III. METHODES GLOBALES D'OPTIMISATION

–Méthodes de recherche aléatoire

–Algorithme du recuit simulé

–Algorithmes génétiques

III. METHODES GLOBALES D'OPTIMISATION

III.1 Méthodes de recherche aléatoire

III.1.1 Méthode des sauts aléatoires

soit $x = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$ avec $l_i \le x_i \le u_i \quad i = 1 \cdots n$

soit $r = \begin{bmatrix} r_1 \\ \vdots \\ r_n \end{bmatrix}$ avec r_i variables aléatoires uniformément

distribuées sur $[0,1]$

alors $x = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} l_1 + r_1(u_1 - l_1) \\ \vdots \\ l_n + r_n(u_n - l_n) \end{bmatrix}$

III. METHODES GLOBALES D'OPTIMISATION

III.1 Méthodes de recherche aléatoire

III.1.2 Méthode de la promenade aléatoire

$$x_{k+1} = x_k + \alpha_k d_k \quad \text{avec} \quad d_k \text{ vecteur aléatoire}$$

Algorithme : Initialisation : $x_0 \quad \alpha_0 \quad k = 0 \quad cc = 1$

1° Soient $\{r_i\}$ n variable s aléatoires uniforméme nt distribuée s sur $[-1, 1]$

si $R = \sqrt{r_1^2 + r_2^2 + \cdots r_n^2} > 1 \Rightarrow$ nouveau $\{r_i\}$

$$d_k = \frac{1}{R}\begin{bmatrix} r_1 \\ \vdots \\ r_n \end{bmatrix} \quad R \leq 1$$

III. METHODES GLOBALES D'OPTIMISATION

III.1 Méthodes de recherche aléatoire

III.1.2 Méthode de la promenade aléatoire

Algorithme (suite)

2° $x_{k+1} = x_k + \alpha_k d_k$

3° $f(x_{k+1}) < f(x_k) \Rightarrow k = k+1 \quad cc = 1$ => Retour en 1°

sinon si $cc < N_{cc} \Rightarrow cc = cc + 1$ => Retour en 1°

sinon $\alpha_k = \dfrac{\alpha_k}{2} \quad cc = 1$ => Retour en 1°

si $\alpha_k < \varepsilon \Rightarrow$ stop

III. METHODES GLOBALES D'OPTIMISATION

III.1 Méthodes de recherche aléatoire

Avantages des méthodes de recherche aléatoire

1. $f(x)$ peut être discontinue et non dérivable
2. Recherche un minimum global
3. Utilisable quand toute les autres méthodes échouent
4. Première étape avant un algorithme plus performant
   (un maillage initial peut également être réalisé)

Inconvénients des méthodes de recherche aléatoire

1. Convergence très lente surtout près de la solution
2. Explosion combinatoire pour les problèmes de grande dimension

=> Première étape avant un algorithme plus performant

III. METHODES GLOBALES D'OPTIMISATION

III.2 Algorithme du recuit simulé *(simulated annealing)*

Analogie avec un processus thermique en physique
de la matière : le recuit

* faire fondre un solide
* le refroidir lentement

Algorithme de Métropolis : simulation du recuit

- Soit un état x_k de niveau d'énergie E_k
- Perturbation aléatoire $\Rightarrow x_{k+1}$ avec niveau E_{k+1}
- si $E_{k+1} \leq E_k \Rightarrow x_{k+1}$ est le nouvel état

  si $E_{k+1} > E_k \Rightarrow x_{k+1}$ est le nouvel état

  avec probabilité $e^{-\frac{(E_{k+1}-E_k)}{k_B T}}$

k_B = constante de Boltzmann
T = température (Kelvin)

III. METHODES GLOBALES D'OPTIMISATION

III.2 Algorithme du recuit simulé *(simulated annealing)*

Analogie en optimisation

* $f(x) = \mathrm{E}$ niveau d'énergie
* x = état du système
* Paramètre de contrôle = température

$$\Rightarrow P\left(x_{k+1} \text{ soit accepté}\right) = \begin{cases} 1 & \text{si} \quad f(x_{k+1}) \le f(x_k) \\ e^{-\frac{(f(x_{k+1}) - f(x_k))}{c_k}} & \text{sinon} \end{cases}$$

c_k paramètre de contrôle

III. METHODES GLOBALES D'OPTIMISATION

III.2 Algorithme du recuit simulé *(simulated annealing)*

Algorithme du recuit simulé

Initialisation : $x_0 \quad c_0 \quad L_0 \quad k = 0$

Itérations : * pour $i = 1 \rightarrow L_k$

générer aléatoirement x_{k_i} dans un voisinage x_k

si $f(x_{k_i}) \leq f(x_k)$ alors $x_k = x_{k_i}$

sinon si $e^{-\frac{(f(x_{ki}) - f(x_k))}{c_k}} > v.a.$ uniforme sur $[0,1]$

alors $x_k = x_{k_i}$

* $k = k + 1 \quad x_{k+1} = x_k$
* modifier L_k
* modifier c_k

III. METHODES GLOBALES D'OPTIMISATION

III.2 Algorithme du recuit simulé *(simulated annealing)*

Avantages : * possibilité d'échapper d'un minimum local
* agitation ajustable avec la température

Procédure de refroidissement

1° Initialisation du paramètre de contrôle

c_0 grand $\Rightarrow$ taux d'acceptation $\tau \approx 1$

2° Décroissance du paramètre de contrôle

Changement lent $c_{k+1} = \alpha\, c_k$ avec $0{,}9 < \alpha < 1$

3° Valeur finale du paramètre de contrôle

x_k constant durant une séquence L_k

4° Nombre de transitions testées: L_k

Si on souhaite un nombre fixe de transitions acceptées
=> $\lim_{k\to\infty} L_k = \infty$ => limite supérieure $L_{\max}$

III. METHODES GLOBALES D'OPTIMISATION

III.2 Algorithme du recuit simulé *(simulated annealing)*

Exemple: Refroidissement en temps polynomial

1° Valeur initiale c_0

m_1 transitions $\Big| f(x_{k_i}) \le f(x_k)$ et m_2 transitions proposées $\Big| f(x_{k_i}) > f(x_k)$

$$\Delta\bar{f} = \frac{1}{m_2} \sum_{\substack{i=1 \\ f(x_{ki})>f(x_k)}}^{m_2} \left[f(x_{k_i}) - f(x_k) \right]$$

moyenne de l'augmentation possible

$$\Rightarrow \quad \text{taux d'acceptation} \quad \tau = \frac{m_1 + m_2 e^{-\Delta\bar{f}/c_0}}{m_1 + m_2} \neq \tau_0$$

$$\Rightarrow \quad c_0 = \frac{\Delta\bar{f}}{\ln \dfrac{m_2}{m_2\tau_0 - (1-\tau_0)m_1}}$$

Itérations jusqu'à la convergence de τ

III. METHODES GLOBALES D'OPTIMISATION

III.2 Algorithme du recuit simulé *(simulated annealing)*

Refroidissement en temps polynomial (suite)

2° Décroissance de c_k

$$c_{k+1} = \frac{c_k}{1 + \frac{c_k \ln(1+\delta)}{3\sigma_k}} \quad \text{avec} \quad \bar{f}_k = \frac{1}{L_k} \sum_{i=1}^{L_k} f(x_{k_i})$$

$$\sigma_k = \sqrt{\frac{1}{L_k} \sum_{i=1}^{L_k} (f(x_{k_i}) - \bar{f}_k)^2}$$

δ paramètre de distance

3° Valeur finale $\quad \frac{c_k}{\bar{f}_0} \quad \frac{\bar{f}_{(k-1)} - \bar{f}_k}{c_{k-1} - c_k} < \varepsilon$

4° Nombre de transitions $\quad L_k = L \quad > 100 \quad \forall k$

III. METHODES GLOBALES D'OPTIMISATION

III.3 Algorithmes génétiques

Principe

Population initiale *P(0)* avec $x \in P(0) \Rightarrow x \in \Omega$

Naissance d'une population *P(1)* par croisements et mutations

La population *P(1)* va engendrer ensuite une population *P(2)*, etc…

III.3.1 Chromosomes (algorithmes génétiques classiques)

Codage des points de Ω en suites de L symboles tirés d'un alphabet.

Exemple: alphabet binaire (0,1)

=> Sélection d'une population initiale *P(0)* de chromosomes

III. METHODES GLOBALES D'OPTIMISATION

III.3 Algorithmes génétiques

III.3.2 Sélection: Formation d'une population intermédiaire $M(k)$ de même taille que $P(k) = N$

1° Evaluation de $f(x)$ pour chaque individu/élément de $P(k)$

2° Définition d'une fonction $F(x)$ = fonction efficacité « Fitness »

$$\text{Exemple}: F(x_i) = 2 - p + 2(p-1)\left[\frac{R(x_i)-1}{N-1}\right] \quad p \in [1,2] \text{ pression sélective}$$

$R(x_i) = \text{rang de } x_i \text{ dans } P(k)$

$$\Rightarrow R(x_i) = \begin{cases} 1 & si \quad x_i \text{ le plus faible} \\ N & si \quad x_i \text{ le plus fort} \end{cases}$$

$\Rightarrow F(x_i)$ normalisé

3° Sélection: chaque point de $M(k)$ est égal à un point $x_i(k) \in P(k)$ avec la probabilité : $F(x_i(k)) \Big/ \sum_{i=1}^{N} F(x_i(k))$

III. METHODES GLOBALES D'OPTIMISATION

III.3 Algorithmes génétiques

III.3.3 Evolution: Application des opérations de croisement et de mutation

1° Croisement

Sélection aléatoire de paires dans $M(k)$ => Parents

A. Cas de chromosomes binaires

Echange d'une partie du chromosome => Enfants

Exemple: Croisement en position 4 avec L=6 :

000 000 et 111 111 => 000 011 et 111 100

Avec sélection aléatoire de la position de la coupure

B. Cas d'éléments réels

Exemple: Enfant = Parent_1 + α (Parent_2 – Parent_1)

Avec α sélectionné aléatoirement dans [-d, 1+d] avec d > 0 (d=0,25)

III. METHODES GLOBALES D'OPTIMISATION

III.3 Algorithmes génétiques

2° Mutations

Perturbation aléatoire => Sortir d'un minimum local

A. Cas binaire : Remplacer aléatoirement des bits dans les chromosomes Enfants avec une probabilité P_m

B. Cas réel : Ajouter une perturbation aléatoire Δ (vecteur) aux descendants obtenus par croisement

III.3.4 Réinsertion: Construire une nouvelle population $P(k+1)$

Insérer les descendants des individus de $M(k)$ dans $P(k)$ => $P(k+1)$

1° Remplacer tous les individus de $P(k)$

2° Remplacer les individus les plus faibles de $P(k)$

3° Introduire les individus les plus forts de $M(k)$

III. METHODES GLOBALES D'OPTIMISATION

III.3 Algorithmes génétiques

Algorithme

Initialisation: k = 0 P(0)

Itérations:

1° Evaluer *P(k)*
2° Test arrêt
3° Sélection de *M(k)*
4° Evolution de *M(k)* par Croisements + Mutations
5° Réinsertion des descendants de *M(k)* => *P(k+1)*
6° *k = k + 1*