

# TABLE DES MATIERES

<b>REMERCIEMENTS.....</b>	<b>I</b>
<b>TABLE DES MATIERES .....</b>	<b>II</b>
<b>LISTE DES FIGURES.....</b>	<b>VI</b>
<b>LISTE DES TABLEAUX .....</b>	<b>VII</b>
<b>LISTE DES ABREVIATION.....</b>	<b>VIII</b>
<b>INTRODUCTION GENERALE .....</b>	<b>1</b>
<b>Chapitre 1. ETAT DE L'ART SUR L'OPTIMISATION.....</b>	<b>3</b>
1.1. Introduction .....	3
1.2. Historique du problème d'optimisation.....	3
1.3. Définition du problème d'optimisation .....	5
1.3.1. Appellation .....	6
1.3.2. Exemple d'utilisation de l'optimisation .....	6
1.3.3. Notion de minimum .....	6
1.4. Classification des problèmes d'optimisation.....	8
1.4.1. Les variables.....	8
1.4.2. La fonction objective.....	11
1.4.3. Formulation du problème .....	15
1.5. Exemple de quelques problèmes d'optimisation rencontré fréquemment .....	17
1.5.1. Problème d'optimisation linéaire avec contrainte .....	17
1.5.2. Problème non-linéaire avec contrainte .....	17
1.5.3. Problèmes convexes .....	18
1.5.4. Problèmes non-convexes.....	18
1.6. Résolution du problème d'optimisation .....	18

1.7. Conclusion.....	19
<b>Chapitre 2. LES METHODES CLASSIQUES .....</b>	<b>20</b>
2.1. Introduction .....	20
2.2. Les problèmes d’optimisation linéaire .....	21
2.2.1. Existence de solution.....	23
2.2.2. Condition d’optimalité .....	23
2.2.3. Méthode du simplexe .....	24
2.2.4. Méthode du Tableau.....	26
2.3. Les problèmes d’optimisation non linéaire .....	28
2.3.1. Problèmes sans contraintes.....	29
2.3.2. Problèmes avec contraintes .....	39
2.4. Conclusion.....	49
<b>Chapitre 3. LES METHODES MODERNES .....</b>	<b>50</b>
3.1. Introduction .....	50
Les métaheuristiques .....	50
3.2. Recuit simulé.....	51
3.2.1. Le traitement thermique .....	52
3.2.2. Principe du recuit simulé.....	54
3.2.3. Algorithme du Recuit Simulé.....	56
3.3. Recherche tabou .....	57
3.3.1. Le randonneur malchanceux .....	57
3.3.2. Principe de la Recherche Tabou.....	57
3.3.3. Techniques d’améliorations de l’algorithme de base.....	59
3.3.4. Algorithme de la Recherche Tabou.....	60
3.4. Optimisation par essais particulières.....	61
3.4.1. Les vols d’oiseaux .....	62
3.4.2. Principe de l’algorithme d’optimisation par essais particulières .....	62
3.4.3. Algorithme de l’optimisation de l’Essaim Particulaire .....	66
3.5. Colonie des fourmis .....	67
3.5.1. Les fourmis réelles .....	67

3.5.2. La fourmi artificielle .....	69
3.5.3. Principe de l’algorithme de colonie des fourmis.....	70
3.5.4. Algorithme .....	73
3.6. Algorithme génétique.....	73
3.6.1. Notion de génétique des populations .....	74
3.6.2. Principe de l’algorithme génétique .....	75
3.6.3. Algorithme .....	82
3.7. Conclusion.....	82
<b>Chapitre 4. APPLICATION : OPTIMISATION DE LA PUISSANCE ACTIVE DANS LE RESEAU ELECTRIQUE.....</b>	<b>83</b>
4.1. Introduction .....	83
4.2. Généralités sur le réseau d’énergie électrique.....	83
4.2.1. Les éléments constitutifs d’un réseau.....	84
4.2.2. Les fonctions du réseau .....	85
4.2.3. Structure des réseaux électriques .....	85
4.3. L’écoulement de puissance .....	87
4.3.1. Classification des nœuds .....	87
4.3.2. Equations du réseau électrique en régime permanent .....	88
4.3.3. Résolution des équations.....	90
4.4. Optimisation de la puissance active d’un réseau.....	96
4.4.1. La fonction objective de l’optimisation de la puissance active.....	96
4.4.2. Les contraintes du réseau correspondantes à l’optimisation de la puissance active ..97	
4.5. Représentation du réseau.....	100
4.6. Résultat.....	101
4.6.1. Résultat du calcul de la répartition de puissance (load flow).....	101
4.6.2. Résultat de l’optimisation de la puissance active.....	104
4.6.3. Comparaison des résultats .....	106
4.7. Conclusion.....	107
<b>CONCLUSION GENERALE .....</b>	<b>108</b>
<b>REFERENCES .....</b>	<b>i</b>

**ANNEXES.....v**

ANNEXE A : CALCUL DE LA REPARTITION DE PUISSANCE PAR LA METHODE DE GAUSS SEIDEL.....v

ANNEXE B : CALCUL DES PERTES ACTIVES TOTALES .....viii

ANNEXE C : DETERMINATION DE LA MATRICE D'ADMITTANCE .....viii

ANNEXE D : OPTIMISATION PAR LA METHODE DE RECUIT SIMULE.....ix



# LISTE DES FIGURES

Figure 1-1 : Chemin le plus court trouvé par Héron d'Alexandrie

Figure 1-2 : Ensemble convexe

Figure 1-3 : Ensemble non convexe

Figure 1-4 : Fonction convexe

Figure 1-5 : Fonction non convexe

Figure 3-1 : Courbe de la température en fonction du temps d'un traitement thermique

Figure 3-2 : Déplacement d'une particule

Figure 3-3: Sélection du chemin plus court par une colonie de fourmis

Figure 3 -4: Les animaux de même espèces adaptés et non adaptés à l'environnement

Figure 3-5: Probabilité de chaque individu placé sur une roulette de casino.

Figure 3-6: croisement avec un point de crossover

Figure 3-7: croisement avec 2 points de crossover

Figure 4-1: réseau à structure maillée

Figure 4-2 : réseau à structure bouclée

Figure 4-3 : réseau à structure arborescente

Figure 4-4 : Organigramme de calcul de l'écoulement de puissance

Figure 4-5 : Organigramme de l'optimisation par la méthode de recuit simulé

Figure 4-6: Représentation du réseau à étudier

Figure 4-7 : Représentation des puissances sur le réseau

Figure 4-8 : Pertes actives totales en fonction de l'itération

Figure 4-9 : Représentation du réseau avec les nouvelles valeurs des puissances

Figure 4-10: Comparaison de l'amplitude des tensions avant et après optimisation

# LISTE DES TABLEAUX

Tableau 2-1 : Ecriture d'un problème d'optimisation linéaire sous forme d'un tableau

Tableau 2-2 : Mise en évidence de la colonne et ligne pivot

Tableau 2-3 : Tableau obtenu après division de la ligne pivot par le pivot

Tableau 4-1: Classification des réseaux en fonction du niveau de tension

Tableau 4-2 : les données et inconnues des nœuds

Tableau 4-3: Les fonctions objectives permettant de minimiser pertes totales en puissances actives

Tableau 4-4: Tableau d'impédance de ligne et d'admittance de charge par rapport à la terre.

Tableau 4-5 : Tableau de planification.

Tableau 4-6 :  $Y_{bus}$  obtenu après calcul

Tableau 4-7 : Les tensions des nœuds

Tableau 4-8 : Les Puissances dans les nœuds

Tableau 4-9 : Les pertes de puissances actives et réactives

Tableau 4-10 : Les tensions des nœuds après optimisation

Tableau 4-11 : Les pertes de puissances actives et réactives après l'optimisation

Tableau 4-12 : Comparaison des résultats obtenus par le load flow et l'optimisation

# LISTE DES ABREVIATIONS

- B Susceptance
- G Conductance
- H Matrice Hessienne
- Max Maximum
- Min Minimum
- P Puissance active
- PV Nœud générateur
- PQ Nœud charge
- pu Per unit
- Q Puissance réactive
- R Résistance
- $\mathbb{R}$  Ensemble des entiers relatifs
- S Puissance apparente
- V Tension
- VA Volt Ampère
- VAR Volt Ampère Réactif
- X Réactance
- Y Admittance
- Z Impédance

# INTRODUCTION GENERALE

Avec le contexte mondial actuel où la concurrence ne cesse de s'accroître d'une part et la majoration sans cesse du prix des ressources et de la main d'œuvre d'autre part, l'extension de la production ne s'avère plus la seule solution pour avoir le maximum de profit. En effet, pour avoir de l'avantage sur le marché, l'unique solution est d'essayer de réviser le rendement dans un but de répondre à un double objectif visant l'amélioration de la qualité des produits proposés tout en préservant le bon fonctionnement et la réduction des coûts de production pour en profiter l'augmentation du bénéfice. Mais cette tâche est difficile, en effet son but est de trouver un compromis entre plusieurs facteurs : besoins à satisfaire, l'efficacité, la rentabilité...

Les entreprises opérant dans le domaine de la production et de la répartition de l'énergie électrique sont vraiment touchées par ce problème. En effet, le rendement faible du réseau de transport et de distribution en est souvent la cause, ce fait est constaté par les manques à gagner dans le bilan de l'entreprise. Ce rendement faible est parfois causé par l'augmentation des pertes de puissance dans les lignes de transports, ce qui conduit à l'insuffisance de l'énergie arrivée aux abonnés. La solution pour ce problème est alors d'optimiser le rendement en minimisant les pertes dans les lignes de transport.

Les problèmes d'optimisation sont alors une discipline incontournable dans presque tous les domaines comme dans l'électricité, la mécanique, l'économie, ....

Pour ces raisons, au cours des dernières décennies, une part importante des activités de recherche a été consacrée dans le domaine de l'optimisation, cette dernière est alors un sujet central de nombreuses études pour pouvoir résoudre un grand nombre de problèmes d'aide à la décision pouvant en effet être décrits sous la forme de problèmes d'optimisation.

Ce mémoire part du constat simple que les problèmes d'optimisation occupent une place de plus en plus importante dans les processus décisionnels. C'est pourquoi, on a choisi d'étudier le problème d'optimisation dans ce mémoire intitulé : « Synthèse des méthodes d'optimisations – Application à l'optimisation de la puissance active dans le réseau électrique »

Ce travail aura donc pour objectif d'étudier les différents types de méthode d'optimisation et appliquer une des méthodes à optimiser la puissance active dans le réseau électrique.

Pour mener à bien cette étude, on a divisé cet ouvrage en quatre grands chapitres :

Le premier chapitre concerne l'état de l'art à travers lequel on parle les différentes évolutions qui ont marqué l'histoire de l'optimisation depuis la première formulation du problème d'optimisation.

Le deuxième chapitre illustre les méthodes anciennes pour la résolution des problèmes d'optimisation.

Le troisième chapitre sera consacré au développement des méthodes modernes.

Et dans le dernier chapitre, on appliquera la méthode de recuit simulé à l'optimisation de la puissance active dans un réseau.

# Chapitre 1. ETAT DE L'ART SUR L'OPTIMISATION

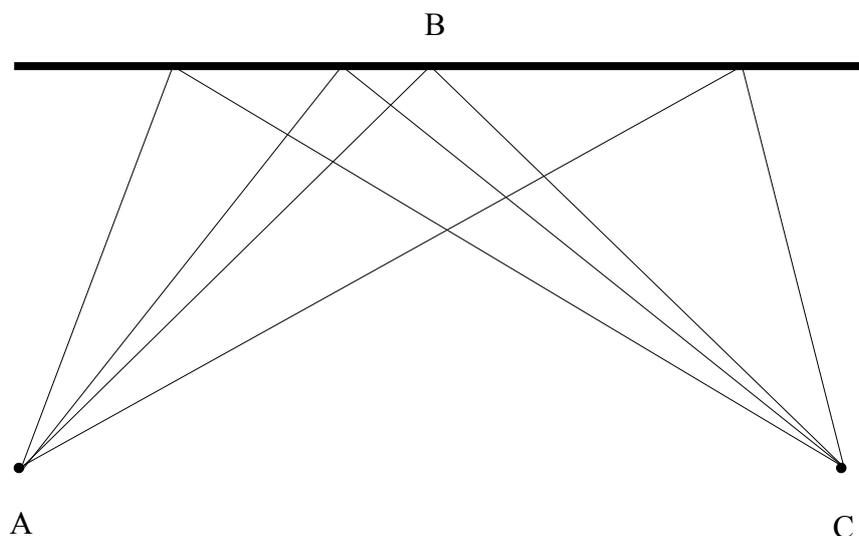
## 1.1. Introduction

L'optimisation est un sujet qui préoccupe les chercheurs, c'est pourquoi, nous allons donner quelques généralités pour avoir une idée sur le concept de l'optimisation, Le but de ce chapitre est donc de donner un survol rapide de quelques questions liées aux problèmes d'optimisation. On va répondre dans ce chapitre ce qu'on appelle optimisation, son histoire et par la suite, on va donner les critères qui permettent de classifier chaque problème d'optimisation et on terminera ce chapitre par donner quelques problèmes d'optimisation et d'énumérer les méthodes permettant de résoudre ces problèmes.

## 1.2. Historique du problème d'optimisation [46]

IIIe siècle : Les premiers problèmes d'optimisation auraient été formulés par Euclide dans son ouvrage historique *Éléments*.

VI siècle : Dans son recherche sur le domaine de l'optique, Héron d'Alexandrie a énoncé dans *Catoptrica* le principe du plus court chemin.



*Figure 1-1 : Chemin le plus court trouvé par Héron d'Alexandrie*

D'après la figure ci-dessus, le chemin le plus court pour aller de A à C en passant par le miroir est celui passant par le point B où l'angle d'incidence et l'angle réfléchi sont égaux.

XIIe siècle : L'apparition du calcul différentiel entraîne l'invention de techniques d'optimisation, ou du moins en fait ressentir la nécessité.

XVIe siècle : Issue de ses travaux avec Leibniz, Newton met au point une méthode itérative permettant de trouver les extrémums locaux d'une fonction en faisant intervenir la notion de dérivée. Cette nouvelle notion permet de grandes avancées dans l'optimisation de fonction car le problème est ramené à la recherche des racines de la dérivée.

Durant le XVIIIe siècle, les travaux des mathématiciens Euler et Lagrange mènent au calcul des variations, une branche de l'analyse fonctionnelle regroupant plusieurs méthodes d'optimisation. Ce dernier a inventé une technique d'optimisation sous contraintes: Les multiplicateurs de Lagrange.

XIXe siècle est marqué par l'intérêt croissant des économistes pour les mathématiques. Ceux-ci mettent en place des modèles économiques qu'il convient d'optimiser, ce qui accélère le développement des mathématiques. Depuis cette période, l'optimisation est devenue un pilier des mathématiques appliquées et le foisonnement des techniques est tel qu'il ne saurait être résumé en quelques lignes.

En 1947 : L'algorithme du simplexe fut proposé par G. B. Dantzig comme méthode de résolution générale des programmes linéaires. La solution optimale est approchée par étapes ou itérations successives.

1950 a été marqué par l'invention de la méthode du gradient conjugué par Cornelius Lanczos et Magnus Hestenes.

1960 : Rosenbrock a inventé les méthodes de recherche dans une famille de directions consistant à minimiser successivement la fonction coût dans une famille de  $n$  directions, par une suite d'évaluations. Cette approche a été modifiée par Powell en 1964 et par Swann la même année.

1965 : La méthode de Nelder-Mead qui est un algorithme d'optimisation non-linéaire a été publiée par Nelder et Mead. C'est une méthode numérique qui minimise une fonction dans un espace à plusieurs dimensions. Et cette méthode a été revisitée récemment par Dennis et Torczon en 1991

1975 : Holland a posé les bases de la technique d'optimisation appelée "algorithmes génétiques". Mais c'est Goldberg (1989) qui, par la suite s'est investi dans l'étude des AG et a développé la forme actuelle que nous connaissons.

1983 : inspirée d'un processus utilisé en métallurgie, les 3 chercheurs de la société IBM, S. Kirkpatrick, C.D. Gelatt et M.P. Vecchi ont mis au point la méthode du recuit simulé et indépendamment par V. Cerny en 1985.

1990 : remarquant le comportement des fourmis, Marco Dorigo et al proposé le premier algorithme de colonies de fourmis pour résoudre le problème de recherche de chemins optimaux dans un graphe

1995 : Russel Eberhart, ingénieur en électricité et James Kennedy, socio-psychologue, s'inspirent du monde du vivant pour mettre en place une méta-heuristique : l'optimisation par essaim particulaire.

1986 : Glover a développée dans un cadre particulier la méthode de recherche tabou. (et indépendamment par Hansen en 1986), c'est une méthode heuristique de recherche locale utilisée pour résoudre des problèmes complexes et/ou de très grande taille.

### 1.3. Définition du problème d'optimisation [43]

Le problème d'optimisation est un problème consistant à rechercher l'ensemble des paramètres permettant d'obtenir le meilleur résultat.

D'un point de vue mathématique, on peut poser le problème d'optimisation de la façon suivante :

Soit une fonction  $f$  définie par :

$$f : S \rightarrow \mathbb{R} \quad (1-1)$$

Qui associe à chacun des éléments  $x$  de l'ensemble  $S$  à un nombre réel  $y$  dans l'ensemble des entiers relatifs  $\mathbb{R}$ .

D'une manière générale, un problème d'optimisation consiste à la recherche de la meilleure solution qui satisfait au mieux le problème ci-dessous par une action adaptée

$$\begin{aligned} & \text{Minimiser } f(x) \\ & \text{Sous contraintes } x \in S \end{aligned} \quad (1-2)$$

Le problème d'optimisation se résume alors à trouver l'élément de  $S$  dont le coût  $y$  est minimal (ou maximal, mais cela ne change rien à la difficulté ou aux types de méthodes employées)

Dans le cadre de la recherche de l'optimum, on a deux grandes variantes de problème qui sont la maximisation et la minimisation d'une fonction objective, cependant, on peut remarquer que maximiser une fonction  $f$  est équivalent à minimiser  $-f$

$$\max f = \min(-f) \quad (1-3)$$

Il découle de cette observation que tout problème de maximisation peut être réduit immédiatement à un problème de minimisation (et inversement) en multipliant la fonction objective par -1.

Dans la suite de cette étude, on peut alors réduire notre étude par le problème de minimisation.

### 1.3.1. Appellation

- $f(x)$  est appelé : fonction objective, fonction de coût ou critère d'optimisation
- $x$  est appelé variables de décision
- $y = f(x)$  qui est l'image de la fonction objective dans  $\mathbb{R}$  est appelé cout.
- Nous désignerons souvent par l'appellation : ensemble admissible ou encore de domaine admissible l'ensemble S

### 1.3.2. Exemple d'utilisation de l'optimisation

- En physique : les problèmes d'optimisations physiques qui se posent souvent sont la minimisation d'une énergie;
- En finance : dans ce domaine, le problème majeur est qu'on veut optimiser les gains (ou minimiser les risques) sous certaines contraintes (puisque la caisse n'a pas une taille infinie) ;
- Dans l'industrie :
  - \* Optimiser la taille et l'emplacement de composants électroniques sur un circuit occupe actuellement l'attention des chercheurs;
  - \* Minimiser les chutes lors du découpage d'éléments dans un morceau de tissu ;
  - \* Gérer les stocks de façon optimale ;
- Analyse de données : faire passer une courbe ayant certaines propriétés (par exemple ayant une pente positive) au plus proche de certains points occupe la principale utilisation de l'optimisation.

Ce sont des exemples d'utilisation de l'optimisation dans la vie courante

### 1.3.3. Notion de minimum

Du point de vue général, minimum veut dire la valeur la plus petite. Pour une fonction, la valeur minimum est la valeur de l'image où cette fonction descend par une partie inférieure avant de remonter. Cependant parmi les minimums qui existent, on peut distinguer :

- Un minimum local
- Un minimum global

Soient l'ensemble  $S \subseteq \mathbb{R}^n$  et une fonction  $f : S \rightarrow \mathbb{R}$ . Les minimums locaux et minimums globaux de la fonction  $f$  sur l'ensemble  $S$  sont définis de la manière suivante :

### 1.3.3.1. Minimum local

Par définition, un vecteur  $x^* \in S$  est un minimum local de  $f$  sur l'ensemble  $S$  s'il a un coût plus faible que celui de ses voisins. Formellement,  $x^*$  est un minimum local de  $f$  sur  $S$  si  $\exists \epsilon > 0$  tel que :

$$f(x^*) \leq f(x) \quad \forall x \in S \text{ avec } |x - x^*| < \epsilon \quad (1-4)$$

Où  $|v|$  désigne la norme du vecteur  $v$ .

Le minimum local est strict si

$$f(x^*) < f(x) \quad \forall x \in S \text{ avec } |x - x^*| < \epsilon \quad (1-5)$$

### 1.3.3.2. Minimum global

Un vecteur  $x^* \in S$  est un minimum global de  $f$  sur l'ensemble  $S$  s'il a un coût plus faible que celui de tous les autres vecteurs dans  $S$ . Formellement,  $x^*$  est un minimum global de  $f$  sur  $S$  si

$$f(x^*) \leq f(x) \quad \forall x \in S \quad (1-6)$$

Le minimum global est strict si

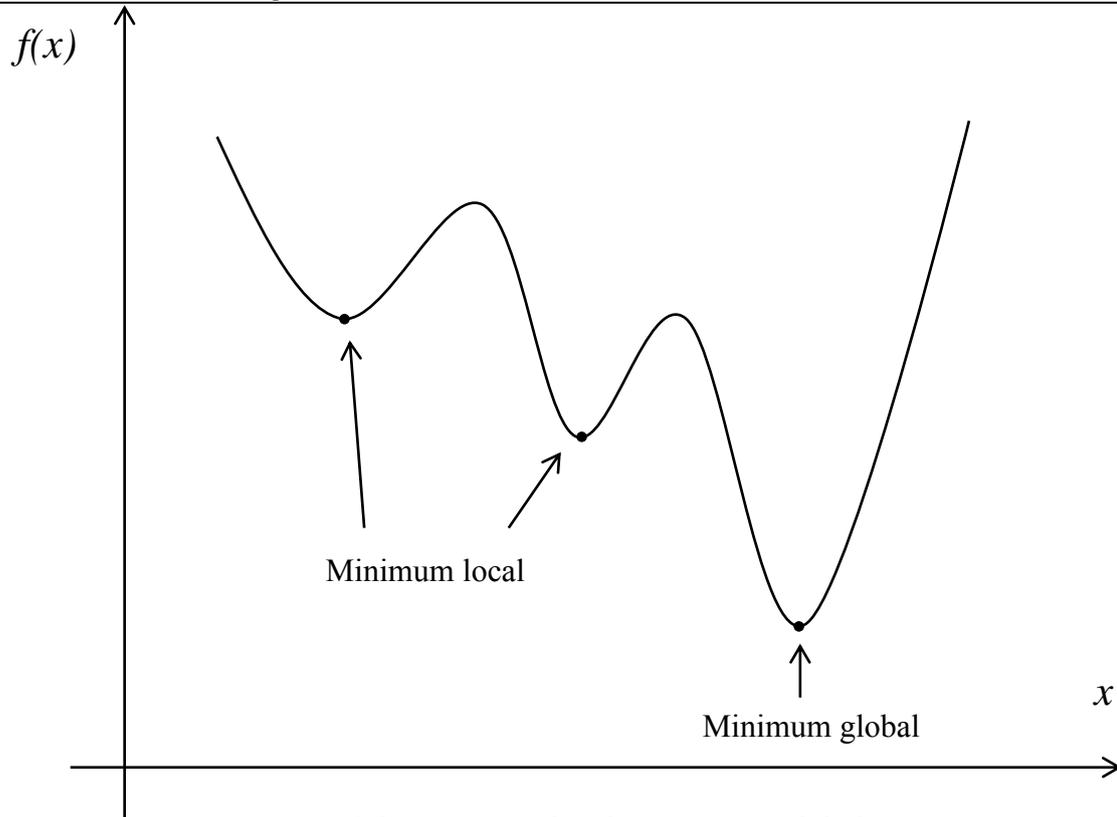
$$f(x^*) < f(x) \quad \forall x \in S \quad (1-7)$$

Dans le cas d'une fonction objective convexe, il n'y a pas de distinction entre minimum local et global : tout minimum local est également global, comme l'établit le théorème suivant.

#### ***Théorème :***

*Soit  $f : S \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$  une fonction convexe définie sur l'ensemble convexe  $S$ .*

*Alors, tout minimum local de  $f$  sur  $S$  est également un minimum global. Si  $f$  est strictement convexe, alors il existe au plus un minimum global de  $f$ .*



*Figure 1-2 : Minimum local et minimum global*

#### 1.4. Classification des problèmes d'optimisation [4]

On dispose jusqu'à présent plusieurs méthodes pour résoudre un problème d'optimisation. Avant de choisir la méthode d'optimisation à appliquer pour un problème donné, il faudra d'abord définir quel genre de problème on a avant d'appliquer la méthode adéquate pour résoudre ce dernier puisque chaque méthode est spécifique à un problème. Il existe de nombreuses sortes de problèmes d'optimisation et ceux-ci peuvent être classifiés par :

- Les variables en jeu
- Le type de la fonction objective  $f(x)$
- La formulation du problème

On va donner quelques aperçus sur ces 3 critères de classification

##### 1.4.1. Les variables

##### 1.4.1.1. Nombre de variable

Rappelons d'abord la définition de l'optimisation

Soit :  $f: S \rightarrow \mathbb{R}$

Minimiser  $f(x)$

$x \in S$

Ici, la variable qui permet d'influencer le but à atteindre est  $x$ . Cette variable à minimiser peut représenter un ou des paramètres physiques, électriques, économiques,....

Un problème peut être mono-variable c'est-à-dire ne manipule qu'une seule variable ou multi-variable dont plusieurs variables sont à tenir compte en même temps, on désigne par  $x_1, \dots, x_n$  les  $n$  variables du problème et c'est aux concepteurs de définir les variables du problème.

#### 1.4.1.2. Type de variables

Dans les problèmes d'optimisation rencontrés habituellement, on distingue deux types de variables différentes :

- Les variables discrètes
- Les variables continues

##### 1.4.1.2.1. Les variables discrètes

Les variables discrètes sont les variables contenues dans le domaine des mathématiques discrètes.

**Exemple :**

- variables entières :  $X = \mathbb{N}^n$ , Variables binaires :  $X = \{0,1\}^n$
- Les variables booléennes

##### 1.4.1.2.2. Les variables continues

Contrairement à la variable discrète, la variable est classée variable continue si l'ensemble  $S$  forme un ensemble continu.

**Exemple :**

- $X = \mathbb{R}_+^n, X = \{x \in \mathbb{R}^n \mid \|x\| \leq 1\}$
- Variable réelle

Puisque dans notre domaine, c'est-à-dire le domaine de l'électricité, plus précisément le réseau électrique, la variable fréquemment utilisée est la variable contenue dans l'ensemble des entiers relatif  $\mathbb{R}$ , alors par la suite, on va se focaliser dans ce type de variable.

---

Pour l'ensemble  $S \subseteq \mathbb{R}^n$  qui est le plus utilisé dans cette étude, on a la caractéristique:

### Ensembles convexes

Soient  $x$  et  $y$  deux points tel que  $x, y \in \mathbb{R}^n$ , l'ensemble

$$[x, y] = \{z = \lambda x + (1 - \lambda)y \mid 0 \leq \lambda \leq 1\}$$

Est appelé segment avec les extrémités  $x$  et  $y$ .

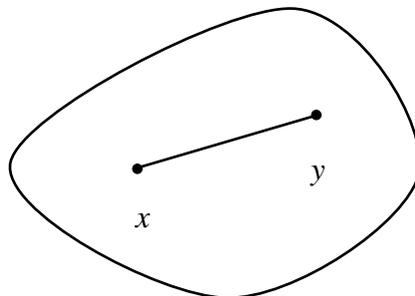
Un ensemble  $S \subseteq \mathbb{R}^n$  est appelé ensemble convexe si le segment joignant toute paire de points appartenant à  $S$  appartient également et entièrement à  $S$ .

$$x, y \in S, 0 \leq \lambda \leq 1 \Rightarrow \lambda x + (1 - \lambda)y \in S$$

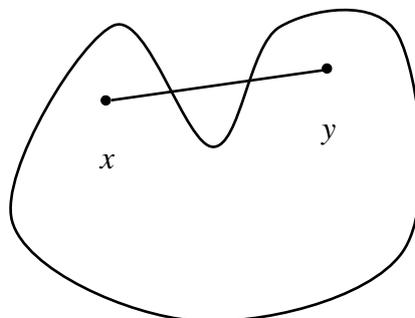
Quelques exemples d'ensembles convexes dans  $\mathbb{R}^n$ :

- $\emptyset$  (l'ensemble vide),
- $\mathbb{R}^n$  (La totalité de l'espace  $\mathbb{R}^n$ ),
- $\mathbb{R}_+^n$  (L'ensemble des vecteurs dont les composantes positives ou nulles),
- $\mathbb{R}_{++}^n$  (L'ensemble des vecteurs dont les composantes sont strictement positives),
- $B_r(a) = \{x \mid \|x - a\| < r \text{ et } \widehat{B}_r\}$  et  $\widehat{B}_r(a) = \{x \mid \|x - a\| \leq r \text{ et } \widehat{B}_r\}$  (les boules ouvertes et fermées de rayon  $r$  centrées en  $a$ )

Dans  $\mathbb{R}$ , il est facile de constater qu'un ensemble convexe est forcément un intervalle, ouvert ou fermé à gauche ou à droite, éventuellement infini (soit  $[a, b]$ ,  $]a, b]$ ,  $[a, b[$ ,  $]a, b[$  avec  $a < b$  et éventuellement  $a = -\infty$  et/ou  $b = +\infty$ ).



*Figure 1-2 : Ensemble convexe*



*Figure 1-3 : Ensemble non convexe*

**Remarque :**

Il est à noter qu'il n'existe pas de notion d'ensemble concave.

### 1.4.2. La fonction objective

#### 1.4.2.1. Définition

La fonction  $f(x)$  nommée fonction objective est la fonction à minimiser, c'est donc la fonction principale à manipuler dans le problème qui est le but à atteindre pour le décideur, en d'autre terme, c'est le cœur du problème. Elle peut représenter toutes les fonctions ou critères qui peuvent donner avantage en déterminant son optimum.

Elles peuvent être par exemple un comportement électrique, un coût de fabrication, une durée de développement, un rendement ou bien une performance,....

Il existe des optimisations monocritères pour les optimisations à une seule fonction objective et des optimisations multi-objectifs appelées aussi optimisations multicritères où le nombre des fonctions objectives est supérieur ou égal à 2 mais durant ce travail, on va se limiter dans les optimisations monocritères, on abrège simplement par optimisation.

Les caractéristiques de la fonction objective à tenir en compte dans un problème d'optimisation sont :

- La continuité
- La dérivabilité
- La linéarité
- La convexité

#### 1.4.2.2. Continuité

Considérons une fonction  $f: S \rightarrow \mathbb{R}^m$  où  $S$  est un sous-ensemble de  $\mathbb{R}^n$ . Soient  $x_0 \in S$

La fonction  $f$  est dite continue en  $x_0$  si  $f$  est définie en  $x_0$  et si

$$\lim_{x \rightarrow x_0} f(x) = f(x_0)$$

$$x \rightarrow x_0$$

Nous dirons que  $f$  est continue en  $x_0$  sur un ensemble  $S$  si  $f$  est continue en tout point de  $S$ .

On le note  $f \in C^0(S)$

#### 1.4.2.3. Dérivabilité

On se place dans des espaces vectoriels normés de dimension finie : on notera toujours  $\|v\|$  la norme, indépendamment de l'espace utilisé. Dans l'espace  $\mathbb{R}^n$ , on note  $(e_1, \dots, e_n)$  la base

canonique et on identifie le plus souvent une application linéaire et sa matrice dans la base canonique. Par ailleurs, on notera  $A^T$  la transposée d'une matrice  $A$ . L'espace  $\mathbb{R}^n$  est naturellement muni d'un produit scalaire canonique, noté  $\langle u, v \rangle$ , et défini pour deux vecteurs  $x, y \in \mathbb{R}^n$  par  $\langle x, y \rangle = x^T y$ .

#### 1.4.2.3.1. Différentielle

Soient  $S$  un ouvert de  $\mathbb{R}^n$  et  $a \in S$ . Une fonction  $f: S \rightarrow \mathbb{R}^p$  est dite différentiable en  $a$  s'il existe une application linéaire continue  $f: \mathbb{R}^n \rightarrow \mathbb{R}^p$  et une fonction  $\varepsilon: \mathbb{R}^n \rightarrow \mathbb{R}^p$  telles que

- Pour tout  $h \in \mathbb{R}^n$  tel que  $a + h \in S$ ,  $f(a + h) - f(a) = L(h) + \|h\|\varepsilon(h)$ ;
- $\lim_{h \rightarrow 0} \|\varepsilon(h)\| = 0$

Si  $L$  existe, elle est unique : on la note  $Df(a)$  (ou  $f'(a)$ ,  $df(a)$ ,  $D_a f \dots$ ) et on l'appelle différentielle de  $f$  en  $a$ .

On peut noter aussi :

$$f(a + h) = f(a) + Df(a)h + o(\|h\|) \quad (1-8)$$

La fonction  $f$  est dite différentiable, si elle est différentiable en tout point de  $S$ .

#### Exemple

- Si  $f$  est une application linéaire sur  $S = \mathbb{R}^n$  alors  $f$  est différentiable de différentielle  $f$ .
- Si  $f: \mathbb{R}^n \rightarrow \mathbb{R}$  est définie pour tout  $x \in \mathbb{R}^n$  par  $f(x) = x^T A x$  avec  $A$  une matrice  $n \times n$  à coefficients réels, alors  $f$  est différentiable de différentielle  $Df(a)h = a^T(A + A^T)h$ .

Considérons maintenant  $f: S \rightarrow \mathbb{R}^p$  est différentiable en  $a$ , on note  $\frac{\partial f}{\partial x_i(a)}$  ou  $\partial_i f(a)$  le vecteur

$Df(a)e_i \in \mathbb{R}^p$ . Grâce à la linéarité de la différentielle, on a pour tout  $h = (h_1, \dots, h_n) \in \mathbb{R}^n$

$$Df(a)h = \sum_{i=1}^n \frac{\partial f}{\partial x_i}(a)h_i \quad (1-9)$$

#### 1.4.2.3.2. Gradient

On se place ici dans le cas où  $p = 1$ , c'est-à-dire  $f: S \rightarrow \mathbb{R}$ . On considère à nouveau un ouvert  $S$  de  $\mathbb{R}^n$ . Si  $f$  est une application  $S \rightarrow \mathbb{R}$  différentiable en un point  $a \in U$ , telle que  $Df(a)$  n'est

pas l'application nulle, alors il existe un unique vecteur  $\nabla f(a) \in \mathbb{R}^n$ , appelé gradient de  $f$  en  $a$ , vérifiant pour tout  $h \in \mathbb{R}^n$ ,  $Df(a)h = \langle \nabla f(a), h \rangle$ .

On peut aisément vérifier que

$$\nabla f(a) = \left( \frac{\partial f}{\partial x_1(a)}, \dots, \frac{\partial f}{\partial x_n(a)} \right) \quad (1-10)$$

**Exemple**

Si  $f$  est une fonction linéaire, de matrice  $u^n \in \mathbb{R}^{1 \times n}$  (c'est à dire que  $f(x) = u^T x$  pour tout  $x \in \mathbb{R}^n$ ), alors  $\nabla f(a) = u$ . Si  $f(x) = x^T A x$ , alors  $\nabla f(a) = (A + A^T)a$

1.4.2.3.3. Différentielle seconde et Hessienne

Soient  $U$  un ouvert de  $\mathbb{R}^n$  et  $a \in U$ . Une fonction  $f : U \rightarrow \mathbb{R}$  est dite deux fois différentiable en  $a$  si les conditions suivantes sont vérifiées :

- $f$  est différentiable sur un voisinage  $V$  de  $a$  dans  $U$  ;
- L'application  $Df : V \rightarrow (\mathbb{R}^n)^*$ ,  $u \rightarrow Df(u)$  (on note  $(\mathbb{R}^n)^*$  l'espace vectoriel des formes linéaires sur  $\mathbb{R}^n$ , qui s'identifie par dualité à  $\mathbb{R}^n$  lui-même) est différentiable en  $a$ .

On note alors  $D^2f(a) = D(Df)(a)$  (application linéaire de  $\mathbb{R}^n$  dans  $(\mathbb{R}^n)^*$ ) la différentielle seconde ainsi obtenue. Cette application linéaire est représentée dans les bases canoniques (base canonique de  $\mathbb{R}^n$  et base duale canonique de  $(\mathbb{R}^n)^*$ ) par une matrice carrée appelée matrice Hessienne de  $f$  en  $a$ , souvent notée  $\nabla^2 f(a)$ .

On note  $\frac{\partial^2 f}{\partial x_i \partial x_j}(a)$  le vecteur  $\frac{\partial}{\partial x_i} \left( \frac{\partial f}{\partial x_j} \right)(a)$  et on abrège  $\frac{\partial^2 f}{\partial x_i \partial x_i}(a)$  en  $\frac{\partial^2 f}{\partial x_i^2}(a)$ . On peut alors aisément vérifier que

$$\nabla^2 f(a) = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2}(a) & \frac{\partial^2 f}{\partial x_2 \partial x_1}(a) & \dots & \frac{\partial^2 f}{\partial x_n \partial x_1}(a) \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_1 \partial x_n}(a) & \frac{\partial^2 f}{\partial x_2 \partial x_n}(a) & \dots & \frac{\partial^2 f}{\partial x_n^2}(a) \end{pmatrix} \quad (1-11)$$

Soit  $S \subseteq \mathbb{R}^n$  un ouvert et  $f : S \rightarrow \mathbb{R}$ .

On dit que  $f$  est de classe  $C^m$  sur  $S$  ( $f \in C^m(S)$ ) si toutes les dérivées partielles jusqu'à l'ordre  $m$  existent et sont continues.

#### 1.4.2.4. La linéarité

La linéarité désigne un rapport de dépendance très simple entre plusieurs variables : la variable  $z$  dépend linéairement des variables  $x_1, \dots, x_n$  ou on dit encore qu'elle s'exprime comme combinaison linéaire de ces variables, quand il existe des constantes  $a_1, \dots, a_n$  tels qu'on ait la relation :

$$z = a_1x_1 + \dots + a_nx_n \quad (1-12)$$

Une fonction linéaire est définie par :

$$f(x_i) = \sum_{i=1}^n a_i x_i \quad (1-13)$$

#### 1.4.2.5. La convexité

##### 1.4.2.5.1. Fonctions convexes

Géométriquement, une fonction est convexe sur un intervalle  $S$  si un segment tiré de tout point  $(x, f(x))$  à un autre point  $(y, f(y))$  est au-dessus de  $f$  dans  $S$ .

Soit  $S$  un ensemble convexe, Une fonction  $f$  définie sur un ensemble convexe  $S$  est dite convexe si :

$$\forall x, y \in S, \forall t \in [0; 1], \quad f(tx + (1 - t)y) \leq tf(x) + (1 - t)f(y) \quad (1-14)$$

La fonction est dite strictement convexe si :

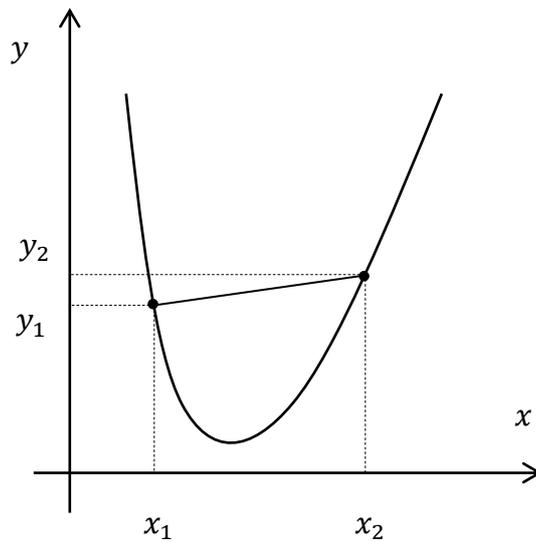
$$\forall x, y \in S, x \neq y \quad \forall t \in [0; 1], \quad f(tx + (1 - t)y) < tf(x) + (1 - t)f(y) \quad (1-15)$$

##### 1.4.2.5.2. Fonction concave

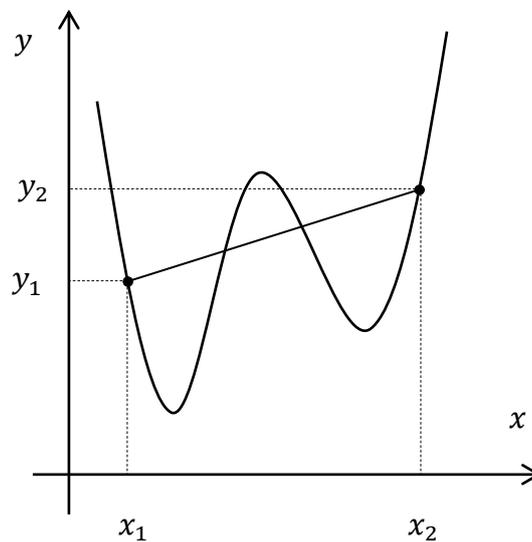
Une fonction est concave si  $-f$  est convexe. Il est facile de voir que chaque fonction linéaire est convexe et concave.

##### 1.4.2.5.3. Fonction non convexe

Une fonction est non convexe si elle n'est ni convexe ni concave. Un exemple familier est la fonction de sinus.



*Figure 1-4 : Fonction convexe*



*Figure 1-5 : Fonction non convexe*

On peut montrer aisément qu'une fonction  $f$  est convexe si

$$\forall x \in S, \quad f''(x) \geq 0 \quad (1-16)$$

Ou strictement convexe si

$$\forall x \in S, \quad f''(x) > 0 \quad (1-17)$$

### 1.4.3. Formulation du problème

Par définition, les contraintes sont les conditions à respecter par une solution pour que celle-ci soit valide. L'ensemble des zones de recherche où les contraintes de conceptions sont vérifiées est nommé espace réalisable ou domaine admissible. Inversement, l'espace irréalisable ou

domaine interdit, désigne l'ensemble des zones où au moins une des contraintes n'est pas respectée.

Les contraintes dans les problèmes d'optimisation décrivent la taille de l'ensemble de recherche, plus le nombre de contrainte augmente, plus la taille du domaine de définition de la fonction objective diminue.

Selon la formulation des problèmes d'optimisation, on peut distinguer l'optimisation sans contrainte et l'optimisation sous contrainte.

#### 1.4.3.1. Optimisation sans contrainte

C'est un problème dont aucune contrainte d'inégalité ou d'égalité n'est présente dans le problème. Par conséquent, un problème d'optimisation sans contrainte peut se formuler de façon suivante :

$$\begin{aligned} & \text{Minimiser } f(x) \\ & x \in S \subseteq \mathbb{R}^n \end{aligned} \tag{1-18}$$

On voit que le domaine de recherche pour une optimisation sans contrainte est l'ensemble  $S$  tout entier.

L'optimisation sans contrainte est rarement utilisée dans les applications.

#### 1.4.3.2. Optimisation avec contraintes

C'est une classification qui exige au moins une contrainte d'inégalité ou d'égalité dans le problème.

Généralement un problème mathématique d'optimisation avec contrainte s'écrit de la manière suivante :

$$\begin{aligned} & \text{Minimiser } f(x) \quad x \in S \subseteq \mathbb{R}^n \\ & \text{Sous les contraintes :} \\ & c_i(x) = 0 \quad i = 1, \dots, m \\ & d_j(x) \geq 0 \quad j = 1, \dots, p \end{aligned} \tag{1-19}$$

Les fonctions  $d_j(x)$ ,  $j = 1, \dots, p$  et  $c_i(x)$ ,  $i = 1, \dots, m$  sont respectivement des contraintes d'inégalités et d'égalités associées à la faisabilité du dispositif et à son adaptation au cahier des charges. Elles peuvent également représenter des relations entre les variables ajoutées par le concepteur pour garantir le bon conditionnement du problème.

Les fonctions  $f(x)$ ,  $c_i(x)$ , et  $d_j(x)$  peuvent être linéaires, non linéaires, continues ou discontinues. De ce fait une solution optimale du problème sera déduite à partir d'une méthode adéquate.

### 1.5. Exemple de quelques problèmes d'optimisation rencontrés fréquemment [4]

#### 1.5.1. Problème d'optimisation linéaire avec contrainte

Un problème d'optimisation linéaire est la recherche du maximum d'une fonction linéaire sur un domaine délimité par des contraintes linéaires.

Un problème d'optimisation linéaire prend donc la forme :

$$\begin{aligned} & \text{Minimiser } f = c^T x \\ & \text{Sous contrainte} \\ & Ax(\geq, =, \leq)b \end{aligned} \tag{1-20}$$

Où A et b sont respectivement une matrice et vecteur définissant les contraintes et c un vecteur reliant la fonction objective  $f$  aux variables x.

Les variables utilisées dans ce problème prennent des valeurs réelles. Le système de contraintes définissant un domaine éventuellement vide de solutions admissibles, l'objectif est d'obtenir la solution optimale quand il en existe vis-à-vis de la fonction objective.

Des résultats d'algèbre linéaire ont montré que le domaine des solutions admissibles est convexe (sauf s'il est vide) et que la solution optimale est un sommet du domaine (ou une face s'il y a plusieurs solutions optimales).

#### 1.5.2. Problème non-linéaire avec contrainte

Un problème d'optimisation avec contraintes non linéaires prend la forme :

$$\begin{aligned} & \text{Minimiser } f(x) \\ & \text{Sous les contraintes} \\ & g(x_i) \leq 0, \quad i = 1, \dots, m \\ & h(x_i) \leq 0, \quad i = 1, \dots, p \end{aligned} \tag{1-21}$$

La fonction objective et/ou les contraintes utilisées dans ce problème n'ont pas de forme linéaire. Les variables des problèmes non-linéaires peuvent prendre des valeurs réelles, entières ou booléennes.

La résolution de ces problèmes n'est généralement pas aisée puisque la propriété de convexité du domaine des solutions n'est plus valable.

### 1.5.3. Problèmes convexes

Un problème convexe d'optimisation est un problème où toutes les contraintes sont des fonctions convexes, et l'objectif est une fonction convexe si on minimise, ou une fonction concave si on maximise.

Si  $f: S \rightarrow \mathbb{R}^n$  est convexe avec  $S$  convexe, alors, en cas d'existence d'un minimum, ce minimum est unique (optimal global).

### 1.5.4. Problèmes non-convexes

Un problème d'optimisation non convexe est un problème où les contraintes et/ou la fonction d'adaptation sont non convexes. Un tel problème peut avoir des régions admissibles multiples et des points optimaux localement dans chaque région. Ceci présente alors une difficulté par rapport à l'ensemble convexe.

#### **Remarque :**

Il faut cependant préciser qu'en général un problème d'optimisation dans n'importe quelle discipline, ne peut être résolu avec suffisamment de cohérence sans une bonne connaissance du dispositif à traiter. En effet, les méthodes d'optimisation ne sont pas des outils magiques, mais doivent être considérées comme des outils d'aide à la recherche de solutions.

## 1.6. Résolution du problème d'optimisation [29]

Comme on l'a déjà énoncé dans ce chapitre, La nature des variables, des domaines de définition et des critères ou objectif à optimiser va influencer le choix de la méthode d'optimisation à utiliser.

Les méthodes d'optimisations sont classées en deux grandes :

- Les méthodes classiques
- Les méthodes modernes

## 1.7. Conclusion

Dans ce chapitre on a énoncé un aperçu global sur le problème d'optimisation. Pour avoir des résultats optimaux à ces problèmes, plusieurs méthodes appartenant chacune aux deux classes énoncées précédemment sont à notre disposition. Le choix d'une méthode ne se fait pas par hasard, en effet quelques conditions sont réunies avant l'utilisation d'une méthode.

Dans le chapitre suivant, on va essayer de développer les méthodes d'optimisation classiques en énonçant les hypothèses correspondantes et leurs principes de fonctionnement.

# Chapitre 2. LES METHODES CLASSIQUES

## 2.1. Introduction

Les méthodes classiques sont constituées par les méthodes déterministes dites exactes regroupent une des grandes classes des méthodes d'optimisation. Ces méthodes se caractérisent par le fait qu'elles permettent d'obtenir une solution dont l'optimalité est garantie en un temps fini même en cas d'une certaine complexité mathématique. En effet, ceci est obtenu en parcourant systématiquement l'espace de recherche (ensemble des solutions admissibles). Malgré cette garantie, l'efficacité de ces méthodes dépend d'un certain nombre d'hypothèses concernant le problème. Ces méthodes déterministes nécessitent par exemple un certain nombre de caractéristiques de la fonction objective, comme la stricte convexité, la continuité ou encore la dérivabilité. Des méthodes d'optimisation déterministes avec leurs hypothèses d'utilisation et leurs principes seront détaillées dans ce chapitre pour bien les comprendre et les distinguer parmi toutes les méthodes d'optimisation existantes.

### ***Hypothèse générale:***

- Etant donné que dans le domaine d'application en génie électrique, on manipule dans la plupart du temps des variables appartenant à l'ensemble des entiers relatifs. Alors dans ce cas, dans la suite, le domaine de recherche est un sous espace de  $\mathbb{R}^n$ .

$$U \subseteq \mathbb{R}^n$$

- Pour assurer la convergence vers un minimum global, la fonction objective doit être convexe.

Les méthodes déterministes sont généralement employées pour les cas suivants :

- Les problèmes d'optimisation linéaire
- Les problèmes d'optimisation non linéaire

## 2.2. Les problèmes d'optimisation linéaire [24]

Un problème d'optimisation linéaire également désigné par le nom de programmation linéaire est un problème d'optimisation comme les autres dans lesquelles la fonction objective et les fonctions définissant les contraintes sont linéaires.

Un problème d'optimisation linéaire peut prendre plusieurs formes à savoir : la forme générale, la forme canonique et la forme standard, il faut aussi noter qu'un problème sous forme générale peut être transformé d'abord en un problème équivalent sous forme canonique, puis en un problème équivalent sous forme standard.

Un problème d'optimisation linéaire sous forme générale est un problème de la forme :

*Maximiser*  $f(x)$

*Sous la contrainte :*

$$x \in U$$

(2-1)

$$U = \{x \in R^n | g_1(x) \leq 0, \dots, g_m(x) \leq 0\}$$

**Remarque :**

- Minimiser une fonction objective  $f$  quelconque revient à Maximiser la fonction  $-f$
- Les contraintes de la forme  $g_1(x) \geq 0$  ou même  $g_1(x) = 0$  peuvent être écrites respectivement sous la forme  $(-g_i(x) \leq 0)$  et  $(g_i(x) \leq 0 \text{ et } -g_i(x) \leq 0)$ .

Un problème d'optimisation linéaire sous forme canonique est un problème de la forme :

*Maximiser :*

$$f(x) = \sum_{j=1}^n c_j x_j$$

(2-2)

*Sous les contraintes :*

$$\sum_{j=1}^n a_{ij} x_j \leq b_i \quad i = 1, \dots, m$$

$$x_j \geq 0 \quad i = 1, \dots, n$$

En écriture matricielle, on a :

Maximiser  $cx$

Sous les contraintes : (2-3)

$$Ax \leq b$$

$$x \geq 0$$

Où :

–  $c$  est une matrice ligne :  $c = (c_1 \dots c_n)$

–  $x$  une matrice colonne :  $x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$

–  $a$  une matrice de dimension  $n \times m$  :  $A = (a_{ij})_{1 \leq i \leq n, 1 \leq j \leq m}$

–  $b$  une matrice colonne :  $b = \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix}$

A tout problème d'optimisation linéaire sous forme canonique, on peut associer un problème d'optimisation linéaire sous forme standard de la manière suivante :

Soit le système sous forme canonique ci-dessus,

En supposant que les lignes de  $A$  sont linéairement indépendantes, c'est-à-dire :

$$\text{rang}(A) = m$$

$$n \geq m$$

On pose  $p = m$  et  $q = m + n$

La forme standard équivalente est la suivante :

Maximiser :

$$f(x) = \sum_{j=1}^n c_j x_j \tag{2-4}$$

Sous les contraintes :

$$x_{q+i} + \sum_{j=1}^n a_{ij} x_j = b_i \quad i = 1, \dots, p$$

$$x_j \geq 0 \quad i = 1, \dots, q$$

Où  $x_{q+i}$   $i = 1, \dots, p$  sont appelés variables d'écart.

Un problème d'optimisation linéaire sous forme standard s'écrit sous forme matricielle comme :

Maximiser  $cx$

Sous les contraintes :

$$Ax \leq b$$

$$x \geq 0$$

(2-5)

Où

$$- A = \begin{pmatrix} a_{1j} & \dots & a_{1j} & 1 & 0 & \dots & 0 \\ a_{2j} & \dots & a_{2j} & 0 & 1 & \dots & 0 \\ \vdots & \dots & \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{mj} & \dots & a_{mj} & 0 & 0 & \dots & 1 \end{pmatrix}$$

$$- x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \\ \vdots \\ x_{m+n} \end{pmatrix}$$

$$- c = (c_1 \quad \dots \quad c_n \quad 0 \quad \dots \quad 0)$$

### 2.2.1. Existence de solution

Avant la résolution d'un problème d'optimisation linéaire, il est primordial de connaître si le problème en question admet ou pas une solution.

En général, on a 2 cas dans laquelle la résolution d'un problème d'optimisation linéaire n'aboutit à aucune solution finie :

- Le premier cas est lorsque l'intersection des contraintes constitue un ensemble vide. Par convention, la valeur optimale du problème d'optimisation vaut  $+\infty$  et on dit alors que le problème n'est pas réalisable.
- Le second cas se présente lorsque le problème d'optimisation linéaire est réalisable mais en faisant la résolution, on trouve  $-\infty$  comme valeur optimale. On dit que le problème n'est pas borné ou est non borné dans ce cas de figure.

### 2.2.2. Condition d'optimalité

**Condition nécessaires et suffisantes**

Soient :

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

$$y \in \mathbb{R}^m$$

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

Le point  $x^*$  est solution du problème linéaire si et seulement si les vecteurs  $y$  et  $s$  existe tels que :

$$\begin{aligned} A^T y + s &= c \quad s \geq 0 \\ Ax &= b \quad x \geq 0 \\ x^T s &= 0 \end{aligned} \tag{2-6}$$

Les variables  $s$  et  $y$  dans ces conditions s'appellent multiplicateurs de Lagrange ou variable duale. L'ensemble des triplets  $(x, y, s)$  forment ce qu'on appelle ensemble des solutions primales duales

Ces conditions sont obtenues dans le cas particulier de la théorie générale d'optimisation différentiable en dimension finie c'est-à-dire la condition de Karush, Kuhn et Tucker

### 2.2.3. Méthode du simplexe

La méthode du simplexe est un algorithme proposé en 1947 par G. B. Dantzig pour résoudre les problèmes d'optimisation linéaire. La solution optimale est approchée par étapes grâce à des itérations successives.

#### **Principe**

Les grandes lignes de la méthode du simplexe sont les suivantes :

##### – Etape 1

Mettre le problème sous forme standard en introduisant des variables d'écart

*Maximiser :*

$$z = c_1 x_1 + c_2 x_2 + \dots + c_n x_n$$

*Sous les contraintes :*

$$x_{n+1} = b_1 + a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n \tag{2-7}$$

$$x_{n+2} = b_2 + a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n$$

⋮

$$x_{n+m} = b_m + a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n$$

Où les variables de gauches c'est-à-dire  $x_{n+1} \quad x_{n+2} \quad \dots \quad x_{n+m}$  sont appelés **variable de base**, et les variables de droite qui sont  $x_1 \quad x_2 \quad \dots \quad x_n$  sont appelés **variable hors base**.

##### – Etape 2

Déterminer une solution de base initiale.

Une solution de base est avant tout une solution admissible ; elle satisfait l'ensemble des contraintes et conditions de signe.

En tenant compte de la première étape, toute solution de base comporte deux catégories de solution pour les variables.

→ Les solutions pour les variables hors base

D'une manière générale, initialement, on prend 0 comme solution de toutes les variables hors base.

→ Les solutions pour les variables dans la base

Elles sont obtenues en remplaçant les valeurs prises comme solutions pour les variables hors base dans les  $m$  équations définissant les contraintes.

De façon générale, si un problème comprend  $m$  contraintes et  $n$  variables réelles, pour qu'une solution soit une solution de base, il faut et il suffit qu'elle soit solution admissible et qu'elle admette au moins  $n$  variables hors base et au plus  $m$  variables dans la base.

### – Etape 3

Tester l'optimalité de cette solution de base s'il s'agit ou non de la solution optimale, Si cette solution de base est la solution optimale, alors le problème est résolu, si elle ne l'est pas, alors on passe à l'étape suivante.

### – Etape 4

Changer de solution de base puis reprendre l'étape 3.

### ***Choix d'une solution de base meilleure***

→ On étudie à partir de la fonction cout  $f(x)$  ; quelle sera le variable à changer permettant d'accroître au maximum le profit.

La variable choisie sera appelée variable entrante.

Le critère de sélection de Dantzig de la variable entrante consiste, dans la fonction économique exprimée exclusivement en fonction des variables hors-base, à sélectionner la variable affectée du coefficient strictement positif le plus élevé, Cette règle est appelée règle du plus grand gain marginal

Soit  $x_k$  la variable choisie.

→ Sans modifier les autres variables, on essaye de trouver la contrainte la plus forte permettant d'augmenter  $x_k$  en gardant les variables d'écart positif;

Soit  $x_{n+l} = b_l + a_{l1}x_1 + a_{l2}x_2 + \dots + a_{ln}x_n$  cette contrainte.

→ Augmenter  $x_k$  jusqu'à ce que l'on ait  $x_{n+l} = 0$

A ce moment  $x_{n+l}$  est devenu variable hors base et sera appelé variable sortante et  $x_k$  qui est un variable entrante devient à son tour une variable de base.

→ Exprimer ensuite les nouvelles variables de base ainsi que  $z$  en fonction des nouvelles hors base.

→ En remplaçant  $x_{n+l}$  par 0 dans la nouvelle et en gardant les valeurs des variables hors base par ses anciennes valeurs, on obtient une nouvelle solution réalisable.

Chaque changement de solution de base constitue une itération.

#### 2.2.4. Méthode du Tableau

Lorsqu'on essaye de trouver le maximum à la main (sous condition que  $n$  et  $m$  ne sont pas trop grands) la tâche peut être sensiblement facilitée par une disposition astucieuse des calculs, sous forme de tableaux. Cette disposition peut également aider à mieux comprendre le fonctionnement de la méthode du simplexe, et constituer un guide pour la réalisation d'un programme d'ordinateur effectuant le travail automatiquement.

##### **Exemple :**

Soit à résoudre le système suivant :

$$\max \{400x_1 + 600x_2 + 800x_3 + 1000x_4\}$$

Sous contraintes :

$$8x_1 + 10x_2 + 12x_3 + 15x_4 \leq 6000$$

$$2x_1 + 2x_2 + 4x_3 + 5x_4 \leq 1500 \quad (2-8)$$

$$x_3 + x_4 \leq 300$$

$$x_1 + x_2 + x_3 + x_4 \leq 450$$

$$x_1, x_2, x_3, x_4 \geq 0$$

Ecriture sous forme de tableau

$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$	$x_8$	
8	10	12	15	1	0	0	0	6000
2	2	4	5	0	1	0	0	1500
0	0	1	1	0	0	1	0	300
1	1	1	1	0	0	0	1	450
400	600	800	1000	0	0	0	0	0

*Tableau 2-1 : Ecriture d'un problème d'optimisation linéaire sous forme d'un tableau*

**Itération 1**

On essaye de trouver la colonne et la ligne pivot

Colonne du pivot : maximum positif de la dernière ligne (critère de Dantzig)

Ligne du pivot : celle qui minimise  $\frac{S_i}{R_i}$  parmi toutes les lignes avec  $R_i$  positif ( $R_i$ , respectivement  $S_i$  sont les éléments de la  $i$ ème ligne se trouvant sur la colonne pivot, respectivement la dernière colonne.)

Et l'intersection de la ligne et de la colonne pivot nous donne le pivot

$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$	$x_8$	
8	10	12	15	1	0	0	0	6000
2	2	4	5	0	1	0	0	1500
0	0	1	1	0	0	1	0	300
1	1	1	1	0	0	0	1	450
400	600	800	1000	0	0	0	0	0

*Tableau 2-2 : Mise en évidence de la colonne et ligne pivot*

Diviser la ligne pivot par le coefficient pivot

$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$	$x_8$	
8	10	12	15	1	0	0	0	6000
$\frac{2}{5}$	$\frac{2}{5}$	$\frac{4}{5}$	1	0	$\frac{1}{5}$	0	0	300
0	0	1	1	0	0	1	0	300
1	1	1	1	0	0	0	1	450
400	600	800	1000	0	0	0	0	0

*Tableau 2-3 : Tableau obtenu après division de la ligne pivot par le pivot*

Annuler les autres coefficients de la colonne pivot :

$$L_i = L_i - R_i \cdot L_{pivot} \quad (2-9)$$

$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$	$x_8$	
2	4	0	0	1	-3	0	0	1500
$\frac{2}{5}$	$\frac{2}{5}$	$\frac{4}{5}$	1	0	$\frac{1}{5}$	0	0	300
$-\frac{2}{5}$	$-\frac{2}{5}$	$\frac{1}{5}$	0	0	$-\frac{1}{5}$	1	0	0
$\frac{3}{5}$	$\frac{3}{5}$	$\frac{1}{5}$	0	0	$-\frac{1}{5}$	0	1	150
0	200	0	0	0	-200	0	0	-30000

*Tableau 2-4 : Tableau obtenu après annulation des autres coefficients de la colonne pivot*

**Itération 2 :**

Le tableau *Tableau 2-4* précédent devient le tableau initial pour l'itération 2

*Tableau 2-2* ← *Tableau 2-4*

Et ainsi de suite jusqu'à ce que tous les coefficients de la dernière ligne soient tous négatifs, à ce moment, l'optimum est atteint.

2.3. Les problèmes d'optimisation non linéaire [42]

D'une manière générale, les problèmes d'optimisation non-linéaire peuvent être divisés en deux grandes classes :

- Les problèmes sans contraintes
- Les problèmes avec contraintes

Avant d'exposer les solutions permettant de résoudre ces 2 problèmes d'optimisation non linéaires, il convient d'abord de souligner qu'on ne peut pas espérer de concevoir une méthode d'optimisation capable de résoudre efficacement tous les problèmes d'optimisation non-linéaire puisque ces problèmes sont trop divers. En fait il y a de nombreuses méthodes, et chacune d'elle est orientée vers une certaine famille restreinte des problèmes d'optimisation. Par contre la plupart des algorithmes de résolution des problèmes d'optimisation avec ou sans contrainte fonctionnent selon un schéma général consistant, à chaque itération, à se rapprocher du minimum par la résolution d'un sous-problème de minimisation. Evidemment, cette stratégie n'a de sens que si ces sous-problèmes sont plus faciles à résoudre que le problème initial.

### 2.3.1. Problèmes sans contraintes

Un problème est dit sans contraintes quand le domaine de recherche est l'ensemble  $\mathbb{R}^n$  tout entier. En effet, aucune contrainte inégalité ou égalité n'est présente. La forme générique d'un problème d'optimisation non linéaire sans contraintes est donc :

$$\begin{aligned} & \text{Minimiser : } f(x) \\ & x \in \mathbb{R}^n \\ & f(x) \text{ non linéaire} \end{aligned} \tag{2-10}$$

#### 2.3.1.1. Conditions d'optimalité

Soit

$$\begin{aligned} & f: \mathbb{R}^n \rightarrow \mathbb{R}, \\ & x \in \mathbb{R}^n \end{aligned}$$

Une fonction continuellement différentiable et  $x^* \in \mathbb{R}^n$ .

##### 2.3.1.1.1. Condition nécessaire du premier ordre

Sachant que  $f'(x; d) = \nabla f(x) \cdot d$

**Théorème :**

Soient  $f: \mathbb{R}^n \rightarrow \mathbb{R}$  continuellement différentiable et  $x \in \mathbb{R}^n$  S'il existe un vecteur  $d$  tel que la dérivée directionnelle de  $f$  dans la direction  $d$  au point  $x^*$  (notée  $f'(x^*; d)$ ) est strictement inférieure à zéro, alors  $d$  est une direction de descente de  $f$  en  $x^*$ .

Réciproquement à ce théorème, s'il existe un vecteur  $d$  tel que  $\nabla f(x^*) \cdot d < 0$ , alors  $d$  est une direction de descente de  $f$  en  $x^*$ .

Si  $x^*$  est un minimum local de  $f$  alors  $\nabla f(x^*) = 0$ .

#### 2.3.1.1.2. Condition nécessaire du second ordre :

***Théorème:***

Soient  $f: \mathbb{R}^n \rightarrow \mathbb{R}$  deux fois continuellement différentiable et  $x^* \in \mathbb{R}^n$ . Si  $x^*$  est un minimum local de  $f$ , alors  $\nabla f(x^*) = 0$  et  $\nabla^2 f(x^*)$  est semi définie positive.

Dans le cas où  $f$  est deux fois continuellement différentiable,

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

Si de plus  $f$  est convexe, cette condition nécessaire du premier ordre est également suffisante pour que  $x^*$  soit un minimum global.

#### 2.3.1.1.3. Conditions suffisantes

Les conditions précédentes doivent être remplies par tous les minimums locaux ; cependant, tout vecteur qui vérifie ces conditions n'est pas obligatoirement un minimum local.

***Théorème :*** (condition suffisante du second ordre).

Soient  $f: \mathbb{R}^n \rightarrow \mathbb{R}$  deux fois continuellement différentiable et  $x^* \in \mathbb{R}^n$ . Si  $\nabla f(x^*) = 0$  et  $\nabla^2 f(x)$  est définie positive, alors  $x^*$  est un minimum local de  $f$ .

Soit

$f: \mathbb{R}^n \rightarrow \mathbb{R}$  deux fois continuellement différentiable

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

Les conditions suffisantes pour qu'un vecteur  $x^*$  soit un minimum local de  $f$  sont :

$\nabla f(x^*) = 0$  et  $\nabla^2 f(x^*)$  est définie positive

Selon le cas, pour résoudre le problème d'optimisation non linéaire sans contraintes, on a les méthodes suivantes :

- Les méthodes basées sur le gradient
- Les méthodes de Newton
- Les méthodes utilisant des directions conjuguées
- La méthode du simplexe (ou méthode de Nelder et Mead)

**Hypothèse**

$f$  différentiable à l'exception des méthodes de directions conjuguées (sauf dans le cas particulier de la méthode du gradient conjugué) et de la méthode de simplexe.

2.3.1.2. La méthode de descente basée sur le gradient

Historiquement, les méthodes de gradient sont les plus anciennes. Ces méthodes permettent de résoudre des problèmes d'optimisation non linéaires.

**Hypothèse**

La fonction objective  $f: \mathbb{R}^n \rightarrow \mathbb{R}$  et  $x \in \mathbb{R}^n$  est une fonction différentiable de  $\mathbb{R}^n$  dans  $\mathbb{R}$

**Principe**

La méthode de descente basée sur le gradient est une des méthodes d'optimisations permettant de minimiser une fonction objective différentiable de  $\mathbb{R}^n$  dans  $\mathbb{R}$ .

Elle se repose sur le principe de la descente itérative par le biais d'un algorithme itératif, c'est-à-dire partant d'un vecteur  $x_0 \in \mathbb{R}^n$ , on génère une suite de vecteur  $x_1, x_2, \dots$  de  $\mathbb{R}^n$  en respectant la décroissance de la fonction  $f$  à chaque itération :

$$f(x_{k+1}) < f(x_k) \quad \forall k \in \mathbb{N}$$

Pour les méthodes basées sur le gradient, les vecteurs sont générés de la manière suivante :

$$x_{k+1} = x_k + \alpha_k d_k \tag{2-11}$$

Où

- $d_k$  est ici une direction de descente
- $\alpha_k > 0$  et tend vers 0

**Direction de descente**

Soient :

$$f: \mathbb{R}^n \rightarrow \mathbb{R} \text{ et } x \in \mathbb{R}^n$$

Un vecteur  $d$  est une direction de descente de  $f$  au point  $x$  si  $\exists \alpha_{max} > 0$  tel que  $f(x + \alpha \cdot d) < f(x) \forall \alpha \in [0, \alpha_{max}]$

**Théorème**

Soit  $f: \mathbb{R}^n \rightarrow \mathbb{R}$  continuellement différentiable et  $x^* \in \mathbb{R}^n$ . S'il existe un vecteur  $d$  tel que la dérivée directionnelle de  $f$  dans la direction  $d$  au point  $x^*$  (notée  $f'(x^*; d)$ ) est strictement inférieure à zéro, alors  $d$  est une direction de descente de  $f$  en  $x^*$

En outre, selon, une propriété connue de l'analyse,

$$f'(x^*; d) = \nabla f(x) \cdot d \quad (2-12)$$

Inversement, on peut donc énoncer que s'il existe un vecteur  $d$  tel que  $\nabla f(x^*) \cdot d < 0$ , alors  $d$  est une direction de descente de  $f$  en  $x^*$ .

La méthode du gradient s'agit lors du cas particulier où

$$d_k = -\nabla f(x^*) \quad (2-13)$$

Les suites de vecteurs sont ainsi successivement générées par la méthode du gradient de la manière suivante :

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k) \quad \alpha_k > 0 \quad (2-14)$$

Le vecteur  $x_0$  peut être choisi arbitrairement, il est à noter quand même que l'idéal est que ce vecteur soit proche d'une solution.

Le problème est maintenant le choix de  $\alpha_k$ . On a différentes stratégies pour ce choix, celle la plus utilisée est désignée sous l'appellation « règle de minimisation ».

Elle consiste à choisir, à chaque itération,  $\alpha_k$  comme étant la solution optimale du problème de minimisation monodimensionnelle de  $f$  le long de la demi-droite définie par le point  $x_k$  et la direction  $-\nabla f(x_k)$ . Un tel sous-problème peut être résolu de diverses manières, par exemple par la méthode de Newton.

Le point minimum est obtenu lorsque  $\nabla f(x^*) = 0$  puisque dans ce cas  $x_{k+1} = x_k$ .

### Algorithme

[Initialisation]

Poser  $k = 0$  ;

Choisir  $x_0$  dans un voisinage de  $x_{min}$  ;

Choisir  $\varepsilon > 0$  ;

[Traitement]

Tant que  $(x_{k+1} - x_k \geq \varepsilon)$  faire

Résoudre par simplexe  $\min \{x_k - \alpha_k \nabla f(x_k)\}$  ;

Poser  $x_{k+1} = x_k - \alpha_k \nabla f(x_k)$  ;

Poser  $k = k + 1$  ;

Fin tant que

Retourner  $x_k$  ;

### 2.3.1.3. La méthode de Newton

#### **Hypothèses**

- $f$  est de classe  $C^2$  c'est à dire  $f$  est deux fois continuellement différentiable
- La matrice Hessienne de  $f$  est inversible

#### **Principe**

La méthode de Newton est aussi une méthode itérative permettant de minimiser à chaque itération, l'approximation quadratique de la fonction objective.

D'abord, essayons de rapprocher  $f$  dans le voisinage de  $x_k$  par son développement de Taylor de second ordre :

$$g_k(x) \approx f(x_k) + \nabla f(x_k) \cdot (x - x_k) + \frac{1}{2} (x - x_k) \cdot \nabla^2 f(x_k) \cdot (x - x_k) \quad (2-15)$$

La condition nécessaire d'avoir le minimum de  $g_k(x)$  est :

$$\nabla g_k(x) = 0 \quad (2-16)$$

Soit :

$$\nabla g_k(x) = \nabla f(x_k) + \nabla^2 f(x_k) \cdot (x - x_k) = 0 \quad (2-17)$$

On choisit alors le point  $x_{k+1}$  comme minimum de  $g_k(x)$ . Par ailleurs, le minimum de  $g$  est réalisé lorsque  $\nabla g_k(x) = 0$  soit  $\nabla g_k(x_{k+1}) = 0$ , on a alors

$$x_{k+1} = x_k - [\nabla^2 f(x_k)]^{-1} \nabla f(x_k) \quad (2-18)$$

$\nabla^2 f(x_k) = H[f](x_k)$  : Matrice Hessienne de  $f$

La relation entre  $x_{k+1}$  et  $x_k$  précédente définit les itérations de la méthode de Newton.

Pourtant, La méthode nécessite l'Hessienne de la fonction  $f$  et son inverse, ce qui entraîne une complication énorme en terme de calcul.

On a alors recours à l'équation équivalente suivante :

$$x_{k+1} = x_k + d_k \quad (2-19)$$

Où  $d_k$  est l'unique solution du système linéaire :

$$H[f](x_k) d_k = -\nabla f(x_k) \quad (2-20)$$

$d_k$  est appelée direction de Newton.

**Algorithme**

Poser  $k = 0$  ;  
 Choisir  $x_0$  dans un voisinage de  $x_{min}$  ;  
 Choisir  $\varepsilon > 0$  ;  
 Tant que  $(x_{k+1} - x_k \geq \varepsilon)$  faire  
     Résoudre le système linéaire  $H[f](x_k)d_k = -\nabla f(x_k)$  ;  
     Poser  $x_{k+1} = x_k + d_k$  ;  
     Poser  $k = k + 1$  ;  
 Fin tant que  
 Retourner  $x_k$  ;

**Les méthodes de quasi-Newton**

La difficulté pour le calcul de la matrice Hessien et de son inverse est souvent un facteur de blocage pour la méthode de Newton, pour éviter le calcul difficile de ces matrices, les méthodes de quasi-Newton permettent de générer une séquence de matrices symétriques définies positives qui sont des approximations toujours améliorées de la matrice Hessien ou de son inverse au point  $x_k$ .

Une méthode de quasi-Newton a pour but de résoudre le problème du type :

$$\begin{aligned} x_{k+1} &= x_k + \alpha_k d_k \\ d_k &= -B_k g_k \quad \alpha > 0 \\ B_k &= H_k'^{-1} \quad g_k = \nabla f(x_k) \end{aligned} \tag{2-21}$$

$B_k$  : Matrice approximative du Hessien de  $f$  en  $x_k$

Pour pouvoir remplacer la matrice Hessien et son inverse par la matrice  $H_k'$  et  $B_k$ , ces 2 matrices doivent vérifier la relation de quasi-Newton suivante :

$$H_{k+1}(x_{k+1} - x_k) = \nabla f(x_{k+1}) - \nabla f(x_k) \tag{2-22}$$

Où

$$x_{k+1} - x_k = B_{k+1}(\nabla f(x_{k+1}) - \nabla f(x_k)) \tag{2-23}$$

**Principe**

Supposant qu'à l'itération  $k + 1$ ,  $B_k$  est connue, la formule suivante est appliquée pour mettre à jour  $B_k$  afin d'obtenir  $B_{k+1}$ .

On dispose alors 3 méthodes de mise à jour et qui vérifient la relation de quasi-Newton:

Partant de la relation (2-23), on a

$$y_k = H'_k s_k \quad \text{et} \quad s_k = B_{k+1} y_k \quad (2-24)$$

Avec :

$$s_k = x_{k+1} - x_k$$

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

a. Méthode de Broyden

La Formule de Broyden est une formule de Correction de Rang 1 :

$$B_{k+1} = B_k + \frac{(s_k - B_k y_k)(s_k - B_k y_k)^T}{(s_k - B_k y_k)^T y_k} \quad (2-25)$$

Le problème avec cet algorithme est que si la matrice  $B_k$  est symétrique et définie positive, la matrice  $B_{k+1}$  ne l'est pas forcément.

b. Méthode de Davidon, Fletcher et Powell (DFP)

La formule de Davidon, Fletcher et Powell suivante est une formule de correction de rang 2.

$$B_{k+1} = B_k + \frac{s_k s_k^T}{s_k^T y_k} - \frac{B_k y_k y_k^T B_k}{y_k^T B_k y_k} \quad (2-26)$$

La matrice  $B_{k+1}$  obtenue par cette formule est définie positive sous certaines conditions.

c. Méthode de Broyden, Fletcher, Goldfarb et Shanno (BFGS)

La formule de mise à jour de Broyden, Fletcher, Goldfarb et Shanno est une formule de correction de rang 2 qui s'obtient à partir de la formule de Davidon, Fletcher et Powell précédente en intervertissant les rôles de  $s_k$  et  $y_k$ .

La formule obtenue permet de mettre à jour une approximation  $H'_k$  du Hessien possédant les mêmes propriétés, à savoir  $H'_{k+1} > 0$  si  $H'_k > 0$  et vérifiant la relation de quasi-Newton :

$$y_k = H'_k s_k$$

$$H'_{k+1} = H'_k + \frac{y_k y_k^T}{y_k^T s_k} - \frac{H'_k s_k s_k^T H'_k}{s_k^T H'_k s_k} \quad (2-27)$$

2.3.1.4. Les méthodes utilisant des directions conjuguées

Les méthodes de direction conjuguée sont des méthodes de descente pouvant résoudre les problèmes quadratiques en au plus  $n$  itérations.

Ces méthodes n'utilisent pas la dérivée de la fonction objective sauf le cas exceptionnel comme les méthodes du gradient conjugué.

**Hypothèses**

$f(x)$  est une fonction quadratique

**Principe**

Soient :

- Une fonction quadratique

$$f(x) = \frac{1}{2}x \cdot Ax + b \cdot x + c \quad x \in \mathbb{R}^n \quad (2-28)$$

$A \in \mathbb{R}^{n \times n}$  est symétrique et définie positive

$b \in \mathbb{R}^n$  et  $c \in \mathbb{R}$

- Un ensemble de  $n$  directions non nulles  $\{d_0, d_1, \dots, d_{n-1}\}$  A-conjuguées c'est-à-dire :

$$d_i \cdot Ad_j = 0 \quad \forall i, j \text{ tels que } i \neq j$$

Ces directions conjuguées se construisent comme suit :

Ces directions A-conjuguées peuvent être générées par la procédure de *Gram-Schmidt* à l'aide d'un ensemble de vecteurs linéairement indépendants quelconque  $\xi_0, \dots, \xi_k$  de telle façon que les sous-espaces générés par ces 2 ensembles sont égaux.

La procédure fonctionne de la manière suivante :

En commençant par  $d_0 = \xi_0$

A la construction de  $d_{i+1}$  on a déjà construit les  $i + 1$  premières directions conjuguées, alors,

$$d_{i+1} = \xi_{i+1} + \sum_{m=0}^i c_{(i+1)m} d_m \quad (2-29)$$

Les coefficients  $c_{(i+1)m}$  sont choisis afin d'assurer la A-conjugaison des directions  $d_0, \dots, d_{i-1}$  ainsi obtenu, c'est-à-dire

$$d_{i+1} \cdot Ad_j = 0 \quad \forall j = 0, \dots, i$$

La méthode consiste à générer la suite de vecteurs  $x_{k+1}$  à partir d'un vecteur de départ  $x_0$  choisi arbitrairement où

$$x_{k+1} = x_k + \alpha_k d_k \quad k = 0, \dots, n - 1 \quad (2-30)$$

Les  $\alpha_k$  sont obtenus par minimisation monodimensionnelle le long de  $d_k$  à chaque itération.

Le résultat attendu pour les méthodes utilisant les directions conjuguées est qu'à chaque itération, la fonction objective  $f$  est minimisée sur le sous espace généré par les  $k$  premières directions conjuguées utilisées par l'algorithme.

A la  $n^{\text{ième}}$  itération, le minimum global de  $f$  sera inclus dans ce sous espace puisque à cette itération, l'espace vectoriel généré sera  $\mathbb{R}^n$ .

### **La méthode du gradient conjugué**

Cette méthode est une des familles des méthodes des directions conjuguées exploitant les vecteurs gradients pour construire les directions conjuguées.

Les vecteurs linéairement indépendants sont donc :

$$\xi_0 = -\nabla f(x_0), \dots \dots \xi_{n-1} = -\nabla f(x_{n-1}) \quad (2-31)$$

#### 2.3.1.5. La méthode du simplexe ou méthode de Nelder et Mead

Cette méthode est très différente de la méthode du simplexe utilisée pour le problème d'optimisation linéaire.

#### **Principe**

Différemment des méthodes précédentes qui consistent à chaque itération à générer un point  $x_{k+1}$  à partir d'un point  $x_k$  sur le long d'une direction, la méthode de simplexe fonctionne suivant le principe ci-dessous :

Soient :

- $x_0, x_1, \dots, x_n$ ,  $n + 1$  points appelé simplexe
- $f(x_{min})$  le point parmi les  $n + 1$  dont le cout est minimum

$$f(x_{min}) = \min_{i=0, \dots, n} f(x_i) \quad (2-32)$$

- $f(x_{max})$  le point parmi les  $n + 1$  dont le cout est maximum

$$f(x_{max}) = \max_{i=0, \dots, n} f(x_i) \quad (2-33)$$

- $x_c$  le centre du polyèdre formé par les points du simplexe sauf  $x_{max}$

$$x_c = \frac{1}{n} (-x_{max} + \sum_{i=0}^n x_i) \quad (2-34)$$

La méthode consiste à remplacer  $x_{max}$  par un autre point  $x_{nouv}$  dont le cout est inférieur. Pour se faire, le point situé sur la droite passant par  $x_{max}$  et  $x_c$ , et symétrique à  $x_{max}$  par rapport à  $x_c$  appelé point de réflexion  $x_{ref}$  est d'abord calculé, puis on procède aux tests permettant d'effectuer les opérations suivantes :

- Réflexion

- Expansion
- Contraction

$$x_{ref} = 2x_c - x_{max} \quad (2-35)$$

→ **Si**  $f(x_{min}) > f(x_{ref})$ ,

On procède à l'expansion

Le point  $x_{exp}$  est alors calculé :

$$x_{exp} = 2x_{ref} - x_c \quad (2-36)$$

$x_{nouv}$  est défini comme suit :

$$x_{nouv} = \begin{cases} x_{exp} & \text{si } f(x_{exp}) < f(x_{ref}) \\ x_{ref} & \text{sinon} \end{cases} \quad (2-37)$$

→ **Si**  $\max_{x_i \neq x_{max}} f(x_i) > f(x_{ref}) \geq f(x_{min})$

On effectue la réflexion :

$$x_{nouv} = x_{ref} \quad (2-38)$$

→ **Si**  $f(x_{ref}) \geq \max_{x_i \neq x_{max}} f(x_i)$

On opère à la contraction :

$x_{nouv}$  est calculé comme suit :

$$x_{nouv} = \begin{cases} \frac{1}{2}(x_{max} + x_c) & \text{si } f(x_{max}) \leq f(x_{ref}) \\ \frac{1}{2}(x_{ref} + x_c) & \text{sinon} \end{cases} \quad (2-39)$$

**Algorithme**

```

Choisir  $x_0, x_1, x_2, \dots, x_n$ 
Définir  $x_{min}$  et  $x_{max}$  ;
Choisir  $\varepsilon > 0$  ;
Tant que  $(x_{max} - x_{min} \geq \varepsilon)$  faire
Si  $f(x_{min}) > f(x_{ref})$ 
    Calculer  $x_{exp} = 2x_{ref} - x_c$ 
    Si  $f(x_{exp}) < f(x_{ref})$ 
        Poser  $x_{max} = x_{exp}$ 
    Sinon
        Poser  $x_{max} = x_{ref}$ 
    Fin Si
Si  $\max_{x_i \neq x_{max}} f(x_i) > f(x_{ref})$  et  $f(x_{ref}) \geq f(x_{min})$ 
    Poser  $x_{max} = x_{ref}$ 
Fin Si
Si  $f(x_{ref}) \geq \max_{x_i \neq x_{max}} f(x_i)$ 
    Si  $f(x_{max}) \leq f(x_{ref})$ 
        Poser  $x_{max} = \frac{1}{2}(x_{max} + x_c)$ 
    Sinon
        Poser  $\frac{1}{2}(x_{ref} + x_c)$ 
    Fin Si
Fin Si
    Poser  $k = k + 1$  ;
Fin tant que
Retourner  $x_{min}$  ;

```

**2.3.2. Problèmes avec contraintes**

Un problème est dit problème non linéaire contraint quand le domaine de recherche n'est plus l'ensemble  $\mathbb{R}^n$  tout entier mais un sous ensemble de  $\mathbb{R}^n$ . En effet, le problème est en général posé sous la forme :

Minimiser :  $f(x)$

$$x \in U \subset \mathbb{R}^n \quad (2-40)$$

$f(x)$  non linéaire

$U$  est défini par des collections d'inégalités  $g_i(x) \leq 0$  et d'égalités  $h_i(x) = 0$

$$U = \{x | g_i(x) \leq 0, h_i(x) = 0, \text{ avec } i = 1, \dots, m \text{ et } j = 1, \dots, l\}$$

### 2.3.2.1. Conditions d'optimalité

#### 2.3.2.1.1. Condition nécessaire de Fritz John

Soient :

- \* Un ensemble  $U \subset \mathbb{R}^n$  et  $\neq \emptyset$ , défini par une collection de  $m$  inégalités  $g_i(x) < 0$  et  $r$  égalités  $h_i(x) = 0$  avec  $g_i : \mathbb{R}^n \rightarrow \mathbb{R} \quad i: 1, 2, \dots, m$  et  $h_i : \mathbb{R}^n \rightarrow \mathbb{R} \quad i: 1, 2, \dots, r$  continuellement différentiables  $\forall i$ .
- \*  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  continuellement différentiable et un vecteur  $x^* \in U$
- \*  $I$  l'ensemble des contraintes d'inégalités actives en  $x^*$  c'est-à-dire,  $I = \{i | g_i(x^*) = 0\}$

Si  $x^*$  est un minimum local de  $f$  dans  $U$ , alors il existe des scalaires  $\mu_0, \mu_1, \dots, \mu_m$  et  $\lambda_1, \dots, \lambda_r$  différents de zéro, tels que :

$$\mu_0 \nabla f(x^*) + \sum_{i \in I} \mu_i \nabla g_i(x^*) + \sum_{i=1}^r \lambda_i \nabla h_i(x^*) = 0 \quad (2-41)$$

$$\mu_0, \mu_i \geq 0 \quad \forall i \in I$$

Les scalaires  $\mu_i$  et  $\lambda_i$  sont appelés **multiplicateurs de Lagrange** et la fonction définie ci-dessus est parfois appelée **fonction de Lagrange**.

#### 2.3.2.1.2. Condition nécessaire de Kuhn-Tucker

Dans la condition précédente, on peut remarquer que si  $\mu_0 = 0$ , cette condition ne comporte aucune information concernant la fonction objective, ce qui signifie que la condition de Fritz John ne nous fournit pas d'information relative à notre problème. Ceci implique que  $\mu_0$  doit être strictement supérieur à 0.

Afin de garantir que  $\mu_0 > 0$ , Kuhn-Tucker a évoqué la nouvelle hypothèse suivante sur les contraintes :

Les gradients des contraintes sous forme d'égalités et des contraintes sous forme d'inégalités actives au point considéré soient linéairement indépendants, ce qui signifie qu'il ne peut exister de combinaison linéaire non triviale de ceux-ci dont la somme vaut le vecteur nul.

Avec cette hypothèse et en prenant arbitrairement  $\mu_0 = 1$ , on peut énoncer le théorème de Kuhn-Tucker sur la condition nécessaire d'optimalité :

Soient :

- Un ensemble  $U \subset \mathbb{R}^n$  et  $\neq \emptyset$ , défini par une collection de  $m$  inégalités  $g_i(x) < 0$  et  $r$  égalités  $h_i(x) = 0$  avec  $g_i : \mathbb{R}^n \rightarrow \mathbb{R} \quad i: 1, 2, \dots, m$  et  $h_i : \mathbb{R}^n \rightarrow \mathbb{R} \quad i: 1, 2, \dots, r$  continuellement différentiables  $\forall i$ .
- $f : \mathbb{R}^n \rightarrow \mathbb{R}$  continuellement différentiable et un vecteur  $x^* \in U$
- $I$ , l'ensemble des contraintes d'inégalités actives en  $x^*$  c'est-à-dire,  $I = \{i | g_i(x^*) = 0\}$

Si  $x^*$  est un minimum local de  $f$  dans  $U$ , alors il existe des scalaires  $\mu_1, \dots, \mu_m$  et  $\lambda_1, \dots, \lambda_r$  différents de zéro, tels que :

$$\nabla f(x^*) + \sum_{i \in I} \mu_i \nabla g_i(x^*) + \sum_{i=1}^r \lambda_i \nabla h_i(x^*) = 0 \quad (2-42)$$

$$\mu_i \geq 0 \quad \forall i \in I$$

#### 2.3.2.1.3. Condition suffisante

La condition nécessaire de Kuhn-Tucker suffisante si la fonction objective  $f$  et les contraintes  $g_i$  sont convexes.

Les principales méthodes itératives pour résoudre les problèmes de cette forme sont la classe de méthodes suivantes :

- Les méthodes de directions admissibles
- Les méthodes de pénalités
- Programmation quadratique séquentielle

Par la suite, on va développer un par un ces méthodes.

#### 2.3.2.2. Les méthodes de directions admissibles

C'est une méthode permettant de résoudre un problème de minimisation non linéaire avec contrainte en partant d'un point initial admissible.

### **Principe**

Etant donné un élément  $x_k \in U$ , la méthode consiste à se déplacer dans une direction  $d_k$  vers un autre point  $x_{k+1} \in U$  et de cout inférieur à  $x_k$ . Ce déplacement se repose sur la propriété suivante :

$$\begin{aligned} x_{k+1} &= x_k + \alpha_k d_k \quad x_{k+1} \in U \\ f(x_{k+1}) &< f(x_k) \end{aligned} \tag{2-43}$$

#### a. Détermination de $\alpha_k$

$\alpha_k$  est un scalaire positif

Après avoir déterminé la direction  $d_k$ ,  $\alpha_k$  s'obtient par une minimisation monodimensionnelle de  $x_k + \alpha_k d_k$  pour avoir un déplacement optimal dans  $U$ .

Pour que  $x_{k+1}$  reste dans  $U$ , il est nécessaire d'imposer une limite supérieure à la valeur de  $\alpha_k$ .

#### b. Détermination de $d_k$

On a plusieurs façons de déterminer la direction  $d_k$ . Dans ce manuscrit, on va présenter 2 méthodes permettant facilement de l'obtenir :

- La méthode de Franck et Wolfe
- La méthode de Zoutendijk

#### 2.3.2.2.1. Description de la méthode de Franck et Wolfe

### **Hypothèses**

- Les contraintes sont linéaires
- $U$  est borné

### **Principe**

La manière de générer une direction  $d_k$  selon la méthode de Franck et Wolfe est de minimiser la dérivée directionnelle de  $f$  dans la direction de  $d_k$  (comme dans la méthode du gradient vue précédemment) tout en respectant la condition de descente  $\nabla f(x_k) \cdot d_k < 0$  et de ne pas sortir de l'ensemble des solutions admissibles  $U$ .

On pose  $d_k = \bar{x}_k - x_k$ ,  $\bar{x}_k \in U$

Le sous problème de recherche de  $d_k$  peut être présenté comme suit :

$$\begin{aligned} \text{Minimiser : } & \nabla f(x_k) \cdot (\bar{x} - x_k) \\ & \bar{x} \in U \end{aligned} \tag{2-44}$$

Où  $\bar{x}$  est la solution de ce sous problème

Puisque  $\nabla f(x_k) \cdot (\bar{x} - x_k)$  est linéaire et les contraintes du problème de départ le sont aussi, alors on a face à un problème linéaire et qui peut être résolu par l'algorithme du simplexe.

Une fois  $d_k$  trouvé, l'optimisation monodimensionnelle le long de  $d_k$  devra être faite sous l'intervalle  $\alpha_k \in [0,1]$

Le critère d'arrêt est que la direction générée est égale au vecteur nul, soit  $d_k = 0$

### **Algorithme**

Poser  $k = 0$  ;

Choisir  $x_0$  dans un voisinage de  $x_{min}$  ;

Choisir  $\varepsilon > 0$  ;

Tant que  $(x_{k+1} - x_k \geq \varepsilon)$  faire

Résoudre par simplexe  $\min\{\nabla f(x_k) \cdot (x - x_k)\}$  ;

$d_k = (x - x_k)$  ;

Résoudre par simplexe  $\min\{x_k + \alpha_k d_k\}$  ;

Poser  $x_{k+1} = x_k + \alpha_k d_k$  ;  $x_{k+1} \in U$

Poser  $k = k + 1$  ;

Fin tant que

Retourner  $x_k$  ;

#### 2.3.2.2.2. Description de la méthode de Zoutendijk

Différemment de la méthode de Franck et Wolfe précédent qui ne traite que les problèmes à contraintes linéaires, la méthode de Zoutendijk pourra traiter un problème même si des contraintes non linéaires sont présentes.

##### a. Cas des contraintes linéaires

Même concept que la méthode de Franck et Wolfe, la méthode de Zoutendijk permet aussi de trouver à chaque itération, la direction  $d_k$  en minimisant la dérivée directionnelle de la fonction objective  $f$  au point  $x_k$ .

En effet, les composantes de la direction  $d_k$  seront déterminées dans le sous-programme linéaire et ces dernières sont ensuite normées afin d'empêcher le problème d'être non borné.

Considérant que  $x_k$  est situé à l'intérieur de  $U$ , le sous problème de recherche de direction peut être posé comme suit :

Minimiser :  $\nabla f(x_k) \cdot d_k$

Sous contraintes

$$d_i \leq 1 \quad \forall i \in \{1, \dots, n\}$$

$$d_i \geq -1 \quad \forall i \in \{1, \dots, n\}$$

(2-45)

Au cours de l'itération,  $x_k$  pourra se trouver sur la frontière de  $U$  où une ou plusieurs contraintes du problème original y sont actives. Pour éviter une direction menant vers des points non admissibles, il faut ajouter au programme linéaire précédent des surplus de contraintes. L'exploitation du gradient des contraintes actives est une solution pour ce problème.

Soit  $I$  l'ensemble des contraintes actives en  $x_k$  :  $I = \{i | g_i(x_k) = 0\}$

Le sous problème précédent est finalement présenté comme suit :

Minimiser :  $\nabla f(x_k) \cdot d_k$

Sous contraintes

$$\nabla g_i(x_k) \cdot d \leq 0 \quad \forall i \in I$$

$$-1 \leq d_i \leq 1 \quad \forall i \in \{1, \dots, n\}$$

(2-46)

Les variables de ce problème forment les composantes de la direction  $d_k$  utilisées pour la résolution du problème initial.

#### b. Cas des contraintes non linéaires

En raison de la non linéarité de  $g_i$  au point  $x_k$ , il se peut que tout déplacement dans cette direction conduise en un point non admissible. L'approche précédente doit donc être modifiée pour pouvoir prendre en compte ce type de contraintes.

#### ***Théorème :***

*Considérons le problème de minimisation de  $f(x)$ , sous contraintes  $\nabla g_i \leq 0 \quad i = 1, \dots, m$ . Soit  $x_k$  une solution admissible. Supposons, de plus, que  $f$  et les  $g_i$  sont continues et continuellement différentiables.*

*Si  $\nabla f_i(x_k) \cdot d \leq 0$  et  $\nabla g_i(x_k) \cdot d \leq 0$ , alors  $d$  est une direction de descente admissible de  $f$  en  $x_k$ .*

Pour trouver un vecteur  $d$  satisfaisant ces deux inégalités strictes, une possibilité est de minimiser le maximum de la valeur  $\nabla f(x_k) \cdot d$  et des valeurs  $\nabla g_i(x_k) \cdot d$ . En dénotant ce minimum par  $z$  et en introduisant les restrictions interdisant au problème d'être non borné, nous obtenons le problème suivant :

Minimiser  $z$

Sous les contraintes

$$\nabla f_i(x_k) \cdot d - z \leq 0 \quad \forall i \in I \quad (2-47)$$

$$\nabla h_i(x_k) \cdot d - z \leq 0 \quad \forall i \in I$$

$$-1 \leq d_i \leq 1 \quad \forall i \in \{1, \dots, n\}$$

Si la valeur optimal de  $< 0$  , alors  $d$  est manifestement une direction de descente admissible. Sinon  $x_k$  satisfait la condition de Fritz John ( $x_k$  satisfait la condition de Fritz John si et seulement si la valeur optimale de la fonction objective du programme linéaire précédent est égale à zéro).

Une fois  $x_{k+1}$  défini, le processus est recommencé

Critère d'arrêt : la méthode de Zoutendijk se termine si la valeur optimale de la fonction objective de ce programme linéaire est nulle.

### 2.3.2.3. Méthode de pénalités

Les méthodes de pénalités sont très employées grâce à la simplicité de leurs principes et aussi leurs efficacités.

#### **Principe**

Le concept de base de la méthode est de transformer le problème sous contrainte en une séquence de problème sans contrainte de façon à ajouter une pénalité à la fonction objective quand une contrainte est violée. Après cette transformation, le problème ainsi obtenu pourra être résolu par les méthodes de résolution de problème sans contrainte énoncé précédemment.

Le problème de départ est donc transformé par le suivant :

$$\begin{aligned} \text{Minimiser : } & f(x) + r \cdot p(x) \\ & x \in \mathbb{R}^n \end{aligned} \quad (2-48)$$

$p(x)$  est la fonction de pénalité, dépendant des contraintes  $g_i(x)$

$r$  est un coefficient réel strictement positif appelé coefficient de pénalité

La méthode de pénalité peut être divisée en 2 grandes méthodes :

- La méthode de pénalités intérieures
- La méthode de pénalités extérieures

### 2.3.2.3.1. Les méthodes de pénalité intérieure (ou méthodes de barrière)

Dans cette section, en partant d'un point initial  $x_0$ , le minimum est approché à chaque itération depuis l'intérieur de  $U$ .

#### **Hypothèses**

- La fonction objective  $f$  et les contraintes  $g_i$  sont des fonctions continues de  $\mathbb{R}^n$  dans  $\mathbb{R}$ .
- $U$  n'est défini que par des collections d'inégalités  $g_i(x) \leq 0$
- Le point initial  $x_0$  est situé à l'intérieur de  $U$

#### **Description**

Dans le cas de la pénalité intérieure, la fonction de pénalité  $p(x)$  est définie de sorte que plus la contrainte devient active, c'est-à-dire plus  $x_k$  se rapproche de la frontière de l'ensemble admissible, plus la fonction de pénalisation croît vers l'infini.

Par conséquent, les fonctions de pénalités les plus répandues pour la méthode de pénalité intérieure sont :

- La logarithmique :

$$p(x) = - \sum_{i=1}^m \ln[-g_i(x)] \quad (2-49)$$

- L'inverse :

$$p(x) = - \sum_{i=1}^m \frac{1}{g_i(x)} \quad (2-50)$$

Pour la fonction de pénalité inverse, le problème à résoudre sera donc de la forme :

*Minimiser*

$$f(x) - r_k \sum_{i=1}^m \frac{1}{g_i(x)} \quad (2-51)$$

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

Pour pouvoir se rapprocher davantage des limites de l'admissibilité, les coefficients de pénalité  $r_k$  doivent décroître à chaque itération c'est-à-dire  $0 < r_{k+1} < r_k$  et  $r_k \rightarrow 0$  lorsque  $k \rightarrow \infty$ . Ceci implique que le terme  $f(x) + r \cdot p(x)$  tend vers  $f(x)$  lorsque  $k \rightarrow \infty$ .

Ce coefficient peut être généré par la formule :

$$\begin{aligned} r_{k+1} &= \alpha \cdot r_k \\ 0 < \alpha < 1 \end{aligned} \tag{2-52}$$

### 2.3.2.3.2. Les méthodes de pénalité extérieure

Pour la méthode de pénalité extérieure, le minimum est approché à chaque itération depuis l'extérieure de  $U$ .

#### **Description**

Dans ce cas, afin d'éliminer les positions non admissible, la valeur de la fonction de pénalité  $p(x)$  à ajouter à la fonction objective est 0 si  $x_k$  est dans l'ensemble admissible, elle est supérieure à 0 dans le cas contraire.

On a alors la fonction de pénalité suivante :

$$p(x) = \sum_{i=1}^m [(g_i(x))^2 \cdot w_i(g_i(x))] \tag{2-53}$$

Où

$$w_i(g_i(x)) = \begin{cases} 0 & \text{si } g_i(x) \leq 0 \\ 1 & \text{si } g_i(x) > 0 \end{cases} \tag{2-54}$$

En effet, la fonction  $w_i(g_i(x))$  a pour rôle d'ignorer la pénalité si la contrainte  $g_i(x)$  est respectée.

Cette fonction de pénalité peut être mise sous la forme suivante :

$$p(x) = \sum_{i=1}^m \max^2(0, g_i(x)) \tag{2-55}$$

Finalement, Le problème sans contrainte à résoudre est:

*Minimiser :*

$$\left[ f(x) - r_k \sum_{i=1}^m \max^2(0, g_i(x)) \right] \tag{2-56}$$

Afin d'amplifier ou de diminuer la valeur, les coefficients de pénalité  $\{r_k\}$  doivent être croissant à chaque itération, c'est-à-dire,  $0 < r_k < r_{k+1}$ , avec  $r_k \rightarrow \infty$  quand  $k \rightarrow \infty$ .

La formule suivante est applicable pour avoir ces coefficients :

$$\begin{aligned} r_{k+1} &= \beta \cdot r_k \\ \beta &> 1 \end{aligned}$$

### 2.3.2.4. Programmation quadratique séquentielle (PQS)

La programmation quadratique séquentielle est l'une des méthodes à utiliser pour résoudre des problèmes généraux d'optimisation non linéaire avec contraintes.

#### **Hypothèses**

Toutes les fonctions (c'est-à-dire la fonction objective et les fonctions de contraintes) sont continues et différentiables

Les problèmes à résoudre avec cette méthode est de la forme :

*Minimiser* :  $f(x)$

*Sous les contraintes*

$$h_i(x) = 0 \quad i = 1, \dots, n \quad (2-57)$$

$$g_i(x) \geq 0 \quad i = n + 1, \dots, m$$

$$x_{inf} \leq x \leq x_{sup}$$

#### **Principe**

La méthode, basée sur le principe itérative et la résolution de sous problème de la programmation quadratique, ce sous problème est obtenu en utilisant une approximation quadratique du Lagrangien et par linéarisation des contraintes.

Après approximation, on peut écrire le sous problème sous la forme :

*Minimiser* :

$$\frac{1}{2} d^T B_K d + \nabla f(x_k)^T d$$

*Sous les contraintes*

$$\nabla h_i(x_k)^T d + h_i(x_k) = 0 \quad (2-58)$$

$$\nabla g_j(x_k)^T d + g_j(x_k) \geq 0$$

$$x_{inf} - x_k \leq d \leq x_{sup} - x_k$$

Où  $B_K$  est une approximation de la matrice Hessienne définie positive.

On utilise la recherche linéaire pour trouver le nouveau point  $x_{k+1}$

$$x_{k+1} = x_k + \alpha d_k \quad (2-59)$$

Tel qu'une "fonction modale" aura une valeur de la fonction inférieure au nouveau point. Ici, la fonction de Lagrange est utilisée comme une fonction modale.

Lorsque l'optimalité n'est pas accomplie,  $B_K$  est mis à jour d'après la formule BFGS.

## 2.4. Conclusion

Les méthodes déterministes vues ci-dessus permettent d'obtenir des solutions dont l'optimalité est garantie. Ceci n'est pourtant efficace qu'à des problèmes spécifiques qui sont en général de taille petite puisqu'elles se caractérisent par l'augmentation du temps de calcul de façon exponentielle avec la taille du problème. Sur des problèmes non convexes qui sont parfois de grande taille, il est très difficile de trouver l'optimum global en un temps raisonnable à cause de l'exploration de tout l'espace de recherche. Pour remédier à ces problèmes, il est conseillé de résoudre le problème avec les méthodes modernes qui sont en générales adaptées à des problèmes pareils. Dans le chapitre suivant, on traitera les méthodes d'optimisation modernes.

# Chapitre 3. LES METHODES MODERNES

## 3.1. Introduction

Certains problèmes d'optimisation demeurent hors de portée des méthodes classiques. Un certain nombre de caractéristiques peuvent en effet être problématiques, comme l'absence de convexité stricte, l'existence de discontinuités, une fonction non dérivable, présence de bruit, etc.

Dans de tels cas, le problème d'optimisation est dit difficile, car aucune des méthodes classiques n'est capable de le résoudre exactement en un temps raisonnable, on devra alors faire appel à des autres méthodes permettant une optimisation approchée.

À partir des années 1980, de nouvelles méthodes modernes sont apparues pour la résolution de ces optimisations difficiles.

Il s'agit de méthodes stochastique, contrairement aux méthodes classiques ou déterministes, ne nécessitent ni point de départ, ni à la connaissance du gradient de la fonction objective pour atteindre la solution optimale. Elles s'appuient sur des mécanismes de transitions probabilistes et aléatoires qui explorent efficacement l'espace de recherche et convergent vers l'optimum global. Leur nature aléatoire implique que plusieurs exécutions successives de ces méthodes conduisent à des résultats différents pour une même initialisation du problème d'optimisation. Cependant, elles demandent un nombre important d'évaluations de la fonction objective en comparaison avec les méthodes déterministes exploitant la dérivée de la fonction objective. Dans ce chapitre, on va étudier chaque méthode moderne existante jusqu'à présent.

### Les métaheuristiques [29]

Les métaheuristiques sont des algorithmes stochastiques itératifs qui utilisent alternativement des phases d'exploitation, d'exploration et d'apprentissage (mémorisation) pour rechercher la solution optimale. Elles partent en général d'une solution arbitraire, puis progressent dans la recherche jusqu'à ce qu'un critère d'arrêt spécifié soit atteint. Elles visent à converger vers ce qu'on appelle la *niche* de l'optimum global (i.e., une sous-population similaire dans le voisinage

de la solution optimale), sans prétendre pour autant d'atteindre l'optimum global. Ces algorithmes essaient donc de trouver une approximation de la meilleure solution. La qualité de la solution obtenue résulte donc d'un compromis avec le temps de calcul. Pour améliorer l'efficacité de la recherche, des méthodes déterministes sont souvent utilisées pour générer des solutions de base servant à l'initialisation des algorithmes métaheuristiques. Un des avantages bien connu des métaheuristiques est leur capacité à résoudre les problèmes sans connaissance a priori des formulations mathématiques de ces derniers. En pratique, il suffit de pouvoir associer à une ou plusieurs variables la valeur d'une fonction donnée, comme par exemple le résultat d'une simulation. Cela est bien différent des méthodes déterministes (exemple : méthode du gradient) qui sont essentiellement basées sur le calcul ou l'approximation des dérivées.

Les métaheuristiques peuvent être classées de nombreuses façons.

On peut distinguer :

→ Celles qui travaillent avec une population de solutions, les exemples les plus connus des méthodes qui travaillent avec une population sont :

- Les algorithmes génétiques
- L'optimisation par essaim de particules et
- Les algorithmes de colonies de fourmis présentent

→ Celles qui ne manipulent qu'une seule solution à la fois. Les méthodes qui tentent itérativement d'améliorer une solution sont appelées méthodes de recherche locale ou méthodes de trajectoire. Ces méthodes construisent une trajectoire dans l'espace des solutions en tentant de se diriger vers des solutions optimales. Les exemples les plus connus de ces méthodes sont :

- La recherche Tabou
- Recuit Simulé.

L'origine ainsi que le principe de fonctionnement de ces 5 métaheuristiques seront détaillés dans la suite de ce chapitre.

### 3.2. Recuit simulé [12] [43]

Lors de la simulation du refroidissement de matériaux, Metropolis a publié en 1953 La première version du recuit simulé. Cependant, l'algorithme de résolution de problème d'optimisation que l'on connaît aujourd'hui a été mise au point par trois chercheurs de la société IBM, S. Kirkpatrick et al, C.D. Gelatt et M.P. Vecchi en 1983 grâce aux publications de Metropolis et indépendamment par V. Černý en 1985.

Le recuit simulé est une métaheuristique probabiliste permettant d'approcher l'optimum global d'une fonction objective. Cette technique est souvent utilisée lorsque le calcul de la solution optimale exacte demanderait un temps de calcul trop important, le recuit simulé est alors utilisé pour trouver une solution approchée dans un temps raisonnable. Le nom et le principe de cette métaheuristique sont inspirés directement de la technique du recuit en métallurgie.

### 3.2.1. Le traitement thermique

Les traitements thermiques sont constitués par un certain nombre d'opérations combinées de chauffage et de refroidissement ayant pour but :

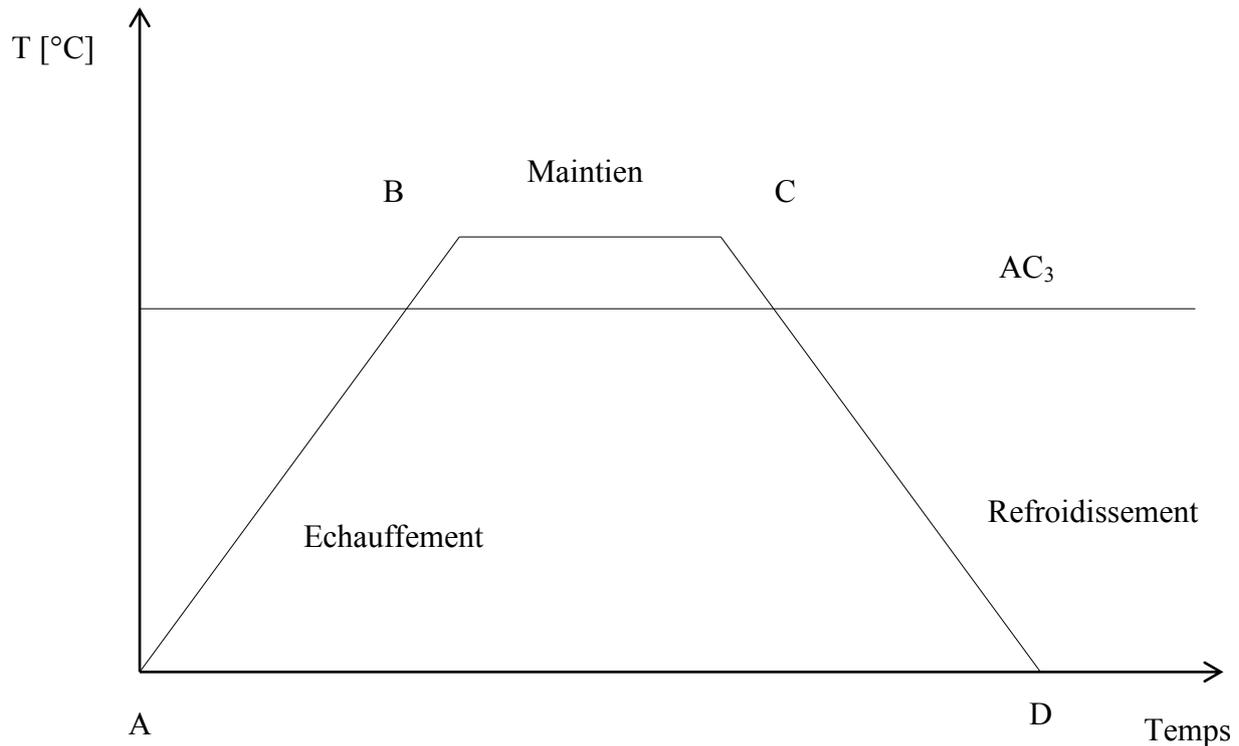
- D'améliorer les caractéristiques des matériaux et rendre ceux-ci plus favorables à un emploi donné, comme : Augmentation de la résistance à la rupture et de la limite élastique en donnant une meilleure tenue de l'élément, augmentation de la dureté, permettant à certaines pièces de mieux résister à l'usure ou aux chocs,.....
- De régénérer un métal qui présente un grain grossier (affiner les grains, homogénéiser la structure) cas des matériaux ayant subi le forgeage.
- De supprimer les tensions internes (écrouissage) des matériaux ayant subi une déformation plastique à froid (emboutissage, fluotournage).

Effectuer un traitement thermique sur une pièce, est alors de faire subir à celle-ci une variation de la température en fonction du temps. Selon la *figure 3-1, page 53*, le procédé de traitement thermique se compose de

AB : L'échauffement à des températures supérieures aux températures de transformation (par exemple : AC3).

BC : Maintient à une température définie.

CD : Refroidissement avec une vitesse donnée : lente (dans le four, à l'air), assez rapide (dans l'huile) ou très rapide (dans l'eau).



*Figure 3-1 : Courbe de la température en fonction du temps d'un traitement thermique*

Le principe du traitement thermique est de modifier la structure et les propriétés mécaniques d'un alliage par des opérations de chauffage jusqu'à une certaine température, de maintien à cette température, et suivies d'un refroidissement à une vitesse plus ou moins accélérée suivant le traitement thermique à procéder dont :

- Le recuit.
- La normalisation.
- La trempe.
- Le revenu et le vieillissement.

Les principaux facteurs qui permettent de distinguer entre les différents types de traitements thermiques sont la température de chauffage, de maintien et la vitesse de refroidissement.

Parmi ces traitements thermiques, c'est le recuit qui nous intéresse dans ce paragraphe,

### Le recuit

Les aciers possèdent un caractère apte d'acquérir grâce à des traitements thermiques variés, toute une gamme de propriétés très différentes. Les recuits en général amèneront les alliages en équilibre physico-chimique et mécanique. Ils tendent à réaliser l'équilibre structural en faisant

disparaître les états hors d'équilibre résultants des traitements thermiques et mécaniques antérieurs.

Le recuit correspond aux valeurs maximales des caractéristiques de ductilité. (Résilience et allongement) et aux valeurs minimales des caractéristiques de résistance (dureté, limite élastique, charge à la rupture). Le recuit a donc pour but de : diminuer la dureté d'un acier trempé, obtenir le maximum d'adoucissement pour faciliter l'usinage ou les traitements mécaniques, régénérer un métal écroui ou surchauffé, homogénéiser les textures hétérogènes et réduire les contraintes internes

Le cycle thermique d'un recuit comprend :

Un chauffage jusqu'à une température dite de recuit qui dépend du type de recuit à réaliser.

Un maintien isotherme à la température de recuit ou des oscillations autour de cette température.

Un refroidissement très lent généralement à l'air calme. La vitesse de refroidissement doit être inférieure à la vitesse critique de recuit,

En effet, lorsqu'on chauffe un métal solide, il devient liquide à une certaine température, dans ce cas les atomes qui le composent ont vu leur degré de liberté augmenter. Inversement lorsque l'on baisse la température le degré de liberté diminue jusqu'à obtenir un solide. Maintenant suivant la façon dont on diminue la température on obtient différent solide :

- Baisse brutale de la température (La trempe), cela produit une structure amorphe, un verre. On a alors un minimum local d'énergie.
- Baisse progressive de la température de façon à atteindre le minimum global d'énergie. On obtient dans ce cas un cristal.

Lorsque la baisse progressive est trop rapide on a alors des défauts au niveau du cristal. Le recuit permet de redonner de la liberté aux atomes pour tenter d'atteindre un nouvel état dynamique. Le recuit permet alors de diminuer les défauts d'un matériau.

Un refroidissement naturel ne permet pas d'atteindre la configuration optimale des atomes pour obtenir la meilleure stabilité.

### 3.2.2. Principe du recuit simulé

La modélisation des processus physiques établi par Métropolis a été exploitée par S. Kirkpatrick et al. afin d'adopter la théorie de l'optimisation. En effet, il a constaté que la recherche des minima locaux était de même nature dans la physique et dans l'informatique.

De manière analogue, pour le recuit simulé, nous parlerons de l'énergie du système, nous cherchons à atteindre la configuration du système qui possède l'énergie la plus faible.

A chaque itération du recuit simulé, la solution actuelle est remplacée par une autre solution issue du voisinage de la solution initiale. Cette nouvelle solution est acceptée avec une probabilité dépendant à la fois de la différence entre les valeurs de coût des deux solutions et d'un paramètre  $T$  appelé température. Cette température décroît progressivement au cours du temps selon une loi déterminée préalablement. Il y a donc toujours une probabilité de dégrader la solution lors d'une itération. Cette stratégie permet d'éviter d'être piégé dans un minimum local. Le recuit simulé est une technique générique qui peut être appliquée aussi bien à des problèmes discrets que des problèmes continus.

L'algorithme, le recuit simulé examine à chaque itération un état voisin  $s'$  de l'état courant  $s$  et détermine de façon stochastique si le système doit exécuter la transition vers cet état ou rester dans la configuration actuelle.

La probabilité de transition depuis un état  $s$  vers un état  $s'$  est donnée par une fonction d'acceptation  $P(e, e', T)$ , cette fonction s'est inspiré de la mécanique statistique en utilisant en particulier la distribution de Boltzmann.

Ainsi, la probabilité  $p$  qu'un système physique passe d'un niveau d'énergie  $E_1$  à un niveau  $E_2$  est donnée par :

$$p = e^{-\frac{\Delta_E}{kT}} \quad (2-1)$$

$K$  : constante de Boltzmann

$T$  : Température absolue, la température du système,  $e$  l'énergie de l'état initial et  $e'$  l'énergie de l'état voisin.

Initialement  $T$  est très élevée donc l'algorithme va souvent dégrader la solution (sauts facilités, phase d'exploration). Au fil des itérations,  $T$  décroît et les dégradations importantes ne sont plus acceptées ( $T$  devient de plus en plus petit).

Généralement, si la nouvelle solution est meilleure, la probabilité de transition est 1, mais ce n'est pas toujours le cas selon les versions de l'algorithme. Si l'énergie de l'état voisin est plus forte, on utilise la règle de Metropolis-Hastings précédente

$\Delta_E = E_2 - E_1$  est la différence d'énergie entre les deux états.

Ce processus amène progressivement le système vers une configuration de plus faible énergie.

Cette étape est répétée jusqu'à ce que le système soit figé sur la même solution pendant plusieurs itérations ou lorsque qu'une limite d'itérations a été dépassée.

Le voisinage d'un état est généré en effectuant des modifications élémentaires de la configuration du système. Pour obtenir de bons résultats il est indispensable que ces modifications soient les

plus simples possibles, les deux configurations doivent rester proches. Par exemple dans le cas du voyageur de commerce, une modification élémentaire est la permutation de deux villes adjacentes. L'exploration du voisinage est l'étape la plus importante du recuit, elle doit se faire de façon optimisée.

### 3.2.3. Algorithme du Recuit Simulé

[Initialisation]

Fournir une solution initiale  $X$

$X_{\text{Min}} = X$

$F_{\text{Min}} = F(X_{\text{Min}})$

Fournir une Température Initiale

[Traitement]

**Tant que** (Température > Température Finale) faire

**Tant que** (Nb de répétition > Nb de répétition Finale) faire

        Choisir  $Y$  dans le voisinage de  $X$  ;

        Calculer  $dF = F(Y) - F(X)$  ;

**Si**  $dF < 0$  alors

$X \leftarrow Y$  ;

**Si**  $F(X) < F(X_{\text{Min}})$

$X_{\text{Min}} \leftarrow X$  ;

$F_{\text{Min}} \leftarrow F(X_{\text{Min}})$  ;

**Fin Si**

**Sinon**

        Tirer  $p$  dans  $[0,1]$  ;

**Si**  $p \leq \exp(-\frac{dF}{T})$

$X \leftarrow Y$  ;

**Fin Si**

**Fin Si**

**Fin Tant que**

$T \leftarrow g(T)$  ;

**Fin tant que**

Retourner  $X_{\text{Min}}$

### 3.3. Recherche tabou

L'algorithme de la Recherche Tabou est une métaheuristique originalement développée par Fred Glover dans les années 80, puis elle est devenue très classique en optimisation combinatoire. Elle est basée sur des idées simples, mais reste quand même efficace pour les résolutions des problèmes d'optimisation.

Elle se distingue des méthodes de recherche locale simple par un certain nombre de règles et de mécanismes lui permettant de surmonter l'obstacle des extremums locaux, tout en évitant les problèmes de cycles.

#### 3.3.1. Le randonneur malchanceux

La meilleure analogie sans doute pour expliquer l'idée derrière l'algorithme de Recherche Tabou serait la fable des randonneurs. Imaginons un randonneur malchanceux perdu dans la montagne. Il voudrait rejoindre le point de plus basse altitude puisqu'il sait qu'une équipe de secours passe régulièrement par ce point. Il ne sait pas quelle est l'altitude de son objectif surtout un brouillard l'empêche de voir loin. Face à un tel problème, la méthodologie de la recherche Tabou lui permettrait d'atteindre son objectif. Partant de n'importe quelle position du randonneur, la Recherche Tabou lui proposera de se déplacer vers le point qui le mènera à la plus basse altitude que le brouillard ne sera plus un empêchement pour lui. Une fois arrivé, il devra chercher de nouveau parmi la prochaine plus basse position qu'il peut apercevoir et qu'il ne l'ait pas encore visitée. Il arrive des cas où la position choisie soit plus haute que la position actuelle du randonneur. Le fait de remonter à cette nouvelle position, peut s'avérer une bonne décision puisque cette action peut mener le randonneur à de nouvelles positions encore plus basses que celle qu'il a trouvé depuis son départ. En poursuivant ainsi de suite, le randonneur se déplacera d'un point à un autre tout en mémorisant les T dernières positions qui lui seront taboues.

Il est vrai que la fable des randonneurs favorise la compréhension de l'algorithme Tabou. Mais dans le cadre d'un travail de recherche, il est impératif de passer à une description formelle et rigoureuse de cet algorithme.

#### 3.3.2. Principe de la Recherche Tabou [26]

Elle est basée aux méthodes de recherche locale simples par le recours à un historique des solutions visitées, qui aide de s'extraire d'un minimum local, mais, pour éviter d'y retomber périodiquement, certaines solutions sont rendues taboues, c'est-à-dire Garder des traces du passé pour mieux s'orienter dans le futur

Dans une première phase, la méthode de recherche Tabou peut être vue comme une généralisation des méthodes d'amélioration locale. En effet, en partant d'une solution quelconque  $x$  appartenant à l'ensemble de solutions, nous nous dirigeons vers une solution  $x'$  appartenant au voisinage  $V(x)$  de  $x$ . L'algorithme explore donc itérativement l'espace de solutions. Afin de choisir le meilleur voisin  $x'$ , l'algorithme évalue la fonction objective  $f$  en chaque point  $V(x)$ , et retient le voisin qui améliore la valeur de  $f$ , ou celui qui la dégrade le moins.

La différence de la méthode de recherche Tabou, par rapport aux autres méthodes locales, réside dans le fait que l'on retient le meilleur voisin, même si celui-ci est plus mauvais que la solution dont il est le voisin direct afin d'échapper au minimum local.

L'algorithme Recherche Tabou, utilise une mémoire pour conserver pendant un moment la trace des dernières meilleures solutions déjà inspectées. Ces solutions sont déclarées taboues, d'où le nom de la méthode. Elles sont stockées dans une liste de longueur  $L$  donnée, appelée liste Tabou. Une nouvelle solution n'est acceptée que si elle n'appartient pas à cette liste Tabou. Ce critère d'acceptation d'une nouvelle solution évite le rebouclage de l'algorithme, durant la visite d'un nombre de solutions au moins égal à la longueur de la liste Tabou, et il dirige l'exploration de la méthode vers des régions du domaine de solutions non encore visitées. La taille de cette liste est bornée à un nombre maximum et s'il ne reste plus de place libre pour insérer le nouvel élément, on élimine la plus ancienne solution trouvée depuis le début de la recherche.

À part le fait que l'algorithme Tabou a besoin de mémoriser la liste taboue pour éviter de cycler, la Recherche Tabou mémorise aussi tout au long de son évolution la meilleure solution  $S_{min}$  rencontrée. En effet, à la fin de la recherche, lorsque le critère d'arrêt est vérifié,  $S_{min}$  nous permettra de retrouver la meilleure solution visitée depuis le lancement de la recherche.

Par ailleurs, il est primordial de bien définir les 3 paramètres suivants à cause de leurs influences sur les résultats de la recherche en qualité et en temps.

a. Le premier est la taille de la liste taboue

Malgré l'apport positif que fournit cette liste, un mauvais paramétrage de sa longueur risque de lui faire perdre toute son utilité. D'un côté, plus la taille de cette liste est petite, plus le phénomène de cyclage reprend vie. D'un autre côté, une liste taboue trop longue limitera énormément l'espace de recherche. Face à ce dilemme, il est généralement conseillé de procéder à une étude empirique dans l'optique de fixer la bonne taille de la liste taboue.

b. Le second élément à définir est la fonction  $V$  qui détermine le voisinage

La détermination de cette fonction a une grande influence sur l'évolution de la recherche. En effet, une mauvaise définition du voisinage pourrait ralentir considérablement l'évolution de l'algorithme. Donc la question à laquelle il faudra répondre pour définir le voisinage est : partant d'une solution  $s$ , comment choisir le sous ensemble des solutions  $x' \in V(x)$ ?

c. La troisième configuration est la condition d'arrêt

Ce dernier joue un rôle essentiel du point de vue de la qualité de la solution et du point de vue de son temps de réponse. On se retrouve ici face à un second dilemme : une condition d'arrêt rapidement atteignable ne laissera pas assez de temps au Tabou pour trouver l'optimum global recherché. À l'inverse, un critère d'arrêt peu probable prolonge trop longtemps la recherche même si l'optimum absolu est atteint.

Pour améliorer l'algorithme afin d'obtenir une solution globale en temps réduits, il existe certaines techniques à appliquer

### 3.3.3. Techniques d'améliorations de l'algorithme de base

Dans cette section, nous présentons les trois principales techniques permettant d'améliorer les performances et l'efficacité de l'algorithme Tabou.

#### 3.3.3.1. Critère d'aspiration

Le critère d'aspiration fût introduit la première fois en 1986 par Fred Glover. En 1989, il publie un autre article sur son Algorithme Tabou où il explique entre autre le fonctionnement du critère d'aspiration. L'idée derrière cette technique amélioratrice est d'accepter certains mouvements récemment effectués et qui en temps normal, ne seraient pas acceptés par les règles du Tabou. Le fait qu'on gère ici des mouvements plutôt que des solutions, nous permet de mémoriser uniquement ces mouvements au lieu de mémoriser des solutions complètes qui alourdissent considérablement la recherche. Cette problématique est principalement rencontrée lors de la vérification de présence d'une solution voisine dans la liste taboue. Mais la mémorisation des mouvements ne présente pas que des avantages. Le fait de mémoriser des mouvements peut amener à visiter la même solution plusieurs fois et cela sans enfreindre l'interdiction qu'impose la liste taboue. Un autre inconvénient au fait de mémoriser des mouvements est qu'il se peut que la liste taboue ne permette pas de visiter des solutions qui n'ont jamais été explorées, même si celles-ci peuvent être des optimums. Face à ce dernier défaut relatif à la mémorisation des mouvements, l'utilisation du critère d'aspiration est d'une grande utilité. En effet, en libérant de

la liste taboue uniquement les mouvements satisfaisant le critère d'aspiration, on peut trouver des solutions meilleures que la meilleure solution rencontrée depuis le début. Donc, le critère d'aspiration peut décider quand un mouvement est bénéfique et donne ainsi la permission à l'algorithme d'entreprendre ce mouvement même s'il est interdit.

### 3.3.3.2. Technique de diversification :

Fred Glover nous explique bien cette technique dans son article. En résumé ce processus consiste à effectuer des mouvements visant à varier l'échantillon de recherche. En effet il s'avère intéressant de diversifier les zones de recherches dans l'espoir de couvrir le plus possible de zones dans l'espace  $N$ . Généralement, cette technique est appelée lorsque la recherche se trouve bloquée aux alentours d'un optimum local. Grâce à la diversification, une solution sera générée, à partir de laquelle l'algorithme reprendra sa recherche.

### 3.3.3.3. Technique d'intensification :

Inversement à la technique de diversification, l'intensification effectue des mouvements permettant d'améliorer rapidement la solution. En effet, en favorisant la recherche aux alentours des zones les plus prometteuses, ce processus peut amener à des solutions de meilleure qualité. Comme son nom l'indique, cette technique intensifie et localise plutôt la recherche tout prêt des solutions avantageuses dans l'espoir de les améliorer et d'atteindre des solutions meilleures que  $S^*$ .

### 3.3.4. Algorithme de la Recherche Tabou

On va garder tous les points du voisinage - ou une partie échantillonnée de ces points si l'ensemble est trop grand (cas d'une fonction continue par exemple) – qu'on appelle  $N(s)$ . On va retirer de ces points ceux qui sont dans la liste tabou  $T(s, k)$  pour obtenir  $N(s, k) = N(s) - T(s, k)$ . On calcule la valeur de la fonction  $f$  à minimiser pour chacun de ces points de  $N(s, k)$ , on trie les points par ordre de  $f$  croissant (le 1er est donc le meilleur du voisinage, mais pas forcément meilleur que le point courant qui n'est évidemment pas inclus dans le voisinage). Si le 1er point est meilleur que la meilleure solution obtenue jusqu'ici, on le garde. Ensuite on met tous les points du voisinage dans la liste tabou. Et on itère jusqu'à par exemple ne plus avoir de changement (ou un autre critère).

Ainsi, à chaque itération, l'algorithme tabou choisit le meilleur voisin non tabou, même si celui-ci dégrade la fonction.

On peut aussi ajouter un critère d'« aspiration », qui déterminera si un point de la liste tabou (ou

plusieurs, donc un sous ensemble  $A(s, k)$  peut quand même être utilisé. qui permet d'utiliser un point tabou car il remplirait quand même une condition désirée (par exemple, si un mouvement interdit par la liste tabou conduit à une valeur de la fonction  $f$  qui serait meilleure que celle obtenue jusqu'ici. Evidemment, cette configuration ne peut pas survenir si on garde des points dans la liste tabou : si un point de la liste tabou a la meilleure valeur de  $f$ , il a forcément été compté auparavant. Des critères d'aspirations plus sophistiqués peuvent être utilisés pour ce critère d'aspiration.

### Algorithme

[Initialisation]

Choisir  $s_{min}$

$f_{min} \leftarrow f(S)$  ;

$\emptyset \leftarrow T$  ;

[Traitement]

**Tant que** (Nb de répétition > Nb de répétition Maximale)

Générer un sous-ensemble  $V(s)$  de solution au voisinage de  $s$

**Pour**  $s \in V(s)$

Choisir  $s' \in V(s)$  qui minimise  $f$  ;

**Fin Pour**

$s_{min} \leftarrow s'$  ;

Mettre à jour  $T$

**Fin tant que**

### 3.4. Optimisation par essais particuliers [31] [33] [44]

L'optimisation par essais de particules est une métaheuristique d'optimisation stochastique, pour les fonctions non-linéaires, elle a été inventée par le socio-psychologue James Kennedy et l'ingénieur en électricité Russel Eberhart en 1995, elle est basée sur la reproduction d'un comportement social et la collaboration entre des individus souvent appelées particules. Cette méthode est alors inspirée du comportement social des animaux évoluant en essaim.

Plusieurs scientifiques ont créé des modèles interprétant le mouvement des vols d'oiseaux et des bancs de poissons. Plus particulièrement, Reynolds et Heppner et al. ont présenté des simulations sur un vol d'oiseaux. Reynolds était intrigué par l'aspect esthétique du déplacement des oiseaux en groupe et Heppner, un zoologue, était intéressé à comprendre les règles permettant à un grand

nombre d'oiseaux de voler en groupe : soit de voler sans se heurter, de changer soudainement de direction, de s'écarter et de se rapprocher de nouveau. Cette étude a grandement inspiré le développement de l'algorithme d'optimisation par essais particuliers.

#### 3.4.1. Les vols d'oiseaux

Lors de simulations des modèles mathématiques décrivant les vols d'oiseaux, Wilson a suggéré que ces types de modèles pourraient très bien s'appliquer à la recherche de points caractéristiques dans un espace de recherche. Sa réflexion se base sur le fait que, lors de l'installation d'une mangeoire à oiseaux dans une cour, même si aucun oiseau ne l'a jamais visitée, après quelques heures de patience un grand nombre d'oiseaux viendront y manger. Lors des simulations de Wilson, la volée d'oiseaux cherchait une mangeoire dans un espace donné et finissait par découvrir son emplacement. En utilisant les algorithmes de modélisation de Heppner et de Reynolds, et en modifiant le modèle mathématique de Wilson, Kennedy et Eberhart ont transformé le tout en un vol d'oiseaux cherchant la « mangeoire » la plus grosse dans un lot de mangeoires contenues dans une région prédéterminée.

L'algorithme d'optimisation par essaim de particules a ainsi vu le jour.

L'algorithme par essaim de particules est généralement introduit en relatant son développement conceptuel.

Tel que mentionné, cet algorithme a vu le jour sous la forme d'une simulation simplifiée d'un milieu social, tel que le déplacement des oiseaux à l'intérieur d'une volée. Pour cet algorithme une redéfinition des termes est nécessaire; une population représente le vol d'oiseaux et un agent ou particule représente chaque oiseau de la volée.

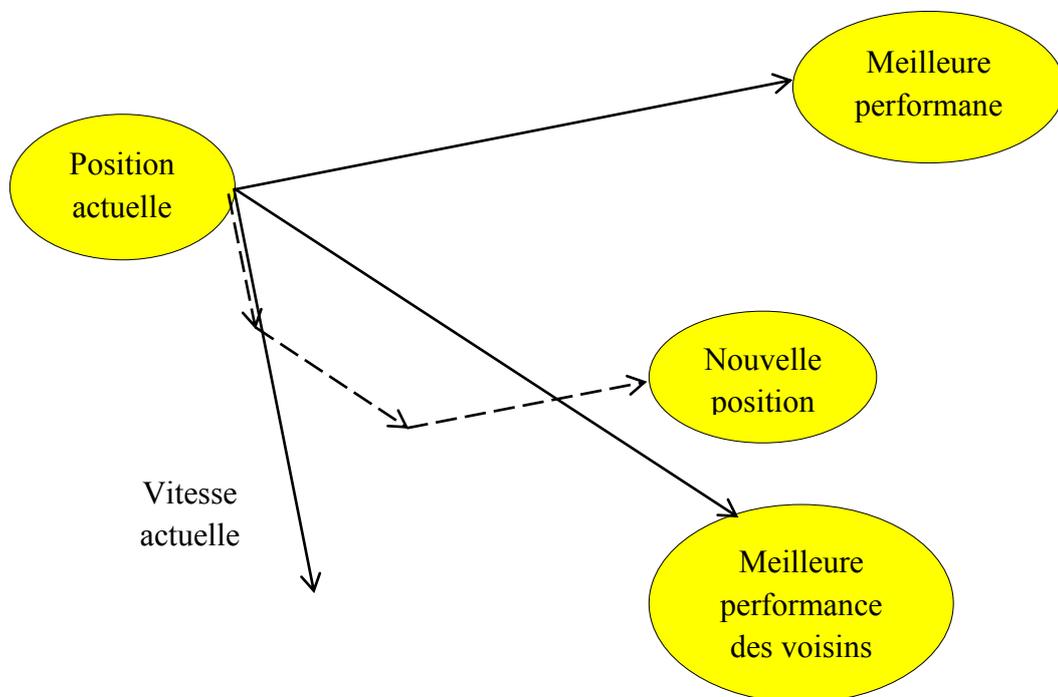
#### 3.4.2. Principe de l'algorithme d'optimisation par essais particuliers

Ce comportement social s'appuie sur l'analyse de l'environnement et du voisinage, ce qui constitue une méthode de recherche d'optimum par l'observation des tendances des particules voisines.

Le concept principal de l'optimisation par essaim de particules est que les solutions potentielles sont accélérées vers les meilleures solutions.

En effet, chaque particule cherche à optimiser ses chances en suivant la tendance de son voisinage qu'il modère par ses propres vécus. Les particules évaluent de manière itérative l'aptitude des solutions candidates et se souviennent de l'endroit où elles ont eu leur meilleure valeur de fitness. Chaque particule dispose d'une mémoire concernant sa meilleure solution visitée ainsi que la capacité de communiquer avec les particules de son entourage. A chaque

itération, les particules se déplacent en tenant compte de leur meilleure position, mais aussi de la meilleure position de ses voisins. L'objectif est de modifier leur trajectoire pour qu'elles se rapprochent le plus possible de l'optimum. Cet optimum est obtenu par une procédure itérative sur la base des processus de mouvement et de l'intelligence dans un système évolutif. L'optimisation par essaim de particules met l'accent sur la coopération plutôt que sur la compétition et il n'y a pas de sélection (au moins dans la version de base), l'idée étant qu'une particule encore aujourd'hui pauvre mérite d'être préservée, peut-être justement parce que c'est elle qui saura trouver la meilleure solution, précisément parce qu'elle agit hors de l'espace prometteur a priori. Selon certains résultats, cette technique ne souffre pas de problèmes rencontrés par d'autres méthodes évolutionnaires.



*Figure 3-2 : Déplacement d'une particule*

La performance de chaque particule (c.-à-d. la convergence de la particule vers l'optimum global) est mesurée en utilisant une fonction de forme physique qui change selon le problème d'optimisation.

Chaque particule dans l'essaim est représentée par les caractéristiques suivantes :

- Sa position actuelle: c'est-à-dire ses coordonnées dans l'ensemble de définition et la valeur de la fonction objective qui lui correspondant.

$x_i$  : La position actuelle de la particule  $i$ .

- Sa meilleure position : c'est la valeur obtenue par la particule et ses coordonnées.

$y_i$  : La meilleure position personnelle de la particule  $i$ .

- Sa vitesse: cette donnée, recalculée à chaque itération de l'algorithme permet de déduire la position suivante de la particule. Elle est fonction de la meilleure position de la particule depuis le début de la recherche, du voisin le mieux positionné à l'instant actuel et de la vitesse précédente de la particule.

$v_i$  : La vitesse courante de la particule  $i$ .

- Ses voisins: c'est un ensemble de particule qui influe sur ses déplacements, en particulier celui qui est le mieux positionné.

$\hat{y}_i$  : La meilleure position de voisinage de la particule  $i$ .

$pbest_i$  : La valeur de fitness de sa meilleure solution

Lors de l'initialisation de l'optimisation par essaim de particules, chaque particule est positionnée aléatoirement dans l'espace de recherche. Pour définir ce positionnement il faut connaître la plage des valeurs de chaque dimension (paramètres) à optimiser.

Puis, la vitesse initiale de chaque particule est aléatoirement choisie. Pour choisir cette vitesse, l'algorithme requiert une vitesse maximale absolue pour chaque dimension. La vitesse peut être négative mais le signe n'indique que le sens dans lequel la particule se déplace.

Il est fortement conseillé d'initialiser la première particule à l'aide d'une valeur connue. Lors de l'optimisation des paramètres dans l'espace de recherche, ceux-ci possèdent généralement des valeurs par défaut. Ces valeurs deviennent le point de départ de la recherche. L'initialisation de la première particule avec de telles valeurs permet d'obtenir, dès la première itération, une valeur de pertinence de base. Cette valeur peut servir de référence pour fins de comparaison et d'estimation de la performance en optimisation obtenue à l'aide de l'optimisation par essaim de particules.

La meilleure position personnelle de la particule  $i$  est la meilleure position (c- à- d, celle ayant pour résultat la meilleure valeur de forme physique) visitée par la particule  $i$  jusqu'ici. La mise à jour de la meilleure position de la particule est comme suit :

$$y_i(t + 1) = \begin{cases} y_i(t) & \text{si } f(x_i(t + 1)) \geq f(y_i(t)) \\ x_i(t + 1) & \text{si } f(x_i(t + 1)) < f(y_i(t)) \end{cases} \quad (2-2)$$

Si Nous nous référons à la version globale de l'optimisation par essaim de particules, où toutes les particules de l'essaim sont considérées comme voisines de la particule  $i$ , on peut adopter la notation  $gbest$  (global best) qui est la meilleur solution.

Pour ce modèle gbest, la meilleure particule est déterminée à partir de l'essaim entier par le choix de la meilleure position personnelle. Si la position de la meilleure particule est notée par le vecteur  $\hat{y}_i$  alors:

$$\hat{y}_i(t)\{y_0, y_1, \dots, y_s\} = \min\{f(y_0(t)), f(y_1(t)), \dots, f(y_s(t))\} \quad (2-3)$$

Où  $s$  indique la taille de l'essaim.

L'étape de la mise à jour de vitesse est indiquée pour chaque dimension  $j \in 1, \dots, N_d$  par conséquent,  $v_{ij}$  représente le  $j^{\text{ème}}$  éléments du vecteur de vitesse de la  $i^{\text{ème}}$  particule.

Ainsi la vitesse de la particule  $i$  est mise à jour en utilisant l'équation suivante :

$$v_{ij}(t + 1) = wv_{ij}(t) + \rho_1(t) (y_{i,j}(t) - x_{i,j}(t)) + \rho_2(t)(\hat{y}_j(t) - x_{i,j}(t)) \quad (2-4)$$

Où :  $w$  est le facteur d'inertie.

$\rho_1$  et  $\rho_2$  sont les constantes d'accélération tiré aléatoirement

La position de la particule  $i$ ,  $x_i$  est mise à jour par l'équation suivante :

$$x_i(t + 1) = x_i(t) + v_i(t + 1) \quad (2-5)$$

C'est le vecteur vitesse qui dirige le processus de recherche et reflète la "sociabilité" des particules.

Si l'on considère  $N$  particules et que chaque particule compare sa nouvelle position à sa meilleure position obtenue, c'est-à-dire qu'on ne prend pas compte du voisinage puisqu'on utilise uniquement l'amélioration obtenue sur la particule elle-même, cela donne l'algorithme suivant.

### 3.4.3. Algorithme de l'optimisation de l'Essaim Particulaire

[Initialisation]

N : nombre de particules

$x_i$  : Position de la particule  $P_i$

$v_i$  : Vitesse de la particule  $P_i$

$pbest_i$  : meilleure fitness obtenue pour la particule  $P_i$

$y_i$  : Position de la particule  $P_i$  pour la meilleure fitness

Initialiser aléatoirement la population

$\rho$  : Valeur aléatoire positive

[Traitement]

**Pour** chaque particule  $i \in 1, \dots, N$  Faire

    Initialiser aléatoirement  $x_i$ ;

    Initialiser aléatoirement  $v_i$  (ou bien  $v_i = 0$ ) ;

$y_i = x_i$  ;

**Fin pour**

[Traitement]

**Tant que** (Nb de répétition > Nb de répétition Maximale)

**Pour**  $i$  de 1 à N faire

**Si** ( $f(x_i) > pbest_i$ ) Alors

$pbest_i \leftarrow f(x_i)$  ;

$y_i \leftarrow x_i$  ;

**Fin Si**

$v_i(t + 1) \leftarrow v_i(t) + \rho(y_i(t) - x_i(t))$  ;

$x_i(t + 1) \leftarrow x_i(t) + v_i(t + 1)$  ;

**Fin Pour**

**Fin Tant que**

### 3.5. Colonie des fourmis [9] [10]

#### 3.5.1. Les fourmis réelles

##### 3.5.1.1. Vie sociale des fourmis

Les fourmis vivent, dans des fourmilières creusées sous le sol ou dans les arbres. La fourmilière est divisée en castes constituées d'individus qui diffèrent par quelques centaines de caractéristiques morphologiques ayant un rôle chacune. Dans la caste des ouvrières, on trouve des femelles stériles avec des caractéristiques physiques adaptées à leur fonction spécifique et assurant diverses tâches comme l'élevage des jeunes larves, la construction et l'entretien du nid et la récolte de la nourriture. Chez les fourmis charpentières, les plus grandes ouvrières sont chargées de la défense du nid, celles de taille intermédiaire, de la recherche de la nourriture et, les plus petites, de s'occuper du couvain.

Mais, en cas de nécessité, n'importe quelle ouvrière peut prendre en charge n'importe quelle autre activité. Chez diverses fourmis, comme les *magnans* d'Afrique, il existe une caste supplémentaire, celle des soldats ayant une grosse tête et d'énormes mandibules pour assurer la défense de la société ou pour attaquer les ennemis. La caste des reproducteurs comprend des femelles ou reines et des mâles. Beaucoup plus grosses et plus grandes que les ouvrières, les reines naissent avec des ailes qu'elles perdent après l'accouplement. Par contre, les mâles sont plus petits et pourvus d'ailes et meurent après l'accouplement. Mâle et reine sont les seuls individus sexués contrairement aux autres individus. La reproduction est assurée par la reine ou les reines faisant partie de la colonie selon un mode de reproduction assez original dans le règne animal appelé *parthénogenèse arrhénotoque*. Ce mode permet à la reine, quand elle est fécondée par un mâle, d'engendrer des femelles, donc des ouvrières, mais également d'engendrer des mâles si elle n'est pas fécondée, mâles qui pourront, par la suite, éventuellement la féconder. Au cours de leur vie, les reines pondent plusieurs millions d'œufs. Ainsi, les fourmis passent, au cours de leur vie, par quatre stades : œuf, larve, nymphe et adulte. Les œufs minuscules, blancs ou jaunâtres, éclosent de deux à six semaines après la ponte et donnent naissance à des larves blanches dépourvues de pattes et immobiles. Au bout d'un temps variant de quelques semaines à plusieurs mois, les larves se transforment en nymphes, souvent enveloppées dans un cocon sécrété à la fin du stade larvaire.

Enfin, les adultes succèdent au stade nymphal. Ces larves sont nourries, nettoyées et protégées par les ouvrières tout au long de leur développement. En échange, ces dernières lèchent une substance que les larves sécrètent à la surface de leur corps.

### 3.5.1.2. Communication entre les fourmis

Les fourmis possèdent de nombreuses glandes de sécrétions jouant plusieurs rôles.

Les glandes de Dufour et de Pavan sont utilisées pour le marquage des chemins et voies d'exploration autour de la fourmilière par des substances odorantes ou pour se rassembler.

Les glandes métapleurales dont le produit est perceptible par les antennes sont utilisées pour la reconnaissance et sécrètent aussi des substances antiseptiques.

Les glandes mandibulaires sécrètent une substance dite «d'alarme» ou «d'alerte » et attractive ou répulsive selon la dose émise et fait fuir les ouvrières étrangères.

Les glandes tergaux ou pygidiales permettant aux fourmis d'entrouvrir leur membrane inter segmentaire du dos de l'abdomen pour libérer une phéromone sexuelle. Chez les fourmis des bois, la glande à poison peut propulser, parfois, à une distance de un mètre de l'acide formique qui sert d'arme de défense chimique.

Dans tous les cas, ces phéromones qui sont des substances volatiles et qui fonctionnent comme des signaux chimiques constituent des messages d'odeur échangés par les fourmis par le biais de leurs antennes pour communiquer entre elles. Ces signaux chimiques sont formés d'un type d'hydrocarbures et véhiculent, à la fois, l'information sur l'espèce, la société, la caste et le stade de développement auxquelles appartiennent les individus rencontrés. C'est une véritable carte d'information génétique.

C'est un système de communication très sophistiqué qui regroupe donc, la communication sexuelle, la communication reine - ouvrière, le recrutement, l'alarme et la défense ainsi que les phéromones territoriales.

Toutefois, la communication par l'intermédiaire des pattes reste aussi très fréquente et de nature souvent utile. Une ouvrière peut amener une autre fourmi à régurgiter de la nourriture liquide en étendant une patte antérieure sur le labium provoquant, de ce fait, un réflexe vomitif dont l'autre fourmi peut profiter pour se nourrir.

En plus de ces deux moyens de communication, les fourmis disposent, aussi, de deux modalités différentes pour échanger les messages sonores : les stridulations et les émissions sonores sans des organes stridulatoires.

Dans le cas des messages sonores dus aux stridulations, le signal sonore est à l'origine d'un crissement aigu, dû aux frottements d'un mince grattoir transversal situé sur sa taille, contre un plateau de fines crêtes parallèles, situé sur la surface adjacente de l'abdomen.

Cette stridulation peut remplir différentes fonctions, selon l'espèce et les circonstances comme, par exemple, émettre des signaux de détresse, renseigner sur la qualité de la nourriture, provoquer un signal de renforcement.

Il existe également, des émissions sonores sans organes stridulatoires. Certaines espèces de fourmis l'emploient lors d'un éventuel danger pour donner l'alerte.

Quant à la communication visuelle, elle était, pour la fourmi, une des formes couramment utilisée au stade primaire de l'évolution, mais elle l'est de moins en moins. Ceci vient du fait de l'évolution considérable des fourmis qui implique une communication plus efficace. Cependant, certaines espèces s'en servent encore pour des situations bien particulières. En effet, si une ouvrière tombe sur de trop grosses graines, elle tourne frénétiquement autour de celles-ci jusqu'à ce que d'autres ouvrières la voient et viennent l'aider.

Ce sont tous ces différents types de communication qui permettent à la fourmi d'être un insecte social ayant la capacité de se partager le travail en accomplissant les tâches de recherche de nourriture, de défense, de l'entretien et de la construction du nid, de l'entretien des larves et leur approvisionnement en nourriture etc. Toutes ces activités doivent être assurées simultanément pour la survie, le développement et le bon fonctionnement de la colonie. Ce bon fonctionnement est assuré par la caste des ouvrières qui se chargent donc de la défense ou de la guerre, de l'entretien de la colonie et ainsi de la construction de la fourmilière, des soins apportés à la reine et aux couvains, de la quête de nourriture.

### 3.5.2. La fourmi artificielle

Ainsi, de la fourmi réelle, les chercheurs ont franchi le pas vers la conception de la fourmi artificielle.

Les algorithmes de colonies de fourmis sont nés à la suite d'une constatation : les insectes sociaux en général, et les fourmis en particulier, résolvent naturellement des problèmes relativement complexes. Les biologistes ont étudié comment les fourmis arrivent à résoudre collectivement des problèmes trop complexes pour un seul individu, notamment les problèmes de choix lors de l'exploitation de sources de nourriture. En effet, de la recherche de nourriture de la fourmi réelle, le chercheur a conçu des algorithmes d'optimisation combinatoire et de routage dans les réseaux. De la division de travail entre fourmis réelles, est déduit le problème d'allocation des tâches. A partir du rangement des larves et des cadavres de la fourmi réelle, c'est le partitionnement de graphes, la classification automatique et le tri collectif qui en ont découlé. Du transport coopératif des fourmis réelles, on est passé à la robotique distribuée notamment l'algorithme de colonie des fourmis.

### 3.5.3. Principe de l'algorithme de colonie des fourmis

Les chercheurs se sont intéressés surtout sur le mode de communication des fourmis qui sont la phéromone. Comme énoncé plus précédemment Les fourmis ont la particularité d'employer pour communiquer des substances volatiles appelées phéromones. Elles sont attirées par ces substances, qu'elles perçoivent grâce à des récepteurs situés dans leurs antennes. Ces substances sont nombreuses et varient selon les espèces. Les fourmis peuvent déposer des phéromones au sol, grâce à une glande située dans leur abdomen, et former ainsi des pistes odorantes, qui pourront être suivies par leurs congénères.

Les fourmis utilisent les pistes de phéromone pour marquer leur trajet, par exemple entre le nid et une source de nourriture. Une colonie est ainsi capable de choisir (sous certaines conditions) le plus court chemin vers une source à exploiter, sans que les individus aient une vision globale du trajet.

En effet, les fourmis le plus rapidement arrivées au nid, après avoir visité la source de nourriture, sont celles qui empruntent les chemins les plus courts. Ainsi, la quantité de phéromone présente sur le plus court trajet est légèrement plus importante que celle présente sur le chemin le plus long. Or, une piste présentant une plus grande concentration en phéromone est plus attirante pour les fourmis, elle a une probabilité plus grande d'être empruntée. La piste courte va alors être plus renforcée que la longue, et à terme, sera choisie par la grande majorité des fourmis.

On constate que le choix s'opère par un mécanisme d'amplification d'une fluctuation initiale. Cependant, il est possible qu'en cas d'une plus grande quantité de phéromone déposée sur les grandes branches, au début de l'expérience, la colonie choisisse le plus long parcours.

D'autres expériences, avec une autre espèce de fourmis, ont montré que si les fourmis sont capables d'effectuer des demi-tours sur la base d'un trop grand écart par rapport à la direction de la source de nourriture, alors la colonie est plus flexible et le risque d'être piégé sur le chemin long est plus faible.

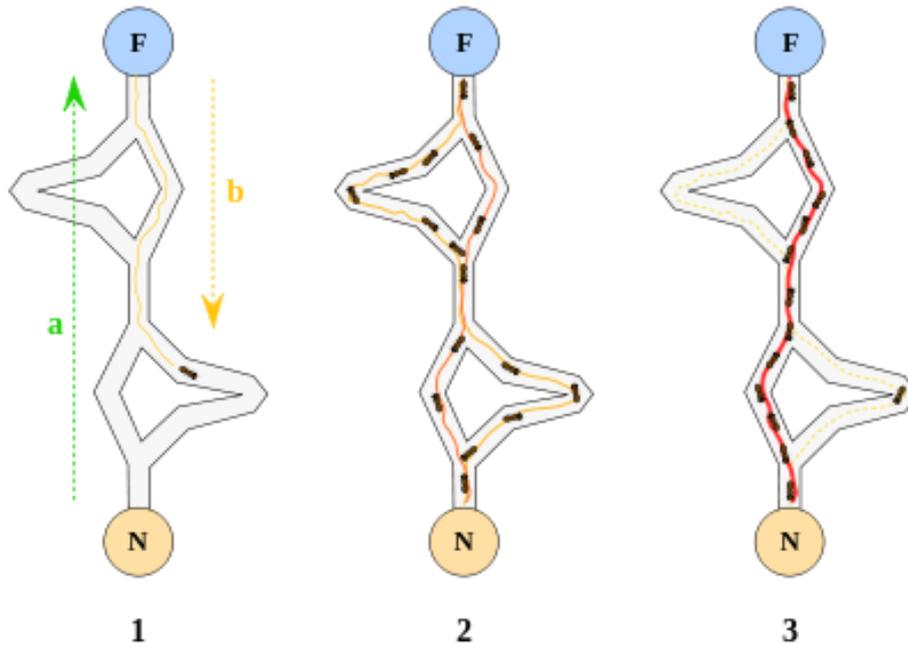


Figure 3-3 : Sélection du chemin plus courte par une colonie de fourmis [46]

Le premier algorithme des colonies de fourmis nommé « Ant System » (AS) a été conçu pour résoudre le problème du voyageur de commerce.

Le problème du voyageur de commerce consiste à trouver une tournée de coût minimal reliant  $n$  villes données. Le voyageur de commerce ne doit passer qu'une seule fois par chaque ville et revenir à son point de départ à la fin. Le coût de la tournée est la longueur totale.

Les différentes étapes de l'algorithme sont les suivantes :

Chaque fourmi  $k$  ( $k = 1, \dots, m$ ) parcourt le graphe et construit un trajet complet de  $n = N$  étapes (on note  $|N|$  le cardinal de l'ensemble  $N$ ). Chaque fourmi  $k$  construit alors une tournée.

Initialisation :

Le nombre d'itération  $t_{max}$  et le nombre de fourmis  $k$  sont fixés.

La première ville  $i$  est sélectionnée selon une loi uniforme : il s'agit de la ville de départ.

Puis la ville suivante  $j$  est sélectionnée selon la probabilité

$$p_{ij}^k(t) = \frac{[\varphi_{ij}(t)]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{l \in J_i^k} [\varphi_{il}(t)]^\alpha \cdot [\eta_{il}]^\beta} \quad \text{Si } j \in J_i^k \quad (2-6)$$

$$p_{ij}^k(t) = 0 \quad \text{Si } j \notin J_i^k \quad (2-7)$$

$\eta_{ij} = \frac{1}{d_{ij}}$  : visibilité, c'est l'inverse de la distance entre les villes, Cette valeur est utilisée pour diriger le choix des fourmis vers des villes proches, et éviter les villes trop lointaines

$\varphi_{ij}$  : Valeur qui représente l'intensité de la phéromone sur la piste

$\alpha$  : paramètre contrôlant l'importance de l'intensité de la piste de phéromone, Si  $\alpha = 0$  seule la visibilité de la ville est prise en compte alors la ville la plus proche est donc choisie à chaque itération.

$\beta$  : paramètre contrôlant l'importance accordée à la visibilité.  $\beta = 0$  seules les pistes de phéromone jouent.

Lorsque la tournée est construite, chaque fourmi dépose une quantité de phéromones sur l'ensemble de son parcours :

$$\Delta_{\varphi_{ij}}^k(t) = \frac{Q}{L^k(t)} \quad \text{Si } (i,j) \in T^k(t) \quad (2-8)$$

$$\Delta_{\varphi_{ij}}^k(t) = 0 \quad \text{Si } (i,j) \notin T^k(t) \quad (2-9)$$

$T^k(t)$  : est le trajet effectué par la fourmi k à l'itération t,

$L^k(t)$  : la longueur de la tournée

Q : un paramètre fixé

Puisque en réalité, les phéromones sont volatiles alors dans l'algorithme, une partie des phéromones sont alors évaporées à chaque itération, de façon à empêcher l'algorithme d'être piégé dans un minimum local. Cette évaporation consiste alors à atténuer l'intensité de phéromone du chemin grâce à un coefficient d'évaporation.

$$\varphi_{ij}(t+1) = (1-\rho) \cdot \varphi_{ij}(t) + \sum_{k=1}^m \Delta_{\varphi_{ij}}^k(t) \quad (2-10)$$

m : nombre de fourmis

$\rho$  : le taux d'évaporation

### 3.5.4. Algorithme

[Initialisation]

$\varphi_{ij} \leftarrow \varphi_0 \quad \forall (i, j) ;$

Placer aléatoirement chaque fourmi sur une ville

[Traitement]

**Pour**  $t=1$  à  $t=t_{\max}$  faire

**Pour** Chaque fourmi  $k$  faire

        Choisir une ville  $j$  selon

        Si  $j \in J_i^k$

$$p_{ij}^k(t) = \frac{[\varphi_{ij}(t)]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{l \in J_i^k} [\varphi_{il}(t)]^\alpha \cdot [\eta_{il}]^\beta};$$

        Sinon

$$p_{ij}^k(t) = 0;$$

        Fin Si

        Déposer une piste  $\Delta_{\varphi_{ij}}^k(t)$  sur le trajet  $T^k(t)$  selon

        Si  $(i, j) \in T^k(t)$

$$\Delta_{\varphi_{ij}}^k(t) = \frac{Q}{L^k(t)} ;$$

        Sinon

$$\Delta_{\varphi_{ij}}^k(t) = 0 ;$$

        Fin Si

**Fin pour**

    Evaporer les pistes selon

$$\varphi_{ij}(t+1) = (1 - \rho) \cdot \varphi_{ij}(t) + \sum_{k=1}^m \Delta_{\varphi_{ij}}^k(t) ;$$

**Fin pour**

### 3.6. Algorithme génétique [39]

Les algorithmes génétiques sont des algorithmes d'optimisation stochastique, ils sont de plus en plus utilisés. On les retrouve désormais dans l'industrie, le monde de l'entreprise ou la recherche. Ils permettent de résoudre un large éventail de problèmes difficiles n'ayant pas de solutions calculables en temps raisonnable de façon analytique ou algorithmique.

Les algorithmes génétiques présentent les avantages suivants : ce sont des méthodes robustes à l'initialisation (c'est-à-dire que leurs convergences ne dépendent pas de la valeur initiale), qui permettent de déterminer l'optimum global d'une fonctionnelle ou de s'en approcher.

Les algorithmes génétiques ont été conçus dans les années 1970 par J. Holland en simulant l'évolution des espèces, la théorie développée par Charles Darwin. De Jongh a ensuite appliqué ces algorithmes à l'optimisation de fonctions à valeurs réelles. Les algorithmes génétiques tentent alors de simuler le processus d'évolution naturelle en s'appuyant sur des techniques dérivées de la génétique et en suivant le modèle darwinien dans un environnement donné.

### 3.6.1. Notion de génétique des populations

La génétique des populations est l'étude de la distribution et des changements de fréquences des versions d'un gène (allèles) dans les populations d'êtres vivants, sous l'influence des pressions évolutives comme :

#### 3.6.1.1. La mutation

C'est le changement de l'information génétique codée par l'ADN, le résultat des mutations qui font apparaître de nouveaux allèles

#### 3.6.1.2. La migration

Ce sont l'occasion de transmission d'allèles d'une population à l'autre, c'est à dire le passage d'individus d'une population à une autre, par exemple sous forme de grains de pollen chez les plantes. Elle peut modifier les fréquences alléliques lorsque celles de la population qui fournit les migrants de celles de la population qui les reçoit.

#### 3.6.1.3. La dérive génétique

Provoque des variations de fréquence des allèles de façon interne aux populations. Ceci est l'effet du hasard (particulièrement efficace quand la population est d'effectif réduit)

Par définition, un gène est l'unité d'hérédité contrôlant la manifestation et la transmission d'un caractère héréditaire

#### 3.6.1.4. La sélection

C'est le principal agent causal de l'évolution. Il s'agit d'un couple formé par un complexe de facteurs du milieu et une population locale d'une espèce. Le complexe de facteur du milieu favorise ou défavorise le bien-être et la reproduction chez les membres de la population.

### ***La sélection naturelle de Darwin***

Alors que plusieurs théories évolutives existaient déjà sous le nom de transformisme, Charles Darwin (1809-1882) propose ce mécanisme que l'on désigne sous le terme de darwinisme ou sélection darwinienne.

La sélection naturelle est l'un des mécanismes qui causent l'évolution des espèces. Ce mécanisme est particulièrement important du fait qu'il explique l'adaptation des espèces aux milieux au fil des générations. La théorie de la sélection naturelle permet d'expliquer et de comprendre comment l'environnement influe sur l'évolution des espèces et des populations en sélectionnant les individus les plus adaptés et constitue donc un aspect fondamental de la théorie de l'évolution.

### ***Principe***

En général, dans une population d'individus d'une même espèce, il existe des différences plus ou moins importantes entre ces individus. A cause de ces différences, certains individus portent des variations qui leur permettent de se reproduire davantage que les autres, dans un environnement précis et puisque lors d'une reproduction, les caractéristiques des individus doivent être héréditaires, c'est-à-dire qu'elles puissent être transmises à leur descendance. De ces mécanismes, les individus les plus adaptés voient leur fréquence s'accroître d'une génération à l'autre, ils auront donc une descendance plus nombreuse tandis que ceux moins adaptés voient leur fréquence diminuer au fil des générations, et en général finissent par disparaître.



*Figure 3-4: Les animaux de même espèce adaptés et non adaptés à l'environnement*

### **3.6.2. Principe de l'algorithme génétique**

Les algorithmes génétiques s'intéressent à un ensemble de solutions simultanément.

Par analogie avec la génétique, un algorithme génétique procède par une multitude d'itérations où chaque itération consiste à tirer au sort deux parents parmi des milliers de solutions (génotypes) plus ou moins bonnes créés au hasard, selon une distribution favorisant les individus les plus adaptés c'est-à-dire les solutions au problème qui sont les plus optimales survivent davantage que celles qui le sont moins. Un opérateur de croisement combine ensuite les deux chromosomes parents pour construire un ou deux enfants, pouvant à leur tour être modifiés

aléatoirement par un opérateur de mutation. Les enfants servent à construire la génération suivante ou remplacent directement des individus de la population. Le processus est répété jusqu'à ce qu'un critère d'arrêt, défini par l'utilisateur, soit vérifié.

Les sections suivantes présentent les caractéristiques essentielles de l'algorithme génétique : les chromosomes, leur évaluation, les méthodes de croisements et les mutations. Ces caractéristiques peuvent différencier chaque algorithme génétique selon l'implémentation de la population initiale, la gestion des générations et les critères d'arrêt.

#### 3.6.2.1. Le codage

Chaque paramètre d'une solution est assimilé à un gène, toutes les valeurs qu'il peut prendre sont les allèles de ce gène, on doit trouver une manière de coder chaque allèle différent de façon unique.

Un chromosome est une suite de gènes, on peut par exemple choisir de regrouper les paramètres similaires dans un même chromosome (chromosome à un seul brin) et chaque gène sera repérable par sa position : son locus sur le chromosome en question.

Chaque individu est représenté par un ensemble de chromosomes, et une population est un ensemble d'individus.

L'efficacité d'un algorithme génétique dépend du choix du codage d'un chromosome, trouver une structure de données et un codage adéquats est l'un des objectifs les plus importants.

Il y a trois principaux types de codage utilisables, et on peut passer de l'un à l'autre relativement facilement :

##### 3.6.2.1.1. Le codage binaire

C'est le plus utilisé. Chaque gène dispose du même alphabet binaire  $\{0, 1\}$ . Un gène est alors représenté par un entier long (32 bits), les chromosomes qui sont des suites de gènes sont représentés par des tableaux de gènes et les individus de notre espace de recherche sont représentés par des tableaux de chromosomes.

##### 3.6.2.1.2. Le codage réel

Cela peut-être utile notamment dans le cas où l'on recherche le maximum d'une fonction réelle.

#### 3.6.2.1.3. Le codage de Gray

Dans le cas d'un codage binaire on utilise souvent la "distance de Hamming" comme mesure de la dissimilarité entre deux éléments de population, cette mesure compte les différences de bits de même rang de ces deux séquences.

#### 3.6.2.2. L'opérateur évaluation et sélection

La phase d'évaluation peut être paralléliser puisque l'évaluation d'un individu ne dépend pas de celle des autres individus, le résultat fourni par la fonction d'évaluation va permettre de sélectionner et de définir quels seront les individus de P qui vont être dupliqués dans la nouvelle population P' et vont servir de parents.

Cette méthode permet de s'assurer que les individus performants seront conservés, alors que les individus peu adaptés seront progressivement éliminés de la population.

Soit n le nombre d'individus de P, on doit en sélectionner n/2 (l'opérateur de croisement nous permet de repasser à n individus).

Cet opérateur est peut-être le plus important puisqu'il permet aux individus d'une population de survivre, de se reproduire ou de mourir. En règle générale, la probabilité de survie d'un individu sera directement reliée à son efficacité relative au sein de la population.

Dans la littérature, il existe de nombreuses méthodes de sélection, complexes ou non, adaptées à certains types de problèmes. Voici quatre types de méthodes de sélection différentes :

- La sélection par roulette
- La méthode "élitiste"
- La sélection par tournois,
- La sélection universelle stochastique.

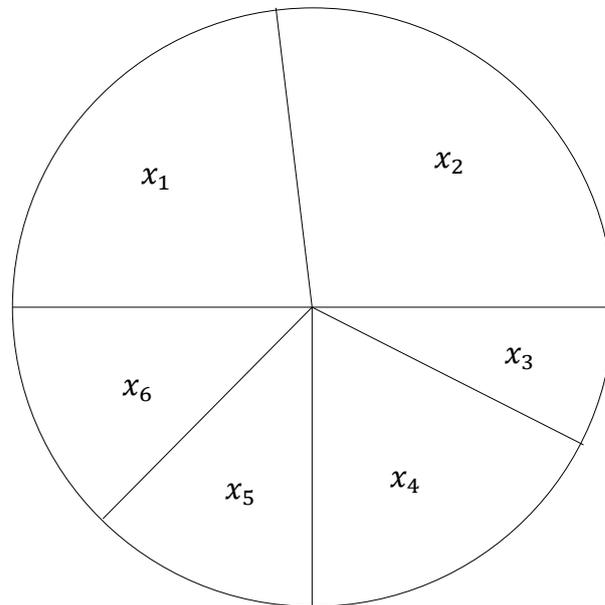
##### 3.6.2.2.1. La sélection par roulette

Cette méthode est la plus courante.

Avec cette méthode chaque individu a une chance d'être sélectionné proportionnelle à sa performance, donc plus les individus sont adaptés au problème, plus ils ont de chances d'être sélectionnés.

On peut comparer cette méthode de sélection à une roulette de casino sur laquelle sont placés tous les individus de la population, l'angle allouée à chacun des individus étant en relation avec leur valeur d'évaluation.

On fait tourner la roue et quand elle cesse de tourner on sélectionne l'individu correspondant au secteur désigné par une sorte de "curseur", curseur qui pointe sur un secteur particulier de celle-ci après qu'elle se soit arrêté de tourner.



*Figure 3-5 : Probabilité de chaque individu placé sur une roulette de casino.*

#### 3.6.2.2.2. La méthode élitiste

Cette méthode consiste d'abord à trier de manière décroissante la population selon l'évaluation. Ensuite, chaque individu se voit associé un rang en fonction de sa position. Ainsi le plus mauvais individu aura le rang 1, le suivant 2, et ainsi de suite jusqu'au meilleur individu qui aura le rang N, pour une population de N individus.

La sélection par rang d'un individu est identique à la sélection par roulette, mais les proportions sont en relation avec le rang plutôt qu'avec la valeur de l'évaluation, de ce fait, les meilleurs individus ne diffèrent pas énormément des plus mauvais, ce qui implique qu'elle conduit à une convergence plus lente vers la bonne solution.

#### 3.6.2.2.3. La sélection par tournois

Cette méthode est celle avec laquelle on obtient les résultats les plus satisfaisants.

Le principe de cette méthode est le suivant : on effectue un tirage avec remise de deux individus de  $P$ , et on les fait "combattre". Celui qui a la valeur d'évaluation la plus élevée l'emporte avec une probabilité  $p$  comprise entre 0.5 et 1. On répète ce processus  $n$  fois de manière à obtenir les  $n$  individus de  $P'$  qui serviront de parents.

La variance de cette méthode est élevée et le fait d'augmenter ou de diminuer la valeur de  $p$  permet respectivement de diminuer ou d'augmenter la pression de la sélection.

#### 3.6.2.2.4. La sélection universelle stochastique

Cette méthode semble être très peu utilisée et qui plus est possède une variance faible, donc introduit peu de diversité, nous n'entrerons donc pas dans les détails, on se contentera d'exposer sa mise en œuvre : On prend l'image d'un segment découpé en autant de sous-segments qu'il y a d'individus. Les individus sélectionnés sont désignés par un ensemble de points équidistants.

#### 3.6.2.3. L'opérateur de croisement

C'est par analogie avec la notion d'hybridation de Darwin qu'a été conçu cet opérateur.

Le croisement utilisé par les algorithmes génétiques est la transposition informatique du mécanisme qui permet, dans la nature, la production de chromosomes qui héritent partiellement des caractéristiques des parents.

Son rôle fondamental est de permettre la recombinaison des informations présentes dans le patrimoine génétique de la population.

Cet opérateur est appliqué après avoir appliqué l'opérateur de sélection sur la population  $P$ ; on se retrouve donc avec une population  $P'$  de  $n/2$  individus et on doit doubler ce nombre pour que notre nouvelle génération soit complète.

On va donc créer de manière aléatoire  $n/4$  couples et on les fait se "reproduire".

Les chromosomes (ensembles de paramètres) des parents sont alors copiés et recombinaison de façon à former deux descendants possédant des caractéristiques issues des deux parents.

L'opérateur de croisement favorise l'exploration de l'espace de recherche. En effet, considérons deux gènes  $A$  et  $B$  pouvant être améliorés par mutation. Il est peu probable que les deux gènes améliorés  $A'$  et  $B'$  apparaissent par mutation dans un même individu. Mais si un parent porte le gène mutant  $A'$  et l'autre le gène mutant  $B'$ , l'opérateur de croisement permettra de combiner rapidement  $A'$  et  $B'$  et donc de créer un nouvel individu possédant cette combinaison, combinaison grâce à laquelle il est possible qu'il soit encore plus adapté que ses parents.

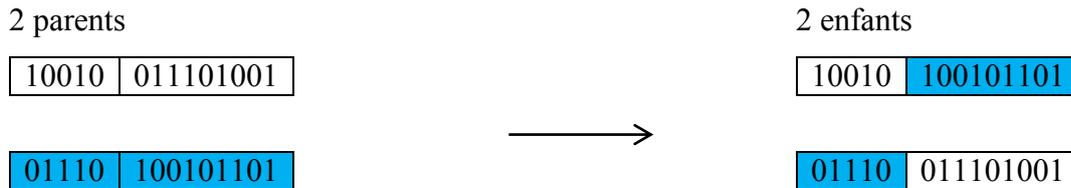
L'opérateur de croisement assure donc le brassage du matériel génétique et l'accumulation des mutations favorables. En termes plus concrets, cet opérateur permet de créer de nouvelles combinaisons des paramètres des composants.

Voici deux méthodes de croisement classiques : le croisement en un point et le croisement en deux points.

### 3.6.2.3.1. Croisement en un point

Considérons des chromosomes constitués de  $M$  bits. Initialement on tire au hasard une position de croisement. On échange ensuite les deux sous chaînes terminales de chacun des deux chromosomes parents P1 et P2, ce qui produit deux nouveaux chromosomes enfants E1 et E2.

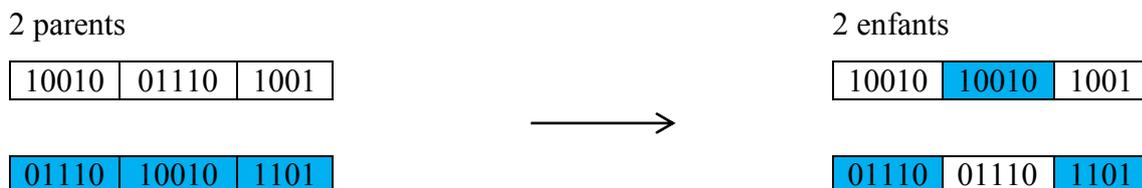
. Les chromosomes des parents sont alors copiés et recombinaison de façon à former deux descendants possédant des caractéristiques issues des deux parents. On forme ainsi la génération  $t + 1$ .



*Figure 3-6: croisement avec un point de crossover*

### 3.6.2.3.2. Croisement en deux points

On choisit aléatoirement deux points de croisement. On échange ensuite les deux sous-chaînes situées entre les deux points de croisement de chacun des deux chromosomes parents P1 et P2, ce qui produit deux nouveaux chromosomes enfants E1 et E2.



*Figure 3-7: croisement avec 2 points de crossover*

### 3.6.2.4. L'opérateur de mutation

Cet opérateur consiste à changer la valeur allélique d'un gène avec une probabilité  $P_m$  très faible, généralement comprise entre 0.01 et 0.001.

On peut aussi prendre  $P_m = \frac{1}{l}$  où  $l$  est la longueur de la chaîne de bits codant notre chromosome.

Une mutation consiste simplement en l'inversion d'un bit (ou de plusieurs bits, mais vu la probabilité de mutation c'est extrêmement rare) se trouvant en un locus bien particulier et lui aussi déterminé de manière aléatoire

L'opérateur de mutation modifie donc de manière complètement aléatoire les caractéristiques d'une solution, ce qui permet d'introduire et de maintenir la diversité au sein de notre population de solutions. Cet opérateur joue le rôle d'un "élément perturbateur", il introduit du "bruit" au sein de la population.

Cet opérateur dispose de 3 grands avantages :

- Il garantit la diversité de la population, ce qui est primordial pour les algorithmes génétiques.
- Il permet d'éviter un phénomène connu sous le nom de dérive génétique.
- Il permet de limiter les risques d'une convergence prématurée causée par exemple par une méthode de sélection élitiste imposant à la population une pression sélective trop forte.

#### 3.6.2.5. L'opérateur de remplacement

Cet opérateur est le plus simple, son travail consiste à réintroduire les descendants obtenus par application successive des opérateurs de sélection, de croisement et de mutation (la population  $P'$ ) dans la population de leurs parents (la population  $P$ ).

Cet opérateur va remplacer une certaine proportion de ceux-ci, proportion pouvant bien sûr être choisie. Le rapport entre le nombre d'individus nouveaux allant être introduits dans la population  $P$  et le nombre d'individus de cette population est connu sous le nom de génération gap.

On trouve essentiellement 2 méthodes de remplacement différentes :

##### 3.6.2.5.1. Le remplacement stationnaire

Dans ce cas, les enfants remplacent automatiquement les parents sans tenir compte de leurs performances respectives, et le nombre d'individus de la population ne varie pas tout au long du cycle d'évolution simulé, ce qui implique donc d'initialiser la population initiale avec un nombre suffisant d'individus. Ce remplacement pourra être fait de 2 façons différentes : en remplaçant la totalité de la population  $P$  par la population  $P'$  ou en remplaçant une certaine proportion d'individus de  $P'$  qui remplaceront leurs parents dans  $P$ .

##### 3.6.2.5.2. Le remplacement élitiste

Dans ce cas, on garde au moins l'individu possédant les meilleures performances d'une génération à la suivante. En général, on peut partir du principe qu'un nouvel individu (enfant) prend place au sein de la population que s'il remplit le critère d'être plus performant que le moins performant des individus de la population précédente. Donc les enfants d'une génération ne remplaceront pas nécessairement leurs parents comme dans le remplacement stationnaire et par la même taille de la population n'est pas figée au cours du temps. Ce type de stratégie

améliore les performances des algorithmes évolutionnaires dans certains cas. Mais présente aussi un désavantage en augmentant le taux de convergence prématurée.

### 3.6.3. Algorithme

[Initialisation]

$$P = \begin{cases} I_1 = (x_1^1, x_2^1, \dots, x_m^1) \\ I_2 = (x_1^2, x_2^2, \dots, x_m^2) \\ \vdots \\ I_n = (x_1^n, x_2^n, \dots, x_m^n) \end{cases}$$

[Traitement]

Codage

**Tant que**  $k < k_{max}$  (Nb de répétition < Nb de répétition Maximale)

**Pour**  $i = 1$  à  $n$

$$y_i = F(I_i) \text{ //probabilité de sélection}$$

$$p_i = \frac{y_i}{\sum_{i=1}^n y_i}$$

**Fin pour**

$$P_{k+1}^1 = \alpha P_k^1 + (1 - \alpha) P_k^2 \text{ //Croisement barycentrique}$$

$$P_{k+1}^2 = (1 - \alpha) P_k^1 + \alpha P_k^2 \text{ //Croisement barycentrique}$$

Mutation

Remplacement

$$k = k + 1$$

**Fin Tant que**

### 3.7. Conclusion

Dans ce chapitre on a pu identifier que parmi les méthodes d'optimisation modernes qui sont des méthodes stochastiques, il existe deux grandes classes, celles qui travaillent avec une population de solution comme les algorithmes génétiques, l'optimisation par essaim de particules, les algorithmes de colonies de fourmis et celles qui ne manipulent qu'une seule solution comme la recherche tabou et recuit simulé. Avec le caractère stochastique des méthodes modernes, on peut explorer l'espace de recherche efficacement afin de guider la recherche vers la solution optimale puisque on a vu que les métaheuristiques peuvent contenir des mécanismes qui permettent d'éviter d'être bloqué dans des régions de l'espace de recherche.

# Chapitre 4. APPLICATION :

## OPTIMISATION DE LA PUISSANCE ACTIVE DANS LE RESEAU ELECTRIQUE

### 4.1. Introduction

L'acheminement de l'énergie électrique génère des pertes importantes en puissance active, ces pertes doivent être minimisées afin d'avoir un réseau performant. Minimiser les pertes totales en puissances actives dans un réseau est un enchaînement de plusieurs disciplines, en effet pour y arriver à cet objectif, on va d'abord donner un aperçu général d'un réseau électrique et c'est ensuite qu'on calcule la répartition de puissance en fonction des données du réseau et c'est à partir de ce calcul qu'on puisse dégager la fonction permettant de définir les pertes en puissance active. Nous terminerons ce chapitre par une discussion des résultats de la comparaison des pertes initiales avec les pertes après optimisation.

### 4.2. Généralités sur le réseau d'énergie électrique [15]

Les premiers réseaux électriques sont apparus dans la première moitié du XXème siècle. Leurs développements furent d'abord anarchiques, chaque gestionnaire de réseaux développant ses moyens de distribution.

Un réseau électrique est l'ensemble de tous les dispositifs permettant la production et l'acheminement de l'énergie électrique de la source jusqu'aux consommateurs d'électricité.

## 4.2.1. Les éléments constitutifs d'un réseau

### 4.2.1.1. La centrale électrique

Habituellement, on appelle centrale électrique l'usine de production d'électricité. Les centrales électriques ont pour objectif de transformer les différentes sources d'énergie naturelle qui existent en énergie électrique. Les centrales électriques regroupent l'ensemble des éléments des unités de production comme les alternateurs, les moteurs, les turbines,...

On peut distinguer les centrales qui exploitent l'énergie fossile et nucléaire et ceux qui exploitent l'énergie renouvelable.

### 4.2.1.2. Les postes électriques

Appelé aussi nœud du réseau, c'est le point où la connexion des lignes électrique sont faites.

Dans ces postes électriques, on trouve :

- L'interconnexion entre les lignes de même niveau de tension :

Cela permet d'interconnecter l'énergie électrique issue des différentes centrales électriques et de répartir l'énergie sur les différentes lignes reliées avec ce poste.

- La transformation de l'énergie : les transformateurs permettent de passer d'un niveau de tension à un autre :

Le transport d'électricité entraîne des pertes dues à l'effet Joule au niveau des conducteurs électriques, ces pertes dépendent de l'intensité  $I$ , de la tension  $U$  et de la résistance  $R$  de la ligne.

Pour le transport de tension triphasée on a :

$$P_{\text{pertes Joule}} = R \cdot I^2 = R \cdot \frac{P_{\text{elec}}^2}{3U^2} \quad (4-1)$$

Pour une même puissance électrique transmise par la ligne et à résistance égale, les pertes par effet Joule diminuent donc comme le carré de la tension: elles sont divisées par quatre quand la tension double. Ainsi, il convient d'élever la tension lors du transport de l'énergie électrique et de le rabaisser à la destination au moyen des transformateurs éleveurs et abaisseur.

### 4.2.1.3. Les moyens de transport d'électricité

Comme l'électricité ne se stocke pas en grande quantité, la production doit être adaptée à la consommation. C'est pourquoi l'énergie produite doit être acheminée en temps réel jusqu'aux consommateurs. L'acheminement de l'énergie sur les longues distances est assuré par les lignes électriques. Elles sont constituées de 3 phases, et chaque phase peut être constituée d'un faisceau

de plusieurs conducteurs (de 1 à 4) espacés de quelques centimètres afin de limiter l'effet couronne qui entraîne des pertes en lignes, différentes des pertes Joule.

#### 4.2.2. Les fonctions du réseau

Du point de vue physique, le réseau électrique est caractérisé par le niveau de tension:

Le réseau de transport

Le réseau de répartition

Le réseau de distribution

Ces niveaux tensions sont illustrés par le tableau suivant :

Classification de la tension	Intervalle de tension
Basse Tension (BT)	$110[V] \leq U_n < 1[kV]$
Moyenne Tension (MT)	$1[kV] \leq U_n < 33[kV]$
Haute Tension (HT)	$U_n \geq 33[kV]$

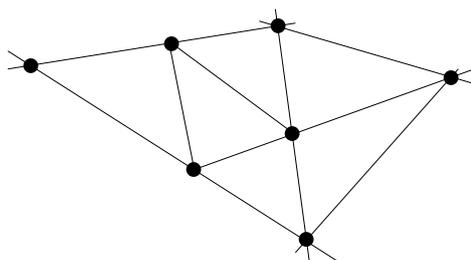
*Tableau 4-1: Classification des réseaux en fonction du niveau de tension*

#### 4.2.3. Structure des réseaux électriques

Un réseau électrique peut être structuré de manières suivantes :

##### 4.2.3.1. Structure maillée :

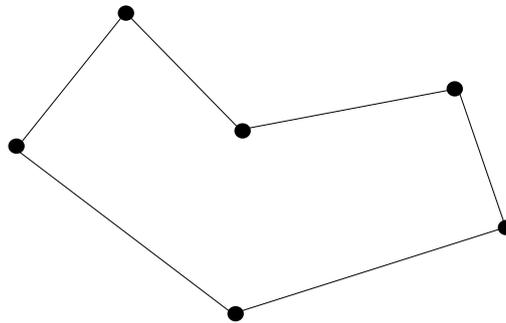
Cette structure apporte une grande sécurité sur l'approvisionnement, en effet, même en cas de défaillance d'un poste, l'alimentation n'est pas interrompue puisque les postes électriques sont reliés entre eux par de nombreuses lignes électriques.



*Figure 4-1: réseau à structure maillée*

#### 4.2.3.2. Structure radiale ou bouclée:

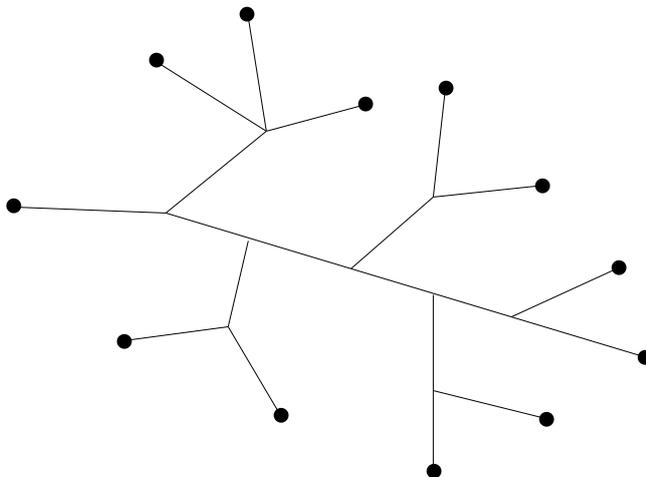
Cette structure offre la sécurité d'alimentation élevée mais reste quand même inférieure à celle de la structure maillée.



*Figure 4-2 : réseau à structure bouclée*

#### 4.2.3.3. Structure arborescente:

Le niveau de sécurité d'alimentation apporté par cette structure est faible puisqu'un défaut sur la ligne ou sur le poste principal engendre une coupure de l'ensemble des clients en aval.



*Figure 4-3 : réseau à structure arborescente*

Chaque type de structure possède des spécificités et des modes d'exploitation très différents. Les grands réseaux d'énergie utilisent tous ces types de structure. Dans les niveaux de tension les plus élevés, on utilise la structure maillée : c'est le réseau de transport. Dans les niveaux de tension inférieurs, la structure bouclée est utilisée en parallèle de la structure maillée : c'est le réseau de répartition. Enfin, pour les plus bas niveaux de tension, la structure arborescente est quasiment exclusivement utilisée : c'est le réseau de distribution.

### 4.3. L'écoulement de puissance [45]

Dans un réseau électrique, Le problème de l'écoulement de puissance consiste à calculer :

- Les tensions complexes au niveau de tous les jeux de barres des différents types de nœuds.
- Les puissances actives et réactives transmises d'un nœud à un autre.
- Les puissances totales injectées en chaque nœud.
- Les pertes totales active et réactive dans le réseau.

Avant d'entamer le calcul de la répartition de puissance, il est primordial de classer tous les nœuds du réseau par leurs fonctions.

#### 4.3.1. Classification des nœuds

Les puissances apparentes injectées  $S$  dépendent des différents types de nœuds du réseau. Ces derniers se subdivisent comme suit :

- **Les nœuds de production** : Ce sont des nœuds reliés aux centrales de production et dont la consommation est relativement négligeable et pour lesquelles la puissance active  $P$  et le module de la tension  $E$  sont connus.
- **Les nœuds de consommation** : Ce sont des nœuds reliés seulement aux charges qui sont caractérisées par les puissances actives  $P$  et réactive  $Q$  qui sont connues.
- **Les nœuds mixtes** : Ce sont des nœuds reliés aux générateurs de puissances active et réactive et aux charges.

#### **Nœud Balancier ou nœud de référence :**

Le bilan de puissance active est :

$$\sum P_G = \sum P_C + \text{pertes actives du réseau} \quad (4-2)$$

Puisqu'on ne connaît pas encore les pertes de puissance active dans toutes les lignes, on ne peut pas encore imposer la puissance active dans tous les nœuds, mais on peut quand même imposer tous les nœuds sauf un nœud de production réglable pour pouvoir compenser les pertes actives dans les lignes. Ce nœud de production réglable est nommé nœud balancier

Le choix du nœud de référence ou du nœud balancier se fera parmi les nœuds production selon le critère ci-dessous :

Il n'est pas arbitraire mais se base sur le nœud qui est relié à la centrale la plus puissante et équipée de moyens de régulation très fiables.

La tension du nœud de référence est fixée et maintenue constante durant le calcul avec une tolérance de 0 à 8% ou 1 à 1,08 en valeur réduite. Les puissances active et réactive de ce nœud seront déterminées après le calcul des tensions complexes des autres nœuds.

Les types de nœuds peuvent être résumés dans le tableau suivant :

Type de nœud	Variable données	Variable inconnues
Nœud de production	P et V	Q et $\delta$
Nœud de consommation	P et Q	V et $\delta$
Nœud balancier	V et $\delta$	P et Q
Nœud mixte	P et Q	Q et $\delta$

*Tableau 4-2 : Les variables données et inconnues des nœuds*

#### 4.3.2. Equations du réseau électrique en régime permanent

Pour la résolution d'un problème de répartition de charges, il est plus commode de travailler avec les admittances plutôt qu'avec les impédances. Nous commencerons par un bref rappel des formules relatives à l'application de la méthode dite « de la matrice d'admittance » pour le calcul d'un réseau électrique quelconque.

Lorsqu'un réseau électrique se trouve en régime permanent établi, il peut être modélisé par un système d'équations linéaires ayant la forme :

$$\bar{I} = \bar{Y} \cdot \bar{U} \quad (4-3)$$

Où

$\bar{I}$  : Vecteur complexe des courants injectés au nœud.

$\bar{Y}$  : Matrice admittance complexe.

$\bar{U}$  : Vecteur complexe des tensions en chaque nœud.

La valeur des composantes de la matrice d'admittance est établie par inspection de la manière suivante :

L'admittance propre « $Y_{ii}$ », associée au nœud 'i', est égale à la somme des admittances des branches incidentes à ce nœud.

$$Y_{ii} = |Y_{ii}| \angle \theta_{ii} = |Y_{ii}| (\cos \theta_{ii} + j \sin \theta_{ii}) = G_{ii} + B_{ii} \quad (4-4)$$

L'admittance de transfert ou admittance mutuelle « $Y_{ki}$ », associée aux nœuds 'k' et 'i', est égale à l'admittance de la branche qui joint ces deux nœuds, changée de signe.

$$Y_{ij} = |Y_{ij}| \angle \theta_{ij} = |Y_{ij}| (\cos \theta_{ij} + j \sin \theta_{ij}) = G_{ij} + B_{ij} \quad (4-5)$$

Si le réseau admet  $n$  nœuds,  $\bar{I}$  et  $\bar{U}$  auront  $N$  composantes complexes chacun et  $\bar{Y}$  sera une matrice complexe de dimensions  $n \times n$ , dont les éléments  $Y_{ij}$  seront calculés à partir des caractéristiques des composants du réseau.

En général pour simplifier le calcul, il est préférable d'utiliser la méthode faisant intervenir les puissances actives et réactives injectées au niveau de chaque nœud telle que :

$$\bar{S}_i = P_i + jQ_i = \bar{U}_i * \bar{I}_i^* \quad (4-6)$$

$$\bar{S}_i^* = P_i - jQ_i = \bar{U}_i^* * \bar{I}_i \quad (4-7)$$

Où

$P_i$  : La puissance active injectée au nœud i.

$Q_i$  : La puissance réactive injectée au nœud i.

$\bar{U}_i^*$  : Le complexe conjugué du vecteur tension  $\bar{U}_i$ .

Pour la formulation de la puissance active et réactive, nous devons définir les quantités suivantes. Pour le bus i on a :

$$V_i = |V_i| \angle \delta_i = |V_i| (\cos \delta_i + j \sin \delta_i) \quad (4-8)$$

Si le système compte  $n$  bus, le courant injecté au bus i est donné par :

$$I_i = Y_{i1}V_1 + Y_{i2}V_2 + \dots + Y_{in}V_n$$

$$I_i = \sum_{k=1}^n Y_{ik}V_k \quad (4-9)$$

Nous notons que par convention, le courant entrant dans un bus est de signe positif, et celui qui en sort est de signe négatif. Ce qui implique que la puissance active et réactive entrant dans un bus sera à son tour de signe positif

Compte tenu de la relation précédente, la puissance apparente est alors:

$$\begin{aligned}\bar{S}_i^* &= P_i - jQ_i = U_i^* \cdot I_i = U_i^* \sum_{k=1}^n Y_{ik} \cdot U_k \\ &= U_i^* (Y_{i1} \cdot U_1 + Y_{i2} \cdot U_2 + \dots + Y_{in} \cdot U_n)\end{aligned}\quad (4-10)$$

$$\begin{aligned}P_i - jQ_i &= \sum_{k=1}^n |Y_{ik} \cdot V_i \cdot V_k| (\cos \delta_i - j \sin \delta_i) (\cos \theta_{ik} + j \sin \theta_{ik}) (\cos \delta_k \\ &\quad + j \sin \delta_k)\end{aligned}\quad (4-11)$$

On sait que :

$$\begin{aligned}(\cos \delta_i - j \sin \delta_i) (\cos \theta_{ik} + j \sin \theta_{ik}) (\cos \delta_k + j \sin \delta_k) &= (\cos \delta_i - \\ j \sin \delta_i) [\cos(\theta_{ik} + \delta_k) + j \sin(\theta_{ik} + \delta_k)] &= \cos(\theta_{ik} + \delta_k - \delta_i) + \\ j \sin(\theta_{ik} + \delta_k - \delta_i)\end{aligned}\quad (4-12)$$

Et nous pouvons exprimer les composantes réelles et imaginaires de la puissance injectée en chaque nœud de la manière suivante :

$$P_i = \sum_{k=1}^n |Y_{ik} \cdot V_i \cdot V_k| \cdot \cos(\theta_{ik} + \delta_k - \delta_i)\quad (4-13)$$

$$Q_i = - \sum_{k=1}^n |Y_{ik} \cdot V_i \cdot V_k| \cdot \sin(\theta_{ik} + \delta_k - \delta_i)\quad (4-14)$$

### 4.3.3. Résolution des équations

A ce stade, il existe plusieurs façons de résoudre le système.

En exprimant les équations relatives aux  $P_i$  et  $Q_i$  connus ( $P_i$  pour les nœuds PV des générateurs ;  $P_i$  et  $Q_i$  pour les nœuds PQ des charges et aucune pour le nœud PV), nous obtenons un système d'équation dont la résolution est généralement plus complexe au fur et à mesure que le nombre de nœuds croît. La résolution manuelle d'un tel problème n'est envisageable que pour un nombre de nœuds très réduit. Les systèmes plus complexes nécessiteront un soutien numérique à la résolution.

Pour la résolution de ces équations, on a à notre disposition 2 méthodes différentes : la méthode de Newton-Raphson et celle de Gauss – Seidel.

La méthode de Newton-Raphson exploite les équations (4-13) et (4-14) tandis que Gauss Seidel est basée sur l'équation (4-15) qui est une modification de l'équation (4-10).

Sachant la puissance active et réactive dans un bus, on peut écrire:

$$U_i = \frac{1}{Y_{ii}} \left[ \frac{P_{i,inj} - jQ_{i,inj}}{U_i^*} - Y_{i1} \cdot U_1 - Y_{i2} \cdot U_2 - \dots - Y_{in} \cdot U_n \right] \quad (4-15)$$

Les deux méthodes utilisent des estimations des variables inconnues comme valeurs initiales pour les itérations et ces valeurs seront corrigés progressivement au cours de l'itération jusqu'à la solution exacte

### Méthode de Gauss – Seidel

Soit le système de  $n$  équation à  $n$  inconnues suivantes:

$$\begin{aligned} x_1 &= f_1(x_1, x_2, \dots, x_n) \\ x_2 &= f_2(x_1, x_2, \dots, x_n) \\ x_i &= f_i(x_1, \dots, x_i, \dots, x_n) \\ &\vdots \\ x_n &= f_n(x_1, x_2, \dots, x_n) \end{aligned} \quad (4-16)$$

On commence par donner des valeurs approximatives au variable, c'est-à-dire  $x_1^0, x_2^0, \dots, x_n^0$ :

Itération 1 :

Pour  $x_1$

$$x_1^1 = f_1(x_1^0, x_2^0, \dots, x_n^0) \quad (4-17)$$

Pour  $x_2$

$$x_2^1 = f_2(x_1^1, x_2^0, x_3^0, \dots, x_n^0) \quad (4-18)$$

Pour  $x_3$

$$x_3^1 = f_3(x_1^1, x_2^1, x_3^0, \dots, x_n^0) \quad (4-19)$$

Pour  $x_i$

$$x_i^1 = f_i(x_1^1, x_2^1, \dots, x_{i-1}^1, x_i^0, x_{i+1}^0, \dots, x_n^0) \quad (4-20)$$

Itération 2 :

Pour  $x_1$

$$x_1^2 = f_1(x_1^1, x_2^1, \dots, x_n^1) \quad (4-21)$$

Pour  $x_2$

$$x_2^2 = f_2(x_1^2, x_2^1, x_3^1, \dots, x_n^1) \quad (4-22)$$

Pour  $x_3$

$$x_3^2 = f_3(x_1^2, x_2^2, x_3^1, \dots, x_n^1) \quad (4-23)$$

Pour  $x_i$

$$x_i^2 = f_i(x_1^2, x_2^2, \dots, x_{i-1}^2, x_i^1, x_{i+1}^1, \dots, x_n^1) \quad (4-24)$$

Itération k :

Pour  $x_1$

$$x_1^k = f_1(x_1^{k-1}, x_2^{k-1}, \dots, x_n^{k-1}) \quad (4-25)$$

Pour  $x_2$

$$x_2^k = f_2(x_1^k, x_2^{k-1}, x_3^{k-1}, \dots, x_n^{k-1}) \quad (4-26)$$

Pour  $x_3$

$$x_3^k = f_3(x_1^k, x_2^k, x_3^{k-1}, \dots, x_n^{k-1}) \quad (4-27)$$

Pour  $x_i$

$$x_i^k = f_i(x_1^k, x_2^k, \dots, x_{i-1}^k, x_i^{k-1}, x_{i+1}^{k-1}, \dots, x_n^{k-1}) \quad (4-28)$$

On note :

$$\Delta x_i^k = x_i^k - x_i^{k-1} \quad (4-29)$$

### **Test d'arrêt**

Le calcul itératif persiste tant que le critère d'arrêt n'est pas satisfait. Ce dernier est défini la comparaison du maximum des valeurs absolues des variations de même variable entre deux itérations successives et une certaine précision de calcul.

$$\text{Max} (|\Delta x_1^k|, |\Delta x_2^k|, |\Delta x_3^k|, \dots, |\Delta x_{n-1}^k|, |\Delta x_n^k|) \leq \varepsilon \quad (4-30)$$

Avec l'expérience pour des problèmes particuliers, le nombre d'itérations exigés pour la convergence peut être réduit si nous changeons l'ancien valeur par

$$x_i^k = x_i^{k-1} + \sigma \Delta x_i \quad (4-31)$$

Le coefficient  $\sigma$  est appelé facteur d'accélération et est choisi arbitrairement. Pour les problèmes ordinaires de load flow,  $\sigma = 1,6$  converge rapidement vers la solution

### **Expression des pertes actives**

Bilan de puissance active :

$$Pertes\ actives = \sum P_G - \sum P_C \quad (4-32)$$

Expression en fonction des composantes réelles et imaginaires des tensions

Soit deux nœud p et q reliés par une ligne

Posons

$$U_p = e_p + jf_p \quad (4-33)$$

Et

$$U_q = e_q + jf_q \quad (4-34)$$

Le transit de puissance de p vers q a comme expression :

$$\bar{S}_{pq} = P_{pq} + jQ_{pq} = U_p \cdot I_{pq}^* \quad (4-35)$$

Soit :

$$\bar{S}_{pq}^* = \bar{U}_p^* \cdot \bar{I}_p = \bar{U}_p^* \cdot \left[ (\bar{U}_p - \bar{U}_q) Y_{pq} + \bar{U}_p \cdot \frac{Y_{tpq}}{2} \right] \quad (4-36)$$

$$\bar{S}_{pq}^* = (e_p - jf_p) \left[ (e_p + jf_p - e_q - jf_q) Y_{pq} + (e_p + jf_p) \frac{Y_{tpq}}{2} \right] \quad (4-37)$$

Le transit de puissance de q vers p a comme expression

$$\bar{S}_{qp} = P_{qp} + jQ_{qp} = U_q \cdot I_{qp}^* \quad (4-38)$$

Soit :

$$\bar{S}_{qp}^* = \bar{U}_q^* \cdot \bar{I}_q = \bar{U}_q^* \cdot \left[ (\bar{U}_q - \bar{U}_p) Y_{pq} + \bar{U}_q \cdot \frac{Y_{tpq}}{2} \right] \quad (4-39)$$

$$\bar{S}_{qp}^* = (e_q - jf_q) \left[ (e_q + jf_q - e_p - jf_p) Y_{pq} + (e_q + jf_q) \frac{Y_{tpq}}{2} \right] \quad (4-40)$$

La perte de puissance dans la branche qui relie p et q est la somme algébrique du transit de puissance de p vers q et de q vers p

$$\bar{S}_{pertes} = P_{pertes} + jQ_{pertes} = \bar{S}_{pq}^* + \bar{S}_{qp}^* \quad (4-41)$$

En remplaçant (4-37) et (4-40) dans (4.41), on a :

$$\bar{S}_{pertes} = \left[ (e_p - e_q)^2 + (f_p - f_q)^2 \right] Y_{pq} + (e_p^2 + e_q^2 + f_p^2 + f_q^2) \frac{Y_{tpq}}{2} \quad (4-42)$$

Puisque  $Y = G + jB$

(4-42) devient :

$$\bar{S}_{pertes} = \left[ (e_p - e_q)^2 + (f_p - f_q)^2 \right] G_{pq} + (e_p^2 + e_q^2 + f_p^2 + f_q^2) \frac{G_{tpq}}{2} + \quad (4-43)$$

$$j \left[ \left[ (e_p - e_q)^2 + (f_p - f_q)^2 \right] B_{pq} + (e_p^2 + e_q^2 + f_p^2 + f_q^2) \frac{B_{tpq}}{2} \right]$$

$$P_{pertes} + jQ_{pertes} = \left[ (e_p - e_q)^2 + (f_p - f_q)^2 \right] G_{pq} + (e_p^2 + e_q^2 + f_p^2 + f_q^2) \frac{G_{tpq}}{2} + j \left[ \left[ (e_p - e_q)^2 + (f_p - f_q)^2 \right] B_{pq} + (e_p^2 + e_q^2 + f_p^2 + f_q^2) \frac{B_{tpq}}{2} \right] \quad (4-44)$$

En identifiant la partie réelle, on a la perte de puissance active dans la branche pq

$$P_{pertes} = \left[ (e_p - e_q)^2 + (f_p - f_q)^2 \right] G_{pq} \quad (4-45)$$

Les pertes totales de puissance actives dans un réseau a donc pour expression :

$$P_{pertes\ tot} = \sum_{k=1}^{nl} \left[ (e_p - e_q)^2 + (f_p - f_q)^2 \right] G_{pq} \quad (4-46)$$

$nl$  : Nombre des lignes de transport

$G_{pq}$  : Conductance de la  $kième$  ligne reliant les nœuds p et q

$e_p, e_q$  : Partie réelle de la tension des nœuds p et q

$f_p, f_q$  : Partie imaginaire de la tension des nœuds p et q

Expression en fonction des modules et arguments

On peut montrer simplement que les pertes totales de puissance active a pour expression :

$$P_{pertes} = \sum_{k=1}^{nl} [V_p^2 + V_q^2 - 2V_p V_q \cos \theta_{pq}] G_{pq} \quad (4-47)$$

$V_p$  : Amplitude de la tension au nœud p

$V_q$  : Amplitude de la tension au nœud q

$\theta_{pq}$  : Déphasage de la tension au nœud p par rapport à la tension au nœud q

$G_{pq}$  : Conductance de la  $kième$  ligne reliant les nœuds p et q

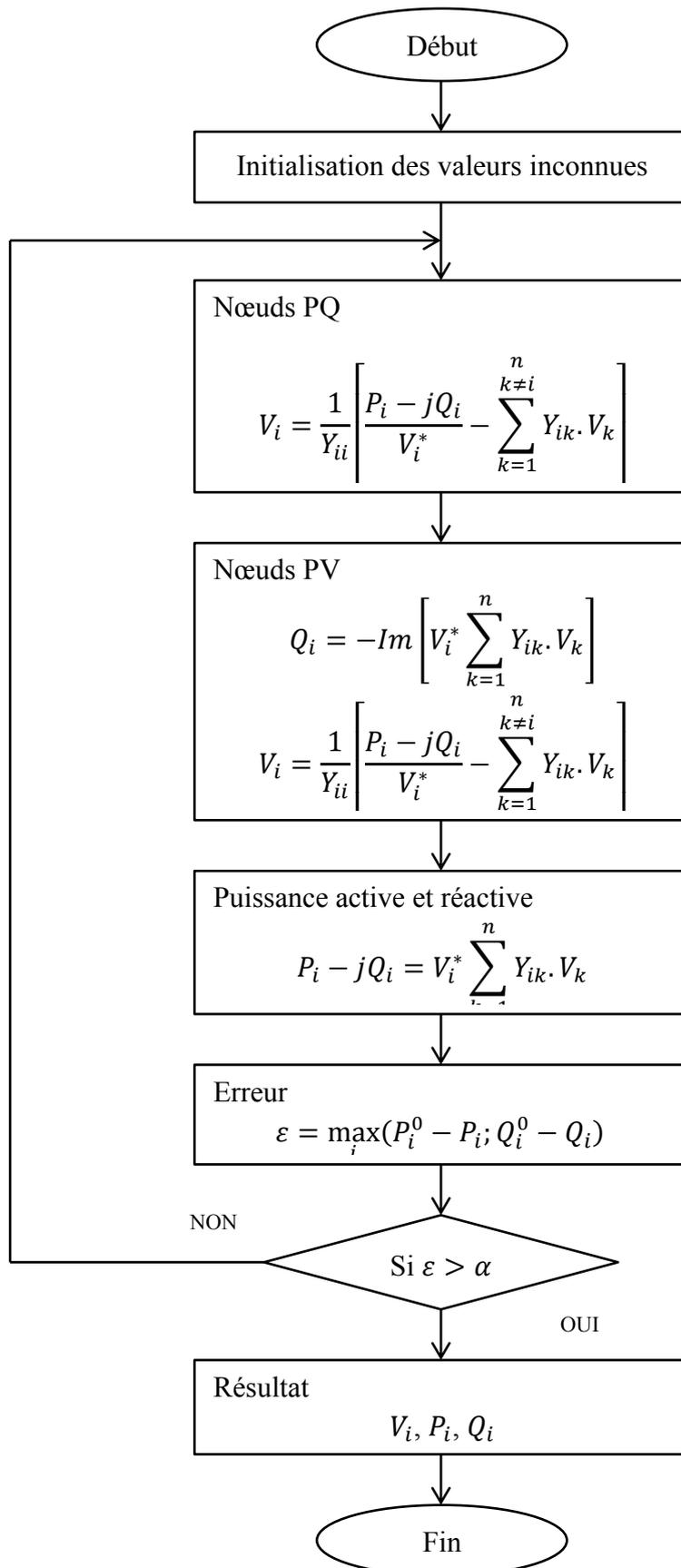


Figure 4-4 : Organigramme de calcul de l'écoulement de puissance

#### 4.4. Optimisation de la puissance active d'un réseau

L'optimisation de la puissance active dans un réseau a généralement pour but de réduire les pertes de puissance lors de l'acheminement de l'énergie électrique. On est donc ici face à un problème de minimisation. Dans le chapitre 1 de ce mémoire, on a énoncé que dans un problème d'optimisation, on doit d'abord définir la fonction objective à minimiser puis de donner toutes les contraintes du problème.

##### 4.4.1. La fonction objective de l'optimisation de la puissance active

Notre objectif étant de réduire les pertes actives totales du réseau. D'après ce qu'on a vu dans le paragraphe calcul des pertes actives, on a deux expressions pour calculer les pertes en puissances actives, ce qui implique qu'on a alors deux choix sur la fonction objective à utiliser :

###### Choix n°1

Minimiser

$$\sum_{k=1}^{nl} [(e_p - e_q)^2 + (f_p - f_q)^2] G_{pq} \quad (4-48)$$

$nl$  : Nombre des lignes de transport

$G_{pq}$  : Conductance de la  $k$ ème ligne

$e_p, e_q$  : Partie réelle de la tension des nœuds p et q

$f_p, f_q$  : Partie imaginaire de la tension des nœuds p et q

###### Choix n°2

Minimiser

$$F = \sum_{k=1}^{nl} [V_p^2 + V_q^2 - 2V_p V_q \cos \theta_{pq}] G_{pq} \quad (4-48)$$

$V_p$  : Amplitude de la tension au nœud p

$V_q$  : Amplitude de la tension au nœud q

$\theta_{pq}$  : Déphasage de la tension au nœud p par rapport à la tension au nœud q

$G_{pq}$  : Conductance de la  $k$ ème ligne reliant les nœuds p et q

#### 4.4.2. Les contraintes du réseau correspondantes à l'optimisation de la puissance active

On a deux sortes de contraintes :

##### 4.4.2.1. La contrainte d'égalité

Ces contraintes reflètent les lois physiques régissant le système électrique. Elles sont représentées par les équations non linéaires de l'écoulement de puissance. Il faut que la somme des puissances active et réactive injectées dans chaque jeu de barres soit égale à zéro.

Les contraintes égalités assurent alors l'équilibre de la puissance générée et de la puissance absorbée par tous les charges et les pertes du réseau :

$$\sum_{i=1}^{ng} P_G = \sum_{j=1}^{nc} P_C + \sum_{k=1}^{nl} P_{pertes} \quad (4-49)$$

$ng$  : Nombre de nœud de production

$nc$  : Nombre ce nœud de consommation

$nl$  : Nombre de branche de liaison

##### 4.4.2.2. Les contraintes d'inégalités

En pratique, on ne doit pas dépasser les limites des éléments physiques du réseau électrique tels que la puissance active générée par les nœuds de production, les tensions dans tous les nœuds du réseau, on peut citer ces contraintes d'inégalité comme suite :

- La puissance active du nœud de référence :

$$P_{G \text{ réf}}^{\min} \leq P_{G \text{ réf}} \leq P_{G \text{ réf}}^{\max}$$

- Les tensions dans les nœuds de consommation

$$V_{C,i}^{\min} \leq V_{C,i} \leq V_{C,i}^{\max} \quad i: \text{nombre des nœuds de consommation}$$

- Les puissances réactives générées par les nœuds de production

$$Q_{G i}^{\min} \leq Q_{G i} \leq Q_{G i}^{\max} \quad \text{où } i: \text{nombre des générateurs}$$

Le problème d'optimisation de la puissance active pourra être formulé dans le tableau suivant :

Expression de la fonction objective en fonction des composantes réelles et imaginaires des tensions	Expression de la fonction objective en fonction des modules et arguments des tensions
<p><i>Minimiser</i></p> $F = \sum_{k=1}^{nl} [(e_p - e_q)^2 + (f_p - f_q)^2] G_{pq}$ <p><i>Sous les contraintes</i></p> $P_{G \text{ réf}}^{\min} \leq P_{G \text{ réf}} \leq P_{G \text{ réf}}^{\max}$ $V_{C,i}^{\min} \leq V_{C,i} \leq V_{C,i}^{\max}$ $Q_{G i}^{\min} \leq Q_{G i} \leq Q_{G i}^{\max}$	<p><i>Minimiser</i></p> $F = \sum_{k=1}^{nl} [V_p^2 + V_q^2 - 2V_p V_q \cos \theta_{pq}] G_{pq}$ <p><i>Sous les contraintes</i></p> $P_{G \text{ réf}}^{\min} \leq P_{G \text{ réf}} \leq P_{G \text{ réf}}^{\max}$ $V_{C,i}^{\min} \leq V_{C,i} \leq V_{C,i}^{\max}$ $Q_{G i}^{\min} \leq Q_{G i} \leq Q_{G i}^{\max}$

*Tableau 4-3: Les fonctions objectives permettant de minimiser pertes totales en puissances actives*

Pour notre étude, on a choisi d'optimiser la puissance active en utilisant l'expression de la fonction objective en fonction des modules et arguments des tensions.

Pour optimiser cette fonction objective, on a choisi d'utiliser la méthode de recuit simulé déjà étudiée dans le paragraphe 3.2

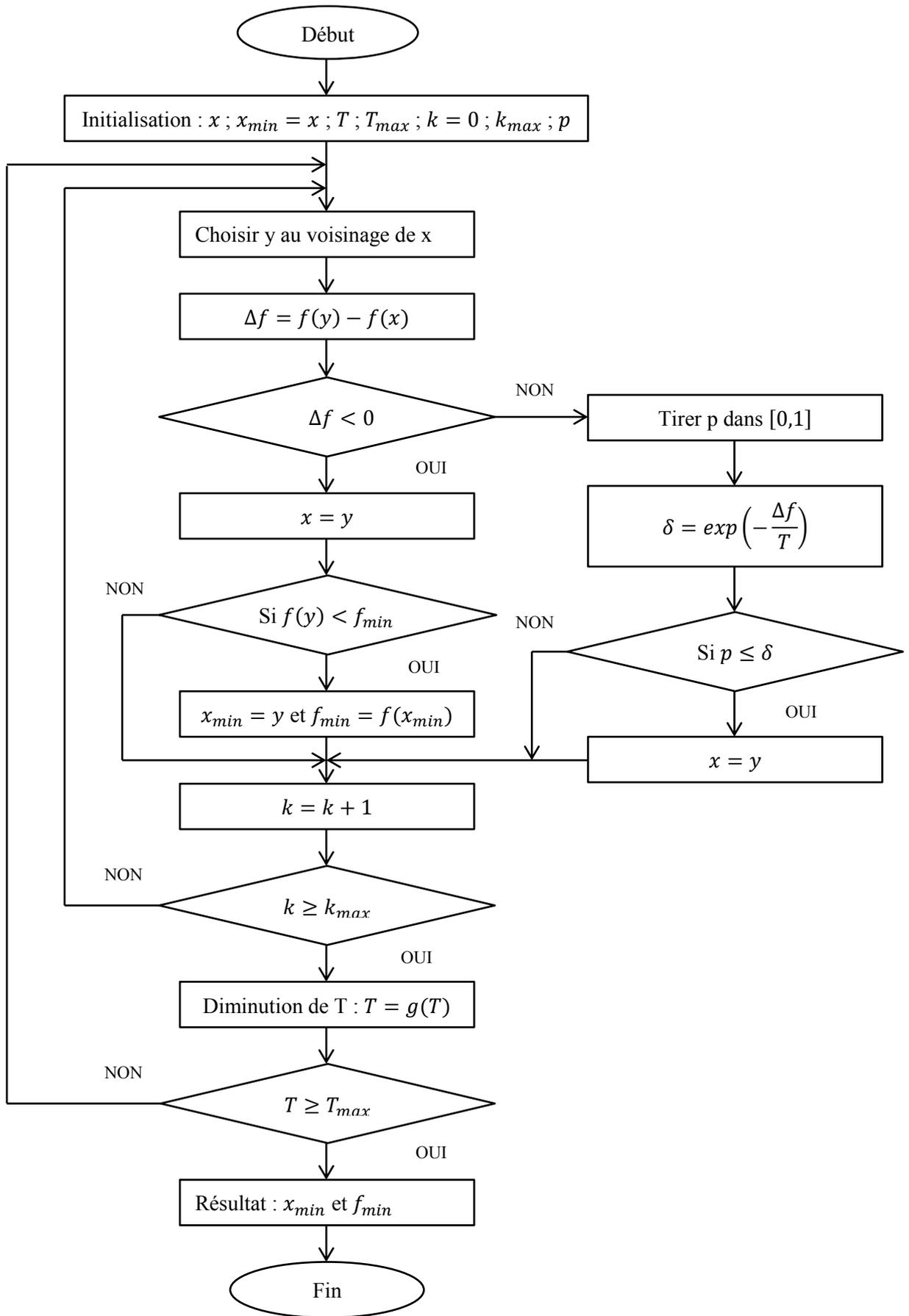


Figure 4-5 : Organigramme de l'optimisation par la méthode de recuit simulé

### 4.5. Représentation du réseau

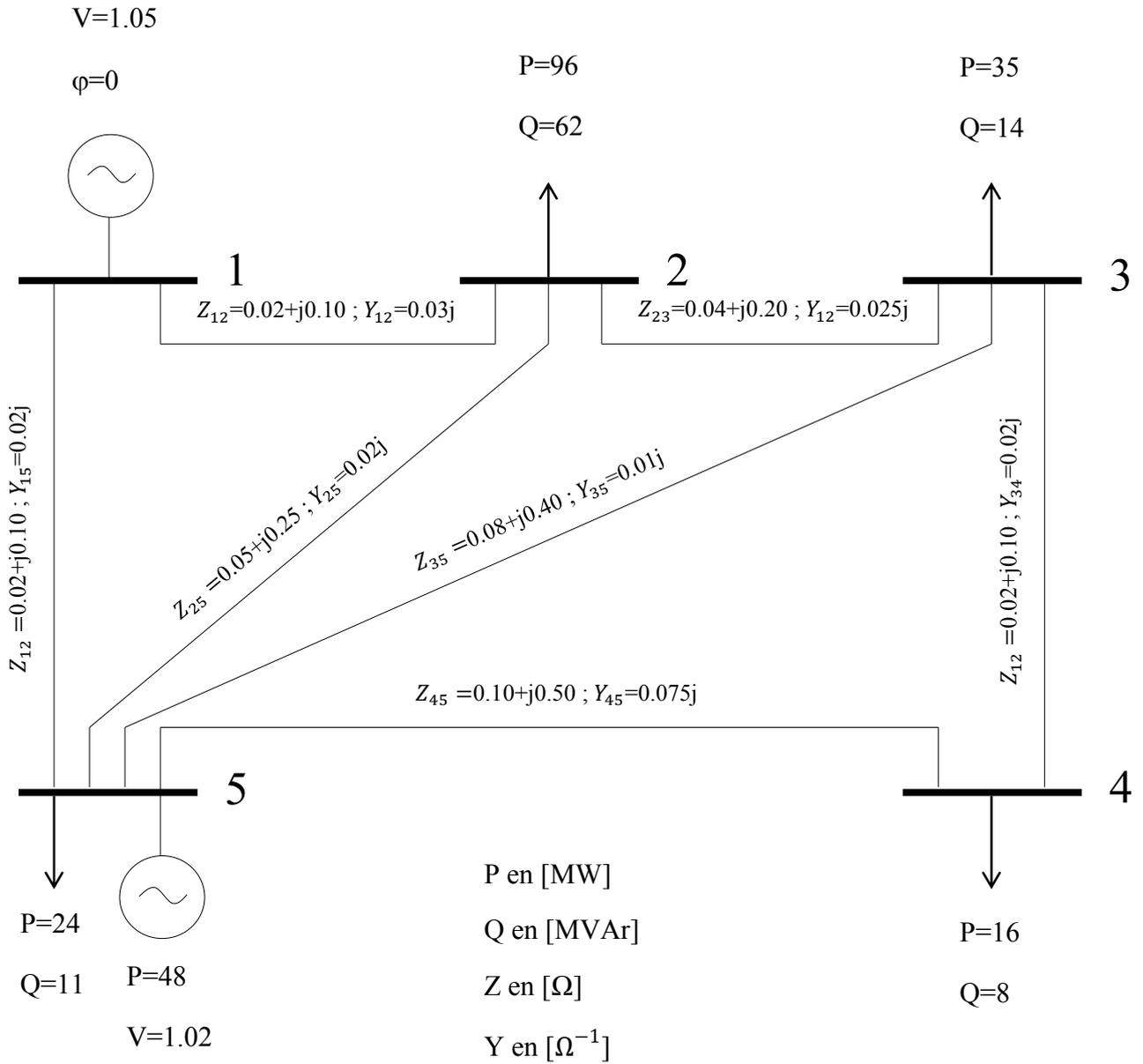


Figure 4-6: Représentation du réseau à étudier

Ligne	Impédance $Z_{pq}$ [pu]	Admittance de charge $Y_{pq}/2$ [pu]
1-2	0.02+j0.10	0+j0.030
1-5	0.05+j0.25	0+j0.020
2-3	0.04+j0.20	0+j0.025
2-5	0.05+j0.25	0+j0.020
3-4	0.05+j0.25	0+j0.020
3-5	0.08+j0.40	0+j0.010
4-5	0.10+j0.50	0+j0.075

Tableau 4-4: Tableau d'impédance de ligne et d'admittance de charge par rapport à la terre.

Nœud	Puissance de produite		Puissance de la charge		Tension	
	P [MW]	Q [MVAR]	P [MW]	Q [MVAR]	Amplitude	Phase
1	-	-	0	0	1,05	0
2	0	0	96	62	-	-
3	0	0	35	14	-	-
4	0	0	16	8	-	-
5	48	-	24	11	1,02	-

Tableau 4-5 : Tableau de planification.

#### 4.6. Résultat

Dans ce mémoire, on a utilisé le logiciel de calcul Matrix Laboratory (Matlab) pour le calcul de la répartition de puissance ainsi que l'optimisation proprement dit de la puissance active

##### 4.6.1. Résultat du calcul de la répartition de puissance (load flow)

Le calcul de la répartition de puissance a été effectué avec l'algorithme de Gauss Seidel.

La matrice d'admittance :  $Y_{bus}$  :

Nœud	1	2	3	4	5
1	2.69-13.41j	-1.92+9.61j	0	0	-0.76+3.84j
2	-1.92+9.61j	3.65-18.19j	-0.96+4.80j	0	-0.76+3.84j
3	0	-0.96+4.80j	2.21-11.00j	-0.76+3.84j	-0.48+2.40j
4	0	0	-0.76+3.84j	1.15-5.67j	-0.38+1.92j
5	-0.76+3.84j	-0.76+3.84j	-0.48+2.40j	-0.38+1.92j	2.40-11.89j

Tableau 4-6 :  $Y_{bus}$  obtenu après calcul

4.6.1.1. Les tensions des nœuds

Nœud	Amplitude [pu]	Phase [degré]
1	1.05	0
2	0.98	-0.08
3	0.97	-0.12
4	0.98	-0.12
5	1.02	-0.05

Tableau 4-7 : Les tensions des nœuds

4.6.1.2. Les puissances des nœuds

Nœud	Puissance active [MW]	Puissance réactive [MVar]
1	126.59	57.10
2	-95.99	-62.00
3	-35.00	-14.00
4	-16.00	-8.00
5	24.00	5.86

Tableau 4-8 : Les Puissances dans les nœuds

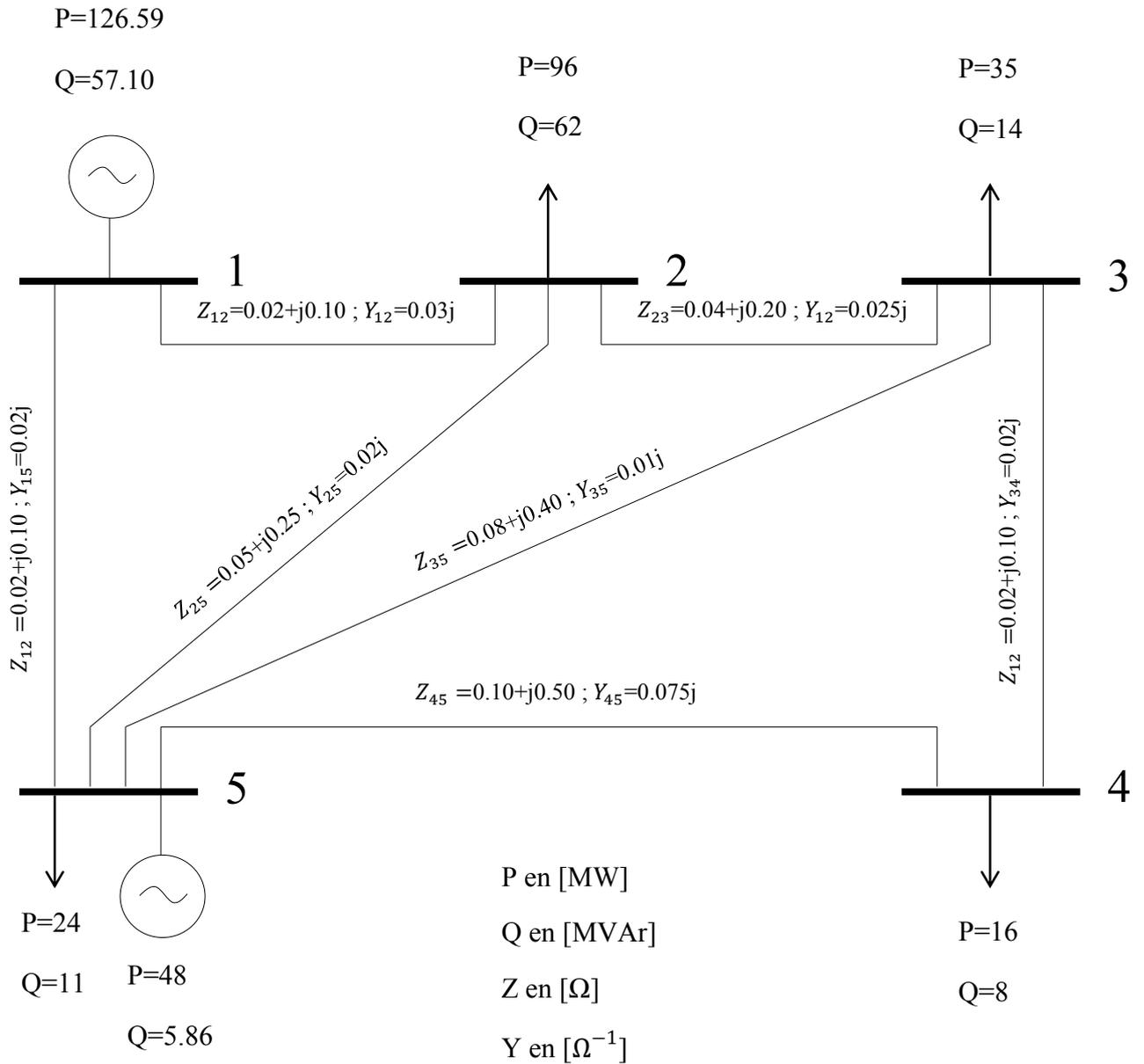


Figure 4-7 : Représentation des puissances sur le réseau

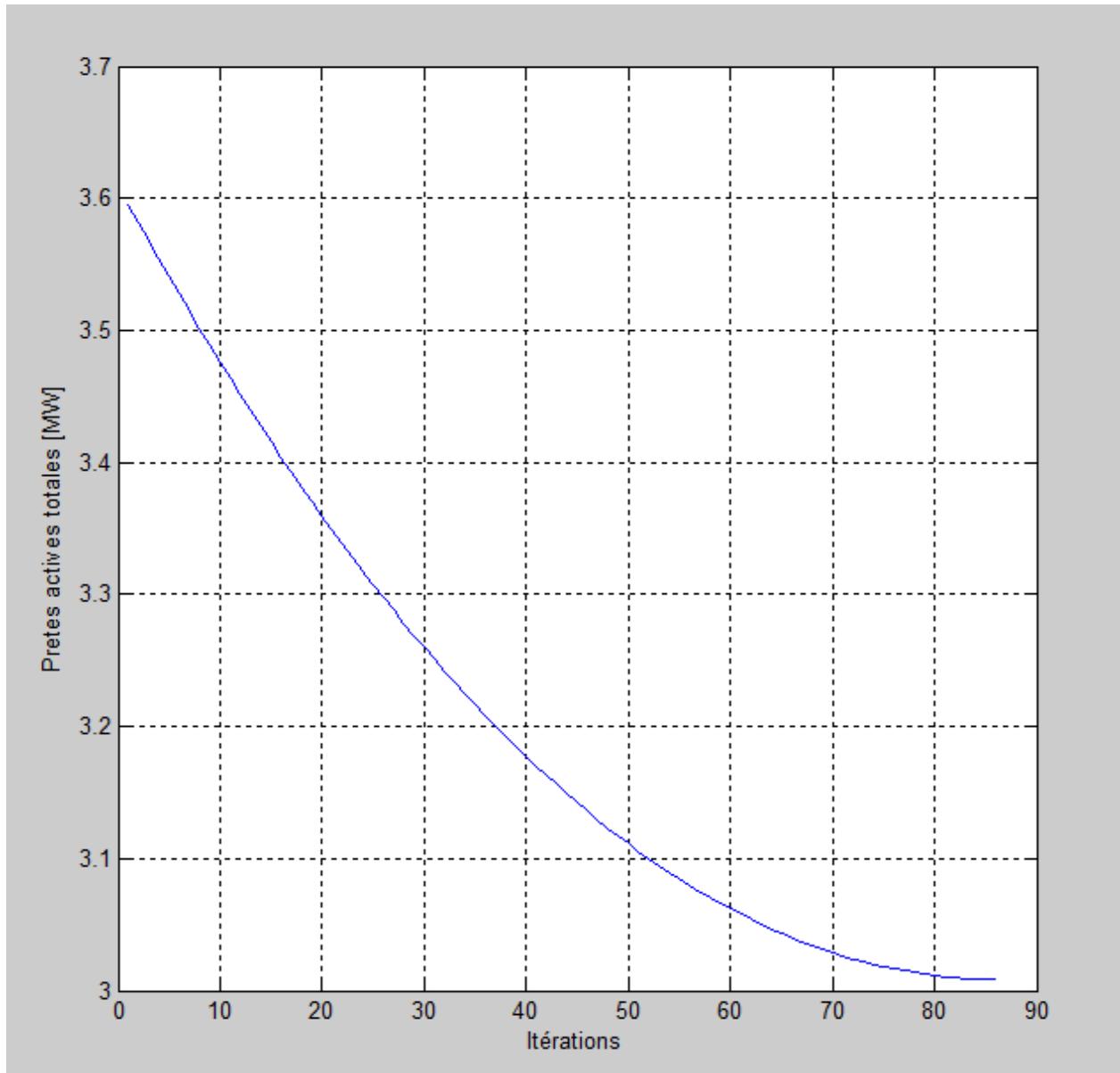
4.6.1.3. Les pertes de puissances

Les pertes actives [MW]	Les pertes réactives [Mvar]
3.60	-22.30

Tableau 4-9 : Les pertes de puissances actives et réactives

#### 4.6.2. Résultat de l'optimisation de la puissance active

Nombre d'itérations : 86



*Figure 4-8 : Pertes actives totales en fonction de l'itération*

Cette figure montre qu'au fil de l'itération, la méthode de recuit simulé minimise progressivement la fonction objective. Le programme s'arrête aux 86ème itérations où le minimum est atteint.

#### 4.6.2.1. Les tensions des nœuds

Nœud	Amplitude [pu]	Phase [degré]
1	1.13	0
2	1.09	-0.07
3	1.10	-0.10
4	1.12	-0.11
5	1.16	-0.05

*Tableau 4-10 : Les tensions des nœuds après optimisation*

#### 4.6.2.2. Les pertes de puissances

Les pertes actives [MW]	Les pertes réactives [Mvar]
3.01	-36.07

*Tableau 4-11 : Les pertes de puissances actives et réactives après l'optimisation*

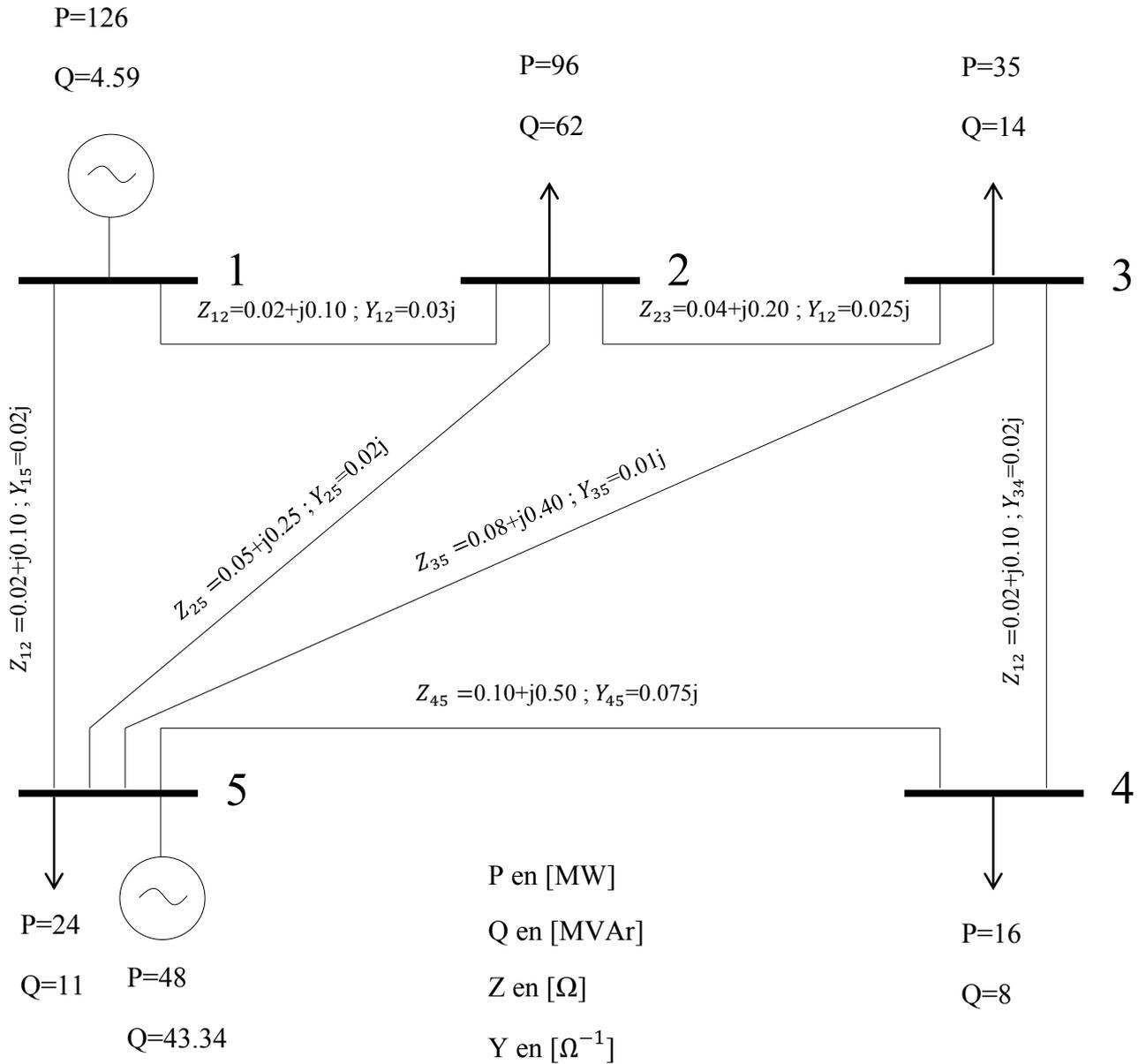


Figure 4-9 : Représentation du réseau avec les nouvelles valeurs des puissances

#### 4.6.3. Comparaison des résultats

	Pertes de puissance active
Résultat obtenu après répartition de puissance utilisant la méthode de Gauss Seidel	3.60
Résultat obtenu après optimisation utilisant la méthode de recuit simulé	3.01

Tableau 4-12 : Comparaison des résultats obtenus par le load flow et l'optimisation

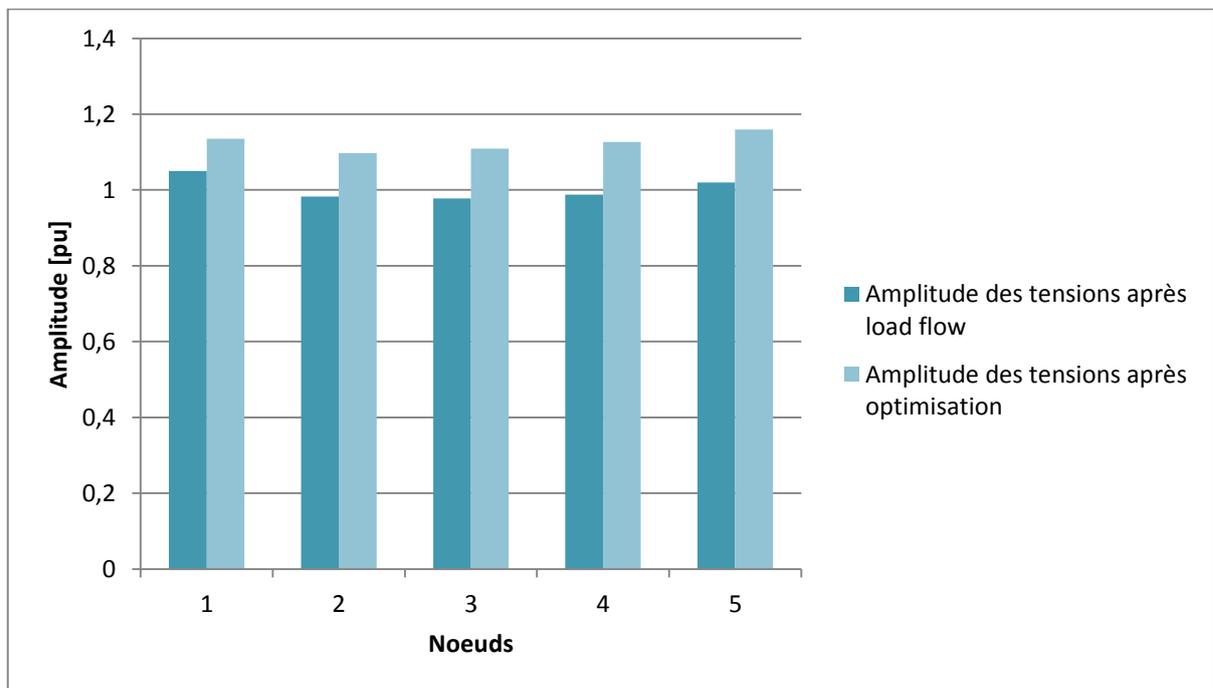


Figure 4-10: Comparaison de l'amplitude des tensions avant et après optimisation

### Interprétation

Après avoir analysé les résultats obtenus, nous abordons la discussion suivante.

D'après les résultats obtenus sur les comparaisons précédentes, on voit qu'après l'optimisation, les tensions des nœuds sont augmentées et par conséquent on a diminué les pertes de 16.38%, ce qui signifie qu'après avoir minimisé on a économisé une puissance de 590 [KW].

### 4.7. Conclusion

Dans ce chapitre, on a réduit les pertes de puissance active dans un réseau, pour arriver à cet objectif, on a d'abord calculé la répartition de puissance de ce réseau par l'algorithme d'itération de Gauss Seidel qui est une méthode simple et efficace et même rapide pour obtenir les paramètres du réseau. Ce calcul nous a permis de déduire les pertes obtenues par le transit de puissance dans les lignes, ensuite les paramètres obtenus par la répartition de puissance ont été exploités comme point de départ ou solution initiale de la méthode de recuit simulé. Après optimisation, on a pu réduire jusqu'à 19,44% les pertes totales de puissance active dans le réseau. Pour conclure, il est avantageux de procéder à l'optimisation de la puissance active avant de planifier un réseau électrique.

# CONCLUSION GENERALE

Au cours de ce mémoire, nous avons eu l'occasion d'exposer la synthèse de toutes les méthodes d'optimisation.

Dans le présent travail, notre principal objectif est alors de maîtriser les différentes méthodes d'optimisations.

Le premier chapitre nous a permis d'avoir une idée générale sur le problème d'optimisation, on a alors développé dans ce chapitre le concept de base de l'optimisation, ceci nous a donné l'occasion de pouvoir définir, de classer, de différencier, et enfin de formuler un problème d'optimisation.

Le second chapitre concerne la méthode ancienne pour la résolution d'un problème d'optimisation, ce chapitre nous a donné l'opportunité de se familiariser sur les méthodes manipulant des fonctions linéaire et non linéaire. En effet, la méthode de Simplexe qui est connue par son efficacité à résoudre des problèmes linéaires a été abordée dans ce chapitre, sans oublier les méthodes de résolution des problèmes non linéaires avec et sans contraintes basées sur le principe de descente.

Le troisième chapitre a été consacré pour l'optimisation moderne ou optimisation stochastique. Cette méthode est très intéressante grâce à son mécanisme de transition probabiliste et aléatoire qui permet l'exploration de toute l'espace de recherche et la convergence vers l'optimum global. Cette méthode est alors connue sur son avantage de résoudre les problèmes dont la fonction objective n'est pas dérivable.

Dans le dernier chapitre, on a testé la méthode du Recuit Simulé pour optimiser les pertes en puissance active dans un réseau d'énergie électrique. Pour se faire, on a d'abord calculé la répartition de puissance dans un réseau donné par l'algorithme itératif de Gauss Seidel en dégageant les pertes actives enregistrées, cette valeur a ensuite été comparée avec celle obtenue après minimisation de pertes totales et les résultats ont montré qu'il est possible de réduire les pertes actives totales dans un réseau et qu'il est toujours avantageux pour une entreprise qui produit et distribue de l'énergie électrique d'optimiser son rendement en minimisant les pertes par l'acheminement de l'énergie.

En guise de conclusion, l'étude dans ce mémoire m'a permis non seulement de comprendre le concept de base de l'optimisation avec sa résolution mais aussi de m'avoir donné une

opportunité de me pencher sur les concepts mathématiques de base et également de me familiariser sur la programmation. D'autre part, constaté lors de ce mémoire, il serait plus favorable de résoudre en même temps dans un seul problème la minimisation des pertes en puissance active, la compensation de la puissance réactive et le dispatching économique qui est le domaine d'application de l'optimisation multi-objective ou optimisation multicritère.

# REFERENCES

[1] : ABDERREZEK Hadjer « contribution à la commande robuste des systèmes non linéaires » mémoire de magistère présenté à l'Université Ferhat Abbas -Setif-1 ufas1 (Algérie) - octobre 2010

[2] : Alain BERRO « Optimisation multi objectif et stratégies d'évolution en environnement dynamique », Thèse de doctorat à l'Université des sciences Sociales Toulouse I - Décembre 2001

[3] : Anatoli Iouditski « OPTIMISATION Analyse convexe Théorie de programmation non-linéaire » Notes de cours à Université Joseph Fourier

[4] : Atousa ASSADI-HAGHI « Contribution au développement de méthodes d'optimisation structurelle pour la conception assistée par ordinateur de composants et de circuits hyperfréquences », Thèse de doctorat de l'UNIVERSITE DE LIMOGES - Mars 2007

[5] : BELLOUFI Abderrahim « Optimisation du processus d'usinage à l'aide de la programmation non linéaire (P.N.L) » Mémoire de magistère - Février 2008

[6] : Benjamin Monmege « Cours optimisation » - 29 février 2012

[7] : Charles-Edmond BICHOT « Elaboration d'une nouvelle métaheuristique pour le partitionnement de graphe : la méthode de fusion-fission. Application au découpage de l'espace aérien » Thèse présenté à l'Institut national polytechnique de Toulouse – Novembre 2007

[8] : COSTANZO Andrea « Optimisation par colonies de fourmis » Article - mai 2006

[9] : Johann Dréo « Métaheuristicques d'optimisation vues sous l'angle de l'échantillonnage de distribution », École Nationale Supérieure des Mines de Saint-Étienne

[10] : D. BELFADEL « De la fourmi réelle à la fourmi artificielle » Article publié à l'Université Mouloud MAMMERI de Tizi-Ouzou

- [11] : Edouard GENGEMBRE « Contribution à la construction de méta-modèles pour l'optimisation sous contraintes de critères énergétiques : Application à la conception des bâtiments » Thèse de doctorat présenté à L'Université de Toulouse – Décembre 2011
- [12] : F. Malrieu « Recuit simulé et le voyageur de commerce » Article publié à Université Rennes I – Mars 2008
- [13] : Fabien PICAROUGNE « Expérimentations de méthodes parallèles de recherche d'information sur internet » Article publié à Université de Tours – 2005
- [14] : Frédéric de Gournay & Aude Rondepierre « Introduction à l'Optimisation Numérique » Cours à l'INSA Toulouse
- [15] : FRIDJAT Zineddine et TAMMA Mohammed Elhadi « Application des algorithmes génétiques à l'optimisation de la production énergie active dans réseau électrique » Mémoire présenté à Université d'EL-Oued – Juin 2014
- [16] : HERNANE Soumeya « Deux stratégies parallèles de l'optimisation par colonie de fourmis » Article publié à l'USTO – 2007
- [17] : Ionel Sorin CIUPERCA « COURS OPTIMISATION » Cours à l'ISFA
- [18] : Jean-Marc Alliot « Algorithmes génétiques » Article – Mars 2005
- [19] : Jean-Pol Guillement « Résumé d'Optimisation » - 2011
- [20] : K. Lenin & B. Ravindranath Reddy & IIM. Surya Kalavathi « Reduction of Active Power Loss and Improvement of Voltage Profile by Using Intellectual Particle Swarm Optimization » Article de IJRECT - 2014
- [21] : Kamal BOUDJELABA « Contribution à la conception des filtres bidimensionnels non récursifs en utilisant les techniques de l'intelligence artificielle : application au traitement d'images » Thèse présenté à l'Université d'Orléans - Juin 2014
- [22] : Kanagasabai Lenin & Bhumanapally Ravindhranath Reddy & Munagala Suryakalavathi « Reduction of real power loss by using Kudu Herd Algorithm » Article de Jawaharlal Nehru Technological University Kukatpally – 2014

- [23] : Laetitia Jourdan « Métaheuristiques pour l'extraction de connaissances : Application à la Génomique » Thèse de doctorat présenté à 'Université de Lille – Novembre 2003
- [24] : Laurent SMOCH « Méthodes d'Optimisation », Licence Professionnelle, Université du Littoral - Septembre 2011
- [25] : Marino WIDMER « Les metaheuristiques : des outils performants pour les problèmes industriels » Article publié à Université de Fribourg – Avril 2001
- [26] : Mehdi Jaoua « Algorithme de recherche tabou pour la planification optimale d'une campagne marketing sur les moteurs de recherche » Mémoire présenté à l'Université de Montréal – Août 2014
- [27] : Nabil MANCER « Contribution à l'optimisation de la puissance réactive en présence de dispositifs de compensation dynamique (FACTS) Mémoire présenté à l'Université Mohamed Khider – Biskra – Janvier 2012
- [28] : Nicolas MONMARCHE « Algorithmes de fourmis artificielles : Applications à la classification et à l'optimisation » Thèse de doctorat présenté à l'Université de Tours – Décembre 2000
- [29] : Omessaad HAJJI « Contribution au développement de méthodes d'optimisation stochastique. Application à la conception des dispositifs électrotechniques », Thèse de doctorat de l'Université des sciences et Technologies de Lille - Décembre 2003.
- [30] : OULD-SAADY Youcef « Conception d'une Métaheuristique Réactive » Mémoire présenté à l'Université Mohamed Boudiaf – 2012
- [31] : Philippe Henniges « PSO pour l'apprentissage supervisé des réseaux neuronaux de type fuzzy artmap » mémoire présenté à université du Québec – Juillet 2006
- [32] : R. Herbin « Optimisation », Cours télé-enseignement, Université Aix-Marseille 1 - Décembre 2011
- [33] : Rabah BENABID « Optimisation Multiobjectif de la Synthèse des FACTS par les Particules en Essaim pour le Contrôle de la Stabilité de Tension des Réseaux Electriques » Mémoire présenté à l'Université Amar Telidji, Laghouat – Septembre 2007
- [34] : Selvaraj Ramkumar « Algorithmes génétiques » Article – Avril 2007

- [35] : Shishir Dixit « Optimal Allocation of SVC for Minimization of Power Loss and Voltage Deviation using NSGA-II » Article du Department of Electrical Engineering MITS, Gwalior, India – 2014
- [36] : Sidi Mohamed Douiri « Cours des Méthodes de Résolution Exactes Heuristiques et Métaheuristiques » Cours
- [37] : Sofiene Kachroudi « Commande et optimisation pour la régulation du trafic urbain multimodale sur de grands réseaux urbains » Thèse Présentée à L'université d'Evry Val d'Essone – Janvier 2010
- [38] : Soumana Beidi HAMMA « Etude de methodes numeriques d'optimisation globale » Thèse de doctorat de l'université paul sabatier - Mars 1992
- [39] : Souquet Amédée & Radet Francois-Gérard « ALGORITHMES GENETIQUES » TE de fin d'année – Juin 2004
- [40] : T. Hariharan, S. Raja & S. Hareesh « Real Power Loss Minimization using Fuzzy logic Controller » Article
- [41] : Tebani Wissame « Compensation optimale de l'énergie réactive dans les réseaux de distribution » Mémoire présenté à l'Université Ferhat Abbas – Sétif – Juin 2012
- [42] : Thierno DIALLO « Etude et illustration de méthodes itératives d'optimisation non linéaire », Projet de master de l'Ecole Polytechnique Fédérale de Lausanne - Avril 2006
- [43] : Xavier ANTOINE « Introduction à l'optimisation : aspects théoriques, numériques et algorithmes », Institut National Polytechnique de Lorraine – 2007
- [44] : Yann COOREN « Perfectionnement d'un algorithme adaptatif d'Optimisation par Essaim Particulaire. Applications en génie médical et en électronique » Thèse de doctorat de L'UNIVERSITÉ PARIS 12 VAL DE MARNE - Novembre 2008
- [45] : ZEHANI RIM « Contribution à l'analyse des méthodes de répartition optimale des puissances dans un réseau de transport d'énergie » Mémoire de Magister présenté à l'UNIVERSITE BADJI MOKHTAR-ANNABA – 2007
- [46] : [www.wikipedia.org/optimisation/](http://www.wikipedia.org/optimisation/)

# ANNEXES

## ANNEXE A : CALCUL DE LA REPARTITION DE PUISSANCE PAR LA METHODE DE GAUSS SEIDEL

```
'REPARTITION DE PUISSANCE PAR LA METHODE DE GAUSS SEIDEL'  
clear all  
clc  
%Nombre de bus P-Q  
np=input('nombre de BUS de type P-Q = ');  
%Nombre de bus P-V  
ng=input('nombre de BUS de type P-V = ');  
%Nombre de bus total  
n=np+ng+1;  
  
%matrice zz  
for i=1:1:n  
    for j=1:1:n  
        disp(['zz(', num2str(i), ';', num2str(j), ')=']);  
        zz(i,j)=input('');  
    end  
end  
  
%Matrice ych  
for i=1:1:n  
    for j=1:1:n  
        disp(['ych(', num2str(i), ';', num2str(j), ')=']);  
        ych(i,j)=input('');  
    end  
end  
  
d2r=pi/180;  
w=100*pi;  
  
%vecteur P  
for i=1:1:np  
    disp(['p(', num2str(i), ')=']);  
    p(i)=input('');  
end  
  
%vecteur Q  
for i=1:1:np  
    disp(['q(', num2str(i), ')=']);  
    q(i)=input('');  
end  
  
%Tension v
```

```
for i=1:1:np
    disp(['v(', num2str(i), '=']);
    v(i)=input('');
end

%Angle
for i=1:1:np
    disp(['fi(', num2str(i), '=']);
    fi(i)=input('');
end

% La matrice d'admittance
[ybus]=ybus(zz,ych);
ybus;
g=real(ybus);b=imag(ybus);

acc=input('Enter the acceleration constant: ');
del=1;indx=0;

% Itération de Gauss Seidel
while del>1e-6
    % Tension du noeud P-Q
    for i=2:np+1
        vint1=(p(i)-j*q(i))/conj(v(i));
        vint2=0;
        for k=1:n
            if (i==k)
                vint2=vint2+0;
            else
                vint2=vint2+ybus(i,k)*v(k);
            end
        end
        vt=(vint1-vint2)/ybus(i,i);
        v(i)=v(i)+acc*(vt-v(i));
    end

    %Puissance reactive du BUS P-V
    vint3=0;
    for i=2+np:1:n
        vint3=0;
        for k=1:1:n
            vint3=vint3+ybus(i,k)*v(k);
        end
        qq(i)=-imag(conj(v(i))*vint3);
    end

    %Tension du noeud P-V
    for i=2+np:1:n
        vint1=(p(i)-j*qq(i))/conj(v(i));
        vint2=0;
```

```
    for k=1:1:n
        if (i==k)
            vint2=vint2+0;
        else
            vint2=vint2+ybus(i,k)*v(k);
        end
    end
    vt=(vint1-vint2)/ybus(i,i);
    v(i)=abs(v(i))*vt/abs(vt);
end

% Puissance apparente
for i=1:5
    sm=0;
    for k=1:5
        sm=sm+ybus(i,k)*v(k);
    end
    s(i)=conj(v(i))*sm
end

% L'erreur
delp=p-real(s)';
delq=q+imag(s)';
delpq=[delp(2:n);
        delq(2:np+1)];
del=max(abs(delpq));
indx=indx+1;
if indx==1
    pause
end

end

%pertes en puissance active
x=[real(v(1)) imag(v(1));
    real(v(2)) imag(v(2));
    real(v(3)) imag(v(3));
    real(v(4)) imag(v(4));
    real(v(5)) imag(v(5))];
y=ybus;

'LES PERTES ACTIVES',[S]=pertes(x,y),pause
'NOMBRE D"ITERATION',indx,pause
'TENSION FINALE',abs(v)',pause
'DEPHASAGE DE LA TENSION [DEGRE]',angle(v)'/d2r,pause
'PUISSANCE ACTIVE [MW]',(real(s)+[0 0 0 0 0.24])*100,pause
'PUISSANCE REACTIVE [MVar]',(-imag(s)+[0 0 0 0 0.11])*100
```

**ANNEXE B : CALCUL DES PERTES ACTIVES TOTALES**

```
% 'CALCUL DES PERTES ACTIVES'
function [S] = loss(v,v1,y)
%UNTITLED Summary of this function goes here
% Detailed explanation goes here
v=abs(v)
for i=1:1:5
    for k=1:1:5
        if i>k
            f(i,k)=(v(k)^2+v(i)^2-2*v(k)*v(i)*cos(angle(v1(i))-
angle(v1(k))))*y(i,k);
        else
            f(i,k)=0;
        end
    end
end
s1=sum(f);
S=sum(s1);
end
```

**ANNEXE C : DETERMINATION DE LA MATRICE D'ADMITTANCE**

```
'CALCUL DE LA MATRICE D"ADMITTANCE'
function [ybus]=ybus(zz,ych)

for i=1:5
    for k=1:5
        if zz(i,k) == 0
            ybus(i,k)=0;
        else ybus(i,k)=-1/zz(i,k);
        end
    end
end
for i=1:5
    ysum=0;
    csum=0;
    for k=1:5
        ysum=ysum+ybus(i,k);
        csum=csum+ych(i,k);
    end
    ybus(i,i)=csum-ysum;
end
y=ybus;
end
```

**ANNEXE D : OPTIMISATION PAR LA METHODE DE RECUIT SIMULE**

```
'OPTIMISATION PAR LA METHODE DE RECUIT SIMULE'  
clear all  
clc  
  
%Nombre de bus P-Q  
np=input('nombre de BUS de type P-Q = ');  
%Nombre de bus P-V  
ng=input('nombre de BUS de type P-V = ');  
%Nombre de bus total  
n=np+ng+1;  
  
%matrice zz  
for i=1:1:n  
    for j=1:1:n  
        disp(['zz(',num2str(i),',';','num2str(j),')=']);  
        zz(i,j)=input('');  
    end  
end  
  
%Matrice ych  
for i=1:1:n  
    for j=1:1:n  
        disp(['ych(',num2str(i),',';','num2str(j),')=']);  
        ych(i,j)=input('');  
    end  
end  
  
%Matrice d'admittance  
[ybus]=ybus(zz,ych);  
y=-real(ybus);  
%Données obtenues à partir du calcul de la répartition de puissance  
v=[1.05;1;1;1;1.02];  
[v1]=gs(v,ybus);  
vmin=v1;  
  
%initialisation de T  
T=1;  
%initialisation de p  
p=0.9;  
%initialisation de Smin  
[S1]=loss(v1,y);  
Smin=S1;  
S2=0;  
m=0;  
  
while S1>=S2  
    if m==0
```

```
        alea=0.01*[0.1 0 0 0 0.16];
    else
        alea=0.01*[0.1 0 0 0 0.165];
    end
    v2=v1+alea;
    [v2]=gs(v2',ybus);

%generation de x1 et x2
[S1]=loss(v1,y);
[S2]=loss(v2,y);
diff=S2-S1;
if diff<=0
    v1=v2;
    if S2<=Smin
        vmin=v1;
        Smin=S2;
    end
else
    prob=exp(-diff/T);
    if prob>p
        v1=v2;
    else
        %pause
    end
end
end
T=1/m;
m=m+1;
courbe(m)=S1*100;
Smin;
end
v1
v2
v3=vmin';
% Calcul de P et Q
for i=1:5
    sm=0;
    for k=1:5
        sm=sm+ybus(i,k)*v3(k);
    end
    s(i)=conj(v3(i))*sm;
end
s
m=[1:1:m];
plot(m,courbe(m)),grid
xlabel('Itérations') , ylabel('Pretes actives totales [MW]')
abs(vmin)
angle(vmin)
'LES PERTES ACTIVES TOTALES', Smin
'LES PERTES REACTIVES TOTALES', Qloss=sum(imag(s))
```

Titre : « SYNTHÈSE DES MÉTHODES D'OPTIMISATIONS – APPLICATION A L'OPTIMISATION DE LA PUISSANCE ACTIVE DANS LE RESEAU ELECTRIQUE PAR LA METHODE DE RECUIT SIMULE»

Auteur : RAMBOAMAMPIANINA Samitiana

E-mail : [rsamitiana@gmail.com](mailto:rsamitiana@gmail.com)

Téléphone : (+261) 33 17 965 66 – (+261) 34 03 450 90

## **RESUME**

Ce mémoire intitulé « Synthèse des méthodes d'optimisations – Application à l'optimisation de la puissance active dans le réseau électrique. » a pour principal objectif la maîtrise des toutes les méthodes d'optimisation. Pour cela, des recherches ont été menées sur le concept de l'optimisation en essayant de mettre en exergue la formulation d'un problème d'optimisation et la distinction du problème d'optimisation. Pour la résolution du problème, on a développé les hypothèses d'utilisation et le principe de chaque méthode qui sont généralement classé en 2 grandes groupes, les méthodes classiques qui sont des méthodes déterministes permettant de résoudre mathématiquement les problèmes et les méthodes modernes qui sont des méthodes stochastiques fournissant une solution globale. Pour appliquer ces théories, on s'est proposé d'optimiser les pertes de puissance actives dans un réseau électrique test à 5 nœuds par la méthode de recuit simulé. Le résultat obtenu nous a montré la possibilité de réduire les pertes actives dans un réseau électrique.

Mots clés : Optimisation, méthode stochastique, méthode déterministe, Calcul de la répartition de puissance, optimisation de la puissance active

## **ABSTRACT**

This memory entitled "Synthesis of the optimizations methods – Application to the optimization of the real power in an electrical network." Is a main objective for the master of all the methods of optimization. For that, research was undertaken on the concept of optimization while trying to highlight the formulation of a problem of optimization and the distinction of the problem of optimization. For the resolution of the problem, we are developed the hypothesis of use and the principle of each method which are generally classified in 2 large groups, the methods traditional which are deterministic methods making it possible to solve the problems mathematically and the modern methods which are stochastic methods providing a global solution. To apply these theories, we are proposed to optimize the real power losses in an electrical supply network test with 5 bus by the method of simulated annealing. The result obtained showed us the possibility of reducing the real power losses in an electrical supply network.

Key words: Optimization, stochastic method, deterministic method, load flow calculus, real power optimization