

**REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE**

**MINISTERE DE L'ENSEIGNEMENT SUPERIEUR  
ET DE LA RECHERCHE SCIENTIFIQUE**

**UNIVERSITE DJILLALI LIABES  
SIDI BEL ABBES**

Faculté Des Sciences De L'ingénieur  
Département d'Electrotechnique

Thèse présentée par :  
**NAAMA BAKHTA**

**Pour l'obtention du diplôme de :  
Docteur Es Science En Electrotechnique**

Intitulée :

**CONTRIBUTION A L'EVALUATION ET AU  
PERFECTIONNEMENT DES METHODES META HEURISTIQUES  
D'OPTIMISATION COMBINATOIRE. APPLICATION A  
L'OPTIMISATION DES PUISSANCES ACTIVES D'UN RESEAU  
D'ENERGIE ELECTRIQUE**

---

Acceptée sur proposition du jury :

Mr S. HADJERI	Professeur, UDL, SBA	PRESIDENT
Mr A. CHAKER	Professeur, ENSET, ORAN	EXAMINATEUR
Mr M. RAHLI	Professeur, USTO, ORAN	EXAMINATEUR
Mme FZ. GHARBI	Maitre de conférences, UDL, SBA	EXAMINATEUR
Mr H. BOUZEBODJA	Maitre de conférences, USTO, ORAN	RAPPORTEUR
Mr Y. RAMDANI	Professeur, UDL, SBA	CO-ENCADREUR

## Remerciements

Je tiens tout d'abord à remercier Mr BOUZEBOUDJA HAMID, mon directeur de thèse, pour m'avoir accueillie encadrée, soutenue en prodigué de nombreux conseils tout au long de ces années de thèse. Je lui suis particulièrement reconnaissante de m'avoir laissée une grande liberté scientifique, ce qui m'a permis de recevoir une initiation idéale et privilège de la recherche.

Je remercie vivement Mr RAMDANI YOUCEF, co-encadreur de cette thèse, qui m'a beaucoup appris et beaucoup apportée, sur le plan scientifique. Qu'il m'a donnée le courage de faire plus sans aucun doute.

J'adresse mes sincères remerciement au président de jury Mr S. HADJERI, et les membres Mr A. CHAKER, Mr M. RAHLI et Mme FZ. GHARBI qui ont accepté de participer au jury en tant qu'examineurs.

Je suis reconnaissante envers mes collègues d'université d'USTO pour les bons moments. Mes remerciements vont aussi à tous les collègues de l'université de Sidi Bel Abbes, université de Mascara. Je remercie particulièrement les personnes qui m'ont permis de résoudre des problèmes administratifs.

Un grand remerciement à monsieur LAHDEB MOHAMED de l'université de Languate pour m'avoir fait confiance, pour ces précieuses aides et remarques.

J'aimerais remercier du fond de mon cœur mes parents pour leur soutien moral, ma famille qui a toujours porté un intérêt à ce que je faisais.

*Je dédie cette thèse à ma famille, mes collègues et à ma petite la plus précieuse khadidja.*

*J'ai trouvé en toi une fée qui illumine mon cœur.*

NAAMA BAKHTA.

**SOMMAIRE**

**Chapitre 1- Introduction Générale**

1-1) Etat de la recherche..... 01  
1-2) Contribution de ce travail..... 02  
1-3) Plan de la thèse..... 03

**Chapitre 2- Ecoulement Statique des Charges**

2-1) Introduction..... 05  
2-2) Principe de fonctionnement d'un réseau électrique..... 06  
2-3) Réglages de la fréquence et de la puissance..... 08  
    2-3-1/ Réglage primaire..... 09  
    2-3-2/ Réglage secondaire..... 10  
    2-3-3/ Réglage tertiaire..... 11  
2-4) Formulation du problème ..... 11  
2-5) Analyse de l'écoulement des puissances..... 14  
2-6) Modélisation du réseau..... 15  
2-7) Détermination de la matrice admittance..... 16  
2-8) Détermination des courants..... 18  
2-9) Détermination des puissances..... 19  
  
2-10) Caractéristique des équations d'écoulement statique des charges..... 19  
2-11) Principe de base de la solution d'écoulement statique des charges..... 20  
2-12) Méthode numérique de l'écoulement statique des charges..... 20  
    2-12-1/ Méthode de Gauss Seidel..... 20  
    2-12-2/ Algorithme de Gauss Seidel..... 21  
2-13) Ecoulement optimal des charges (optimal load flow)..... 22  
2-14) Conclusion..... 23

**Chapitre 3- Les Méthodes d'Optimisation**

3-1) Introduction..... 24  
3-2) Optimisation continue..... 25  
    3-2-1/ Programmation non linéaire..... 25  
        3-2-1-a) Programmation non linéaire avec contraintes..... 25  
            3-2-1-a-1/ Méthode des coûts marginaux..... 25  
    3-2-2/ Programmation non linéaire sans contraintes..... 28  
        3-2-2-1/ Méthode de Gradient..... 28  
        3-2-2-2/ Méthode de Gradient Conjugue..... 28  
        3-2-2-3/ Méthode de Newton..... 29  
        3-2-2-4/ Méthode de Quasi-Newton..... 30  
        3-2-2-5/ Algorithme de Davidon-Fletcher-Powel..... 31  
        3-2-2-6/ Algorithme de Broydon- Fletcher- Goldfarb- Shanno..... 32  
3-3) Programmation linéaire..... 32  
    3-3-1/ Méthode de simplexe..... 32  
3-4) Méthode de pénalité..... 35  
3-5) Optimisation par métaheuristiques..... 38  
3-6) Conclusion..... 38

## Chapitre 4- Métaheuristiques pour l'Optimisation Combinatoire et L'affectation Sous Contraintes

4-1) Introduction.....	39
4-2) Définition de l'optimisation combinatoire .....	39
4-3) Recherche locale et recherche globale.....	40
4-4) Classification des métaheuristiques .....	41
4-5) Métaheuristiques.....	43
4-5-1/ Méthodes de voisinages .....	43
4-5-1-1/ Recuit Simule .....	44
4-5-1-2/ Recherche Tabou .....	46
4-5-1-2-1/ Critère d'aspiration.....	48
4-5-1-2-2/ Intensification.....	49
4-5-1-2-3/ diversification.....	49
4-5-1-2-4/ Exemple d'optimisation des fonctions par la Recherche Tabou.....	49
4-5-2/ Algorithmes évolutifs .....	52
4-5-2-1/ Algorithmes Génétiques.....	53
4-5-2-1-1/ Codage des individus.....	54
4-5-2-1-2/ Opérateurs génétiques.....	55
4-5-2-1-3/ Exemple d'optimisation des fonctions par un Algorithme Génétique.....	59
4-5-2-2/ Programmation évolutive .....	61
4-5-2-3/ Stratégies d'évolution.....	62
4-5-3/ Les algorithmes de colonies de fourmi.....	62
4-6) Avantages et inconvénients des méthodes métaheuristiques.....	64
4-7) Les algorithmes hybrides.....	65
4-8) Quelle métaheuristiques utilisées ?.....	66
4-9) Conclusion.....	66

## Chapitre 5- Application des Algorithmes Génétiques dans la Répartition Optimale des Puissances Actives

5-1) Introduction.....	67
5-2) Présentation du problème .....	67
5-3) Codage binaire .....	68
5-3-a/ <i>Création de la population initiale</i> .....	69
5-3-b/ <i>Fonction d'évaluation</i> .....	69
5-3-c/ <i>opérateur de sélection</i> .....	69
5-3-d/ <i>opérateur de croisement</i> .....	69
5-3-e) <i>opérateur de mutation</i> .....	69
5-4) Codage réel.....	70
5-4-a/ <i>opérateur de croisement</i> .....	71
5-4-b/ <i>opérateur de mutation</i> .....	71
5-5) Illustration .....	72
5-6) Conclusion.....	82



**Chapitre 6-Application de la Recherche Tabou dans la Répartition Optimale des Puissance Actives**

6-1) Introduction.....	83
6-2) Organigramme de la recherche tabou.....	84
6-3) Illustration .....	85
6-4) Conclusion.....	89

**Chapitre 7- Les Algorithmes Hybrides**

7-1) Introduction.....	90
7-2) Hybridation de l'Algorithme Génétique avec QN.....	90
7-3) Illustration .....	92
7-4) Hybridation de l'AG avec la Recherche Tabou.....	96
7-5) Hybridation de la méthode RT avec QN.....	100
7-6) Conclusion.....	103

<b>CONCLUSION GENERALE</b> .....	104
<b>BIBLIOGRAPHIE</b> .....	106
<b>ANNEXE</b> .....	111

**LISTE DES FIGURES**

**Figure 2.1.** Principe de fonctionnement d'un réseau électrique

**Figure 2.2.** Caractéristique de régulateur primaire

**Figure 2.3.** Caractéristique de régulateur secondaire

**Figure 2.4.** Fonction coût linéaire par morceaux

**Figure 2.5.** Le réseau électrique sous la forme simplifié

**Figure 3.1.** Les méthodes d'optimisation

**Figure 4.1.** Exemple d'une fonction objectif à une variable.

**Figure 4.2.** Classement des méthodes de résolution

**Figure 4.3.** Principe des méthodes heuristiques

**Figure 4.4.** Principe de base du recuit simulé

**Figure 4.5.** L'algorithme du recuit simulé

**Figure 4.6.** Principe de base de la recherche tabou

**Figure 4.7.** L'algorithme de la recherche tabou

**Figure 4.8.** Graphe de la fonction  $F_2$

**Figure 4.9.** Graphe de la fonction  $F_3$

**Figure 4.10.** Graphe de la fonction  $F_4$

**Figure 4.11.** Schéma de principe des algorithmes évolutionnaires

**Figure 4.12.** L'algorithme génétique de base.

**Figure 4.13.** Technique du croisement en 1 point

**Figure 4.14.** Technique du croisement en 2 points

**Figure 4.15.** Technique de croisement uniforme

**Figure 4.15.** Technique de croisement uniforme

**Figure 4.16.** Technique de mutation

**Figure 4.17.** Graphe de la fonction  $F_1$

**Figure 4.18.** Graphe de la fonction  $F_2$

**Figure 4.19.** Graphe de la fonction  $F_3$

**Figure 4.20.** Graphe de la fonction  $F_4$

**Figure 4.21.** Exemple du mécanisme de fourmi

**Figure 5.1.** Organigramme de l'algorithme génétique / (Codage binaire)

**Figure 5.2.** Organigramme de l'algorithme génétique / (Codage réel)

**Figure 5.3.** Coût en fonction du nombre d'itérations / Réseau IEEE 5 / (codage binaire)

**Figure 5.4.** Coût en fonction du nombre d'itérations/ Réseau IEEE-5 / (codage réel)

**Figure 5.5.** Coût en fonction du nombre d'itérations / Réseau IEEE-30 / (codage binaire)

**Figure 5.6.** Coût en fonction du nombre d'itérations / Réseau IEEE-30 / (codage réel)

**Figure 5.7.** Coût en fonction du nombre d'itérations/ Réseau IEEE-57/ (codage binaire)

**Figure 5.8.** Coût en fonction du nombre d'itérations/ Réseau IEEE-57 / (codage réel)

**Figure 6.1.** Schéma tabou

**Figure 6.2.** Organigramme de la recherche tabou

**Figure 6.3.** Courbe du coût total/ Réseau IEEE-5

**Figure 6.4.** Courbe du coût total / Réseau IEEE-30

**Figure 6.5.** Courbe du coût total / Réseau IEEE-57

**Figure 7.1.** Organigramme de l'hybridation d'un AG avec QN

**Figure 7.2.** Coût en fonction du nombre d'itérations/ Réseau IEEE-5/

**Figure 7.3.** Coût en fonction du nombre d'itérations/ Réseau IEEE-30/

**Figure 7.4.** Coût en fonction du nombre d'itérations/ Réseau IEEE-57/

**Figure 7.6.** Organigramme de l'algorithme génétique hybride avec la recherche tabou

**Figure 7.7** Coût en fonction du nombre d'itérations / Réseau IEEE-5/

**Figure 7.8.** Coût en fonction du nombre d'itérations / Réseau IEEE-30/

**Figure 7.9.** Coût en fonction du nombre d'itérations/ Réseau IEEE-57/

**Figure 7.10.** Organigramme de l'hybridation RT-QN

**LISTE DES TABLEAUX**

**Tableau 2.1.** Les nœuds dans un réseau électrique.

**Tableau 4.1.** Résultats d'optimisation des quatre fonctions

**Tableau 5.1.** Résultats d'un AG/ Réseau IEEE 5/ (codage binaire)

**Tableau 5.2.** Résultats d'un AG/ Réseau IEEE 5/ (codage réel)

**Tableau 5.3.** Résultats d'un AG / Réseau IEEE-30 / (codage binaire)

**Tableau 5.4.** Résultats d'un AG / Réseau IEEE-30 / (codage réel)

**Tableau 5.5.** Résultats d'un AG / Réseau IEEE-57 / (codage binaire)

**Tableau 5.6.** Résultats d'un AG / Réseau IEEE-57 / (codage réel)

**Tableau 5.7.** Comparaison entre l'AG et QN / Réseau IEEE-5/

**Tableau 5.8.** Comparaison entre l'AG et QN / Réseau IEEE-30/

**Tableau 5.9.** Comparaison entre l'AG et QN / Réseau IEEE-57/

**Tableau 6.1.** Résultats de RT /Réseau IEEE-5/

**Tableau 6.2.** Résultats de RT /Réseau IEEE-30/

**Tableau 6.3.** Résultats de RT /Réseau IEEE-57/

**Tableau 6.4.** Comparaison entre RT, QN et AG / Réseau IEEE-5 /

**Tableau 6.5.** Comparaison entre RT, QN et AG / Réseau IEEE-30/

**Tableau 6.6.** Comparaison entre RT, QN et AG / Réseau IEEE-57 /

**Tableau 7.1** Hybridation AG-QN/ Réseau IEEE-5/

**Tableau 7.2** Hybridation AG-QN/ Réseau IEEE-30/

**Tableau 7.3** Hybridation AG-QN/Réseau IEEE-57/

**Tableau 7.4** Hybridation AG-RT /Réseau IEEE-5/

**Tableau 7.5** Hybridation AG -RT /Réseau IEEE-30/

**Tableau 7.6** Hybridation AG-RT /Réseau IEEE-57/

**Tableau 7.8.** Hybridation RT-QN /Réseau IEEE-5/

**Tableau 7.9** Hybridation RT-QN /Réseau IEEE-30 /

**Tableau 7.10** Hybridation RT-QN /Réseau IEEE-57/

**Tableau 7.11** Résultats du réseau IEEE-5

**Tableau 7.12** Résultats du réseau IEEE-30

**Tableau 7.13** Résultats du réseau IEEE-57

***CHAPITRE I***  
***INTRODUCTION GENERALE***

## INTRODUCTION GENERALE

### 1-1) ETAT DE RECHERCHE

L'énergie électrique occupe une place très importante dans les branches de l'économie moderne et de la vie courante. La consommation de l'énergie électrique augmente considérablement. Il est admis d'une manière générale, que depuis le début du dix-neuvième siècle l'énergie électrique consommée dans le monde double en moyenne tous les dix ans. Le rôle des systèmes d'énergie électriques est de fournir aux utilisateurs le produit électricité au moindre coût dans des conditions de qualité et de sécurité satisfaisantes.

La demande croissante sur l'énergie électrique a conduit à l'extension et à la complexité des réseaux électriques, ce qui a poussé les chercheurs dans ce domaine à opter pour un choix convenable des meilleures solutions possibles.

Pour une bonne exploitation du réseau, nous devons résoudre les problèmes d'ordre techniques et économiques, ce qui nécessite l'amélioration de la gestion de l'énergie électrique en réduisant d'une part le coût de production et d'autre part en gardant l'équilibre entre la production et la consommation.

Pour réaliser ces objectifs dans le secteur de l'électricité, les chercheurs n'ont pas cessé de développer des algorithmes qui permettent de programmer la production des centrales électriques d'une manière optimale. Ces algorithmes basés sur des méthodes mathématiques d'optimisation sont continuellement améliorés dans le but d'augmenter leurs performances afin qu'ils soient exploités d'une façon instantanée au niveau des dispatchings. Vers la fin des années 60, deux méthodes ont été introduites, soit la méthode du hessien [SAS 68] et celle du gradient réduit [DOM 68, PES 68]. Dans ces deux méthodes, les contraintes de sécurité sont prises en compte à l'aide de pénalités incluses dans la fonction objectif. Parallèlement est apparue une autre famille de méthodes basées sur la programmation linéaire et la programmation non linéaire. L'inconvénient de ces méthodes est le risque de convergence vers un optimum local surtout si la fonction objective est non- linéaire ou bien ses dérivées sont difficiles à calculer.

Le rôle primordial de toute entreprise de production d'énergie électrique est d'assurer en tout temps et en tout lieu la couverture des puissances actives et réactives demandées par les utilisateurs, d'offrir une sécurité d'alimentation élevée, et de garantir une qualité acceptable de l'énergie livrée avec un coût aussi faible que possible. Afin de respecter ces exigences, il est nécessaire de prendre un certain nombre de décisions, dont l'échelonnement temporel va de seconde aux quelques minutes.

La première décision est de garder l'équilibre instantané entre la production et la consommation et cela pour un réglage primaire qui est le réglage de puissance active, ensuite par un réglage secondaire qui fait intervenir les puissances échangées avec les réseaux voisins.

A l'échelle de quelques minutes, on introduit un réglage dit tertiaire ou *dispatching économique*, qui tient compte des coûts de production et des contraintes de sécurité (tension des nœuds et transits sur les branches). Il s'agit donc d'une optimisation en régime permanent qui est reprise à intervalles réguliers plusieurs fois par heure, en fonction de la variation des charges [PIE 83].

La formulation générale du problème de la répartition optimale des puissances actives et réactives (optimal power flow « OPF ») est caractérisé par la minimisation de la fonction coût (fonction objectif) en tenant compte de certaines contraintes.

Beaucoup d'algorithmes d'OPF ont été développés pour résoudre ce problème. Chacun d'eux a ses propres caractéristiques favorables, telles que le temps de la solution et les propriétés de la convergence [ARV 71, TSA 71, RAH 85, RAH 95 a, BOU 04, BOU 05 b ...]. Au début des années 80, c'est la naissance des nouvelles méthodes dites les *métaheuristiques d'optimisation combinatoire*.

En optimisation combinatoire, de nombreux problèmes s'avèrent souvent difficiles à résoudre de manière exacte. Ceci n'est pas du à un manque de connaissance mathématique, mais plutôt à des problèmes techniques. En effet, la résolution d'un problème dans lequel on considère des instances de taille comparable à celles rencontrées dans la pratique conduit souvent à se heurter à des problèmes de taille mémoire et de temps de calcul trop importants. De sorte à contre ces obstacles, il faut recourir à des méthodes d'approximation de la solution. On ne cherchera plus à obtenir forcément la meilleure solution mais plutôt une solution de bonne qualité obtenue en un temps minimal. Deux principales classes d'algorithmes répondent à ce besoin :

- ✓ Les algorithmes de type constructif qui construisent leur solution à partir d'un élément sans jamais remettre en question les choix passés. Ces algorithmes fournissent souvent des solutions de mauvaise qualité.
- ✓ Les méthodes d'exploration locale qui partent d'une solution réalisable cherchent à l'améliorer pas à pas. Ces méthodes restent coûteuses en termes de ressources informatiques.

Cependant, ces deux approches présentent le défaut de ne pas pouvoir progresser au delà du premier optimum local rencontré. Ainsi, de nouvelles métaheuristiques ont été créées pour y faire face : Recuit Simulé, Algorithmes Génétiques, la Recherche Tabou, ...etc. Une nouvelle technique d'optimisation devenue aujourd'hui un enjeu important, c'est l'hybridation, à tel point que toutes les différentes métaheuristiques que nous avons citées, sont maintenant perçues comme des points de départ au développement de nouveaux algorithmes d'optimisation.



## 1-2) CONTRIBUTIONS DE CE TRAVAIL

Ce travail est une contribution à la résolution du problème de la répartition optimale des puissances actives dans un réseau d'énergie électrique par les métaheuristiques d'optimisation combinatoire.

Dans un premier temps, nous appliquons la méthode de l'intelligence artificielle à l'aide des Algorithmes Génétiques pour la minimisation du coût de production de l'énergie électrique par un codage binaire naturel puis par un codage réel. Le choix du type de codage est un paramètre primordial car l'efficacité d'un Algorithme Génétique dépend principalement de la manière dont elle capte les régularités de l'espace de recherche. Les opérateurs génétiques (sélection, croisement, mutation) représentent les paramètres les plus importants pour l'évaluation de la fonction objective. Le choix de la probabilité de croisement et de mutation est un facteur typique qui permet d'évaluer la fonction objective.

Ensuite, nous utilisons une deuxième méthode métaheuristique d'optimisation combinatoire pour la minimisation du coût de production de l'énergie électrique c'est la Recherche Tabou. A l'inverse des algorithmes génétiques cette méthode utilise un point initial au lieu d'un ensemble des points (population). La Recherche Tabou utilise une mémoire de stockage des solutions courantes dans une liste de longueur L données, appelée liste tabou. Le critère d'aspiration consiste à tester si la solution produite de statut tabou présente un coût inférieur à celui de la meilleure solution trouvée. Le critère d'intensification consiste à approfondir la recherche dans certaines régions du domaine, identifiées comme susceptible de contenir un optimum global. Le critère de diversification permet à l'algorithme de bien explorer l'espace des solutions, et d'éviter que le processus de recherche ne soit trop localisé et laissé de grandes régions du domaine totalement inexplorées.

Les critères de diversification et d'intensification sont des concepts complémentaires, qui enrichissent la méthode de Recherche Tabou et la rendent plus robuste et plus efficace.

Dans le but de construire un algorithme hybride robuste, efficace et réalisable nous présentons une hybridation entre : un Algorithme Génétique simple et une méthode analytique quasi-Newtonienne, un Algorithme Génétique simple et la Recherche Tabou et enfin entre la Recherche Tabou et une méthode analytique quasi-Newtonienne.

## 1-3) PLAN DE LA THESE

Cette thèse présente une étude générale sur la répartition optimale des puissances actives dans un réseau d'énergie électrique d'une part, d'autre part présente une application des méthodes d'optimisations combinatoires pour la minimisation du coût de production de l'énergie électrique comparées avec des méthodes analytiques.

L'organisation générale de cette thèse est articulée autour de 7 chapitres

Dans le chapitre 1 nous présentons une introduction générale

Le chapitre 2 présente la répartition optimale des puissances actives avec toutes les contraintes de sécurité et la formulation mathématique de la fonction coût (fonction objective).

Le chapitre 3 expose les différentes méthodes d'optimisation avec leurs algorithmes.

Le chapitre 4 se présente comme une référence des méthodes métaheuristiques d'optimisation combinatoires.

Le chapitre 5 est consacré à l'application des Algorithmes Génétiques dans la répartition optimale des puissances actives dans un réseau électrique. Une simulation a été faite sur des réseaux à 5, 30 et 57 nœuds.

Le chapitre 6 présente l'application d'une méthode métaheuristique qui est la Recherche Tabou pour minimiser le coût de production.

Le chapitre 7 propose une hybridation entre les méthodes métaheuristiques de base et les métaheuristiques hybridées avec les méthodes de quasi-Newton.

En fin nous clôturons ce travail par une conclusion générale.

***CHAPITRE II***  
***L'ÉCOULEMENT STATIQUE DES***  
***CHARGES***

## ÉCOULEMENT STATIQUE DES CHARGES

### 2-1) INTRODUCTION

L'un des états les plus importants d'un réseau électrique est son mode d'opération en régime permanent. Afin d'obtenir des informations sur ces points d'opération et d'être en mesure de les gérer pour des raisons de sécurité, de fiabilité ou d'économie, on introduit dans ce chapitre le concept général de la modélisation des systèmes électriques en traitant deux champs:

- l'écoulement statique des charges ou de puissance
- l'écoulement optimal des charges.

Les calculs d'écoulement statique des charges fournissent les écoulements de puissance et les profils de tensions (amplitude et phase) reliés aux conditions d'opération des barres (nœuds) et des lignes du système en régime permanent. A chaque barre du réseau sont associés quatre paramètres: puissances active et réactive, l'amplitude et la phase de tension.

Les considérations premières pour le développement d'un programme d'ordinateur sont :

- la formulation d'une description mathématique du problème
- l'utilisation d'une méthode numérique de solution.

La formulation mathématique (modélisation) résulte en un groupe d'équations non linéaires reliant les variables aux éléments d'une matrice caractérisant la physique du système. Ces équations sont établies en utilisant les relations de nœud ou de boucle. Leurs coefficients dépendent de la sélection du courant ou de la tension comme variable de base, alors une matrice d'admittances ou d'impédances est utilisée.

La solution du système d'équations décrivant le modèle est basée sur une méthode numérique itérative à cause de leurs non linéarités. Elle doit satisfaire les lois de Kirchoff dont l'une ou l'autre est utilisée comme test de convergence. Les contraintes introduites dans la solution sont les limites physiques des génératrices et les tolérances permises sur les amplitudes et les phases des tensions de barre du réseau. Différentes techniques de solution sont développées afin de faire ressortir la particularité de la formulation. Ce sont:

- la méthode de Gauss
- la méthode de Gauss-Seidel
- la méthode de Newton-Raphson

Puisque la démarche vise à contrôler le profil des tensions de barre, donc gérer un état de fonctionnement du système, on étudie la théorie qui permet d'atteindre cet objectif à partir de critères déterminés: l'écoulement optimal des charges (optimal load flow). Dans sa formulation générale, c'est un problème d'écoulement des charges dans lequel les coûts

de combustible où d'autres variables de performance regroupées à l'intérieur d'une fonction objective, sont optimisés. Le modèle mathématique doit satisfaire des contraintes d'égalité (équations standard d'écoulement statique des charges) et des contraintes d'inégalité d'ingénierie (limites inférieure et supérieure des tensions et des puissances générées) ou de sécurité (angle maximal de transfert de puissance).

Lorsque les coûts des combustibles sont minimisés, l'algorithme de la répartition optimale sert pour le dispatching économique. Dans ce cas, il détermine les valeurs optimales des puissances actives et réactives de sortie de toutes les génératrices, celles des autres sources de puissances réactives et l'ajustement des prises des transformateurs. Lors de la formulation d'un modèle pour fin d'étude de stabilité, de fiabilité ou de sécurité du système, la fonction objective doit être développée en fonction des variables significatives. Alors, selon l'approche adoptée et compte tenu de la nature des relations obtenues, ce problème d'optimisation est résolu de façon numérique, par des méthodes itératives directes (Gradient, Simplex, etc.) ou indirectes (Programmation quadratique par exemple).

Cependant, avant toute formulation mathématique de l'écoulement statique, on doit comprendre le fonctionnement du réseau et réunir ses éléments constitutifs à l'aide de lois physiques qui gouvernent leur interaction, afin d'en arriver à un modèle efficace.

## 2-2) PRINCIPE DE FONCTIONNEMENT D'UN RESEAU ELECTRIQUE

Le rôle essentiel d'un réseau électrique est de pourvoir les puissances actives et réactives demandées par les équipements variés qui y sont connectés. Les points de production et de consommation sont reliés entre eux par des lignes de transmission. Pour mieux comprendre le principe de fonctionnement, considérons le système de distribution à deux barres de la figure (2-1 a). Les barres sont alimentées par des unités de production qui génèrent respectivement les puissances  $S_{G1}$  et  $S_{G2}$ . Les demandes se définissent à chaque barre comme étant  $S_{D1}$  et  $S_{D2}$ . Les deux barres sont connectées par une ligne de transmission Figure (2-1 b). Celle-ci est représentée par son modèle en  $\pi$  caractérisé par une impédance série,  $Z$ , et deux admittances shunt  $y_{shi}$ . Les deux tensions de barres sont symbolisées respectivement par  $V_1$  et  $V_2$ .

Dans la formulation du modèle, on combine les puissances générées et demandées à chaque barre pour obtenir une puissance nette  $S_i$ , définie comme la différence entre la production et la consommation de puissance de la barre  $i$  en question. Cette puissance nette est considérée comme injectée dans la barre par une "source de puissance de barre" dont le symbole est introduit dans la Figure (2-1 c).

Pour le système à deux barres, ces puissances injectées sont:

$$S_1 = P_1 + jQ_1 = P_{G1} - P_{D1} + j(Q_{G1} - Q_{D1}) \quad (2-1)$$

$$S_2 = P_2 + jQ_2 = P_{G2} - P_{D2} + j(Q_{G2} - Q_{D2}) \quad (2-2)$$

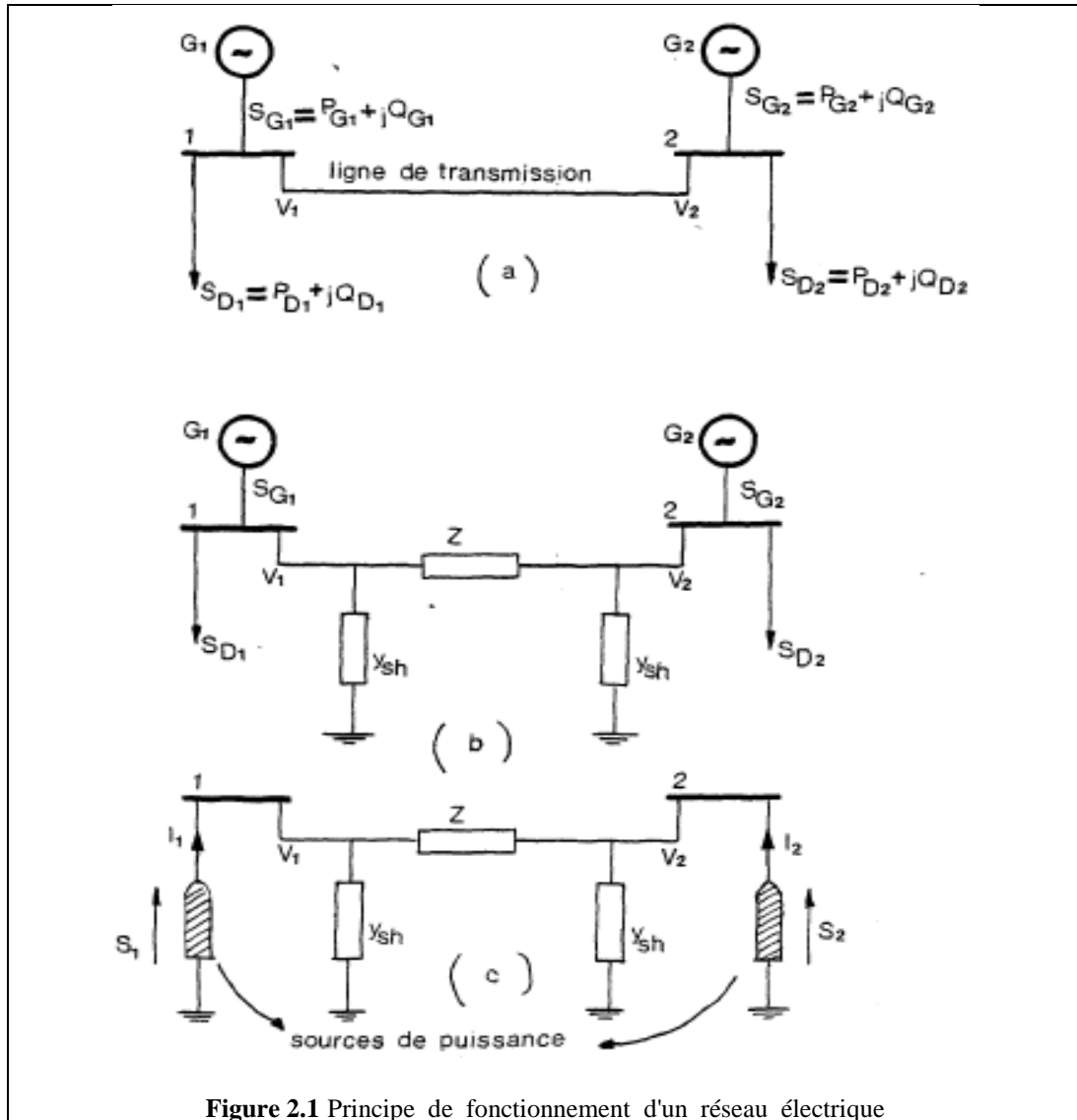


Figure 2.1 Principe de fonctionnement d'un réseau électrique

Le système décrit fonctionne de la manière suivante:

En agissant sur le couple moteur, ce qui peut être accompli à l'aide des régulateurs de chaque turbine, on établit un équilibre entre la puissance active générée, la puissance active demandée plus les pertes actives de transmission. Le critère essentiel pour maintenir un tel équilibre est une fréquence constante.

Par une manipulation du courant de champ de chaque rotor, ainsi que la force électromotrice du stator, on établit un parfait équilibre entre la puissance réactive générée dont une portion provient des lignes, la puissance réactive demandée plus les pertes

réactives. La constance de l'amplitude des tensions de barre est le critère essentiel pour maintenir un tel équilibre.

Le rôle de la ligne de transmission est de fournir un chemin de transfert du surplus de puissance d'une barre pour aller compenser la sur-demande de l'autre et ou de servir comme branche de secours pour les circonstances critiques.

Le circuit de la figure (2-1c) décrit un circuit électrique élémentaire très simplifié par rapport à un réseau ordinaire.

Pour concevoir le modèle du système, on commence par établir une continuité entre le courant équivalent au rapport de la puissance injectée et de la tension de nœud et celui entrant dans les composants de la ligne de transmission. On obtient :

$$\frac{S_1^*}{V_1^*} = V_1 Y_{sh1} + \frac{V_1 - V_2}{Z_{ser}} \quad (2-3)$$

$$\frac{S_2^*}{V_2^*} = V_2 Y_{sh2} + \frac{V_2 - V_1}{Z_{ser}} \quad (2-4)$$

De la théorie des réseaux,  $Y_{sh}$  et  $Z_{ser}$ , paramètres constants des lignes, peuvent être regroupés à l'intérieur des matrices décrivant la constitution physique du système. C'est l'étape initiale de la modélisation.

### 2-3) REGLAGES DE LA FREQUENCE ET DE LA PUISSANCE

La continuité d'un système exige des investissements humains et financiers importants afin de répondre aux exigences suivantes :

- Assurer au client la puissance dont il a besoin
- Fournir une tension stable
- Fournir une fréquence stable
- Fournir l'énergie à un prix acceptable
- Maintenir des normes de sécurité
- Veiller à la protection de l'environnement

La réalisation des quatre premiers points est assurée par le réglage des réseaux. En effet l'énergie électrique n'étant pas stockable pratiquement, la puissance des générateurs doit à chaque instant être exactement adaptée à la consommation, de sorte que, devant toute variation de la consommation, les générateurs doivent réagir afin de maintenir cet équilibre. Les conditions de cette adaptation permanente se présentent de façons différentes, il s'agit de compenser de faibles écarts ou les variations importantes de la consommation.

On adjoint un régulateur de vitesse qui consiste à ne pas laisser la fréquence s'écarte trop loin de la valeur de consigne, en adaptant le plus rapidement possible la puissance produite à la puissance demandée.

### 2-3-1 / Réglage primaire

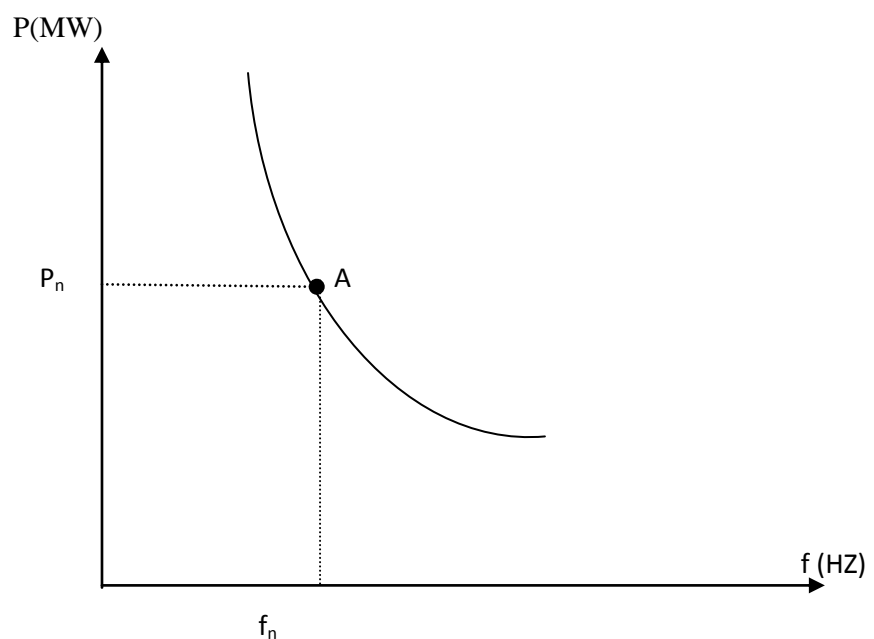
Le réglage primaire concerne la régulation de vitesse, que subit un groupe isolé indépendamment des l'autres.

Il nous permet de donner une caractéristique ayant pour statisme  $\tau = \frac{\Delta w}{\Delta P}$  (figure 2-2) donnant la relation entre la variation de fréquence et la variation de la puissance.

Pour les groupes fonctionnant en parallèle, les variations des puissances sont fonction des statismes des groupes, et des puissances nominales.

Si tous les groupes ont le même statisme, la variation de puissance se répartira proportionnellement aux puissances nominales.

Si ces derniers sont différents, la répartition se fait inversement au statisme.



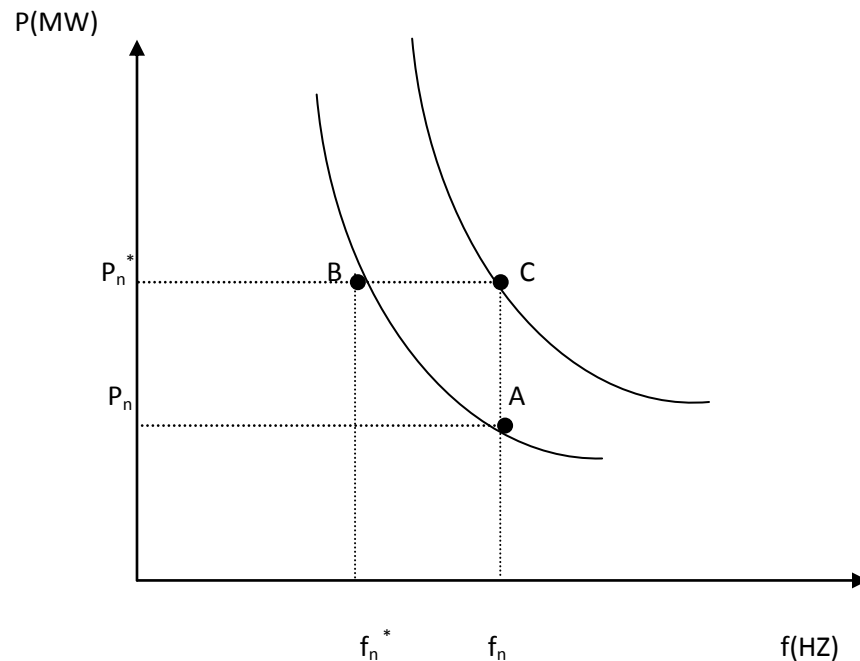
**Figure 2.2.** Caractéristique de régulateur primaire



### 2-3-2 / Réglage secondaire

La liaison tachymètre vannage donne une caractéristique de réglage ayant pour statisme  $\tau = \frac{\Delta w}{\Delta P}$ . Une telle caractéristique ne permet pas de maintenir la fréquence à sa valeur de consigne en cas de surcharge. On adjoint un dispositif de réglage, dit réglage secondaire ou de compensation qui permet de déplacer cette caractéristique de statisme par un mécanisme de changement de vitesse (figure 2-3).

En changeant la valeur de la puissance  $P_n$ , nous observons un changement de la fréquence  $f_n$ . Comme le changement de la fréquence ne doit pas être grand, il va falloir limiter le changement de puissances. Dans ce cas, une partie de cette puissance est prise par un autre régulateur. Une variation consiste à la détermination d'un nouveau point de travail du régulateur, c'est-à-dire fixer la nouvelle valeur de la puissance appelée comme le montre la figure (2-3)



**Figure 2.3.** Caractéristique de régulateur secondaire

Le point A correspondant à l'ancienne puissance désirée, le point B correspond à la nouvelle puissance désirée mais là nous remarquons que la nouvelle fréquence diminue par rapport à la fréquence nominale ce qui est tout à fait normal.

Pour réajuster la fréquence, il va falloir ramener le point B au point C pour que la puissance reste la même et la fréquence sera égale à la fréquence nominale. Cette opération est réalisée par le régulateur secondaire [RAH 95 a].

### 2-3-3 /Réglage tertiaire

Les réglages primaire et secondaire avaient pour but de faire face aux variations aléatoires et non prévisibles des consommations en maintenant ainsi près que possible de leur valeur de consigne :

- D'une part la fréquence du réseau
- D'autre part les échanges avec les réseaux voisins.

Les écarts sont, pour la plupart, aléatoires et de moyenne nulle, mais certains peuvent devenir durables par exemple pour compenser :

- Une variation durable du niveau de consommation.
- La mise hors service d'un groupe générateur à la suite d'un incident
- L'augmentation des pertes dans le réseau résultant de ces deux inconvénients.

Le réglage tertiaire permet une nouvelle répartition économique des puissances entre les groupes en service quand les variations de la charge dépassent les limites d'action du régulateur secondaire.

### 2-4) FORMULATION DU PROBLEME

La gestion optimale des productions actives et réactives est une fonction de plus en plus importante des centres de conduite des réseaux, dans le but d'accroître la sécurité d'alimentation et dans d'exploiter judicieusement les ressources existantes en minimisant les coûts de production et les pertes.

Pour assumer pleinement cette tâche, il est important de bien définir les objectifs qui serviront de critères d'optimisation. On peut dégager les buts suivants :

- Proposer un programme de production qui respecte toutes les contraintes de courants de branche et de tensions, dans un but de sécurité avant toute autre recherche de gain économique (analyse de sécurité) ;
- Minimiser le coût de productions actives et réactives incluant les pertes dans le réseau (dispatching économique) ;
- Réajuster les programmes de production lors de défaillance d'un ou plusieurs éléments du réseau (groupe, ligne ou transformateur), afin de ramener le réseau de l'état d'alerte à l'état sain [LIA 78].

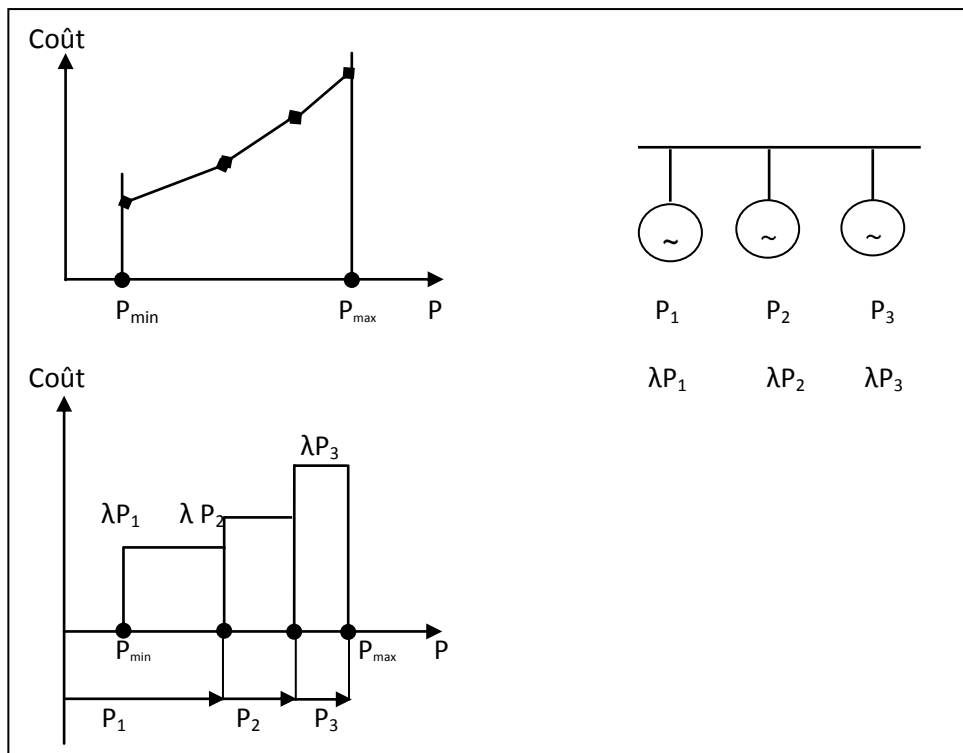
Les objectifs précités peuvent être formulés tels quels pour résoudre les problèmes de planification, soit :

- Retarder le renforcement du réseau par une répartition judicieuse des puissances transitées ;
- Retarder l'investissement de nouveaux moyens de production ;
- Le cas échéant, minimiser le coût des renforcements et des nouveaux moyens de production.

La recherche d'un programme de production qui respecte toutes les contraintes suivant diverses contingences et qui accroît de ce fait la sécurité d'alimentation est sans doute l'objectif primordial. En effet, le coût des conséquences économiques en cas de défaillance est beaucoup plus élevé que le coût de production.

La minimisation du coût de la production active est un problème qui intéresse particulièrement les réseaux à prédominance thermique.

Les coûts absolus ou relatifs des productions actives et réactives ont été admis a priori linéaire et positif en absorption comme en production. Toutefois, on peut les définir librement selon les divers objectifs que l'on souhaite atteindre. La même hypothèse a été faite pour les écarts des réglers des transformateurs par rapport à leur position initiale. Les coûts marginaux et les pénalités sont donc constants, si l'on exclut la correction apportée par les facteurs de pénalité. A toute fin pratique, on peut admettre que la fonction coût de la production active d'une centrale thermique est linéaire, mais on peut toujours approximer une fonction convexe par plusieurs segments de droites (fonction linéaire par morceaux). On associe alors à chaque segment une nouvelle variable (figure ci-dessous), sans relation physique avec les divers groupes d'une centrale.



**Figure 2.4.** Fonction coût linéaire par morceaux

Avec :

P : production active participant au réglage tertiaire.

$\lambda P$  : coût marginal horaire ou pénalité de la production active.

Dans le cas exceptionnel d'une fonction coût non convexe ou discontinue, les méthodes de programmation linéaire ou non linéaire sont inexactes ou inapplicables, et il faut recourir à d'autres méthodes, telles que la programmation dynamique [CHA 79].

Si le coût marginal de la production d'énergie active est une notion bien définie que l'on peut quantifier avec précision (on affecte parfois un coût marginal à la production hydraulique pour valoriser l'énergie accumulée), il n'en est pas de même pour le coût marginal de l'énergie réactive qui est de 10 à 1000 fois inférieur à celui de l'énergie active [MOR 73]. En effet, ce coût représente les pertes actives dues aux transits des puissances réactives dans le réseau, qui sont prises en compte à l'aide de facteurs de pénalité dans la fonction objectif. La même remarque est applicable à l'effet de la variation des rapports de transformation sur les pertes actives. Comme le coût de la production d'énergie réactive est considérablement plus faible que celui de l'énergie active, le gain économique réalisé par son optimisation n'est pas spectaculaire, mais reste appréciable. Par contre, on tire avantageusement parti de la diversité des moyens de production et de la souplesse de réglage en qualité et rapidité.

Les courbes donnant le coût de production de chaque centrale en fonction de la puissance qu'il débite ont été déterminées expérimentalement.

La formation analytique de ces courbes est celle d'un polynôme de degré « n » et qui s'écrit sous la forme suivante :

$$C(P_G) = a_0 + a_1 P_G + a_2 P_G^2 + \dots + a_n P_G^n \quad (2-5)$$

Dans la pratique, la fonction coût se présente sous forme d'un polynôme du deuxième degré [RAH 95 a, CHA 95]:

$$C(P_G) = a_0 + a_1 P_G + a_2 P_G^2 \quad (2-6)$$

Les coefficients de ce polynôme sont calculés par les méthodes d'interpolation de Lagrange, Newton, ... etc.)

La fonction du coût totale de production de l'énergie électrique exprimée en dinars par heure est donnée par l'expression suivante :

$$F = \sum_{i=1}^{NG} C_i(P_{Gi}) \quad (2-7)$$

Les puissances actives doivent être choisies de telle sorte à minimiser la fonction coût de production totale en tenant compte de certaines contraintes.

Le problème peut être posé de la manière suivante :

$$\text{Minimiser } \sum_{i=1}^{NG} C_i(P_{Gi}) \quad (2-8)$$

Sous les contraintes :

$$\sum_{i=1}^{NG} P_{Gi} - \sum_{j=1}^{NC} P_{Chj} - P_L = 0 \quad (2-9)$$

$$P_{Gi}^{\min} \leq P_{Gi} \leq P_{Gi}^{\max} \quad (2-10)$$

Où :

$NG$  : nombre total de nœud producteur ;

$NC$  : nombre total de nœud consommateurs ;

$P_{Gi}$  : puissance active générée au nœud  $i$  ;

$P_{Chj}$  : puissance active consommée au nœud  $j$  ;

$P_{Gi}^{\max}$  : puissance active maximale générée au nœud  $i$  ;

$P_{Gi}^{\min}$  : puissance active minimale générée au nœud  $i$  ;

$P_L$  : les pertes totales actives dans la ligne.

## 2-5) ANALYSE DE L'ÉCOULEMENT DES PUISSANCES

Le calcul de l'écoulement de puissances dit aussi calcul de la répartition des charges (load flow) permet de déterminer :

- Les tensions complexes aux niveaux des différents nœuds ;
- Les puissances transitées d'un nœud à un autre ;
- Les puissances injectées à chaque nœud ;
- Les pertes actives et réactives dans le réseau électrique.

Pour résoudre le problème de l'écoulement de puissances, il existe deux méthodes, l'une dite des mailles, l'autre dite des nœuds. Cette dernière méthode est préférable car elle prend en considération la matrice admittance  $[Y]$ , qui est une matrice creuse, de même elle est facile à introduire les données du problème.

Le développement de l'outil informatique a permis d'élaborer plusieurs méthodes, on peut citer les méthodes de Gauss Seidel et de Newton-Raphson.

## 2-6) MODELISATION DU RESEAU

La résolution du problème de l'écoulement des puissances dans tout système électrique nécessite un modèle mathématique pour calculer les différents paramètres du réseau électrique. [RAH 95 a, LAO]

Soit le réseau électrique donné par la forme simplifié comme montre la figure (2-5).

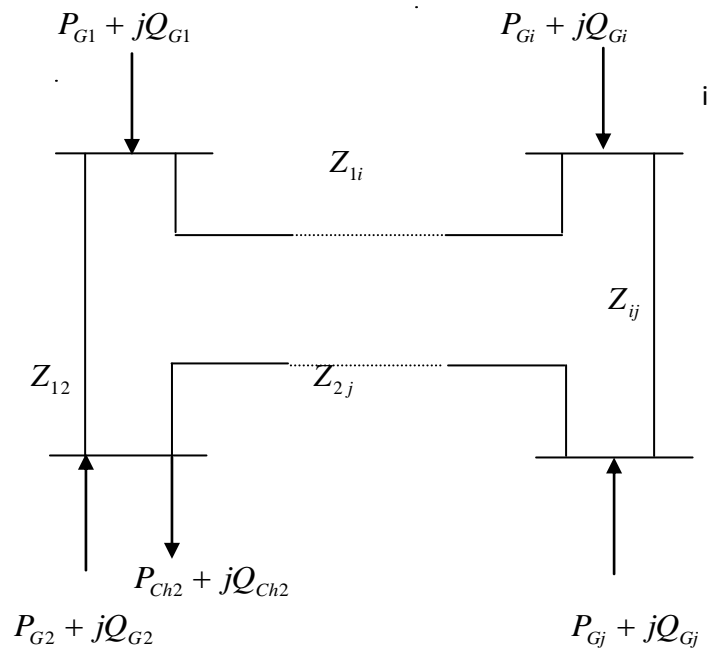


Figure 2.5. Le réseau électrique sous la forme simplifié

En régime permanent, l'équation linéaire du réseau est donnée par :

$$\bar{I} = \bar{Y} \cdot \bar{V} \quad (2-12)$$

Où :

$\bar{I}$  : vecteur complexe des courants injectés au nœud ;

$\bar{Y}$  : matrice admittance complexe ;

$\bar{V}$  : vecteur complexe des tensions en chaque nœud ;

Le tableau ci-dessous résume les différents types de nœuds constituant le réseau électrique.

Type de nœuds	données	inconnues
Nœuds producteurs	$P$ et $ V $	$Q$ et $\delta$
Nœuds consommateurs	$P$ et $Q$	$ V $ et $\delta$
Nœud de bilan	$ V $ et $\delta$	$P$ et $Q$

Tableau 2-1 Les nœuds dans un réseau électrique.

### 2-7) DETERMINATION DE LA MATRICE ADMITTANCE

On considère le schéma d'une branche entre deux nœuds  $i$  et  $j$  :

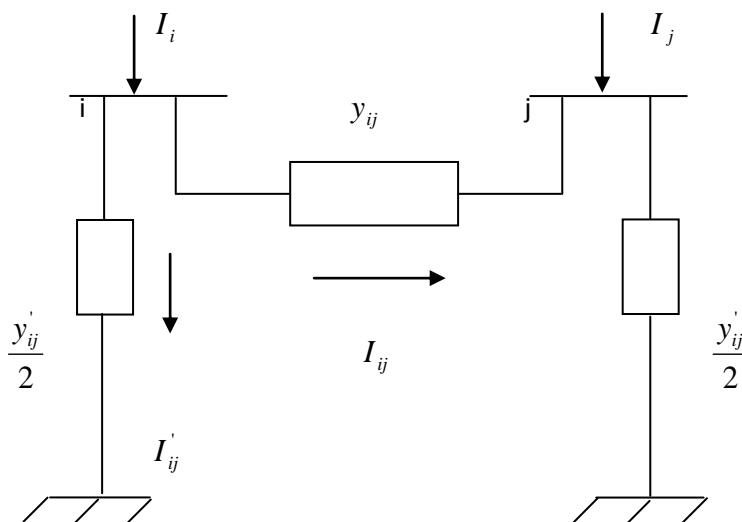


Figure 2.6.

Avec:

$I_i$  : Courant injecté au nœud i ;

$I_{ij}$  : Courant transitant du nœud i au nœud j ;

$I'_{ij}$  : courant de fuite au nœud i ;

$y_{ij}$  : Admittance de ligne i-j ;

$\frac{y'_{ij}}{2}$  : Admittance shunt des nœuds i et j ;

$E_i, E_j$  : Tensions aux nœuds i et j ;

L'utilisation de la méthode des nœuds nécessite la transformation des impédances des branches du réseau en admittances ; pour cela, nous posons :

$$y_{ij} = \frac{1}{Z_{ij}} = \frac{1}{R_{ij} + jX_{ij}} = \frac{R_{ij}}{R_{ij}^2 + X_{ij}^2} - j \frac{X_{ij}}{R_{ij}^2 + X_{ij}^2} \quad (2-13)$$

$$\text{D'où : } y_{ij} = g_{ij} - jb_{ij} \quad (2-14)$$

Avec :

$$\begin{cases} g_{ij} = \frac{R_{ij}}{R_{ij}^2 + X_{ij}^2} \\ b_{ij} = \frac{X_{ij}}{R_{ij}^2 + X_{ij}^2} \end{cases} \quad (2-15)$$

Où :

$g_{ij}$  : appelée conductance

$b_{ij}$  : appelée susceptance

L'admittance propre du nœud i donnée par :

$$Y_{ii} = \sum_{\substack{j=1 \\ j \neq i}}^n (y_{ij} + \frac{y'_{ij}}{2}) \quad (2-16)$$

Et l'admittance mutuelle entre le nœud i et le nœud j :

$$Y_{ij} = -y_{ij} \quad (2-17)$$



## 2-8) DETRMINATION DES COURANTS

Les équations qui régissent le réseau par l'application de la loi des nœuds peuvent être données par la formule suivante :

$$I_i = \sum_{\substack{j=1 \\ i \neq j}}^n (I_{ij} + I'_{ij}) \quad (2-18)$$

L'expression du courant transmit du nœud i vers le nœud j :

$$I_{ij} = \sum_{\substack{j=1 \\ i \neq j}}^n y_{ij} (E_i - E_j) \quad (2-19)$$

L'expression du courant de fuite à la terre :

$$I'_{ij} = \sum_{\substack{j=1 \\ i \neq j}}^n \left( \frac{y'_{ij}}{2} \cdot E_i \right) \quad (2-20)$$

On déduit donc l'expression du courant au nœud i :

$$I_i = \sum_{\substack{j=1 \\ j \neq i}}^n y_{ij} (E_i - E_j) + \sum_{\substack{j=1 \\ j \neq i}}^n \left( \frac{y'_{ij}}{2} \cdot E_i \right) \quad (2-21)$$

D'où

$$I_i = E_i \cdot \sum_{\substack{j=1 \\ j \neq i}}^n \left( y_{ij} + \frac{y'_{ij}}{2} \right) - \sum_{\substack{j=1 \\ j \neq i}}^n y_{ij} \cdot E_j \quad (2-22)$$

On trouve ainsi l'équation générale du courant :

$$I_i = E_i \cdot Y_{ii} + \sum_{\substack{j=1 \\ j \neq i}}^n Y_{ij} \cdot E_j \quad (2-23)$$

D'une façon générale, on aura :

$$I_i = \sum_{j=1}^n Y_{ij} E_j \quad (2-24)$$

D'où la forme matricielle du courant :

$$[I] = [Y][E] \quad (2-25)$$

## 2-9) DETERMINATION DES PUISSANCES

La puissance apparente injectée au nœud  $i$  est donnée par :

$$S_i^* = P_i - jQ_i = E_i^* \cdot I_i \quad (2-26)$$

On remplace l'équation (2-24) dans l'équation (2-26) on aura :

$$S_i^* = P_i - jQ_i = E_i^* \cdot \sum_{j=1}^n Y_{ij} E_j \quad (2-27)$$

Sachant que :

$$\begin{cases} E_i = e_i + jf_j \\ Y_{ij} = G_{ij} - jB_{ij} \\ E_i^* = e_i - jf_i \end{cases} \quad (2-28)$$

Et l'équation de la puissance apparente sera :

$$S_i^* = P_i - jQ_i = (e_i - jf_i) \cdot \sum_{j=1}^n (G_{ij} - jB_{ij})(e_j + jf_j) \quad (2-29)$$

On en déduit les expressions des puissances actives et réactives :

$$\begin{cases} P_i = \left[ \sum_{j=1}^n e_i (e_j G_{ij} + f_j B_{ij}) + f_i (f_j G_{ij} - e_j B_{ij}) \right] \\ Q_i = \left[ \sum_{j=1}^n f_i (e_j G_{ij} + f_j B_{ij}) - e_i (f_j G_{ij} - e_j B_{ij}) \right] \end{cases} \quad (2-30)$$

## 2-10) CARACTERISTIQUE DES EQUATIONS D'EDOULEMENT STATIQUE DES CHARGES

En observant la relation (2-30), on constate que:

1. Les équations sont algébriques, car elles représentent un modèle statique du système, ou un système opérant en régime permanent.
2. Les équations sont non linéaires, donc difficilement résolubles de façon analytique, d'où la nécessité d'utiliser une méthode numérique de solution par ordinateur.
3. Généralement, dans l'analyse des systèmes, les équations relient le courant et la tension, ces équations relient la puissance et la tension.

Par conséquent, il s'agit de réduire le nombre d'inconnues, de  $6N$  à  $2N$  en spécifiant  $4N$  variables afin d'égaliser le nombre d'équations à celui des variables. En principe les  $2N$  variables restantes pourront être déterminées.

## 2-11) PRINCIPE DE BASE DE LA SOLUTION D' ECOULEMENT STATIQUE DES CHARGES

Après avoir classifié les 6 N variables, la solution du système d'équation formé par (2-30) peut être obtenue en procédant comme suit:

Étape 1. A partir de la connaissance de la demande de la clientèle, on possède toutes les informations requises sur les 2 N variables incontrôlables.

Étape 2. On spécifie alors 2 N variables de contrôle; par exemple les puissances générées.

Étape 3. Les 2 N variables qui restent constituent les inconnues. A l'aide de 2 N équations.

## 2-12) METHODE NUMERIQUE DE SOLUTION D' ECOULEMENT STATIQUE DES CHARGES

Pour résoudre les équations d'écoulement statique des charges, un grand nombre de techniques numériques ont déjà été utilisées.

Plusieurs méthodes itératives ont été appliquées, on peut citer : la méthode de Gauss-Seidel, la méthode de Newton-Raphson, la méthode de relaxation...etc.

### 2-12-1/ METHODE DE GAUSS-SEIDEL

Cette méthode permet de résoudre un système d'équations non linéaire en utilisant la matrice admittance, on suppose initialement des tensions pour tous les nœuds excepté le nœud de bilan où la tension maintenue constante.

On peut exprimer les courants pour chaque nœud par la relation suivante :

$$I_i = \frac{S_i^*}{E_i^*} = \frac{P_i - jQ_i}{E_i^*} \quad i=1,2,\dots,n \quad (2-31)$$

Où :

$n$  : désigne le nombre de nœuds dans le réseau ;

$S_i^*$  : le conjugué de la puissance apparente injectée au nœud  $i$  ;

$E_i^*$  : le conjugué de la tension au nœud  $i$  ;

$P_i$  : la puissance active injectée au nœud  $i$  ;

$Q_i$  : la puissance réactive injectée au nœud  $i$  ;

En remplace la valeur du courant dans l'équation (2-31) on aura :

$$I_i = \frac{P_i - jQ_i}{E_i^*} = Y_{ii} \cdot E_i + \sum_{\substack{j=1 \\ j \neq i}}^n Y_{ij} \cdot E_j \quad i \neq s \quad (2-32)$$

L'expression de la tension pour chaque nœud est :

$$E_i = \frac{1}{Y_{ii}} \left[ \frac{P_i - jQ_i}{E_i^*} - \sum_{j=1}^{i-1} Y_{ij} \cdot E_j - \sum_{j=i+1}^n Y_{ij} \cdot E_j \right] \quad (2-33)$$

On pose :

$$\begin{cases} KL_i = \frac{P_i - jQ_i}{Y_{ii}} \\ YL_{ij} = \frac{Y_{ij}}{Y_{ii}} \end{cases} \quad (2-34)$$

D'où l'expression de la tension pour chaque nœud :

$$E_i^{k+1} = \frac{KL_i}{[E_i^k]^*} - \sum_{j=1}^{i-1} YL_{ij} \cdot E_j^{k+1} - \sum_{j=i+1}^n YL_{ij} \cdot E_j^k \quad i \neq s \quad (2-35)$$

Pour accélérer la convergence de la méthode, on introduit un facteur d'accélération  $\alpha$  :

$$E_i^{k+1} = E_i^k + \alpha \cdot \Delta E_i^k \quad (2-36)$$

Avec

$$\Delta E_i^k = E_i^{k+1} - E_i^k \quad (2-37)$$

## 2-12-2/ ALGORITHME DE GAUSS SEIDEL

Etape1 :

Formation de la matrice admittance [Y]

Etape2 :

Estimation des valeurs initiales des tensions nodales  $E_i^0 \quad i=1, \dots, n$

Etape3 :

Détermination des paramètres  $KL_i$  et  $YL_{ij} \begin{cases} i=1, \dots, n \\ j=1, \dots, n \end{cases}$

-initiation des itérations  $k=0$

Etape4 :

Calcul itératif des tensions pour chaque nœud suivant la relation :

$$E_i^{k+1} = \frac{KL_i}{[E_i^k]^*} - \sum_{j=1}^{i-1} YL_{ij} \cdot E_j^{k+1} - \sum_{j=i+1}^n YL_{ij} \cdot E_j^k$$

On calcul l'écart entre les valeurs d'une même tension trouvée aux itérations qui se suivent :

$$\Delta E_i^{(k)} = E_i^{(k+1)} - E_i^{(k)}$$

On introduit le facteur d'accélération  $\alpha$  pour réduire le nombre d'itérations.

Etape5 :

Une fois le test de convergence est vérifiée ( $Max \Delta E^{(k)} \leq \epsilon$ ), les valeurs des tensions de la dernière itération sont retenues, on calcul :

- les puissances transitées :  $S_{ij}^* = E_i^* (E_i - E_j) y_{ij} + E_i^* \cdot E_i \cdot \frac{y_{ij}'}{2}$

-les puissances injectées :  $S_i = \sum_{\substack{j=1 \\ j=i}}^n S_{ij}$

-les pertes :  $S_l = \sum_{i=1}^n S_i$

Sinon aller à l'étape 4.

## 2-13) ECOULEMENT OPTIMAL DES CHARGES (OPTIMAL LOAD FLOW)

De l'étude de la section (2-10), on se rend compte que la demande totale d'un réseau est indépendante de l'opérateur du système et peut être satisfaite de différentes manières. Cependant, il est nécessaire de choisir une seule configuration. Cette situation oblige à spécifier deux variables par barre, ensuite de décider de l'ajustement approprié des paramètres restants. Le choix d'un critère optimal est toujours subjectif, mais incombe la formulation d'un problème dit optimal.

Les méthodes numériques utilisées pour résoudre le problème d'écoulement optimal des charges avec contraintes peuvent être divisées en deux grandes catégories:

- les méthodes directes
- les méthodes indirectes.

Les méthodes directes sont des méthodes de recherche qui ouvrent directement sur le problème original, orientant leur recherche d'une région dite réalisable vers la solution optimale. D'une itération à l'autre, la fonction objective diminue continuellement, ce sont :

- la programmation linéaire, méthode du Simplex
- méthode de la direction réalisable
- méthode du gradient
- méthode du gradient réduit.

Les méthodes indirectes supposent que le problème possède au départ les conditions nécessaires de continuité et de différentiabilité, et solutionnent le problème transformé ou un groupe de formulations approximatives du même problème sans contrainte, caractérisé par des équations représentant les conditions d'optimalité. Les plus rencontrées sont :

- la programmation quadratique récursive
- la programmation linéaire récursive
- les méthodes des multiplicateurs.

## 2-14) CONCLUSION

Le problème d'écoulement statique des charges d'un réseau électrique peut être formulé avec ou sans contraintes. Le développement des relations de tout modèle conduit à des équations non linéaires. Compte tenu de la complexité des systèmes (nombre de barres et de lignes élevé), les méthodes de solution sont toujours itératives.

En général, connaissant la consommation totale de puissance, la solution de l'écoulement statique contient les étapes suivantes: (1) formuler le modèle mathématique c'est-à-dire, de développer des relations entre puissances générées, (2) spécifier les contraintes du système, (3) calculer les écoulements de charge dans chaque ligne. Vu la simplicité et l'efficacité de la méthode de Gauss-Seidel, on a appliqué cette dernière pour le calcul de l'écoulement des charges dans la suite de cette thèse.

***CHAPITRE III***  
***LES METHODES D'OPTIMISATION***

## LES METHODES D'OPTIMISATION

### 3-1) INTRODUCTION

L'optimisation est un ensemble de techniques permettant de trouver les variables qui rendent optimale une fonction de réponse, appelée « fonction objectif ». Sur le plan mathématique, cela correspond à la recherche des extrémums de fonctions à une ou plusieurs variables. Dans le domaine des sciences appliquées, il s'agit en général de trouver l'optimum de la réponse d'opérations industrielles ou d'expériences de laboratoire.

Les méthodes d'optimisations les plus courantes, sont les méthodes de programmation linéaire et les méthodes de programmation non linéaire.

La méthode de programmation non linéaire a été la première méthode à connaître un essor remarquable depuis la seconde guerre mondiale attirant l'attention des chercheurs ; les solutions qu'elles offrent couvrent un large champ d'application.

Dans les années quatre vingt, d'autres méthodes ont été fréquemment utilisées citons [LAO] :

- Des méthodes de programmations successives
- Des méthodes de lagrangien augmenté
- Des méthodes de programmation quadratique, mentionnons les méthodes de **Newton** et quasi-**Newton**

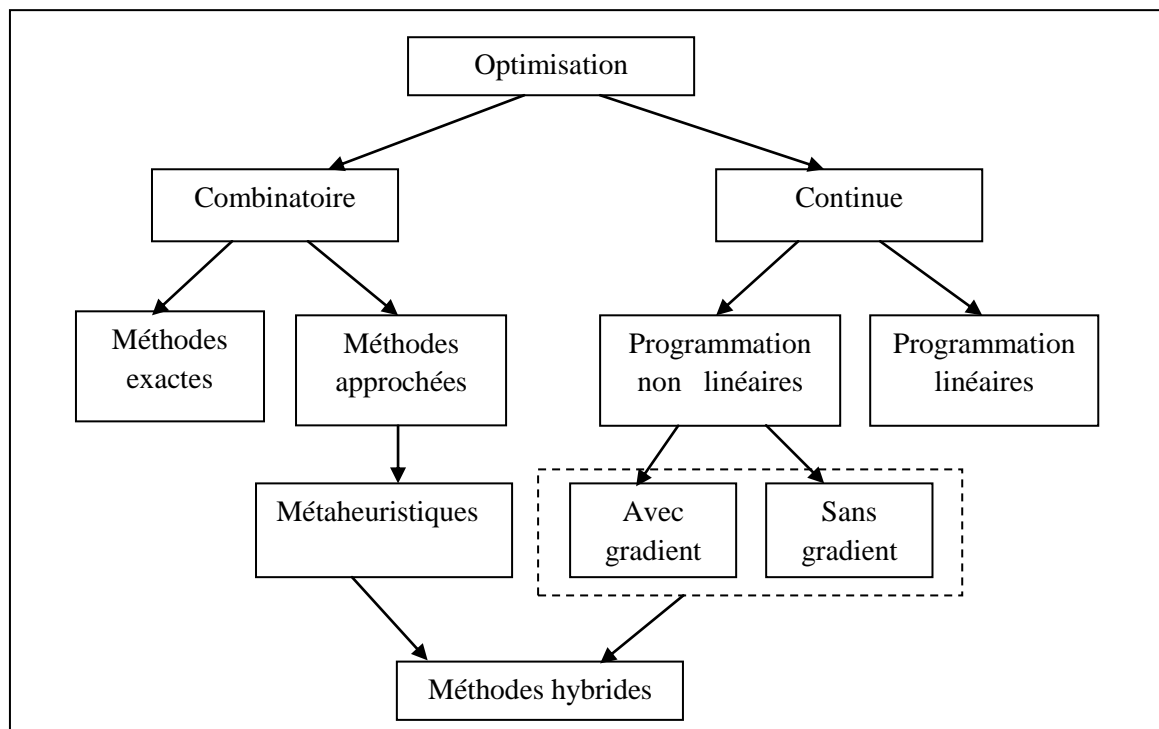


Figure 3.1. Les méthodes d'optimisation



### 3-2) OPTIMISATION CONTINUE

Parmi les méthodes d'optimisations continues, on peut citer la programmation linéaire et la programmation non linéaire. Dans la programmation non linéaire on trouve deux sortes, programmation avec contraintes et sans contraintes.

#### 3-2-1/ PROGRAMMATION NON LINEAIRE

##### 3-2-1-a) PROGRAMMATION NON LINEAIRE AVEC CONTRAINTES

##### 3-2-1-a-1/ METHODE DES COUTS MARGINAUX

La fonction de Lagrange associé au problème est par définition :

$$L(p, \lambda) = \sum_{i=1}^{NG} C_i(P_{Gi}) + \lambda \left( \sum_{i=1}^{NG} P_{Gi} - \sum_{j=1}^{NC} P_{Chj} - P_L \right) \quad (3-1)$$

Où :

$\lambda$  : Multiplicateur de Lagrange.

La condition nécessaire pour que la fonction  $L$  soit minimale est :

$$\frac{\partial L}{\partial P_{Gi}} = 0 \quad i = 1, \dots, NG \quad (3-2)$$

$$\frac{\partial L}{\partial \lambda} = 0 \quad (3-3)$$

L'équation (3-3) n'est que la contrainte d'égalité

$$\frac{\partial L}{\partial P_{Gi}} = \frac{\partial C_i}{\partial P_{Gi}} + \lambda \left( 1 - \frac{\partial P_L}{\partial P_{Gi}} \right) = 0 \quad (3-4)$$

D'où:

$$\lambda = \frac{-\partial C_i / \partial P_{Gi}}{1 - \partial P_L / \partial P_{Gi}} \quad (3-5)$$

$$\lambda = \frac{\partial C_i}{\partial P_{Gi}} \cdot \frac{1}{\frac{\partial P_L}{\partial P_{Gi}} - 1} \quad (3-6)$$

On pose:

$$(ITL)_i = \frac{\partial P_L}{\partial P_{Gi}} = 2 \cdot \sum_{j=1}^n P_j \alpha_{ij} + \beta_{ij} Q_j \quad (3-7)$$

$$(IC)_i = \frac{\partial Ci}{\partial P_{Gi}} = \lambda [(ITL)_i - 1] \quad (3-8)$$

Avec:

$$\begin{cases} \alpha_{ij} = \frac{R_{ij}}{|E_i| |E_j|} \cos \delta_{ij} \\ \beta_{ij} = \frac{-R_{ij}}{|E_i| |E_j|} \sin \delta_{ij} \end{cases} \quad (3-9)$$

$(IC)_i$  : représente le coût incrémental de l'unité « i »

$(ITL)_i$  : représente les pertes incrémentales associés à l'unité « i »

$\frac{1}{\frac{\partial P_L}{\partial P_{Gi}} - 1}$  : le facteur de pénalité pour le nœud « i »

$P_j$  : la puissance active injectée au nœud « j »

$Q_j$  : la puissance réactive injectée au nœud « j »

$R_{ij}$  : résistance entre le nœud i et j

$\delta_{ij}$  : le déphasage de la tension entre le nœud i et j

$|E_i|$  : le module de la tension au nœud « i »

$|E_j|$  : le module de la tension au nœud « j »

**Algorithme de résolution**Etape1 :

Déterminer la matrice impédance [Z] par inversion de la matrice admittance [Y].

Etape2 :

Calculer les coefficients  $\alpha_{ij}$  et  $\beta_{ij}$  pour  $i=1, \dots, n$  et  $j=1, \dots, n$ .

Etape3 :

Calculer les pertes incrémentaux de transmission  $(ITL)_i$  en chaque nœud producteur.

Etape4 :

Initialiser le coefficient de Lagrange  $\lambda^0$ .

Etape5 :

Calculer les coûts marginaux  $(IC)_i$  en chaque nœud producteur et par suite on détermine les puissances actives optimales par l'expression suivante :

$$P_{Gi}^{opt} = \frac{(IC)_i - B_i}{2A_i}$$

Avec :

$A_i$  et  $B_i$  sont les coefficients de la fonction coût.

Etape6 :

Vérifier les contraintes :

$$\sum_{i=1}^{NG} P_{Gi} - \sum_{j=1}^{NC} P_{Chj} - P_L = 0$$

$$P_{Gi}^{\min} \leq P_{Gi} \leq P_{Gi}^{\max}$$

Etape7 :

Cette étape sert à choisir le nouveau coefficient «  $\lambda$  » de la prochaine itération, dans le cas où l'équation du bilan ne serait pas vérifiée ;

Si par exemple :  $\sum_{i=1}^{NG} P_{Gi}^k > P_{Ch} + P_L$ , on choisi  $\lambda^{k+1} < \lambda^k$ , ( $\lambda = \lambda - \Delta\lambda$ ) aller à l'étape 5

Si par exemple :  $\sum_{i=1}^{NG} P_{Gi}^k < P_{Ch} + P_L$ , on choisi  $\lambda^{k+1} > \lambda^k$ , ( $\lambda = \lambda + \Delta\lambda$ ) aller à l'étape 5

$\Delta\lambda$  : le pas de déplacement.

### 3-2-2/ PROGRAMMATION NON LINEAIRE SANS CONTRAINTES

#### 3-2-2-1/ METHODE DE GRADIENT

Elle repose sur les démarches suivantes : [MIN 83]

On part d'un point  $x^0$  et on calcule le gradient  $\nabla f(x^0)$  en  $x^0$ , comme  $\nabla f(x^0)$  indique la direction de plus grande augmentation de  $f$ , on se déplace d'une quantité  $\lambda_0$  dans la direction opposée au gradient, et on définit le point :

$$x^1 = x^0 - \lambda_0 \frac{\nabla f(x^0)}{\|\nabla f(x^0)\|} \quad (3-10)$$

La procédure est répétée et engendre les points  $x^0, x^1, \dots, x^k$  suivant la relation :

$$x^{k+1} = x^k - \lambda_k \frac{\nabla f(x^k)}{\|\nabla f(x^k)\|} \quad (3-11)$$

Où :  $\lambda_k > 0$

#### 3-2-2-2/ METHODE DE GRADIENT CONJUGUE

On suppose ici que la fonction à minimiser est quadratique de la forme suivante [MIN 83]:

$$q(x) = \frac{1}{2} x^T A x + b^T x + c \quad (3-12)$$

Où:

A: une matrice  $n \times n$  définie positive et symétrique ;

b : un vecteur de  $n$  éléments ;

c : une constante.

L'idée de la méthode est de construire progressivement les directions  $d_0, d_1, \dots, d_k$  par rapport à la matrice A. à chaque étape  $k$  la direction  $d_k$  est obtenue par combinaison linéaire du gradient  $-\nabla q(x^k)$  en  $x^k$ , et des directions précédentes  $d_0, d_1, \dots, d_{k-1}$ .

En notons  $g_k = \nabla q(x^k)$  le gradient de la fonction  $q$  en  $x^k$ , l'algorithme de résolution de la méthode prend la forme suivante :

**Algorithme de résolution**Etape1 :

Soit  $x^0$  le point de départ,  $g_0 = \nabla q(x^0) = Ax^0 + b$

Poser  $d_0 = -g_0, k=0$

Etape2 :

A l'itération  $k$ , on est au point  $x^k$

Définir  $x^{k+1} = x^k + \lambda_k \cdot d_k$

Avec  $\lambda_k = \frac{g_k^T \cdot d_k}{d_k^T \cdot A \cdot d_k}$

Puis  $d_{k+1} = -g_{k+1} + \beta_k d_k$

Avec  $\beta_k = \frac{g_k^T \cdot A \cdot d_k}{d_k^T \cdot A \cdot d_k}$

Etape3 :

Test d'arrêt sinon aller à l'étape 2.

**3-2-2-3/ METHODE DE NEWTON**

Cette méthode consiste à remplacer la fonction  $f$  par son approximation quadratique au voisinage du point  $x^k$  [MIN 83].

$$q(x) = f(x^k) + \nabla f^T(x^k)(x - x^k) + \frac{1}{2}(x - x^k)^T \nabla^2 f(x^k)(x - x^k) \quad (3-13)$$

On prend  $x^{k+1}$  comme le point minimum de  $q(x)$ , on supposant que la fonction  $f$  est deux fois continûment différentiable.

$$q(x^{k+1}) = f(x^k) + \nabla f^T(x^k)(x^{k+1} - x^k) + \frac{1}{2}(x^{k+1} - x^k)^T \nabla^2 f(x^k)(x^{k+1} - x^k) \quad (3-14)$$

On suppose que la fonction  $f$  admet un minimum unique  $x^{k+1}$ , (si  $\nabla^2 f(x^k)$  est une matrice définie positive) défini par :

$\nabla q(x^{k+1}) = 0$ , ce qui conduit au système linéaire suivant :

$$\nabla f(x^k) = -\nabla^2 f(x^k)(x^{k+1} - x^k) \quad (3-15)$$

D'où la formule itérative :

$$x^{k+1} = x^k - [\nabla^2 f(x^k)]^{-1} \cdot \nabla f(x^k) \quad (3-16)$$

Lorsqu'on veut appliquer cette méthode sur une fonction quelconque, ils apparaissent des difficultés dus essentiellement au fait qu'elle ne possède pas de propriété de convergence globale, pour remédier à ces difficultés, il faut introduire des modifications, on peut agir par exemple sur le pas de déplacement en utilisant la formule itérative du type :

$$x^{k+1} = x^k - \lambda_k [\nabla^2 f(x^k)]^{-1} \cdot \nabla f(x^k) \quad (3-17)$$

Où :  $\lambda_k$  est un scalaire choisi de façon à ce que la norme  $\|x^{k+1} - x^k\|$  ne soit pas trop grande. On peut aussi le choisir de façon à ce que  $x^{k+1}$  minimise  $g(\lambda) = f(x^k + \lambda d_k)$  dans la direction :

$$d_k = -[\nabla^2 f(x^k)]^{-1} \cdot \nabla f(x^k) \quad (3-18)$$

### 3-2-2-4/METHODE DE QUASI-NEWTON

Le principe de la méthode consiste à généraliser la fonction itérative de newton. [MIN 83]

$$x^{k+1} = x^k - \lambda_k [\nabla^2 f(x^k)]^{-1} \cdot \nabla f(x^k) \quad (3-19)$$

La limitation importante de la méthode de newton consiste dans la restriction :  $\nabla^2 f$  définie positive.

Cette méthode consiste à remplacer  $[\nabla^2 f(x^k)]^{-1}$  par une matrice  $H_k$  définie positive donnant la direction de déplacement à partir du gradient  $\nabla f(x^k)$ , d'où une formule itérative du type :

$$x^{k+1} = x^k - \lambda_k H_k \cdot \nabla f(x^k) \quad (3-20)$$

$\lambda_k$  : est choisi de façon à minimiser  $g(\lambda) = f(x^k + \lambda d_k)$  dans la direction  $d_k = -H_k \cdot \nabla f(x^k)$ .

Il existe beaucoup de variantes dans le choix de la formule de mise à jour de la matrice  $H_k$ . La formule de correction qui permet d'obtenir la matrice  $H_{k+1}$ , à partir de la matrice  $H_k$  utilise les nouvelles informations obtenues lors de l'étape  $k$  de l'algorithme.

$$H_{k+1}[\nabla f(x^{k+1}) - \nabla f(x^k)] = x^{k+1} - x^k \quad (3-21)$$

Avec:

$$\begin{cases} \delta_k = x^{k+1} - x^k \\ \gamma_k = \nabla f(x^{k+1}) - \nabla f(x^k) \end{cases} \quad (3-22)$$

$$H_{k+1} = H_k + \frac{(\delta_k - H_k \gamma_k)(\delta_k - H_k \gamma_k)^T}{\gamma_k^T (\delta_k - H_k \gamma_k)} \quad (3-23)$$

### 3-2-2-5/ ALGORITHME DE DAVIDON-FLETCHER-POWEL (D.F.P)

Cet algorithme utilise la formule de correction suivante: [MIN 83, RAH 95 b]

$$H_{k+1} = H_k + \frac{\delta_k \cdot \delta_k^T}{\delta_k^T \cdot \gamma_k} - \frac{H_k \cdot \gamma_k \cdot \gamma_k^T \cdot H_k}{\gamma_k^T \cdot H_k \cdot \gamma_k} \quad (3-24)$$

Où : le point  $x^{k+1}$  est obtenu de  $x^k$  par déplacement dans la direction  $d_k = -H_k \cdot \nabla f(x^k)$

Etape1:

$x^0$  point de départ. Choisir  $H_0$  définie positive quelconque (par exemple la matrice unité)  $k=0$

Etape2:

A l'itération  $k$ , déterminer la direction de déplacement

$$d_k = -H_k \cdot \nabla f(x^k)$$

Déterminer  $x^{k+1}$  comme le minimum de  $f(x^k + \theta d_k)$  pour  $\theta \geq 0$

Calculer  $\delta_k = x^{k+1} - x^k$ , et  $\gamma_k = \nabla f(x^{k+1}) - \nabla f(x^k)$ .

$$\text{Puis } H_{k+1} = H_k + \frac{\delta_k \cdot \delta_k^T}{\delta_k^T \cdot \gamma_k} - \frac{H_k \cdot \gamma_k \cdot \gamma_k^T \cdot H_k}{\gamma_k^T \cdot H_k \cdot \gamma_k}$$

Etape3:

Test d'arrêt ou retour à l'étape2.

### 3-2-2-6/ ALGORITHME DE BROYDON-FLETCHER-GOLDFARD-SHANNO (B.F.G.S)

Cet algorithme utilisé, pour construire une approximation de l'inverse du hessien, une formule de correction directement dérivée de la formule (3-24) [MIN 83].

$$H_{k+1} = H_k + \frac{\gamma_k \cdot \gamma_k^T}{\gamma_k^T \cdot \delta_k} - \frac{H_k^T \cdot H_k}{\delta_k^T \cdot H_k \cdot \delta_k} \quad (3-25)$$

L'algorithme de B.F.G.S se déduit alors directement de l'algorithme de D.F.P.

### 3-3) PROGRAMMATION LINEAIRE

La programmation linéaire a pour but d'optimiser une fonction linéaire de plusieurs variables. Ces derniers étant liés par ailleurs par un certain nombre de contraintes. C'est une branche très importante de la recherche opérationnelle, il existe des algorithmes évolués permettant de résoudre des problèmes à un grand nombre de variables comme la méthode de simplexe.

#### 3-3-1/METHODE DE SIMPLEXE

La méthode du simplexe est un algorithme qui nous permet de calculer le minimum d'une fonction linéaire à partir d'une solution de base admissible satisfaisante les contraintes.

Soit la fonction à minimiser :

$$\text{Min } F = (C_0 + \sum_{j=1}^n C_j X_j) \quad (3-26)$$

$$\text{Sous les contraintes : } \left\{ \sum_{i=1}^m a_{ij} X_j \leq b_j, \quad X_j \geq 0 \right.$$

La fonction  $F$ , et les contraintes peuvent s'écrire aussi :

$$\text{Min } F = \min (C_0 + C_1 X_1 + C_2 X_2 + \dots + C_n X_n) \quad (3-27)$$

Sous les contraintes :

$$\begin{cases} a_{11} X_1 + a_{12} X_2 + \dots + a_{1n} X_n \leq b_1 \\ a_{m1} X_1 + a_{m2} X_2 + \dots + a_{mn} X_n \leq b_m \end{cases} \quad (3-28)$$



Sous la forme matricielle on a :

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_n \end{bmatrix} \leq \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix} \quad (3-29)$$

Les inégalités sont transformées aux égalités par l'introduction de variables d'écart :  $X_{n+1}, X_{n+2}, \dots, X_{n+m}$ .

$$\begin{cases} a_{11}X_1 + a_{12}X_2 + \dots + a_{1n}X_n + X_{n+1} = b_1 \\ a_{21}X_1 + a_{22}X_2 + \dots + a_{2n}X_n + X_{n+2} = b_2 \\ \dots \\ a_{m1}X_1 + a_{m2}X_2 + \dots + a_{mn}X_n + X_{n+m} = b_m \end{cases} \quad (3-30)$$

Le système matriciel devient :

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} & 1 & 0 & \dots & 0 \\ a_{21} & a_{22} & \dots & a_{2n} & 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} & 0 & \dots & \dots & 1 \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_{n+1} \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix} \quad (3-31)$$

On pose:

$$A(m,n) = \begin{bmatrix} a_{11} & \dots & a_{1n} \\ a_{m1} & \dots & a_{mn} \end{bmatrix};$$

$$A(m,n+m) = \begin{bmatrix} a_{11} & \dots & a_{1n} & 1 & 0 & \dots & 0 \\ a_{m1} & \dots & a_{mn} & 0 & 0 & \dots & 1 \end{bmatrix};$$

$$C(n) = (C_1, C_2, \dots, C_n);$$

$$C(n,n+m) = (C_1, C_2, \dots, C_{n+1}, \dots, C_{n+m})$$

$$\Gamma(m) = (C_{n+1}, \dots, C_{n+m})$$

$$B(m,n) = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & \dots & & 1 \end{bmatrix}$$

Avec :

$A(m,n)$  : matrice contenant les coefficients de  $X_j, j=1, \dots, n$ .

$A(m,n+m)$  : matrice contenant les coefficients de  $X_j$  après introduction des variables d'écart.

$C(n)$  : vecteur contenant les coefficients de la matrice Z.

$C(n,n+m)$  : vecteur contenant les coefficients de la matrice Z après introduction des variables d'écart.

$B(m,n)$  : la matrice de base.

$\Gamma(m)$  : vecteur formé de  $C_j$  tel que  $j \geq n+1$

$Y(m)$  : vecteur des  $b_i$

### Algorithmes de résolution

Soit  $\text{Min } F = \min(C_0 + C_1 X_1 + \dots + C_n X_n)$

$$\begin{bmatrix} a_{11} & \dots & a_{1n} \\ \dots & \dots & \dots \\ a_{m1} & \dots & a_{mn} \end{bmatrix} \begin{bmatrix} X_1 \\ \vdots \\ X_n \end{bmatrix} \leq \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix}$$

Après l'introduction des variables d'écart on aura le système suivant :

$$\begin{bmatrix} a_{11} & \dots & a_{1n} & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_{m1} & \dots & a_{mn} & 0 & \dots & 1 \end{bmatrix} \begin{bmatrix} X_1 \\ \vdots \\ X_{n+1} \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix}$$

$$B(m,n) = \begin{bmatrix} 1 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 \end{bmatrix} \text{ matrice de base}$$

$$\Gamma(m) = (0, 0, \dots, 0)$$

Etape 1 :

On calcul les nouveaux coefficients de  $F$

$$\bar{Y}(m) = B^{-1}(m,n).Y(m) : \text{nouveaux coefficients } b_i$$

$$\bar{A}(m,n+m) = B^{-1}(m,n).A(m,n+m) : \text{nouveaux coefficients } X_j$$

$$\bar{C}(n+m) = C(n+m) - \Gamma(m).A(m,n+m) : \text{nouveaux coefficients de } F$$

Etape 2 :

Si tous les  $C_j$  sont positifs ou nuls donc nous sommes à l'optimum puisque tous les  $X_j$  sont positifs ou nuls. La valeur minimale de la fonction coût  $F$  est la valeur précédente. (à la première étape) .

Etape3 :

S'il existe un seul  $C_j$  négatif, la valeur correspondante à  $C_j$  sera celle qui entrera dans la nouvelle base  $B(m,n)$ . Il s'agit de savoir quelle est la variable d'écart qui sortira de  $B(m,n)$ . Pour cela calculons le  $\min X_j$  comme suit :  $Y(m) - A_j X_j \geq 0$ ,  $\forall A_j \geq 0$

Etape4 :

La nouvelle valeur de  $F$  est  $F = F + C_j \cdot \min X_j$

Etape5 :

On effectue le produit :  $w(m) = A(m, n+m) \cdot v(n+m)$

$$\text{Avec } v(n+m) = \begin{bmatrix} 0 \\ \min X_j \\ 0 \end{bmatrix}$$

Etape6 :

On compare chaque élément de  $w$  à l'élément correspondant de  $Y(m)$   
 -s'il n'y a aucune égalité entre les éléments de  $w(m)$  et ceux de  $Y(m)$ , la valeur de  $F$  est celle trouvée dans l'étape 4.  
 -si une seule égalité  $y_k = w_k$ , la variable qui sort de la base  $b$  est  $X_{n+k}$ .

Etape7 :

On calcule la matrice  $B(m,n)$ , puis on commence par l'étape 1.  
 le coefficient de  $X_{n+k}$  est remplacé par  $X_j$  et  $A$  remplace  $\bar{A}$ ,  $Y$  remplace  $\bar{Y}$ ,  $C$  remplace  $\bar{C}$ .

**3-4) METHODE DE PENALITE**

Les méthodes de pénalités sont des méthodes simples et efficaces. Leur principe consiste à ramener le problème d'optimisation avec contraintes en un problème d'optimisation sans contraintes. [MIN 83]

Soit le problème :

$$\min \{F(x), x \in D\} \quad (3-32)$$

Où

$$D = G \cap H \quad (3-33)$$

Avec

$$G = \{x / g_i(x) \geq 0, \quad i = 1, 2, 3 \dots m\}$$

Et

$$H = \{x / h_j(x) = 0, \quad j = 1, 2, 3 \dots n\}$$

G et H étant respectivement les ensembles de contraintes de types inégalité et égalité.

#### a- Méthode de pénalité extérieure [CHA]

Dans cette méthode, on introduit les deux types de contraintes égalités et inégalités. La fonction objective  $F_m$  s'écrit sous la forme suivante :

$$F_m = F(x) + E(r_k, G, H) \quad (3-34)$$

Où :  $E(r_k, G, H)$  est le terme de pénalisation extérieure.

$$E(r_k, G, H) = \frac{1}{r_k} \sum_{i=1}^m D_i g_i^2(x) + \frac{1}{r_k} \sum_{j=1}^n B_j h_j^2(x) \quad (3-35)$$

On aura donc  $F_m$  sous la forme suivante :

$$F_m = F(x) + \frac{1}{r_k} \sum_{i=1}^m D_i g_i^2(x) + \frac{1}{r_k} \sum_{j=1}^n B_j h_j^2(x) \quad (3-36)$$

Avec

$$\begin{cases} D_i > 0, \text{ si } g_i(x) < 0 \\ D_i = 0, \text{ si } g_i(x) \geq 0 \end{cases}$$

Et

$$\begin{cases} B_j > 0, \text{ si } h_j(x) \neq 0 \\ B_j = 0, \text{ si } h_j(x) = 0 \end{cases}$$

$D_i$  et  $B_j$  sont des constantes.

**b- Méthode de pénalité intérieure [CHA]**

Le principale inconvénient de la méthode de pénalité extérieure est que l'optimum  $x^*$  est approché vers l'extérieur ce qui a conduit de chercher une autre méthode de pénalité dans les quelles l'optimum est approché vers l'intérieur (d'où le nom de pénalité intérieure). [MIN 83]

La pénalisation intérieure peut être appliquée uniquement dans le cas, où on a des contraintes de type inégalité.

La fonction objective  $F_m$  s'écrit sous la forme suivante :

$$F_m = F(x) + I(r_k, G) \quad (3-37)$$

Où

$I(r_k, G)$  est le terme de pénalisation intérieure avec

$$I(r_k, G) = r_k \sum_{i=1}^m \frac{A_i}{g_i(x)} \quad (3-38)$$

On aura donc  $F_m$  sous la forme suivante :

$$F_m = F(x) + r_k \sum_{i=1}^m \frac{A_i}{g_i(x)} \quad (3-39)$$

Avec

$$\begin{cases} A_i > 0, & \text{si } g_i(x) \geq 0 \\ A_i = 0, & \text{si } g_i(x) < 0 \end{cases}$$

$A_i$  : une constante

**c- Méthode de pénalité mixte [CHA]**

Elle englobe les termes de pénalisation intérieure représentés par  $I(r_k, G)$  et les terme de pénalisation extérieure représentés par  $E(r_k, G, H)$ .

La fonction objective s'écrit sous la forme suivante :

$$F_m = F(x) + r_k \sum_{i=1}^m \frac{A_i}{g_i(x)} + \frac{1}{r_k} \sum_{i=1}^m D_i g_i^2(x) + \frac{1}{r_k} \sum_{j=1}^n B_j h_j^2(x) \quad (3-40)$$

### **3-5) OPTIMISATION PAR METAHEURISTIQUES**

Les métaheuristiques constituent une classe de méthodes approchées adaptables à un très grand nombre de problèmes combinatoires et de problèmes d'affectations sans contraintes. Elles ont révélé leur grande efficacité pour fournir des solutions approchées de bonne qualité pour un grand taille, c'est pour quoi l'étude de ces méthodes est actuellement en plein développement.

### **3-6) CONCLUSION**

Une étude sur les différentes méthodes d'optimisation à été réalisée. Pour notre travail de recherche nous nous sommes intéressés par les classes des métaheuristiques, à savoir les méthodes de voisinages et les algorithmes évolutifs.

***CHAPITRE IV***  
***METAHEURISTIQUES POUR***  
***L'OPTIMISATION***  
***COMBINATOIRE ET L'AFFECTATION***  
***SOUS CONTRAINTES***

## **METAHEURISTIQUES POUR L' OPTIMISATION COMBINATOIRE ET L'AFFECTION SOUS CONTRAINTES**

### **4-1) INTRODUCTION**

L'optimisation combinatoire occupe une place très importante en recherche opérationnelle, en métaheuristiques discrètes et en informatique. Son importance se justifie d'une part par la grande difficulté des problèmes d'optimisation et d'autre part par de nombreuses applications pratiques pouvant être formulées sous la forme d'un problème d'optimisation combinatoire. Bien que les problèmes d'optimisation combinatoire soient souvent faciles à définir, ils sont généralement difficiles à résoudre. En effet, la plupart de ces problèmes appartiennent à la classe des problèmes non linéaires (NP) difficiles et ne possèdent donc pas à ce jour de solution algorithmique efficace valable pour toutes les données.

Etant donnée l'importance de ces problèmes, de nombreuses méthodes de résolution ont été développées en recherche opérationnelle (RO) et en intelligence artificielle (IA). Ces méthodes peuvent être classées sommairement en deux grandes catégories : les méthodes exactes (complètes) qui garantissent la complétude de la résolution et les méthodes approchées (incomplètes) qui perdent la complétude pour gagner en efficacité.

Le principe essentiel d'une méthode exacte consiste à énumérer, souvent de manière implicite, l'ensemble des solutions de l'espace de recherche. Pour améliorer l'énumération des solutions, une telle méthode dispose de techniques pour détecter le plus tôt possible des échecs (calculs de bornes) et d'heuristiques spécifiques pour orienter les différents choix. Parmi les méthodes exactes, on trouve la plupart des méthodes traditionnelles (développées depuis une trentaine d'années) telles les techniques de séparation et évaluation progressive (SEP) ou les algorithmes avec retour arrière. Les méthodes exactes ont permis de trouver des solutions optimales pour des problèmes de taille raisonnable. Malgré les progrès réalisés (notamment en matière de la programmation linéaire en nombres entiers), comme le temps de calcul nécessaire pour trouver une solution risque d'augmenter exponentiellement avec la taille du problème, les méthodes exactes rencontrent généralement des difficultés face aux applications de taille importante.

Les méthodes approchées constituent une alternative très intéressante pour traiter les problèmes d'optimisation de grande taille si l'optimalité n'est pas primordiale. En effet, ces méthodes sont utilisées depuis longtemps par de nombreux praticiens. On peut citer les méthodes gloutonnes et l'amélioration itérative.

### **4-2) DEFINITION DE L'OPTIMISATION COMBINATOIRE**

Un problème d'optimisation combinatoire est défini par un ensemble d'instances. A chaque instance du problème est associé un ensemble discret de solution  $S$ , un sous-ensemble de  $X$  de  $S$  représentant les solutions admissibles (réalisables) et une fonction de coût  $f$  (ou fonction objectif) qui assigne à chaque solution  $s$  appartient à  $X$  le nombre réel



(ou entier)  $f(s)$ . Résoudre un tel problème consiste à trouver une solution  $s^*$  appartenant à  $X$  optimisant la valeur de la fonction de coût  $f$ . Une telle solution  $s^*$  s'appelle une *solution optimale* ou un *optimum global*.

### 4-3) RECHERCHE LOCALE ET RECHERCHE GLOBALE

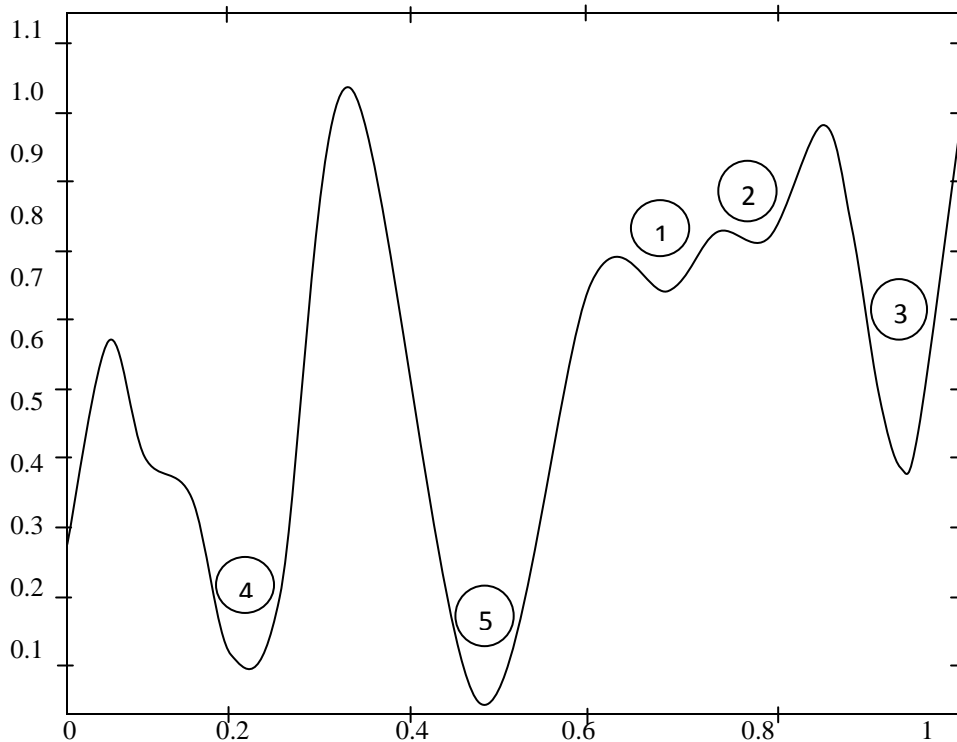
La méthode d'optimisation globale vise la détermination des optima globaux du problème, en évitant le piègeage dans l'un de ses optima locaux. Cette dénomination présente néanmoins une certaine ambiguïté, car on rencontre souvent dans la littérature la dénomination de « méthode locale », qui fait cette fois référence au mécanisme de recherche, lorsqu'il procède par voisins successifs.

Nous pouvons partager les méthodes en deux catégories. Celles qui permettent de déterminer un minimum local, ces méthodes sont appelées locales, et celles qui s'efforcent de déterminer un optimum global, ces méthodes sont appelées méthodes de recherche globale.

Nous donnons un exemple plus simple pour traiter l'optimum global et l'optimum local d'une fonction objectif à une seule variable.

Sur un ensemble  $X$  de solutions  $x$ , avec  $X \subset \mathbb{R}$ , on cherche à optimiser une fonction  $f$ , appelée *fonction objectif*.

Un exemple d'une telle fonction à une dimension est présenté dans la figure ci-dessous. Il s'agit d'une fonction à une variable réel  $x$ , pouvant varier entre 0 et 1, dont on cherche le minimum.



**Figure 4.1.** Exemple d'une fonction objectif à une variable.

La fonction  $f$  est un polynôme en  $x$  dont les coefficients  $a_j$  ont été calculés par une méthode de moindres carrés à partir d'une liste de données :

$$f(x) = \sum_{j=0}^{20} a_j \cdot x^j$$

Cette fonction  $f$  présente plusieurs caractéristiques intéressantes pour l'étude d'un algorithme d'optimisation :

- ✓ Un minimum global (5) :  $x^* = 0.4666$  ;  $f(x) = 0.01048$  ;
- ✓ Un minimum local (4) presque aussi bas que le minimum global, mais situé dans une vallée plus large ;
- ✓ Plusieurs autres minima locaux (1, 2, 3), situés dans des vallées plus au moins étroites et profondes.

#### • GENERALISATION DES FONCTIONS

Soit  $x$  un vecteur de dimension finie  $n$  dont les composants  $x_i$  vérifient  $a_i \leq x_i \leq b_i$ , avec  $i=1, \dots, n$ , où  $a_i$  et  $b_i$  sont les composantes données de deux vecteurs  $\mathbf{A}$  et  $\mathbf{B}$ , de dimension  $n$ . Les vecteurs  $\mathbf{A}$  et  $\mathbf{B}$  définissent un domaine hyper-rectangulaire que l'on note  $X \subset \mathbb{R}^n$ .

Soit la fonction  $f$  à valeurs réelles définie comme suit :

$$f : X \rightarrow \mathbb{R}$$

$$x \rightarrow f(x)$$

Nous nous intéressons à la recherche des minima d'une fonction, puisqu'un problème de maximisation peut toujours se ramener à un problème de minimisation.

Le problème considéré est celui de trouver le ou un minimum global de  $f$ , et d'éviter d'être piégé dans un minimum local.

$$f(x^*) \text{ est un minimum local } \Leftrightarrow \left[ \exists \varepsilon > 0 / \forall x \in X : \|x - x^*\| < \varepsilon \Rightarrow f(x) \geq f(x^*) \right] \text{ et } x^* \in X$$

$$f(x^*) \text{ est un minimum global } \Leftrightarrow \forall x \in X : f(x) \geq f(x^*) \text{ et } x^* \in X$$

$$\text{Pour un minimum global : } f(x^*) = y^* = \min_{x \in X} \{f(x)\}$$

#### 4-4) CLASSIFICATION DES METAHEURISTIQUES

On peut faire la différence entre les métaheuristiques qui s'inspirent de **phénomènes naturels** comme par exemple la recherche tabou, et celles qui s'inspirent de **phénomènes ou la théorie de l'évolution** comme les Algorithmes Génétiques et les algorithmes des fourmis.

Une telle classification ne semble cependant pas très utile et est parfois difficile à réaliser. En effet, il existe de nombreuses métaheuristiques récentes qu'il est difficile de classer dans l'une des 2 catégories. Certains se demanderont par exemple si l'utilisation d'une mémoire dans la méthode Tabou n'est pas directement inspirée de la nature.

Une autre façon de classer les métaheuristiques est de distinguer celles qui travaillent avec une **population de solutions** de 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**. La méthode Tabou, le Recuit Simulé et la Recherche à Voisins Variables sont des exemples typiques de méthodes de trajectoire. Ces méthodes construisent une trajectoire dans l'espace des solutions en tentant de se diriger vers des solutions optimales. L'exemple le plus connu de méthode qui travaille avec une population de solutions est l'algorithme génétique.

Un très grand nombre de résolutions existent en RO et en IA pour l'optimisation combinatoire et l'affectation sous contraintes. La figure (4-2) met en parallèle les méthodes représentatives développées en RO et en IA, avec à titre indicatif la date approximative d'apparition de chaque méthode.

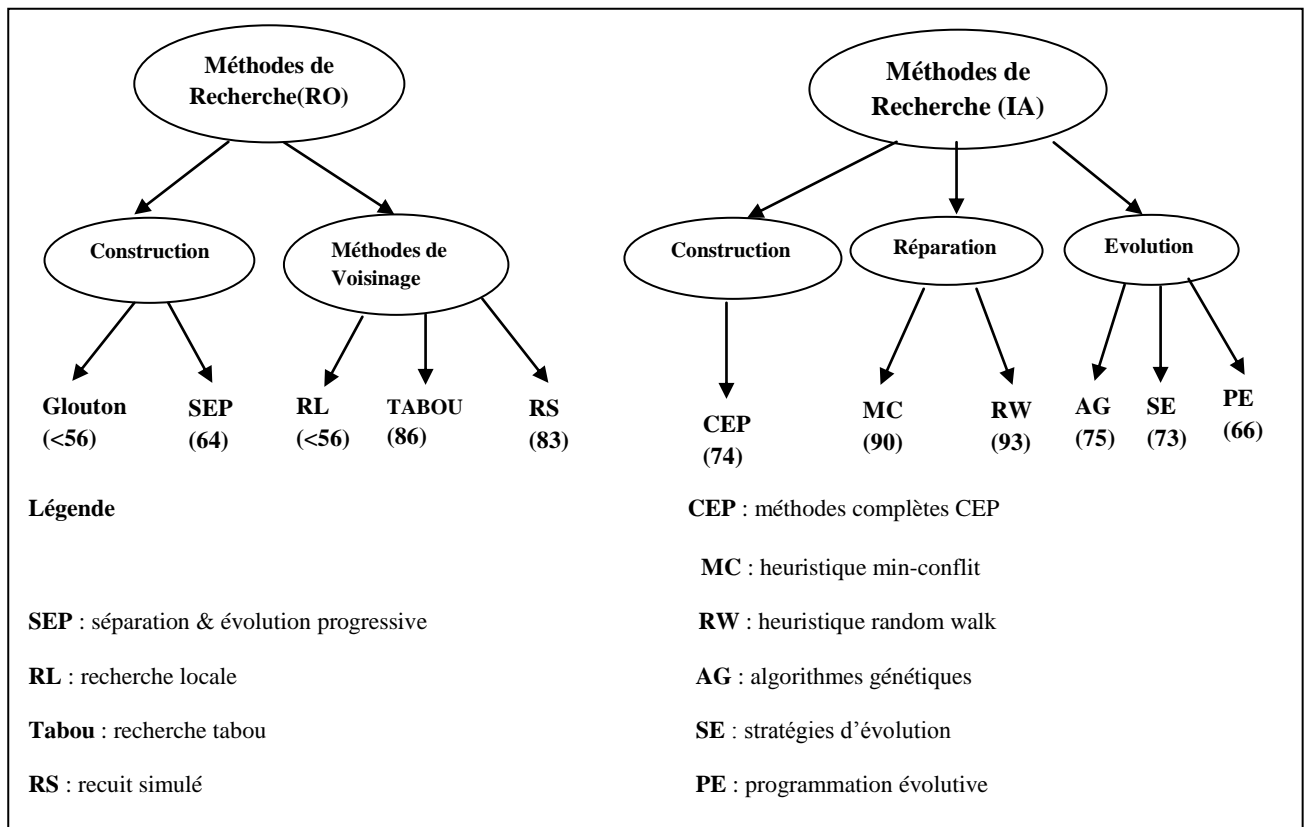


Figure 4.2. Classement des méthodes de résolution

## 4-5) METAHEURISTIQUES

Depuis une dizaine d'années, des progrès importants ont été réalisés avec l'apparition d'une nouvelle génération de méthodes approchées puissantes et générales, souvent appelées *métaheuristiques*. Le mot « heuristique » vient du grec *heurein* (découvrir), *les heuristiques sont des méthodes qui cherchent à approcher une solution optimale*, on les appelle par fois *méthodes approchées* [ERI 07].

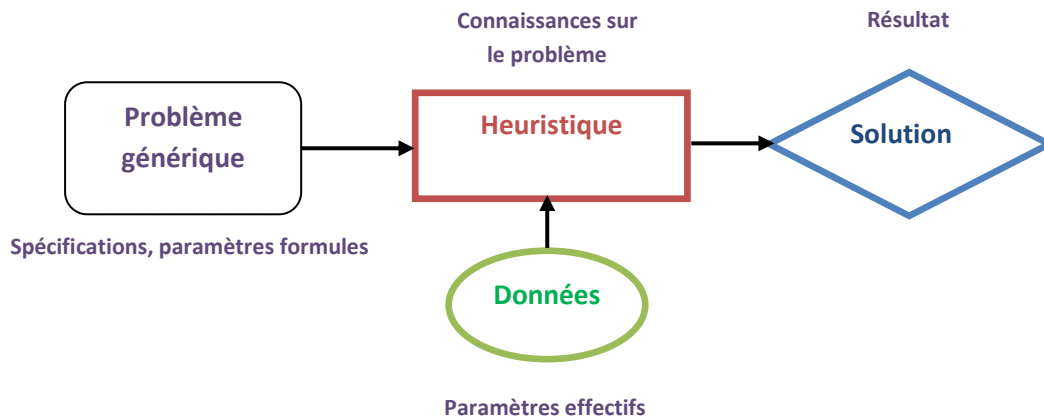


Figure 4.3. Principe des méthodes heuristiques

Une métaheuristique est constituée d'un ensemble de concepts fondamentaux (par exemple, la liste tabou, et les mécanismes d'intensification et de diversification pour le métaheuristique tabou), qui permettent d'aider à la conception de méthodes heuristiques pour un problème d'optimisation. Ainsi les métaheuristiques sont adaptables et applicables à une large classe de problèmes [ERI 07].

Les métaheuristiques sont représentées essentiellement par *les méthodes de voisinage* (le recuit simulé et la recherche tabou) et *les algorithmes évolutifs* (les algorithmes génétiques et les stratégies d'évolution). Grâce à ces métaheuristiques, on peut proposer aujourd'hui des solutions approchées pour des problèmes d'optimisations classiques de grande taille et pour de très nombreuses applications qu'il était impossible de traiter auparavant. On constate, depuis ces dernières années, que l'intérêt porté aux métaheuristiques augmente continuellement en recherche opérationnelles et en intelligence artificielle.

### 4-5-1/ METHODES DE VOISINAGES

Les méthodes de voisinage sont fondées sur la notion de voisinage. Nous avons donc introduire d'abord cette notion fondamentale ainsi que quelques notions associées. Le choix du voisinage en optimisation combinatoire a une grande influence sur le résultat de l'application d'une métaheuristique. C'est d'ailleurs cette constatation qui est à l'origine de la méthode dite à voisinage variable.

La définition d'un voisinage dépend beaucoup du problème considéré. On définit le voisinage d'une solution par une transformation élémentaire permettant de passer une

solution  $S_1$  à une autre solution  $S_2$  proche avec une faible modification de la structure de la solution  $S_1$ . Cette opération est dite mouvement.

#### 4-5-1-1/ RECUIT SIMULE

Le recuit simulé (SA) « Simulated Annealing » [KIR 83, CER 85] trouve ses origines dans la thermodynamique. Cette méthode est issue d'une analogie entre le phénomène physique de refroidissement lent d'un corps en fusion, qui le conduit à un état solide, de basse énergie. Il faut abaisser lentement la température, en marquant les paliers suffisamment longs pour que le corps atteigne l'équilibre thermodynamique à chaque palier de température. Pour les matériaux, cette basse énergie se manifeste par l'obtention d'une structure régulière, comme dans les cristaux et l'acier.

Idée générale : poursuivre tout de même la recherche mais : 1) de façon non déterministe, 2) les mouvements améliorants sont toujours acceptés, 3) plus un mouvement dégrade la solution, plus la probabilité de la choisir est basse [ERI 07].

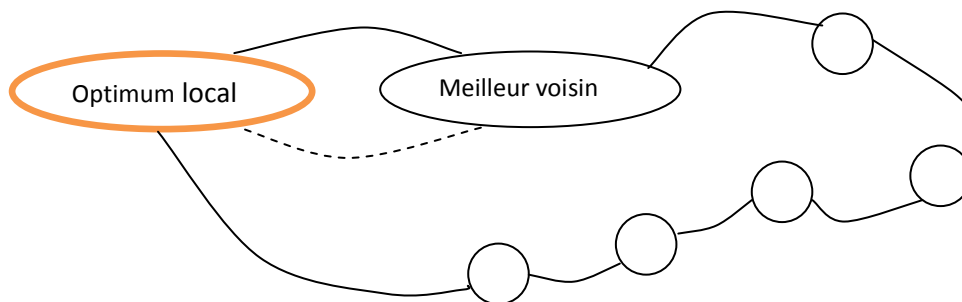


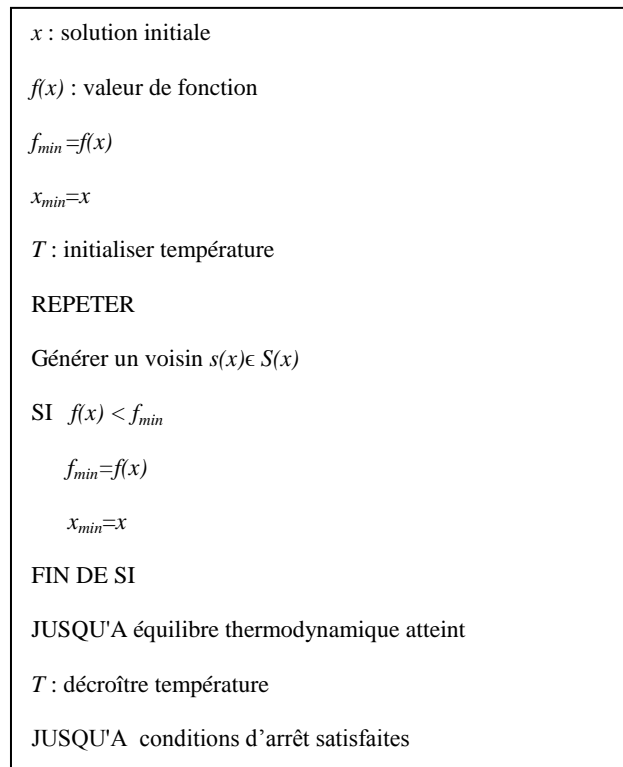
Figure 4.4. Principe de base du recuit simulé

L'analogie exploitée par le recuit simulé consiste à considérer une fonction  $f$  à minimiser comme fonction d'énergie, et une solution  $\mathbf{x}$  peut être considérée comme un état donné de la matière dont  $f(\mathbf{x})$  est l'énergie.

Pour une « température »  $T$  donnée, à partir d'une solution courante  $\mathbf{x}$ , on considère une transformation élémentaire qui changerait  $\mathbf{x}$  en  $s(\mathbf{x})$ . Si cette perturbation induit une diminution de la valeur de la fonction objective  $f$ ,  $\Delta f = f(s(\mathbf{x})) - f(\mathbf{x}) < 0$  elle est acceptée. Dans le cas contraire, si  $\Delta f = f(s(\mathbf{x})) - f(\mathbf{x}) \geq 0$ , la perturbation est acceptée tout de même avec une probabilité  $p = \exp \frac{-\Delta f}{T}$ .

Le paramètre de contrôle de  $T$  est la « température » du système, qui influe sur la probabilité d'accepter une solution plus mauvaise. A une température élevée, la probabilité d'acceptation d'un mouvement quelconque tend vers 1 : presque tous les changements sont acceptés. L'algorithme équivaut alors à une marche aléatoire dans l'espace des configurations. Cette température est diminuée lentement au fur à mesure du déroulement de l'algorithme pour simuler le processus de refroidissement des matériaux, et sa

diminution est suffisamment lente pour que l'équilibre thermodynamique soit maintenu. L'algorithme est présenté dans la figure suivante :



**Figure 4.5.** L'algorithme du recuit simulé

L'efficacité du recuit simulé dépend fortement du choix de paramètres de contrôle qui sont :

- La valeur initiale de la température,
- La fonction de décroissance de la température,
- Le critère de changement de palier de température,
- Les critères d'arrêt.

Pour le calcul de la température de départ, plusieurs méthodes ont été proposées. Une des méthodes est basée sur l'observation de la variation moyenne de la fonction  $f$ . À partir d'une solution initiale  $x_0$ , on génère par transformation élémentaires aléatoires, un certain nombre de solution  $x_0'$  (environ 50 à 100) telles que  $f(x_0') > f(x_0)$ , et on calcule la variation  $\langle \Delta f \rangle_{init}$ . Une température initiale  $T_{init}$  est calculée de façon à accepter au départ une certaine proportion  $P_{init}$  de mouvements dégradant la fonction  $f$ . pour une température initiale « moyenne », la valeur de  $P_{init}$  est de 0.5. La valeur de  $T_{init}$  est déduite de la formule suivante :

$$P_{init} = \exp \frac{-|\langle \Delta f \rangle_{init}|}{T_{init}} \quad (4-1)$$

Le rôle de la température  $T$  au cours du processus de recuit simulé est très important. Un forte décroissant de la température risque de piéger l'algorithme dans un minimum local, alors qu'une faible décroissance au début du processus entraîne une convergence Très lente de l'algorithme. Un compromis pour adapter la décroissance de la température à l'évolution du processus consiste à utiliser une variation logarithmique. La loi logarithmique de décroissance de la température, qui assure la convergence théorique du recuit simulé, est la suivante :

$$T_k = \frac{\mu}{\log(1+k)} \quad (4-2)$$

Où  $k$  est le nombre de paliers de températures effectuées, et  $\mu$  une constante positive. En pratique, on adapte souvent une décroissance géométrique  $T_{k+1} = \alpha T_k$ , avec ( $0 < \alpha < 1$ ), car la loi précédente induit un temps de calcul prohibitif.

Pour le changement de palier de température, on peut simplement spécifier un nombre de transformations, acceptées ou non, au bout duquel la température est abaissée.

#### 4-5-1-2/ RECHERCHE TABOU

La Recherche Tabou (**RT**) « *tabu search* » est une métaheuristique originalement développée par Glover [GLO 86] et indépendamment par Hansen [HAN 86], sous l'appellation de « *steepest ascent mildest descent* ». Elle est basée sur des idées simples, mais elle est néanmoins très efficace. Cette méthode combine une procédure de recherche locale avec un certain nombre de règles et de mécanismes permettant à celle-ci de surmonter l'obstacle des optima locaux, tout en évitant de cycliser. Elle a été appliquée avec succès pour résoudre de nombreux problèmes difficiles d'optimisation combinatoire.

Idée générale : poursuivre la recherche au – delà du premier optimum local en choisissant le meilleur voisin. Utilise une mémoire pour guider le processus de recherche dans le but d'éviter de retourner à une solution déjà visitée.

Mémoriser les solutions déjà visitées, ou mémoriser l'inverse de mouvements récemment effectués, ou mémoriser des caractéristiques de solutions ou de mouvements récemment effectués. Les composantes mémorisées deviennent interdites (taboues) [ERI 07].

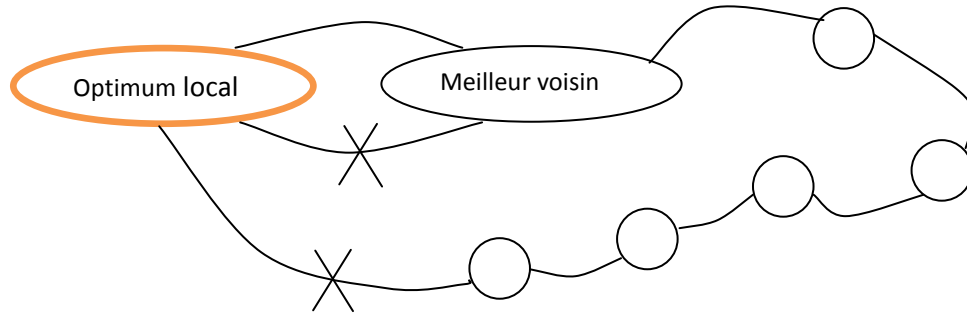


Figure 4.6. Principe de base de la recherche tabou

Dans une première phase, la méthode de Recherche Tabou peut être vue comme une généralisation des méthodes d'amélioration locales. En effet, en partant d'une quelconque  $x$  appartenant à l'ensemble de solutions  $X$ , on se déplace vers une solution  $s(x)$  située dans le voisinage de  $[S(x)]$  de  $x$ . Donc l'algorithme explore itérativement l'espace de solutions  $X$ .

Afin de choisir le meilleur voisin  $s(x)$  dans  $S(x)$ , l'algorithme évalue la fonction objective  $f$  en chaque point  $s(x)$ , et retient le voisin qui améliore la valeur de la fonction objectif  $f$ , ou au pire celui qui la dégrade le moins.

L'originalité de la méthode de Recherche Tabou, par rapport aux méthodes locales, qui s'arrêtent dès qu'il n'y a plus de voisin  $s(x)$  permettent d'améliorer la valeur de la fonction objectif  $f$ , réside dans le fait que l'on retient le meilleur voisin, même si celui-ci est plus mauvais que la solution d'où l'on vient. Ce critère autorisant les dégradations de la fonction objectif évite à l'algorithme d'être piégé dans un minimum local. Mais il induit un risque de cyclage. En effet, lorsque l'algorithme a quitté un minimum quelconque par acceptation de la dégradation de la fonction objectif, il peut revenir sur ses pas, à l'itération suivante.

Pour régler ce problème, l'algorithme a besoin d'une mémoire pour conserver pendant un moment la trace des dernières meilleures solutions déjà visitées. Ces solutions sont déclarées *tabou*, d'où le nom de la méthode. Elles sont stockées dans une liste de longueur  $L$  données, 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 cyclage 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. L'algorithme de la recherche tabou est présenté dans la figure (4.7).

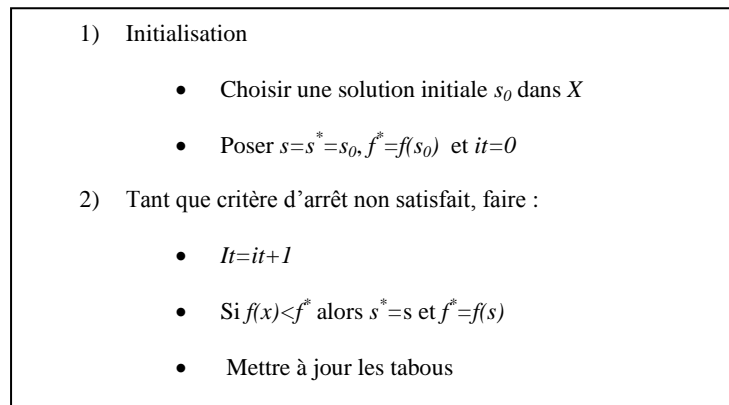
La liste tabou est généralement gérée comme une liste « circulaire » : on élimine à chaque itération la solution la plus ancienne, en la remplaçant par la nouvelle solution retenue. Mais le codage d'une telle liste est encombrant, car il faudrait garder en mémoire tous les éléments qui définissent une solution. Pour pallier cette contrainte, on remplace la liste tabou de solutions interdites par une liste de « transformations interdites », en introduisant la transformation inverse d'une transformation faite récemment.



La liste taboue varie selon le problème à traiter, c'est une donnée primordiale. Si la liste taboue est trop petite, elle risque de conduire à un cyclage, et si elle est trop grande elle peut interdire des transformations intéressantes.

#### 4-5-1-2-1/ Critère d'aspiration

Le remplacement de la liste tabou des solutions visitées par la liste des transformations élémentaires  $\{x, s(x)\}$  conduit non seulement à l'interdiction de revenir vers des solutions précédentes (on évite le cyclage court), mais aussi vers un ensemble de solutions dont plusieurs peuvent ne pas avoir été visitées jusqu'ici. Il est donc primordial de corriger ce défaut et de trouver un moyen de lever l'interdiction de l'acceptation d'une transformation élémentaire  $\{x, s(x)\}$  déjà effectuée (donc appartenant à la liste tabou), sous un certain critère, appelé *critère d'aspiration*. Cette correction permet aussi de revenir à une solution déjà visitée et de redémarrer la recherche dans une autre direction. Cette idée est développée dans [GLO 89, GLO 95, GLO 97].



**Figure 4.7.** L'algorithme de la recherche tabou

Le critère d'aspiration le plus simple et le plus couramment utilisé consiste à tester si la solution produite de statut tabou présente un coût inférieur à celui de la meilleure solution trouvée jusqu'à présent. Si cette situation se produit, le statut tabou de la solution est levé. Ce critère est évidemment très sévère, il ne devrait pas être vérifié très souvent, donc il apporte peu de changements à la méthode. D'autres critères d'aspiration plus complexes peuvent être envisagés. L'inconvénient de recourir trop souvent à l'aspiration est qu'elle peu détruire, dans une certaine mesure, la protection offerte par la liste tabou vis-à-vis du cyclage [Lah 07].

Notons que, dans le cas d'une liste tabou de solutions, le concept de critère d'aspiration n'est pas intéressant. Toute annulation du statut tabou d'une solution se trouve dans la liste tabou pourrait conduire l'algorithme au cyclage.

#### **4-5-1-2-2/ Intensification**

L'intensification consiste à approfondir la recherche dans certaines régions du domaine, identifiées comme susceptible de contenir un optimum global. Cette intensification est appliquée périodiquement, et pour une durée limitée. Pour mieux intensifier la recherche dans une zone bien localisée, plusieurs stratégies sont proposées dans la littérature.

La plus simple consiste à retourner à l'une des meilleures solutions trouvée jusqu'à présent, puis de reprendre la recherche à partir de cette solution, en réduisant la longueur de la liste tabou pour un nombre limité d'itérations. Dans ce cas, on adapte la procédure de recherche tabou, en élargissant le voisinage de la solution courante (en augmentant la taille de l'échantillon  $S(x)$ ), tout en diminuant le pas des transformations. On peut aussi remplacer simplement l'heuristique tabou par une autre méthode plus puissante, ou mieux adaptée, pour une recherche locale.

#### **4-5-1-2-3/ Diversification**

La diversification permet à l'algorithme de bien explorer l'espace des solutions, et d'éviter que le processus de recherche ne soit trop localisé et laissé de grandes régions du domaine totalement inexplorées. La plus simple des stratégies de diversification consiste à interrompre périodiquement l'acheminement normal de la procédure tabou, et à la faire redémarrer à partir d'une autre solution, choisie aléatoirement, ou « intelligemment ». Une autre méthode consiste à biaiser la fonction d'évaluation  $f$ , en introduisant un terme qui pénalise les transformations effectuées fréquemment, afin de favoriser des transformations nouvelles. Ce type de stratégies de diversification peut être utilisé de façon continue, sans interrompre la procédure de Recherche Tabou.

En résumé, nous dirons que la diversification et l'intensification sont des concepts complémentaires, qui enrichissent la méthode de Recherche Tabou et la rendent plus robuste et plus efficace [LAHk 07].

#### **4-5-1-2-4/ Exemple d'optimisation des fonctions par la Recherche Tabou**

##### **❖ Fonction à une variable :**

Soit la fonction  $F_1$  à une variable à maximiser dans l'intervalle [7.45 7.55],  
 $F_1(x) = 15x - x^2$

##### **❖ Fonction à deux variables à minimiser :**

Soit la fonction  $F_2$  à deux variables à minimiser dans l'intervalle ( $x \in [-0.2 \ 0.2]$  et  $y \in [-0.2 \ 0.2]$ ),  $F_2(x, y) = x^2 + 2y^2 - 0.3\cos(3\pi x) - 0.4\cos(4\pi y) + 0.7$

Soit la fonction  $F_3$  à deux variables à minimiser dans l'intervalle ( $x \in [-5 \ 5]$  et  $y \in [-5 \ 5]$ ),  $F_3(x, y) = 1 - \frac{1}{2}(x^2 + y^2)$

Soit la fonction  $F_4$  à deux variables à minimiser dans l'intervalle  $(x \in [4 \ 6] \text{ et } y \in [4 \ 6])$ ,  $F_4(x, y) = (x - y)^2 + \left[ \frac{(x + y - 10)}{3} \right]^2$

Les paramètres de contrôle de la recherche tabou sont :

- Nombre d'itération :  $i_{\text{max}} = 200$
- Longueur de la liste taboue :  $L = 5$
- Nombre de points de recherche autour du voisinage:  $M = 10$
- Nombre de diversifications:  $i_{\text{diver}} = 1$

Les résultats de simulation des fonctions sont :

$$F_1 = 56.2500$$

$$x = 7.5013$$

$$F_2 = 0.0125$$

$$x = 0.0073$$

$$y = 0.0188$$

$$F_3 = -22.9386$$

$$x = 4.9088$$

$$y = 4.8765$$

$$F_4 = 0.0052$$

$$x = 5.0365$$

$$y = 5.0938$$

Les courbes de la variation des fonctions  $F_2$ ,  $F_3$  et  $F_4$  sont respectivement :

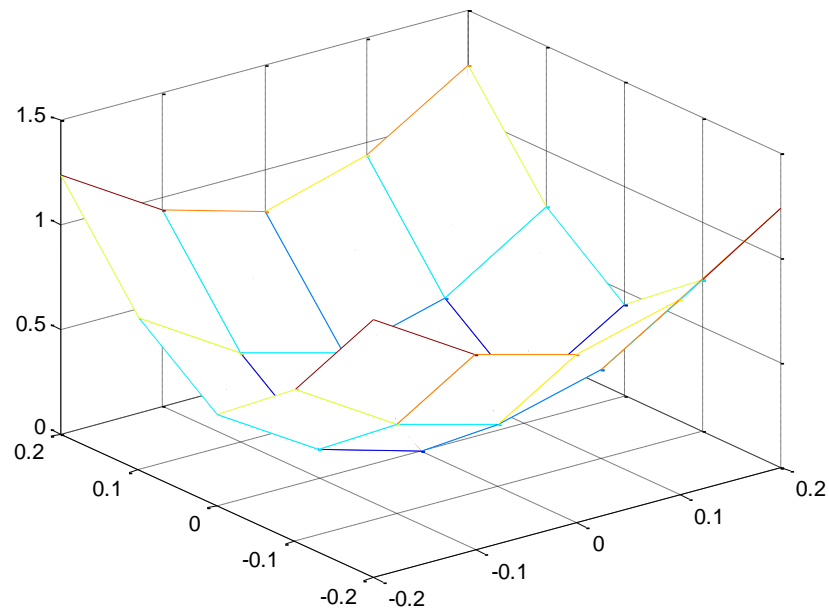


Figure 4.8. Graphe de la fonction  $F_3$

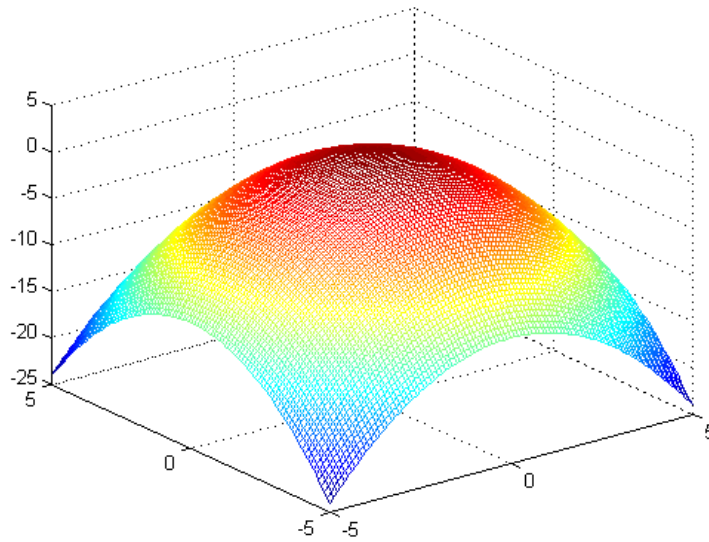


Figure 4.9. Graphe de la fonction  $F_2$

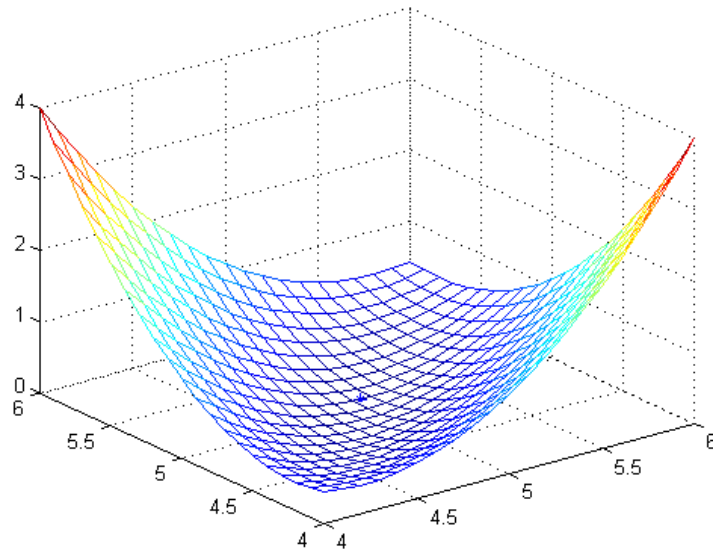


Figure 4.10. Graphe de la fonction  $F_4$

#### 4-5-2/ ALGORITHMES EVOLUTIFS

Le terme « algorithmes évolutifs » englobe une autre classe assez large de métaheuristiques. Ces algorithmes sont basés sur le principe du processus d'évolution naturelle [DEJ 93, BÄC 93a, SCH 97b]. Les méthodes évolutives doivent leur nom à l'analogie avec les mécanismes d'évolution des espèces vivantes.

Un algorithme évolutif est composé de trois éléments essentiels : 1) *une population* constituée de plusieurs individus représentant des solutions potentielles (configurations) du problème donné ; 2) un *mécanisme d'évolution* de l'adaptation de chaque individu de la population à l'égard de son environnement extérieur ; 3) un *mécanisme d'évolution* composé d'opérateurs permettant d'éliminer certains individus et de produire de nouveaux individus à partir des individus sélectionnés.

Du point de vue opérationnel, un algorithme évolutif débute avec une population initiale souvent générée aléatoirement et répète ensuite un cycle d'évolution composé de 3 étapes séquentielles : 1) mesurer l'adaptation (la qualité) de chaque individu de la population par le mécanisme d'évolution, 2) sélectionner une partie des individus, et 3) produire de nouveaux individus par des recombinaisons d'individus sélectionnés. Ce processus se termine quand la condition d'arrêt est vérifiée.

On peut distinguer trois grandes familles d'algorithmes évolutifs : les algorithmes génétiques, la programmation évolutive et les stratégies d'évolution. Ces méthodes se différencient par leur manière de représenter les données et par leur façon de faire évoluer la population d'une génération à l'autre.

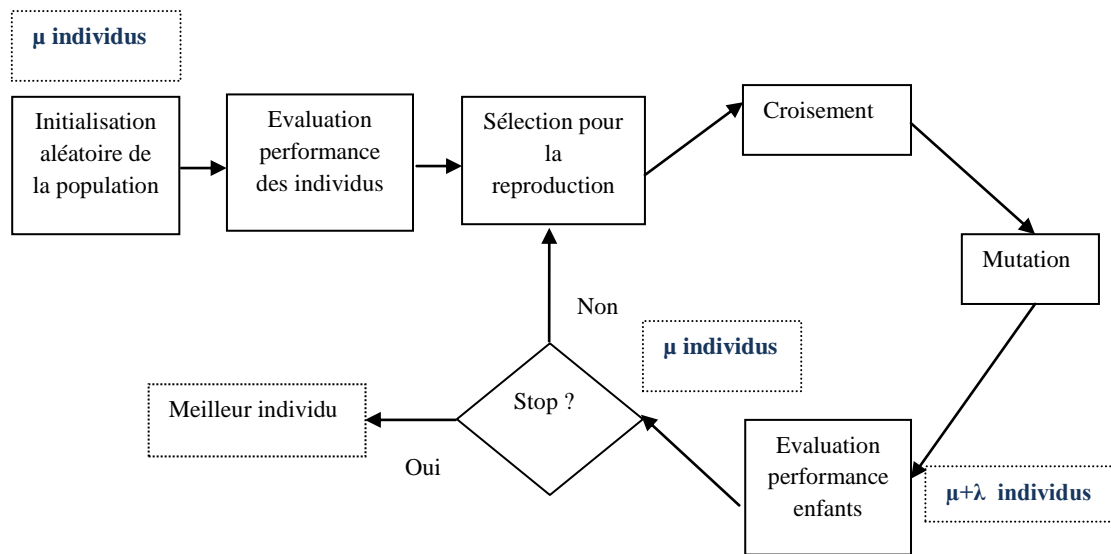


Figure 4.11. Schéma de principe des algorithmes évolutionnaires

#### 4-5-2-1/ ALGORITHMES GENETIQUES

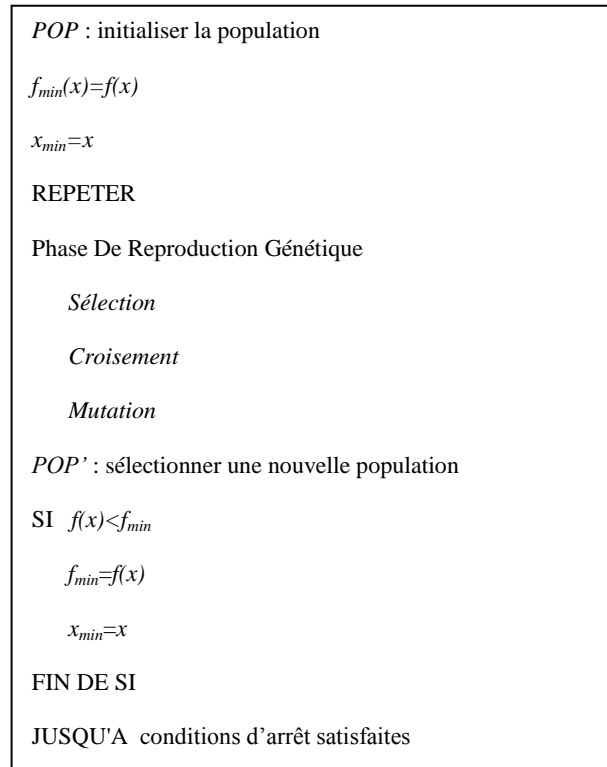
Les principes fondamentaux des Algorithmes Génétiques (AG) « *genetic algorithms* » ont été exposés par Holland [HOL 75]. Ces algorithmes s'inspirent du fonctionnement de l'évolution naturelle, notamment la sélection de Darwin, et la procréation selon les règles de Mendel. La sélection naturelle, que Darwin appelle l'élément « propulseur » de l'évolution, favorise les individus d'une population qui sont le mieux adaptés à un *environnement*. La sélection est suivie de la procréation, réalisée à l'aide de croisements et de mutations au niveau du patrimoine génétique des individus (ou « génotype ») [HOL 75, 92, GOL 89, DEJ 94], constitué d'un ensemble de gènes. Ainsi deux individus « parents », qui se croisent, transmettent une partie de leur patrimoine génétique à leurs descendants. Le génotype de l'enfant que celui-ci est plus au moins bien adapté à l'environnement. S'il est bien adapté, il a une plus grande chance de procréer dans la génération future. Au fur et à mesure des générations, on sélectionne les individus les mieux adaptés, et l'augmentation du nombre des individus bien adaptés fait évoluer la population entière.

Nous présentons d'abord le principe de base des algorithmes génétiques, puis nous décrirons les différents codages et nous présentons ensuite les opérateurs génétiques.

Dans les Algorithmes Génétiques, on essaie de simuler le processus d'évolution d'une population. On part d'une population  $N$  solutions du problème présentées par des individus. Cette population choisie aléatoirement est appelée initiale. Le degré d'*adaptation* d'un individu à l'environnement est exprimé par la valeur de la fonction coût (fonction objectif)  $f(\mathbf{x})$ , où  $\mathbf{x}$  est la solution que l'individu représente. On dit qu'un individu est d'autant mieux adapté à son environnement, que le coût de la solution qu'il représente est plus faible. Au sein de cette population, intervient alors la *sélection* au hasard d'un ou deux parents, qui produisent une nouvelle solution (nouvelle population), à

travers les *opérateurs génétiques*, tels que le *croisement* et la *mutation*. En itérant ce processus, on produit une population plus riche en individus mieux adaptés.

La figure (4-12) présente le principe d'un algorithme génétique de base.



**Figure 4.12.** L'algorithme génétique de base.

Cet algorithme comporte trois phases distinctes :

- ✓ La production de la population d'individus la mieux adaptée pour contribuer à la reproduction de la génération suivante (version artificielle de la sélection naturelle) ;
- ✓ La phase de reproduction, qui exploite essentiellement les opérateurs de croisement et de mutation ;
- ✓ La stratégie de remplacement des populations parent et enfant par la génération suivante.

#### **4-5-2-1-1/ Codage des individus**

Plusieurs codes d'informations sont utilisés. Les plus fréquemment utilisés sont le code *binaire naturel*, le code *binaire en Gray* et le code *réel*.

**a. Codage binaire**

Ce codage consiste, pour un individu donné, à concaténer toutes ses variables codées en binaire. La chaîne binaire (par exemple 1000/0110/1101), correspond à un individu défini par 3 variables (8, 6,13) en codage binaire naturel sur 4bits chacune. Ce codage binaire présente plusieurs avantages : alphabet minimum, facilité de mise en place. Néanmoins, ce codage présente trois inconvénients majeurs :

- ✓ Les performances de l'algorithme sont diminuées lorsque la longueur de chaîne augmente ;
- ✓ Deux nombres décimaux voisins (exemple : 7 et 8) peuvent être très éloignés dans le codage binaire (1000 et 0111) : falaise de Hamming. Ce problème peut être réglé en remplaçant le code binaire naturel par le code binaire de Gray.

**b. Codage réel**

Ce codage consiste simplement à la concaténation des variables  $x_i$  d'un individu  $x$ . Exemple un individu  $x$  (25, 31,8) est codé 25/31/8. Ce codage présente des avantages majeurs. Il est plus précis que le codage binaire et l'espace de recherche est le même que l'espace de problème. Il a le mérite d'être simple à utiliser, l'évaluation de la fonction coût est plus rapide. Le codage réel évite de faire le transcodage du binaire naturel ou de Gray vers les réels à chaque évaluation.

**4-5-2-1-2/ Opérateurs génétiques**

La phase de reproduction exploite principalement deux opérateurs : le croisement et la mutation. Elle comporte aussi l'opérateur de sélection et l'opérateur de production de la génération suivante, à partir des populations parent et enfant.

** Opérateur de sélection**

La sélection consiste à choisir les paires d'individus qui vont participer à la reproduction de la population future. La fonction de sélection calcule une probabilité de sélection pour chaque individu, en fonction de sa qualité et de tous Les autres individus dans la population. On trouve dans la littérature trois principes de sélection courante: Sélection ordonnée, Roulette biaisée et la Sélection uniforme. [DOC a]



- a) **Sélection Ordonnée (Rank Selection)** : Chaque individu de la population évolue par la fonction d'adaptation. Les valeurs de la fonction d'adaptation obtenues pour l'ensemble des individus seront classées dans un ordre croissant ou décroissant. Les meilleurs individus seront donc sélectionnés.
- b) **Roulette Biaisée (Roulette Wheel Selection)** : Cette méthode attribue à chaque individu une probabilité de survie «  $pi$  » proportionnelle à son adaptation «  $fi$  » dans la population. [HAN, DOC a, DOC c]

$$pi = \frac{f(x_i)}{\sum_{j=1}^{Npop} f(x_j)} \quad (4-3)$$

Où :

$\sum_{j=1}^{Npop} f(x_j)$  représente la somme de toutes les valeurs des fonctions d'adaptation de chaque individu de la population.

$Npop$  : représente la taille de la population.

Lors de la phase de sélection, les individus sont sélectionnés aléatoirement en respectant les probabilités «  $pi$  » associées pour former la population de la nouvelle génération. Ceci s'effectue pour le calcul d'une probabilité de sélection cumulée «  $qi$  » telle que :

$$qi = \sum_{j=1}^i pi \quad (4-4)$$

Puis en génère aléatoirement un réel «  $r$  » sur l'intervalle  $[0,1]$   $Npop$  fois de suite. L'individu est sélectionné lorsque  $qi-1 < r < qi$ .

- c) **Sélection Uniforme (Uniform Selection)** : C'est une technique très simple qui consiste à sélectionner un individu aléatoirement de la population. La probabilité «  $pi$  » pour qu'un individu soit sélectionné est définie par [ DOC c] :

$$pi = \frac{1}{N_{pop}} \quad (4-5)$$

#### Opérateur de croisement

Principal opérateur agissant sur la population des parents est le croisement, qui est appliqué avec une certaine probabilité, appelée taux de croisement notée  $p_c$

(typiquement proche de l'unité). Le croisement consiste à choisir deux individus représentés par leurs chaînes de gènes, tirés au hasard dans la population courante, et à définir aléatoirement un ou plusieurs points de croisement. Les nouvelles chaînes sont alors créées en interchangeant les différentes parties de chaque chaîne. Il existe plusieurs types de croisement, les plus courants sont :

- a) **croisement binaire en un point** : C'est le plus simple et le plus classique pour des codages utilisant un alphabet à faible cardinalité, comme le codage binaire. On choisit au hasard un point de croisement, pour chaque couple (Figure 4-13). Notons que le croisement s'effectue directement au niveau binaire, et non pas au niveau des gènes. Un chromosome peut donc être coupé au milieu d'un gène [DOC b].

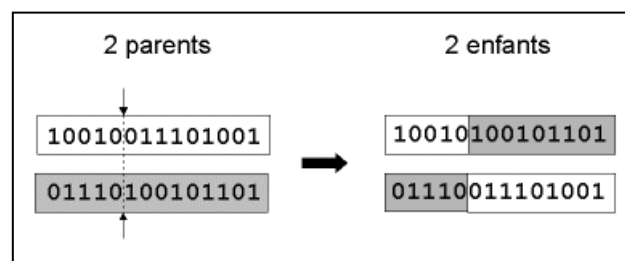


Figure 4.13. Technique du croisement en 1 point

- b) **croisement binaire en deux points** : On choisit au hasard deux points de croisement pour chaque couple (Figure 4-14) [DOC b].

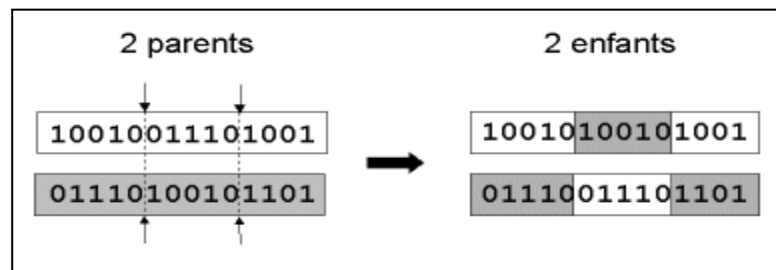
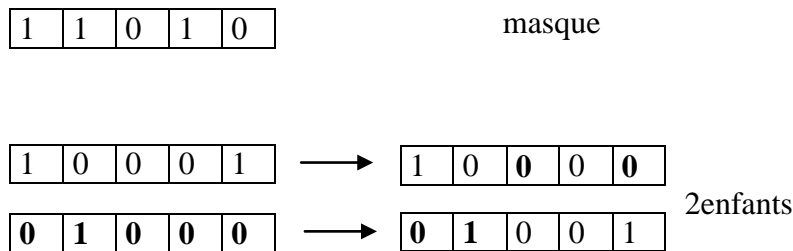


Figure 4.14. Technique du croisement en 2 points

- c) **Croisement uniforme** : Cette technique est complètement différente des deux techniques précédentes. Un masque de croisement est généré aléatoirement pour chaque couple d'individu ou pour chaque génération. Les valeurs de ce masque sont binaires, sa taille est identique à celle du chromosome.

Le fonctionnement du croisement uniforme est le suivant : Si la valeur du bit du masque est égale à 1, la valeur du gène du parent 1 est copiée chez l'enfant 1, et si la valeur du bit du masque est égale à 0, la valeur du gène de parent 2 est copiée à

l'enfant 1. Les valeurs des gènes de l'enfant 2 sont les suivantes : les valeurs du gène du parent 1 lorsque la valeur du bit du masque est égale à 0 et les valeurs du gène du parent 2 lorsque la valeur du bit du masque est égale à 1 [DOC c].



**Figure 4.15.** Technique de croisement uniforme

### **Croisement réel**

Le croisement réel ne se différencie du croisement binaire que par la nature des éléments qu'il altère : ce ne sont plus des bits qui sont échangés à droite du point de croisement, mais des variables réelles. [DOC b]

Soit, après la sélection des deux parents  $P_1$  et  $P_2$ , deux nouveaux enfants (descendants) sont créés  $C_1$  et  $C_2$  :

$$\begin{cases} C_1 = \alpha P_1 + (1 - \alpha) P_2 \\ C_2 = (1 - \alpha) P_1 + \alpha P_2 \end{cases} \quad (4-6)$$

Avec «  $\alpha$  » est un chiffre généré aléatoirement.

### **Opérateur de mutation**

L'opérateur de mutation protège les algorithmes génétiques des prématurées d'informations pertinentes. Elle permet d'introduire une certaine information dans la population, qui a pu être perdue lors de l'opération de croisement. Ainsi elle participe au maintien de la diversité, utile à une bonne exploration du domaine de recherche. L'opérateur de mutation s'applique avec certaine probabilité, appelée taux de mutation notée  $p_m$ , typiquement compris entre 0.05 et 0.10. Ce faible taux de mutation permet de dire que la mutation est considérée comme un mécanisme d'adaptation secondaire pour les algorithmes génétiques. Dans le codage binaire, la mutation consiste à changer un bit 1 par le bit 0 et *vice versa*, pour chaque bit de la chaîne, avec une probabilité  $p_m$ . La figure (4-16) représente l'opération de mutation pour un individu représenté par une chaîne binaire codée sur 14 bits. La

probabilité de mutation  $p_m$  est indépendante de la longueur du codage des variables des individus.

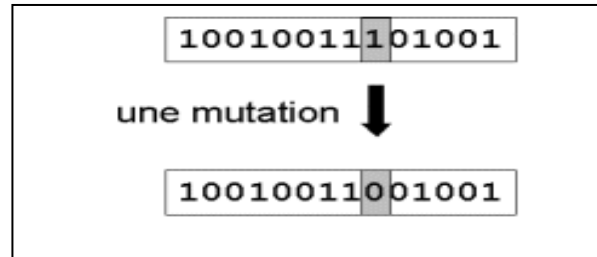


Figure 4.16. Technique de mutation

#### 4-5-2-1-3/ Exemple d'optimisation des fonctions par un Algorithme Génétique simple :

On applique la méthode d'un Algorithme Génétique simple pour calculer le maximum de la fonction  $F_1$  d'une seule variable et le minimum des fonctions  $F_2$ ,  $F_3$  et  $F_4$  à deux variables utilisées dans le paragraphe (4-5-1-2) :

Les paramètres de contrôle de l'algorithme génétique sont :

- Type de codage : Codage binaire
- Taille de la population :  $N_{pop} = 30$
- Probabilité de croisement :  $pc = 0.75$
- Croisement double
- Probabilité de mutation :  $pm = 0.06$
- Mutation simple
- La longueur de chromosome :  $l = 12$  bits
- Nombre de génération :  $maxgen = 300$

$$F_1(x) = 15x - x^2$$

$$F_2(x, y) = x^2 + 2y^2 - 0.3\cos(3\pi x) - 0.4\cos(4\pi y) + 0.7$$

$$F_3 = 1 - \frac{1}{2}(x^2 + y^2)$$

$$F_4(x, y) = (x - y)^2 + \left[ \frac{(x + y - 10)}{3} \right]^2$$

Les résultats de simulation des fonctions sont :

$$F_1 = 56.2475$$

$$x = 7.55$$

$$F_2 = 3.8057 \cdot 10^{-4}$$

$$x = 0.0035$$

$$y = 0.0025$$

$$F_3 = -23.7081$$

$$x = 4.9878$$

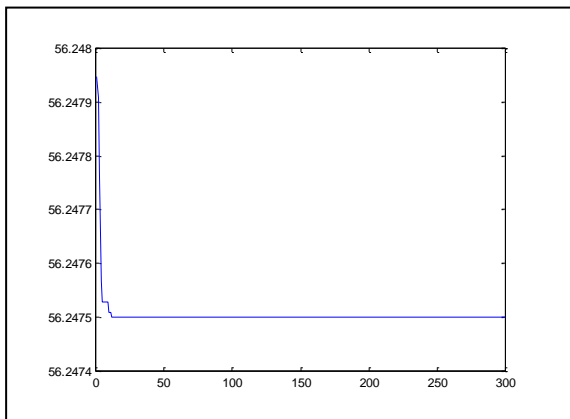
$$y = 4.9536$$

$$F_4 = 0.006023$$

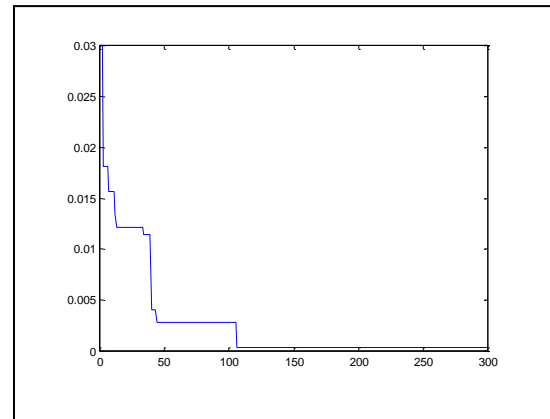
$$x = 5.0159$$

$$y = 5.0339$$

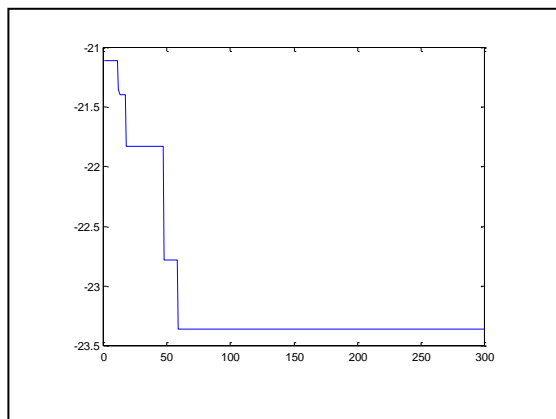
Les courbes de variation de chaque fonction vs le nombre d'itération sont respectivement :



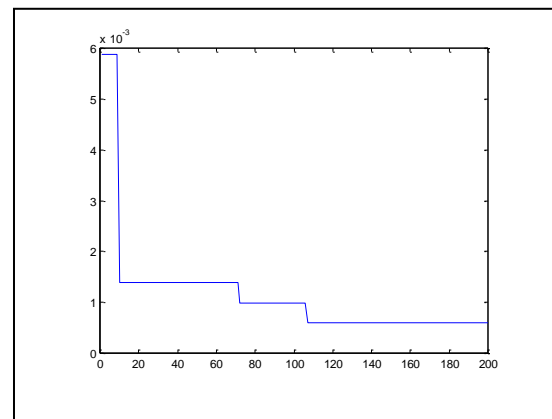
**Figure 4.17.** Graphe de la fonction  $F_1$



**Figure 4.18.** Graphe de la fonction  $F_2$



**Figure 4.19.** Graphe de la fonction  $F_3$



**Figure 4.20.** Graphe de la fonction  $F_4$

Le tableau suivant donne une comparaison entre les quatre fonctions citées ci-dessus traitées par deux algorithmes, l'algorithme génétique et la recherche tabou.

	AG		RT			
	$X_{opt}$		$F_{opt}$	$X_{opt}$		$F_{opt}$
$F_1$	7.55		56.2475	7.5013		56.2500
$F_2$	0.0035	0.0025	3.8057e-4	0.0073	0.0188	0.0125
$F_3$	4.9878	4.9536	-23.7081	4.9088	4.8765	-22.9386
$F_4$	5.0159	5.0339	0.006023	5.0365	5.0938	0.0052

**Tableau 4.1.** Résultats d'optimisation des quatre fonctions

### Discussions des résultats :

Les deux méthodes ont donné presque les mêmes résultats pour la maximisation de la première fonction ( $F_1$ ). La minimisation des trois dernières fonctions par la Recherche Tabou a donné des bons résultats en comparant avec les algorithmes génétiques.

### 4-5-2-2/ PROGRAMMATION EVOLUTIVE

La programmation évolutive (**EP**) « *evolutionary programming* », s'appuie sur un codage approprié du problème à résoudre et sur les opérations de mutation adaptées au codage [FOG 66, 94]. Le codage d'un tel algorithme dépend du problème à résoudre. Par exemple pour un problème d'optimisation dans le domaine des réels, les individus d'une population seraient des vecteurs de réels.

Un cycle d'évolution typique pour la programmation évolutive est le suivant : chaque configuration de la population courante est copiée dans une nouvelle population. Les configurations sont ensuite mutées, conduisant à de nouvelles configurations. L'ensemble des configurations entre ensuite dans une étape de compétition pour survivre dans la génération suivante.

La programmation évolutive a été initialement introduite pour simuler l'intelligence qui est définie sur l'hypothèse suivant : la caractéristique principale de l'intelligence est la capacité d'adaptation comportementale d'un organisme à son environnement [FOG 66]. Selon un modèle très simpliste, cette tâche de simulation revient à prédire une séquence de symboles appartenant à un alphabet fini à partir des séquences déjà observées. Dans ce but, des automates d'états finis sont choisis pour représenter les individus d'une population. Ainsi, à chaque automate de la population est donnée une série de symboles pris dans une séquence déjà observée et la sortie de l'automate est mesurée par rapport au résultat déjà connu. Cette mesure constitue l'adaptation de l'individu. L'étape suivante consiste à créer, pour chaque individu, un individu enfant par une mutation aléatoire de l'individu parent.

La mutation est assurée par une des opérations suivantes : changer le symbole d'une sortie, changer un état de transition, ajouter ou supprimer un état et changer l'état initial.

Aujourd'hui, la programmation évolutive s'est adaptée à l'optimisation combinatoire et a produit des résultats intéressants pour certains problèmes [FOG 94, BÄC 93b].

#### **4-5-2-3/ STRATEGIES D'EVOLUTION**

Les stratégies d'évolution (*ES*) « *evolution strategies* » [REC 73, SCH 81, BÄC 91, 95] sont conçues dès le départ pour résoudre des problèmes d'optimisation continus. Dans un algorithme de la stratégie d'évolution (SE), les individus sont des points (vecteurs réels). Comme la programmation évolutive, les SE n'utilisent que la mutation et la sélection.

L'algorithme le plus simple, noté (1+1)-ES, manipule un seul individu. A chaque génération (itération), l'algorithme génère par mutation un individu enfant à partir de l'individu parent et sélectionne l'un ou l'autre pour le conserver dans la population (selon l'adaptation de chaque individu). Le processus s'arrête quand la condition d'arrêt est vérifiée, définie souvent par le nombre d'itérations, le temps de calcul réalisé ou l'écart entre deux individus de deux itérations successives.

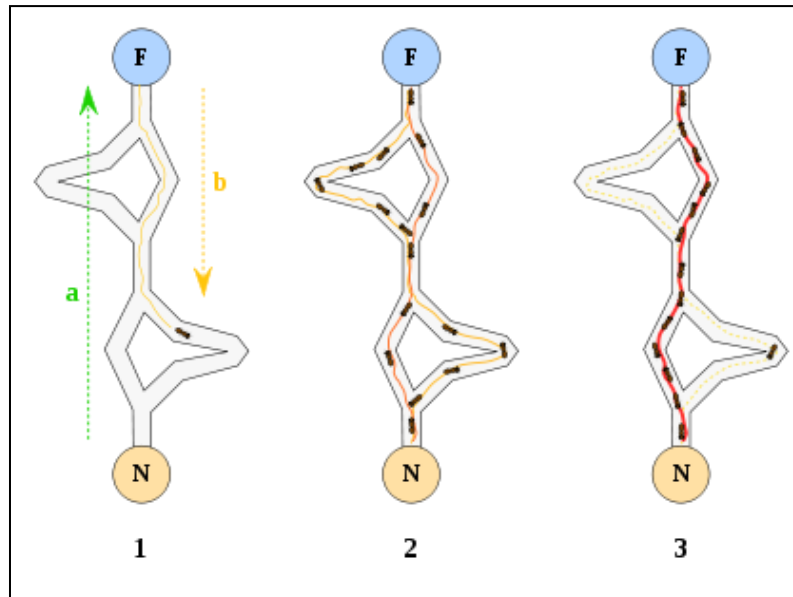
Cet algorithme (1+1)-ES se généralise en un seul algorithme (m+1)-ES qui signifie que « m » parents génèrent 1 enfant à chaque génération et qu'une sélection ramène ensuite la population de m+1 individus à un m individus [BÄC 91]. De plus, la recombinaison a été également introduite dans ces algorithmes [BÄC 93a].

#### **4-5-3) LES ALGORITHMES DE COLONIES DE FOURMI**

Les algorithmes de colonies de fourmis (*ACO*) « Ant Colony Optimization » sont des algorithmes inspirés du comportement des fourmis et qui constituent une famille de métaheuristiques d'optimisation.

Initialement proposé par Marco Dorigo *et al.* dans les années 1990 [COL 91, DOR 92], pour la recherche de chemins optimaux dans un graphe, le premier algorithme s'inspire du comportement des fourmis recherchant un chemin entre leur colonie et une source de nourriture. L'idée originale s'est depuis diversifiée pour résoudre une classe plus large de problèmes et plusieurs algorithmes ont vu le jour, s'inspirant de divers aspects du comportement des fourmis.

L'idée originale provient de l'observation de l'exploitation des ressources alimentaires chez les fourmis. En effet, celles-ci, bien qu'ayant individuellement des capacités cognitives limitées, sont capables collectivement de trouver le chemin le plus court entre une source de nourriture et leur nid.



**Figure 4.21.** Exemple du mécanisme de fourmi

1) la première fourmi trouve la source de nourriture (F), via un chemin quelconque (a), puis revient au nid (N) en laissant derrière elle une piste de phéromone (b). 2) les fourmis empruntent indifféremment les quatre chemins possibles, mais le renforcement de la piste rend plus attractif le chemin le plus court. 3) les fourmis empruntent le chemin le plus court, les portions longues des autres chemins perdent leur piste de phéromones.

Des biologistes ont ainsi observé, dans une série d'expériences menées à partir de 1989 [GOS 89, DEN 90], qu'une colonie de fourmis ayant le choix entre deux chemins d'inégale longueur menant à une source de nourriture avait tendance à utiliser le chemin le plus court.

Un modèle expliquant ce comportement est le suivant :

1. une fourmi (appelée « éclaireuse ») parcourt plus ou moins au hasard l'environnement autour de la colonie ;
2. si celle-ci découvre une source de nourriture, elle rentre plus ou moins directement au nid, en laissant sur son chemin une piste de phéromones ;
3. ces phéromones étant attractives, les fourmis passant à proximité vont avoir tendance à suivre, de façon plus ou moins directe, cette piste ;
4. en revenant au nid, ces mêmes fourmis vont *renforcer* la piste ;
5. si deux pistes sont possibles pour atteindre la même source de nourriture, celle étant la plus courte sera, dans le même temps, parcourue par plus de fourmis que la longue piste ;
6. la piste courte sera donc de plus en plus renforcée, et donc de plus en plus attractive ;
7. la longue piste, elle, finira par disparaître, les phéromones étant volatiles ;
8. à terme, l'ensemble des fourmis a donc déterminé et « choisi » la piste la plus courte.



Les fourmis utilisent l'environnement comme support de communication : elles échangent indirectement de l'information en déposant des phéromones, le tout décrivant l'état de leur « travail ». L'information échangée a une portée *locale*, seule une fourmi située à l'endroit où les phéromones ont été déposées y a accès. Ce système porte le nom de « stigmergie », et se retrouve chez plusieurs animaux sociaux (il a notamment été étudié dans le cas de la construction de piliers dans les nids de termites).

#### **4-6) AVANTAGES ET INCONVENIENTS DES METHODES METAHEURISTIQUES**

##### ***c. Recuit simulé***

Le Recuit Simulé présente l'avantage d'offrir des solutions de bonne qualité, tout en restant simple à programmer et à paramétrer. Il offre autant de souplesse d'emploi que l'algorithme de recherche local classique : on peut inclure facilement des contraintes dans le corps du programme.

Aarts, Korst et Laarhoven (1997) ont par ailleurs démontré que, sous certaines conditions de décroissance de la température, l'algorithme du Recuit Simulé converge en probabilité vers un optimum global lorsque le nombre d'itérations tend vers l'infini.

L'un des inconvénients du Recuit Simulé est qu'une fois l'algorithme piégé à basse température dans un minimum local, il lui est impossible de s'en sortir tout seul.

##### ***d. Recherche tabou***

La méthode Tabou est une méthode de recherche locale, et la structure de son algorithme de base est finalement assez proche de celle du Recuit Simulé, donc on passera de l'un à l'autre facilement, avec l'avantage, par rapport au Recuit Simulé, d'avoir un paramétrage simplifié : dans un première temps, le paramétrage consistera d'abord à trouver une valeur indicative  $t$  d'itérations pendant lesquelles les mouvements sont interdits. Il faudra également décider d'une stratégie de mémorisation à long terme – sur la qualité des solutions par exemple.

En revanche, la méthode Tabou exige une gestion de la mémoire de plus en plus lourde à mesure que l'on voudra raffiner le procédé en mettant en place des stratégies de mémorisation complexe.

L'efficacité de la méthode Tabou fait qu'elle est largement employée dans les problèmes d'optimisation combinatoire : elle a été testée avec succès sur les grands problèmes classiques (voyageur de commerce, ordonnancement d'ateliers) et elle est fréquemment appliquée sur les problèmes de constitution de planning, de routage, d'exploration géologique, etc.

### *e. Les algorithmes génétiques*

D'abord, les algorithmes génétiques sont coûteux en temps de calcul, puisqu'ils manipulent plusieurs solutions simultanément. C'est le calcul de la fonction de performance qui est le plus pénalisant, et on optimise généralement l'algorithme de façon à éviter d'évaluer trop souvent cette fonction.

Ensuite, l'ajustement d'un Algorithme Génétique est délicat. L'un des problèmes les plus caractéristiques est celui de la *dérive génétique*, qui fait qu'un bon individu se met, en l'espace de quelques générations, à envahir toute la population. On parle dans ce cas de convergence prématurée, qui revient à lancer à une recherche locale autour d'un minimum... qui n'est pas forcément l'optimum attendu. Les méthodes de sélection proportionnelle peuvent en particulier favoriser ce genre de dérive. Un autre problème surgit lorsque les différents individus se mettent à avoir des performances similaires : les bons éléments ne sont alors plus sélectionnés, et l'algorithme ne progresse plus.

Le grand avantage des Algorithmes Génétiques est qu'ils parviennent à trouver de bonnes solutions sur des problèmes très complexes, et trop éloignés des problèmes combinatoires classiques pour qu'on puisse tirer profit de certaines propriétés connues. Ils doivent simplement déterminer entre deux solutions quelle est la meilleure, afin d'opérer leurs sélections. On les emploie dans les domaines où un grand nombre de paramètres entrent en jeu, et où l'on a besoin d'obtenir de bonnes solutions en quelques itérations seulement – dans les systèmes de régulation de transport en temps réel par exemple.

Par ailleurs, les Algorithmes Génétiques se prêtent bien, du fait de leur traitement simultané de solutions, à la recherche d'**optimum multiples** : en créant une fonction de coût partagée, dont la valeur dépend partiellement de la distance entre les individus, on voit se former graduellement des sous-populations d'individus, qui se stabilisent autour des différents pics de la fonction objectif.

### *f. Les algorithmes de fourmis*

L'algorithme de colonies de fourmi offre beaucoup de souplesse, et il a été possible de l'adapter à tous les grands problèmes d'optimisation combinatoires.

Par ailleurs, l'algorithme de colonies de fourmi se parallélise de façon très naturelle, en affectant par exemple un processus différent pour traiter la marche de chaque fourmi.

Les algorithmes de colonies de fourmi ont été appliqués avec succès sur les problèmes d'affectation quadratique, de routage et de planification, et ils ont été la source d'inspiration de nouvelles métaheuristiques.

## **4-7) LES ALGORITHMES HYBRIDES**

Le mode d'hybridation qui semble le plus fécond concerne la combinaison entre les méthodes de voisinage, l'approche d'évolution et d'autres méthodes. L'idée essentielle de

cette hybridation consiste à exploiter pleinement la puissance de recherche de méthodes de voisinage et de recombinaison des algorithmes évolutifs sur une population de solutions. Un tel algorithme utilise une ou plusieurs méthodes de voisinage sur les individus de la population pendant un certain nombre d'itérations ou jusqu'à la découverte d'un ensemble d'optima locaux et invoque ensuite un mécanisme de recombinaison pour créer de nouveaux individus.

Les algorithmes hybrides sont sans doute parmi les méthodes les plus puissantes. Malheureusement, les temps de calcul nécessaires peuvent devenir prohibitifs à cause du nombre d'individus manipulés dans la population.

#### **4-8) QUELLE METAHEURISTIQUE UTILISER ?**

Le premier problème pratique qui se pose à un utilisateur confronté à une application concrète est d'effectuer un choix parmi les différentes métaheuristiques disponibles. Ce choix est d'autant plus difficile qu'il n'existe pas de comparaison systématique et fiable des différentes métaheuristiques. Cependant, il est possible de caractériser les métaheuristiques selon quelques critères généraux, de manière à faciliter ce choix. Les critères de comparaisons retenus sont les suivants :

- facilité d'adaptation au problème
- possibilité d'intégrer des connaissances spécifiques au problème,
- qualité des meilleures solutions trouvées,
- rapidité, c'est-à-dire temps de calcul nécessaire pour trouver une telle solution

Le critère de qualité correspond à la meilleure qualité qu'il est possible d'obtenir par une exécution prolongée. Le critère de rapidité représente le temps de calcul typiquement nécessaire pour obtenir une telle solution.

Tout l'art du concepteur d'une heuristique résidera dans l'assemblage judicieux des différents principes que nous avons vus (intensification, diversification, hybridation...). La connaissance d'éléments de théorie connus en algorithmique, ou dans le domaine des méthodes dites exactes, permet d'améliorer les performances, selon les caractéristiques du problème à résoudre.

#### **4-9) CONCLUSION**

Nous avons présenté une synthèse de principales métaheuristiques : les méthodes de voisinage, les algorithmes évolutifs et les algorithmes hybrides. Nous avons proposé une analyse de ces métaheuristiques en dégagant les idées fondamentales et avançons des pistes pour guider le choix d'une métaheuristique en pratique.

***CHAPITRE V***  
***APPLICATION DES ALGORITHMES***  
***GENETIQUE DANS LA REPARTITION***  
***OPTIMALE DES PUISSANCES ACTIVES***

## APPLICATION DES ALGORITHMES GENETIQUES DANS LA REPARTITION OPTIMALE DES PUISSANCES ACTIVES

### 5-1) INTRODUCTION

Les Algorithmes Génétiques, à l'inverse des méthodes traditionnelles de recherche numériques de type gradient ne sont pas basés sur une approche analytique mais sur une approche heuristique. En cela, peu d'information est nécessaire pour l'utilisation de l'espace de recherche possible et un critère d'efficacité. Les spécialistes en recherche opérationnelle ont très vite remarqué le potentiel de l'utilisation des algorithmes génétiques comme outil de résolution numérique.

Dans ce chapitre, nous allons nous intéresser principalement à l'application des AG dans la répartition optimale des puissances actives. Nous allons appliquer et comparer deux types d'AG simple, l'un avec un codage binaire, l'autre avec un codage réel.

### 5-2) PRESENTATION DU PROBLEME

Soit la fonction objective à minimiser :

$$\text{Minimiser } \sum_{i=1}^{NG} C_i(P_{Gi}) \quad (5-1)$$

Sous les contraintes :

$$\sum_{i=1}^{NG} P_{Gi} - \sum_{j=1}^{NC} P_{Chj} - P_L = 0 \quad (5-2)$$

$$P_{Gi}^{\min} \leq P_{Gi} \leq P_{Gi}^{\max} \quad (5-3)$$

Les contraintes inégalités seront prises en considération dans l'algorithme génétique par contre la contrainte égalité n'est pas prise en considération, pour cette raison qu'on va utiliser une méthode basée sur la transformation du problème original avec contraintes en un problème auxiliaire sans contraintes. Nous appliquons la pénalisation on aura une fonction sans contraintes à minimiser ( $F_m$ ), cette dernière tient compte de la fonction coût  $F$  et la contrainte égalité [NAA 05 a].

$$F_m(P_G, r_k) = \sum_{i=1}^{NG} C_i(P_{Gi}) + \frac{1}{r_k} \cdot B \cdot h^2(P_G) \quad (5-4)$$

Avec  $r_k$  : coefficient de pénalité

$h(P_G)$  est la contrainte égalité

$B$  : une constante, tel que :

$$\begin{cases} B > 0 & \text{si } h(P_G) \neq 0 \\ B = 0 & \text{si } h(P_G) = 0 \end{cases} \quad (5-5)$$

$$h(P_G) = \sum_{i=1}^{NG} P_{Gi} - P_{Ch} - P_L = 0$$

$P_{Ch}$  : puissance active totale consommée

$P_L$  : perte totale active dans le réseau.

### 5-3) CODAGE BINAIRE

La première étape lors de l'application d'un AG est le choix du codage des paramètres sous forme de chaînes maniabiles. La représentation choisie est primordiale car l'efficacité d'un AG dépend principalement de la manière dont elle capte les régularités de l'espace de recherche. Nous avons utilisé un AG avec une représentation binaire de l'individu, rappelons que la transformation la plus simple d'une chaîne binaire "a" en un nombre entier "x" s'opère par la règle suivante :

$$x = \sum_{i=1}^l a_i 2^{l-i} \quad (5-6)$$

Où :  $l$  : nombre de bits

$a_i$  : une chaîne binaire

$x$  : la valeur réelle.

Nous codons les variables  $P_{Gi}$  par les chaînes binaires de 12 bits :

La chaîne 000000000000 code  $P_{Gi} = P_{Gi}^{\min}$

La chaîne 111111111111 code  $P_{Gi} = P_{Gi}^{\max}$

Par exemple le chromosome A = {1011} vaut  $1*2^3 + 0*2^2 + 1*2^1 + 1*2^0 = 11$ .

La transformation d'une chaîne binaire A = { $a_1, \dots, a_s$ } en un nombre réel x peut alors s'exécuter en deux étapes :

- Conversion (base 2 en base 10) :  $P_G' = \sum_{i=1}^l a_i 2^{l-i}$
- Recherche du nombre réel correspondant :

$$P_G = P_G^{\min} + P_G' \frac{P_G^{\max} - P_G^{\min}}{2^{l-1}} \quad (5-7)$$

$$\text{Où : } P_G^{\min} \leq P_G \leq P_G^{\max}$$

### **5-3-a / création de la population initiale**

La population initiale  $P(0)$  est composée de  $N_{pop} = 30$  chaînes générées aléatoirement. Pour ce qui est de la phase d'initialisation, la procédure est assez simple. Elle consiste en un tirage aléatoire de  $N_{pop}$  individus dans l'espace des individus permis, respectant la contrainte inégalité (5-3).

### **5-3-b/ fonction d'évaluation**

L'évaluation est la détermination de l'aptitude de l'individu, on calculant la valeur de sa fonction objective, après que chaque chaîne ait été décodée en nombre réel. Dans notre cas, l'évaluation consiste à calculer la valeur de la fonction  $F_m$  par l'équation (5-4) :

### **5-3-c/ opérateur de sélection**

La sélection permet d'identifier statistiquement les meilleurs individus d'une population et d'éliminer les mauvais. Sur la base du résultat de l'évaluation, l'algorithme détermine la probabilité de sélection de chaque individu dans la nouvelle population. L'individu le moins performant (qui donne la valeur max de  $F_m$ ) va être éliminé, par contre l'individu le plus performant (qui donne la valeur min de  $F_m$ ) apparaît de nouveau.

### **5-3-d/ opérateur de croisement**

Après avoir sélectionné aléatoirement deux parents, nous choisissons, là aussi aléatoirement un point de coupe. On génère ainsi deux nouveaux individus en inversant les gènes des deux parents, à partir du point de coupe.

### **5-3-e/ opérateur de mutation**

Cette fonction consiste à modifier aléatoirement un bit de l'individu, c'est-à-dire chaque bit de l'ensemble des individus de la population à une probabilité  $P_m$  de subir une mutation à chaque génération.

### Organigramme de la méthode (AG –codage binaire)

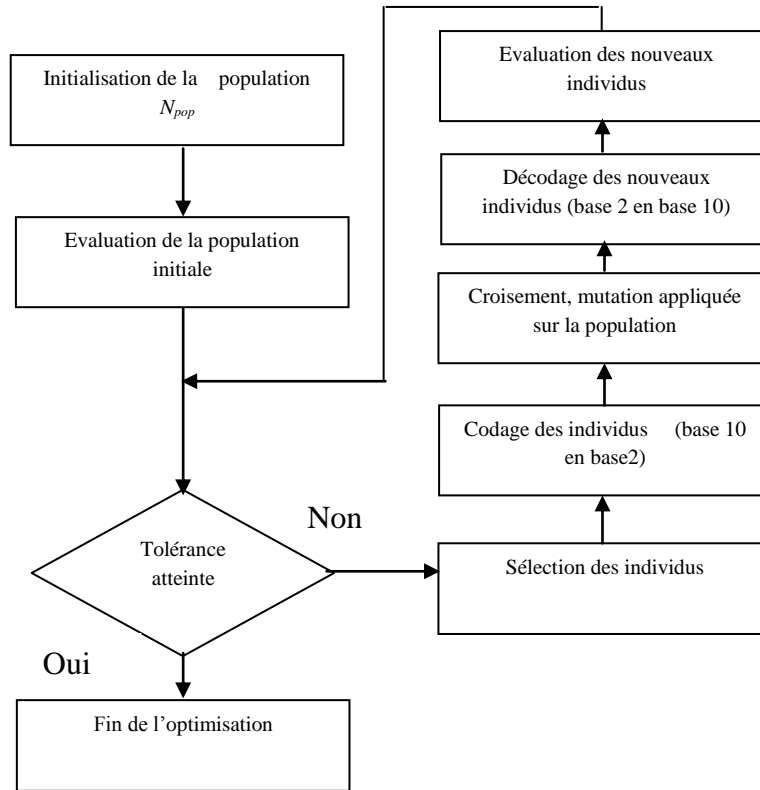


Figure 5.1. Organigramme de l'Algorithme Génétique / (Codage binaire)

#### 5-4) CODAGE REEL

L'AG que nous avons utilisé par la suite est représenté par des paramètres manipulant des valeurs réelles. Une des améliorations moyennes consiste alors à se servir des nombres réels directement. Les résultats donnés par **Michel Wicz** [SPA 99] montrent que la représentation binaire aboutit souvent à une faible précision et qu'en règle générale, le gain en temps de calcul est positif. La conclusion à laquelle il arrive est naturellement qu'une représentation plus naturelle du problème offre des solutions plus efficaces. En utilisant le codage réel, l'individu n'est alors plus qu'un chiffre à valeur réelle dans l'espace des valeurs permises. Nous avons utilisé les mêmes procédures précédentes d'initialisation, évaluation et la sélection. L'évaluation s'effectue sans passer par le décodage.



#### 5-4-a/opérateur de croisement

Il existe plusieurs types de croisement, nous avons utilisé le croisement arithmétique décrit au chapitre (4). Ce type de croisement effectue une simple combinaison linéaire entre deux parents.

#### 5-4-b/opérateur de mutation

Nous avons appliqué une mutation uniforme identique à celle du codage binaire. Ainsi, chaque variable  $P_{Gi} \in \mathfrak{R}$  est changée selon une certaine probabilité en un nombre aléatoire tiré dans une distribution uniforme sur l'intervalle  $[P_{Gi}^{\min}, P_{Gi}^{\max}]$ .

#### Organigramme de la méthode (AG –codage réel)

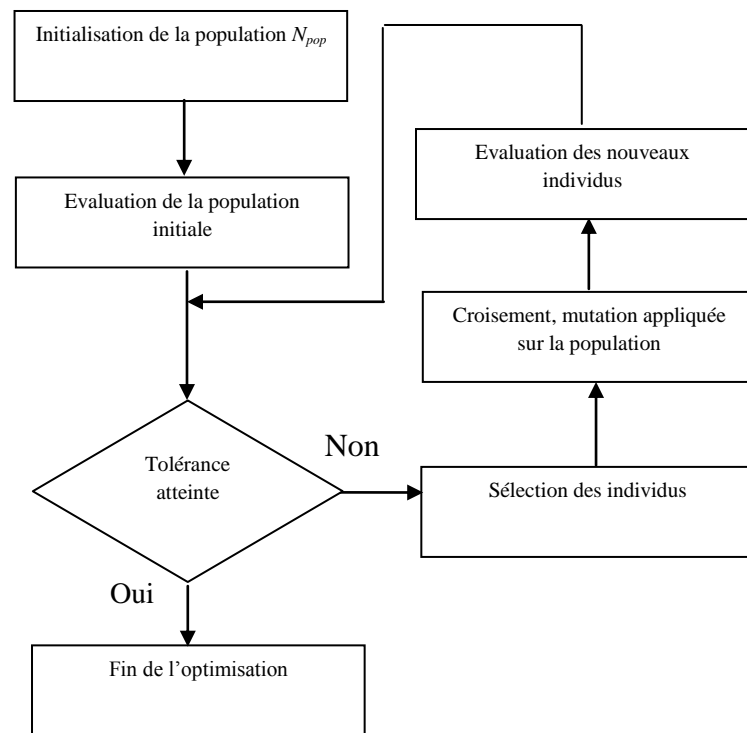


Figure 5.2. Organigramme de l'Algorithme Génétique / (Codage réel)

### 5-5) ILLUSTRATION

Nous avons appliqué un AG simple sur trois réseaux différents, un réseau à 5 nœuds (IEEE-5), un réseau à 30 nœuds (IEEE-30) et un réseau à 57 nœuds (IEEE-57).

Les données de chaque réseau, les tensions planifiées ainsi que les puissances générées et consommées sont données dans l'annexe (A).

#### a / réseau à 5 nœuds :

Le réseau IEEE-5 comportant 3 nœuds générateurs, dont les fonctions coûts exprimées en Dollars par heures sont données par les expressions suivantes [TSA 71]:

$$F_1 (P_{G1}) = 0.0060 P_{G1}^2 + 2.0 P_{G1} + 140$$

$$F_2 (P_{G2}) = 0.0075 P_{G2}^2 + 1.8 P_{G2} + 120$$

$$F_3 (P_{G3}) = 0.0070 P_{G3}^2 + 1.5 P_{G3} + 80$$

Sous les contraintes :

$$30 \leq P_{G1} \leq 120 ; 20 \leq P_{G2} \leq 80 ; 10 \leq P_{G3} \leq 60$$

Où les puissances générées sont exprimées en MW, la puissance apparente de base est de 100MVA.

Le choix des paramètres de contrôle de l'algorithme génétique pour un codage binaire est le suivant :

- Taille de la population :  $N_{pop} = 30$
- Probabilité de croisement :  $pc = 0.75$
- Croisement en double
- Probabilité de mutation :  $pm = 0.06$
- Mutation simple
- La longueur du chromosome :  $l = 12$  bits
- Nombre de génération :  $maxgen = 300$
- Coefficient de pénalité :  $rk = 0.052$

Le choix des paramètres de contrôle de l'algorithme génétique pour un codage réel est le suivant :

- Taille de la population :  $N_{pop} = 30$
- Probabilité de croisement :  $pc = 0.8$
- Probabilité de mutation :  $pm = 0.058$
- Nombre de génération :  $maxgen = 300$
- Coefficient de pénalité :  $rk = 0.02$

Nous présentons dans le tableau suivant les résultats de l'Algorithme Génétique en codage binaire et réel. Nous avons effectué dix essais avec des conditions initiales différentes [NAA 04].

N° d'essais	$P_{G1}^{opt}$ (MW)	$P_{G2}^{opt}$ (MW)	$P_{G3}^{opt}$ (MW)	F (\$/h)
1	50.6813	61.4945	55.7998	699.6164
2	44.5055	74.4029	49.0598	699.2006
3	41.6264	68.8205	57.5092	699.1800
4	47.9121	73.1722	46.9109	699.4000
5	47.4725	67.6044	52.8938	698.9608
6	48.0000	68.8059	51.1600	698.9876
7	40.4835	70.9011	56.5934	699.1430
8	45.3407	68.7619	53.8462	698.9118
9	45.5165	68.1758	54.2979	698.9810
10	41.8462	71.6630	54.4689	699.0230

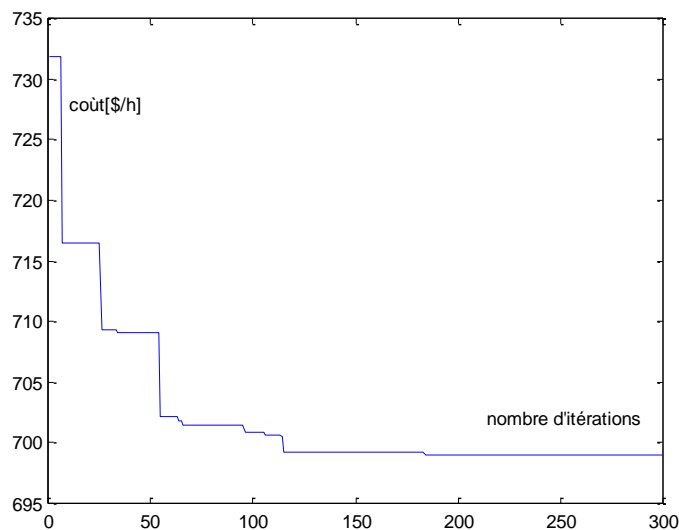
**Tableau 5.1.** Résultats d'un AG/ Réseau IEEE-5/ (codage binaire)

N° d'essais	$P_{G1}^{opt}$ (MW)	$P_{G2}^{opt}$ (MW)	$P_{G3}^{opt}$ (MW)	F (\$/h)
1	49.1863	67.4978	51.2944	699.0535
2	53.6145	60.8601	53.5006	699.8947
3	47.4716	70.8435	49.6569	699.0461
4	53.7975	68.9315	45.2340	699.8840
5	45.1926	69.6710	53.1172	698.9127
6	51.7749	67.0453	49.1480	699.3588
7	45.7845	69.0550	53.1418	698.9178
8	41.9756	68.1355	57.8608	699.1621
9	46.9026	70.1532	50.9255	698.9659
10	53.2489	67.3667	47.3706	699.5919

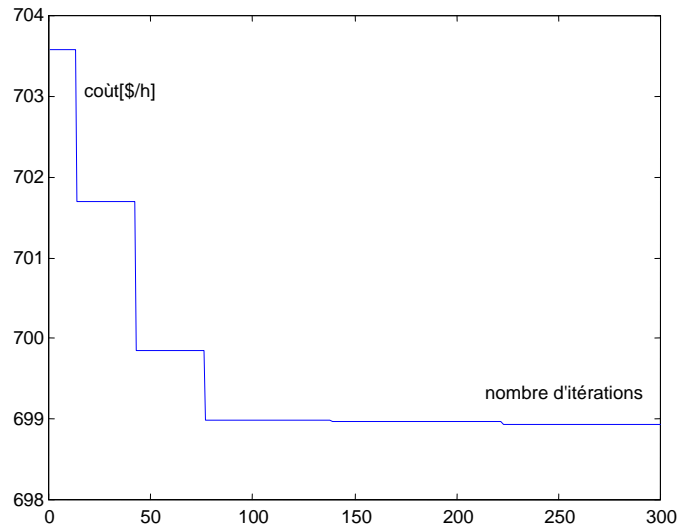
**Tableau 5.2.** Résultats d'un AG/Réseau IEEE-5/ (codage réel)

Après dix essais on remarque que, les résultats s'avancent au voisinage d'une même valeur optimale du coût.

La courbe donnant le coût total en fonction du nombre d'itération est donnée par la figure suivante :



**Figure 5.3.** Coût en fonction du nombre d'itérations/ Réseau à IEEE-5/ (codage binaire)



**Figure 5.4.** Coût en fonction du nombre d'itérations/ Réseau IEEE-5/ (codage réel)

#### **b / réseau à 30 nœuds :**

Considérons le réseau IEEE-30, comportant 6 nœuds générateurs, dont les fonctions coûts exprimées en Dollars par heures sont données par les expressions suivantes [BOU 04] :

$$F_1 (P_{G1}) = 0.00375 P_{G1}^2 + 2.00 P_{G1} + 0.0$$

$$F_2 (P_{G2}) = 0.01750 P_{G2}^2 + 1.75 P_{G2} + 0.0$$

$$F_5 (P_{G5}) = 0.06250 P_{G5}^2 + 1.00 P_{G5} + 0.0$$

$$F_8 (P_{G8}) = 0.00834 P_{G8}^2 + 3.25 P_{G8} + 0.0$$

$$F_{11} (P_{G11}) = 0.02500 P_{G11}^2 + 3.00 P_{G11} + 0.0$$

$$F_{13} (P_{G13}) = 0.02500 P_{G13}^2 + 3.00 P_{G13} + 0.0$$

Sous les contraintes :

$$50 \leq P_{G1} \leq 200 ; 20 \leq P_{G2} \leq 80 ; 15 \leq P_{G5} \leq 50 ;$$

$$10 \leq P_{G8} \leq 35 ; 10 \leq P_{G11} \leq 30 ; 12 \leq P_{G13} \leq 40$$

Où les puissances générées sont exprimées en MW, la puissance apparente de base est de 100MVA.

Le choix des paramètres de contrôle de l'algorithme génétique pour un codage binaire est le suivant :

- Taille de la population :  $N_{pop} = 30$
- Probabilité de croisement :  $pc = 0.75$
- Croisement double
- Probabilité de mutation :  $pm = 0.01$
- Mutation simple
- La longueur du chromosome :  $l = 12$  bits
- Nombre de génération :  $maxgen = 300$
- coefficient de pénalité  $rk=0.062$

Le choix des paramètres de contrôle de l'algorithme génétique pour un codage réel est le suivant :

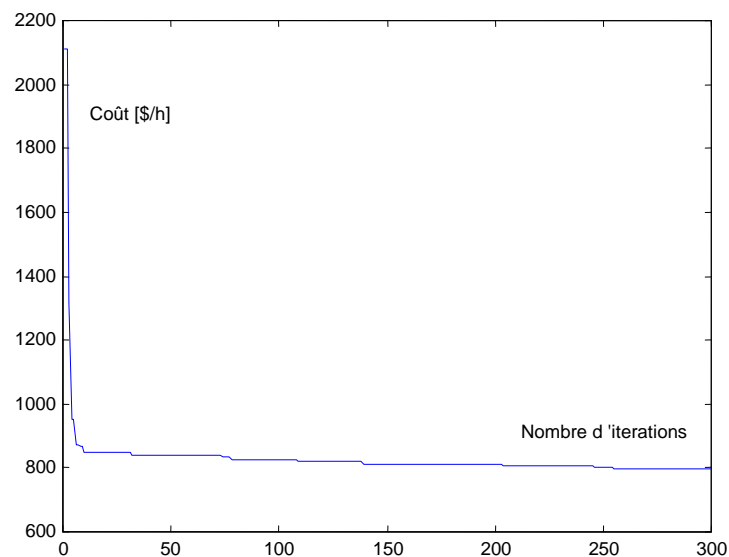
- Taille de la population :  $N_{pop} = 30$
- Probabilité de croisement :  $pc = 0.8$
- Probabilité de mutation :  $pm = 0.06$
- Nombre de génération :  $maxgen = 300$
- coefficient de pénalité  $rk=0.052$

N° d'essais	$P_{G1}^{opt}$ (MW)	$P_{G2}^{opt}$ (MW)	$P_{G5}^{opt}$ (MW)	$P_{G8}^{opt}$ (MW)	$P_{G11}^{opt}$ (MW)	$P_{G13}^{opt}$ (MW)	F (\$/h)
1	189.3407	43.7949	17.5470	15.9341	12.7448	12.9368	799.3739
2	185.3997	49.0839	23.1075	10.0000	12.0195	12.7264	799.5154
3	184.4285	49.3581	18.8727	11.9275	14.8806	12.8438	799.3578
4	186.4538	47.2114	21.6735	13.9565	11.0111	12.0240	798.6837
5	186.5848	48.7220	21.6415	11.9507	10.3548	13.0808	798.7700
6	186.9172	43.2949	18.9428	17.2530	13.3535	12.5356	799.5142
7	190.1316	50.3014	17.0608	12.2758	10.1042	12.4267	798.6289
8	186.4213	48.8085	21.3919	13.4174	10.0575	12.2297	798.5051
9	186.3300	48.4142	18.0508	15.0317	12.5051	12.0000	798.7975
10	184.6763	51.6669	19.2512	13.4357	11.0370	12.2457	798.6231

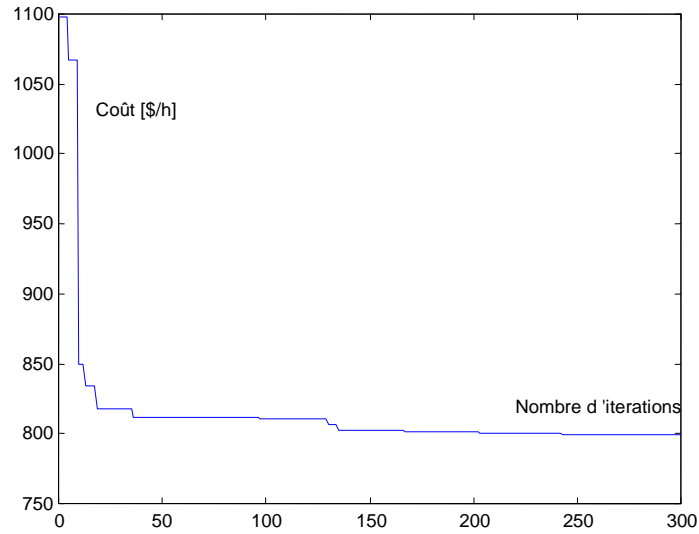
**Tableau 5.3.** Résultats d'un AG / Réseau IEEE-30 / (codage binaire)

N° d'essais	$P_{G1}^{opt}$ (MW)	$P_{G2}^{opt}$ (MW)	$P_{G5}^{opt}$ (MW)	$P_{G8}^{opt}$ (MW)	$P_{G11}^{opt}$ (MW)	$P_{G13}^{opt}$ (MW)	F (\$/h)
1	184.0723	48.7749	20.8817	15.3744	11.2184	12.0000	798.6668
2	186.6276	49.5368	20.9489	11.5572	11.5932	12.0000	798.9341
3	190.6294	49.2130	18.4158	11.6343	10.2268	12.2121	798.2691
4	183.7953	49.8256	17.9917	16.2606	11.0904	13.3249	799.1517
5	182.1444	47.1955	20.2732	17.8545	11.2209	13.6293	799.2644
6	185.6805	45.5362	19.2828	18.3384	11.4535	12.0000	798.9662
7	181.4069	49.6531	18.1893	18.9142	11.4412	12.6800	799.4959
8	179.2506	48.2774	19.2882	23.2781	10.0719	12.1474	799.8729
9	183.7130	48.0867	20.8475	17.6938	10.0000	12.0000	798.9077
10	186.6289	49.1543	19.2867	14.3670	10.8810	12.0000	798.3289

**Tableau 5.4.** Résultats d'un AG / Réseau IEEE-30 / (codage réel)



**Figure 5.5.** Coût en fonction du nombre d'itérations/réseau IEEE-30/ (codage binaire)



**Figure 5.6.** Coût en fonction du nombre d'itérations /Réseau IEEE-30/ (codage réel)

### **c / réseau à 57 nœuds :**

Considérons le réseau IEEE-57 [ZIM 97], comportant 7 nœuds générateurs, dont les fonctions coût et les contraintes sont :

$$F_1 (P_{G1}) = 0.0775795 P_{G1}^2 + 20 P_{G1} + 0.0$$

$$F_2 (P_{G2}) = 0.01 P_{G2}^2 + 40 P_{G2} + 0.0$$

$$F_3 (P_{G3}) = 0.25 P_{G3}^2 + 20 P_{G3} + 0.0$$

$$F_6 (P_{G6}) = 0.01 P_{G6}^2 + 40 P_{G6} + 0.0$$

$$F_8 (P_{G8}) = 0.022222 P_{G8}^2 + 20 P_{G8} + 0.0$$

$$F_9 (P_{G9}) = 0.01 P_{G9}^2 + 40 P_{G9} + 0.0$$

$$F_{12} (P_{G12}) = 0.03225 P_{G12}^2 + 20 P_{G12} + 0.0$$

Sous les contraintes :

$$00 \leq P_{G1} \leq 575.88 ; 00 \leq P_{G2} \leq 100 ; 00 \leq P_{G3} \leq 140 ;$$

$$00 \leq P_{G6} \leq 100 ; 00 \leq P_{G8} \leq 550 ; 00 \leq P_{G9} \leq 100 ; 00 \leq P_{G12} \leq 410$$

Où les puissances générées sont exprimées en MW, la puissance apparente de base est de 100MVA.



Le choix des paramètres de contrôle de l'algorithme génétique pour un codage binaire est le suivant :

- Taille de la population :  $N_{pop} = 30$
- Probabilité de croisement :  $pc = 0.75$
- Croisement double
- Probabilité de mutation :  $pm = 0.006$
- Mutation simple
- La longueur du chromosome :  $l = 12$  bits
- Nombre de génération :  $maxgen = 300$
- coefficient de pénalité  $rk=0.021$

Le choix des paramètres de contrôle de l'algorithme génétique pour un codage réel est le suivant :

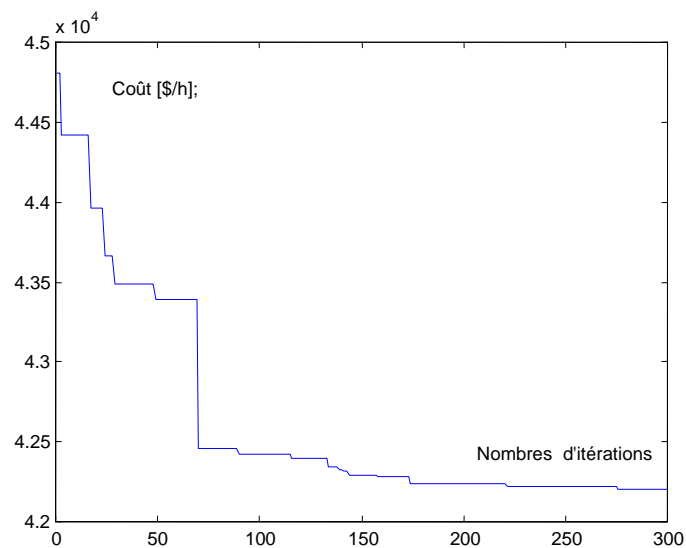
- Taille de la population :  $N_{pop} = 30$
- Probabilité de croisement :  $pc = 0.8$
- Probabilité de mutation :  $pm = 0.06$
- Nombre de génération :  $maxgen = 300$
- coefficient de pénalité  $rk=0.02$

N° d'essais	$P_{G1}^{opt}$ (MW)	$P_{G2}^{opt}$ (MW)	$P_{G3}^{opt}$ (MW)	$P_{G6}^{opt}$ (MW)	$P_{G8}^{opt}$ (MW)	$P_{G9}^{opt}$ (MW)	$P_{G12}^{opt}$ (MW)	F (\$/h)
1	150.8724	99.3558	41.5390	58.5062	508.2641	76.7431	343.4302	42201
2	123.7492	100.0000	42.5717	79.9277	500.7638	82.2581	349.3376	42202
3	129.3564	80.8672	39.9242	100.0000	501.8550	74.5974	352.1661	42202
4	137.3584	72.5982	45.4724	79.5469	491.6909	87.2824	364.7849	42202
5	154.0044	67.6816	39.2318	96.0795	488.3288	81.8469	351.4369	42202
6	148.0344	88.6822	38.4853	92.7300	509.6622	63.1278	338.0750	42206
7	147.8791	78.9803	37.4986	90.7987	513.0241	63.9562	346.5537	42205
8	146.1640	58.3586	48.6160	100.0000	514.4569	72.9971	338.0635	42205
9	147.3341	93.3161	45.0470	66.8531	517.8520	67.5603	340.7799	42205
10	138.7134	99.9144	39.2119	100.0000	513.4709	62.4044	324.9279	42201

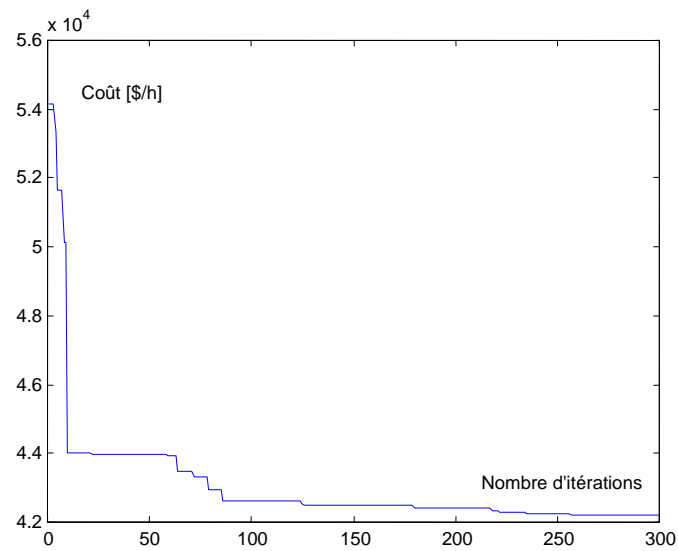
**Tableau 5.5.** Résultats d'un AG / Réseau IEEE-57 / (codage binaire)

N° d'essais	$P_{G1}^{opt}$ (MW)	$P_{G2}^{opt}$ (MW)	$P_{G3}^{opt}$ (MW)	$P_{G6}^{opt}$ (MW)	$P_{G8}^{opt}$ (MW)	$P_{G9}^{opt}$ (MW)	$P_{G12}^{opt}$ (MW)	F (\$/h)
1	143.4971	88.0681	42.9087	90.1611	523.0149	56.2870	334.7155	42205
2	146.2465	74.3569	50.7484	75.0733	508.5059	75.3609	348.3446	42201
3	153.3561	91.2861	43.3192	74.8330	515.7542	63.5133	336.6148	42206
4	148.4455	67.1039	44.5189	67.2251	519.3938	90.5703	341.3649	42203
5	144.6749	89.3485	39.5946	64.0400	480.2198	100.0000	360.8382	42204
6	144.9055	94.1925	47.7452	49.7160	506.4844	88.5840	347.1230	42204
7	147.8722	89.9957	36.8012	99.9962	504.3230	56.6821	343.0062	42202
8	152.2993	66.8059	48.7061	70.0455	506.9163	88.6207	345.2897	42204
9	137.3781	89.7463	39.7459	61.7831	507.9423	82.4414	359.6605	42205
10	157.7194	65.3578	46.7609	89.8808	495.9181	81.6604	341.3659	42202

**Tableau 5.6.** Résultats d'un AG / Réseau IEEE-57 / (codage réel)



**Figure 5.7.** Coût en fonction du nombre d'itérations/ Réseau IEEE-57/ (codage binaire)



**Figure 5.8.** Coût en fonction du nombre d'itérations/Réseau IEEE-57/ (codage réel)

Nous présentons dans les tableaux ci-dessous une comparaison des résultats obtenus par l'AG et la méthode de Quasi-Newton (QN) [BOU 06, NAA 04].

	$P_{G1}^{opt}$ (MW)	$P_{G2}^{pt}$ (MW)	$P_{G3}^{opt}$ (MW)	F (\$/h)
AG codage binaire	45.3407	68.7619	53.8462	698.9118
AG codage réel	45.1926	69.6710	53.1172	698.9127
QN	70.6216	58.5665	38.7881	705.0929

**Tableau 5.7.** Comparaison entre l'AG et QN / Réseau IEEE-5/

	$P_{G1}^{opt}$ (MW)	$P_{G2}^{opt}$ (MW)	$P_{G5}^{opt}$ (MW)	$P_{G8}^{opt}$ (MW)	$P_{G11}^{opt}$ (MW)	$P_{G13}^{opt}$ (MW)	F (\$/h)
AG codage binaire	186.4213	48.8085	21.3919	13.4174	10.0575	12.2297	798.5051
AG codage réel	190.6294	49.2130	18.4158	11.6343	10.2268	12.2121	798.2691
QN	138.3960	56.2886	25.2812	28.3377	21.2439	24.5741	830.4255

**Tableau 5.8.** Comparaison entre l'AG et QN / Réseau IEEE-30/

	$P_{G1}^{opt}$ (MW)	$P_{G2}^{opt}$ (MW)	$P_{G3}^{opt}$ (MW)	$P_{G6}^{opt}$ (MW)	$P_{G8}^{opt}$ (MW)	$P_{G9}^{opt}$ (MW)	$P_{G12}^{opt}$ (MW)	F (\$/h)
AG codage binaire	138.7134	99.9144	39.2119	100.0000	513.4709	62.4044	324.9279	42201
AG codage réel	146.2465	74.3569	50.7484	75.0733	508.5059	75.3609	348.3446	42201
QN	328.1308	91.9337	110.0241	91.9337	317.4165	91.9337	247.3093	46934

**Tableau 5.9.** Comparaison entre l'AG et QN / Réseau IEEE-57/

## 5-6) Conclusion

Selon les résultats obtenus lors des trois applications, nous avons remarqué que les valeurs optimales du coût sont obtenues par les Algorithmes Génétiques.

La méthode de QN dans les applications converge plus rapidement, mais donne un coût nettement supérieur par rapport à l'Algorithme Génétique.

Les Algorithmes Génétiques peuvent constituer une alternative intéressante lorsque les méthodes d'optimisation traditionnelles (méthodes analytiques) ne parviennent pas à fournir efficacement des résultats fiables.

***CHAPITRE VI***  
***APPLICATION DE LA RECHERCHE***  
***TABOU DANS***  
***LA REPARTITION OPTIMALE DES***  
***PUISSANCES ACTIVES***

## APPLICATION DE LA RECHERCHE TABOU DANS LA REPARTITION OPTIMALE DES PUISSANCES ACTIVES

### 6-1) INTRODUCTION

Un problème d'optimisation combinatoire est un problème qui peut s'exprimer par une fonction (dite coût) avec ou sans contraintes, à minimiser ou à maximiser sur un ensemble de définition fini ou dénombrable. C'est le cas de nombreux problèmes, dans des domaines d'applications très variés, qu'ils soient scientifiques ou techniques. La résolution satisfaisante d'un problème d'optimisation difficile, qui comporte un grand nombre de solution sous optimales, justifie souvent le recours à une « métaheuristique » puissante. Ces méthodes d'optimisation permettent de rechercher les optimums globaux d'un problème d'optimisation, en évitant le piègeage dans l'un de ses minimums locaux.

L'optimisation est un ensemble de techniques permettant de trouver les valeurs des variables qui rendent optimale une fonction de réponse, appelée aussi « fonction objectif ». Sur le plan mathématique, cela correspond à la recherche des extrémums de fonctions à plusieurs variables.

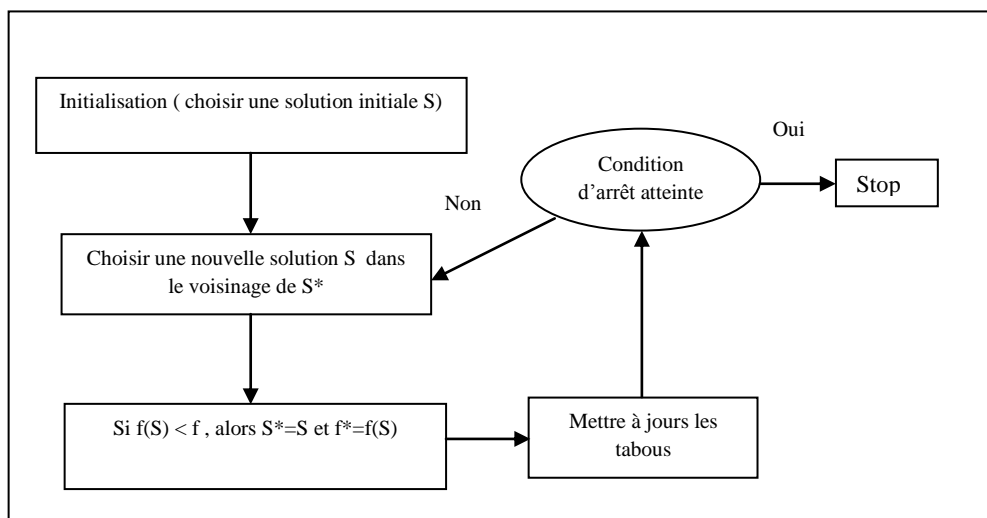
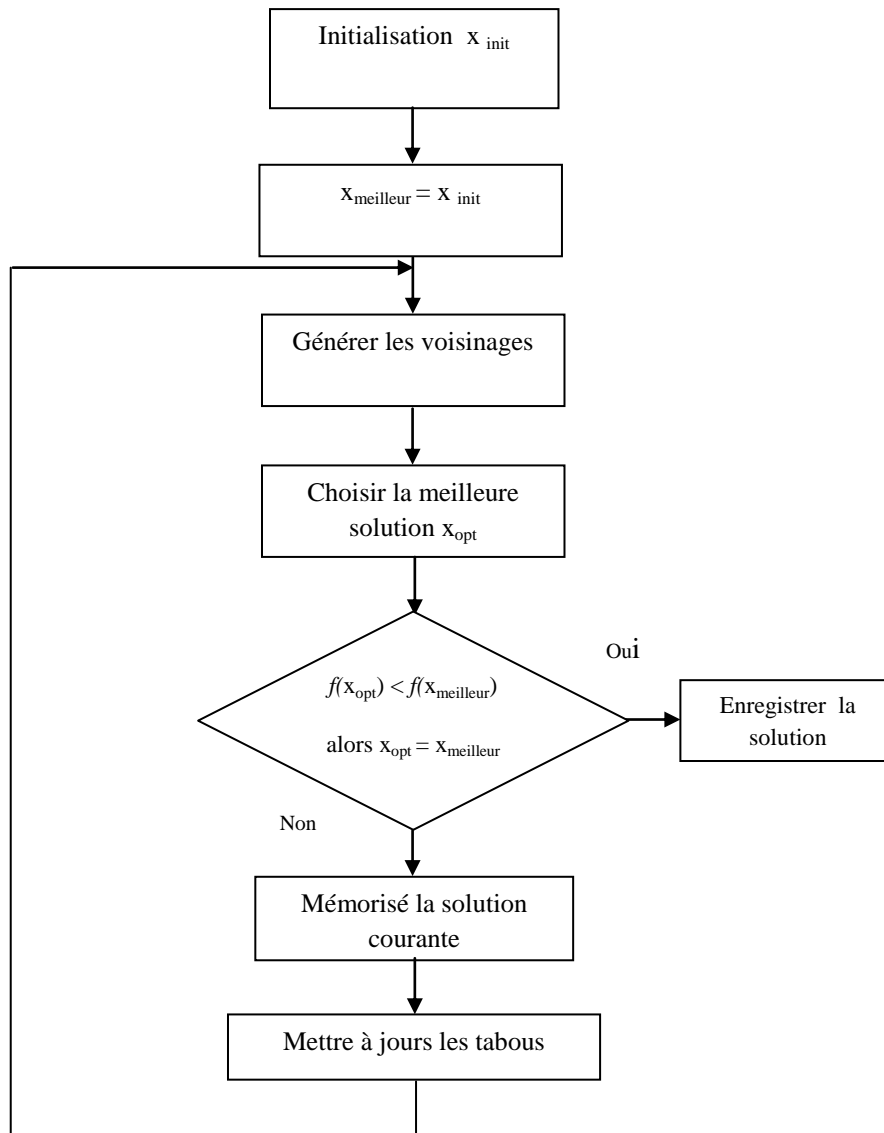


Figure 6.1. Schéma tabou

Dans ce chapitre nous nous intéressons à la méthode de la recherche tabou appliquées à l'optimisation des puissances actives dans les réseaux électriques IEEE-5, IEEE-30 et IEEE-57.

**6-2) ORGANIGRAMME DE LA RECHERCHE TABOU****Figure 6.2.** Organigramme de la recherche tabou

### 6-3) ILLUSTRATION

Nous avons appliqué la Recherche Tabou sur des réseaux différents, IEEE-5, IEEE-30 et IEEE-57.

Les paramètres de contrôle de l'algorithme de Recherche Tabou pour un réseau IEEE-5 sont :

- Nombre d'itération :  $i_{max}= 200$
- Longueur de la liste taboue :  $L= 5$
- Nombre de points de recherche autour du voisinage:  $M= 12$
- Nombre de diversifications:  $idiver = 1$
- Facteur de pénalité  $rk=1$

Les paramètres de contrôle de l'algorithme de Recherche Tabou pour un réseau IEEE-30 sont :

- Nombre d'itération :  $i_{max}= 200$
- Longueur de la liste taboue :  $L= 10$
- Nombre de points de recherche autour du voisinage:  $M= 20$
- Nombre de diversifications:  $idiver = 1$
- Facteur de pénalité  $rk=0.1$

Les paramètres de contrôle de l'algorithme de Recherche Tabou pour un réseau IEEE-57 sont :

- Nombre d'itération :  $i_{max}= 200$
- Longueur de la liste taboue :  $L=10$
- Nombre de points de recherche autour du voisinage:  $M= 45$
- Nombre de diversifications:  $idiver = 1$
- Facteur de pénalité  $rk=0.1$

Dans le but de montrer l'efficacité de l'algorithme de Recherche Tabou nous avons effectué plusieurs essais, les résultats sont présentés dans les tableaux (6.1), (6.2), (6.3)



N° d'essais	$P_{G1}^{opt}$ (MW)	$P_{G2}^{opt}$ (MW)	$P_{G3}^{opt}$ (MW)	F (\$/h)
1	51.5061	76.1626	40.3850	700.9029
2	51.5061	76.1626	40.3850	700.9029
3	51.5061	76.1626	40.3850	700.9029
4	51.5061	76.1626	40.3850	700.9029
5	51.5061	76.1626	40.3850	700.9029

**Tableau 6.1.** Résultats de RT /Réseau IEEE-5/

N° d'essais	$P_{G1}^{opt}$ (MW)	$P_{G2}^{opt}$ (MW)	$P_{G3}^{opt}$ (MW)	$P_{G4}^{opt}$ (MW)	$P_{G5}^{opt}$ (MW)	$P_{G6}^{opt}$ (MW)	F (\$/h)
1	191.9536	40.8547	17.8429	13.2328	12.5646	15.8470	801.0630
2	191.9536	40.8547	17.8429	13.2328	12.5646	15.8470	801.0630
3	191.9536	40.8547	17.8429	13.2328	12.5646	15.8470	801.0630
4	191.9536	40.8547	17.8429	13.2328	12.5646	15.8470	801.0630
5	191.9536	40.8547	17.8429	13.2328	12.5646	15.8470	801.0630

**Tableau 6.2.** Résultats de RT /Réseau IEEE-30/

N° d'essais	$P_{G1}^{opt}$ (MW)	$P_{G2}^{opt}$ (MW)	$P_{G3}^{opt}$ (MW)	$P_{G6}^{opt}$ (MW)	$P_{G8}^{opt}$ (MW)	$P_{G9}^{opt}$ (MW)	$P_{G12}^{opt}$ (MW)	F (\$/h)
1	253.3952	97.4596	94.1011	40.5436	528.1956	94.2480	170.6614	44760
2	253.3952	97.4596	94.1011	40.5436	528.1956	94.2480	170.6614	44760
3	253.3952	97.4596	94.1011	40.5436	528.1956	94.2480	170.6614	44760
4	253.3952	97.4596	94.1011	40.5436	528.1956	94.2480	170.6614	44760
5	253.3952	97.4596	94.1011	40.5436	528.1956	94.2480	170.6614	44760

**Tableau 6.3.** Résultats de RT /Réseau IEEE-57/

L'évolution de la fonction coût en fonction du nombre d'itérations est représentée par les figures (6-3), (6-4) et (6-5) pour les réseaux IEEE-5, IEEE-30 et IEEE-57 respectivement

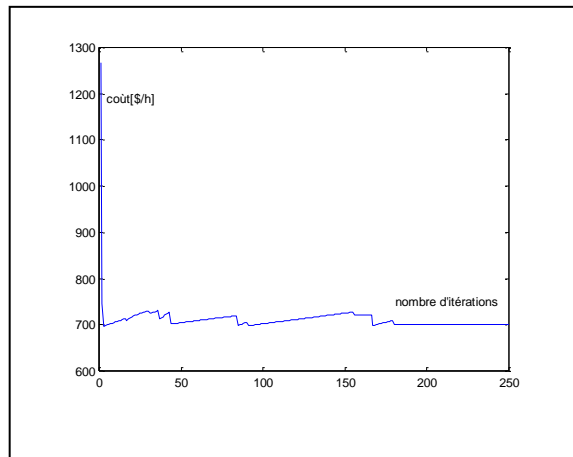


Figure 6.3. Courbe du coût total / Réseau IEEE-5

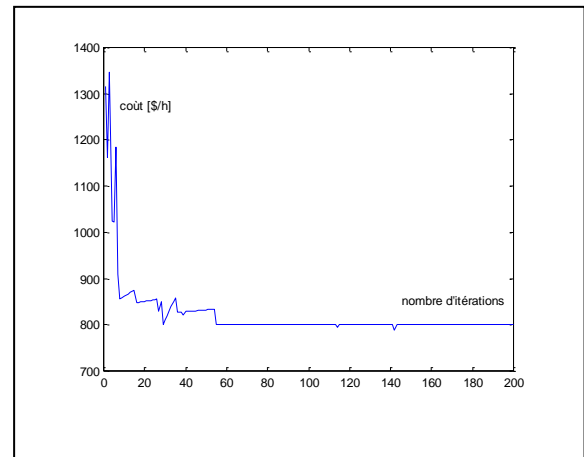


Figure 6.4. Courbe du coût total / Réseau IEEE-30

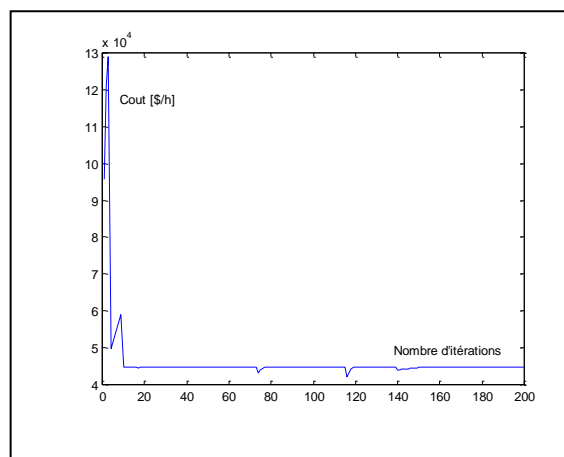


Figure 6.5. Courbe du coût total / Réseau IEEE-57

### Discussion des résultats :

Les résultats trouvés par la méthode de la Recherche Tabou confirment bien la validité et l'efficacité de cette méthode quelque soit le changement des conditions initiales dans les différents réseaux. Ces résultats ne s'éloignent pas en comparant aux résultats trouvés par les algorithmes génétiques.

Pour démontrer la puissance de la méthode Recherche Tabou, nous avons comparé cette dernière avec les résultats de l'AG et la méthode de Quasi-Newton (QN) [NAA 04, BOU 05 a, BOU 06].

	$P_{G1}^{opt}$ (MW)	$P_{G2}^{pt}$ (MW)	$P_{G3}^{opt}$ (MW)	F (\$/h)	T(s)
RT	51.5061	76.1626	40.3850	700.9029	1.10
QN	70.6216	58.5665	38.7881	705.0929	0.55
AG	41.8462	71.6630	54.4689	699.0230	2.24
	47.4716	70.8435	49.6569	699.0461	1.89

**Tableau 6.4.** Comparaison entre RT, QN et AG / Réseau IEEE-5 /

	$P_{G1}^{opt}$ (MW)	$P_{G2}^{pt}$ (MW)	$P_{G3}^{opt}$ (MW)	$P_{G4}^{opt}$ (MW)	$P_{G5}^{opt}$ (MW)	$P_{G6}^{opt}$ (MW)	F (\$/h)	T(s)
RT	191.9536	40.8547	17.8429	13.2328	12.5646	15.8470	801.0630	1.31
QN	139.7247	55.9680	23.4143	27.7873	19.9428	22.7129	810.6558	0.85
AG	186.9231	50.7692	17.6068	12.4481	11.5531	13.0120	798.8622	2.54
	184.3717	50.8652	20.3659	11.6856	12.2810	12.7242	799.1404	1.51

**Tableau 6.5.** Comparaison entre RT, QN et AG / Réseau IEEE-30/

	$P_{G1}^{opt}$ (MW)	$P_{G2}^{pt}$ (MW)	$P_{G3}^{opt}$ (MW)	$P_{G6}^{opt}$ (MW)	$P_{G8}^{opt}$ (MW)	$P_{G9}^{opt}$ (MW)	$P_{G12}^{opt}$ (MW)	F (\$/h)	T(s)
RT	253.3952	97.4596	94.1011	40.5436	528.1956	94.2480	170.6614	44760	1.18
QN	134.6126	117.1434	41.9767	117.1435	425.7778	117.1434	324.9086	42295	1.01
AG	138.7134	99.9144	39.2119	100.0000	513.4709	62.4044	324.9279	42201	2.15
	146.2465	74.3569	50.7484	75.0733	508.5059	75.3609	348.3446	42201	1.76

**Tableau 6.6.** Comparaison entre RT, QN et AG / Réseau IEEE-57 /

#### **6-4) CONCLUSION**

Dans ce chapitre nous avons appliqué la méthode de la Recherche Tabou à la répartition optimale des puissances actives afin d'améliorer la fonction coût totale. L'application est faite sur trois réseaux différents.

Les résultats obtenus affirment bien que la méthode de la Recherche Tabou présente un intérêt certain en termes de convergence fiable vers un optimum global et avec une minimisation de coût global réduit. La méthode de la Recherche Tabou converge plus rapidement en comparant avec les algorithmes génétiques.

Pour les différents réseaux, les résultats obtenus, comparés à ceux de la méthode classique QN, sont meilleurs en termes de minimisation de la fonction coût.

Dans le chapitre 7, nous présentons une nouvelle approche d'hybridation entre les métaheuristiques utilités et les la méthode de quasi-Newton, afin de minimiser le coût de combustible de chaque réseau traité en tenant compte les performances de chaque méthode.

***CHAPITRE VI***  
***APPLICATION DE LA RECHERCHE***  
***TABOU DANS***  
***LA REPARTITION OPTIMALE DES***  
***PUISSANCES ACTIVES***

## APPLICATION DE LA RECHERCHE TABOU DANS LA REPARTITION OPTIMALE DES PUISSANCES ACTIVES

### 6-1) INTRODUCTION

Un problème d'optimisation combinatoire est un problème qui peut s'exprimer par une fonction (dite coût) avec ou sans contraintes, à minimiser ou à maximiser sur un ensemble de définition fini ou dénombrable. C'est le cas de nombreux problèmes, dans des domaines d'applications très variés, qu'ils soient scientifiques ou techniques. La résolution satisfaisante d'un problème d'optimisation difficile, qui comporte un grand nombre de solution sous optimales, justifie souvent le recours à une « métaheuristique » puissante. Ces méthodes d'optimisation permettent de rechercher les optimums globaux d'un problème d'optimisation, en évitant le piègeage dans l'un de ses minimums locaux.

L'optimisation est un ensemble de techniques permettant de trouver les valeurs des variables qui rendent optimale une fonction de réponse, appelée aussi « fonction objectif ». Sur le plan mathématique, cela correspond à la recherche des extrémums de fonctions à plusieurs variables.

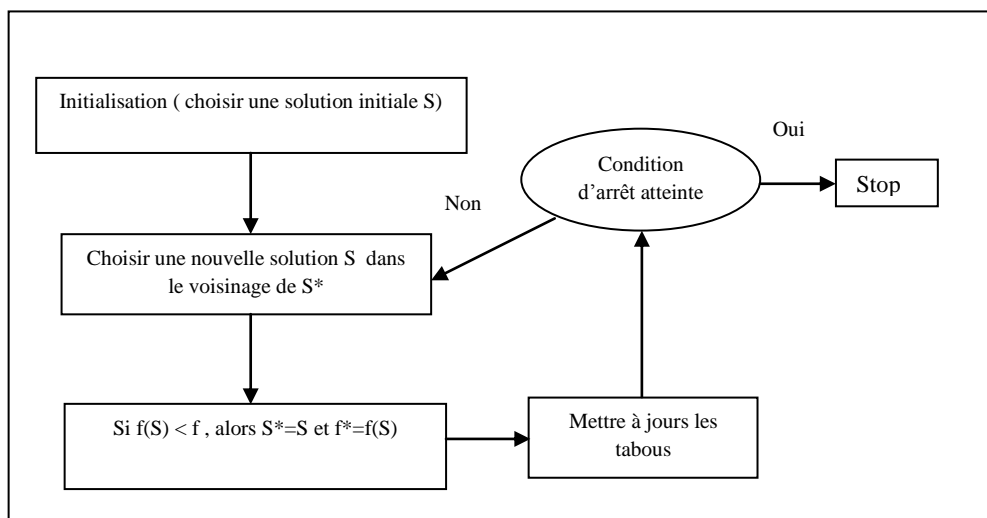


Figure 6.1. Schéma tabou

Dans ce chapitre nous nous intéressons à la méthode de la recherche tabou appliquées à l'optimisation des puissances actives dans les réseaux électriques IEEE-5, IEEE-30 et IEEE-57.

## 6-2) ORGANIGRAMME DE LA RECHERCHE TABOU

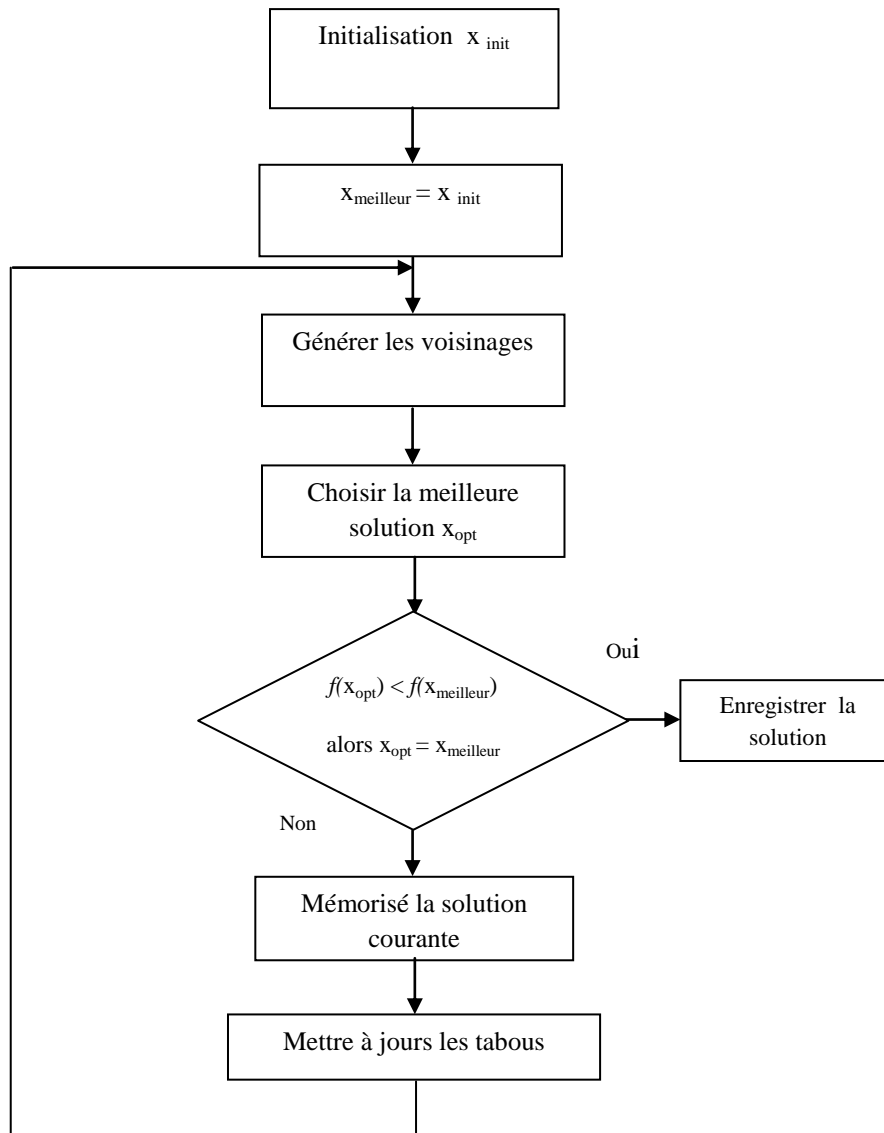


Figure 6.2. Organigramme de la recherche tabou

### 6-3) ILLUSTRATION

Nous avons appliqué la Recherche Tabou sur des réseaux différents, IEEE-5, IEEE-30 et IEEE-57.

Les paramètres de contrôle de l'algorithme de Recherche Tabou pour un réseau IEEE-5 sont :

- Nombre d'itération :  $i_{max}= 200$
- Longueur de la liste taboue :  $L= 5$
- Nombre de points de recherche autour du voisinage:  $M= 12$
- Nombre de diversifications:  $idiver = 1$
- Facteur de pénalité  $rk=1$

Les paramètres de contrôle de l'algorithme de Recherche Tabou pour un réseau IEEE-30 sont :

- Nombre d'itération :  $i_{max}= 200$
- Longueur de la liste taboue :  $L= 10$
- Nombre de points de recherche autour du voisinage:  $M= 20$
- Nombre de diversifications:  $idiver = 1$
- Facteur de pénalité  $rk=0.1$

Les paramètres de contrôle de l'algorithme de Recherche Tabou pour un réseau IEEE-57 sont :

- Nombre d'itération :  $i_{max}= 200$
- Longueur de la liste taboue :  $L=10$
- Nombre de points de recherche autour du voisinage:  $M= 45$
- Nombre de diversifications:  $idiver = 1$
- Facteur de pénalité  $rk=0.1$

Dans le but de montrer l'efficacité de l'algorithme de Recherche Tabou nous avons effectué plusieurs essais, les résultats sont présentés dans les tableaux (6.1), (6.2), (6.3)



N° d'essais	$P_{G1}^{opt}$ (MW)	$P_{G2}^{opt}$ (MW)	$P_{G3}^{opt}$ (MW)	F (\$/h)
1	51.5061	76.1626	40.3850	700.9029
2	51.5061	76.1626	40.3850	700.9029
3	51.5061	76.1626	40.3850	700.9029
4	51.5061	76.1626	40.3850	700.9029
5	51.5061	76.1626	40.3850	700.9029

**Tableau 6.1.** Résultats de RT /Réseau IEEE-5/

N° d'essais	$P_{G1}^{opt}$ (MW)	$P_{G2}^{opt}$ (MW)	$P_{G3}^{opt}$ (MW)	$P_{G4}^{opt}$ (MW)	$P_{G5}^{opt}$ (MW)	$P_{G6}^{opt}$ (MW)	F (\$/h)
1	191.9536	40.8547	17.8429	13.2328	12.5646	15.8470	801.0630
2	191.9536	40.8547	17.8429	13.2328	12.5646	15.8470	801.0630
3	191.9536	40.8547	17.8429	13.2328	12.5646	15.8470	801.0630
4	191.9536	40.8547	17.8429	13.2328	12.5646	15.8470	801.0630
5	191.9536	40.8547	17.8429	13.2328	12.5646	15.8470	801.0630

**Tableau 6.2.** Résultats de RT /Réseau IEEE-30/

N° d'essais	$P_{G1}^{opt}$ (MW)	$P_{G2}^{opt}$ (MW)	$P_{G3}^{opt}$ (MW)	$P_{G6}^{opt}$ (MW)	$P_{G8}^{opt}$ (MW)	$P_{G9}^{opt}$ (MW)	$P_{G12}^{opt}$ (MW)	F (\$/h)
1	253.3952	97.4596	94.1011	40.5436	528.1956	94.2480	170.6614	44760
2	253.3952	97.4596	94.1011	40.5436	528.1956	94.2480	170.6614	44760
3	253.3952	97.4596	94.1011	40.5436	528.1956	94.2480	170.6614	44760
4	253.3952	97.4596	94.1011	40.5436	528.1956	94.2480	170.6614	44760
5	253.3952	97.4596	94.1011	40.5436	528.1956	94.2480	170.6614	44760

**Tableau 6.3.** Résultats de RT /Réseau IEEE-57/

L'évolution de la fonction coût en fonction du nombre d'itérations est représentée par les figures (6-3), (6-4) et (6-5) pour les réseaux IEEE-5, IEEE-30 et IEEE-57 respectivement

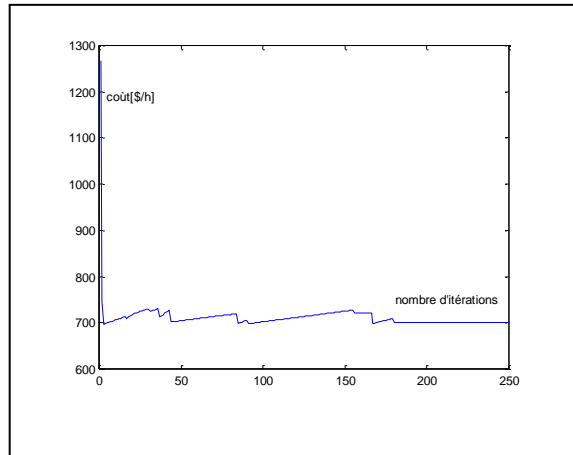


Figure 6.3. Courbe du coût total / Réseau IEEE-5

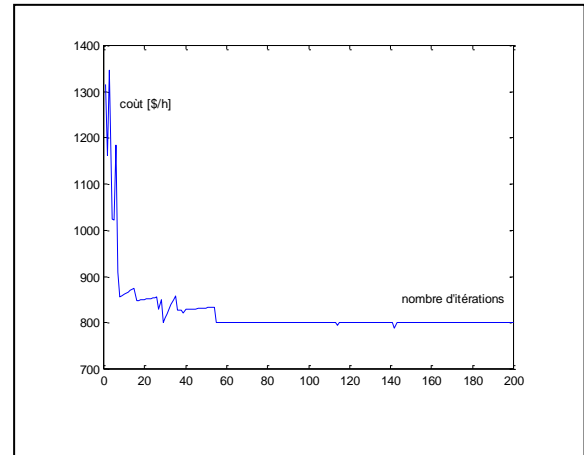


Figure 6.4. Courbe du coût total / Réseau IEEE-30

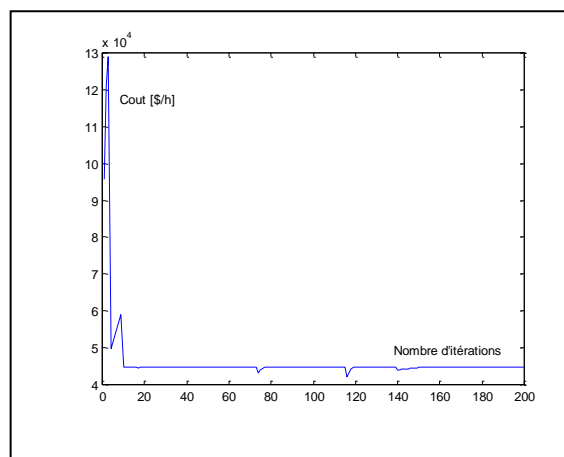


Figure 6.5. Courbe du coût total / Réseau IEEE-57

### Discussion des résultats :

Les résultats trouvés par la méthode de la Recherche Tabou confirment bien la validité et l'efficacité de cette méthode quelque soit le changement des conditions initiales dans les différents réseaux. Ces résultats ne s'éloignent pas en comparant aux résultats trouvés par les algorithmes génétiques.

Pour démontrer la puissance de la méthode Recherche Tabou, nous avons comparé cette dernière avec les résultats de l'AG et la méthode de Quasi-Newton (QN) [NAA 04, BOU 05 a, BOU 06].

	$P_{G1}^{opt}$ (MW)	$P_{G2}^{pt}$ (MW)	$P_{G3}^{opt}$ (MW)	F (\$/h)	T(s)
RT	51.5061	76.1626	40.3850	700.9029	1.10
QN	70.6216	58.5665	38.7881	705.0929	0.55
AG	41.8462	71.6630	54.4689	699.0230	2.24
	47.4716	70.8435	49.6569	699.0461	1.89

**Tableau 6.4.** Comparaison entre RT, QN et AG / Réseau IEEE-5 /

	$P_{G1}^{opt}$ (MW)	$P_{G2}^{pt}$ (MW)	$P_{G3}^{opt}$ (MW)	$P_{G4}^{opt}$ (MW)	$P_{G5}^{opt}$ (MW)	$P_{G6}^{opt}$ (MW)	F (\$/h)	T(s)
RT	191.9536	40.8547	17.8429	13.2328	12.5646	15.8470	801.0630	1.31
QN	139.7247	55.9680	23.4143	27.7873	19.9428	22.7129	810.6558	0.85
AG	186.9231	50.7692	17.6068	12.4481	11.5531	13.0120	798.8622	2.54
	184.3717	50.8652	20.3659	11.6856	12.2810	12.7242	799.1404	1.51

**Tableau 6.5.** Comparaison entre RT, QN et AG / Réseau IEEE-30/

	$P_{G1}^{opt}$ (MW)	$P_{G2}^{pt}$ (MW)	$P_{G3}^{opt}$ (MW)	$P_{G6}^{opt}$ (MW)	$P_{G8}^{opt}$ (MW)	$P_{G9}^{opt}$ (MW)	$P_{G12}^{opt}$ (MW)	F (\$/h)	T(s)
RT	253.3952	97.4596	94.1011	40.5436	528.1956	94.2480	170.6614	44760	1.18
QN	134.6126	117.1434	41.9767	117.1435	425.7778	117.1434	324.9086	42295	1.01
AG	138.7134	99.9144	39.2119	100.0000	513.4709	62.4044	324.9279	42201	2.15
	146.2465	74.3569	50.7484	75.0733	508.5059	75.3609	348.3446	42201	1.76

**Tableau 6.6.** Comparaison entre RT, QN et AG / Réseau IEEE-57 /

#### **6-4) CONCLUSION**

Dans ce chapitre nous avons appliqué la méthode de la Recherche Tabou à la répartition optimale des puissances actives afin d'améliorer la fonction coût totale. L'application est faite sur trois réseaux différents.

Les résultats obtenus affirment bien que la méthode de la Recherche Tabou présente un intérêt certain en termes de convergence fiable vers un optimum global et avec une minimisation de coût global réduit. La méthode de la Recherche Tabou converge plus rapidement en comparant avec les algorithmes génétiques.

Pour les différents réseaux, les résultats obtenus, comparés à ceux de la méthode classique QN, sont meilleurs en termes de minimisation de la fonction coût.

Dans le chapitre 7, nous présentons une nouvelle approche d'hybridation entre les métaheuristiques utilités et les la méthode de quasi-Newton, afin de minimiser le coût de combustible de chaque réseau traité en tenant compte les performances de chaque méthode.

***CHAPITRE VII***  
***LES ALGORITHMES HYBRIDES***

## LES ALGORITHMES HYBRIDES

### 7-1) INTRODUCTION

L'hybridation est une tendance observée dans de nombreux travaux réalisés sur les métaheuristiques ces dix dernières années. Elle permet de tirer profit des avantages cumulés des différentes métaheuristiques, à tel point que les métaheuristiques que nous avons vues jusqu'à présent ne sont plus que des canevas, des points de départ, pour commencer à résoudre un problème d'optimisation.

Une des techniques les plus populaires d'hybridation concerne l'utilisation de métaheuristique de type « trajectoire locale » avec des métaheuristiques à population. La plupart des applications réussies d'Algorithmes Génétiques, ou à colonies de fourmi, sont complétées par une phase de recherche locale, car c'est ce qui leur manque à l'origine. Les individus, les fourmis, tentent d'identifier les zones prometteuses de l'espace des solutions, lesquelles sont ensuite explorées plus en détail par des méthodes de recherche locale.

Une autre manière d'hybrider consiste à exécuter en parallèle plusieurs fois la même métaheuristique, mais avec des paramètres différents. Ces processus parallèles communiquent entre eux régulièrement pour échanger de l'information sur leurs résultats partiels. Cette technique est d'autant plus utilisée que s'est approfondie la recherche sur les architectures parallèles. Toutes les métaheuristiques classiques ont fait l'objet d'une implémentation parallèle, y compris l'algorithme du recuit simulé qui, bien que de nature séquentielle, a pu être parallélisé en le divisant en processus élémentaires.

Enfin une troisième forme d'hybridation combine les métaheuristiques avec des méthodes exactes. Une méthode exacte peut ainsi donner lieu à une technique efficace pour la détermination du meilleur voisin d'une solution (ce qui peut s'avérer plus judicieux que de choisir la meilleure solution parmi un petit échantillon de voisins).

Dans ce chapitre, nous proposons une méthode d'optimisation reposant sur une approche hybride. L'idée de base est de faire une combinaison entre les méthodes analytiques et les métaheuristiques d'une part et d'autre part de faire une combinaison entre les métaheuristiques.

### 7-2) HYBRIDATION D' UN ALGORITHME GENETIQUE AVEC QN

Il existe plusieurs moyens d'hybrider les AG tout en maintenant une structure de programme assez modulaire [REN 95] : Algorithme hybride selon le principe de **Darwin**, Algorithme hybride selon le principe de **Lamarck** et Algorithme hybride **Directe**.

Le principe de fonctionnement du premier algorithme hybride est le suivant : par itérations successives, les AG fournissent graduellement de meilleures estimées initiales pour des QN en ce sens que les points de départ de QN deviennent progressivement ceux qui, bien

placés à l'intérieur du bassin d'attraction du minimum global, évolueront aisément et sûrement vers cet optimum. Au moment d'évaluer la qualité de chaque individu, les AG utilisent les résultats d'un grimpeur QN dont le point initial correspond à cet individu [BOU 06 c].

Le deuxième algorithme hybride fonctionne comme suit: par itérations successives, les AG fournissent graduellement de meilleures solutions, en utilisant QN comme un mécanisme accélérateur grâce à ses propriétés. Au moment d'évaluer la qualité d'adaptation de chaque individu, les AG utilisent les résultats d'un grimpeur QN travaillant avec une estimée initiale correspondant à cet individu. Durant les phases de reproduction et de transformations génétiques (croisement, mutation) pour la génération de nouveaux individus, les AG manipulent les nouvelles solutions trouvées par QN, à la différence de la première variante dans laquelle les AG manipulent les points initiaux fournis aux QN et non les résultats fournis par ces derniers.

L'algorithme hybride **Directe** c'est une combinaison directe des AG et la méthode QN. Dans une première phase les AG explorent l'espace de recherche dans le but de découvrir des sous-espaces promoteurs et de fournir une solution grossière, à savoir une solution située à l'intérieur du bassin d'attraction du minimum global. Dans une seconde phase, QN utilise la meilleure solution fournie par les AG comme estimée initiale et poursuit la recherche suivant son propre mode d'exploitation [BOU 05 a].

## Organigramme de l'hybridation AG-QN

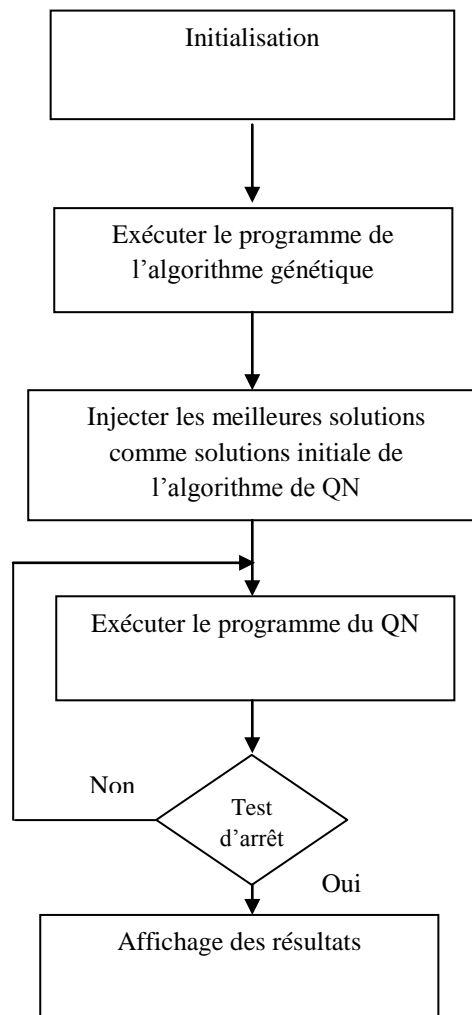


Figure 7.1. Organigramme de l'hybridation d'un AG avec QN

### 7-3) ILLUSTRATION

#### a / Simulation du réseau IEEE-5

Les paramètres de contrôle de l'algorithme sont :

- Type de codage : Codage binaire
- Taille de la population :  $N_{pop} = 30$
- Probabilité de croisement :  $pc = 0.75$
- Croisement double
- Probabilité de mutation :  $pm = 0.06$
- Mutation simple
- La longueur de chromosome :  $l = 12$  bits
- Nombre de génération :  $maxgen = 300$

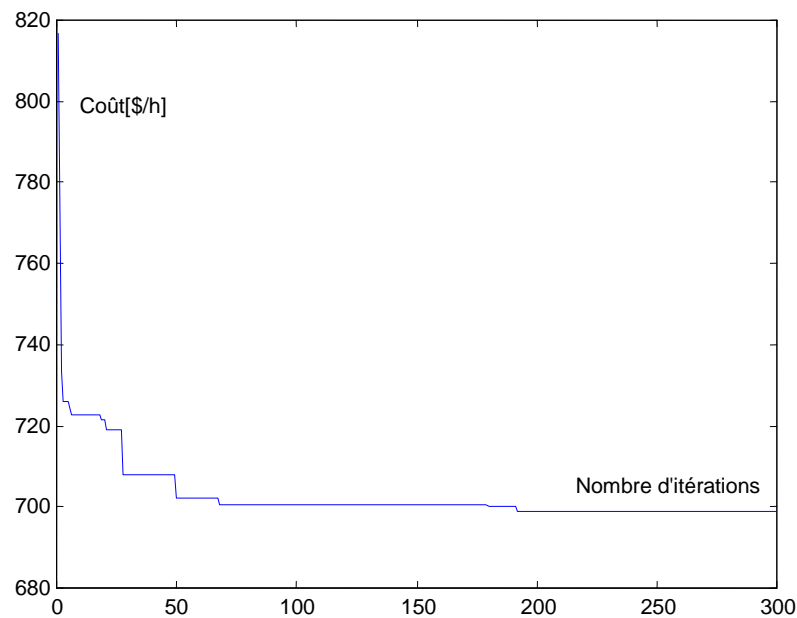


- Coefficient de pénalité :  $r_k=0.08$
- Les résultats de simulation du réseau IEEE-5 sont donnés dans le tableau 7.1

	$P_{G1}^{opt}$ (MW)	$P_{G2}^{opt}$ (MW)	$P_{G3}^{opt}$ (MW)	F (\$/h)
AG /QN hybride	45.2981	69.5705	53.1113	698.9101

**Tableau 7.1** Hybridation AG-QN/Réseau IEEE-5/

La figure 7.2 illustre l'évolution de l'algorithme en fonction des itérations



**Figure 7.2.** Coût en fonction du nombre d'itérations/ Réseau /IEEE-5/

### b/ Simulation du réseau IEEE-30

Les paramètres de contrôle de l'algorithme sont :

- Type de codage : Codage binaire
- Taille de la population :  $N_{pop} = 30$
- Probabilité de croisement :  $p_c = 0.75$
- Croisement double
- Probabilité de mutation :  $p_m = 0.06$
- Mutation simple
- La longueur de chromosome :  $l = 12$  bits
- Nombre de génération :  $maxgen = 300$

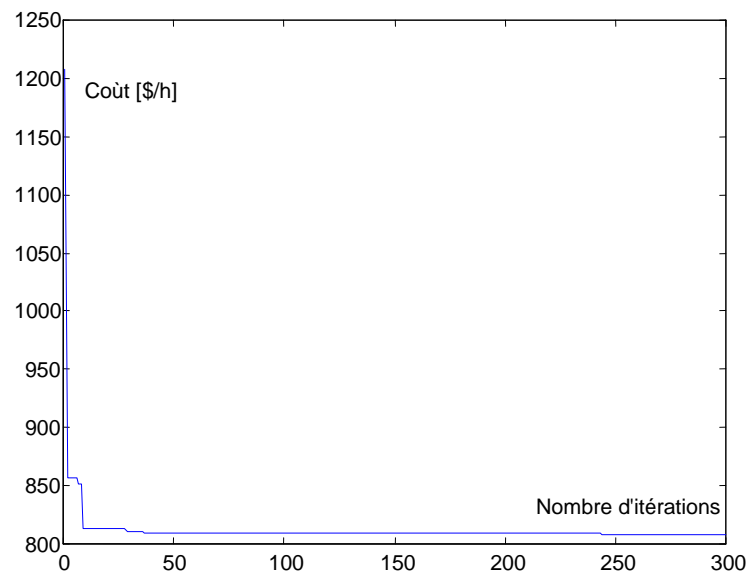
- Coefficient de pénalité :  $r_k=0.08$

Les résultats de simulation du réseau IEEE-30 sont donnés dans le tableau 7.2

	$P_{G1}^{opt}$ (MW)	$P_{G2}^{opt}$ (MW)	$P_{G3}^{opt}$ (MW)	$P_{G4}^{opt}$ (MW)	$P_{G5}^{opt}$ (MW)	$P_{G6}^{opt}$ (MW)	F (\$/h)
AG / QN hybride	194.6288	48.8498	19.6780	12.5748	9.1950	9.1950	804.0205

**Tableau 7.2** Hybridation AG-QN/Réseau IEEE-30/

La figure 7.3 illustre l'évolution de l'algorithme en fonction des itérations



**Figure 7.3.** Coût en fonction du nombre d'itérations/ Réseau /IEEE-30/

### c/ Simulation du réseau IEEE-57

Les paramètres de contrôle de l'algorithme sont :

- Type de codage : Codage binaire
- Taille de la population :  $N_{pop}= 30$
- Probabilité de croisement :  $pc = 0.8$
- Croisement double
- Probabilité de mutation :  $pm = 0.02$
- Mutation simple
- La longueur de chromosome :  $l = 12$  bits

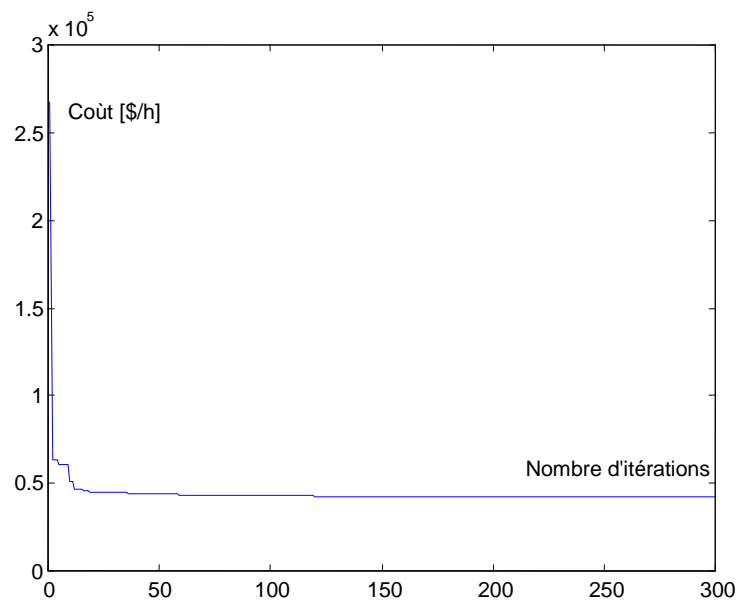
- Nombre de génération :  $\text{maxgen} = 300$
- Coefficient de pénalité :  $r_k=0.05$

Les résultats de simulation du réseau IEEE-57 sont donnés dans le tableau 7.3

	$P_{G1}^{\text{opt}}$ (MW)	$P_{G2}^{\text{opt}}$ (MW)	$P_{G3}^{\text{opt}}$ (MW)	$P_{G6}^{\text{opt}}$ (MW)	$P_{G8}^{\text{opt}}$ (MW)	$P_{G9}^{\text{opt}}$ (MW)	$P_{G12}^{\text{opt}}$ (MW)	F (\$/h)
AG / QN hybride	140.3763	89.0284	43.5614	89.0284	490.0618	89.0283	337.5974	42170

**Tableau 7.3** Hybridation AG-QN/Réseau IEEE-57

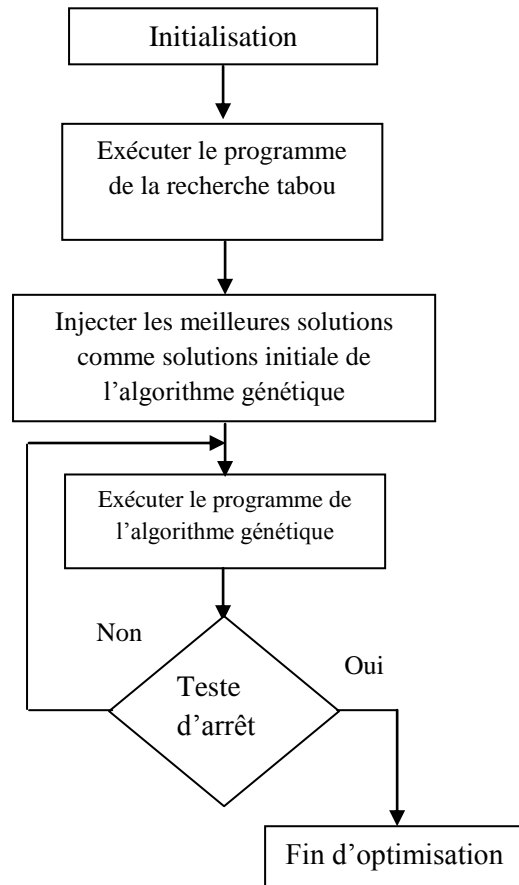
La figure 7.4 illustre l'évolution de l'algorithme en fonction des itérations



**Figure 7.4.** Coût en fonction du nombre d'itérations/ Réseau IEEE-57/

#### 7-4) HYBRIDATION DE L'AG AVEC LA RECHERCHE TABOU

Dans cette hybridation, on lance l'algorithme de la Recherche Tabou pour générer une solution. La meilleure solution sera en suite injectée comme population initiale dans l'algorithme génétique.



**Figure 7.5.** Organigramme de l'algorithme génétique hybride avec la recherche tabou

#### a / Simulation du réseau IEEE-5

Les paramètres de contrôle de l'algorithme hybride sont:

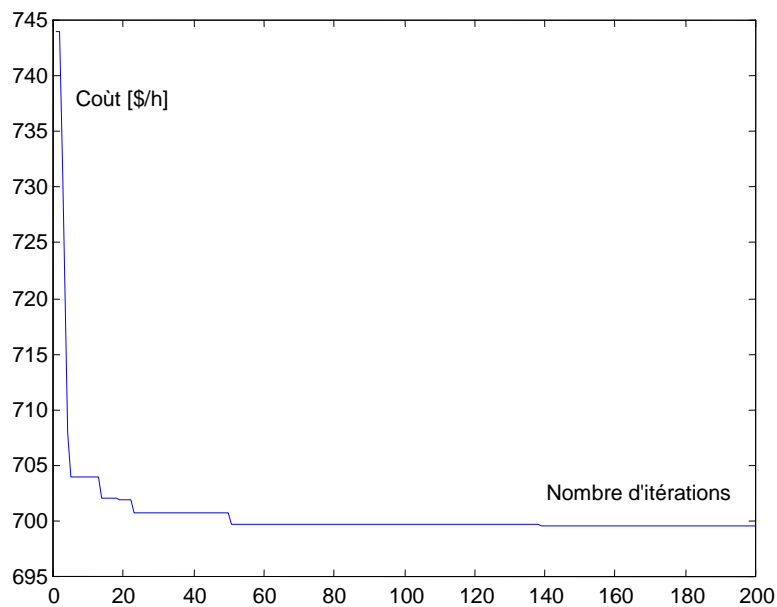
- Type de codage : Codage binaire
- Taille de la population :  $N_{pop} = 30$
- Probabilité de croisement :  $pc = 0.75$
- Croisement double
- Probabilité de mutation :  $pm = 0.06$
- Mutation simple

- La longueur de chromosome :  $l = 12$  bits
- Nombre d'itération :  $i_{\max} = 200$
- Longueur de la liste taboue :  $L = 5$
- Nombre de points de recherche autour du voisinage:  $M = 12$
- Nombre de diversifications:  $i_{\text{diver}} = 1$

	$P_{G1}^{\text{opt}}$ (MW)	$P_{G2}^{\text{opt}}$ (MW)	$P_{G3}^{\text{opt}}$ (MW)	F (\$/h)
AG/RT hybride	46.3516	62.9451	58.6935	699.5103

**Tableau 7.4** Hybridation AG-RT /Réseau IEEE-5/

La figure 7.6 illustre l'évolution de la fonction coût en fonction du nombre d'itérations.



**Figure 7.6** Coût en fonction du nombre d'itérations / Réseau IEEE-5/

**b / Simulation du réseau IEEE-30:**

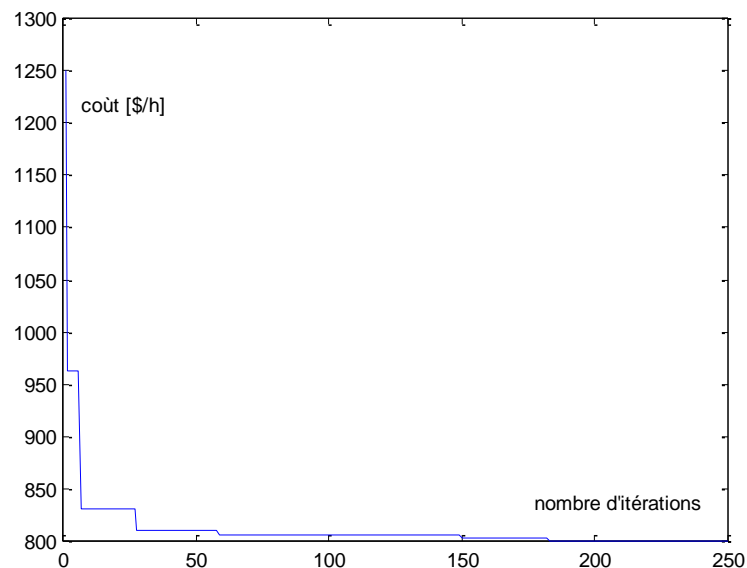
Les paramètres de contrôle de l'algorithme sont :

- Type de codage : Codage binaire
- Taille de la population :  $N_{pop} = 30$
- Probabilité de croisement :  $pc = 0.75$
- Croisement double
- Probabilité de mutation :  $pm = 0.05$
- Mutation simple
- La longueur de chromosome :  $l = 12$  bits
- Facteur de pénalité = 0.095
- Nombre d'itération :  $i_{max} = 200$
- Longueur de la liste taboue :  $L = 10$
- Nombre de points de recherche autour du voisinage:  $M = 20$
- Nombre de diversifications:  $idiver = 1$
- Facteur de pénalité = 0.1

	$P_{G1}^{opt}$ (MW)	$P_{G2}^{opt}$ (MW)	$P_{G3}^{opt}$ (MW)	$P_{G4}^{opt}$ (MW)	$P_{G5}^{opt}$ (MW)	$P_{G6}^{opt}$ (MW)	F (\$/h)
AG/RT hybride	182.4908	44.5421	22.8803	16.1538	13.8291	12.4376	800.2767

**Tableau 7.5** Hybridation AG -RT /Réseau IEEE-30

La figure 7.8 illustre l'évolution de la fonction coût en fonction du nombre d'itérations.



**Figure 7.8.** Coût en fonction du nombre d'itérations/ Réseau IEEE-30/

**c / Simulation du réseau IEEE-57:**

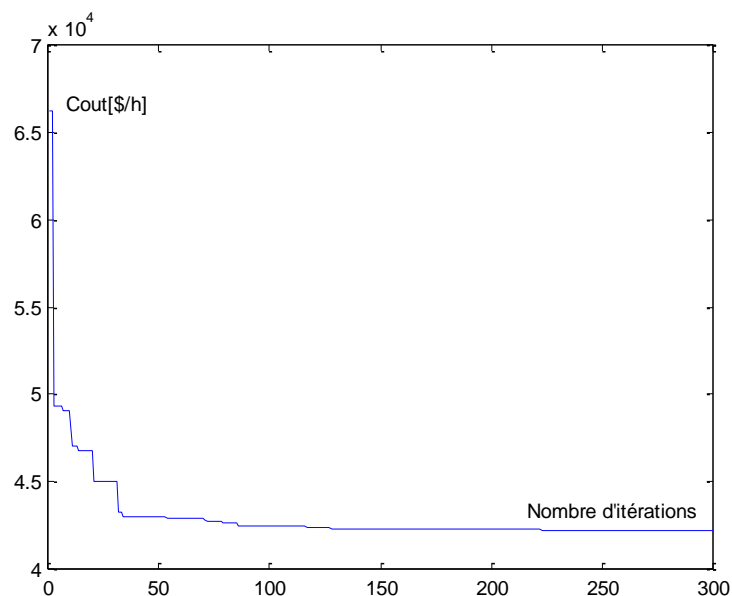
Les paramètres de contrôle de l'algorithme sont :

- Type de codage : Codage binaire
- Taille de la population :  $N_{pop} = 30$
- Probabilité de croisement :  $pc = 0.75$
- Croisement double
- Probabilité de mutation :  $pm = 0.0098$
- Mutation simple
- La longueur de chromosome :  $l = 12$  bits
- Facteur de pénalité = 0.028
- Nombre d'itération :  $i_{max} = 200$
- Longueur de la liste taboue :  $L = 5$
- Nombre de points de recherche autour du voisinage:  $M = 45$
- Nombre de diversifications:  $idiver = 1$
- Facteur de pénalité=0.1

	$P_{G1}^{opt}$ (MW)	$P_{G2}^{opt}$ (MW)	$P_{G3}^{opt}$ (MW)	$P_{G6}^{opt}$ (MW)	$P_{G8}^{opt}$ (MW)	$P_{G9}^{opt}$ (MW)	$P_{G12}^{opt}$ (MW)	F (\$/h)
AG/RT hybride	148.2892	97.1834	42.5183	74.7205	509.5681	75.9688	330.4420	42190

**Tableau 7.6** Hybridation AG-RT /Réseau IEEE-57/

La figure 7.9 illustre l'évolution de la fonction coût en fonction du nombre d'itérations.



**Figure 7.9.** Coût en fonction du nombre d'itérations/ Réseau IEEE-57/

### 7-5) HYBRIDATION DE LA METHODE RT AVEC QN

Dans cette procédure l'algorithme de Recherche Tabou explore l'espace de recherche dans le but de découvrir des sous-espaces prometteurs et de fournir une solution grossière, à savoir une solution située à l'intérieur du bassin d'attraction du minimum global. Dans une seconde phase, QN utilise la meilleure solution fournie par les AG comme estimée initiale et poursuit la recherche suivant son propre mode d'exploitation.

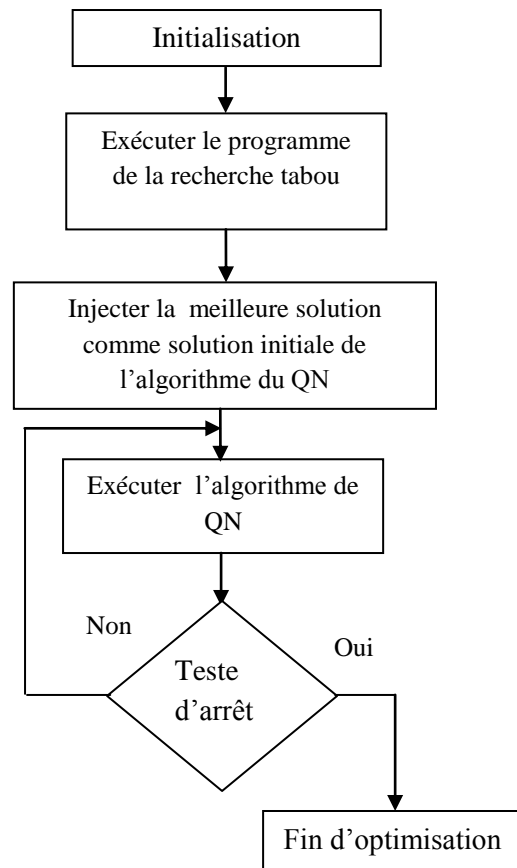


Figure 7.10 Organigramme de l'hybridation RT-QN

#### a / Simulation du réseau IEEE-5

Les paramètres de contrôle de la recherche tabou sont :

- Nombre d'itération :  $i_{max} = 200$
- Longueur de la liste taboue :  $L = 5$
- Nombre de points de recherche autour du voisinage:  $M = 12$
- Nombre de diversifications:  $idiver = 1$



	$P_{G1}^{opt}$ (MW)	$P_{G2}^{opt}$ (MW)	$P_{G3}^{opt}$ (MW)	F (\$/h)
RT/QN hybride	46.0606	68.9860	52.9334	698.9164

**Tableau 7.8.** Hybridation RT- QN /Réseau IEEE-5/

### b / Simulation du réseau IEEE-30

Les paramètres de contrôle de l'algorithme sont :

- Nombre d'itération :  $i_{max} = 200$
- Longueur de la liste taboue :  $L = 10$
- Nombre de points de recherche autour du voisinage:  $M = 20$
- Nombre de diversifications:  $idiver = 1$

	$P_{G1}^{opt}$ (MW)	$P_{G2}^{opt}$ (MW)	$P_{G3}^{opt}$ (MW)	$P_{G4}^{opt}$ (MW)	$P_{G5}^{opt}$ (MW)	$P_{G6}^{opt}$ (MW)	F (\$/h)
TR/QN hybride	192.7569	45.9362	21.5999	13.4604	9.9393	10.4288	804.4716

**Tableau 7.9.** Hybridation RT- QN /Réseau IEEE-30 /

### c / Simulation du réseau IEEE-57

Les paramètres de contrôle de l'algorithme sont :

- Nombre d'itération :  $i_{max} = 200$
- Longueur de la liste taboue :  $L = 10$
- Nombre de points de recherche autour du voisinage:  $M = 45$
- Nombre de diversifications:  $idiver = 1$

	$P_{G1}^{opt}$ (MW)	$P_{G2}^{pt}$ (MW)	$P_{G3}^{opt}$ (MW)	$P_{G6}^{opt}$ (MW)	$P_{G8}^{opt}$ (MW)	$P_{G9}^{opt}$ (MW)	$P_{G12}^{opt}$ (MW)	F (\$/h)
RT/QN hybride	140.3763	89.0284	43.5614	89.0284	490.0618	89.0283	337.5974	42170

**Tableau 7.10.** Hybridation RT- QN /Réseau IEEE-57/

Une analyse rationnelle nécessite une comparaison entre les résultats obtenus respectivement par les métaheuristiques de base et les métaheuristiques hybrides réalisés dans l'étude des réseaux IEEE-5, IEEE-30 nœuds et IEEE-57 nœuds donnée par les tableaux ci-dessous.

	AG		QN	RT	AG/QN	AG/RT	RT/QN
	C. binaire	C .réel					
F (\$/h)	698.9118	698.9127	705.0929	700.9029	698.9010	699.5103	698.9164
T (s)	2.24	1.89	0.55	1.10	2.45	1.35	1.89

**Tableau 7.11.** Résultats du réseau IEEE-5

	AG		QN	RT	AG/QN	AG/RT	RT/QN
	C. binaire	C .réel					
F (\$/h)	798.5051	798.2691	830.4255	801.0630	804.0205	800.2767	804.4716
T (s)	2.54	1.51	0.85	1.31	3 .02	1.99	1.35

**Tableau 7.12.** Résultats du réseau IEEE-30

	AG		QN	RT	AG/QN	AG/RT	RT/QN
	C. binaire	C .réel					
F (\$/h)	42201	42201	46934	44760	42170	42190	42170
T (s)	2.15	1.76	1.01	1.18	3.76	1.29	2.12

**Tableau 7.13.** Résultats du réseau IEEE-57

### Discussion des résultats

L'hybridation entre un algorithme génétique (AG) et une méthode analytique Quasi-Newtonienne (QN) permet de marier leurs avantages et de réduire leurs inconvénients. On utilise à la fois les capacités d'exploration et la puissance d'exploitation de QN [NAA 08 d].

L'hybridation entre les algorithmes génétiques (AG) et la méthode de la Recherche Tabou (RT) a donné des bons résultats dans les différents réseaux traités.

Quelque soit le changement des conditions initiales, l'hybridation entre la Recherche Tabou (RT) et méthode Quasi-Newtonienne (QN) a donné les mêmes résultats dans les trois réseaux traités.

### 7-6) CONCLUSION

Les problèmes d'optimisation requièrent des méthodes robustes, efficaces et flexibles. Afin de s'affranchir de la complexité du problème d'adaptation, de réduire le coût de réalisation et de simplifier le modèle étudié et son système d'alimentation, nous avons proposé l'utilisation de deux méthodes stochastiques originales d'optimisation qui sont : les algorithmes génétiques et la recherche tabou.

## ***CONCLUSION GENERALE***

## Conclusion Générale

Le travail présenté dans cette thèse, porte sur l'application des méthodes métaheuristiques d'optimisation combinatoire dans l'analyse des réseaux électriques. L'utilisation de ces méthodes modernes dans la répartition optimale des puissances actives nécessite une vérification complète des différents paramètres afin de répondre à la minimisation de la fonction objectif en tenant compte des contraintes techniques et économiques.

Une première phase de ce travail a consisté à comprendre, maîtriser puis programmer les métaheuristiques de base (Algorithmes Génétiques, Recherche Tabou). Les programmes conçus ont été d'abord testés et validés sur un certain nombre de fonctions tests avant d'être appliqués au problème d'optimisation de puissance active dans un réseau électrique. Nous avons appliqué pour notre problème d'optimisation un Algorithme Génétique avec deux types de codage : un codage binaire et un codage réel sur trois réseaux standards IEEE-5, IEEE-30 et IEEE-57 nœuds. Les résultats confirment bien la validité et l'efficacité de l'Algorithme Génétique, avec une amélioration du temps de calcul pour la représentation réelle de l'AG.

Dans une seconde phase, nous avons appliqué une autre méthode métaheuristique (Recherche Tabou). Les résultats obtenus présentent un intérêt certain en termes de convergence fiable vers une minimisation de coût global. Une comparaison entre les résultats obtenus des méthodes métaheuristiques et une méthode analytique quasi-Newtonienne montre l'aptitude des stratégies métaheuristiques à atteindre la valeur optimale du coût de production. La méthode quasi-Newton est toujours la plus rapide en temps de calcul.

Dans la dernière partie de ce travail, nous avons proposé une hybridation entre les métaheuristiques de base (AG-RT) et les métaheuristiques avec une méthode analytique quasi-Newtonienne (RT-QN, AG-QN). Les métaheuristiques hybrides étudiées présentent globalement l'avantage de mieux minimiser le coût de production que les métaheuristiques de base, tout en respectant les limites des puissances générées. L'hybridation AG-QN et RT-QN permet d'obtenir des solutions intéressantes du point de vue coût optimal et temps de calcul. Les méthodes métaheuristiques et analytiques de quasi-Newton restent néanmoins complémentaires.

### *Perspectives*

Cependant les conclusions tirées ne peuvent pas être considérées comme étant des conclusions définitives. Certes, elles sont les résultats de l'analyse et interprétation de nombreuses simulations validées, mais qui à notre point de vue peuvent encore être améliorées à travers :

- L'application du problème de la répartition optimale des puissances actives où toutes les contraintes de sécurité sont prises en compte.
- L'utilisation d'autres méthodes métaheuristique telles que le Recuit Simuler, Essaim de Particules...etc.
- Une approche unifiée entre la répartition optimale des puissances actives et réactive, sous des contraintes de sécurité.



***BIBLIOGRAPHIE***

**BIBLIOGRAPHIE**

- [ARV 71] N.V.ARVANTIDIS & J.ROSING, The object function in real power dispatching, IEEE Trans on PAS, Vol Pas 90, July-August 1971.
- [BÄC 91] T.BÄCK, F. HOFFMEISTER, H-P. SCHWEFEL, A Survey of evolutionary strategies. Proc. Of 4<sup>th</sup> Intl. Conference on Genetic Algorithms (ICGA '91), R. BELEW, L. BOOKER (Eds), Morgan Kaufmann, p. 2-9, 1991.
- [BÄC 93a] T. BÄCK, H-P. SCHWEFEL, An overview of evolutionary algorithms for parameter optimization. Evolutionary Computation 1(1): 1-24, 1993.
- [BÄC 93b] T.BÄCK, F. HOFFMEISTER, H-P. SCHWEFEL, (Eds), Applications of evolutionary algorithms. Report of the System Analysis Research Group (Sys), Univ. of Dortmund, ftp://lumpi. Informatik. Uni-dortmund. De/pud/ EA/papers/ ea-app.ps.gs, 1993.
- [BOU 04] T. BOUKTIR, Application de la programmation orientée objet à l'optimisation de l'écoulement de puissance, thèse Es Science, soutenue à Batna 2004.
- [BOU 05 a] H. BOUZEOUDJA, B. NAAMA, A. CHAKER: Traitement Parallèle des méthodes Quasi – Newtoniennes pour l'Optimisation des Puissances Actives. International Conference on Electrotechnics, ICEL, 2005, USTO Oran, November 13-14, 2005
- [BOU 05 b] H.BOUZEOUDJA, A. CHAKER, A. ALLALI, B. NAAMA: Economic Dispatch Solution Using a Real-Coded Genetic Algorithm. Acta journal, Electrotechnica et Informatica, Vol. 5, N°4, 2005, ISSN 1335-8243
- [BOU 06 ] H. BOUZEOUDJA, Optimisation des puissances actives par les algorithmes génétiques, thèse d'état, soutenue à SBA, 2006.
- [CER 85] V.CERNY, A thermodynamical approach to the travelling salesman problem: an efficient simulated annealing algorithm. Journal of Optimization Theory and Applications 45: 41-51, 1985.
- [COL 91] A. COLORNI, M. DORIGO & V. MANIEZZO, Distributed Optimization by Ant Colonies, actes de la première conférence européenne sur la vie artificielle, Paris, France, Elsevier Publishing, 134-142, 1991.
- [CHA 79] P.A. CHAOREL, F. RENAUD, Programmation dynamique appliquée au problème du dispatching économique, Journée d'information sur l'utilisation de l'ordinateur dans l'économie énergétique. ASE, Dättwil, 1979, pp. 185- 200.
- [CHA] WISMER CHATERGY , Introduction to programming , North- Holland



- [CHA 95] A. CHAKER , O.LAOEUR, S.M. DJABER & O.BEKHCHI ,optimisation des puissances actives et réactives d'un réseau électrique , Proceeding of the CMSE S'95 Guelma , 1995.
- [DEN 90] J.L. DENEUBOURG, S. ARON, S. GOSS & J.-M. PASTEELS, The self-organizing exploratory pattern of the Argentine ant, Journal of Insect Behavior, volume 3, page 159, 1990
- [DEJ 93] K.A. DE JONG, SPEARS, On the state of evolutionary computation. Proc. Of International Conference on Genetic Algorithms (ICGA'93), p; 618-623, 1993.
- [DEJ 94] K.A. DE JONG, Genetic Algorithms : A25 year perspective. Dans J.M. Zurada, R.J. Marks II, C. J. Robinson (Eds), Computational Intelligence-Imitating Life, IEEE Press, p. 125-134, 1994
- [DOM 68] H. W. DOMMEL , W.S. TINNEY, Optimal power flow solutions. IEEE Trans. PAS – 87, 1968, pp . 1866-1876.
- [DOC a] document internet : Principe mathématiques et utilisation des algorithmes génétiques [w3.toulouse.inra.fr/Centre/esr/CV/bontemps/WP-Algo\\_Gene.html](http://w3.toulouse.inra.fr/Centre/esr/CV/bontemps/WP-Algo_Gene.html)
- [DOC b] document internet : A. KHIAT & B. SEFIANE , algorithmes génétiques . <http://www-chez.com/psodu>
- [DOC c] document internet : <http://www.livia.estmtt.Ca/common/publication-filtres/2002/Ming-Nadia .PDF>.
- [DOR 92] M. DORIGO, Optimization, Learning and Natural Algorithms, PhD thesis, Politecnico di Milano, Italie, 1992.
- [ERI 07] ERIC TAILLARD, Introduction aux méta-heuristiques, Heig-vd , Haute école d'ingénieure et de gestion du canton de Vaud, 3<sup>e</sup> cycle romand en informatique, Genève, 5-6 Juin 2007 .
- [FOG 66] L.J.FOGEL, A.J. OWENES, M.J. WALSH, Artificial intelligence through simulated evolution, Wiley, New York, 1966.
- [FOG 94] D.B. FOGEL, Evolutionary programming : an introduction and some current directions. Statistics and Computing 4: 11- 129, 1994.
- [GLO 86] F.GLOVER, Future paths for integer programming and links to artificial intelligence. Computers and Operations Research 13: 533-549,1986
- [GLO 89] F.GLOVER, Tabu search: part I ORSA J. on Computing 1 (3): 190-206, 1989.
- [GLO 89] F.GLOVER, Tabu search: part I ORSA J. on Computing 2 (1): 190-206, 1989.
- [GLO 95] F.GLOVER, J.P. KELLY, M, LAGUNA, Genetic algorithms and tabu search-hybrids for optimization. Computers & Operations Research 22, 1995.

- [GOS 89] S. GOSS, S. ARON, J.-L. DENEUBOURG & J.-M. PASTEELS, The self-organized exploratory pattern of the Argentine ant, *Naturwissenschaften*, volume 76, pages 579-581, 1989
- [GOL 89] D.E GOLDBERG, Genetic algorithms in search, optimization, and machine learning, Addison-Wesley, New York, 1989.
- [GLO 97] F.GLOVER, M, LAGUNA, Tabu search, Kluwer Academic Publishers, 1997.
- [HAN 86] P.HANSAN, The steepest ascent mildest descent heuristic for combinatorial programming. Congress on Numerical Methods in Combinatorial Optimization, Capri, Italie, 1986.
- [HOL 75] J.H. HOLLAND, Adaptation in natural and artificial systems. The University of Michigan Press, Ann Arbor, 1975.
- [HOL 92 ] J.H. HOLLAND, Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence. MIT Press/Bradford Books, 2<sup>nd</sup> Edition, Cambridge, MA, 1992.
- [HAN] M. HANNOUN & A. AMROUCHE , Genetica un environnement de développement destiné aux algorithmes génétiques . El Harrach –Alger [www.emse.fr/~hannoun/home/les](http://www.emse.fr/~hannoun/home/les) algorithmes génétiques.
- [KIR 83] S.KIRKPATRICK, C.D. GELATT, P.M. VECCHI, Optimization by simulated annealing. *Science* 220: 671-680, 1983.
- [LAH 07] M. LAHDEB, Théorie et applications de méthodes d'hybridations métaheuristiques dans les réseaux électriques, Mémoire de Magister, soutenu à Laghouat 2007.
- [LAO] M. LAOUER, Optimisation des puissances actives et réactives dans un réseau électrique, Thèse de magister USTO.
- [LIA 78] T.E. DY LIACCO, System security: the computer's role, *IEEE Spectrum*, June 1978, P. 43-56.
- [MOR 73] J.J. MORF, Amélioration de la répartition des puissances réactives. ASE , Utilisation de calculateurs de processus dans les centrales et les réseaux électriques, Lausanne, Suisse, 1973,pp . 281-295.
- [MIN 83] MICHEL MINOUX , Programmation mathématiques théorie et algorithmes , Tome I- Dunod 1983.
- [NAA 04] B. NAAMA, Application des algorithmes génétique dans un réseau d'énergie électrique. Mémoire de magistère, université e Sidi Bel Abbés, Juillet 2004.
- [NAA 05 a] B. NAAMA, H.BOUZEBOUDJA, A.CHAKER, Y.RAMDANI : Optimisation des puissances actives par les algorithmes génétiques, influence du

coefficient de pénalité sur la convergence. Séminaire international sur les systèmes Electromécaniques SISEM'05, Annaba 22-23 novembre 2005.

[NAA 05b] B. NAAMA, H. BOUZEBODJA, A. CHAKER : Etude comparatives des deux méthodes d'optimisations des puissances actives les algorithmes génétiques et la programmation non linéaire. First International conference on Electrical Systems PCSE'05, Oum El Bouaghi 9-11 May 2005.

[NAA 07] B. NAAMA, H. BOUZEBODJA, Y. RAMDANI, A. CHAKER Optimisation des puissances dans un réseau d'énergie électrique par intelligence Artificielle. 4th International Conference on Computer Integrated Manufacturing CIP'2007 Setif 03-04 novembre 2007.

[NAA 08] B. NAAMA, H. BOUZEBODJA, Y. RAMDANI, A. CHAKER : Hybrid Approach to the Economic Dispatch Problem Using a Genetic and a Quasi-Newton Algorithms. Acta journal, Electrotechnica et Informatica, Vol. 8, N°3, 2008, ISSN 1335-8243.

[PIE 83] A. C. PIERRE , Optimisation des puissances actives et réactives par la programmation linéaire dans les réseaux électriques à haute tension. Thèse Es science, soutenue à Lausanne EPFL 1983.

[PES 68] J. PESCHON, D.S. PIERCY, W.S. TINNEY, O.J. TREIT, M. CUENOD, Optimal control of reactive power flows. IEEE Trans. PAS – 87, 1968, pp . 40 – 48.

[RAH 85] M. RAHLI, La commande de la répartition optimale des puissances actives dans un réseau d'énergie électrique par la programmation linéaire, thèse de magister, soutenue à USTO, 1985.

[RAH 95 a] M. RAHLI, contribution à l'étude de la répartition optimale des puissances actives dans un réseau d'énergie électrique, thèse de doctorat d'état USTO 1995.

[RAH 95 b] M. RAHLI, Application d'une nouvelle méthode programmation non linéaire à la répartition économique des puissances actives dans un réseau ouest algérien, Proceeding of the CMSE S'95 Guelma, 1995.

[REC 73] I. RECHENBERG, Evolutionasstrategie : optimierung technischer systeme nach prinziping der biologischen evolution. Formman- Holzboog ? Stuttgart, 1973.

[REN 95] J.M. RENDERS. Algorithmes génétiques et réseaux de neurones. Addison-Wesley, 1995.

[SAS 68] A.M. SASSON, Non linear programming applications to power Systems. Proc. Symposium Helors- Ifors, Athens , 1968.

[SCH 97b] M. SCHOENAUER, Z. MICHALEWICZ, Evolutionary computation: an introduction. Control and Cybernetics 26 (3) : 307-338, 1997.

[SCH 81] H.P. SCHWEFEL, Numerical optimization of computer models, John Wiley & Sons, 1981.

[SPA 99] A. SPALANZANI , Algorithmes évolutionnaire pour l'études de la robustesse des systèmes de reconnaissance automatique de la parole . Thèse de doctorat d'état. 1999 Université Joseph Fourier- Grenoble I

[TSA 71] N. F. TSANG , A technique for optimizing real and reactive power schedules . University of Arkansas Fayetteville, Arkansas 1971.

[ZIM 97]R. ZIMMERMAN, D. GAN. Matpower: A MATLAB power system simulation package. 1997

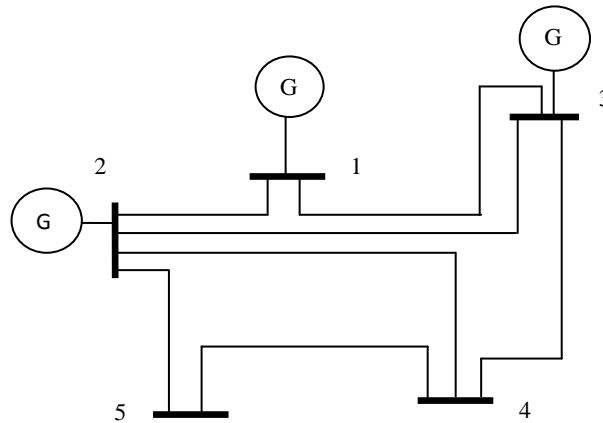
[http:// www.pserc.cornell.edu/matpower/](http://www.pserc.cornell.edu/matpower/)

## ANNEXE

## Les données des réseaux

## 1) Les données du réseau IEEE-5 :

Soit le schéma d'un réseau IEEE 5-nœuds dont 3 nœuds générateurs est donné par la figure suivante :



**Figure 1.** Réseau IEEE 5-nœuds

pq	$Z_{pq}$ (PU)	$y_{pq}/2$ (PU)
1-2	$0.02 + j0.06$	$0 + j0.030$
1-3	$0.08 + j0.24$	$0 + j0.025$
2-3	$0.06 + j0.18$	$0 + j0.020$
2-4	$0.06 + j0.18$	$0 + j0.020$
2-5	$0.04 + j0.12$	$0 + j0.015$
3-4	$0.01 + j0.03$	$0 + j0.010$
4-5	$0.08 + j0.24$	$0 + j0.025$

**Tableau 1.** Les données des lignes

Les tensions planifiées ainsi que les puissances générées et consommées sont classées dans le tableau suivant :

Nœuds	Tensions (PU)	Puissances générées		Puissances consommées	
		$P_G$ (MW)	$Q_G$ (Mvars)	$P_{ch}$ (MW)	$Q_{ch}$ (Mvars)
1	1.060 + j0.00	-	-	0	0
2	1.055 - j0.042	40.0	30.0	20	10
3	1.042 - j0.067	30.0	10.	45	15
4	1.038 - j0.075	0	0	40	5
5	1.025 - j0.096	0	0	60	10

**Tableau 2.** Valeurs planifiées

N° du nœud	$P_G^{\min}$ [MW]	$P_G^{\max}$ [MW]	a [\$/MW <sup>2</sup> h]	b [\$/MW <sup>2</sup> h]	c [\$/MW <sup>2</sup> h]
1	30	120	0.0060	2.0	140
2	20	80	0.0075	1.8	120
3	10	60	0.0070	1.5	80

**Tableau 3.** Coefficients des fonctions coût et les limites des puissances

La consommation totale :  $P_{ch} = 165$  MW

Les résultats de l'écoulement de puissance par la méthode de Gauss- Seidel (avec une précision de  $10^{-5}$ ) sont :

Les pertes actives totales :  $P_L = 2.9762$  MW

Les pertes réactives totales :  $Q_L = -22.9404$  Mvar

## 2) Les données du réseau IEEE-30 :

Soit le schéma d'un réseau IEEE 30 –nœuds dont 6 nœuds générateurs est donné par la figure suivante :

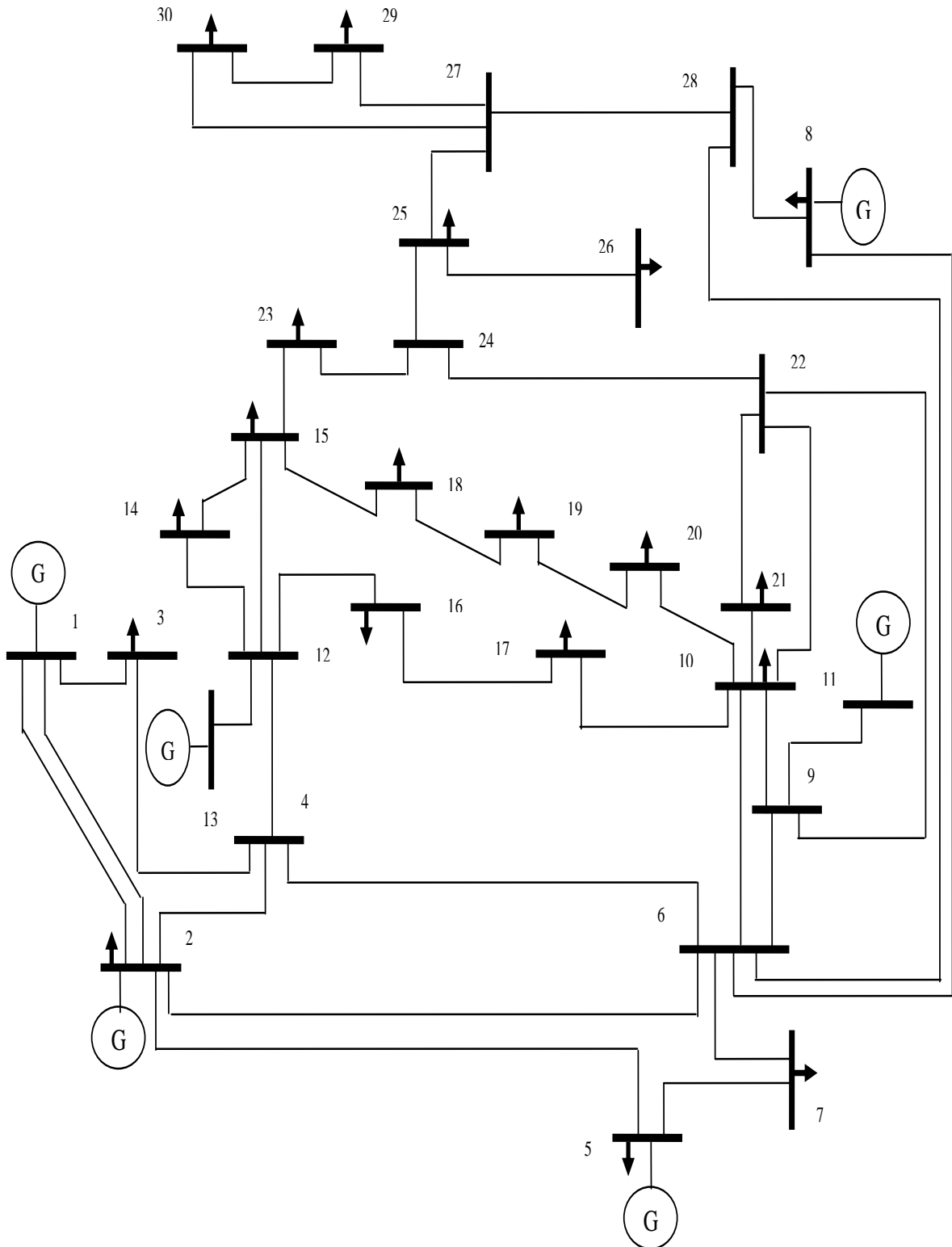


Figure 2. Réseau IEEE 30-nœuds

pq	$Z_{pq}$ (PU)	$y_{pq}/2$ (PU)
1-2	0.02+j0.06	0+ j0.03
1-3	0.05+j0.19	0+ j0.02
2-4	0.06+j0.17	0+ j0.02
3-4	0.01+j0.04	0.00
2-5	0.05+j0.2	0+ j0.02
2-6	0.06+j0.18	0+ j0.02
4-6	0.01+j0.04	0.00
5-7	0.05+j0.12	0+ j0.01
6-7	0.03+j0.08	0+ j0.01
6-8	0.01+j0.04	0.00
6-9	0.00+j0.21	0.00
6-10	0.00+j0.56	0.00
9-11	0.00+j0.21	0.00
9-10	0.00+j0.11	0.00
4-12	0.00+j0.26	0.00
12-13	0.00+j0.14	0.00
12-14	0.12+j0.26	0.00
12-15	0.07+j0.13	0.00
12-16	0.09+j0.2	0.00
14-15	0.22+j0.2	0.00
16-17	0.08+j0.19	0.00
15-18	0.11+j0.22	0.00
18-19	0.06+j0.13	0.00
19-20	0.03+j0.07	0.00
10-20	0.09+j0.21	0.00
10-17	0.03+j0.08	0.00
10-21	0.03+j0.07	0.00
10-22	0.07+j0.15	0.00
21-22	0.01+j0.02	0.00



15-23	0.10+j0.20	0.00
22-24	0.12+j0.18	0.00
23-24	0.13+j0.27	0.00
24-25	0.19+j0.33	0.00
25-26	0.25+j0.38	0.00
25-27	0.11+j0.21	0.00
28-27	0.00+j0.40	0.00
27-29	0.22+j0.42	0.00
27-30	0.32+j0.60	0.00
29-30	0.24+j0.45	0.00
8-28	0.06+j0.20	0+ j0.02
6-28	0.02+j0.06	0+ j0.01

**Tableau 4.** Les données des lignes

Les tensions planifiées ainsi que les puissances générées et consommées sont classées dans le tableau suivant :

Nœuds	Tensions (PU)	Puissances générées		Puissances consommées	
		$P_G$ (MW)	$Q_G$ (Mvars)	$P_{Ch}$ (MW)	$Q_{Ch}$ (Mvars)
1	1.06+j0.00	-	-	0.00	0.00
2	1.00+J0.00	60.00	23.00	21.7	12.7
3	1.00+J0.00	0.00	0.00	4.2	1.2
4	1.06+j0.00	0.00	0.00	7.6	1.6
5	1.00+J0.00	30.0	50.0	94.2	19.0
6	1.00+J0.00	0.00	0.00	0.00	0.00
7	1.00+J0.00	0.00	0.00	22.8	10.9
8	1.00+J0.00	20.0	20.0	30.0	30.0
9	1.00+J0.00	0.00	0.00	0.00	0.00
10	1.00+J0.00	0.00	0.00	5.8	2.0

11	1.00+J0.00	15.0	13.0	0.00	0.00
12	1.00+J0.00	0.00	0.00	11.2	7.5
13	1.00+J0.00	15.0	20.0	0.00	0.00
14	1.00+J0.00	0.00	0.00	6.2	1.6
15	1.00+J0.00	0.00	0.00	8.2	2.5
16	1.00+J0.00	0.00	0.00	3.5	1.8
17	1.00+J0.00	0.00	0.00	9.0	5.8
18	1.00+J0.00	0.00	0.00	3.2	0.9
19	1.00+J0.00	0.00	0.00	9.5	3.4
20	1.00+J0.00	0.00	0.00	2.2	0.7
21	1.00+J0.00	0.00	0.00	17.5	11.2
22	1.00+J0.00	0.00	0.00	0.00	0.00
23	1.00+J0.00	0.00	0.00	3.2	1.6
24	1.00+J0.00	0.00	0.00	8.7	6.7
25	1.00+J0.00	0.00	0.00	0.00	0.00
26	1.00+J0.00	0.00	0.00	3.5	2.3
27	1.00+J0.00	0.00	0.00	0.00	0.00
28	1.00+J0.00	0.00	0.00	0.00	0.00
29	1.00+J0.00	0.00	0.00	2.4	0.9
30	1.00+J0.00	0.00	0.00	10.6	1.9

**Tableau 5.** Valeurs planifiées

N° du nœud	$P_G^{\min}$ [MW]	$P_G^{\max}$ [MW]	a [\$/MW <sup>2</sup> h]	b [\$/MW <sup>2</sup> h]	c [\$/MW <sup>2</sup> h]
1	50	200	0.00375	2.00	0.00
2	20	80	0.0175	1.75	0.00
5	15	50	0.0625	1.00	0.00
8	10	35	0.00834	3.25	0.00
11	10	30	0.025	3.00	0.00
13	12	40	0.025	3.00	0.00

**Tableau 6.** Coefficients des fonctions coût et les limites des puissances

La consommation totale :  $P_{ch} = 285.200$  MW

En utilisant la méthode de Gauss- Seidel, avec une précision de  $10^{-5}$ , les résultats de l'écoulement de puissance sont :

Les pertes actives totales :  $P_L = 8.9215$  MW

Les pertes réactives totales :  $Q_L = 3.0459$  Mvar

### 3) Les données du réseau IEEE-57 :

Soit le schéma d'un réseau IEEE 57 -nœuds dont 7 nœuds générateurs est donné par la figure suivante :

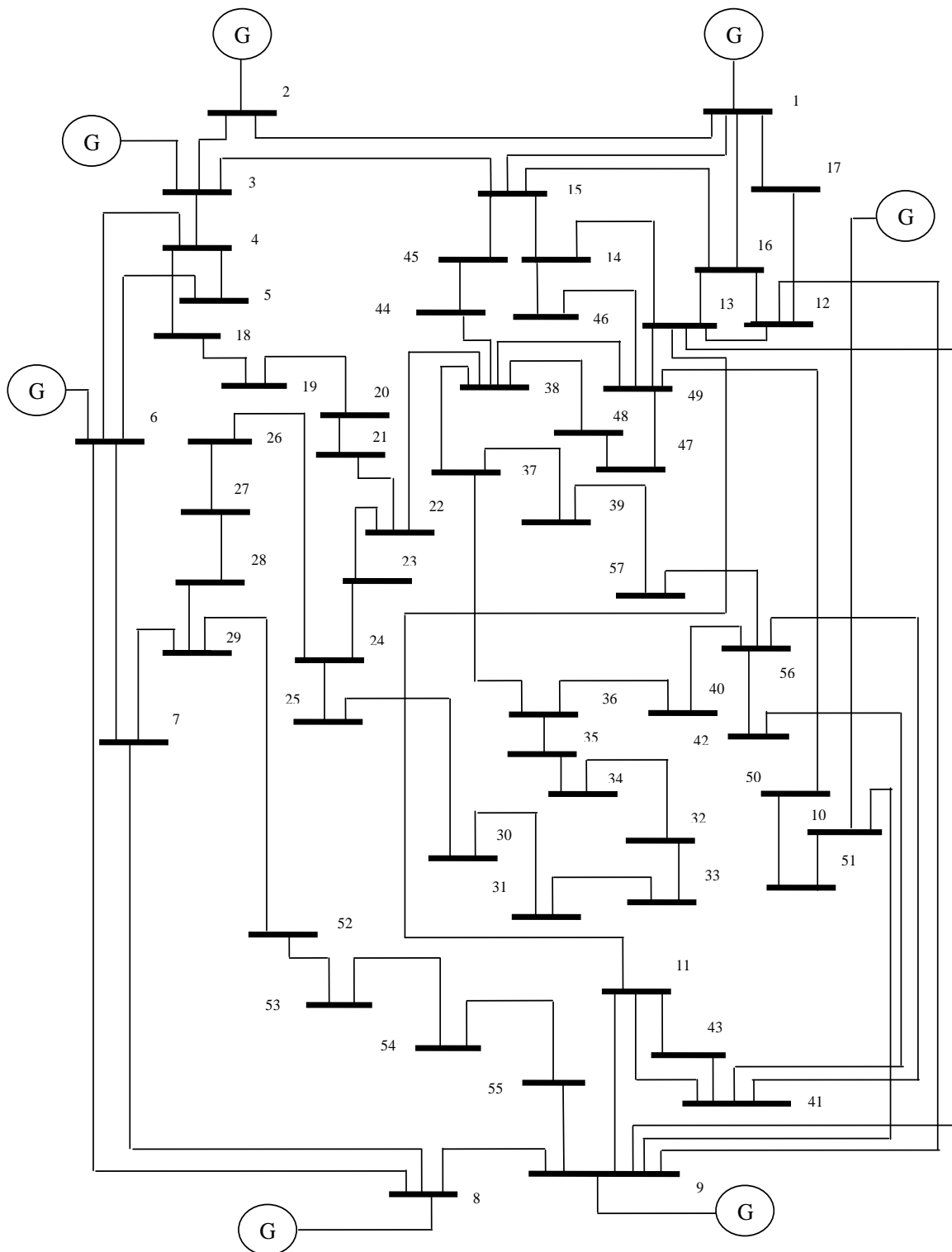


Figure3. Réseau IEEE 57-nœuds

pq	$Z_{pq}$ (PU)	$y_{pq}/2$ (PU)
1-2	0.0083+j0.028	0.129
2-3	0.0298+j0.085	0.0818
3-4	0.0112+j0.0366	0.038
4-5	0.0625+j0.132	0.0258
4-6	0.043+j0.148	0.0348
6-7	0.02+j0.102	0.0276
6-8	0.0339+j0.173	0.047
8-9	0.0099+j0.0505	0.0548
9-10	0.0369+j0.1679	0.044
9-11	0.0258+j0.0848	0.0218
9-12	0.0648+j0.295	0.0772
9-13	0.0481+j0.158	0.0406
13-14	0.0132+j0.0434	0.011
13-15	0.0269+j0.0869	0.023
1-15	0.0178+j0.091	0.0988
1-16	0.0454+j0.206	0.0546
1-17	0.0238+j0.108	0.0286
3-15	0.0162+j0.053	0.0544
4-18	0+j0.555	0
5-6	0.0302+j0.0641	0.0124
7-8	0.0139+j0.0712	0.0194
10-12	0.0277+j0.1262	0.0328
11-13	0.0223+j0.0732	0.0188
12-13	0.0178+j0.058	0.0604
12-16	0.018+j0.0813	0.0216
12-17	0.0397+j0.179	0.0476
14-15	0.0171+j0.0547	0.0148

18-19	$0.461+j0.685$	0
19-20	$0.283+j0.434$	0
21-20	$0+j0.7767$	0
21-22	$0.0736+j0.117$	0
22-23	$0.0099+j0.0152$	0
23-24	$0.166+j0.256$	0.0084
24-25	$0+j1.182$	0
24-26	$0+j0.0473$	0
26-27	$0.165+j0.254$	0
27-28	$0.0618+j0.0954$	0
28-29	$0.0418+j0.0587$	0
7-29	$0+j0.0648$	0
25-30	$0.135+j0.202$	0
30-31	$0.326+j0.497$	0
31-32	$0.507+j0.755$	0
32-33	$0.0392+j0.036$	0
34-32	$0+j0.953$	0
34-35	$0.052+j0.078$	0.0032
35-36	$0.043+j0.0537$	0.0016
36-37	$0.029+j0.0366$	0
37-38	$0.0651+j0.1009$	0.002
37-39	$0.0239+j0.0379$	0
36-40	$0.03+j0.0466$	0
22-38	$0.0192+j0.0295$	0
11-41	$0+j0.749$	0
41-42	$0.207+j0.352$	0
41-43	$0+j0.412$	0
38-44	$0.0289+j0.0585$	0.002
15-45	$0+j0.1042$	0

14-46	$0+j0.0735$	0
46-47	$0.023+j0.068$	0.0032
47-48	$0.0182+j0.0233$	0
48-49	$0.0834+j0.129$	0.0048
49-50	$0.0801+j0.128$	0
50-51	$0.1386+j0.22$	0
10-51	$0+j0.0712$	0
13-49	$0+j0.191$	0
29-52	$0.1442+j0.187$	0
52-53	$0.0762+j0.0984$	0
53-54	$0.1878+j0.232$	0
54-55	$0.1732+j0.2265$	0
11-43	$0+j0.153$	0
44-45	$0.0624+j0.1242$	0.004
40-56	$0+j1.195$	0
56-41	$0.553+j0.549$	0
56-42	$0.2125+j0.354$	0
39-57	$0+j1.355$	0
57-56	$0.174+j0.26$	0
38-49	$0.115+j0.177$	0.003
38-48	$0.0312+j0.0482$	0
9-55	$0+j0.1205$	0

**Tableau 7.** Les données des lignes

Les tensions planifiées ainsi que les puissances générées et consommées sont classées dans le tableau suivant :

Nœuds	Tensions (PU)		Puissances générées		Puissances consommées	
			$P_G$ (MW)	$Q_G$ (Mvars)	$P_{Ch}$ (MW)	$Q_{Ch}$ (Mvars)
1	1.04	0	-	-	55	17
2	1.01	-1.18	0.0	-0.8	3	88
3	0.985	-5.97	40.0	-1.0	41	21
4	0.981	-7.32	0.0	0.0	0	0
5	0.976	-8.52	0.0	0.0	13	4
6	0.98	-8.65	0.0	0.8	75	2
7	0.984	-7.58	0.0	0.0	0	0
8	1.005	-4.45	450.0	62.1	150	22
9	0.98	-9.56	0.0	2.2	121	26
10	0.986	-11.43	0.0	0.0	5	2
11	0.974	-10.17	0.0	0.0	0	0
12	1.015	-10.46	310.0	128.5	377	24
13	0.979	-9.79	0.0	0.0	18	2.3
14	0.97	-9.33	0.0	0.0	10.5	5.3
15	0.988	-7.18	0.0	0.0	22	5
16	1.013	-8.85	0.0	0.0	43	3
17	1.017	-5.39	0.0	0.0	42	8
18	1.001	-11.71	0.0	0.0	27.2	9.8
19	0.97	-13.2	0.0	0.0	3.3	0.6
20	0.964	-13.41	0.0	0.0	2.3	1
21	1.008	-12.89	0.0	0.0	0	0
22	1.01	-12.84	0.0	0.0	0	0
23	1.008	-12.91	0.0	0.0	6.3	2.1
24	0.999	-13.25	0.0	0.0	0	0



25	0.982	-18.13	0.0	0.0	6.3	3.2
26	0.959	-12.95	0.0	0.0	0	0
27	0.982	-11.48	0.0	0.0	9.3	0.5
28	0.997	-10.45	0.0	0.0	4.6	2.3
29	1.01	-9.75	0.0	0.0	17	2.6
30	0.962	-18.68	0.0	0.0	3.6	1.8
31	0.936	-19.34	0.0	0.0	5.8	2.9
32	0.949	-18.46	0.0	0.0	1.6	0.8
33	0.947	-18.5	0.0	0.0	3.8	1.9
34	0.959	-14.1	0.0	0.0	0	0
35	0.966	-13.86	0.0	0.0	6	3
36	0.976	-13.59	0.0	0.0	0	0
37	0.985	-13.41	0.0	0.0	0	0
38	1.013	-12.71	0.0	0.0	14	7
39	0.983	-13.46	0.0	0.0	0	0
40	0.973	-13.62	0.0	0.0	0	0
41	0.996	-14.05	0.0	0.0	6.3	3
42	0.966	-15.5	0.0	0.0	7.1	4.4
43	1.01	-11.33	0.0	0.0	2	1
44	1.017	-11.86	0.0	0.0	12	1.8
45	1.036	-9.25	0.0	0.0	0	0
46	1.05	-11.89	0.0	0.0	0	0
47	1.033	-12.49	0.0	0.0	29.7	11.6
48	1.027	-12.59	0.0	0.0	0	0
49	1.036	-12.92	0.0	0.0	18	8.5
50	1.023	-13.39	0.0	0.0	21	10.5
51	1.052	-12.52	0.0	0.0	18	5.3
52	0.98	-11.47	0.0	0.0	4.9	2.2
53	0.971	-12.23	0.0	0.0	20	10

54	0.996	-11.69	0.0	0.0	4.1	1.4
55	1.031	-10.78	0.0	0.0	6.8	3.4
56	0.968	-16.04	0.0	0.0	7.6	2.2
57	0.965	-16.56	0.0	0.0	6.7	2

**Tableau 8.** Valeurs planifiées

N° du nœud	$P_G^{\min}$ [MW]	$P_G^{\max}$ [MW]	a [\$/MW <sup>2</sup> h]	b [\$/MW <sup>2</sup> h]	c [\$/MW <sup>2</sup> h]
1	0.0	575.88	0.077	20.0	0.0
2	0.0	100.00	0.010	40.0	0.0
3	0.0	140.00	0.250	20.0	0.0
6	0.0	100.00	0.010	40.0	0.0
8	0.0	550.00	0.022	20.0	0.0
9	0.0	100.00	0.010	40.0	0.0
12	0.0	410.00	0.032	20.0	0.0

**Tableau 9.** Coefficients des fonctions coût et les limites des puissances

La consommation totale :  $P_{ch} = 1250.8$  MW

Les résultats de l'écoulement de puissance par la méthode de Gauss Seidel, pour une précision  $\varepsilon = 10^{-5}$  sont :

Les pertes actives totales :  $P_L = 27.882$  MW

Les pertes réactives totales :  $Q_L = -116.39$  Mvar

## الملخص:

هذه الرسالة تناولت استخدام الميتاهورستيك وتهجماتها. في المرحلة الأولى تمثلت في فهم و برمجة الميتاهورستيك الأساسية (الخوارزميات، الجينية، سرب الجسميات الأمثل، البحث المحظور، الانصهار الزائف).  
IEEE-5bus, هذه البرامج اختبرت على عدد من الدوال البسيطة ثم قمنا بتطبيقها على مسألة تدفق الاستطاعة الكهربائية في المرحلة الثانية قمنا بمزج الميتاهورستيك من خلال التهجين الأول على شكل أفواج IEEE-30 bus, IEEE57 bus و النهج الثاني المزج التكاملي  
AG/TS , AG/QN, TS/QN

**كلمات مفتاحية:** الميتاهورستيك، تهجين، الخوارزميات الجينية، سرب الجسميات، الانصهار الأمثل المحظور، نظام الطاقة الكهربائية.

## Résumé

Cette thèse présente l'utilisation des méthodes métaheuristiques dans la répartition optimale des puissances actives. La première phase consiste à comprendre, maîtriser puis programmer les métaheuristiques de base (Algorithmes Génétiques, Recherche Tabou). Les programmes conçus ont été d'abord testés et validés sur un certain nombre de fonctions puis appliqués au problème d'optimisation de puissance active dans un réseau électrique.

Nous avons appliqué pour notre problème d'optimisation un Algorithme Génétique avec deux types de codage un codage binaire et un codage réel sur trois réseaux standards IEEE-5, IEEE-30 et IEEE-57 nœuds.

Dans la deuxième partie, nous avons appliqué une autre méthode métaheuristique (Recherche Tabou). Une comparaison entre les résultats obtenus des méthodes métaheuristiques et une méthode analytique quasi-Newtonienne à été faite sur les différents réseaux traités.

La dernière partie présente une hybridation entre les métaheuristiques de base (AG-RT) et les métaheuristiques avec une méthode analytique quasi-Newtonienne (RT-QN, AG-QN).

**Mots-clés :** Métaheuristiques, Optimisation Combinatoire, Algorithmes Génétiques, Recherche Tabou, Hybridation, quasi-Newtonien.