

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR 16/03
ET DE LA RECHERCHE SCIENTIFIQUE

ÉCOLE NATIONALE POLYTECHNIQUE



المدرسة الوطنية المتعددة التقنيات
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

Département de Génie électrique

Mémoire du Projet de fin d'études pour l'obtention du diplôme
d'ingénieur d'état en électrotechnique

THÈME

**Application des Algorithmes Génétiques pour la
Répartition Economique de Puissance d'un
Réseau Electrique**

Proposé et dirigé par :

Mr HELLAL.A

Présenté par :

NEMOUCHI Fateh

ALIOUAT Mohammed

PROMOTION 2003

Remerciements

Nous tenons tout d'abord à remercier Mr A.hellal, qui nous a proposé un sujet intéressant et qui a consacré beaucoup de son temps à notre travail. Grâce a ses nombreuses remarques, toujours intéressantes, notre recherche a toujours été diriger dans la bonne direction.

Nous tenons aussi à remercier Mr O.Ben Ouered, enseignant à l'Université de Blida, qui nous a aidé énormément dans notre travail.

Nous tenons également à remercier tous les membres de l'équipe de service conduite système de SONELGAZ, pour leur accueil très chaleureux et leur disponibilité constante.

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

المدرسة الوطنية المتعددة التقنيات
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

أهداء

أهدي هذا العمل المتواضع إلى :

الوالدة الكريمة حفظها الله

الأخت العزيزة و أبناءها وزوجها

كل عائلتي

زميل الصعاب فاتح

كل الأصحاب : مراد, أحمد, كمال, محمد...

كل رفقاء الدراسة

كل من يحبني

كمال

*D*édicaces

A mes chers parents.

A mes frères

A ma sœur.

A mes tantes et oncles et leurs enfants.

A toute ma famille.

A mon binôme Mohammed et mes très chers amis :

A tous mes amis de l'ENP.

A la promotion Electrotechnique 2003.

A tous ouleds houmti (Mouloud, Hichem, Mustapha, Zouhir,

Nabil,...) .

A Ceux qui m'aime.

الموضوع تطبيق اللوغارتميات الجينية للتوزيع الاقتصادي لطاقة شبكة كهربائية.

ملخص هذا العمل يتمثل في تطبيق اللوغارتميات الجينية لحل مشكلة التوزيع الاقتصادي للطاقة المستهلكة على مجموع مولدات شبكة كهربائية بعد الدراسة المعمقة لمبدأ هذه اللوغارتميات قمنا بتطبيقها و قد اثبتت نتائج حسنة.

الكلمات المفتاحية: اللوغارتميات الجينية ، التوزيع الاقتصادي للطاقة ، شبكة كهربائية.

Sujet : l'application des algorithmes génétiques pour la répartition économique de puissance d'un réseau électrique .

Résumé : Ce travail consiste à l'application des algorithmes génétiques pour la résolution de problème de la répartition économique de puissance consommée sur l'ensemble des centrales d'un réseau électrique , après l'étude approfondie de principe de fonctionnement de ces algorithmes, nous avons implémenter un algorithme génétique et il a prouvé une bonne résultat.

Mots clés : algorithmes génétiques, répartition économique de puissance, réseau électrique.

Subject : the application of the genetic algorithms for the economic dispatch of power of an electrical network.

Summary : This work consists with the application of the genetic algorithms for the resolution of the economic dispatch problem of consumption power on the whole of the power stations of an electrical network, after the thorough study of principle of these algorithms, we have to implement a genetic algorithm and it proved a good result.

Key words : genetic algorithms, economic dispatch of power, electrical network.

SOMMAIRE

- *Introduction*

CHAPITRE I : Généralités sur les Méthodes d'Optimisation

I-1- Introduction

I-2- Fonction objective monovariante et multivariante

1-2-1- cas d'une fonction objective à une variable

1-2-2- cas d'une fonction objective multivariante

I-3- Les méthodes déterministes

1-3-1- Définition

1-3-2- La programmation linéaire

1-3-2-1- La méthode de simplexe

1-3-3- La programmation non linéaire.

1-3-3-1- La méthode de gradient

I-4- Les méthodes non déterministes

1-4-1- Méthodes stochastiques

1-4-1-1- recherche aléatoire (méthode de Monte-Carlo)

1-4-1-2- Amélioration itérative

1-4-2- Méthodes pseudo-aléatoires

1-4-2-1- Méthode de recuit simulé

1-4-2-2- Approches heuristiques d'optimisation

1-4-2-2-1- Le calcul évolutionnaire

1-4-2-2-2- Les bases biologiques

1-5- Discussion

1-6- Résumé

CHAPITRE II : Les Algorithmes Génétiques

II-1- Introduction

II-2- Organisation et Principe d'un Algorithme Génétique

II-3- Genèse de la population

II-4- Evaluation

II-5- Opérateur de sélection

II-5-1- Sélection par la roue biaisée

II-5-2- Sélection par tournoi

II-6- Codage binaire

II-7- Opérateur de croisement

II-7-1- Croisement en un point

II-7-2- Croisement en deux points

II-7-3- Croisement uniforme

II-8- Opérateur de mutation

II-9- Codage réel

II-10- Opérateur de croisement

II-10-1- Croisement simple

II-10-2- Croisement arithmétique

II-10-3- Croisement heuristique

II-10-4- Croisement MPX

II-11- Opérateur de mutation

II-11-1- Mutation uniforme

II-11-2- Mutation non uniforme

II-11-3- Mutation dans les bornes

II-12- Mécanisme de remplacement

II-12-1- Espace d'échantillonnage régulier

II-12-2- Espace d'échantillonnage élargi

II-13- Optimisation des fonctions simples par les Algorithmes Génétiques

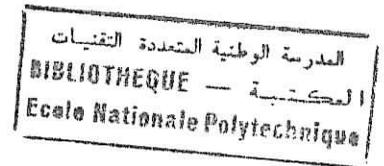
II-13-1- Fonction a une variable

II-13-1-1- Le codage

II-13-1-2- Tirage et évaluation de la population initiale

II-13-1-3- Sélection

II-13-1-4- Le croisement



Chapitre III : Approches Conventionnelles de La Répartition Economique de Charge

III-1- Introduction

III-2- Problème de la répartition économique de charge

III-3- Formulation générale

III-4- Résolution de la répartition économique de charge

III-4-1- Méthode des coûts marginaux

III-4-1-1- Méthode des coûts marginaux sous la seule contrainte
(production=consommation)

III-4-1-2- Méthode des coûts marginaux sous les contraintes imposées aux
groupes de production

III-4-1-3- Méthode des coûts marginaux avec considération des pertes

III-4-2- Méthode graphique

III-4-3- Méthode itérative de lambda variable

III-4-4- Méthode du gradient de premier ordre

III-4-5- Méthode du gradient de second ordre

III-4-6- Méthode de point bas et les facteurs de participation

Chapitre IV : Application des Algorithmes Génétiques Pour La Répartition Economique de Charge

IV-1 Introduction

IV-2 Algorithme génétique adopté

IV-2-1- Codage des variables du problème

IV-2-2- Espace de recherche

IV-2-3- Genèse de la population

IV-2-4- Opérateurs génétiques

IV-2-4-1- L'adaptation

IV-2-4-2- La sélection

IV-2-4-3- Croisement

IV-2-4-4- Mutation

IV-2-5- Constitution de la Nouvelle Génération

IV-3- Compromis exploration / exploitation

IV-4- Réglage des paramètres de l'AG

IV-5- Organigramme d'algorithme génétique

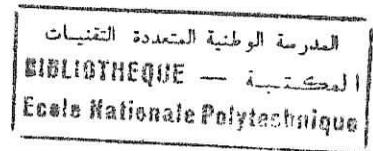
IV-6- Exemple d'application

IV-6-1- Les résultats de l'AG

IV-6-2- Les résultats obtenus avec une méthode de Programmation Non Linéaire

IV-6-3- Comparaison et discussion des résultats

IV-7- Conclusion



المدرسة الوطنية المتعددة التقنيات
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

Introduction

Pour ses besoins domestiques et industriels, le monde moderne a besoin de plus en plus d'énergie et une partie importante de celle-ci est produite et utilisée sous forme d'énergie électrique. L'accroissement continu de la consommation de cette forme d'énergie est d'ailleurs considérée comme une caractéristique de développement de la société.

Malheureusement, les gisements naturels d'électricité n'existent pas !. Alors, pour satisfaire la demande en énergie électrique, qui ne se trouve pas dans la nature sous une forme directement exploitable, il est nécessaire de la produire à partir d'une autre forme d'énergie que l'on appelle « énergie primaire », de la transporter puis de la distribuer.

La figure 1 schématise les trois aspects sous lesquels se présente la fourniture de l'énergie électrique : production, transport, distribution.

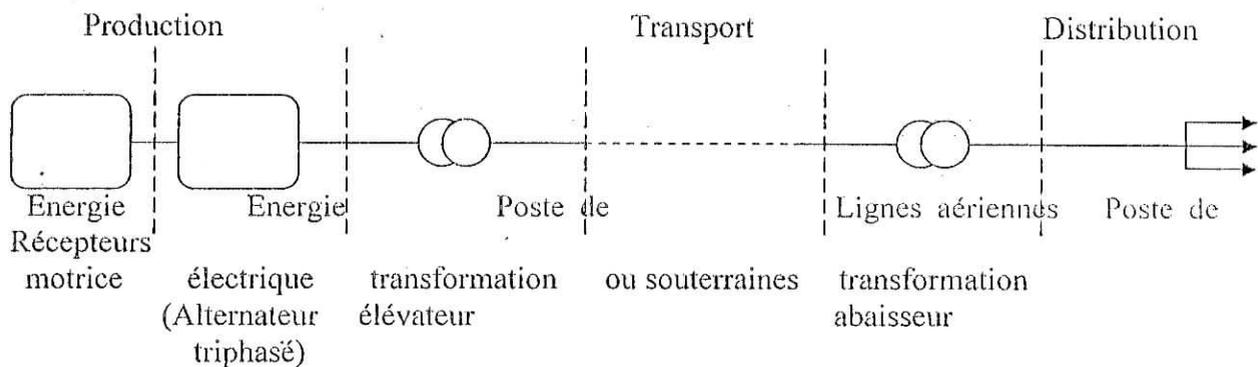


Figure 1 : Schéma simplifié de production, transport et de distribution de l'énergie électrique

La production englobe l'ensemble des centrales, généralement interconnectées. La tension délivrée au niveau des centrales est ensuite élevée grâce à un transformateur élévateur haute tension : cette technique de transport à une tension élevée permet en effet d'acheminer de grandes quantités d'énergie avec un gain économique considérable. Cette tension élevée est, bien sûr, abaissée par la suite grâce à un transformateur abaisseur afin de fournir l'énergie aux consommateurs.

Notons également que la production et le transport de l'électricité se font en triphasé.

En effet, le système triphasé permet aussi un gain économique très important et est adopté dans tous les pays du monde.

La production de l'énergie électrique est obtenue, selon les lois de l'électromagnétique, en faisant tourner un électro-aimant (produisant un champ magnétique) à l'intérieur d'un bobinage fixe où vont se créer des tensions induites, ce qui est le principe des alternateurs des centrales électriques.

Pour obtenir l'énergie électrique désirée, il faut donc absolument faire tourner les alternateurs dans les centrales. De plus, le réseau exige une tension électrique à fréquence fixe, réglée à 50 Hz. Il faut donc faire tourner ces alternateurs à vitesse synchrone devant absolument conduire à une tension à fréquence industrielle.

De nombreuses sources et techniques sont utilisées pour produire de l'énergie électrique, résumées à travers les types de centrales suivantes selon l'énergie primaire utilisée:

- **Les Centrales Thermiques** utilisant de l'énergie calorifique ou thermique produite à partir la combustion de fossiles, fuels ou autres matières premières se distinguent principalement par
 - *les centrales à vapeur* où le combustible (pétrole, charbon, lignite, mazout, gaz naturel ou artificiel,...) est brûlé dans des chaudières qui fournissent de la vapeur aux turbines.
 - *les centrales à gaz (turbine à gaz)*, qui prennent actuellement de l'extension vu les réserves importantes prouvées de la source primaire qui est le gaz naturel et de ses impacts moins nocifs pour l'environnement. principalement comme source d'énergie de point de vue réserve.
 - *les centrales nucléaires*, dont l'énergie primaire est transformée en chaleur (énergie thermique) par la combustion de produits fissiles (uranium, plutonium...).

- **Les centrales hydrauliques** utilisent l'énergie cinétique de l'eau, convenablement captée dans les chutes naturelles ou artificielles des cours d'eau.

- **Les centrales marémotrices** utilisent comme énergie primaire l'énergie potentielle de l'eau de la mer (lors de la marée haute) et celle d'un bassin naturel (à marée basse). Ces centrales, ont le même principe de fonctionnement que les centrales hydrauliques.

- **Les centrales éoliennes** utilisent le vent comme source d'énergie.

- **Les centrales solaires sont de deux types :**
 - *Les centrales photovoltaïques* qui consistent à transformer directement le rayonnement lumineux solaire, c'est-à-dire l'énergie des photons des rayons solaires en énergie électrique à travers des cellules photovoltaïques.

 - *Les centrales solaires héliothermiques* qui consistent à utiliser le principe des centrales thermiques à vapeur en exploitant l'énergie thermique d'un fluide colporteur chauffé par le rayonnement solaire à travers des parois vitrées (effet de serre).

- **Les centrales géothermiques qui exploitent** la chaleur interne de la terre, entretenue par la désintégration d'éléments radioactifs dans les couches profondes du globe. La géothermie est considérée avoir l'avantage d'être une source d'énergie renouvelable.

Notons que quelque soit l'énergie primaire utilisée, l'aspect économique jouera un rôle important dans le choix de telle ou telle technique. De nombreux facteurs comme le coût des installations ou le choix du site d'implantation sont loin d'être négligeables. De plus il faut également tenir en compte du rendement de la centrale afin de ne pas gaspiller le combustible utilisé.

Le coût de production d'un kilowattheure d'électricité est donc différent d'une centrale à l'autre.

La Consommation d'énergie électrique appelée sur un réseau est la somme des demandes ou charges des appareils branchés à chaque instant. Elle est donc variable en permanence. Cette variation apparaît généralement continue à l'échelle des réseaux de distribution et à fortiori à l'échelle de la production.

Afin d'avoir un réseau électrique robuste et d'assurer une bonne qualité d'énergie aux consommateurs, la production d'énergie électrique doit être capable de s'adapter à cette variation, ce qui implique un réglage et un contrôle centralisés du réseau [pél.8].

Ce réglage est pour faire face aux variations aléatoires ou non prévisibles des consommations en maintenant aussi près que possible de leur valeur de consigne :

- la fréquence du réseau
- la production des unités en service
- les échanges avec les réseaux voisins

La production des usines génératrices « en réglage » s'écarte donc nécessairement *des prévisions faites la veille*. Les écarts sont, pour la plupart, aléatoires, 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
- ou encore l'augmentation des pertes dans le réseau résultant de ces deux événements.

Il en résulte, donc, *des variations du coût total de production*.

Afin de contrôler ces variations du coût total de production de façon à le rendre à tout moment minimal, ou tout au moins aussi proche que possible du minimum, compte tenu des contraintes imposées au réseau et aux groupes de production, *un réglage économique des productions* est nécessaire.

Le Réglage Economique des Productions (*Répartition Economique de Charge*) :

Quand un réseau électrique doit répondre à une certaine courbe de charge, il adapte sa production de puissance en service à celle consommée. Une répartition judicieuse de la puissance absorbée par le réseau sur un certain nombre ou l'ensemble des centrales interconnectées, permet d'assurer à chaque instant la charge demandée, compte tenu des contraintes imposées au réseau et aux groupes de production.

En effet, pour une charge donnée, plusieurs répartitions sont réalisables du point de vue sécurité de réseau. Mais en introduisant l'aspect économique dans l'exploitation d'un réseau électrique, la question qui se pose est la suivante :

laquelle de toutes ces répartitions réalisables donne un coût total de production minimum, ou tout au moins aussi proche que possible du minimum.

On a finalement à résoudre un problème d'optimisation (de minimisation) avec contrainte.

Différentes méthodes sont utilisées pour cela, mais ne sont valables, chacune, que dans des conditions bien déterminées ou avec des hypothèses particulières.

Vu l'impact économique important de l'aspect de répartition de puissance dans un réseau électrique, surtout pour les grands réseaux, et la complexité de ce problème, des méthodes d'optimisation robustes et fiables sont souhaitées et deviennent même nécessaires.

A travers une recherche bibliographique assez riche, un certain nombre de travaux font apparaître de nouvelles techniques et méthodes appliquées dans le domaine que l'on peut résumer ainsi :

- Il a été prouvé théoriquement et expérimentalement que les **Algorithmes Génétiques** [Gol.6], [Ren.9] sont des procédures robustes d'exploration d'espaces complexes.
- Que ces algorithmes sont simples d'un point de vue calcul, mais sont cependant très performants dans leur recherche d'amélioration. De plus, ils ne sont pas fondamentalement limités par des hypothèses contraignantes sur le domaine d'exploration (hypothèses concernant la continuité " l'existence de dérivées " l'unicité modalité, entre autres).
- Et que les **Algorithmes Génétiques (AG)** sont maintenant appliqués en un nombre grandissant de domaines des affaires, de la recherche scientifique en général, ainsi que de l'ingénierie.

Apparemment, les techniques basées sur l'utilisation d'algorithmes génétiques possèdent les qualités de méthodes d'optimisation recherchées. Mais il est clair que pour tout scientifique, il faut comprendre :

- le comment et le pourquoi de l'efficacité de ces algorithmes
- ce qui les rend différents des méthodes d'exploration classiques
- les raisons du nombre croissant d'applications
- ainsi que l'étude de l'applicabilité de ces Algorithmes pour la résolution du problème de répartition économique de charge.

Pour parvenir à cette compréhension, notre travail a été effectué selon le plan de travail suivant :

Le Chapitre I, est consacré d'une part, à mentionner des généralités sur les méthodes d'optimisation [Ghi.5],[Gol.6] et [Chr.4], et d'autre part à situer les techniques utilisant les Algorithmes Génétiques parmi les différentes méthodes existantes.

Le Chapitre II, présente la définition et le principe de fonctionnement des algorithmes génétiques, leurs origines, et dans quelle mesure ils s'approchent ou diffèrent des autres procédures d'exploration. Des exemples simples illustrent le principe de fonctionnement à travers un suivi pas à pas de leur résolution [Gol.6], [Hes.7] et [Ren.9].

Le Chapitre III, présente la formulation générale du problème de répartition économique de charge, ainsi qu'un certain nombre de méthodes conventionnelles utilisées [Woo.10] et [Pél.8].

Le Chapitre IV, consiste à l'application de ces algorithmes génétiques au problème de répartition économique de charge.

Nous terminerons finalement par une **Conclusion**.

Généralité sur les Méthodes d'Optimisation

I.1 Introduction

Dans la vie courante, nous sommes fréquemment confrontés à des problèmes d'optimisation plus ou moins complexes. Ces problèmes peuvent être exprimés sous la forme générale d'un "problème d'optimisation". On définit alors une fonction objective (fonction de coût ou fonction profit), que l'on cherche à optimiser (minimiser ou maximiser) par rapport à tous les "paramètres" concernés.

Pour la résolution de ces problèmes, les méthodes utilisées peuvent être classées en méthodes déterministes, et non déterministes. Ces méthodes sont des méthodes de recherche de l'optimum local ou global.

- $f(x^*)$ est un minimum local :

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

- $f(x^*)$ est un minimum global:

$$\Leftrightarrow [\forall x \in X : f(x) \geq f(x^*)] \text{ et } x^* \in X.$$

I.2 Fonction objective monovariante et multivariante

I.2.1 Cas d'une fonction objective à une variable

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

Un exemple d'une telle fonction à une dimension est présenté dans la Figure 1.1. Il s'agit d'une fonction à une variable réel x , pouvant varier entre 0 et 1, dont on cherche le minimum.

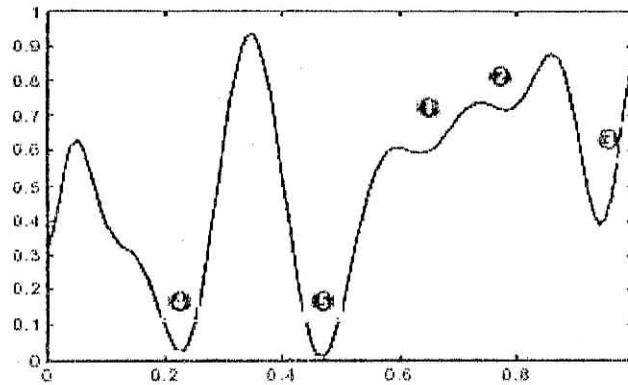


Figure 1.1 : fonction objective a une variable

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.

I.2.2 Cas d'une fonction objective multivariable

La fonction suivante $f(x, y)$ présente une fonction objective a deux variables x et y caractérisée par deux extremums : un local et l'autre global.

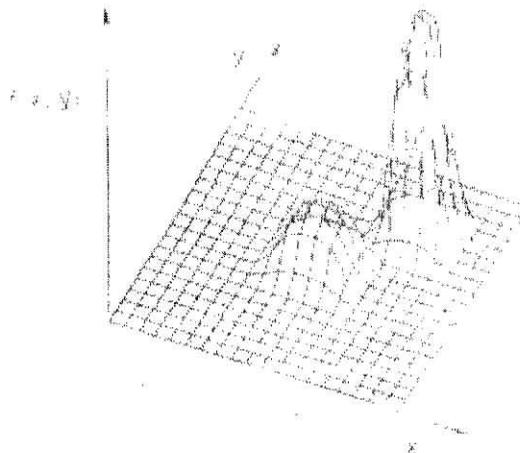


Figure 1.2 : fonction objective à deux variables

I.3 Les méthodes déterministes

I.3.1 Définition

Les méthodes déterministes se caractérisent par une exploration systématique de l'espace de recherche [Aub.1], Elle obéissent généralement à l'algorithme suivante :

1. Choix d'une première solution courante i admissible.
2. Génération d'une solution j dans le voisinage de i .
3. Si $f(j)$ est meilleur que $f(i)$ alors j devient la solution courante puis retour à 2.
4. l'algorithme se termine lorsqu'il n'y a plus d'amélioration de la solution courante.

Les méthodes déterministes peuvent être classées en deux classes : la programmation linéaire et non linéaire.

I.3.2 La programmation linéaire

La programmation linéaire a pour objet l'étude d'une catégorie particulière des problèmes : celle où la fonction à optimiser est une fonction linéaire, les liaisons entre les variables étant également linéaires par rapport aux variables [Ach.2].

I.3.2.1 La méthode du simplexe

L'intérêt principal de la méthode du simplexe qu'elle ne nécessite pas de calcul de gradient. Elle est uniquement basée sur l'évaluation de la fonction. Cela la rend utilisable des fonctions bruitées.

Cette méthode locale effectue une recherche multidirectionnelle dans l'espace d'état.

Soit une fonction f à minimiser, on appelle simplexe de R^n tout ensemble (x_0, x_1, \dots, x_n) de points de R tel que $f(x_0) \geq f(x_i) \forall i \in [1..n]$, x_0 est donc le meilleur élément.

Après la construction d'un simplexe initial, l'algorithme va modifier celui-ci en appliquant des calculs de réflexions, expansions et contractions jusqu'à la validation d'un critère d'arrêt.

I.3.3 La programmation non linéaire

La programmation non linéaire a pour objet l'étude des problèmes où la fonction à optimiser est une fonction non linéaire avec des contraintes linéaire ou non linéaire. Une multitude de méthodes ont été développées, les unes plus complexes que les autres, cependant pour les méthodes de base nous notons la méthode de gradient [Baz.3].

I.3.3.1 La méthode de gradient

Historiquement, les méthodes de gradient sont les plus anciennes. Elles permettent de résoudre des problèmes non linéaires et sont basées sur une l'hypothèse forte : la connaissance de la dérivée de la fonction objectif en chacun des points de l'espace. Cette famille de méthodes procède de la façon suivante :

On choisit un point de départ x_0 et on calcule le gradient $\nabla f(x_0)$ en x_0 , on se déplace d'une quantité λ_0 dans le sens opposé au gradient et on définit le point x_1 .

$$x_1 = x_0 - \lambda \frac{\nabla f(x_0)}{\|\nabla f(x_0)\|}$$

Cette procédure est répétée et engendre les points x_0, x_1, \dots, x_k . Ainsi, pas a pas, la distance entre le point d'indice k et l'optimum diminue.

$$x_{k+1} = x_k - \lambda_k \frac{\nabla f(x_k)}{\|\nabla f(x_k)\|} \quad \forall k, \lambda_k > 0$$

λ_k est le pas de déplacement a chaque itération.

Si λ_k est fixé, on parle de méthode de gradient a pas prédéterminé.

Le désavantage de cette procédure est que la convergence est très dépendante de choix du pas de déplacement. La convergence peut être très lente si le pas est mal choisi.

L'inconvénient majeur de cette méthode est leur hypothèse de dérivabilité qui la rend inapte à traiter beaucoup de problème réel. De plus, leur déplacement déterministe dans l'espace d'état engendre de temps de calcul très importants dans certaines conditions :

- espace d'état très grand.
- Mauvais paramétrage de la méthode (maillage, vecteur du pas de déplacement).
- Piège de certaines fonctions.

I.4 Méthodes non déterministes

I.4.1 Méthodes stochastiques

Les méthodes stochastiques sont les plus récentes et font intervenir des processus aléatoires dans la recherche des solutions optimales. Le développement de l'outil informatique depuis un demi siècle a permis d'utiliser ces méthodes à une grande échelle [GoL.6].

Les algorithmes de recherche aléatoires ont connu un succès grandissant depuis que les chercheurs ont observé les limitations des méthodes déterministes.

Cependant, les procédures aléatoires qui explorent et mémorisent le meilleur élément doivent elles aussi être abandonnées pour des raisons de manque d'efficacité.

Dans notre précipitation à éliminer les méthodes d'exploration purement aléatoires, nous devons surtout faire attention à les distinguer des méthodes pseudo- aléatoires.

Parmi les méthodes aléatoires en peut cité :

I.4.1.1 Recherche aléatoire (méthode de Monte-Carlo)

Cette méthode consiste à tirer à chaque itération une solution au hasard. La fonction objective f est évaluée en ce point. La nouvelle valeur est comparée à la précédente. Si elle est meilleure que la précédente, cette valeur est enregistrée, ainsi que la solution correspondante, et le processus continue. Sinon on repart du point précédent et on recommence le procédé,

jusqu'à ce que les conditions d'arrêt soient atteintes [Ghi.5]. L'algorithme est présenté à la Figure 1.3.

```

x := solution aléatoire
fmin := f(x)
xmin := x

REPETER
x := solution aléatoire
SI f(x) < fmin
  fmin := f(x)
  xmin := x
FIN SI
JUSQU'A conditions d'arrêt satisfaites

```

Figure 1.3 : pseudo code d'un algorithme de recherche aléatoire.

I.4.1.2 Amélioration itérative (méthode de la descente)

Cette méthode vise à déterminer une solution $s(x)$ dans le voisinage de la solution courante x , telle que $f(s(x)) < f_{min}$ (f_{min} désigne la valeur minimale courante de f). La méthode consiste à engendrer, à chaque itération, un N -échantillon, suivant un procédé aléatoire dans le voisinage de la solution courante x . La fonction objectif f est évaluée en chaque point de l'échantillon, et la solution x' correspond à la plus petite valeur de f obtenue [Ghi.5].

$$f(x') = f(s(x)) = \min [f(s_i(x))]. \quad 1 < i < N$$

Cette nouvelle valeur $f(x')$ est comparée à la valeur minimale courante f_{min} . Si elle est meilleure, cette valeur est enregistrée, ainsi que la solution correspondante, et le processus continue. Sinon l'algorithme prend fin, un minimum local est atteint. Cet algorithme est présenté dans la Figure 1.4.

```

x := solution aléatoire
f_min := f(x)
x_min := x

REPETER
engendrer un N-échantillon s_i(x) ∈ voisinage S(x) et
calculer f(s(x)) = min_{1 ≤ i ≤ N} [f(s_i(x))]

SI f(s(x)) < f_min
  f_min := f(s(x))
  x_min := s(x)
SINGN
  sortir de REPETER
FIN SI
FIN REPETER

```

Figure I.4 : pseudo code de la méthode d'amélioration itérative

Après avoir atteint un minimum local, cette procédure peut repartir d'un autre point pris au hasard.

I.4.2 Méthodes pseudo-aléatoires

Les méthodes pseudo-aléatoires sont des méthodes qui utilisent un choix aléatoire comme outil pour guider une exploration hautement intelligente dans l'espace des paramètres codes.

I.4.2.1 Méthode du recuit simulé

Le recuit simulé 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 des 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 [Aub.1].

L'analogie exploitée par le recuit simulé consiste à considérer une fonction f à minimiser comme fonction d'énergie, et une solution x peut être considérée comme un état donné de la matière dont $f(x)$ est l'énergie. Le recuit simulé exploite généralement le critère défini par l'algorithme de Metropolis pour l'acceptation d'une solution obtenue par perturbation de la solution courante.

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

L'algorithme est présenté dans la Figure 1.5.

```

SI  $f(s(x)) \leq f(x)$ 
     $f(x) := f(s(x))$ 
     $x := s(x)$ 
SINON
     $p = \exp \frac{-\Delta f}{T}$ 
     $r :=$  solution aléatoire entre  $[0, 1]$ 
    SI  $r \leq p$ 
         $f(x) := f(s(x))$ 
         $x := s(x)$ 
    FIN SI

```

Figure 1.5 : pseudo code de la règle de Metropolis.

Le paramètre de contrôle 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. Nous présentons dans la Figure 1.6. L'algorithme du recuit simulé.

```

x := solution aléatoire
f(x) := valeur de la fonction

fmin := f(x)
xmin := x

T := initialiser température (assez élevée)
REPETER
  REPETER
    générer un voisin s(x) ∈ voisinage S(x)
    appliquer la règle de Metropolis
    SI f(x) < fmin
      fmin := f(x)
      xmin := x
    FIN DE SI
  JUSQU'A équilibre thermodynamique atteint

  T := décroître température
JUSQU'A conditions d'arrêt satisfaites

```

Figure 1.6 : pseudo code de l'algorithme de recuit simulé

1.4.2.2 Approches heuristiques d'Optimisation

1.4.2.2.1 Le Calcul Evolutionnaire

Le terme *calcul évolutionnaire* se réfère à l'étude des fondements et applications de certaines techniques heuristiques basées sur les principes de l'évolution naturelle. En dépit du fait que ces techniques peuvent être classées en trois principales catégories (Figure 1.7), cette classification est basée sur quelques détails et faits de développement historique plutôt que sur des différences majeures de fonctionnement. En fait, leur base biologique est essentiellement la même.

CE	=	AG	+	SE	+	PE	+	PG
<i>Calcul</i>		<i>Algorithmes</i>		<i>Stratégies</i>		<i>Programmation</i>		<i>Programmation</i>
<i>Evolutionnaire</i>		<i>Génétiques</i>		<i>Evolutionnaires</i>		<i>Evolutionnaire</i>		<i>Génétique</i>
		(Holland, 1975)		(Rechenberg, 1973)		(Fogel, Owens, Walsh, 1966)		(Koza, 1992)

Figure 1.7 : L'équation évolutionnaire.

1.4.2.2 Les bases biologiques

L'origine des algorithmes évolutionnaires a été une tentative d'imitation de certains processus de l'évolution naturelle; Bien que beaucoup de détails de l'évolution biologique ne soient pas complètement compris (et ne le seront pas avant longtemps), il existe des points bien confirmés par l'évidence expérimentale.

- L'évolution est un processus opérant par le biais de chromosomes plutôt que par le biais d'organismes. Les chromosomes sont des outils organiques codant la structure d'un être vivant, i.e. une créature est "construite" en décodant un ensemble de chromosomes.
- La sélection naturelle est un mécanisme reliant des chromosomes avec la performance de l'entité qu'ils représentent, permettant une bonne adaptation aux organismes performants bien adaptés à leur environnement, de se reproduire plus souvent que ceux le sont moins.
- Le processus évolutionnaire prend place durant l'étape de reproduction. Il existe un grand nombre de mécanismes de reproduction dans la nature. Les plus usuels sont les mécanismes de mutation et de recombinaison

Basés sur ces points on peut ainsi se pencher sur les algorithmes génétiques qui font l'objet de notre étude.

• 1.4.2.3 Les algorithmes génétiques

Les principes fondamentaux de ces algorithmes ont été exposés par Holland. Ces algorithmes s'inspirent du fonctionnement de l'évolution naturelle, notamment la sélection de Darwin, et la procréation selon les règles de Mendel [Ren.9] [GoL.6].

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"), 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 fait que celui-ci est plus au moins bien adapté à l'environnement.

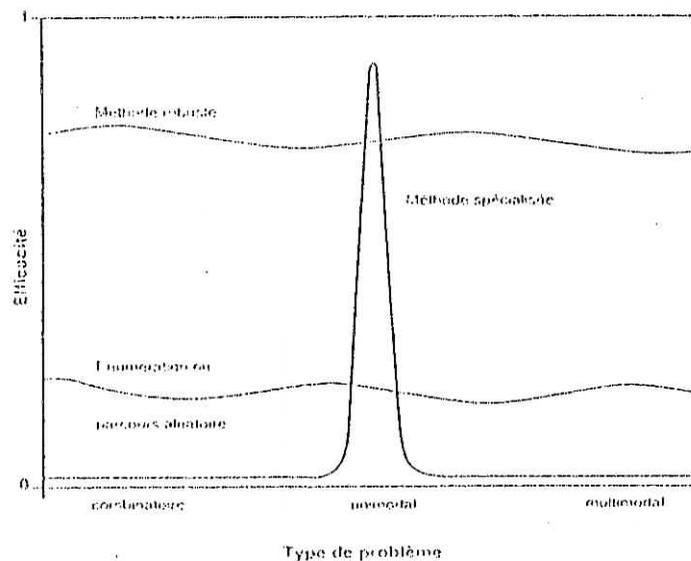
S'il est bien adapté, il a une plus grande chance de procréer dans la génération future. Au fur et à mesure des générations, on sélectionne les individus les mieux adaptés, et l'augmentation du nombre des individus bien adaptés fait évoluer la population entière.

I.5 Discussion

Bien que notre discussion n'est pas fait un examen exhaustif de la pléthore de méthodes d'optimisation traditionnelle, elle nous laisse une désagréable conclusion : les méthodes d'exploration conventionnelle ne sont pas robustes.

Ceci n'implique pas qu'elles sont inutiles. les procédures mentionnées plus haut, ainsi que d'innombrable combinaison ou permutation hybrides ont été appliquées nombreuses fois avec succès, cependant, quand on s'attaque à des problème plus complexes, d'autre sont nécessaires.

Pour éclairer ce point, observons le spectre des problèmes dans la Figure I.8. Cette figure représente un indice d'efficacité idéalisé en fonction d'un continuum de problèmes.



1.6 Résumé

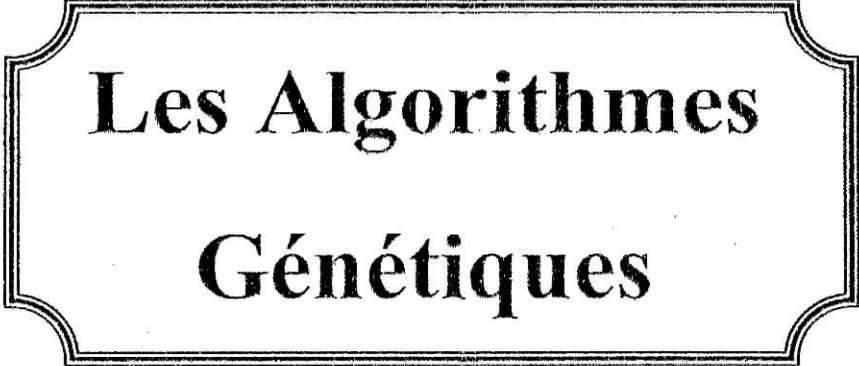
Dans ce chapitre, on a présenté différentes méthodes d'optimisation et leur propriété :

Les méthodes stochastiques permettent une bonne exploration de l'espace de recherche, mais il n'y a pas d'exploitation des résultats déjà obtenus.

Par contre dans Les méthodes déterministes vues ci-dessus sont très efficaces sur des problèmes particuliers et en général de petite taille. Mais sur des problèmes de grande taille, la probabilité de trouver l'optimum global en un temps raisonnable dépend essentiellement de la bonne méthode et les bons paramètres.

Afin de remédier à ces problèmes, des méthodes pseudo-aléatoires ont été élaborées, dont les méthodes basées sur les algorithmes génétiques détaillées dans le chapitre suivant.

CHAPITRE II



Les Algorithmes Génétiques

II.1. Introduction

Les algorithmes génétiques sont inspirés d'observations concernant les mécanismes de la sélection naturelle. Ils ont été conçus à l'origine pour résoudre des problèmes traitant d'adaptation, ce qui a amené les chercheurs à tenter de reproduire certains phénomènes naturels, avec des caractéristiques leur conférant des aptitudes et des règles de comportement efficaces leur autorisant à prendre en compte des conditions très difficiles [Hcs.7] :

- Situation imprévue.
- Environnement inconnu.
- Contraintes changeantes induites par l'environnement.

Les premières réflexions niant l'existence d'une « conception originelle de systèmes naturels figés », déjà adaptés pour toujours à toutes les conditions extérieures, ont été introduites par DARWIN, vers 1860, avec sa théorie de l'évolution des espèces.

L'idée de base de cette théorie est que, sous l'influence des contraintes extérieures, les êtres se sont graduellement auto-modifiés, et ce au travers du processus de reproduction, pour produire des générations d'êtres mieux adaptés à leur milieu naturel. Cette idée porte une notion d'auto-organisation, permettant aux espèces correspond aux observations : les premiers êtres vivants dont nous possédons des traces tangibles (fossiles) sont des systèmes extrêmement simples par rapport aux êtres évolués vivant à notre époque.

Une approche de modélisation plus poussée de cette évolution à travers le processus de reproduction a été formalisée par J.MONOD. Cette approche purement biologique insiste sur la diversité originelle qui s'appuie sur la combinatoire d'un petit nombre de caractéristiques de base. Le filtrage du hasard opéré par les conditions extérieures laisse vivre ou fait mourir alors un être, avant ou après qu'il se soit reproduit.

Les problèmes de traitement de l'information rencontrés dans l'industrie sont aujourd'hui traités par des systèmes figés, ayant fait l'objet d'une phase de conception originelle (appelée ingénierie), cette conception étant destinée à leur conférer des caractéristiques adaptées pour toujours à des conditions d'exploitation connues. Ce mode de

production de systèmes de traitement de l'information est bien sûr extrêmement limitatif, puisqu'il suppose que l'on connaisse parfaitement, lors de la conception, toutes les configurations d'exploitation possibles de système, sachant que celles-ci ne sont pas censées évoluer de manière significative (pas d'usure de machine, pas de modifications industriel).

Les évolutions éventuelles du système sont alors classiquement prises en charge périodiquement par de nouvelles opérations d'ingénierie. Or cette limitation intrinsèque interdit la mise au point de systèmes réellement évolutifs (capable de s'adapter à des conditions d'environnement inconnues ou variables) ou adaptatifs (tenant compte des changements des sources d'éclairage et des capteurs dans un contexte industriel).

Des chercheurs en informatique ont donc étudié la possibilité de donner à des machines l'aptitude d'évoluer spontanément, de manière à s'adapter automatiquement à de nouvelles conditions d'exploitation (usures, changement d'utilisateurs).

Les premiers travaux significatifs sur thème ont été proposés par HOLLAND en 1962. Il a développé la notion de parallélisme implicite (théorie des schémas) véhiculée par ces algorithmes en 1968, et a proposé la formalisation actuelle des algorithmes génétiques dans les années 70.

Lors de la conception de ces algorithmes génétiques, les chercheurs remarquèrent que ceux-ci présentaient des aptitudes remarquables à l'optimisation. Leur comportement semblait même plus robuste que celui de certains algorithmes classiquement dédiés à cette discipline. Cette propriété a été formalisée depuis : les algorithmes génétiques ont fait l'objet de travaux de recherche et ont été utilisés dans nombre d'applications qui démontrent leur adaptation à certains problèmes difficiles de recherche opérationnelle.

La puissance des algorithmes génétiques vient du fait que la technique est robuste et peut s'appliquer à une grande catégorie de problèmes. Incluant ceux qui sont difficiles à résoudre avec les autres méthodes, les algorithmes génétiques ne garantissent pas de trouver la solution optimale globale d'un problème. Mais ils sont généralement capables de trouver des solutions acceptables dans des délais acceptables.

On distingue 4 principaux points qui font la différence fondamentale entre ces algorithmes et les autres méthodes :

- Les algorithmes génétiques utilisent un codage des paramètres, et non les paramètres eux mêmes.
- Les algorithmes génétiques travaillent sur une population de points, au lieu d'un point unique.
- Les algorithmes génétiques n'utilisent que les valeurs de la fonction étudiée, pas sa dérivée, ou une autre connaissance auxiliaire.
- Les algorithmes génétiques utilisent des règles de transition probabilistes, et non déterministes.

II.2 Organisation et Principe d'un algorithme génétique

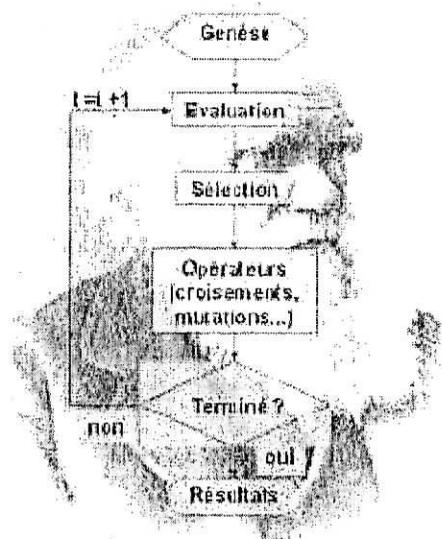


Figure 2.1 : Organisation d'un algorithme génétique

Les algorithmes génétiques (AG) sont des algorithmes itératifs d'optimisation pseudo-aléatoires fondés sur les mécanismes de la sélection naturelle et de la génétique. Leur fonctionnement est extrêmement simple, il manipule une *population* de taille constante. Cette

population est formée de points candidat appelé *individu*. La taille constante de la population entraîne un phénomène de compétition entre les individus. Chaque individu représente le codage d'une solution potentielle au problème à résoudre, il est constitué d'un ensemble d'éléments appelés *chromosome*, qui a leur tour constitué des *gènes* pouvant prendre plusieurs valeurs appartenant à un *alphabet* non forcément numérique.

A chaque itération, appelée *génération*, est créée une nouvelle population avec le même nombre des individus. Cette génération consiste en des individus mieux "*adaptés*" à leur environnement tel qu'il est représenté par la fonction sélective. Au fur et à mesure des générations, les individus vont tendre vers l'optimum de la fonction sélective. La création d'une nouvelle population à partir de la précédente se fait par application des opérateurs génétiques que sont: le croisement et la mutation.

Les opérateurs génétiques imitent le processus d'hérédité des gènes pour les créations des nouveaux descendants à chaque génération. L'opération d'évaluation imite le processus d'évolution Darwinienne pour créer les populations de la génération.

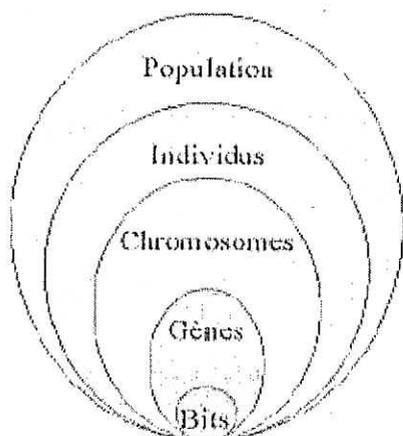


Figure 2.2 : les cinq niveaux d'organisation de l'algorithme génétique.

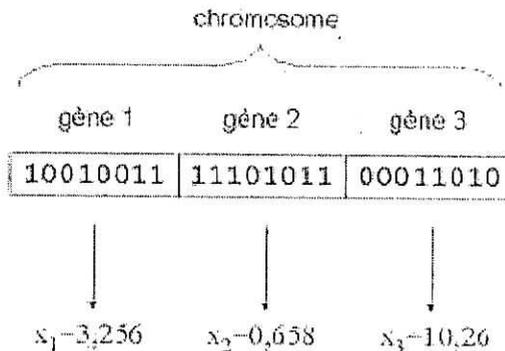


Figure 2.3 : Schéma de codage des variables d'optimisation x_i

II.3 Genèse de la population

La première étape de l'algorithme est la genèse de la population, c'est-à-dire le choix des dispositifs de départ qu'il faut faire évoluer.

On pourrait prendre des individus régulièrement répartis dans l'espace. Néanmoins, une initialisation aléatoire est plus simple à réaliser : les valeurs des gènes sont tirées au hasard selon une distribution uniforme.

II.4 Evaluation

L'évaluation de la nouvelle génération est réalisée par le modèle utilisé. Pour cela, la fonction d'adaptation intervient pour déterminer les meilleurs individus à utiliser dans la prochaine génération : ce sont les valeurs sélectives des solutions de chaque individu qui diront quels sont les meilleurs individus à garder.

La fonction sélective doit rendre des résultats positifs ou nuls pour que l'opérateur de sélection fonctionne convenablement, c'est là la seule contrainte imposée sur elle. La fonction sélective f est formée par la composition de deux fonctions :

$$f(x) = g(o(x))$$

La fonction g peut être une transformation linéaire de $o(x)$

$$f(x) = a.o(x) + b$$

Il peut être intéressant de faire une transformation logarithmique de la fonction objective.

Par exemple pour minimiser une fonction, on a

$$f(x) = b - \log o(x)$$

Cette fonction sélective permet de limiter les écarts de valeur sélective dans les premières générations (limitant ainsi le risque de convergence prématurée). De plus, elle permet d'amplifier les écarts de valeurs sélectives entre les individus pour lesquelles $o(x)$ est faible (que l'on retrouve en grand nombre lorsque l'algorithme commence à converger).

Une autre possibilité, lorsque l'on cherche à maximiser $o(x)$, est d'utiliser une transformation par élévation à une puissance donnée :

$$f(x) = [a.o(x) + b]^\gamma$$

La valeur de γ dépend du problème à traiter.

Il peut être intéressant de faire varier le paramètre γ durant l'exécution de l'algorithme génétique de manière à adapter les écarts des valeurs sélectives comme désiré.

Un algorithme génétique évoluant vers une population d'individus à fortes valeurs sélectives, peut ajuster dynamiquement la fonction sélective f de manière à ce que l'écart entre l'individu de plus forte valeur sélective et celui de plus faible valeur sélective ne se réduise pas trop. On a alors

$$f(x) = a.o(x) + b(t)$$

Une définition possible de $b(t)$ dans le cas où on maximise $o(x)$, est

$$b(t) = -\min\{f(x) / x \in p(t)\}$$

Où $p(t)$ représente la population à l'instant t . On parle dans ce cas de fonction sélective ajustée linéairement (dynamic linear scaled fitness function).

Dans le cas où le problème à résoudre est contraint, les solutions qui ne respectent pas la contrainte doivent être écartées. Plusieurs solutions existent, pour lesquelles:

1. On répète le processus de recombinaison jusqu'à ce qu'un individu respectant les contraintes soit généré.
2. On adapte les opérateurs génétiques de manière à ce que seuls soit générés des individus respectant les contraintes.
3. On pénalise les individus qui ne respectent pas les contraintes en diminuant leur valeur sélective. Pour cela, on calcule une fonction de pénalité (penalty fonction) p que l'on soustrait à la valeur sélective brute de l'individu.

$$f(x) = g(o(x)) - p(x)$$

Pour les problèmes dans lesquels de nombreuses solutions ne sont pas possibles car ne respectant pas la contrainte, il est préférable de retenir la dernière solution.

Notons que dans le cas d'un modèle physique, la majeure partie du temps de calcul sera probablement due à l'exécution de ce modèle. En effet, le reste de l'AG est essentiellement composé de manipulation d'entiers et de bits, donc très rapide.

II.5 Opérateur de Sélection

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

Les individus ayant une grande fitness ont donc plus de chance d'être sélectionnés. On parle alors de sélection proportionnelle.

La sélection fournie la force motrice dans l'algorithmes génétiques, elle est basé sur le principe de la sélection darwinienne et la pression de sélection est critique .a l'une des extrémités, la recherche de la solution optimale va se terminer prématurément. Tandis qu a l'autre extrémité, la progression est plus lente que nécessaire. Typiquement, une faible pression est indiquée, au lancement de la recherche, en faveur d'une large exploration de

l'espace des solutions, tandis que une forte pression de sélection est recommandée, a la fin, en vue d'exploiter les régions les plus prometteuses de l'espace de solutions.

Il existe plusieurs méthodes pour la reproduction. La méthode la plus connue et utilisée est sans doute, la roue de loterie biaisée (*roulette wheel*) de Goldberg et encore la sélection par tournoi.

II .5.1 Sélection par la roue biaisée

Selon cette méthode, chaque individu sera dupliqué dans une nouvelle population proportionnellement à sa valeur d'adaptation. On effectue, en quelque sorte, autant de tirages avec sélection d'un élément dans la population. Ainsi, dans le cas d'un codage binaire, la fitness d'un chromosome particulier étant $f(x)$, la probabilité avec laquelle il sera réintroduit dans la nouvelle population de taille N est :

$$\frac{f(x)}{\sum_{i=1}^N f(x_i)}$$

L'inconvénient majeur de cette méthode repose sur le fait qu'un individu n'étant pas le meilleur peut tout de même dominer la sélection. Elle peut aussi engendrer une perte de diversité par la domination d'un super individu. Un autre inconvénient est sa faible performance vers la fin quand l'ensemble des individus se ressemble. Darwin résume très bien tous ces inconvénients :

Pensez à une situation où une chaîne [individu pour nous] de la population a comparativement une fitness élevée mais n'est pas optimal ou proche de l'optimum. Disons que la fitness de cette chaîne est dix fois plus grande que la fitness moyenne. Il pourrait facilement arriver, après quelques générations, que la population ne soit entièrement constituée que de cette chaîne. Dans un tel cas, l'algorithme génétique n'évoluera plus et l'optimum ne sera pas trouvé. Ce phénomène est appelé "convergence prématurée" et est l'un des problèmes les plus fréquents lors de l'utilisation des algorithmes génétiques.

II .5.2 Sélection par tournoi

On tire un ensemble des individus aléatoirement dans la population et on reproduit le meilleur des ces individus a la nouvelle population. On refait cette procédure jusqu'à ce que la nouvelle population soit complète. Il est tout a fait possible que certains individus participent à plusieurs tournoi, s'il gagnent plusieurs fois dans la population intermédiaire, ce qui favorisera la pérennité de leurs gènes.

Généralement Cette méthode donne de bons résultats.

On note que l'opération de sélection ne crée pas de nouveaux individus dans la population. Ceci est le rôle des opérateurs de croisement et de mutation.

II.6 Codage binaire

Il existe deux principaux type de codage : binaire ou réel. Rappelons que la transformation la plus simple (fonction de décodage d) d'une chaîne binaire A en nombre entier x s'opère par la règle suivante :

$$x = d(A) = \sum_{i=1}^l a_i \cdot 2^{l-i-1}$$

Ainsi le chromosome $A = [1; 0; 1; 1]$ vaut $1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 8 + 2 + 1 = 11$.

Évidemment, la fonction d sera modifiée selon le problème. Ainsi si nous cherchons à maximiser une fonction, une méthode possible serait la suivante (la taille du chromosome dépendant bien évidemment de la précision voulue)

$$x = d(A) = \sum_{i=1}^l a_i \cdot 2^{-i-1}$$

Une autre façon de faire est de choisir d telle que :

$$x = d(A) = \sum_{i=1}^l \frac{a_i \cdot 2^{l-i-1}}{2^{l+1} - 1}$$

Cette dernière règle peut se généraliser. Ainsi, admettons que nous cherchons à maximiser f en fonction d'une variable réelle x . Soit $D = [x_{min}; x_{max}]$, avec $D \in \mathbb{R}$, l'espace de recherche permis avec x_{min} et x_{max} les bornes inférieures et supérieures. Soit $prec$ la précision (chiffre après la virgule) avec laquelle nous cherchons x . Soit $ld = x_{max} - x_{min}$ la longueur de l'intervalle D . Nous devons alors diviser cet intervalle en $ni = ld * 10^{prec}$ sous intervalles égaux afin de respecter la précision.

Exemple : soit $D = [-1; 2]$, nous avons donc $ld = 3$, si nous voulions une précision $prec = 6$, alors il nous faut diviser cet intervalle en $ni = 3000000$ sous intervalles.

Avec s l'entier naturel tel que $2^s > ni$ (dans notre exemple, $s = 22$ car $2^{21} = 2097152 < 3000000 < 2^{22} = 4194304$).

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

1. conversion (base 2 en base 10) : $x' = \sum_{i=1}^s 2^{i-1} a_i$
2. recherche du nombre réel correspondant : $x = x_{min} + x' \frac{x_{max} - x_{min}}{2^s - 1}$

Ce qui revient au même directement en une seule étape par :

$$x = x_{min} + \sum_{i=1}^s \frac{2^{i-1} \cdot a_i \cdot ld}{2^s - 1}$$

II.7 Opérateur de Croisement

L'opérateur de croisement permet la création de nouveaux individus selon un processus très simple. Il permet donc l'échange d'information entre les individus.

Tout d'abord, deux individus, qui forment alors un couple, sont tirés au sein de la nouvelle population issue de la reproduction. Puis un (potentiellement plusieurs) site de croisement est tiré aléatoirement (chiffre entre 1 et $l-1$). Enfin, selon une probabilité *prob* que le croisement s'effectue, les segments finaux (dans le cas d'un seul site de croisement) des deux parents sont alors échangés autour de ce site.

Cet opérateur permet la création de deux nouveaux individus. Toutefois, un individu sélectionné lors de la reproduction ne subit pas nécessairement l'action d'un croisement. Ce dernier ne s'effectue qu'avec une certaine probabilité. Plus cette probabilité est élevée et plus la population subira de changement.

II.7.1 Croisement en un point

Une position k est choisie uniformément entre 1 et $(l-1)$, deux nouvelles chaînes sont créées en échangeant tous les caractères entre les positions $(k+1)$ et l de chaque chaîne de la paire considérée, les nouvelles chaînes peuvent être totalement différentes de leurs parents. À remarquer qu'un croisement n'aura aucune conséquence sur un bit dont les parents ont la même valeur à la même position.

Notons que le croisement s'effectue directement au niveau binaire, et non pas au niveau des gènes. Le point de coupeur il est généralement au milieu.

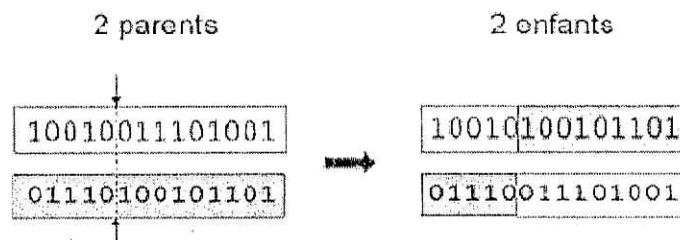


Figure 2.4 : représentation schématique du croisement en 1 point.

II.7.2 Croisement en deux points

Le même principe du croisement en un point il est appliqué au niveau du croisement en deux points à la différence que deux points de coupures sont choisis au hasard comme il est montré dans la Figure 2.5.

Cette opération est généralement considérée comme plus efficace que la précédente.

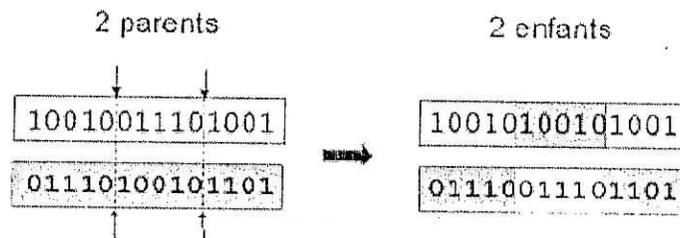


Figure 2.5 : représentation schématique du croisement en 2 points.

II.7.3 Croisement uniforme

Chaque bit du descendant est créé à partir du bit correspondant de l'un des parents. Les bits sont choisis selon un vecteur appelé vecteur masque de croisement. L'ors que il y a un 1 dans le vecteur masque, le bit est copié à partir du premier parent et l'ors qu'il y a un 0 le bit est copié à partir du second parent comme il est montré sur la Figure 2.6.

Le procédé est répété avec d'autres parents pour produire le second descendant, un nouveau vecteur masque est généré aléatoirement pour chaque paire des parents.

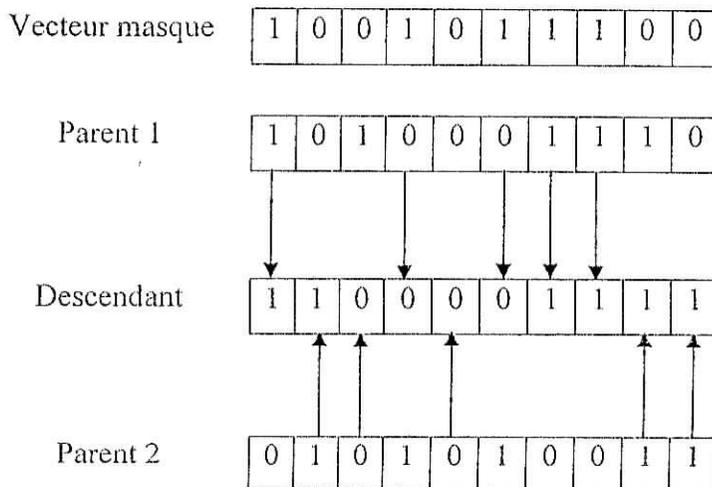


Figure 2.6 : représentation schématique du croisement uniforme.

La probabilité de croisement P_c est définie comme le pourcentage de nombre espéré des individus qui vont subir l'opération de croisement, par rapport à la taille de la population.

Une probabilité de croisement élevée permet une exploration plus large de l'espace de solutions et réduit les chances de fixation sur un faux optimum. Mais si cette probabilité est trop élevée, alors une quantité importante de temps de calcul risque d'être perdu dans l'exploration des régions non prometteuses de l'espace de solutions. Typiquement la probabilité de croisement varie entre 0.6 et 1.0.

II.8 Opérateur de mutation

Il peut arriver que certaines informations disparaissent de la population lors du processus de sélection des individus. L'opérateur de recombinaison ne permet en aucun cas de retrouver ces informations. C'est la raison essentielle qui a conduit à définir un autre opérateur, la *mutation*. Cet opérateur ne crée généralement pas de meilleurs individus, mais il évite l'établissement de populations uniformes incapables d'évoluer. La *mutation simple* consiste à modifier, avec une probabilité P_m faible, les allèles des bits des individus. La Figure 2.7 illustre l'application de l'opérateur de mutation simple sur un individu.

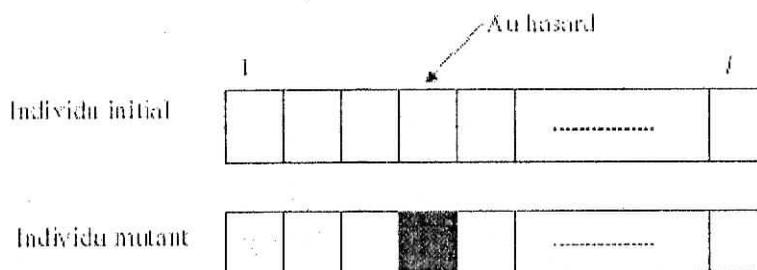


Figure 2.7 : Opération de mutation simple

Sans mutation, il y a un risque de convergence prématurée de l'algorithme vers un extremum local. Cependant, de manière à ne pas empêcher la convergence de l'algorithme, la probabilité de mutation doit rester faible (par exemple 0,001). En fait, la probabilité P_m optimale dépend de la taille de la population et de la longueur des individus.

Le choix de la meilleure probabilité de mutation P_m est délicat. En effet, P_m dépend à la fois de la taille N de la population et de la longueur l des individus. De plus, si P_m décroît avec le temps, les performances de l'algorithme sont meilleures. Enfin, P_m est inversement proportionnelle à la probabilité de croisement. Hesser et Männer proposent

$$P_m = \sqrt{\frac{\alpha}{\beta}} \cdot \frac{e^{-\tau \cdot t}}{N \sqrt{l}}$$

avec α , β , et τ constants.

Fogarty a montré qu'une décroissance exponentielle de P_m au cours de l'exécution de l'A.G. améliore ses performances. Hesser et Männer montrent que P_m doit augmenter si p_c diminue et réciproquement. On peut en conclure d'après le paragraphe précédent que p_c doit augmenter au cours de l'exécution de l'A.G. Ce résultat a déjà été donné par Booker.

II.9 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 efficace que le codage binaire et l'espace de recherche est le même que l'espace du 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 transcoding du binaire naturel vers les réels à chaque évaluation. Néanmoins, il possède deux inconvénients, son alphabet est infini, et il a besoin d'opérateurs appropriés.

A l'aide du codage binaire, toutes les opérations sont assez simples à mettre en place. Malgré tout, quelques inconvénients existent:

– *Il peut être difficile d'adapter ce codage à certains problèmes :* La représentation binaire traditionnelle utilisée pour les algorithmes génétiques crée des problèmes pour les problèmes d'optimisation de grandes dimensions à haute précision numérique. Par exemple, avec 100

variables appartenant au domaine $[-500; 500]$ et dont une précision de 6 chiffres après la virgule est requises, la taille de l'individu est 3000. Cela, en retour, génère un espace de recherche d'environ 1000^6 . Pour de tels problèmes, les algorithmes génétiques basés sur des représentations binaires ont de faibles performances.

- *La distance de Hamming* (La distance de Hamming entre deux chaînes de bits est le nombre de bits qui diffèrent de l'une à l'autre. Ainsi entre 01100101 et 00101100 la distance de Hamming vaut 3) entre deux nombres réels proches peut être grand (exemple : 0111 qui vaut 7 et 1000 qui vaut 8, la distance est de 4). Ce qui crée bien souvent une convergence mais non l'obtention de la valeur optimale.
- Suivant le problème, la résolution de l'algorithme peut être coûteux en temps.
- Le croisement et la mutation peuvent être inadaptés (création d'individus n'appartenant pas à l'espace de recherche).

Une des améliorations majeures consiste alors à se servir de nombres réels directement. Les résultats donnés par l'expérience montrent que la représentation réelle aboutit souvent à une bonne résultat et qu'en règle général le gain en termes de temps de calcul est positif. La conclusion à laquelle il arrive est naturellement qu'une représentation plus réelle du problème offre des solutions plus efficaces.

En utilisant le codage réel, notre individu n'est alors plus qu'un chiffre à valeurs réelles dans l'espace des valeurs permises : $A = a, a \in D \subset \mathbb{R}$. L'opérateur de sélection reste identique à celui de la roue de loterie biaisée ou du tournoi. En revanche, il existe d'autres opérateurs de croisement et de mutation.

II .10 Opérateur de croisement

Soient $Y = (y_1, y_2, y_3)$ et $X = (x_1, x_2, x_3)$ deux membres (vecteur de dimension trois) de la population initiale.

II .10.1 Croisement simple

Cette méthode est identique dans le principe à celle décrite auparavant. Pour ce faire, nous générons un nombre aléatoire r à partir d'une distribution uniforme sur l'ensemble

$f(1, 2, 3)$, et deux nouveaux individus, X' et Y' sont créés selon la règle suivante :

$$x'_i = \begin{cases} x_i & \text{si } i < r \\ y_i & \text{sinon} \end{cases}$$

$$y'_i = \begin{cases} y_i & \text{si } i < r \\ x_i & \text{sinon} \end{cases}$$

II.10.2 Croisement arithmétique

Ce croisement effectue une simple combinaison linéaire entre les parents. Soit, après avoir généré un chiffre aléatoire, $\alpha = U(0, 1)$, les nouveaux parents sont

$$x' = \alpha X + (1 - \alpha)Y$$

$$y' = (1 - \alpha)X + \alpha Y$$

II.10.3 Croisement heuristique

Enfin, il existe aussi le croisement heuristique. Cet opérateur effectue une extrapolation linéaire des deux individus. Un nouvel individu, X' est créé selon le processus suivant (sous l'hypothèse que $X > Y$ en terme de fitness, sinon nous inversons X et Y dans les équations) :

$$X' = X + r(X - Y)$$

$$Y' = X.$$

où/et

$$\text{Faisabilité } \epsilon = \begin{cases} 1 & \text{si } b_1^i < x_i < b_2^i, \forall i \\ 0 & \text{sinon} \end{cases}$$

Où b_1^i et b_2^i sont les bornes autorisées pour x_i , et avec r un nombre aléatoire tiré dans

$U(0; 1)$. Nous devons donc avoir tout le temps $x_i \in [b_1^i ; b_2^i]$. Si X' n'est pas faisable (i.e. faisabilité nulle) alors un nombre r est retiré et la procédure est recommencée jusqu'à ce que la solution soit faisable où qu'un certain nombre d'essais ait été effectué.

Dans le cas où $f(X) = f(Y)$ (même fitness) on reproduit simplement X et Y . Cet opérateur est le seul des croisements à utiliser directement une information reliée à la fitness. Cet opérateur est un croisement unique pour les raisons suivantes :

- il utilise les valeurs de la fonction objective afin de déterminer une direction de recherche.
- il produit seulement un enfant et il peut ne produire aucun enfant.

Il semble que le croisement heuristique contribue à trouver une solution plus précise ; ses principales responsabilités dans la recherche de la solution c'est de faire une recherche dans une direction prometteuse.

II.10.4 Croisement MPX

Ce croisement conserve dans fils i la zone interne de parent i (zone comprise entre $N1$ et $N2$ tirés aléatoirement). Ensuite, pour les cases vides restantes, le k_{ieme} chromosome de fils i sera rempli avec le k_{ieme} de parent j s'il ne provoque pas de doublon. A défaut, on essaiera le k_{ieme} chromosome de parent i ou enfin le premier chromosome de la zone interne de parent j qui ne provoque pas de doublon. Ce croisement est illustré sur la Figure 2.8. ($NB=12$).

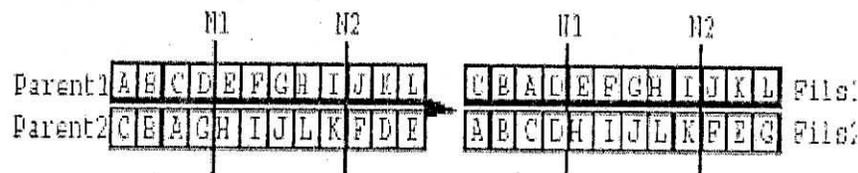


Figure 2.8 : Croisement MPX

II.11 Opérateur de mutation

II.11.1 Mutation uniforme

La mutation uniforme est identique à celle de la mutation binaire. Ainsi, chaque variable $x_i \in X$ est changée selon une certaine probabilité en un nombre aléatoire tiré dans une distribution uniforme sur l'intervalle $[b_1^i ; b_2^i]$, avec b_1^i et b_2^i les bornes inférieures et supérieures pour x_i .

II.11.2 Mutation non uniforme

La mutation non uniforme revient à changer la variable x_i en un nombre tiré dans une distribution non uniforme. Cette nouvelle variable x_i' est telle que :

$$x_i' = \begin{cases} x_i + (b_2^i - x_i)f(G), & \text{si } \alpha < 0.5 \\ x_i - (x_i - b_1^i)f(G), & \text{si } \alpha \geq 0.5. \end{cases}$$

avec

$$f(G) = \left(\alpha' \left(1 - \frac{G}{G_{\max}}\right)\right)^b$$

α, α' : nombres aléatoires $\in (0; 1)$;

G : le chromosome courant;

G_{\max} : le nombre maximum de chromosome (i.e. de création de nouvelle population);

b : un paramètre déterminant le degré de non uniformité.

II.11.3 Mutation dans les bornes

Un dernier opérateur de mutation existe : la mutation dans les bornes. Avec cet opérateur, chaque variable $x_i \in X$ choisie pour muter prend pour valeur l'une des deux bornes b_1^i ou b_2^i avec équiprobabilité. A l'évidence, cet opérateur n'a d'intérêt et d'efficacité que si la solution est proche des bornes de l'espace de recherche.

Notons qu'il est possible de combiner plusieurs opérateurs en même temps.

II.12 Mécanisme de remplacement

II.12.1 Espace d'échantillonnage régulier

Dans l'algorithme génétique original de HOLLAND les parents sont remplacés par leurs descendants juste après leurs naissances, parce que les opérateurs génétiques sont de natures aveugle, les descendantes peuvent être plus mauvaise que leur parents, avec la stratégie de remplacement de chaque parent par son descendant directement, certains individus promoteurs seront perdus du processus d'évolution. Pour résoudre ce problème plusieurs stratégies de remplacement ont été examinées. On suggère qu'à chaque naissance d'un descendant, on remplace un individu choisi aléatoirement dans la population courante. DE JONG a proposé une stratégie d'entassement qui consiste à sélectionner un parent pour mourir, à chaque naissance d'un descendant. Le parent mort est soit le parent le plus faible ou celui qui ressemble le plus fermement au nouveau descendant. En utilisant une comparaison bit à bit pour mesurer la ressemblance. La Figure 2.9 illustre la sélection basée sur l'espace d'échantillonnage régulier.

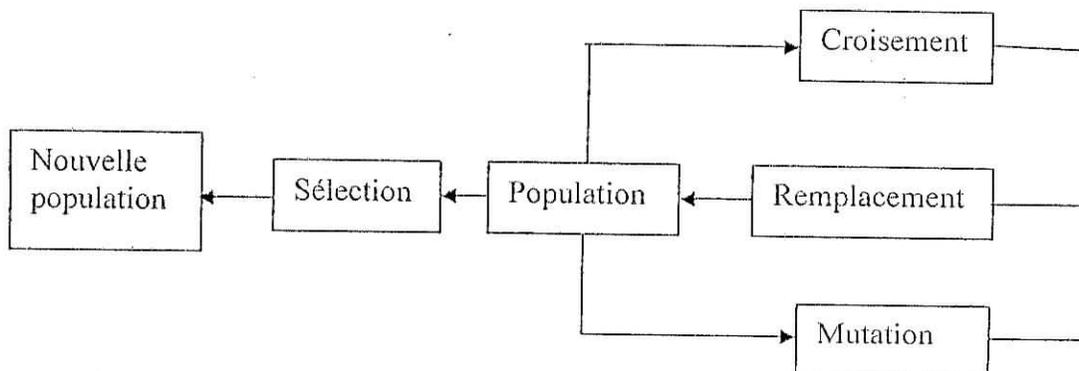


Figure 2.9 : Sélection dans l'espace d'échantillonnage régulier.

II.12.2 Espace d'échantillonnage élargi

Quand la sélection est réalisée dans un espace d'échantillonnage élargi. Les parents et les descendants ont la même chance de survivre. Un cas typique est la sélection $(\lambda + \mu)$. Avec cette stratégie, μ parents et λ descendants sont en compétition pour survivre, et les μ meilleurs

parmi ces descendants et anciens parents sont sélectionnés comme parent de la nouvelle génération. Bien que la plupart des méthodes de sélection soient basées sur l'espace d'échantillonnage régulier. Il est facile de les implémenter dans l'espace d'échantillonnage élargi. La figure 111. Illustre la sélection basée sur l'espace d'échantillonnage élargi.

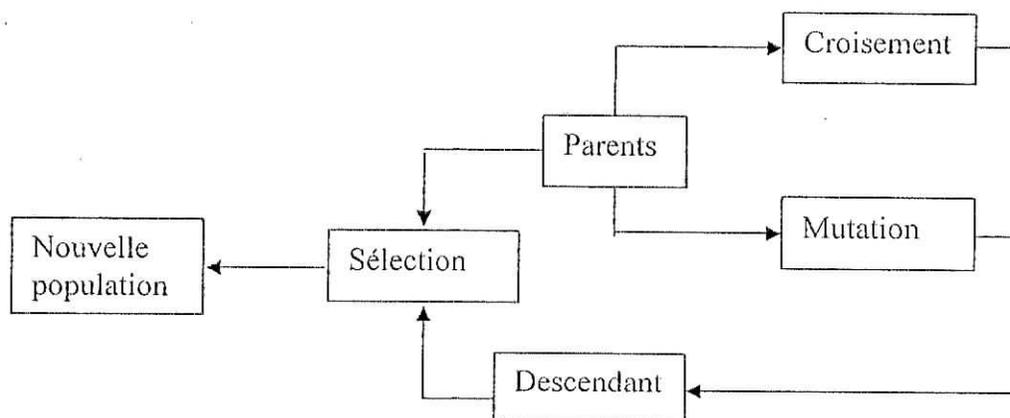


Figure 2.10 : Sélection dans l'espace d'échantillonnage élargi

Remarque

Les opérateurs de l'algorithme génétique sont guidés par un certain nombre de paramètres fixés à l'avance.

La valeur de ces paramètres influence sur la réussite ou non d'un algorithme génétique. Ces paramètres sont les suivants :

1. La taille de la population, N , et la longueur du codage de chaque individu l (dans le cas du codage binaire). Si N est trop grand le temps de calcul de l'algorithme peut s'avérer très important, et si N est trop petit, il peut converger trop rapidement vers un mauvais chromosome.
2. La probabilité de croisement P_c . Elle dépend de la forme de la fonction de fitness. Son choix est en général heuristique (tout comme pour P_m). Plus elle est élevée, plus la population subit de changements importants. Les valeurs généralement admises sont comprises entre 0,6 et 1.

3. La probabilité de mutation P_m . Ce taux est généralement faible puisqu'un taux élevé empêche la convergence de l'algorithme.

II.13 Optimisation des fonctions simples par les algorithmes génétiques

II.13.1 Fonction a une variable

Appliquons étape par étape notre algorithme génétique de base a un problème d'optimisation. Considérons l'objectif de minimiser la fonction $f(x) = (x-15.5)^2 + 4$, ou x peut varier entre 0 et 31 figure 12.

Le minimum peut être calculé en annulant la dérivée de la fonction objective :

$$\frac{df}{dx} = 2.(x - 15.5) = 0 \Rightarrow x = 15.5$$

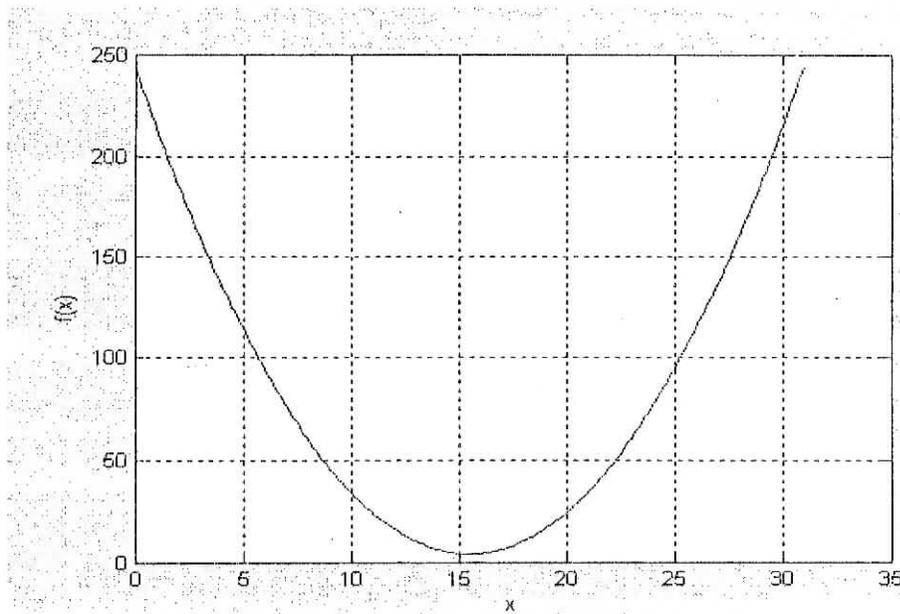


Figure 2.11 : schéma d'une fonction convexe

II.13.1.1 Le codage

Pour utiliser un algorithme génétique, nous devons d'abord coder la variable de décision de notre problème en une chaîne de longueur finie.

Pour ce problème, nous coderons simplement la variable x en un entier naturel de base 2 à 5 chiffres. Avant de continuer notre "simulation", revoyons rapidement le concept d'entier binaire. En tant que créatures avec dix doigts, il nous est facile de manipuler les entiers et l'arithmétique de base 10.

Par exemple : le nombre de 5 chiffres 53095 peut être représenté comme
 $5 * 10^4 + 3 * 10^3 + 0 * 10^2 + 9 * 10^1 + 5 * 1 = 53095$

En arithmétique de base 2, nous bien entendu que deux chiffres (0, 1) et le nombre 10011 se décode en le nombre de base 10.

$$1 * 2^4 + 0 * 2^3 + 0 * 2^2 + 1 * 2^1 + 1 * 2^0 = 19.$$

Avec un entier naturel de 5 chiffres binaires, on peut coder les nombres entre 0(00000) et 31 (11111). Avec une fonction à optimiser et un codage bien définis, nous allons maintenant simuler une génération d'un algorithme génétique avec reproduction, croisement, et mutation.

II.13.1.2 Tirage et évaluation de la population initiale

Pour commencer, nous obtenons une population de taille 4 en tirant à pile ou face 20. en regardant cette population, que l'on peut voir sur la partie gauche du tableau 1 nous observons que les valeurs décodées de x sont données avec les valeurs $f(x)$ de la fonction à optimiser. Pour s'assurer que nous savons comment calculer les valeurs $f(x)$ à partir de la représentation sous forme de chaîne, jetons un œil à la troisième chaîne de la population initiale (11011). En décodant cette chaîne, nous obtenons ainsi $x = 27$ pour la chaîne 11011.

Pour calculer la valeur de la fonction à optimiser, il suffit de prendre
 $f(x) = (x - 15.5)^2 + 4$. Ce qui donne $f(x) = 136.25$. on peut obtenir d'autres valeurs de x et de $f(x)$ de la même manière.

Numéro de chaîne	Population initiale	X(entier positif)	Fonction d'adaptation f(x)	pelect	Nombre attendu	Nombre obtenu
1	10011	19	16.25	0.0766	0.3064	1
2	10110	22	46.25	0.2181	0.8724	1
3	11011	27	136.25	0.6426	2.5704	0
4	01100	12	13.25	0.0625	0.25	2
Somme		80	212	1.00	4.00	4.0
moyenne		20	53	0.25	1.00	1.0
max		27	136.25	0.6426	2.5704	2.0

Après reproduction	Partenaire (Sélection aléatoire)	Lieu de croisement (Sélection aléatoire)	Nouvelle population	x	f(x)
10011	2	3	01011	11	24.25
01100	1	3	10100	20	24.25
10110	4	2	01110	12	16.25
01100	3	2	10100	20	24.25
Somme				63	89
moyenne				15.75	22.25
max				20	24.25

Nombre des générations	cinquième population	x	f(x)
5	01100	12	16.25
	01110	14	6.25
	10001	17	6.25
	10000	16	4.25
Somme		59	33
moyenne		14.75	8.25
max		17	16.25

Figure 2.12 : Un algorithme génétique simple

II .13.1.3 Sélection

Une nouvelle génération de l'algorithme génétique commence par la reproduction. On sélectionne le groupe de reproduction pour la génération suivante en faisant tourner la roue de loterie quatre fois. Une simulation de ce processus à l'aide de tirage au sort a conduit à la copie d'une chaîne 1 et d'une chaîne 2, à deux copies de la chaîne 4, et à aucune copie de la chaîne 3, comme on peut le dire dans la table 1. En comparant ceci avec le nombre attendu de copies ($n \cdot p_{\text{select}_i}$), on peut remarquer que l'on a obtenu ce qu'on pouvait espérer: les meilleurs sont copiés en plus grand nombre, les pires meurent, et les autres restent en nombre constant.

II .13.1.4 Le croisement

Dans le groupe sélectionné pour l'accouplement, le croisement simple a lieu en deux étapes:

- les chaînes sont appariées au hasard, en utilisant des tirages au sort pour regrouper par paires les nouveaux couples.
- on croise les chaînes appariées, en utilisant des tirages au sort pour choisir les lieux de croisement.

En se référant de nouveau au Tableau 1, on voit qu'un choix aléatoire de partenaires a sélectionné un accouplement des chaînes 1 et 2. Le site de croisement étant 3, les deux chaînes 10011 et 01100 s'enjambent et produisent deux nouvelles chaînes 01011 et 10100

Les deux chaînes restantes dans le groupe sont croisées au site 2; les chaînes ainsi produites peuvent être examinées dans le tableau.

II .13.1.5 La mutation

Le dernier opérateur, la mutation, est appliqué bit par bit. Nous supposons ici que la probabilité de mutation est de 0.001. Considérant les 20 bits de notre exemple, on pourrait s'attendre à ce que $20 \cdot 0.001$, c'est-à-dire 0,02 bits, subissent une mutation à chaque génération. Une simulation de ce processus avec cette probabilité indique qu'aucun bit ne

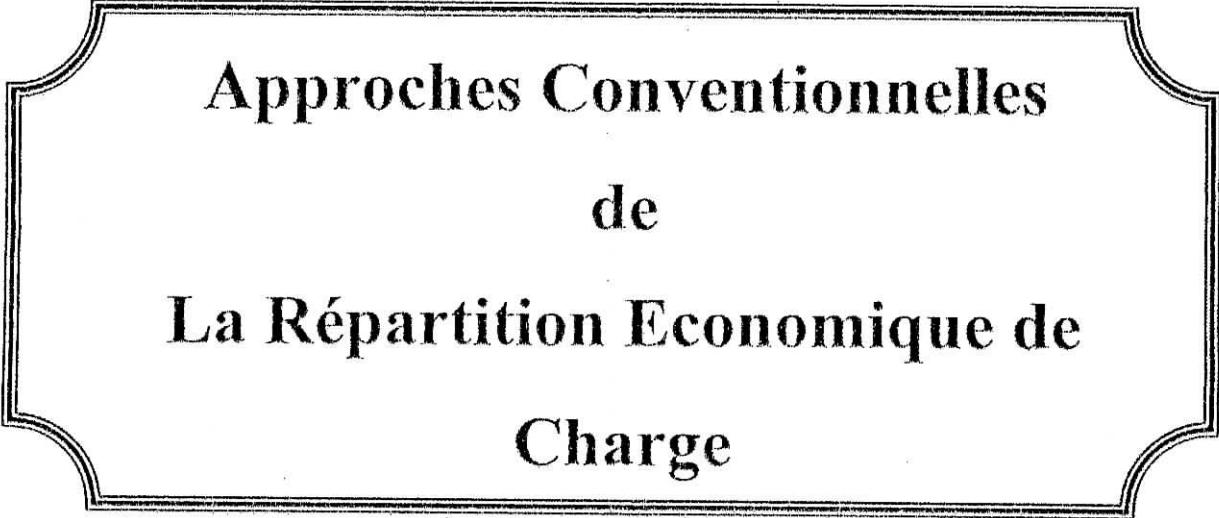
subit de mutation. Il y' a donc pas de bit changé de 0 a 1 ou vice versa pendant cette génération.

II .13.1.6 Retour à la phase d'évaluation

Après reproduction, croisement, et mutation, la nouvelle population est prête à être testée. A cette fin, il suffit de décoder les chaînes nouvellement créées par l'algorithme génétique de base et de calculer la valeur de la fonction d'adaptation pour les valeurs obtenues grâce a ce décodage. Les résultats d'une génération de la simulation sont présentés dans le tableau 1. Bien qu'il soit dangereux se tirer des conclusions pratiques a partir d'un unique essai d'une procédure aléatoire, nous commençons a percevoir la façon dont les algorithmes génétiques combinent les notions très performantes pour atteindre de meilleurs résultat. Dans le tableau, on note qu'à la fois les performances maximales et moyennes sont améliorées dans la nouvelle population. L'adaptation moyenne de la population et passée de 53 a 22. 25. en une génération.

Au court des générations la fonction objective converge bien vers la valeur optimale. On voit clairement que l'adaptation moyenne passe de 22.25 pour la deuxième génération vers 8.25 pour la cinquième génération.

CHAPITRE III



**Approches Conventionnelles
de
La Répartition Économique de
Charge**

III. 1 Introduction

Ce chapitre se penche sur la définition du problème physique à traiter « problème de la répartition économique de charge dans les systèmes d'énergie électrique », ainsi qu'un certain nombre de techniques d'optimisation conventionnelles utilisées pour le résoudre.

Cependant penchons nous un peu sur la formulation globale de problème.

III. 2 Problème de la répartition économique de charge

Le terme général de répartition économique se réfère à des formulations où les coûts de fonctionnement ou d'investissement sont minimisés. Dans notre cas, la répartition économique concerne principalement l'écoulement de charge optimal du point de vue coût, compte tenu des contraintes imposées au réseau et aux groupes de production. La principale contrainte est l'obligation que la somme des capacités des générateurs en service doivent avoir une puissance égale à la demande totale de la charge plus les pertes.

III. 3 Formulation générale

Considérons un réseau d'énergie électrique à N nœuds, où :

N_g : est le nombre total de nœuds de génération dans le réseau.

N_c : est le nombre total de nœuds de charge dans le réseau.

Remarque un nœud de génération peut être à la fois un nœud de génération et un nœud de charge (dans le cas où une charge est très proche de la centrale).

La Répartition Economique de Charge revient à calculer un vecteur de contrôle U (niveaux de génération des unités de production en MW), qui satisfait à la fois la puissance totale appelée demande totale des charges incluant les pertes actives totales dans le réseau, et le coût de production minimum, ou tout au moins une estimation proche.

La formulation générale s'écrit ainsi :

$$\min \quad F_T(P_i) = \sum_{i=1}^{N_g} F_i(P_i) \quad (3.1)$$

$$\text{Sous la contrainte} \quad G(P_i) = 0$$

Où

F_T représente le coût de production total du système qui est la somme des coûts de chacune des différentes unités de production en service (F_T est considérée fonction des seules puissances P_i et est nécessairement une fonction croissante de chacune d'elles),

P_i est le niveau de génération (puissance produite) par le générateur qui débite sur le $i^{\text{ème}}$ nœud.

La contrainte égalité s'écrit :

$$G(P_i) = \sum_{i=1}^{N_g} P_i - \sum_{i=1}^{N_c} C_i - P_L(P_i) = 0 \quad (3.2)$$

C_i : la consommation des charges alimentées à partir du nœud i .

C : vecteur des puissances spécifiées aux nœuds de charge.

P_L : pertes totales dans le réseau de transmission.

Les consommations C_i sont considérées comme des données fixes dans le problème à résoudre (puisque ce problème est résolu selon la prévision de la courbe de charge et non pas en temps réel), donc G est bien une fonction des seules puissances produites P_i .

La fonction $G(P_i)$ est une fonction croissante de chacune des puissances P_i , car il est physiquement impossible que les pertes $P_L(P_i)$ croissent plus vite que P_i .

Les fonctions F et G respectent, en pratique, toujours les propriétés de convexité.

Exemple d'un réseau électrique

Afin de faciliter la compréhension des détails des différentes techniques, l'explication de principe de chaque méthode, est suivi par un exemple d'application.

Nous raisonnerons en prenant l'exemple d'un réseau à trois (3) unités thermiques de génération (voir données en annexe) mais l'extension à tous autres cas et, en particulier, la généralisation à un nombre quelconque de générateurs ne soulève aucune difficulté.

La Figure 3.1 montre la configuration qui sera étudiée dans cette partie.

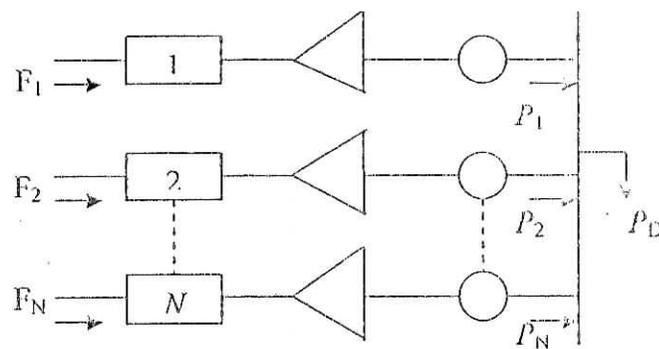


Figure. 3.1 Représentation d'un système à N unités de production.

Ce système se compose de N unités de génération connectées pour satisfaire la charge électrique demandée P_D , égale à la somme des consommations des charges branchés au réseau.

Pour tout le reste du chapitre, nous considérerons un système de $N=3$ unités.

Les unités de production consomment du carburant à un taux spécifique (MBtu/h), qui peut se traduire en R/h, c'est à dire un taux de coût (R étant une unité monétaire quelconque).

Les caractéristiques des trois unités de production sont présentées en annexe.

III. 4 Résolution de problème de la répartition économique de charge

Différentes méthodes approchées sont utilisées pour cela, mais ne sont valables, chacune, que dans des conditions bien déterminées ou avec des hypothèses particulières. comme montré dans ce qui suit.

Le but de la répartition économique de charge est de contrôler les variations du coût total de production de façon à le rendre à tout moment minimal, ou tout au moins aussi proche que possible du minimum, compte tenu des :

- contraintes imposées aux groupes de production.
- pertes de transmission.
- contraintes imposées au réseau de transport.

Pour une meilleure illustration de la complexité du problème et de la manière dont on y fait face, des approches simples et graduelles sont présentées.

III. 4.1 Méthode des Coûts Marginaux

Rappelons que la répartition économique de charge est un problème d'optimisation d'une fonction coût F sous une contrainte égalité G tel que les fonctions F et G respectent, en pratique, toujours les propriétés de convexité. Donc ce problème peut être attaqué formellement en utilisant les méthodes de calcul avancées qui impliquent la fonction de Lagrange, dont la Méthode des Coûts Marginaux

III. 4.1.1 Méthode des Coûts Marginaux sous la seule contrainte

"Production = Consommation"

Dans ce cas, la contrainte essentielle et unique du fonctionnement de ce système est que la somme des puissances produites par les N unités doit être strictement égale à la charge totale demandée.

Mathématiquement parlant, le problème se traduit par la minimisation d'une fonction objective F_T égale au coût de fonctionnement total pour satisfaire la charge demandée. Le problème est

donc de réduire au minimum F_T sous la contrainte que la somme des puissances générées est égale à la charge demandée, les pertes de transmission et autres étant négligées et aucune limite de fonctionnement n'étant introduite.

La formulation de ce problème devient :

$$F_T = \sum_{i=1}^N F_i(P_i) \quad (3.3)$$

$$G = P_D - \sum_{i=1}^N P_i = 0 \quad (3.4)$$

Afin d'établir les conditions nécessaires pour avoir une valeur extrême de la fonction objective, on ajoute la fonction contrainte à la fonction objective après que la fonction contrainte ait été multipliée par un multiplicateur indéterminé λ . Ceci est connu comme la *fonction de Lagrange* et est montré dans l'Eq. 3.3.

$$L = F_T + \lambda G \quad (3.5)$$

Les conditions nécessaires pour obtenir une valeur extrême de la fonction objective résultent de la résolution du système d'équations obtenu en dérivant la fonction de LaGrange par rapport à chacune des variables indépendantes, ces équations étant égales chacune à zéro. On compte ainsi $N+1$ variables dont N valeurs sont les puissances P_i générées plus le multiplicateur indéterminé de Lagrange λ . Notons que la dérivée de la fonction de Lagrange par rapport à λ donne évidemment l'équation de contrainte $G = 0$. Les dérivées premières par rapport aux N variables de puissances sont données par l'équation 3.6.

$$\frac{\partial L}{\partial P_i} = \frac{dF_i(P_i)}{dP_i} - \lambda = 0 \quad \text{avec } i=1, 2, 3, \dots, N \quad (3.6)$$

Cela signifie que la condition nécessaire pour l'existence d'un fonctionnement de ce système avec un coût minimum réside dans le fait que les incréments de coût de toutes les unités soient égaux à une certaine valeur λ . Naturellement, à cette condition nécessaire, s'ajoute l'équation de contrainte que la somme des puissances produites doit être égale à la puissance totale demandée par la charge.

Exemple 3.1

Supposons que nous souhaitons déterminer le point de fonctionnement économique (la répartition économique de charge) pour un système à trois unités (voir données en annexe) pour satisfaire une charge totale de 850 MW, les coûts de carburant étant les suivants .

Unité 1: Coût de carburant = 0,9 R/MBtu

Unité 2: Coût de carburant = 1,0 R/MBtu

Unité 3: Coût de carburant = 1,0 R/MBtu

Ce qui donne

$$F_1(P_1) = H_1(P_1) \times 0,9 = 459 + 6,48 P_1 + 0,00128 P_1^2 \text{ R/h}$$

$$F_2(P_2) = H_2(P_2) \times 1,0 = 310 + 7,85 P_2 + 0,00194 P_2^2 \text{ R/h}$$

$$F_3(P_3) = H_3(P_3) \times 1,0 = 78 + 7,97 P_3 + 0,00482 P_3^2 \text{ R/h}$$

En utilisant Eq. (3.6), les conditions pour une répartition optimale sont :

$$\frac{dF_1}{dP_1} = 6.48 + 0.00256 P_1 = \lambda$$

$$\frac{dF_2}{dP_2} = 7.85 + 0.00388 P_2 = \lambda$$

$$\frac{dF_3}{dP_3} = 7.97 + 0.00964 P_3 = \lambda$$

et

$$P_1 + P_2 + P_3 = 850 \text{ MW}$$

La résolution de λ donne :

$$\lambda = 8,284 \text{ R / MWh}$$

et les puissances générées sont donc respectivement :

$$P_1 = 704,6 \text{ MW} \quad P_2 = 111,8 \text{ MW} \quad P_3 = 32,6 \text{ MW}$$

Cette solution vérifie la contrainte exigeant que la production totale soit égale à 850 MW.

La solution du problème telle qu'elle vient d'être exposée peut conduire à une solution irréalisable, si la puissance active imposée à certains groupes est incompatible avec sa sécurité (surcharge ou puissance inférieure au minimum technique).

Il est donc nécessaire de prendre en considération les limites techniques imposées aux groupes de production.

III. 4.1.2 Méthode des coûts marginaux sous les contraintes imposées aux groupes de production

En plus de la condition que les incréments de coût de toutes les unités soient égaux à une certaine valeur λ , et l'égalité entre la somme des puissances produites et la puissance demandée par la charge, on doit ajouter une double contrainte inégalité pour chacune des puissances générées, cette double inégalité contraignant chacune des unités de production à ne pas dépasser ses limites inférieure et supérieure autorisées.

Ces conditions et inégalités peuvent se résumer par l'ensemble des équations suivantes.

$$\begin{aligned} \frac{dF_i}{dP_i} &= \lambda && N \text{ équations} \\ P_{i, \min} &\leq P_i \leq P_{i, \max} && 2N \text{ inégalités} \\ \sum_{i=1}^N P_i &= P_D && \text{l'équation contrainte} \end{aligned} \quad (3.7)$$

En identifiant les contraintes inégalités, les conditions nécessaires peuvent être légèrement modifiés comme montré par le système (3.8).

$$\begin{aligned} \frac{dF_i}{dP_i} &= \lambda && \text{si } P_{i, \min} < P_i < P_{i, \max} \\ \frac{dF_i}{dP_i} &\leq \lambda && \text{si } P_i = P_{i, \max} \\ \frac{dF_i}{dP_i} &\geq \lambda && \text{si } P_i = P_{i, \min} \end{aligned} \quad (3.8)$$

Exemple 3.2

Reprenons l'exemple précédent en considérant cette fois ci les limites de production de chacune des unités de production (voir annexe).

Les résultats obtenus donnent :

$$\lambda = 8,284 \text{ R / MWh}$$

et

$$P_1 = 704,6 \text{ MW} \quad P_2 = 111,8 \text{ MW} \quad P_3 = 32,6 \text{ MW}$$

Nous remarquons que cette solution vérifie la contrainte exigeant que la production totale soit égale 850 MW, mais elle reste irréalisable puisque les unités 1 et 3 ne sont pas dans les limites autorisées.

Pour avoir la répartition la plus économique tout en respectant les limites d'unités, appliquons le système (3.8).

Supposons que l'unité 1 est placée à sa production maximale et l'unité 3 à sa production minimale. La répartition devient

$$P_1 = 600 \text{ MW} \quad P_2 = 200 \text{ MW} \quad P_3 = 50 \text{ MW}$$

A partir de (3.8), le coût incrémental λ se limite à celui de l'unité 2 puisque cette dernière n'est pas à l'une ou l'autre de ses limites. Aussi :

$$\lambda = \left. \frac{dF_2}{dP_2} \right|_{P_2=200} = 8.626 \text{ R/ MWh}$$

Nous calculons ensuite l'incrément de coût pour les unités 1 et 3 pour voir s'ils rencontrent les conditions de l'Eq. 3.8.

$$\left. \frac{dF_1}{dP_1} \right|_{P_1=600} = 8.016 \text{ R/ MWh}$$

$$\left. \frac{dF_3}{dP_3} \right|_{P_3=50} = 8.452 \text{ R/ MWh}$$

Notons que l'incrément de coût de l'unité 1 est le plus petit λ . Ainsi l'unité 1 devrait être à son maximum. Cependant l'incrément de coût de l'unité 3 n'est pas plus grand que celui de l'unité 2, ce qui fait que l'unité 3 ne devrait pas être forcée à son minimum. Pour trouver la

répartition optimale, on doit laisser les incréments de coût pour les unités 2 et 3 égaux à λ , comme suit.

$$P_1 = 600 \text{ MW}$$

$$\frac{dF_2}{dP_2} = 7.85 + 0.00388 P_2 = \lambda$$

$$\frac{dF_3}{dP_3} = 7.97 + 0.00964 P_3 = \lambda$$

$$P_2 + P_3 = 850 - P_1 = 250 \text{ MW}$$

Ce qui donne

$$\lambda = 8,576 \text{ R / MWh}$$

et

$$P_1 = 600 \text{ MW} \quad P_2 = 187,1 \text{ MW} \quad P_3 = 62,9 \text{ MW}$$

Notons que cette répartition vérifie les conditions de (3.8), puisque

$$\left. \frac{dF_1}{dP_1} \right|_{P_1=600} = 8.016 \text{ R / MWh}$$

Ce qui est moins que le λ , tandis que $\frac{dF_2}{dP_2}$ et $\frac{dF_3}{dP_3}$ sont tous deux égaux à λ .

Notons alors que toutes les contraintes sont vérifiées, c.-à-d., chaque unité est dans ses limites de production maximale et minimale autorisées, et la somme des puissances produites par les trois unités égale à la production total désirée de 850 MW.

Finalement on peut faire la conclusion suivante :

Si les pertes du réseau sont négligeables par rapport à la puissance totale produite, on considérera leurs variations comme nulles, et les variations du coût total de production ne dépendront, pour un niveau de production donné, que de la répartition de charge entre les groupes.

Ce coût total sera alors minimal si tous les groupes qui ne produisent pas leur puissance maximale fonctionnent à une puissance telle que leurs coûts marginaux de production soient égaux (pour les groupes portés à leur puissance maximale, le coût marginal est plus faible).

Mais cette méthode n'est adaptée qu'aux réseaux à forte prépondérance thermique et à usines réparties et proches des charges (d'où faible importance des pertes de transport d'énergie), tel que le réseau de la Grande-Bretagne.

Mais comme, malheureusement, ce n'est pas toujours le cas, c'est à dire, la plupart des réseaux existants ayant des pertes de transmission non négligeables et même parfois importantes, il faut en tenir compte afin d'établir une répartition économique plus juste.

III. 4.1.3 Méthode des coûts marginaux avec considération des pertes

La Figure 3.2 montre symboliquement tout le système thermique de production d'électricité relié à un nœud de charge à travers un réseau de transmission. Le problème de la répartition économique de charge associé à cette configuration particulière est légèrement plus compliqué, que le cas précédent.

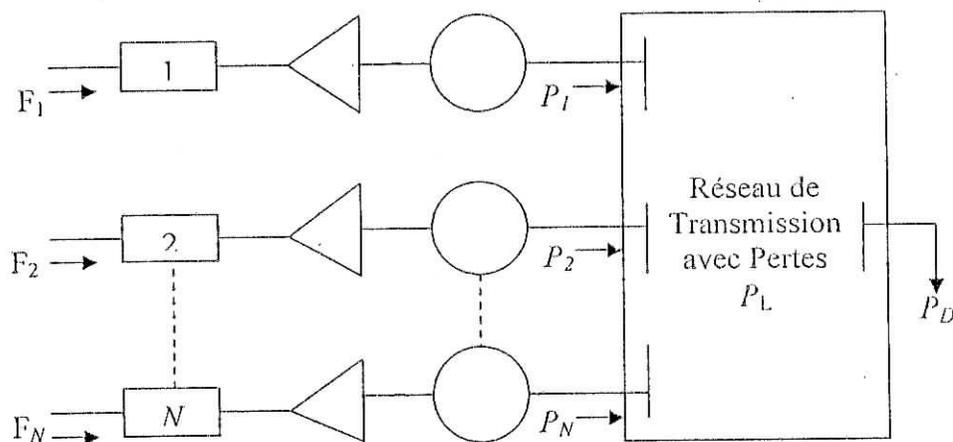


Figure. 3.2 N unités thermiques servant une charge à travers un réseau avec pertes.

Ceci est dû à l'équation contrainte incluant les pertes de réseau. La fonction objective, F_T , est identique à celle défini pour l'Eq.3.3.

Cependant, l'équation de contrainte précédemment montrée dans Eq. 3.4 doit maintenant être modifiée par l'Eq. 3.9.

$$G = P_D + P_L - \sum_{i=1}^N P_i = 0 \quad (3.9)$$

Le même procédé est suivi dans le sens formel d'établir les conditions nécessaires pour un point de fonctionnement à coût minimum. La fonction de Lagrange est montrée par l'Eq. 3.10

$$L = F_T + \lambda G \quad (3.10)$$

En dérivant la fonction de Lagrange par rapport à chacune des N valeurs P_i , on obtient l'Eq. 3.11 :

$$\frac{\partial L}{\partial P_i} = \frac{dF_i(P_i)}{dP_i} - \lambda \left(1 - \frac{\partial PL}{\partial P_i}\right) = 0 \quad \text{avec } i=1, 2, 3, \dots, N \quad (3.11)$$

L'interprétation de cette expression est la suivante :

la répartition des puissances produites P_i qui rend le coût total minimal est telle que les incréments des coût des différents unités de production sont égaux à un facteur près appelé facteur de pénalité.

Ce facteur de pénalité a pour expression $\frac{1}{\left(1 - \frac{\partial PL}{\partial P_i}\right)}$

Et le résultat que l'on vient d'énoncer se résume dans l'Eq.3.12

$$\frac{dF_i}{dP_i} \times \frac{1}{\left(1 - \frac{\partial PL}{\partial P_i}\right)} = \lambda \quad (3.12)$$

Il y a N équations de ce type à satisfaire avec l'équation de contrainte montrée dans Eq. 3.11.

$$\frac{dF_i}{dP_i} \times \frac{1}{\left(1 - \frac{\partial PL}{\partial P_i}\right)} = \lambda \quad (3.11)$$

$$G = P_D + P_L - \sum_{i=1}^N P_i = 0 \quad (3.9)$$

Cette collection, l'Eq. 3.12 Plus l'Eq. 3.9, est connue sous le nom des *équations de coordination*.

Les incréments de coût de production étant connus, il suffit donc, pour résoudre ce problème, de calculer les facteurs de pénalité, ou, ce qui revient au même, au calcul des pertes différentielles ($\frac{\partial PL}{\partial P_i}$)

Il est beaucoup plus difficile de résoudre le présent ensemble d'équations par rapport à l'ensemble précédent où les pertes de transmission sont négligées puisque cette seconde implique le calcul des pertes de réseau de transmission, P_L qui est très complexe car il nécessite le calcul de chacune des impédances de réseau et des courants transitant dans les branches du réseau.

$$PL = \sum r_{ij} I_{ij}^2 \quad (3.13)$$

Les méthodes de résolution proposées ont pour but d'éviter ce calcul.

L'idée de base repose sur le développement de l'expression mathématique des pertes ci-dessus, en explicitant les courants I_{ij} comme une fonction des seules variables indépendantes P_i et la charge P_D , la consommation P_D est considérée comme une donnée fixe dans le problème à résoudre (à chaque instant (la prévision)), ainsi, l'expression des pertes est bien une fonction des seules variables, les puissances produites P_i .

Différentes méthodes approchées sont utilisées pour cela, mais ne sont valables, chacune, que dans des conditions déterminées.

Méthode des coefficients β [Pél.8]

On appelle ainsi une méthode d'usage courant dans certains réseaux dont la production est essentiellement thermique tels que les réseaux de l'Est des U.S.A.

Elle a l'avantage d'être simple, mais elle repose sur des hypothèses qu'on ne peut considérer comme acceptables que dans les réseaux que l'on vient de définir et, même là, à condition d'être rectifiée plusieurs fois par jour.

L'idée de base c'est que les pertes étant une fonction quadratique des courants transités :

$$PL = \sum r_{ij} I_{ij}^2$$

Elles doivent être approximativement aussi une fonction quadratique des puissances produites, car les variations des modules des courants I s'écartent peu, en fonction des puissances actives P_i d'une loi linéaire de variation.

Une expression courante des pertes :

$$P_L = [P_1 \ P_2 \ \dots \ P_N] \begin{bmatrix} B_{11} & B_{12} & \dots & B_{1N} \\ B_{21} & B_{22} & & \vdots \\ \vdots & & \ddots & \\ B_{N1} & \dots & & B_{NN} \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \\ \vdots \\ P_N \end{bmatrix} + [B_{10} \ B_{20} \ \dots \ B_{N0}] \begin{bmatrix} P_1 \\ P_2 \\ \vdots \end{bmatrix} + B_{00} \quad (3.14)$$

Ce qui implique

$$P_L = \sum_{i=1}^N \sum_{j=1}^N P_i \beta_{ij} P_j + \sum_{i=1}^N \beta_{i0} P_i + \beta_{00} \quad (3.15)$$

Avec

β_{ij} : coefficient des pertes de la ligne ij .

β_{i0} : coefficient des pertes des impédances shunt liées au nœud i .

β_{00} : constante représentant les pertes du système lorsque la demande de la charge est nulle, les pertes résultantes des courants de ligne causés par le mouvement de la puissance réactive.

Les coefficients β_{ij} , β_{i0} , β_{00} dépendent des valeurs de V_i , θ_i et Q_i (le module de la tension, sa phase et la puissance réactive injectée dans le réseau à partir de nœud i , respectivement) ce n'est donc que dans des conditions telles que ces grandeurs n'aient que des variations négligeables que la formule (3.15) reste utilisable.

Ainsi, les pertes différentielles auront pour expression

$$\frac{\partial P_L}{\partial P_i} = \sum_{j=1}^N \beta_{ij} P_j + \beta_{i0} \quad (3.16)$$

En pratique, les modules U_i varient peu et les puissances réactives Q_i ont généralement des variations bien corrélées avec celles des P_i , par contre les déphasages θ_i peuvent accuser des variations sensibles dès que les flux de puissance subissent des distorsions dans l'espace.

Si le réseau est essentiellement alimenté par des centrales thermiques, on peut assez bien classer ses états en deux groupes : états d'heures creuses et états d'heures pleines, les angles θ_i ne variant sensiblement que lorsqu'on passe de l'un à l'autre.

Pour les pays à forte production hydraulique, et à consommations dispersées plus ou moins loin des usines génératrices, les coefficients de l'équation (3.16) seraient à modifier trop souvent, et la méthode des coefficients β devient impraticable.

Autres méthodes de calcul des pertes différentielles ont apparues dont :

- La méthode de Brownlee. [Pél.8]
- et
- La méthode du Jacobien. [Pél.8]

Exemple 3.3

Commençons avec les mêmes unités comme pour l'exemple 3.1, en gardant les mêmes coûts de carburant pour les unités 2 et 3, et supposons que le prix du charbon augmenté à 1.1 R/MBtu. La fonction de coût de production pour l'unité 1 devient

$$F_1(P_1) = H_1(P_1) \times 1,1 = 561 + 7,92 P_1 + 0,001562 P_1^2 \text{ R/h}$$

Nous incluons une expression simplifiée des pertes.

$$P_L = 0.00003 P_1^2 + 0.00009 P_2^2 + 0.00012 P_3^2$$

Cette formule simplifiée des pertes suffira pour montrer les difficultés dans le calcul de la répartition optimale pour laquelle les pertes de transmission sont rendu compte. Notons que les formules réelles des pertes sont plus compliquées que celle utilisée dans cet exemple.

L'application de l'Eq. 3.10 et l'Eq 3.11,

$$\frac{dFi}{dPi} = \lambda \left(1 - \frac{\partial PL}{\partial Pi}\right)$$

devient pour P_1

$$7.92 + 0.003124 P_1 = \lambda [1 - 2 (0.00003) P_1]$$

de même pour P_2 et P_3 ,

$$7.85 + 0.00388 P_2 = \lambda [1 - 2 (0.00009) P_2]$$

$$7.97 + 0.00964 P_3 = \lambda [1 - 2 (0.00012) P_3]$$

et

$$P_1 + P_2 + P_3 - 850 - P_L = 0$$

Nous n'avons plus un ensemble d'équations linéaires comme dans l'Exemple 3A. ceci rend nécessaire un procédé plus complexe de résolution de problème comme suit.

Etape 1 On sélectionne un ensemble des valeurs de départ pour P_1 , P_2 et P_3 dont leur somme est égale à la charge.

Etape 2 On calcul les incréments des pertes $\partial P_L / \partial P_i$ aussi bien que les pertes totales P_L . les incréments des pertes et les pertes totales seront considérées constantes jusqu'à ce que nous revenions à l'étape 2.

Etape 3 On calcule la valeur du λ qui donne P_1 , P_2 , et P_3 dont la somme est égale à la charge totale plus les pertes. C'est maintenant aussi simple que les calculs dans l'exemple 3.1 puisque les équations sont encore linéaires.

Etape 4 On compare P_1 , P_2 , et P_3 de l'étape 3 aux valeurs utilisées au début de l'étape 2, s'il n'y a aucun changement significatif pour toutes les valeurs, nous allons à l'étape 5, autrement nous retournons à l'étape 2.

Etape 5 Fait

Utilisant cette procédure, on obtient

Etape 1 Sélectionnons les valeurs de début P_1 , P_2 , et P_3 comme suit

$$P_1 = 400 \text{ MW}$$

$$P_2 = 300 \text{ MW}$$

$$P_3 = 150 \text{ MW}$$

Etape 2 les incréments des pertes sont :

$$\partial P_L / \partial P_1 = 2 (0.00003) 400 = 0.0240$$

$$\partial P_L / \partial P_2 = 2 (0.00009) 300 = 0.0540$$

$$\partial P_L / \partial P_3 = 2 (0.00012) 150 = 0.0360$$

Les pertes totales sont 15.6 MW

Etape 3 Nous pouvons maintenant calculer λ en utilisant les équations suivantes :

$$7.92 + 0.003124 P_1 = \lambda (1 - 0.0240) = \lambda (0.9760)$$

$$7.85 + 0.00388 P_2 = \lambda (1 - 0.0540) = \lambda (0.9460)$$

$$7.97 + 0.00964 P_3 = \lambda (1 - 0.0360) = \lambda (0.9640)$$

et

$$P_1 + P_2 + P_3 - 850 - 15.6 = P_1 + P_2 + P_3 - 865.6 = 0$$

Ces équations sont maintenant linéaires, ainsi nous pouvons calculer λ directement. les résultats sont :

$$\lambda = 9.5252 \text{ R / MWh}$$

et les puissances produites par les générateurs sont :

$$P_1 = 440.68 \text{ MW}$$

$$P_2 = 299.12 \text{ MW}$$

$$P_3 = 125.77 \text{ MW}$$

Etape 4 puisque ces valeurs de P_1 , P_2 , et P_3 sont tout à fait différent des valeurs de début, nous retournant à l'étape 2.

Etape 2 Les incréments des pertes sont recalculés avec les nouvelles valeurs de génération

$$\partial P_L / \partial P_1 = 2(0.00003)440.68 = 0.0264$$

$$\partial P_L / \partial P_2 = 2(0.00009)299.12 = 0.0538$$

$$\partial P_L / \partial P_3 = 2(0.00012)125.77 = 0.0301$$

les pertes totales sont 15.78 MW.

Etape 3 Les nouvelles incréments des pertes et pertes totales sont incorporées dans les équations, et une nouvelle valeur de λ et P_1 , P_2 , et P_3 sont trouvées pour

$$7.92 + 0.003124 P_1 = \lambda (1 - 0.0264) = \lambda (0.9736)$$

$$7.85 + 0.00388 P_2 = \lambda (1 - 0.0538) = \lambda (0.9462)$$

$$7.97 + 0.00964 P_3 = \lambda (1 - 0.0301) = \lambda (0.9699)$$

et

$$P_1 + P_2 + P_3 - 850 - 15.78 = P_1 + P_2 + P_3 - 865.78 = 0$$

Ce qu'il donne

$$\lambda = 9.5275 \text{ R/MWh}$$

et

$$P_1 = 433.94 \text{ MW}$$

$$P_2 = 300.11 \text{ MW}$$

$$P_3 = 131.74 \text{ MW}$$

⋮

Table 3.1 récapitule le processus itératif employé pour résoudre ce problème.

Itération	P_1 (MW)	P_2 (MW)	P_3 (MW)	Pertes (MW)	λ (R/MWh)
début	400.00	300.00	150.00	15.60	9.5252
1	440.68	299.12	125.77	15.78	9.5275
2	433.94	300.11	131.74	15.84	9.5285
3	435.87	299.94	130.42	15.83	9.5283
4	435.13	299.99	130.71	15.83	9.5284

Table 3.1 le processus itérative utilisé pour résoudre l'Exemple 3.3.

III. 4.2 Méthode Graphique

Nous pouvons approcher la solution de problème la répartition économique de Charge dans un système d'énergie électrique à centrales thermiques en négligent les pertes. en considérant la technique graphique montrée dans la Figure 3.3.

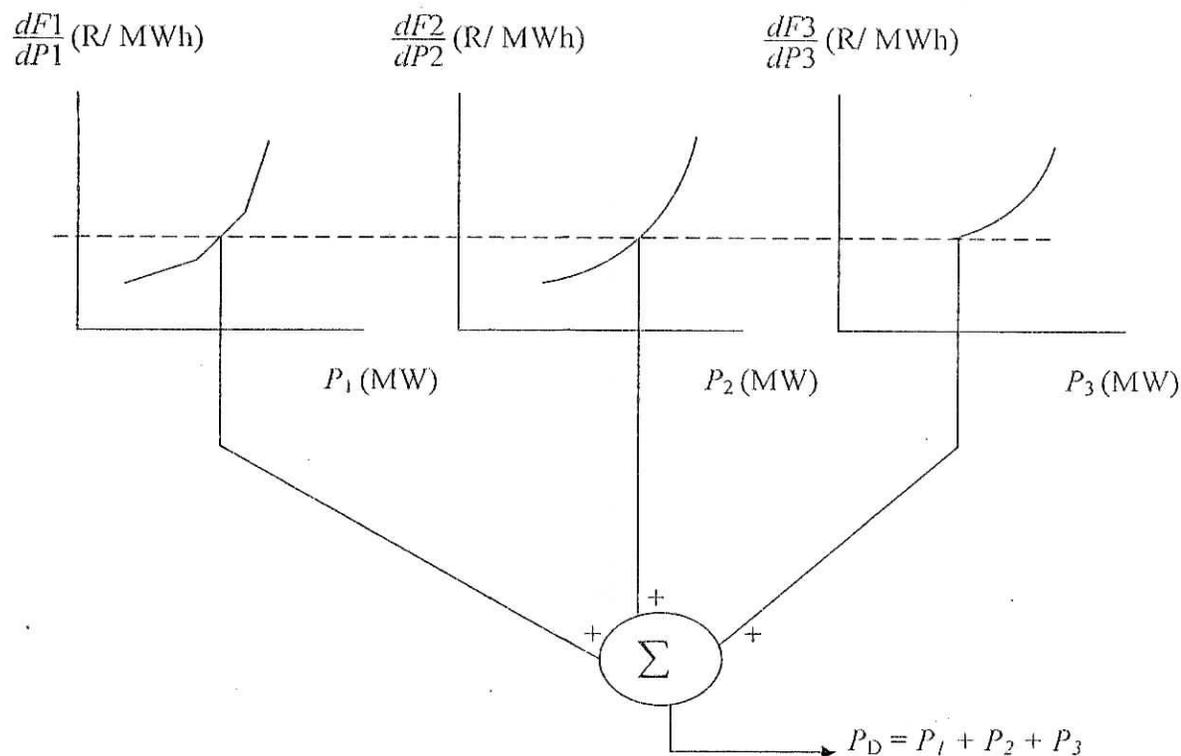


Figure. 3.3 La Solution graphique de la Répartition Economique.

Supposons que nous avons eu un système à trois-unités et nous souhaitons de trouver le point de fonctionnement économique optimal. Une approche consiste à tracer les caractéristiques de l'incrément de coût pour chacune de ces trois unités sur le même graphique, afin d'établir les points de Fonctionnement de chacune de ces trois unités tels que nous avons le coût minimum et en même temps satisfaire la demande indiquée, nous pourrions employer ce papier graphique et une règle pour trouver la solution. C'est-à-dire, nous pourrions assumer un niveau d'incrément de coût (λ) et trouver les sorties de puissance de chacune des trois unités pour cette valeur de l'incrément de coût.

Naturellement, notre première estimation est probablement incorrecte. Si nous avons assumé une valeur de l'incrément de coût telle que la somme des puissances produites est

aussi base, nous devons augmenter la valeur de λ et essayer une autre solution, et nous observons bien sur les limites de fonctionnement sur chacune des unités pour chaque valeur de λ .

avec deux solutions, nous pouvons extrapoler (ou interpoler) les deux solutions pour obtenir de plus près de la valeur désirée de la puissance demandée totale, tel qu'il est montré sur la Figure 3.4

$$\text{Erreur : } e = (P_1 + P_2 + P_3) - P_D$$

(MW)

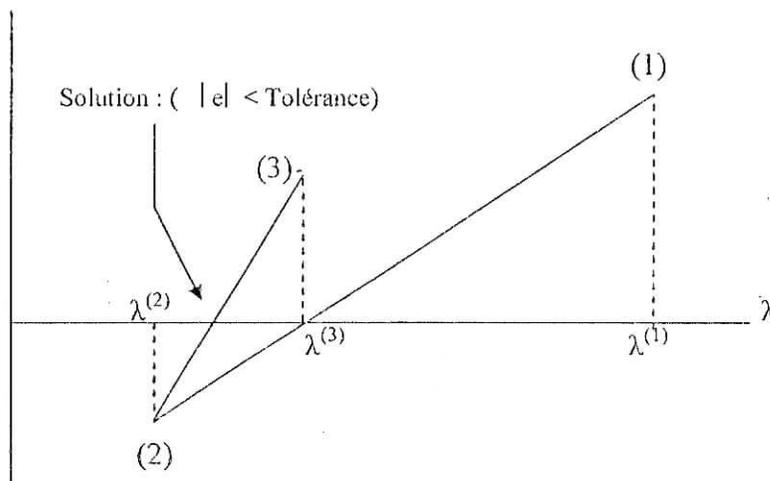


Figure. 3.4 Projections de lambda.

En pratique, pour calculer la répartition des charges qui donne à la fois la puissance totale appelée et le coût de production minimal, on utilise un jeu de réglettes graduées en puissance et en coûts et qui coulissent parallèlement.

Il y a encore quelques années on utilisait de telles réglettes dans les dispatchings d'E.D.F., et pour en faciliter la manœuvre elles étaient installées sur des tambours.

Nous pouvons prolonger cette approche dans le secteur des algorithmes de calcul.

III. 4.3 Méthode Itérative de Lambda Variable

Ce même procédé peut être adopté pour une exécution d'ordinateur comme représenté sur la Figure 3.5

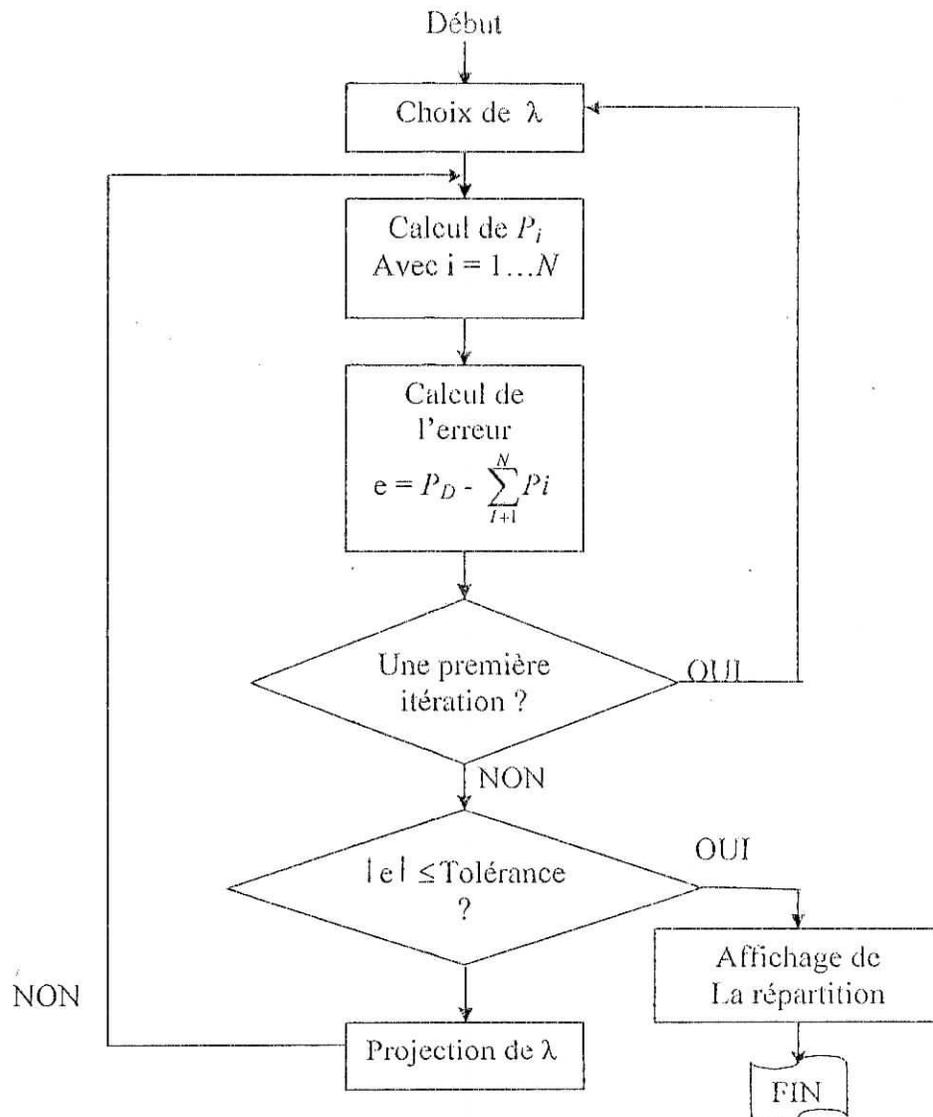


Figure. 3.5 Schéma fonctionnel de la méthode Itérative de Lambda variable.

C'est-à-dire, nous établirons maintenant un ensemble de règles logiques qui nous permettraient d'accomplir le même objectif que nous avons fait juste avec la règle et le papier graphique.

Nous pourrions, par exemple, stocker des tables des données (qui montreraient la valeur de la puissances produites totales pour différents niveaux de l'incrément de coût et la combinaisons des unités) dans l'ordinateur et interpoler entre les points stockés de puissance pour trouver la puissance produite exacte pour une valeur indiquée du l'incrément de coût. un autre approche serait de développer une fonction analytique pour la puissance produite en fonction du l'incrément de coût, de stocker cette fonction (ou ses coefficients) dans l'ordinateur, et employer ceci pour établir la puissance générée par chacune des différentes unités.

Ce procédé est un calcul de type itératif, et nous devons établir des règles d'arrêtes. Deux formes générales des règles d'arrêtes semblent appropriées pour cette application. La première est montrée sur la Figure 3.5 et elle est essentiellement une règle basée sur l'idée de trouver le point de fonctionnement approprié en dessous d'une tolérance indiquée. L'autre, non représenté sur la Figure 3.5, implique de compter le temps d'exécution d'une boucle itérative et de s'arrêter quand le temps maximum est excédé.

Le procédé itératif de lambda variable converge très rapidement pour ce type particulier de problème d'optimisation. Le procédé informatique réel est légèrement plus complexe que cela indiqué sur la Figure 3.5 puisqu'il est nécessaire d'observer les limites de fonctionnement sur chacune des unités pendant le calcul.

Exemple 3.4

Supposons que nous souhaitons employer des fonctions cubiques pour représenter les caractéristiques d'entrée-sortie des installations de production comme suit.

$$H \text{ (MBtu/h)} = A + BP + CP^2 + DP^3 \quad (P \text{ en MW})$$

Pour les trois unités, on trouve le programme optimum en utilisant la méthode itérative de lambda.

	A	B	C	D
Unité 1	749,55	6,95	$9,68 \times 10^{-4}$	$1,27 \times 10^{-7}$
Unité 2	1285,0	7,051	$7,375 \times 10^{-4}$	$6,453 \times 10^{-8}$
Unité 3	1531,0	6,531	$1,04 \times 10^{-3}$	$9,98 \times 10^{-8}$

Table 3.2

Supposons que le coût de carburant est 1,0 R/MBtu pour chaque unité et les limites de fonctionnement sont comme suit.

$$320 \text{ MW} \leq P_1 \leq 800 \text{ MW}$$

$$300 \text{ MW} \leq P_2 \leq 1200 \text{ MW}$$

$$275 \text{ MW} \leq P_3 \leq 1100 \text{ MW}$$

Deux calculs d'échantillon sont montrés, tous les deux emploient l'organigramme de la Figure 3.3. Dans ce calcul, la valeur de λ dans la deuxième itération est toujours placée à 10 % au-dessus ou au-dessous de la valeur de départ selon le signe de l'erreur; pour les itérations restantes, le lambda est projeté comme sur la Figure 3.5.

Le premier exemple montre l'avantage de commencer par une valeur de λ près de la valeur optimale.

$$P_D = 2500 \text{ MW}$$

$$\lambda_{\text{début}} = 8,0 \text{ R/MWh}$$

Itération	Lambda	Génération totale (MW)	P_1	P_2	P_3
1	8,0000	1731,5	494,3	596,7	640,6
2	8,8000	2795,0	800,0	1043,0	952,0
3	8,5781	2526,0	734,7	923,4	867,9
4	8,5566	2497,5	726,1	911,7	859,7
5	8,5586	2500,0	726,9	912,7	860,4

Table 3.3

Le deuxième exemple montre les problèmes oscillants qui peuvent être produits avec une approche itérative de lambda.

$$P_D = 2500 \text{ MW}$$

$$\lambda_{\text{début}} = 10,0 \text{ R/MWh}$$

Itération	Lambda	Génération totale (MW)	P_1	P_2	P_3
1	10,0000	3100,0	800,0	1200,0	1100,0
2	9,0000	2974,8	800,0	1148,3	1026,5
3	5,2068	895,0	320,0	300,0	275,0
4	8,1340	1920,6	551,7	674,5	694,4
5	9,7878	3100,0	800,0	1200,0	1100,0
6	8,9465	2927,0	800,0	1120,3	1006,7
7	6,8692	895,0	320,0	300,0	275,0
8	8,5099	2435,0	707,3	886,1	841,7
9	8,5791	2527,4	735,1	924,0	868,3
10	8,5586	2500,1	726,9	912,8	860,4

Table 3.4

III. 4.4 Méthode du Gradient de Premier Ordre

Beaucoup de procédures d'optimisation ne sont rien plus qu'une énumération ou recherche dirigée, comme est le cas avec les méthodes qui sont connues comme *méthodes de gradient*. Ces méthodes appropriées bien au type de problèmes où il y a beaucoup de compréhension physique de la nature du problème et du comportement de la fonction objective car les variables indépendantes sont ajustées. Les méthodes de gradient ont une caractéristique souhaitable très forte. C'est-à-dire, les techniques de recherche de gradient toujours commencent avec une solution faisable et recherchent la solution optimale le long d'une trajectoire qui maintient une solution faisable à tout moment. Par " la solution faisable " est signifié une solution dans laquelle tous les conditions de contrainte sont réunies.

Il peut y avoir d'autres méthodes de recherche de gradient qui commencent à partir des solutions infaisables et recherchent les solutions qui sont optimales et faisables. Cependant, les techniques de recherche de gradient les plus généralement utilisées fournissent toujours des réponses faisables.

C'est une caractéristique très souhaitable puisque le procédé du calcul par ordinateur peut être interrompu à tout moment et le point de solution le plus récent sera toujours un point raisonnable de fonctionnement dont pour faire l'utilisation. Si la méthode est telle que la fonction objective change de façon monotone pendant que chaque nouveau point de fonctionnement est établi, alors la solution la plus récente est la meilleure trouvée à cette étape particulière du calcul.

L'inconvénient de ce type de procédé d'optimisation est qu'il n'y a aucune " règle d'arrêt " claire. C'est-à-dire, l'établissement " du point optimum est difficile". Habituellement, on permet à la méthode de recherche de gradient de continuer jusqu'à ce que le procédé de recherche ne puisse trouver aucun gain " significatif " additionnel dans la fonction objective ou jusqu'à ce que le nombre d'itérations a dépassé une certaine valeur raisonnablement élevée.

Pour commencer la technique de gradient, supposent que le système d'énergie fonction à un point faisable de fonctionnement. C'est-à-dire, la somme des puissances produites des unités est égale à la charge demandée.

Après, supposons que les puissances générées de chacune de ces unités sont changées par une certaine petite quantité et qu'un nouveau point faisable de fonctionnement est trouvé.

Le développement en série de Taylor de la fonction objective de l'Eq. 3.3 autour du point de fonctionnement faisable initial donne L'équation 3.17

$$\begin{aligned}
 F_T + \Delta F_T &= F_1(P_1) + F_2(P_2) + \dots + F_N(P_N) + \\
 &+ \frac{dF_1}{dP_1} \Delta P_1 + \frac{dF_2}{dP_2} \Delta P_2 + \dots + \frac{dF_N}{dP_N} \Delta P_N + \\
 &+ \frac{1}{2} \left(\frac{d^2 F_1}{dP_1^2} [\Delta P_1]^2 + \frac{d^2 F_2}{dP_2^2} [\Delta P_2]^2 + \dots + \frac{d^2 F_N}{dP_N^2} [\Delta P_N]^2 + \dots \right) + \dots \quad (3.17)
 \end{aligned}$$

Des termes sont montrées dans ce développement en série de Taylor jusqu' à et y compris le deuxième ordre, au premier ordre le changement de la fonction objective est montré dans Eq. 3.18.

$$\Delta F_T = \frac{dF_1}{dP_1} \Delta P_1 + \frac{dF_2}{dP_2} \Delta P_2 + \dots + \frac{dF_N}{dP_N} \Delta P_N \quad (3.18)$$

Nous avons obtenu cette relation en supposant que toutes les termes de second ordre et plus peuvent être négligées relativement aux termes de premier ordre et en soustrayant les frais d'exploitation initiaux de l'équation perturbée.

après, considérons l'équation de contrainte (Eq. 3.4) et permettons à chacune des puissances d'être perturbé d'une certaine petite quantité. L'équation de contrainte est comme suit.

$$\sum_{i=1}^N \Delta P_i = 0 \quad (3.19)$$

Dans cette relation ci-contre les valeurs initiales ont été soustraites de sorte que l'équation de contrainte maintenant déclare simplement que la somme des changements de toutes les puissances produites doit être égale à zéro. Cette équation de contrainte enlève un degré de liberté du problème de sorte qu'au moins une unité doit être choisie comme une variable dépendante. Appelons cette unité dépendante la $x^{ième}$ unité. Dans ce cas, le changement du puissance produite par cette unité dépendante est égale au négative du la somme des changements des puissances produites par les $N-1$ unités restantes comme montré dans l'Eq. 3.20.

$$\Delta P_x = - \sum_{i \neq x}^N \Delta P_i \quad (3.20)$$

Ces deux équations (Eqs 3.19 et 3.20) peuvent être combinées pour établir une équation qui donne le changement de la fonction objective, F_T en fonction du changement du la puissance produite de chacune des $N-1$ machines indépendantes. Ceci est montré dans Eq. 3.21

$$\Delta F_T = \sum_{i \neq x} \left[\frac{dF_i}{dP_i} - \frac{dF_x}{dP_x} \right] \Delta P_i = \sum_{i \neq x} \frac{\partial F_T}{\partial P_i} \Delta P_i \quad (3.21)$$

La procédure d'utilisation de cette équation est indiquée sur la Figure 3.6, qui est un diagramme fonctionnel de la méthode de recherche de gradient de premier ordre.

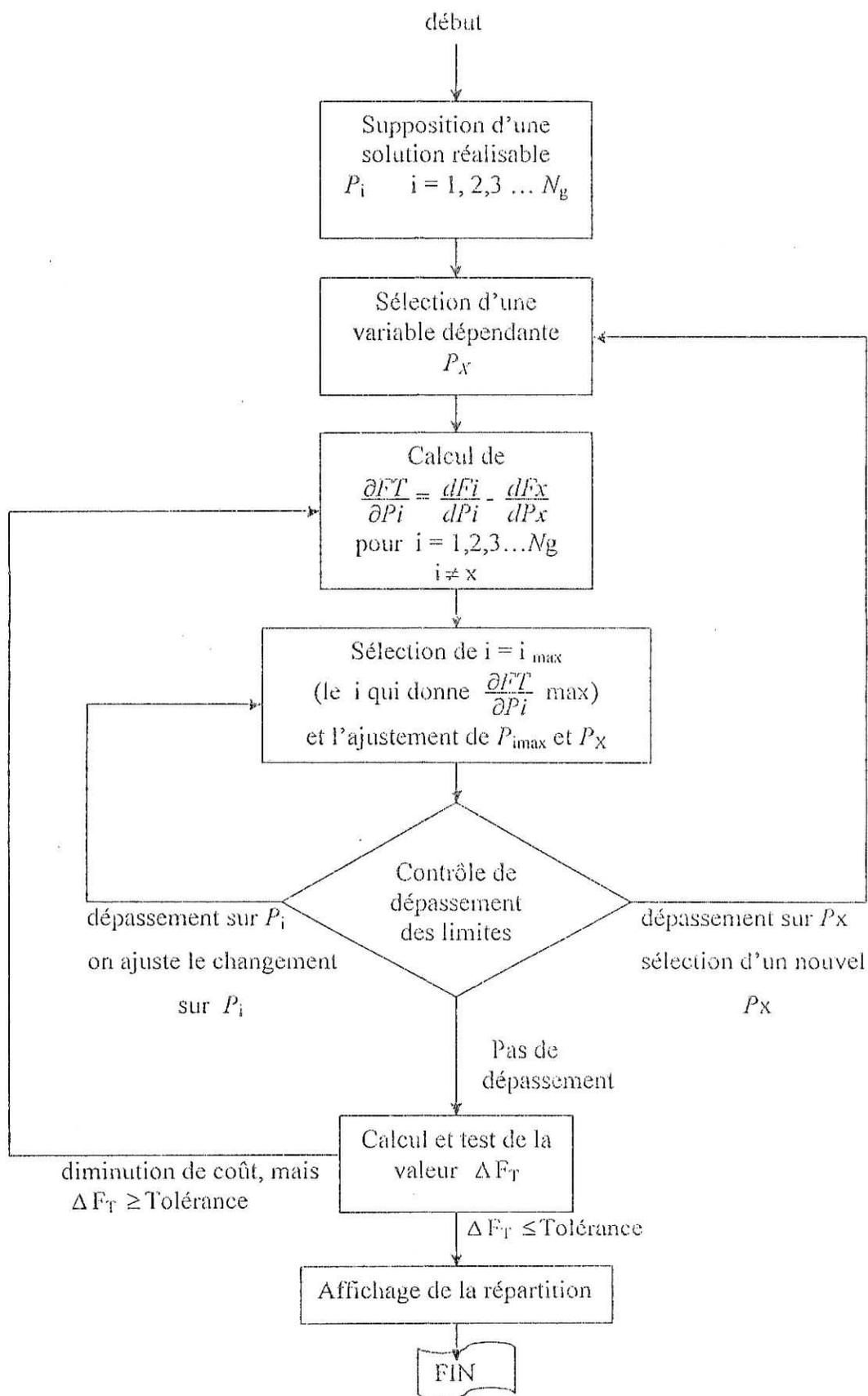


Figure. 3.6 La Répartition Economique de Charge par la Méthode de Gradient de Premier Ordre.

La technique débute avec n'importe quelle solution faisable. C'est-à-dire, une solution qui satisfait l'équation de contrainte (Eq. 3.4). Après, les coefficients dans Eq. 3.21 sont évalués pour ce point de fonctionnement après que la machine dépendante, x a été choisie. Ceci a comme conséquence l'expression numérique de Eq. 3.21, et les coefficients de chacune des $N-1$ puissances variables indiquent le gain dans la fonction objective qui peut être réalisée pour un mouvement d'unité dans la puissance de cette machine particulière. Évidemment, la machine qui gagne la plus grand de la fonction de coût d'exploitation est celle à déplacer d'abord. Cette unité peut être obtenue en trouvant le maximum des valeurs absolues des divers coefficients des changements de la puissance produite.

Quand un mouvement est choisi, la quantité par laquelle est déplacer la puissance doit être vérifiée soigneusement de sorte que ni l'unité indépendante (la $i^{\text{ième}}$ unité) ni l'unité dépendante (la $x^{\text{ième}}$ unité) est déplacée au delà de ses limites de fonctionnement. Les deux unités doivent être déplacées d'une façon égale et opposée de sorte que le nouveau point de fonctionnement soit encore faisable.

Comme la Figure 3.6 montre, les points de fonctionnement devraient être vérifiés après chaque étape, ou de préférence avant chaque étape est fait de manière permanente. Si l'unité dépendante est soulevée contre une limite de fonctionnement ou une limite de fonctionnement pour l'unité dépendante est violée, alors il est temps de retourner à cette étape du calcul où l'unité dépendante est choisie. Ce procédé est tout à fait simple et franc.

Il exige, cependant, un grand nombre d'itérations à venir à une convergence satisfaisante.

Exemple 3.5

Encore, en prenant les données de l'exemple 3.3, résolvons pour une répartition économique de génération par la technique de gradient de premier ordre représentée sur la Figure 3.6. La recherche sera commencée comme suit.

$$\left. \begin{array}{l} P_1 = 400 \text{ MW} \\ P_2 = 300 \text{ MW} \\ P_3 = 150 \text{ MW} \end{array} \right\} \text{Itération 1}$$

Nous pouvons utiliser ce point de départ puisqu'il rencontre la condition

$P_1 + P_2 + P_3 = 850 \text{ MW}$. La variable dépendante, x sera l'unité 3 de puissance générée P_3 . Alors

$$\Delta F_T = \left[\frac{dF_1}{dP_1} - \frac{dF_3}{dP_3} \right] \Delta P_1 + \left[\frac{dF_2}{dP_2} - \frac{dF_3}{dP_3} \right] \Delta P_2$$

$$= (-0.2464) \Delta P_1 + (-0.4020) \Delta P_2$$

et

$$F_T = 8200.47 \text{ R/h}$$

Puisque le plus grand coefficient apparaît avec le ΔP_2 , nous relèverons P_2 . C'est-à-dire, nous souhaitons diminuer F (ΔF négatif), ainsi P_2 croissant (ΔP_2 positif) réalisera ceci puisque son coefficient est négatif. Les conditions de la prochaine itération sont (après avoir augmenté P_2 par 50 MW et avoir diminué P_3 de 50 MW)

$$\left. \begin{array}{l} P_1 = 400 \text{ MW} \\ P_2 = 350 \text{ MW} \\ P_3 = 100 \text{ MW} \end{array} \right\} \text{Itération 2}$$

Ce qu'il donne

$$\Delta F_T = (0.2356) \Delta P_1 + (0.2740) \Delta P_2$$

$$F_T = 8197.27 \text{ R/h}$$

Les itérations successives sont montrées dans le tableau suivant.

Table 3.5

Itération	P_1 (MW)	P_2 (MW)	P_3 (MW)	F_T (R/h)	ΔF_T	ΔP_1 (MW)	ΔP_2 (MW)	ΔP_3 (MW)
1	400	300	150	8200.47	$(-0.2464) \Delta P_1 +$ $(-0.4020) \Delta P_2$	0	+50	-50
2	400	350	100	8197.27	$(0.2356) \Delta P_1 +$ $(0.2740) \Delta P_2$	0	-25	+25
3	400	325	125	8194.65	$(-0.0054) \Delta P_1 +$ $(-0.064) \Delta P_2$	0	+12.5	-12.5
4	400	337.5	112.5	8194.90	$(0.1151) \Delta P_1 +$ $(0.105) \Delta P_2$	-10	0	+10
5	390	337.5	122.5	8194.38	$(-0.01254) \Delta P_1 +$ $(-0.00861) \Delta P_2$	+5	0	-5
6	395	337.5	117.5	8194.48	$(0.05128) \Delta P_1 +$ $(0.0568) \Delta P_2$	0	-2.5	+2.5
7	395	335	120	8194.38	$(0.0272) \Delta P_1 +$ $(0.023) \Delta P_2$			

III. 4.5 Méthode du Gradient de Second Ordre

Nous retournons au problème de trouver le programme économique pour un niveau de demande simple donné. La méthode de gradient de premier ordre montrée précédemment peut être raffinée et rendue plus efficace en employant les termes de second ordre de développement en série de Taylor de coût Total de génération donnée dans Eq. 3.17.

$$\begin{aligned}
 F_T + \Delta F_T &= F_1(P_1) + F_2(P_2) + \dots + F_N(P_N) + \\
 &+ \frac{dF_1}{dP_1} \Delta P_1 + \frac{dF_2}{dP_2} \Delta P_2 + \dots + \frac{dF_N}{dP_N} \Delta P_N + \\
 &+ \frac{1}{2} \left(\frac{d^2 F_1}{dP_1^2} [\Delta P_1]^2 + \frac{d^2 F_2}{dP_2^2} [\Delta P_2]^2 + \dots + \frac{d^2 F_N}{dP_N^2} [\Delta P_N]^2 + \dots \right) + \dots \quad (3.17)
 \end{aligned}$$

Rappelons encore qu'il n'y a aucun mélange dans les deuxièmes dérivés partiels dans le cas habituel. C'est-à-dire, le deuxième dérivé du coût de production d'une unité donnée dépend normalement seulement explicitement de la puissance produite par l'unité, elle-même. C'est-à-dire,

$$\frac{\partial^2 F_i}{\partial P_i \partial P_j} = 0$$

pour $i \neq j$, aussi, la contrainte exigeant l'égalité de la somme des puissances produites des différentes unités et la demande totale doit être traitée comme dans

Eq 3.19 et Eq 3.20 pour que le programme développé ne change pas la fréquence du système.

Ceux-ci sont

$$\sum_{i=1}^N \Delta P_i = 0 \quad (3.19)$$

$$\Delta P_X = - \sum_{i \neq X}^N \Delta P_i \quad (3.20)$$

Si nous substituons la relation de l'Eq. 3.20 dans Eq. 3.17 gardant les termes de premier et de second ordre impliquant les puissances perturbées, alors nous obtenons

$$\Delta F_T = \sum_{i \neq x} \left(\frac{dF_i}{dP_i} - \frac{dF_x}{dP_x} \right) \Delta P_i + \frac{1}{2} \left\{ \frac{d^2 F_1}{dP_1^2} (\Delta P_1)^2 + \frac{d^2 F_2}{dP_2^2} (\Delta P_2)^2 + \dots + \frac{d^2 F_x}{dP_x^2} (\Delta P_1^2 + \Delta P_2^2 + \dots + 2 \Delta P_1 \Delta P_2 + 2 \Delta P_1 \Delta P_3 + \dots) \right\} \quad (3.22)$$

Le changement dans le coût total de fonctionnement ΔF_T , peut être traité en utilisant des méthodes ordinaires de calcul puisque c'est une fonction des $N-1$ changements indépendants des niveaux des puissances produites, ΔP_i , il ne y a aucune autre contrainte autre que les limites des niveaux de production des installations, que nous ignorerons actuellement.

le meilleur point de fonctionnement, sera réalisé quand le dérivé partiel du ΔF_T par rapport a chaque variable indépendante, ΔP_i , est zéro. C'est à dire, les dérivés partiels $\frac{\partial \Delta F_T}{\partial \Delta P_i}$ doit être zéro pour tout $i, i \neq x$. ces dérivés ont comme conséquence un ensemble d'équations simultanées.

$$\begin{aligned} \frac{\partial \Delta F_T}{\partial \Delta P_1} = 0 &= \left(\frac{dF_1}{dP_1} - \frac{dF_x}{dP_x} \right) + \frac{d^2 F_1}{dP_1^2} \Delta P_1 + \frac{d^2 F_x}{dP_x^2} \sum_{i \neq x} \Delta P_i \\ \frac{\partial \Delta F_T}{\partial \Delta P_2} = 0 &= \left(\frac{dF_2}{dP_2} - \frac{dF_x}{dP_x} \right) + \frac{d^2 F_2}{dP_2^2} \Delta P_2 + \frac{d^2 F_x}{dP_x^2} \sum_{i \neq x} \Delta P_i \\ &\vdots \end{aligned} \quad (3.23)$$

la présentation peut être rendue légèrement plus claire en laissant

$$F_i' = \frac{dF_i}{dP_i} \quad \text{et} \quad F_i'' = \frac{d^2 F_i}{dP_i^2} \quad (3.24)$$

tous les deux soient évalués pour le point initial de fonctionnement. Alors les $N-1$ équations simultanées peut être écrites sous une forme matricielle comme suit

$$\begin{bmatrix} F_1'' + F_x'' & F_x'' & F_x'' \dots \\ F_x'' & F_2'' + F_x'' & F_x'' \dots \\ F_x'' & F_x'' & F_2'' + F_x'' \dots \\ \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} \Delta P_1 \\ \Delta P_2 \\ \Delta P_3 \\ \vdots \end{bmatrix} = - \begin{bmatrix} F_1' - F_x' \\ F_2' - F_x' \\ F_3' - F_x' \\ \vdots \end{bmatrix} \quad (3.25)$$

La méthode de recherche de gradient de seconde-order rapporte les composants multidimensionnels du gradient de la fonction de coût par les étapes suivantes.

1. commencez par une solution faisable et calculez les éléments des équations de la matrice.
2. inversez la matrice pour le calcul des montants décalent chaque générateur.
3. vérifiez que la solution ne viole aucune contrainte.
4. contrôlez les valeurs du vecteur $(F' - Fx')$ au nouveau point de fonctionnement pour voir que tout les incréments des coûts sont égaux dans les limites de fonctionnement de chaque unité.

Si les incréments des coûts ne sont pas égaux, le procédé est répété.

Avec une fonction objective quadratique, la méthode de second-order convergera dans une étape supposant une situation sans contrainte. les exemples utilisés en ce chapitre donnent l'impression que la méthode de second order est une qui converge toujours dans une simple étape et ne semble avoir aucun des problèmes informatiques liés aux méthodes de premier ordre, c'est pas vrai !

Ces méthodes toutes ont leurs difficultés. D'une façon générale, en employant des techniques de gradient c'est les contraintes qui provoquent les problèmes.(dans quelques autres techniques d'optimisation il y a plusieurs contraintes semblables entre eux qu'ils sont une bénédiction et servent à réduire la dimensionnalité du problème).

Exemple 3.6

Nous résoudrons le même problème de répartition (que l'Exemple 3.3) avec les mêmes conditions de départ que dans l'Exemple 3.5 en utilisant la méthode de gradient du second-order. Les conditions de départ sont

$$P_1 = 400 \text{ MW}$$

$$P_2 = 300 \text{ MW}$$

$$P_3 = 150 \text{ MW}$$

$$\begin{aligned} F_1' &= 9.1696 \text{ R/h} & F_1'' &= 0.003124 \text{ (R/MWh) / MW} \\ F_2' &= 9.0.140 \text{ R/h} & F_2'' &= 0.00388 \text{ (R/MWh) / MW} \\ F_3' &= 9.4160 \text{ R/h} & F_3'' &= 0.00964 \text{ (R/MWh) / MW} \end{aligned}$$

Utilisant alors l'Eq.3.23,

$$\begin{bmatrix} F_1'' + F_3'' & F_3'' \\ F_3'' & F_2'' + F_3'' \end{bmatrix} \begin{bmatrix} \Delta P1 \\ \Delta P2 \end{bmatrix} = - \begin{bmatrix} F_1' - F_3' \\ F_2' - F_3' \end{bmatrix}$$

Ou

$$\begin{bmatrix} 0.012764 & 0.00964 \\ 0.00964 & 0.01352 \end{bmatrix} \begin{bmatrix} \Delta P1 \\ \Delta P2 \end{bmatrix} = - \begin{bmatrix} -0.2464 \\ -0.4020 \end{bmatrix}$$

La solution est

$$\begin{bmatrix} \Delta P1 \\ \Delta P2 \end{bmatrix} = \begin{bmatrix} -6.8301 \\ 34.603 \end{bmatrix}$$

Puis

$$P_1 = 400 - 6.83 = 393.17 \text{ MW}$$

$$P_2 = 300 + 34.6 = 334.6 \text{ MW}$$

Et par définition

$$P_3 = 850 - P_1 - P_2 = 122.23 \text{ MW.}$$

III. 4.6 Méthode de Point Bas et les Facteurs de Participation

Cette méthode suppose que le problème de la répartition économique doit être résolu à plusieurs reprises en déplaçant les générateurs d'un programme économiquement optimum à l'autre pendant que la charge change par une petite quantité raisonnablement. Nous commençons à partir d'un programme donné *—le point bas*. Après, le programmeur assume un changement de charge et étudie de combien chaque unité de production à besoin d'être

déplacée (c.-à-d., " participant " au changement de charge) pour que la nouvelle charge soit servie au point de fonctionnement le plus économique.

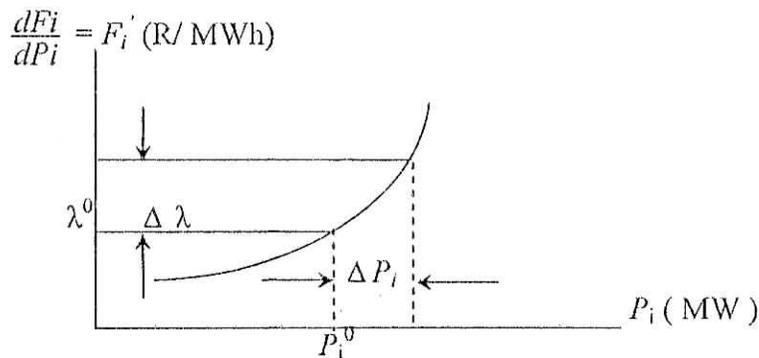


Figure. 3.7 La courbe de l'incrément de coût de la $i^{\text{ème}}$ unité (rapport de $\Delta\lambda$ et de ΔP_i)

Supposons que les premiers et deuxièmes dérivés de la fonction coût par rapport aux puissances produites sont disponibles (c.-à-d., F_i' et F_i'' existent). La courbe de l'incrément de coût de la $i^{\text{ème}}$ unité est donnée sur la Figure 3.7. Pendant que la charge d'unité est changée d'une quantité ΔP_i , l'incrément de coût de système se déplace du λ^0 à $\lambda^0 + \Delta\lambda$.

Pour un petit changement de puissance produite sur cette simple unité, on peut écrire

$$\Delta\lambda_i = \Delta\lambda \cong F_i''(P_i^0) \Delta P_i \quad (3.26)$$

Cela est vrai pour chacune des N unités de système de sorte que,

$$\Delta P_1 = \frac{\Delta\lambda}{F_1''}$$

$$\Delta P_2 = \frac{\Delta\lambda}{F_2''}$$

$$\vdots$$

$$\Delta P_N = \frac{\Delta\lambda}{F_N''}$$

Le changement total de la génération (égale changement total de la demande de système) est, naturellement, égale à la somme des différents changements d'unité. Laissons P_D être toute la demande sur les générateurs (où $P_D = P_{\text{consommateurs}} + \text{pertes}$), alors

$$\Delta P_D = \Delta P_1 + \Delta P_2 + \dots + \Delta P_N = \Delta \lambda \sum_i \left(\frac{1}{F_i^n} \right) \quad (3.27)$$

Les équations, 3.24 et 3.25, peut être employées pour trouver le facteur de participation pour chaque unité comme suit.

$$\left(\frac{\Delta P_i}{\Delta P_D} \right) = \frac{\left(\frac{1}{F_i^n} \right)}{\sum_i \left(\frac{1}{F_i^n} \right)} \quad (3.28)$$

L'exécution par ordinateur d'un tel arrangement de répartition économique est direct. Elle pourrait être faite par Provision des tables de valeurs de F_i^n comme une fonction des niveaux de charge et la conception d'un arrangement simple pour prendre la charge existante plus l'augmentation projetée à recherchent ces données et calculent les facteurs.

Exemple 3.7

Commençants à partir de la solution économique optimale,

$$P_1 = 393,2 \text{ MW}$$

$$P_2 = 334,6 \text{ MW}$$

$$P_3 = 122,2 \text{ MW}$$

Puisque elle rencontre toutes les contraintes, c.-à-d., chaque unité est dans ses limites de production maximales et minimales autorisées, et la somme des puissances produites par les trois unités égale à la production total désirée de 850 MW.

Employent la méthode de facteur de participation pour calculer la répartition économique pour une charge totale de 900 MW.

Utilisant Eq. 3.28,

$$\frac{\Delta P_1}{\Delta P_D} = (0.003124)^{-1} / (0.003124)^{-1} + (0.00388)^{-1} + (0.00964)^{-1} = 0.47$$

De même,

$$\frac{\Delta P_2}{\Delta P_D} = (0.00388)^{-1} / 681.57 = 0.38$$

$$\frac{\Delta P_3}{\Delta P_D} = 103.73 / 681.57 = 0.15$$

La nouvelle valeur de génération est calculée utilisant

$$P_{i \text{ new}} = P_{i \text{ base}} + \left(\frac{\Delta P_i}{\Delta P_D} \right) \Delta P_D \text{ pour } i = 1, 2, 3$$

Alors pour chaque unité

$$P_{1 \text{ new}} = 393.2 + (0.47) (50) = 416.7$$

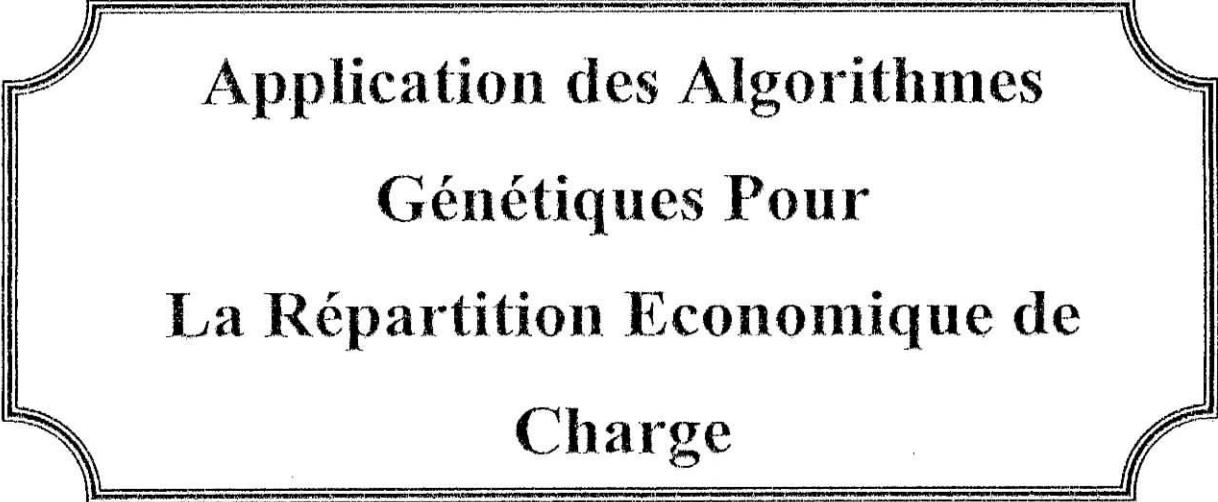
$$P_{2 \text{ new}} = 334.6 + (0.38) (50) = 353.6$$

$$P_{3 \text{ new}} = 122.2 + (0.15) (50) = 129.7$$

Conclusion

Ce chapitre a présenter certaines méthodes conventionnelles appliquées au problème de la repartition économique de charge. Des applications ont été faite sur un même réseau à 3 générateur pour illustrer leur différences.

CHAPITRE IV



**Application des Algorithmes
Génétiques Pour
La Répartition Economique de
Charge**

IV.1 Introduction

Dans ce chapitre nous nous sommes intéressés à l'implémentation d'un algorithme génétique pour la résolution du problème de répartition économique de charge dans un réseau électrique.

En fait, la nécessité et la recherche de trouver et d'appliquer une méthode d'optimisation robuste, le problème de la répartition économique de charge dans un réseau électrique étant caractérisé par un espace de recherche trop grand et une fonction objective très complexe, nous ont mené vers les algorithmes génétiques qui sont un procédé d'optimisation, utilisant un choix aléatoire pour diriger une exploration hautement intelligente dans l'espace de recherche à l'aide d'opérateurs génétiques.

Nous présentons ici une description de l'algorithme génétique adapté à la résolution de ce problème. Notre algorithme détermine chaque composante du vecteur de contrôle U (i.e. le niveau de production nécessaire au niveau de chaque unité de production en service) pour lequel le coût de production total est minimum, ou tout au moins aussi proche que possible du minimum.

IV.2 Algorithme génétique adopté

IV.2.1 Codage des variables du problème

La première étape est de définir et de coder convenablement, clairement et sans ambiguïté les paramètres d'optimisation. Le but de cette application d'AG est de déterminer le vecteur U qui représente la répartition économique de charge. Chaque variante de ce vecteur représente un *individu*, les composantes du vecteur U , P_i , sont les *gènes* de taille $n_{bco}(i)$ chacun. Nous appelons *population* un ensemble d'individus que nous allons faire évoluer.

Le codage binaire a été choisi, pour permettre de traiter des chromosomes de grande précision et ainsi de rendre l'application des opérateurs génétiques simple et directe. De plus, un des avantages du codage binaire est que l'on peut facilement coder toutes sortes de variables : des réels, des entiers, des valeurs booléennes, des chaînes de caractères, etc.

Cela nécessite simplement l'usage de fonctions de codage et décodage pour passer d'une représentation à l'autre.

Rappelons que dans cette étude les N_g variables, P_i , sont supposées réelles. Nous considérons un espace de recherche fini :

$$P_{i, \min} \leq P_i \leq P_{i, \max} \quad \forall i \in [1; N_g]$$

Afin de coder nos variables réelles en binaire, nous discrétisons l'espace de recherche. Ainsi un codage sur 32 bits implique une discrétisation des intervalles en $2^{(32)} - 1 = 4\,294\,967\,294$ valeurs discrètes.

IV.2.2 Espace de recherche

En effet, le problème qu'on cherche à résoudre est un problème d'optimisation avec contrainte égalité.

$$G = P_D - \sum_{i=1}^N P_i = 0$$

Pour faire face au problème numérique posé par cette égalité, nous avons remplacé cette contrainte par l'inégalité

$$\sum_{i=1}^N P_i \geq P_D$$

Pendant, cela, nous a conduit à l'exploration d'un trop grand espace de recherche par rapport à celui qui contient la solution. Il est donc judicieux de limiter cet l'espace de recherche en se basant sur certains aspects pratiques du problème.

En fait, nous savons que la production totale ne doit pas être inférieure à la demande incluant les pertes, P_D , afin d'éviter les problèmes de déstabilisation de la fréquence du réseau. Et elle ne doit pas non plus être trop grande par rapport à P_D car ce cas augmente les pertes du réseau. En se basant sur de telles considérations, nous avons défini l'espace de recherche suivant :

$$0 \leq \sum_{i=1}^N P_i - P_D \leq 0.01 * P_C$$

avec : P_C , Puissance consommée (MW),
 P_L , Pertes de transmission de réseau,
 $P_D = P_C + P_L$, Puissance consommée + Pertes.

Notons que le facteur $0.01 * P_C$ est choisi après plusieurs essais (facteur qui nous a paru le plus adéquat).

IV.2.3 Genèse de la population (Population initiale)

La première étape de l'algorithme est la genèse de la population, c'est-à-dire le choix d'une population de départ que nous allons faire évoluer. On pourrait prendre des individus régulièrement répartis dans l'espace. Néanmoins, une initialisation aléatoire est plus simple à réaliser. Cette population initiale regroupe plusieurs solutions potentielles du problème d'optimisation.

Nous discuterons plus loin la taille 'taipop' de cette population, mais nous pouvons déjà dire qu'elle résultera d'un compromis entre temps de calcul et qualité de la solution.

IV.2.4 Opérateurs génétiques

IV.2.4.1 Adaptation (Evaluation de la fonction objective)

La fonction objective à minimiser est définie par l'équation

$$F_T = \sum_{i=1}^N F_i(P_i) \quad (3.1)$$

Le calcul de cette fonction revient à évaluer le coût de production de chacune des N_g unités de production en service.

IV.2.4.2 Sélection

Nous appelons *génération* la population à un instant t donné. Une fois l'évaluation de la génération effectuée, on opère une sélection à partir de la fonction d'adaptation. Seuls les individus passant l'épreuve de sélection peuvent accéder à la *génération intermédiaire* (*mating pool* en terminologie anglo-saxonne) et s'y reproduire.

En fait, nous avons essayé les deux techniques de sélection les plus utilisées [ref] : sélection par la roue biaisée et sélection par tournoi.

Nous avons constaté qu'on peut éviter l'ensemble des inconvénients induits par la technique de la roue biaisée, (voir paragraphe II.4.1) en utilisant la méthode de :

- **sélection par tournoi** : plusieurs individus sont choisis au hasard et combattent (leur fonction d'adaptation sont comparées) pour accéder à la génération intermédiaire. Le plus adapté l'emporte avec une probabilité, que nous avons prise égale à 1 (une valeur inférieure permet de réduire la pression de sélection si nécessaire). Cette étape est répétée jusqu'à ce que la population intermédiaire soit remplie. Il est tout à fait possible que certains individus participent à plusieurs tournois. S'ils gagnent plusieurs fois, ils auront donc droit d'être copiés plusieurs fois dans la génération intermédiaire, ce qui favorisera la pérennité de leurs gènes.

Les résultats de ce chapitre ont été donc obtenus avec cette dernière méthode.

IV.2.4.3 Croisement

Une fois la génération intermédiaire obtenue, les individus sont aléatoirement répartis en couples. Les chromosomes (ensembles de paramètres) des parents sont alors copiés et recombinaison de façon à former deux descendants possédant des caractéristiques issues des deux parents. On forme ainsi la génération $t+1$.

L'opérateur croisement favorise l'exploration de l'espace de recherche. En effet, le phénomène de croisement est une propriété naturelle de l'ADN. C'est par analogie avec ce dernier qu'ont été conçus les opérateurs de croisement dans les AG. Nous avons testé deux méthodes de croisement classiques : croisement en un point et croisement en plusieurs points.

- **croisement en un point** : on choisit au hasard un point de croisement, pour chaque couple. Notons que le croisement s'effectue directement au niveau binaire. Un chromosome peut donc être coupé au milieu d'un gène. Les chromosomes sont bien sûr généralement beaucoup plus longs.

- **croisement en N_g points** : on choisit au hasard N_g points de croisement répartis uniformément sur les gènes de l'individu, c'est à dire que chaque gène subit un croisement au niveau du point de croisement choisi. Pour cet travail, nous avons utilisé cet opérateur car

il est généralement considéré comme plus efficace que le précédent. Néanmoins nous n'avons pas constaté de différence notable dans la convergence de l'algorithme.

Notons que le croisement est fait avec une probabilité '**probc**'. D'autres formes de croisement existent, du croisement en k points jusqu'au cas limite du croisement uniforme...

IV.2.4.4 Mutation

Nous définissons une *mutation* comme étant l'inversion d'un bit dans un chromosome. Cela revient à modifier aléatoirement la valeur d'un gène de l'individu. Les mutations jouent le rôle de bruit et empêchent le risque d'aboutir à des solutions qui ne seront pas forcément optimales. Autrement dit, les mutations permettent d'assurer une recherche aussi bien globale que locale, selon le poids et le nombre des bits mutés.

Le nombre des bits mutés dépend directement de la probabilité de mutation '**probm**', de la taille de la population et de la longueur des chromosomes des individus (nombre total de bits de la population).

De nombreuses méthodes existent. Souvent la probabilité de mutation '**probm**' est fixée entre 0,001 et 0,01.

La méthode d'*auto-adaptation* des probabilités de mutation énonce que : dans un environnement stable, il est préférable d'avoir un taux de mutation faible, alors que la survie d'une espèce dans un environnement subissant une évolution rapide nécessite un taux de mutation élevé permettant une adaptation rapide. Les taux de mutation d'une espèce dépendent donc de leur environnement.

IV.2.5 Constitution de la nouvelle génération

Une fois les individus ayant subi l'ensemble des opérateurs génétiques, adaptation, sélection, croisement puis mutation, il est tout à fait naturel que certains d'entre eux sortent de l'espace de recherche limité au début de l'exploration.

Pour rester dans le cadre d'une optimisation avec contrainte, une évaluation de l'équation contrainte est faite pour tous ces individus. Seuls les individus vérifiant la contrainte peuvent accéder à la *nouvelle génération* et s'y reproduire.

Le reste de la population est remplacé (aléatoirement) par des individus qui vérifient la contrainte.

IV.3 Compromis exploration / exploitation

Le phénomène le plus prédominant dans une optimisation par les algorithmes génétiques est la fragile opposition entre exploration et exploitation. L'exploration consiste à aller chercher de nouvelles solutions là où on n'a pas encore cherché, alors que l'exploitation consiste à rester le plus proche possible de la meilleure solution déjà rencontrée.

Une bonne méthode d'optimisation devra doser ces deux ingrédients avec un soin tout particulier. En effet, la meilleure piste de recherche peut très bien être toute proche de la meilleure solution trouvée comme elle peut en être très éloignée. Dans les algorithmes génétiques, le dosage exploration / exploitation est pris en charge par la pression de sélection (exercée par l'opérateur de sélection) et par la pression de variation (exercée par les opérateurs de mutation et de croisement). Dans la sélection, la pression détermine le taux de conservation des meilleurs individus. Dans la mutation et le croisement, c'est le taux de variation des chromosomes qui est contrôlé.

Pour assurer le meilleur dosage exploration / exploitation durant la recherche de la répartition économique de charge, nous avons soigneusement choisi la technique de sélection, ainsi que les probabilités de croisement et de mutation.

IV.4 Réglage des paramètres de l'AG

Nous touchons là au délicat problème du réglage des paramètres de l'algorithme. Celui-ci doit être optimisé pour chaque type de problème traité, ce qui a constitué une part importante de notre travail.

Les paramètres d'évolution dans un algorithme génétique sont : la taille de population candidate, le nombre maximum de générations, la probabilité de croisement et la probabilité de mutation.

Trouver théoriquement les valeurs idéales de ces paramètres constitue un défi mathématique insurmontable. Nous avons donc choisi de commencer à partir de valeurs données par la littérature [Ren.9] et [Gol.6] pour les opérateurs classiques et de faire des évolutions expérimentales pour déterminer les meilleurs paramètres possibles.

- **Taille de population candidate 'taipop' :**

Quelle doit être la taille de cette population ? Une population trop petite évoluera probablement vers un optimum local peu intéressant. Une population trop grande permettra une sélection efficace dans toutes les zones prometteuses de l'espace de recherche et d'espérer avoir une bonne distribution des valeurs de la fonction objective, mais sera inutile car le temps de convergence devient excessif. La taille de la population doit être choisie de façon à réaliser un bon compromis entre temps de calcul et qualité du résultat.

Pour le problème qui nous intéressait et avec les moyens de calcul dont nous disposions, nous avons pu constater qu'une population de $20 \cdot N_g$ individus (20 fois le nombre de degrés de liberté) constituait un bon compromis.

Mais il faut être conscient que cette taille de population dépend de la puissance de calcul dont on dispose, des méthodes utilisées (sélection, opérateurs génétiques...), du nombre de variables considérées et de la fonction d'adaptation.

- **Nombre de générations maximums 'Gmax' :**

En ce qui concerne le nombre maximum de générations, il est nécessaire d'avoir un nombre suffisant de boucles générationnelles pour pouvoir parler d'évolution. En effet, la mutation par exemple ne peut fonctionner correctement que si on lui laisse assez de temps. Nous avons choisis un nombre maximum de génération égal à 50.

- **Probabilité de croisement 'probc' :**

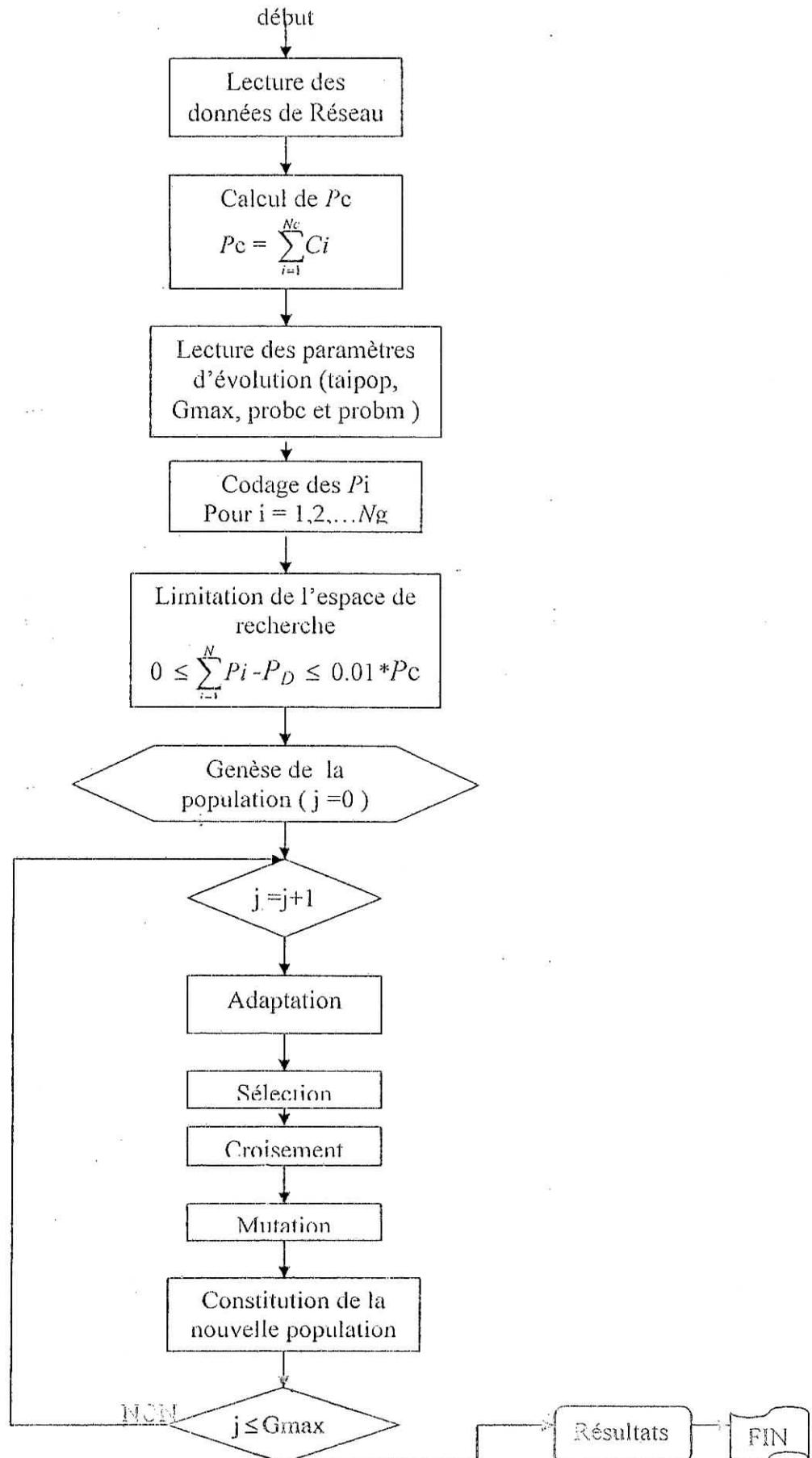
Les résultats de test de notre algorithme génétique ne montrent pas une grande sensibilité à la probabilité de croisement, ce qui nous a poussé à adopter finalement la valeur de 60%, comme probabilité de croisement global.

- **Probabilité de mutation 'probm' :**

Généralement la probabilité de mutation doit être extrêmement faible pour rendre l'optimisation valable. Pour cela et pour prendre en compte la formulation biologique de la méthode d'*auto-adaptation* et l'appliquer à notre cas, nous avons posé la probabilité de mutation égale à 0,001 (valeur qui nous a paru la meilleure après plusieurs essais).

IV.5 Organigramme d'algorithme génétique adopté

L'organigramme suivant présente la procédure de recherche de la répartition économique de charge d'un réseau électrique par un algorithme génétique.



Remarque

En effet, on a programmé deux versions pour ce même organigramme, la seule différence entre ces deux versions est au niveau de la constitution de la nouvelle génération. Pour la première version la constitution de la nouvelle génération se fait par le choix aléatoire des individus à partir de l'espace de recherche, pour la deuxième version il se fait par un choix aléatoire des individus mais à partir de la population initiale.

IV.6 Exemple d'application

On note qu'un format spécifique est utilisé pour introduire les données de réseau au programme.

Le fichier de données de réseau électrique utilisé pour la discussion des résultats de cette partie est montrée à l'annexe. Ce réseau, a trois générateurs, reliés à trois nœuds différents pour servir la charge électrique demandée P_D , mais nous précisons ici que, la généralisation à un nombre quelconque de générateurs ne soulève aucun problème pour le programme.

IV.6.1 Les résultats obtenus avec l'application de l'AG

AG Version N°1

Itération	P_D (MW)	P_1 (MW)	P_2 (MW)	P_3 (MW)	P_{tot} (MW)	Coût (\$/hr)
1	270	77.8	108.2	88.4	274.3	4308.3
2	270	82.5	109.9	81.7	274.1	4303.9

On remarque que cette version donne un bon résultat de point de vue production total et coût de production, mais le problème qui se pose pour cette version est que après deux exécutions différentes on ne retombe jamais sur la même répartition de puissance.

La deuxième version résout ce problème et améliore même les résultats de point de vue production total et coût de production.

AG Version N°2

P_D (MW)	P_1 (MW)	P_2 (MW)	P_3 (MW)	P_{tot} (MW)	Coût (\$/hr)
270	72.1	124.6	76.1	272.9	4274.1

IV.6.2 Les résultats obtenus avec une méthode de Programmation Non Linéaire

Ce même problème est résolu par une méthode de programmation non linéaire, les résultats obtenus sont les suivants :

P_D (MW)	P_1 (MW)	P_2 (MW)	P_3 (MW)	P_{tot} (MW)	Coût (\$/hr)
270	74.27	116.13	81.59	271.99	4245.90

IV.6.3 Comparaison et discussion des résultats

Les résultats obtenus par l'application des AG pour la répartition économique de charge sont très proches de celles obtenus par la méthode de programmation non linéaire.

IV.7 Conclusion

Ces résultats obtenus nous permet de dire que les algorithmes génétiques sont applicables pour la résolution de ce problème Spécifique.



Conclusion

Conclusion

Après un brève passage par des généralités sur les méthodes d'optimisation dans le chapitre 1. Et l'étude approfondie de principe de fonctionnement des algorithmes génétiques au chapitre 2. L'étude de l'applicabilité de ces Algorithmes pour la résolution du problème de réparation économique de charge, est faite a travers la présentation de la formulation générale du problème, ainsi qu'un certain nombre de méthodes conventionnelles utilisées pour le résoudre dans Le troisième chapitre et l'implémentation d'un algorithme génétique adapté présenter au quatrième chapitre.

Nous sommes amenés à ces constatations :

Les Algorithmes Génétiques sont fondamentalement différents des techniques d'optimisation plus conventionnelles selon quatre axes principaux :

- Les AG utilisent un codage des paramètres, et non les paramètres eux mêmes.
- Les AG travaillent sur une population de points, au lieu d'un point unique.
- Les AG n'utilisent que les valeurs de la fonction étudiée, pas sa dérivée, ou une autre connaissance auxiliaire.
- Les AG utilisent des règles de transition probabilistes, et non déterministes.

Et que :

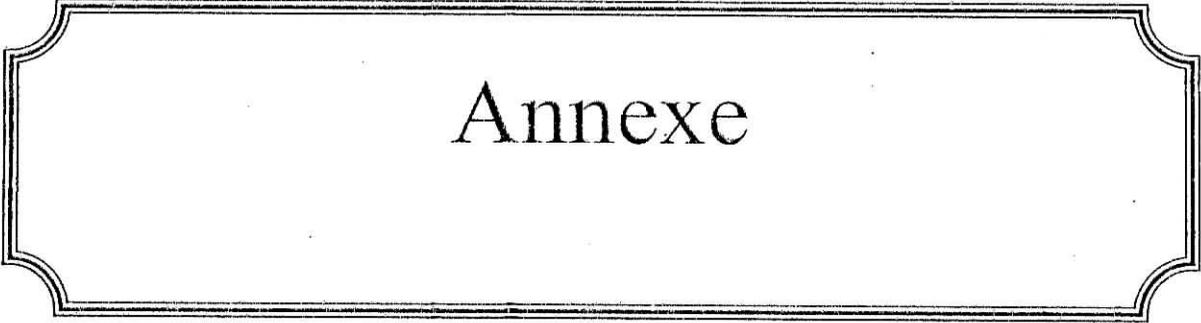
- L'application des algorithmes génétiques pour la réparation de puissance d'un réseau électrique élimine pas mal d'obstacles rencontrés par les méthodes conventionnelles grâce aux différences fondamentales indiquées ci-dessus , surtout si en sache que l'appréciation exacte de la fonction coût est assez complexe et que les informations auxiliaires de cette fonction ne sont pas disponibles ou au moins très difficile à obtenir.

Les résultats obtenus par l'application des AG pour la répartition économique de charge sont très proches de celles obtenus par des méthodes d'optimisation spécifiques a ce problème .Ce qui nous a permet de dire que les algorithmes génétiques sont applicables pour la résolution de ce problème Spécifique.

En fin nous n'oublions pas de rappeler que :

- Chaque problème d'optimisation a une (ou des) méthodes (s) adaptée(s) à son exploration ; cependant selon le problème, les types d'information auxiliaire peuvent varier énormément. Ce n'est donc qu'en refusant d'utiliser cette information que l'on peut espérer développer les procédures générales que nous souhaitons avoir à notre disposition. Mais d'un autre côté, le refus d'utiliser une connaissance spécifique quand elle existe peut limiter la performance d'un algorithmes quand il est comparé avec des méthodes conçues spécialement pour résoudre ce problème.

Pour cela des informations autre que la valeur de la fonction à optimiser sont utilisées, dans ce qu'il est nommé les algorithmes génétiques guidés par la connaissance ; cependant, pour notre étude, nous avons tenir à insister sur l'importance de l'hypothèse de cécité pour la robustesse des *algorithmes génétiques de base*.

A decorative rectangular frame with a double-line border and ornate, curved corners. The word "Annexe" is centered within this frame.

Annexe

Unité 1: Centrale Thermique à Vapeur (Au Charbon)

la puissance maximale autorisée = 600 MW

la puissance minimale autorisée = 150 MW

Courbe de la consommation spécifique- production:

$$H_1 \left(\frac{MBtu}{h} \right) = 510.0 + 7.2 P_1 + 0.00142 P_1^2$$

Unité 2: Centrale Thermique à Vapeur (Au Fuel)

la puissance maximale autorisée = 400 MW

la puissance minimale autorisée = 100 MW

Courbe de la consommation spécifique- production:

$$H_2 \left(\frac{MBtu}{h} \right) = 310.0 + 7.85 P_2 + 0.00194 P_2^2$$

Unité 3: Centrale Thermique à Vapeur (Au Fuel)

la puissance maximale autorisée = 200 MW

la puissance minimale autorisée = 50 MW

Courbe de la consommation spécifique- production:

$$H_3 \left(\frac{MBtu}{h} \right) = 78.0 + 7.97 P_3 + 0.00482 P_3^2$$

```

% Bus Data Format
% 1 bus number (1 to 29997)
% 2 bus type
%   PQ bus           = 1
%   PV bus           = 2
%   reference bus    = 3
%   isolated bus     = 4
% 3 Pd, real power demand (MW)
% 4 Qd, reactive power demand (MVAR)
% 5 Gs, shunt conductance (MW (demanded?) at V = 1.0 p.u.)
% 6 Bs, shunt susceptance (MVAR (injected?) at V = 1.0 p.u.)
% 7 area number, 1-100
% 8 Vm, voltage magnitude (p.u.)
% 9 Va, voltage angle (degrees)
% (-) (bus name)
% 10 baseKV, base voltage (kV)
% 11 zone, loss zone (1-999)
% (+) 12 maxVm, maximum voltage magnitude (p.u.)
% (+) 13 minVm, minimum voltage magnitude (p.u.)
%
% Generator Data Format
% 1 bus number
% (-) (machine identifier, 0-9, A-Z)
% 2 Pg, real power output (MW)
% 3 Qg, reactive power output (MVAR)
% 4 Qmax, maximum reactive power output (MVAR)
% 5 Qmin, minimum reactive power output (MVAR)
% 6 Vg, voltage magnitude setpoint (p.u.)
% (-) (remote controlled bus index)
% 7 mBase, total MVA base of this machine, defaults to baseMVA
% (-) (machine impedance, p.u. on mBase)
% (-) (step up transformer impedance, p.u. on mBase)
% (-) (step up transformer off nominal turns ratio)
% 8 status, 1 - machine in service, 0 - machine out of service
% (-) (% of total VARS to come from this gen in order to hold V at
%      remote bus controlled by several generators)
% 9 Pmax, maximum real power output (MW)
% 10 Pmin, minimum real power output (MW)
%
% Branch Data Format
% 1 f, from bus number
% 2 t, to bus number
% (-) (circuit identifier)
% 3 r, resistance (p.u.)
% 4 x, reactance (p.u.)
% 5 b, total line charging susceptance (p.u.)
% 6 rateA, MVA rating A (long term rating)
% 7 rateB, MVA rating B (short term rating)
% 8 rateC, MVA rating C (emergency rating)
% 9 ratio, transformer off nominal turns ratio ( = 0 for lines )
%   (taps at 'from' bus, impedance at 'to' bus, i.e. ratio = Vf / Vt)
% 10 angle, transformer phase shift angle (degrees)
% (-) (Gf, shunt conductance at from bus p.u.)
% (-) (Bf, shunt susceptance at from bus p.u.)
% (-) (Gt, shunt conductance at to bus p.u.)
% (-) (Bt, shunt susceptance at to bus p.u.)
% 11 initial branch status, 1 - in service, 0 - out of service
%
% (+) Area Data Format
% 1 i, area number
% 2 price_ref_bus, reference bus for that area

```

```

%
% (+) Generator Cost Data Format
%     NOTE: If gen has n rows, then the first n rows of gencost contain
%     the cost for active power produced by the corresponding generators.
%     If gencost has 2*n rows then rows n+1 to 2*n contain the reactive
%     power costs in the same format.
%     1  model, 1 - piecewise linear, 2 - polynomial
%     2  startup, startup cost in US dollars
%     3  shutdown, shutdown cost in US dollars
%     4  n, number of cost coefficients to follow for polynomial
%     (or data points for piecewise linear) total cost function
%     5 and following, cost data, piecewise linear data as:
%         x0, y0, x1, y1, x2, y2, ...
%     and polynomial data as, e.g.:
%         c2, c1, c0
%     where the polynomial is  $c0 + c1*P + c2*P^2$ 
%
%%----- Power Flow Data -----%%
%% system MVA base
baseMVA = 100.0000;

%% bus data
bus = [
1  3  0.0  0.0  0.0  0.0  1  1.0000  0.0000  345.0000
1  1.1000  0.9000;
2  2  0.0  0.0  0.0  0.0  1  1.0000  0.0000  345.0000
1  1.1000  0.9000;
3  2  0.0  0.0  0.0  0.0  1  1.0000  0.0000  345.0000
1  1.1000  0.9000;
4  1  0.0  0.0  0.0  0.0  1  1.0000  0.0000  345.0000
1  1.1000  0.9000;
5  1  90.0000  30.0000  0.0  0.0  1  1.0000  0.0000
345.0000  1  1.1000  0.9000;
6  1  0.0  0.0  0.0  0.0  1  1.0000  0.0000  345.0000
1  1.1000  0.9000;
7  1  100.0000  35.0000  0.0  0.0  1  1.0000  0.0000
345.0000  1  1.1000  0.9000;
8  1  0.0  0.0  0.0  0.0  1  1.0000  0.0000  345.0000
1  1.1000  0.9000;
9  1  125.0000  50.0000  0.0  0.0  1  1.0000  0.0000
345.0000  1  1.1000  0.9000;
];

%% generator data
gen = [
1  0.0000  0.0000  300.0000  -300.0000  1.0000  100.0000
1  250.0000  10.0000;
2  163.0000  0.0000  300.0000  -300.0000  1.0000  100.0000
1  300.0000  10.0000;
3  85.0000  0.0000  300.0000  -300.0000  1.0000  100.0000
1  270.0000  10.0000;
];

%% branch data
branch = [
1  4  0.0000  0.0576  0.0000  250.0000  250.0000
250.0000  0.0000  0.0000  1;
4  5  0.0170  0.0920  0.1580  250.0000  250.0000
250.0000  0.0000  0.0000  1;
5  6  0.0390  0.1700  0.3580  150.0000  150.0000
150.0000  0.0000  0.0000  1;
];

```

3	6	0.0000	0.0586	0.0000	300.0000	300.0000
300.0000		0.0000	0.0000	1;		
6	7	0.0119	0.1008	0.2090	150.0000	150.0000
150.0000		0.0000	0.0000	1;		
7	8	0.0085	0.0720	0.1490	250.0000	250.0000
250.0000		0.0000	0.0000	1;		
8	2	0.0000	0.0625	0.0000	250.0000	250.0000
250.0000		0.0000	0.0000	1;		
8	9	0.0320	0.1610	0.3060	250.0000	250.0000
250.0000		0.0000	0.0000	1;		
9	4	0.0100	0.0850	0.1760	250.0000	250.0000
250.0000		0.0000	0.0000	1;		

];

%%----- OPF Data -----%%

%% area data

area = [

1 5;

];

%% generator cost data

gencost = [

2 1500.00 0.00 3 0.11 5 150;

2 2000.00 0.00 3 0.085 1.2 600;

2 3000.00 0.00 3 0.1225 1 335;

];

Bibliographie

- [Aub.1], Aubin J.P., "*Méthodes explicites de l'optimisation*". Paris : Dunod, 1982.
- [Ach.2], Acher J. et Gardelle J., "*Programmation Linéaire*". Paris : Dunod, 1978.
- [Baz.3], Bazaraa M., "*Non Linear Programming Theory and Algorithms*". New York: John Wiley & Sons, 1980.
- [Chr.4], Christophe J., "*Introduction à l'Optimisation*". Ellipses, 1994.
- [Ghi.5], Ghilardi.J.P., "*Optimisation de la représentation de graphes par approche hybride déterministes et stochastiques*". Marseille : thèse d'doctorat d'état, 2002.
- [GoL.6], Goldberg.E., "*Algorithmes génétiques, exploration, optimisation et apprentissage automatique*". Paris: Addison-Wesley, 1994.
- [Hes.7], Hesser et Manner ., "*Techniques avancées pour le traitement de l'information*". Ellipses, 1994.
- [Pél.8], Pélissier R., "*Réseaux d'énergie électrique*". Paris : Université de Paris, 1972.
- [Ren.9], Renders J.M., "*Algorithmes génétiques et réseau de neuronne*". Paris.
- [Woo.10], Wood A.J. et Wollenberg B.F, "*Power generation operation & control*". New York: John Wiley & Sons, 1983.

De nombreux articles nous ayant servi pour la réalisation de notre travail sont disponibles sur Internet, en particulier sur les sites suivants

- <http://www.aic.nrl.navy.mil:80/galist>: archives de la Navy sur les AG.
- <http://GAL4.GE.UIUC.EDU/illigal.home.html>: illinois Genetic Algorithms Laboratory, dirigé par D.E. Goldberg, un des grands spécialistes des AG.
- <http://garage.cps.msu.edu/>: Genetic Algorithms Research and Applications Group (GARAGE).
- <http://www.natural-selection.com/index.html>: Natural Selection Inc., où l'on trouve les publications de D.B. Fogarty.
- <http://www.shef.ac.uk/~gaipp/mogas.html>: Evolutionary Computation Research Multiobjective Genetic Algorithms (Carlos Fonseca).
- <http://www.supelec.fr/ecole/mesures/pe/keramat.html>: Optimisation Globale avec les Algorithmes Génétiques (Mansour Keramat).