

Chapitre III. Techniques d'optimisation

Les méthodes d'optimisation sont classées en deux grandes familles: les méthodes déterministes et les méthodes stochastiques.

La grande majorité des méthodes de la première famille sont basées sur la connaissance de la direction de recherche donnée par le gradient de la fonction objectif. Ce sont des méthodes locales dont l'optimum dépend fortement de point initial.

Les méthodes stochastiques sont considérées comme des méthodes globales à cause de leur capacité à trouver le minimum global parmi plusieurs minima locaux pour n'importe quel contexte initial sans avoir besoin de connaître le gradient de la fonction objectif. Cependant, le temps de calcul élevé reste l'handicap majeur de ces méthodes.

III.1. Formulation d'un problème d'optimisation

Un problème d'optimisation se présente schématiquement comme suit : une variable physique ou une variable de décision ou de commande doit être choisie de façon optimale, c'est-à-dire de façon à optimiser (minimiser ou maximiser selon le cas) un critère physique (énergie, pertes, couple,...), un critère technique (précision, stabilité,...) ou économique (coût, rentabilité, productivité,...), tout en respectant certaines contraintes intrinsèques à la situation considérée.

Si nous considérons le vecteur X de dimension n , dont les n éléments représentent les variables ou paramètres d'optimisation :

$$X = [x_1, x_2, \dots, x_n]$$

$f(x)$: la fonction objectif

R^n : l'espace de recherche réalisable

Alors, de manière générale un problème d'optimisation peut être écrit sous la forme suivante:

$$\begin{cases} \text{Min } f(x) \in R^n \\ g_i(x) \leq 0 & i = 1, \dots, m \\ h_j(x) = 0 & j = 1, \dots, p \\ x_{k \min} \leq x_k \leq x_{k \max} & k = 1, \dots, n \end{cases} \quad (01)$$

où:

$g_i(x)$ et $h_j(x)$ représentent respectivement les contraintes d'inégalité et d'égalité
 x_{kmin} et x_{kmax} désignent les contraintes de domaine.

Nous donnons ci-dessous quelques définitions concernant le problème (II.1).

III.1.1. Minimum local

Un point x^* appartenant à l'espace de recherche réalisable R^n est un minimum local de la fonction objectif f dans R^n s'il existe un voisinage Δ :

$$\Delta \equiv \{x \in R^n / \|x - x^*\| \leq \delta\} \quad (02)$$

tel que:

$$\forall x \in \Delta \quad f(x) \geq f(x^*) \quad (03)$$

III.1.2. Minimum global

Un point x^* appartenant à l'espace de recherche réalisable R^n est un minimum global de f si:

$$\forall x \in R^n \quad f(x) \geq f(x^*) \quad (04)$$

Le minimum global est le petit minimum local. La figure 1 illustre la notion du minimum local et minimum global.

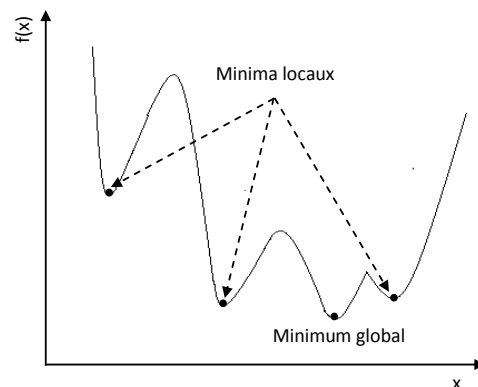


Figure 1 Minima locaux et minimum global d'une fonction f

III.2. Problème d'optimisation non contraint

Un problème d'optimisation sans contraintes est défini sous la forme suivante:

$$\begin{cases} \text{Min } f(x) \in R^n \\ x_{k \min} \leq x_k \leq x_{k \max} \quad k = 1, \dots, n \end{cases} \quad (08)$$

Il est similaire au problème (1) sauf que les fonctions contraintes $g_i(x)$ et $h_j(x)$ ne sont pas définies.

III.2.1. Condition d'optimalité

Pour résoudre un problème d'optimisation non contraint, il est nécessaire de caractériser la solution par des conditions.

Les conditions de premier et second ordre nécessaires pour que x^* soit un minimum local d'un problème non contraint sont données par:

$$\begin{cases} \nabla f(x^*) = 0 \\ H(x^*) \geq 0 \end{cases} \quad (09)$$

avec :

∇f est le gradient de la fonction objectif

$H = \nabla^2 f$ est la matrice de dérivées secondes partielles de f , dite la matrice Hessienne ou le Hessien.

III.3. Méthodes d'optimisation déterministes

Cette classe de méthodes est caractérisée par une évolution prévisible vers la solution, ne laissant aucune place au hasard. En partant de la même configuration initiale, une méthode d'optimisation déterministe fournit le même résultat. Ce sont des méthodes locales, c'est-à-dire qu'elles convergent vers un optimum dépendant uniquement de la configuration initiale, qu'il soit local ou global.

Elles sont classées en trois groupes suivant l'ordre de dérivabilité de la fonction objectif nécessaire pour résoudre le problème d'optimisation.

▪ Méthodes déterministes d'ordre zéro

Elles sont dites aussi méthodes heuristiques ou géométriques. Leur principe est basé sur l'exploration de l'espace de recherche par des essais successifs en recherchant les directions les plus favorables et nécessitent uniquement la connaissance de la fonction objectif. L'intérêt principal de telles méthodes réside dans le fait qu'elles ne nécessitent pas la connaissance des dérivées de la fonction objectif. Pour cela, elles sont souvent utilisées pour la résolution des problèmes discontinus. Dans cette catégorie des méthodes déterministes on trouve principalement : la méthode de Hooke et Jeeves et la méthode du Simplex.

▪ Méthodes déterministes d'ordre un

Ces méthodes nécessitent en plus de l'évaluation de la fonction objectif l'évaluation de son gradient à chaque itération. La connaissance de gradient donne des informations sur la direction de recherche et donc la localisation rapide de

l'optimum. Par contre, elles sont applicables uniquement aux problèmes où la fonction objectif est continûment différentiable. Parmi les méthodes déterministes d'ordre un les plus utilisées on trouve: la méthode de la plus grande pente et la méthode du gradient conjugué.

■ Méthodes déterministes d'ordre deux

Ces méthodes approfondissent, par rapport aux méthodes d'ordre un, l'étude locale de la fonction objectif en utilisant les dérivées secondes pour déterminer la direction de recherche. Les principales méthodes utilisées sont les méthodes de Newton et leurs dérivées, communément appelées méthodes de quasi-Newton.

III.3.1. Méthodes déterministes unidimensionnelles

Les méthodes déterministes unidimensionnelles sont employées pour l'optimisation des problèmes à une seule dimension (un seul paramètre). Leur principe est basé sur la réduction successive de l'intervalle de recherche pour localiser le point optimal. Ces méthodes sont aussi appelées méthodes de recherche linéaire. Parmi ces méthodes, nous présentons ci-dessous la méthode de la section dorée et la méthode de Dichotomie.

La majorité de ces méthodes n'exigent pas que la fonction objectif soit continûment différentiable ni même continue, mais seulement unidimensionnelle.

III.3.1.1. Méthode de la section dorée

La méthode de la section dorée réduit l'intervalle de la solution à chaque itération par un facteur $k=0.6180$. Ce facteur est la section dorée et il est la solution positive de l'équation:

$$1+k = \frac{1}{k} \quad (11)$$

Supposons que la solution optimale de la fonction objectif $f(x)$ se trouve dans l'intervalle $[x_1, x_2]$. En considérant les deux points:

$$x_3 = kx_1 + (1-k)x_2, \quad x_4 = kx_2 + (1-k)x_1 \quad (12)$$

et en comparant les valeurs de $f(x_3)$ et $f(x_4)$, la méthode de la section dorée procède comme suit:

✓ Si $f(x_3) > f(x_4)$; le point optimal se trouve dans l'intervalle $[x_3, x_2]$, ainsi x_3 remplace x_1 .

✓ Si $f(x_3) < f(x_4)$; le point optimale appartient à l'intervalle $[x_1, x_4]$, x_2 est remplacé par x_4 .

On applique cette démarche successivement jusqu'à l'obtention d'un intervalle de longueur inférieure à k .

III.3.1.2. Méthode de Dichotomie

Elle est basée sur le même principe de réduction de l'intervalle de recherche à chaque itération. En considérant le triplet (x_1, x_2, x_3) appartenant à l'intervalle de recherche, tel que $f(x_1) > f(x_2) < f(x_3)$, le principe de la méthode consiste à découper en deux les intervalles $[x_1, x_2]$ et $[x_2, x_3]$ de façon à obtenir de nouveaux points (x_4, x_5) sur lesquels la fonction objectif sera évaluée.

A partir des valeurs de la fonction objectif en ces points, on choisit le nouveau triplet pour le prochain découpage. Ainsi, à chaque itération du processus, nous utilisons deux nouvelles évaluations de la fonction objectif pour réduire l'intervalle de recherche. La figure 2 montre le principe de découpage de l'intervalle de recherche par la méthode de Dichotomie.

Le processus itératif s'arrête lorsque l'intervalle de découpage devient plus petit à une certaine tolérance ε donnée.

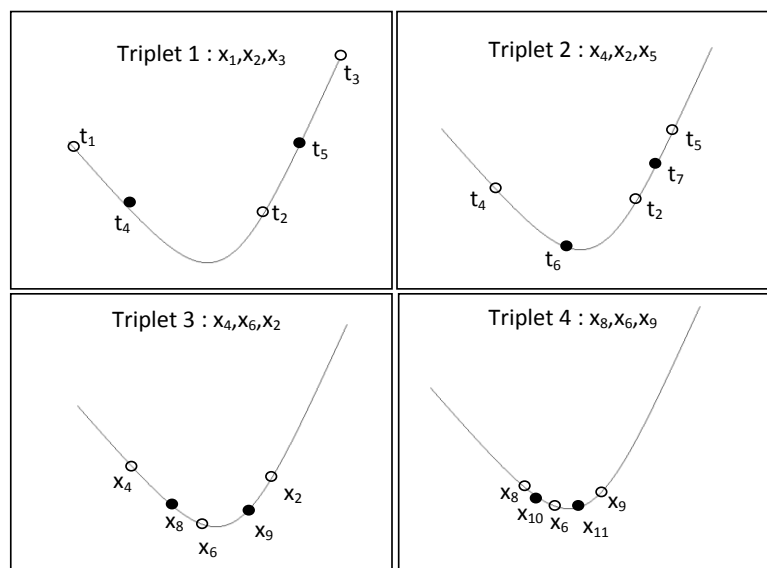


Figure 2. Méthode de Dichotomie.

III.3.2. Méthodes déterministes multidimensionnelles

Les méthodes déterministes multidimensionnelles sont employées pour l'optimisation des problèmes à une ou plusieurs dimensions. Les méthodes d'ordre égal ou supérieur à un nécessitent la connaissance d'une direction de recherche. Généralement et afin d'accélérer la convergence, une méthode de recherche linéaire (méthode unidimensionnelle) est utilisée dans la direction de recherche.

III.3.2.1. Méthode du Simplex

Contrairement aux méthodes qui démarrent d'un point initial, la méthode du simplex démarre d'un simplex original.

Un simplex est une figure géométrique de $(n+1)$ points, n étant le nombre des paramètres d'optimisation $X_i, i=1,n$. A chaque itération de l'algorithme, un nouveau Simplex est généré tout en éliminant le point où la fonction objectif présente sa plus grande valeur. L'application de la méthode du Simplex s'effectue en trois étapes: réflexion, expansion et contraction.

Soit les suppositions initiales suivantes:

- ✓ $f(x)$ la fonction objectif
- ✓ $X=[x_1, x_2, \dots, x_n]$ le vecteur des paramètres de conception.
- ✓ $f_h = \max(f(x_i))$ et $f_l = \min(f(x_i))$
- ✓ $\bar{X}$ est l'isobarycentre des points x_i avec $i \neq h$

a. Réflexion

$$x^* = (1 + \alpha)\bar{x} - \alpha x_h \quad (16)$$

avec α est une constante positive et elle représente le coefficient de réflexion.

b. Expansion

$$x^{**} = \gamma x^* + (1 - \gamma)\bar{x} \quad (17)$$

avec γ est le coefficient d'expansion, il est pris supérieur à un.

c. Contraction

$$x^{***} = \beta x_h + (1 - \beta)\bar{x} \quad (18)$$

Le coefficient de contraction β est compris entre zéro et un.

III.3.2.2. Méthode de la plus grande pente

La méthode de la plus grande pente ou méthode du gradient à pas optimal, est l'une des méthodes les plus classiques pour l'optimisation des fonctions multidimensionnelles. Elle appartient à la classe des méthodes déterministes d'ordre un. Son principe est basé sur le choix d'une direction de descente opposée à celle du gradient. A chaque itération x_k , le gradient de la fonction objectif $f(x)$ est évalué, la direction de recherche choisie est la suivante:

$$d_k = -\nabla f(x_k) \quad (19)$$

L'algorithme de la méthode peut être résumé dans les étapes suivantes:

1. Initialisation: choisir un point initial x_0
2. Direction de recherche : pour l'itération k , mettre $d_k = -\nabla f(x_k)$
3. Critère d'arrêt: pour une tolérance prédéterminée $\varepsilon > 0$,
- Si $\|d_k\| \leq \varepsilon$ arrêter le processus et prendre x_k comme solution optimale

- Si non, continuer

4. Recherche linéaire: utiliser une méthode de recherche linéaire pour déterminer le pas optimal α :

$$f(x_k + \alpha_k d_k) \quad (20)$$

α_k issue de cette minimisation étant le pas optimal, il nous amène à un nouveau point de recherche à chaque itération.

5. Nouvelle approximation: mettre

$$x_{k+1} = x_k + \alpha_k d_k \quad (21)$$

Retourner à l'étape 2

III.3.2.3. Méthode du gradient conjugué

Le gradient conjugué utilise le même principe que la méthode précédente, à la différence que la direction de descente n'est plus donnée par le gradient de la fonction, mais par des directions conjuguées successives. Deux directions h_0 et h_1 sont dites conjuguées par rapport à la matrices Hessienne H d'une fonction, si:

$$h_0^T H h_1 = 0 \quad (22)$$

Les directions conjuguées peuvent être obtenues directement à partir du Hessien de la fonction objectif. Cependant, ce calcul peut représenter un coût important pour la méthode d'optimisation. Pour éviter ce problème, la méthode du gradient conjugué effectue le calcul des directions conjuguées à partir de ces deux équations:

$$h_{k+1} = g_{k+1} + \beta_k h_k \quad (23)$$

$$\beta_k = \frac{g_{k+1}^T g_{k+1}}{g_k^T g_k} \quad (24)$$

où:

h_k et h_{k+1} sont des directions conjuguées

g_k et g_{k+1} sont les directions opposées aux gradients calculés sur les points x_k et x_{k+1} , respectivement.

Ce calcul est uniquement valable si le point x_{k+1} est obtenu à partir d'une minimisation linéaire où h_k représente la direction de recherche, comme nous montre l'équation suivante:

$$x_{k+1} = x_k + \alpha_k h_k \quad (25)$$

où:

α_k est le pas optimal donné par une minimisation linéaire.

Récapitulons la méthode de gradient conjugué dans l'algorithme correspondant suivant:

1. Initialisation: choisir une approximation initiale x_0
2. Critère d'arrêt: pour une tolérance prédéterminée $\varepsilon > 0$,
 - Si $\|\nabla f(x_k)\| \leq \varepsilon$ arrêter le processus et prendre x_k comme solution optimale
 - Sinon, continuer
3. Recherche linéaire: utiliser une méthode de recherche linéaire pour déterminer le pas optimal α_k , tout en minimisant la fonctionnelle:

$$f(x_k + \alpha_k h_k) \quad (26)$$

4. Nouvelle approximation: donnée par l'équation (25)

Plusieurs expressions pour le choix du terme β_k existent, les plus utilisées parmi elles sont:

- ✓ Algorithme de Fletcher-Reeves : le terme β_k est donné par l'expression (24)
- ✓ Algorithme de Polak-Ribière: le terme β_k est donné par l'expression suivante :

$$\beta_k = \frac{(g_{k+1} - g_k)g_{k+1}}{g_k \cdot g_k} \quad (27)$$

III.3.2.4. Méthode de Newton

La méthode de Newton contrairement aux méthodes du premier ordre, est reconnue pour être très efficace, notamment au voisinage du minimum. Son principe pour l'optimisation est de minimiser successivement les approximations au second-ordre de la fonctionnelle f . Elle prend pour direction de recherche la valeur suivante:

$$d = -[\nabla^2 f(x)]^{-1} \nabla f(x) \quad (28)$$

avec $\nabla^2 f(x)$ est la matrice Hessienne.

La principale difficulté de la méthode de Newton réside dans le calcul des dérivées secondes de la fonctionnelle, calcul généralement très coûteux en terme de temps de calcul. Dans le but de surmonter cette difficulté, les méthodes dites de Quasi-Newton, dérivées de la méthode de Newton sont utilisées. Ces méthodes procèdent à une approximation successive du Hessien en utilisant uniquement le gradient de la fonctionnelle.

III.3.2.5. Méthodes Quasi-Newton

Les méthodes de Quasi-Newton consistent à imiter l'algorithme de Newton, mais sans calculer le Hessien de la fonction objectif ni son inverse. Les étapes principales pour l'application d'une méthode de type Quasi-Newton sont données ci-dessous:

1. Initialisation: Soit x_1 l'approximation initiale, $H_1 = I$, $p_k = -H_1 \nabla f(x_1)$, où I est la matrice identité.

2. Recherche linéaire: Déterminer le pas optimal courant α_k par la minimisation de la fonctionnelle suivante: $f(x_k + \alpha_k p_k)$
3. Nouvelle approximation: mettre $x_{k+1} = x_k + \alpha_k p_k$
4. Critère d'arrêt: pour une tolérance prédéterminée $\varepsilon > 0$, si $\|\nabla f(x_{k+1})\| \leq \varepsilon$ arrêter le processus, si non continuer.
5. Nouvelle direction de recherche: soit

$$q_k = \nabla f(x_{k+1}) - \nabla f(x_k), p_{k+1} = -H_{k+1} \nabla f(x_{k+1}) \quad (29)$$

Retourner à l'étape 2.

Pour le calcul du Hessien H_{k+1} à l'itération $k+1$ plusieurs algorithmes existent, les plus employés sont:

✓ L'algorithme de Davidon-Fletcher-Powell (DFP):

$$H_{k+1} = H_k + \alpha_k \frac{p_k p_k^T}{p_k^T q_k} - \frac{H_k q_k q_k^T H_k}{q_k^T H_k q_k} \quad (30)$$

✓ L'algorithme de Broyden-Fletcher-Goldfarb-Shannon (BFGS):

$$H_{k+1} = \alpha_k \frac{p_k p_k^T}{q_k^T p_k} + \left(1 - \frac{p_k q_k^T}{q_k^T p_k}\right) H_k \left(1 - \frac{q_k p_k^T}{q_k^T p_k}\right) \quad (31)$$

III.4. Méthodes d'optimisation stochastiques

Les méthodes stochastiques sont des méthodes où l'approche de l'optimum est entièrement guidée par un processus probabiliste et aléatoire (stochastique). Ces méthodes ont une grande capacité de trouver l'optimum global du problème. Contrairement à la plupart des méthodes déterministes, elles ne nécessitent ni de point de départ, ni la connaissance du gradient de la fonction objectif pour atteindre la solution optimale. Cependant, elles demandent un nombre important d'évaluations de la fonction objectif avant d'arriver à la solution du problème.

Parmi les méthodes stochastiques les plus employées, nous distinguons le Recuit Simulé développé par Kirkpatrick en 1983, la Recherche Tabou développée par Glover en 1989 et les méthodes évolutionnistes comme les Algorithmes Génétiques développés par Holland en 1975 et l'Optimisation par Essaim de Particule développée par Kennedy et Eberhart en 1995.

III.4.1. Algorithmes Génétiques

Les Algorithmes Génétiques ont été proposés par Holland en 1975, puis développés par d'autres chercheurs par la suite. Ils sont actuellement une des méthodes les plus diffusées. La méthode des Algorithmes Génétiques (AG) fait partie d'une famille de méthodes stochastiques appelée Méthodes Évolutionnistes. Cette méthode s'inspire des mécanismes de l'évolution naturelle des êtres vivants.

L'AG est basé sur la traduction mathématique des phénomènes naturels qui sont la reproduction, la survie et l'adaptation des individus, dans le sens que les individus d'une population les mieux adaptés à leur environnement ont une plus grande probabilité de survivre et de se reproduire de génération en génération, en donnant des descendants encore mieux adaptés.

L'AG est actuellement une des méthodes les plus utilisées dans la résolution de problèmes d'optimisation dans de nombreux domaines d'application.

III.4.1.1. Analogie entre l'algorithme génétique et la théorie de l'évolution naturelle

Dans l'AG, l'ensemble des paramètres du problème à optimiser est défini comme étant un individu. Un individu représente une solution particulière au problème à optimiser. Un ensemble d'individus donne naissance à la population. La population représente donc un ensemble de solutions du problème à optimiser. Elle représente aussi un ensemble de différentes configurations de paramètres, donc un sous espace de recherche.

L'adaptation à l'environnement est donnée par la valeur retournée de la fonction objectif. Les générations sont représentées par les itérations du processus d'optimisation. Chaque nouvelle génération ou nouvelle itération comprend une nouvelle population donc une nouvelle configuration d'individus alors un nouveau sous espace de recherche.

Dans le tableau ci-dessous on résume la correspondance entre la théorie d'évaluation naturelle et l'algorithme génétique.

Tableau 1. Analogie entre les AG et la théorie d'évolution naturelle

Théorie d'Evaluation Naturelle	Algorithmes Génétiques
Individu	Ensemble de paramètres
Population	Ensemble de configuration de paramètres
Environnement	Espace de recherche
Adaptation de l'individu	Evaluation de la fonction objectif
Génération	Itérations de la méthode

Dans la figure qui suit on représente les trois niveaux d'organisation de l'algorithme génétique.

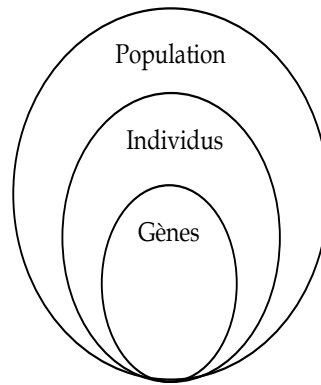


Figure 3. Niveaux d'organisation d'un algorithme génétique.

III.4.1.2. Principe des algorithmes génétiques

L'algorithme génétique est basé sur quatre éléments principaux qui sont : l'évaluation, la sélection, le croisement et la mutation.

Après l'initialisation aléatoire de la première population d'individus qui définit la première génération, on répète successivement les quatre étapes suivantes :

1. L'évaluation des individus par le calcul de leurs fonctions objectifs (mesure de l'adaptation).
2. La sélection des individus reproducteurs : théoriquement les individus qui s'adaptent le mieux à l'environnement défini par la fonction objectif.
3. Application de l'opérateur de croisement. Cet opérateur permet l'exploration de l'espace de recherche.
4. Application de l'opérateur de mutation. Cet opérateur joue un double rôle : explorer l'espace de recherche qui n'a pas pu être atteint par l'opérateur de croisement et réaliser une recherche locale, très proche de la solution en cours.

A la fin de l'étape quatre nous obtiendrons une nouvelle population. Cette population constitue l'ensemble d'individus de la génération (itération) qui suit.

Ces quatre étapes sont répétées autant de fois qu'il y a besoin de générations pour satisfaire un critère d'arrêt. La solution est alors représentée par le meilleur individu de la dernière génération.

III.4.1.3. Mise en œuvre de la procédure des algorithmes génétiques

La mise en œuvre de la procédure des algorithmes génétiques nécessite en premier lieu la modélisation de l'ensemble des étapes qui la constituent. Cette modélisation consiste en la traduction mathématique des différents passages de la procédure. Dans ce qui suit nous développons les différents outils permettant la modélisation et la mise en œuvre de la procédure des AG.

▪ Le codage

Dans l'algorithme génétique de base, tel qu'il a été fondé par Holland, les gènes (paramètres à optimiser) sont formés de 1 et 0. Dans ce cas, chaque paramètre réel

est codée par son équivalent en binaire et l'individu obtenu est représenté par une chaîne codée de plusieurs gènes (paramètres) représentant une solution particulière pour la fonction objectif, figure 4.II.

De nouvelles versions d'AG sont apparues. Elles ne se basent plus sur le codage binaire mais elles travaillent directement sur les paramètres réels. Ces versions sont appelées algorithmes génétiques avec codage réel figure 4.I.

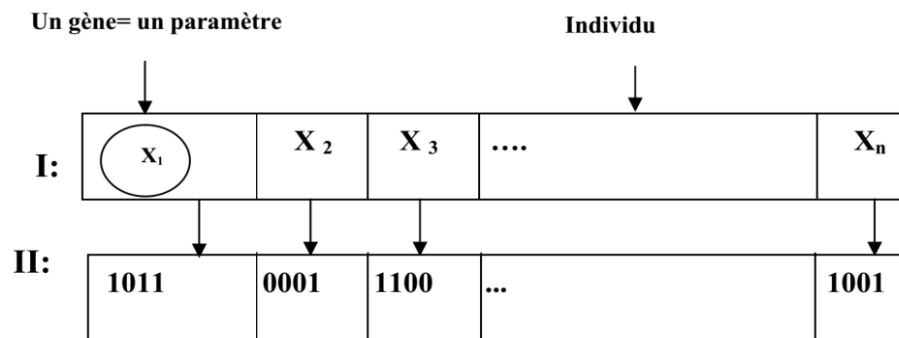


Figure 4. Représentation d'un individu; I. codage réel, II. codage binaire

▪ L'évaluation

La fonction d'adaptation, évaluation, ou fitness, associe une valeur pour chaque individu. Cette valeur a pour but d'évaluer si un individu est mieux adapté qu'un autre à son environnement. Ce qui signifie qu'elle quantifie la réponse fournie au problème pour une solution potentielle donnée. Ainsi les individus peuvent être comparés entre eux.

▪ La Sélection

L'opérateur de sélection est appliqué sur la population courante de façon à sélectionner les individus qui iront former la population de la prochaine génération. La sélection de ces individus est basée sur leur valeur d'adaptation. Ainsi, les individus les plus adaptés sont généralement sélectionnés pour constituer la génération suivante, alors que les plus faibles sont exclus sans avoir la possibilité d'avoir des descendants. Il existe différentes façons d'implémenter un opérateur de sélection, parmi lesquelles nous trouvons la sélection proportionnelle.

La sélection proportionnelle consiste à attribuer à chaque individu i_j un nombre réel p_j qui représente le nombre de descendants attendus pour lui dans la génération suivante.

La valeur de p_j est calculée par:

$$p_j = \frac{f(i_j)}{\sum_{k=1}^N f(i_k)} \cdot N \quad (32)$$

Avec:

N : la taille de la population (le nombre d'individus).

$f(i_j)$: la valeur d'adaptation de l'individu i_j .

La deuxième étape du processus de sélection consiste à convertir la valeur du p_j de chaque individu en un nombre de descendants que chacun d'entre eux aura effectivement dans la prochaine génération. Cette conversion est obtenue à l'aide d'un algorithme qui transforme les valeurs réelles des p_j en valeurs entières. Cet algorithme repose sur le principe de la roue de loterie.

Dans l'algorithme proposé par Holland, on crée une roue de loterie divisée en secteurs proportionnels au p_j de chaque individu. Le nombre de secteur est égal au nombre d'individus N . Ensuite, on fait tourner la roue un nombre de fois égal à la taille de la population donc N fois. À chaque coup, on prend une copie (descendant) de l'individu désigné par l'aiguille de la roue pour faire partie de la nouvelle population. Ainsi, les individus avec un plus grand p_j auront un plus grand secteur dans la roue et en conséquence une plus grande probabilité de participer dans la génération suivante.

La Figure 3 montre une représentation graphique de cet algorithme.

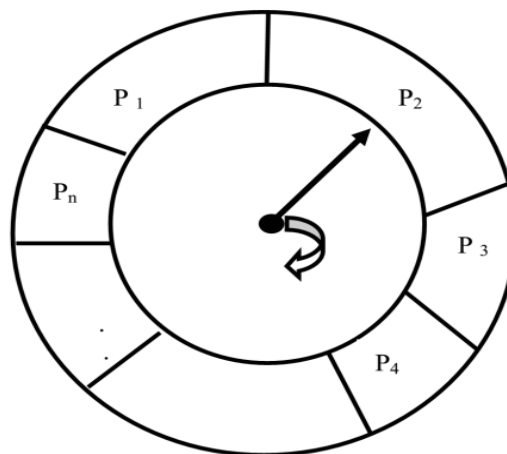


Figure 5. Algorithme de sélection par roue de loterie

A la fin de l'application de l'opérateur de sélection, la nouvelle population, de N individus, contiendra une plus grande proportion des meilleurs individus de la génération précédente. On passe alors à l'étape de reproduction, dans laquelle seront utilisés les opérateurs de croisement et de mutation.

▪ Le croisement

L'opérateur de croisement est utilisé pour échanger les caractéristiques "génétiques" entre les différents individus d'une génération quelconque. Cet échange s'effectue en choisissant deux individus au hasard (parents) qui seront "croisés" avec une certaine probabilité de croisement p_c de façon à générer deux

nouveaux individus (enfants). Les enfants remplaceront leurs parents et formeront la nouvelle population.

Dans le cas d'un codage réel des individus, ce "croisement" peut être obtenu à partir d'un simple échange de paramètres entre les deux parents, comme le montre la Figure 6. La zone de croisement, au niveau de la paire d'individus (parents), est choisie aléatoirement.

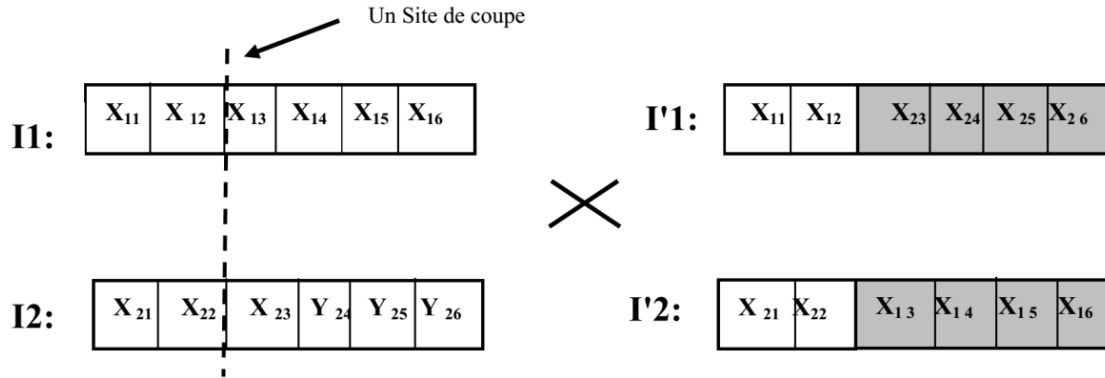


Figure 6. Croisement en un point

Le croisement représenté sur la Figure 6 est du type 1-point. Il existe d'autres implémentations de croisement, tels que le type 2-points, le croisement multiple,.... Malgré ces différentes façons de "croiser" les individus, le but de ces opérateurs reste toujours l'exploration de nouvelles régions de l'espace de recherche à partir de l'échange de caractéristiques entre les individus de la même population.

▪ La mutation

L'opérateur de mutation est appliqué sur les individus d'une population de façon à obtenir d'autres individus avec des nouvelles caractéristiques "génétiques". Dans le cas d'un codage réel, le mécanisme de mutation peut être implémenté en choisissant un individu de la génération courante au hasard et en modifiant un de ses paramètres aléatoirement avec une probabilité de mutation p_m . Ce mécanisme est dénommé mutation uniforme. Le but de l'opérateur de mutation est d'atteindre des nouvelles régions de l'espace de recherche.

La probabilité de mutation est généralement donnée par:

$$p_m = \frac{1}{N\sqrt{L}} \quad (33)$$

avec L est le nombre de paramètres qui forment l'individu.

En utilisant les quatre opérateurs que nous venons de décrire, les meilleurs individus se propagent de génération en génération, en se combinant ou en échangeant leurs meilleures caractéristiques. En favorisant les meilleurs individus,

les régions les plus prometteuses de l'espace de recherche sont explorées, ce qui permet d'atteindre l'optimum global.

Après la mutation, les individus constitueront la nouvelle population de N individus qui donne naissance à la nouvelle génération. Le cycle continue: évaluation, sélection, croisement, mutation, évaluation, etc. jusqu'à la dernière génération fixée. Il y a donc quatre paramètres de base qui doivent être fixés pour assurer le fonctionnement d'un AG: le nombre d'individus dans la population N , la génération maximale G_{max} , les taux de croisement p_c et de mutation p_m . Trouver de bonnes valeurs à ces paramètres est un problème souvent délicat.

III.4.1. Optimisation par essaim de particules

L'optimisation par essaim de particules (PSO) est une technique développée par Eberhart et Kennedy en 1995, elle est inspirée à partir du comportement social des nuées d'oiseaux et des bancs de poissons.

PSO partage beaucoup de similitudes avec des techniques évolutionnaires de calcul telles que les algorithmes génétiques. A la différence des AG, PSO n'a aucun opérateur de reproduction (croisement ou mutation).

PSO est basée sur une population de particules, un essaim regroupe plusieurs particules (individus). Chaque particule prend sa décision en utilisant sa propre expérience et les expériences de leur voisinage.

Comme les algorithmes génétiques, PSO démarre le processus d'optimisation par une population des solutions aléatoires dans l'espace de recherche. La position de chaque particule est représentée par sa coordonnée et également par sa vitesse.

Le déplacement de chaque particule dans l'espace de recherche, est basé sur sa position actuelle et la mise à jour de sa vitesse.

$$S_i^{k+1} = S_i^k + V_i^{k+1} \quad (34)$$

Tel que :

S_i^{k+1}, S_i^k : Position de la particule i à l'itération $k+1$ et k respectivement.

V_i^{k+1} : Vitesse de la particule i à l'itération $k+1$.

Chaque particule dans l'essaim, change sa vitesse suivant deux informations essentielles. Une, est liée à son expérience personnelle, qui est la meilleure position trouvée par la particule durant le processus de recherche $pbest$. La deuxième information, concernant la meilleure position trouvée par tout l'essaim $gbest$.

Le principe de changement de la vitesse est défini par l'équation (4):

$$V_i^{k+1} = wV_i^k + c_1 rand_1 \times (pbest_i - s_i^k) + c_2 rand_2 \times (gbest - s_i^k) \quad (35)$$

D'où:

V_i^k : Vitesse de la particule i à l'itération k ,

W : Fonction de pondération,

C_j : Facteurs de pondération,

$rand$: Nombre aléatoire entre 0 et 1,

S_i^k : Position actuelle de la particule i à l'itération k ,

$pbest_i$: Meilleure position trouvée par la particule i jusqu'à ici,

$gbest$: Meilleure position trouvée par l'essaim jusqu'à ici.

La fonction de pondération w est donnée par l'équation suivante:

$$w = w_{\max} - \frac{w_{\max} - w_{\min}}{iter_{\max}} \times iter \quad (36)$$

Tel que:

$w_{\max}$: Poids initial,

$w_{\min}$: Poids final,

$iter_{\max}$: Nombre d'itérations maximum,

$iter$: Itération courante.

La fonction de pondération w joue un rôle important dans la procédure de recherche. Elle garantit un équilibre entre la recherche locale et la recherche globale, un bon choix de cette fonction augmente l'efficacité de la méthode pour avoir une solution globale. L'expérience a montré que la diminution linéaire de la valeur de w de 0.9 à 0.4 au cours de la procédure de recherche donne des meilleurs résultats.

L'organigramme général du PSO, est présenté selon la procédure suivante:

Etape 1: Génération d'un état initial de chaque particule.

Les points de recherche initiaux, position (S_i^0) et vitesse (V_i^0) pour chaque particule sont habituellement générés aléatoirement dans l'espace de recherche. Le point de recherche courant est placé à $pbest$ pour chaque particule. La meilleure valeur évaluée de $pbest$ est placée à $gbest$.

Etape 2: Recherche d'une nouvelle position pour chaque particule

La valeur de la fonction objective est calculée pour chacun des particules. Si la valeur d'une particule est meilleure que son $pbest$ courant, $pbest$ prend cette nouvelle valeur. Si la meilleure valeur de $pbest$ est meilleure que $gbest$ courant, $gbest$ est remplacé par celle-ci et la particule qui correspond à cette valeur est ainsi stockée.

Etape 3: Modification de chaque point de recherche

Le point de recherche courant de chaque particule est changé en utilisant les équations (34), (35) et (36).

Etape 4: Vérification de l'état de sortie

Le nombre courant d'itération atteint le nombre maximum d'itération $iter_{max}$, alors fin du programme, autrement, retourner à l'étape 2.