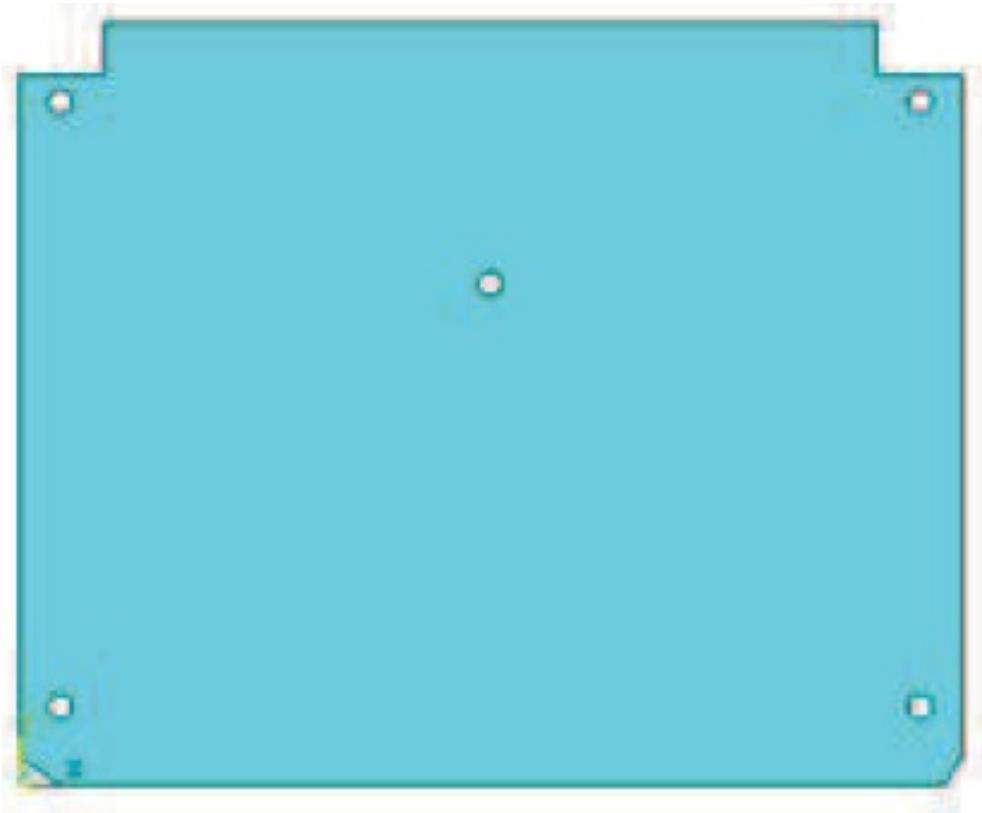


FIGURE 4.5 – Modèle géométrique

## Maillage

Le maillage est une action de former un réseau de disposer une grille qui recouvre, enserre,...etc l'interconnexion assurant une meilleure sécurité de l'alimentation électrique du secteur desservi, grâce à un réseau maillé. Dimension des mailles d'un filet de pêche est prévue par la réglementation.

La figure 4.6 présente le maillage utilisé pour la simulation thermomécanique de la carte électronique[10].

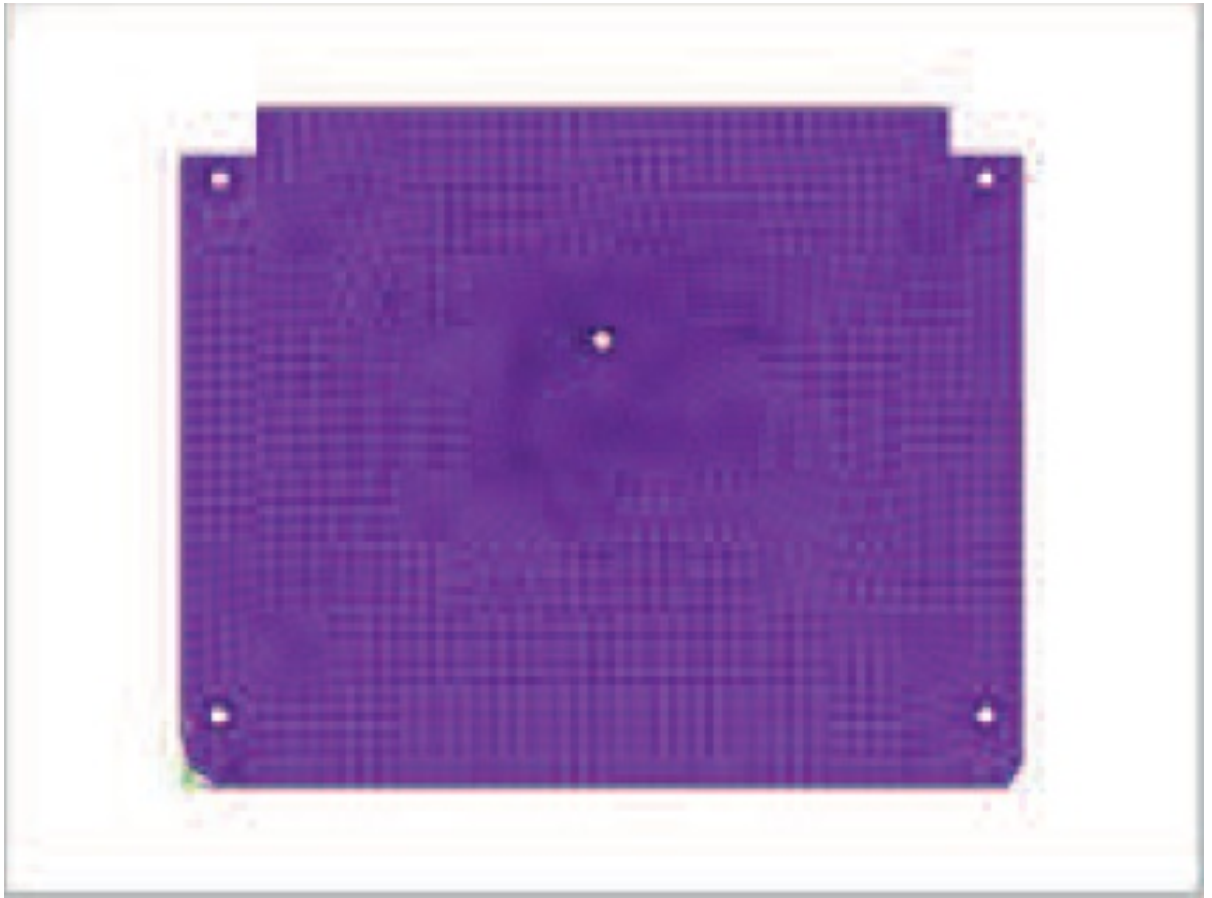


FIGURE 4.6 – Maillage

## Propriétés des matériaux

Les propriétés des matériaux constituant la structure étudiée sont présentées dans le tableau 4.1[10].

## Conditions aux limites et chargement

La structure est encastrée par 4 vis aux 4 coins et par une cinquième vis avec une position qui change aléatoirement pour trouver la position optimale qui minimise la déformation du PCB. La structure est soumise à une température constante[10].

Propriété des matériaux	FR4	CU
Module de Young (GPa)	17	115
Coefficient de Poisson	0.39	0.31
Densité (Kg/m <sup>3</sup> )	1800	8890
CTE (mm/K)	18	17
Module de cisaillement (GPa)	2.4	44

TABLE 4.1 – Propriétés des matériaux

## Calcul du déplacement maximal

Le calcul de la déformation du circuit imprimé en figure 4.7 indique la zone fortement déformée. La position de la cinquième vis du PCB est alors caractéristique puisque le déplacement maximal se produira autour cette zone. Une identification de cette position caractéristique a été réalisée afin d'alimenter le modèle d'optimisation[10].

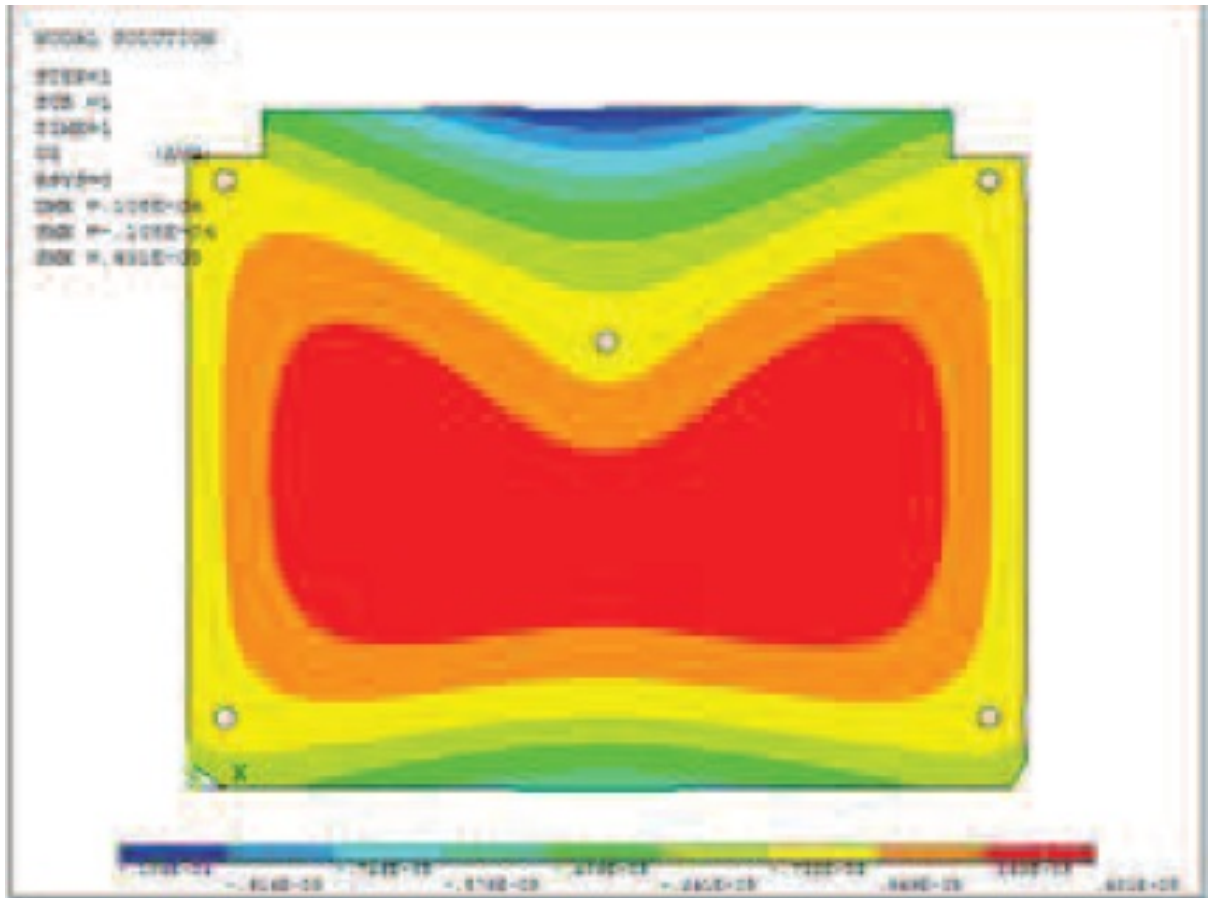


FIGURE 4.7 – Déplacement maximal

## Modèle mathématique

Pour cette application, nous cherchons la position optimale de la cinquième vis de fixation de la plaque embarquée. La plaque est soumise à une température uniforme. Le déplacement maximal  $D_{max}$  de la plaque est fixé[10].

$$\left\{ \begin{array}{l} \text{Minimisation} \quad f(x) = D_{max}(x, y) \\ \text{Sous les contraintes} \\ 10 \leq x \leq 160 \\ 15 \leq y \leq 121 \end{array} \right.$$

	Résultats déterministes
X (mm)	84.25
Y (mm)	89.75
Déplacement maximal (mm)	0.01

TABLE 4.2 – Résultats déterministes

## Résultats et discussion

la méthode d'optimisation déterministe donne de bons résultats et permet à notre carte électronique un bénéfice de déplacement de 0,01mm ; cet avantage permet de minimiser le déplacement maximal de la carte et d'offrir des zones adéquates pour placer des composants électroniques qui sont placés dans la carte électronique.

# Conclusion générale

Ce mémoire a présenté de nouvelles méthodes d'optimisation globale qui s'appuient sur les algorithmes métaheuristiques dont les algorithmes évolutionnaires destinés à résoudre des problèmes d'optimisation difficiles.

L'application traitée dans ce mémoire rentre dans le cadre de l'optimisation en mécanique de structures et l'optimisation des structures mécatronique.

cette application qui est l'optimisation de la position de la vis d'une carte électronique, ce problème est mono-objectif, son but est de trouver le bon positionnement de la vis de fixation de la carte électronique. Sachant que ce positionnement résulte de la minimisation du déplacement maximal qui déforme notre carte.

L'approche d'optimisation déterministe, a permis de minimiser le déplacement maximal de la carte et de localiser des zones adéquates pour assembler des composants électroniques à la carte étudiée.



# Bibliographie

- [1] A. CHIKHI ; N. BOUAFIA. Méthode hybride en optimisation globale. Master's thesis, UMMTO, 07 2013.
- [2] M. DORIGO ; V. MANIEZZO ; A. COLORNI. *Ant system : optimization by a colony of cooperating agents*. IEEE Trans. on Man, Cyber, Part B, 26 :29–41, 1996.
- [3] G. B. DANTZIG. *Programmation linéaire et extensions*. Presses de l'Université Princeton, Prince-tonne, New Jersey, 1963.
- [4] M. DORIGO. *Optimization, learning and natural algorithms*. PhD thesis, Politecnico di Milano, Italy, 1992.
- [5] R.LE RICHEL ; S.MOTTELET ; E.TOUBOUL. *Optimisation locale et globale*. 2010.
- [6] H.HACHIMI. *Hybridation d'algorithmes méthaheuristiques en optimisation globale et leurs applications*. PhD thesis, Ecole Mohammadia d'ingénieurs, Université Mohamed V Agdal, 2013.
- [7] DL. KUNZ ; AS. HOPKINS. *Structured data in structural analysis software*. Comput, Struct, 26 :965–978, 1987.
- [8] G. Kharmanda. *Optimisation et CAO des Structures Fiabiles*. PhD thesis, Université Blaise Pascal Clermont II, France, 2003.
- [9] D. FI. WOLPERT ; W. G. MACREADY. *No free lunch theoremns for optimnization*. IEEE Transactions on Evolutionary Computation, 1 :67–82, 1997.
- [10] S.ASSIF. *Fiabilité et optimisation des structures mécaniques à paramètres incertains : application aux cartes électroniques*. PhD thesis, Autre, 2013.
- [11] S.KIRKPATRICK ;C. GELATT ;M. P. VECCHI. *Optimization by simulated annealing*. Science, 220 :671–680,1983.
- [12] D. DASGUPTA ; R. LERICHE ; M.SCHOENAUER Z. MICHALEWICZ. *Evolutionary algorithms for constrained engineering problems*. Computers and Industrial Engineering Journal, 30, 1996.