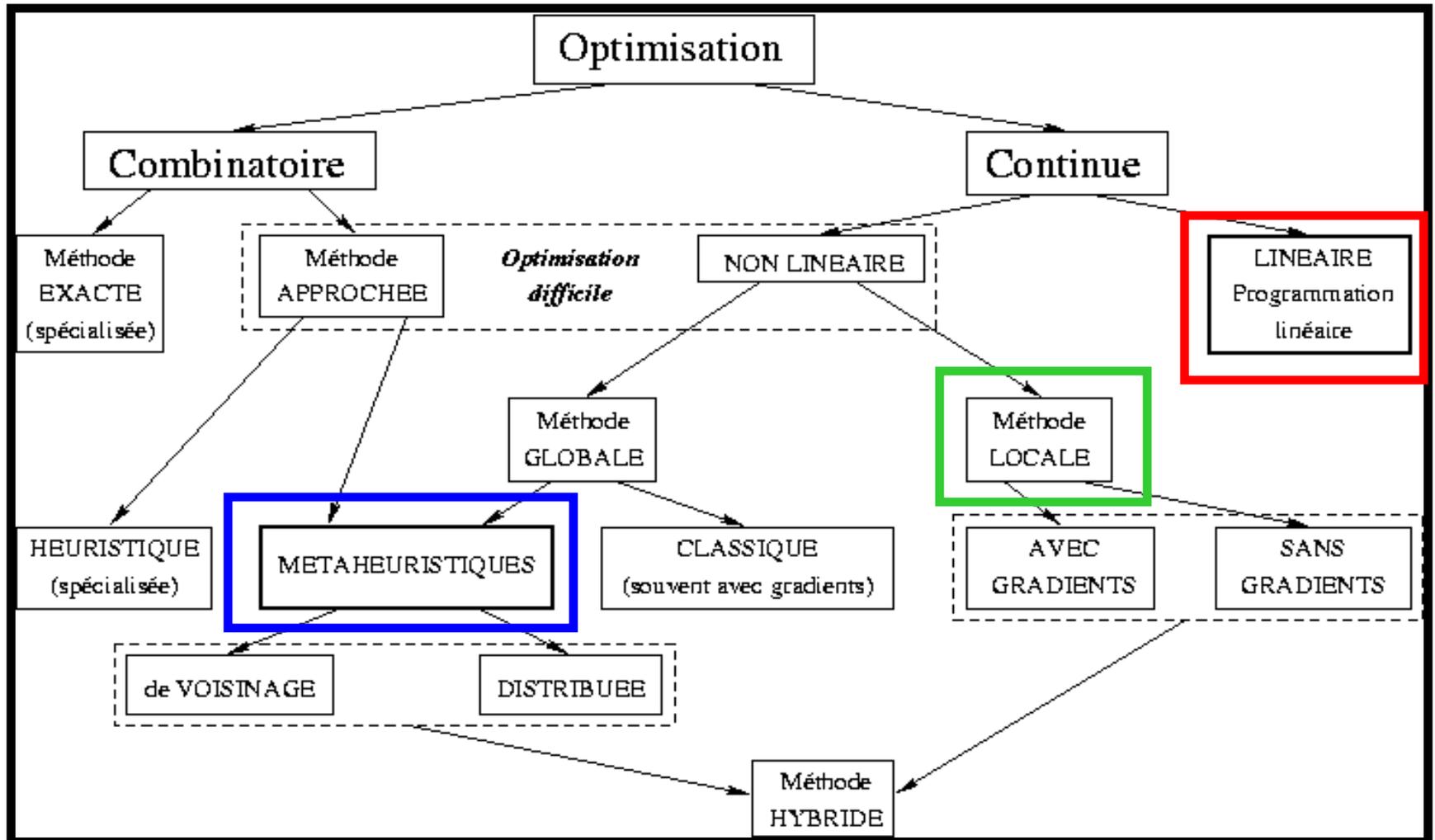


Recherche opérationnelle

CLASSIFICATION DES ALGORITHMES D'OPTIMISATION



Michalewicz (MZ) (n variables) :

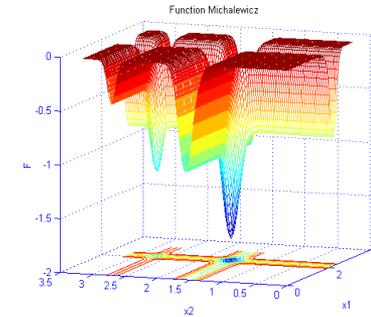
$$MZ(x) = \sum_{i=1}^n \sin(x_i) \left[\sin\left(\frac{i \cdot x_i^2}{\pi}\right) \right]^{20}$$

domaine de recherche : $0 \leq x_i \leq \pi$, $i = 1, n$

$n = 2$, 1 minimum global : $MZ(x^*) = -1.80$

$n = 5$, 1 minimum global : $MZ(x^*) = -4.687$

$n = 10$, 1 minimum global : $MZ(x^*) = -9.68$



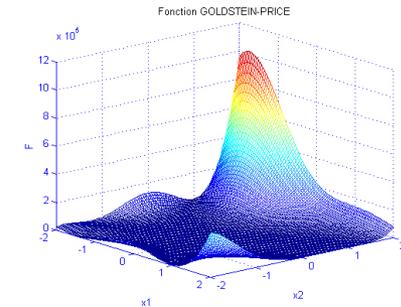
Goldstein-Price (GP) (2 variables) :

$$GP(x) = [1 + (x_1 + x_2 + 1)^2 * (19 - 14x_1 + 13x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] * [30 + (2x_1 - 3x_2)^2 * (18 - 32x_1 + 12x_1^2 - 48x_2 - 36x_1x_2 + 27x_2^2)]$$

domaine de recherche : $-2 \leq x_i \leq 2$, $i = 1, 2$;

4 minimums locaux

1 minimum global : $x^* = (-1, 0)$; $GP(x^*) = 3$.



De Jong F6 (DJ F6) (2 variables) :

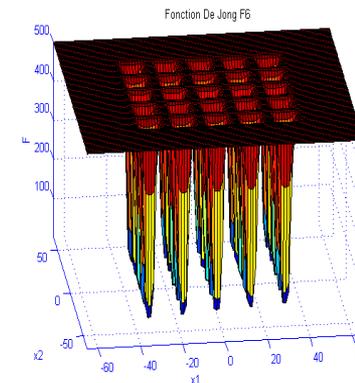
$$F6(x) = \frac{1}{0.002 + \sum_{i=0}^{24} \frac{1}{|4 - X_i|^{2.2} + |4 - Y_i|^{2.2}}}$$

$X_i = [16(i \bmod 5) - 2]$ & $Y_i = 16 [\text{integer}(i/5) - 2]$

domaine de recherché : $-65 \leq x_i \leq 65$, $i = 1, 2$;

25 minimums : $(32j, 32k)$, $j, k = -1, -0.5, 0, 0.5, 1$

minimum global : $x^* = (-32, -32)$; $F6(x^*) \cong 0.9980$



Exemple de fonction objectif

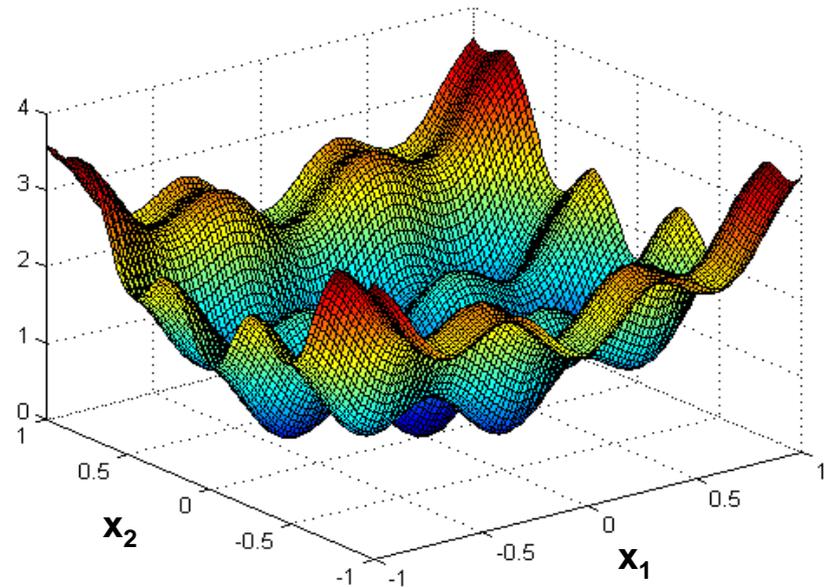
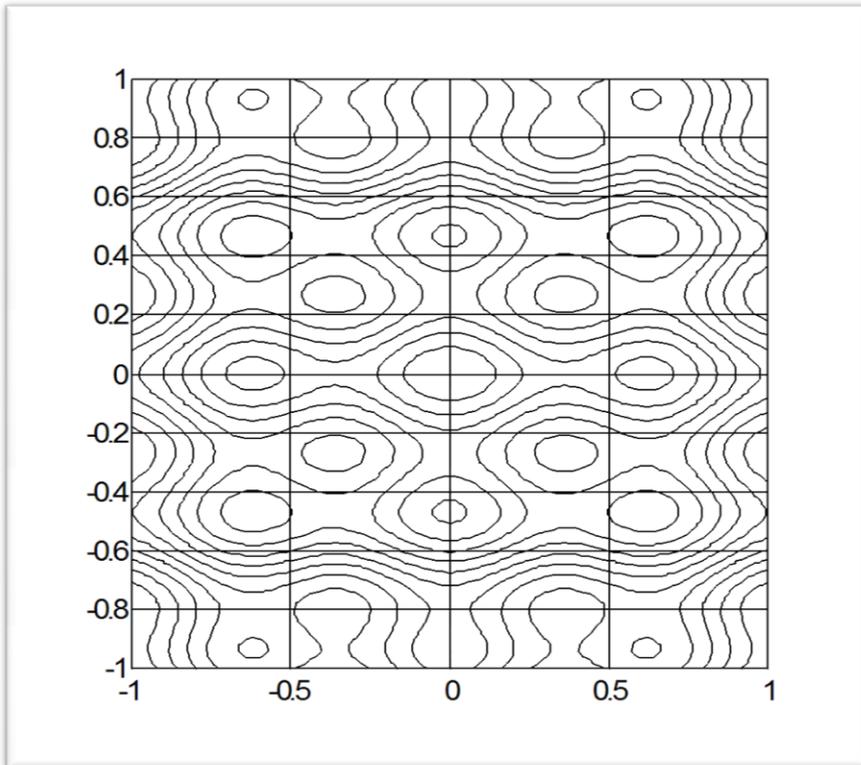
B2 (2 variables) :

$$B2(x_1, x_2) = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1) - 0.4\cos(4\pi x_2) + 0.7 ;$$

domaine de recherche : $-1 < x_j < 1$, $j = 1, 2$;

plusieurs minima locaux ;

1 minimum global : $(x_1, x_2)^* = (0, 0)$; $B2((x_1, x_2)^*) = 0$.



Recherche locale

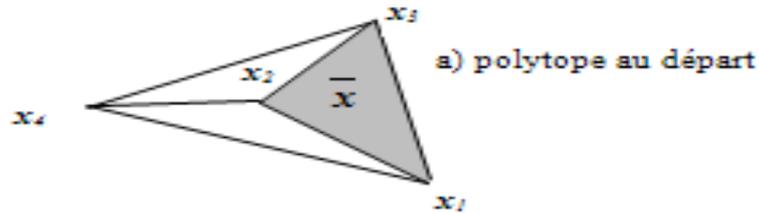
Cas de variables continues

- La méthode de Newton Raphson
- La méthode de la sécante
- La méthode dichotomique
- La méthode du gradient
- La méthode du polytope de Nelder-Mead

- La méthode du polytope de Nelder-Mead

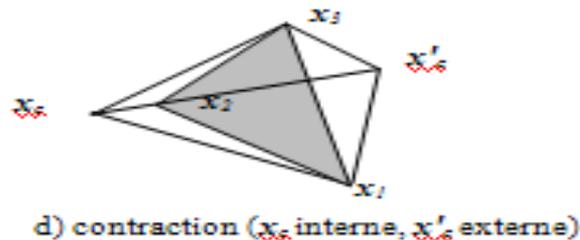
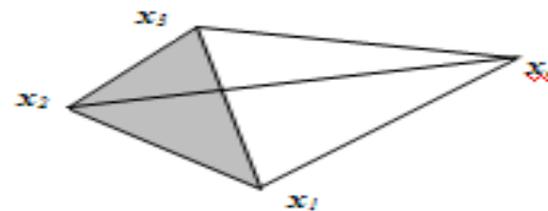
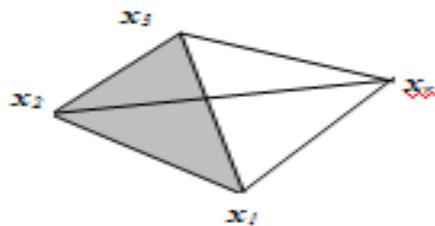
- Un polytope est une figure géométrique de $(n+1)$ points, n étant la dimension du problème
- Le polytope de départ est obtenu par le tirage aléatoire d'un point x_1 dans l'espace solution
- les autres points x_i sont choisis de manière à former une base, généralement une base orthogonale : $x_i = x_1 + \lambda e_i$,
- La méthode du gradient
- La méthode du polytope de Nelder-Mead

Les mouvements possibles dans la méthode du polytope de Nelder-Mead dans \mathbb{R}^3



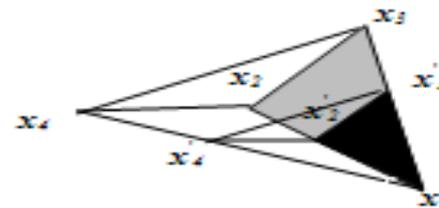
x_1 : représente le sommet où la valeur de la fonction objectif est la plus basse
 x_4 : représente le sommet où la valeur de la fonction objectif est la plus élevée

Soit \bar{x} le centre de gravité des points x_i : $\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$



$$x'_4 = \beta x_{n+1} + (1 - \beta) \bar{x},$$

$$x''_4 = \beta x_4 + (1 - \beta) \bar{x}.$$



Pseudocode de la méthode du polytope de Nelder-Mead

POLYT tableaux de taille $n+1$

POLYT := initialiser polytope

α, β, γ := initialiser

SIMPL := ordonner polytope pour obtenir $P_0 = \langle x_1, x_2, \dots, x_{n+1} \rangle$

REPETER

$$\bar{x} := \frac{1}{n} \sum_{i=1}^n x_i$$

réflexion ($x_r = (1 + \alpha)\bar{x} - \alpha x_{n+1}$)

$x^k := x_r$

SI réflexion acceptée ($f(x_r) < f(x_n)$) alors

SI ($f(x_r) < f(x_1)$)

expansion ($x_e = \gamma x_r + (1 - \gamma)\bar{x}$)

SI expansion acceptée ($f(x_e) < f(x_k)$) alors $x^k := x_e$

SINON $x^t := x_{n+1}$ (pour contraction interne)

SI ($f(x_r) \leq f(x^t)$) alors $x^t := x_r$ (pour contraction externe)

contraction ($x_c = \beta x^t + (1 - \beta)\bar{x}$)

SI contraction acceptée ($f(x_c) \leq f(x_n)$) alors $x^k := x_c$

SINON rétrécissement du simplexe

$$x_j := \frac{(x_j + x_1)}{2} \quad \text{pour } j = 2, \dots, n$$

$$x^k := \frac{(x_{n+1} + x_1)}{2}$$

Trier $f(x_1), f(x_2), \dots, f(x^k)$ pour obtenir $P_k = \langle x_1, x_2, \dots, x_{n+1} \rangle$

JUSQU'A conditions d'arrêt satisfaites

Heuristiques

- RECUIT SIMULÉ
- RECHERCHE AVEC TABOUS
- ALGORITHMES GÉNÉTIQUES
- COLONIES DE FOURMIS ARTIFICIELLES
- ESSAIM PARTICULAIRE

Hypothèses

- problème mono-objectif
- fonction objectif $f(\mathbf{x})$, à minimiser
- seules contraintes : « contraintes de boîte » :

$$x_i^{\text{MIN}} < x_i < x_i^{\text{MAX}}$$

Difficultés spécifiques aux problèmes continus :

(on parle aussi de « problèmes difficiles », mais sans référence ici à la théorie de la complexité...)

- pas d'expression analytique de f
- f « bruitée » :
 - bruit expérimental (exploitation de mesures)
 - « bruit de calcul » numérique
(utilisation d'un simulateur de circuits électroniques)

⇒ gradients
de f non
accessibles

- **f** comporte des non-linéarités
- Nombreux minima locaux
- **hétérogénéité des domaines de définition** des différentes variables
variables de **gammes très étendues** (plus de 10 décades)

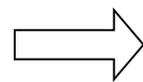
les termes « global » / « local » ne caractérisent pas ici le type de mouvement utilisé (\Rightarrow par exemple, recuit simulé : à la fois local, par son mécanisme, et global, par sa finalité...)

⇒ **recours aux métaheuristiques**,
qui sont toutes à la fois « directes » et « globales »



- l'**aspect « direct »** des métaheuristiques, lié à leur origine combinatoire, n'a pas d'attrait particulier dans le cas discret
- c'est, au contraire, un **avantage très important**, dans le cas continu « difficile »

sauf exception (Essais particulières),
métaheuristiques conçues / cadre discret



nécessité d'une « adaptation »
aux problèmes continus

Principe d'un algorithme stochastique d'optimisation globale

- la perturbation aléatoire :
 - toutes les coordonnées du vecteur solution courant, ou une partie seulement, sont perturbées, plusieurs coordonnées à la fois ou une par une,
 - la transformation suit une loi de distribution particulière dans un voisinage du point courant, par exemple une loi de distribution uniforme,
 - le ou les nouveaux points sont générés selon une loi qui dépend des points précédents et/ou des valeurs précédentes de la fonction objectif, ou ne dépend que du point courant,
 - etc.
- le critère d'acceptation :
 - le ou les nouveaux points sont acceptés selon une certaine loi de probabilité,
 - le ou les nouveaux points sont acceptés selon certaines conditions,
 - le ou les meilleurs points sont conservés pour l'étape suivante,
 - une recherche locale est effectuée à partir du ou des nouveaux points,
 - etc.
- les critères d'arrêt :
 - ils peuvent être liés à la qualité du minimum obtenu :
 - la procédure est arrêtée quand il n'y a plus d'amélioration de la solution après un certain nombre d'itérations,
 - elle s'arrête quand les perturbations ne dépassent plus un certain seuil.
 - ils peuvent être liés à des coûts de calcul ; la procédure s'arrête après :
 - un nombre maximal d'évaluations de la fonction objectif,
 - un nombre d'itérations fixé,
 - un certain temps de calcul fixé,
 - etc.

Les principales métaheuristiques modernes

- La difficulté majeure réside dans la détermination de la taille optimale du pas de discrétisation et de sa direction (résultant des variables sur lesquelles on agit)
- Harmonisation des différents domaines de définition des différentes variables de décision
- Le choix de la loi de discrétisation est un compromis entre deux situations extrêmes :
 - si le pas est trop petit, on n'explore qu'une région limitée de l'espace des configurations, et l'algorithme risque d'être piégé dans un minimum local ;
 - si le pas est trop grand, la recherche devient quasiment aléatoire.
- La meilleure solution peut consister à élaborer une topologie adaptative.

L'algorithme de recuit simulé : historique

Le recuit simulé (Simulated Annealing)

Expériences réalisées par Metropolis et al. dans les années 50 pour simuler l'évolution de ce processus de recuit physique (Metropolis53).

L'utilisation pour la résolution des problèmes d'optimisation combinatoire est beaucoup plus récente et date des années 80 (Kirkpatrick83,Cerny85).

Le recuit simulé est la première métaheuristique qui a été proposée.

Recuit simulé

- L'analogie exploitée par le recuit simulé consiste à considérer une fonction f à minimiser comme fonction d'énergie
- 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.
 - Sinon cette perturbation est acceptée tout de même avec une probabilité

$$p = \exp\left(\frac{-\Delta f}{T}\right)$$

Règle de Metropolis

```
Fonction CritereMetropolis ( $\Delta f$ , T) retourne Booleen
  SI  $\Delta f \leq 0$ 
    Retourne Vrai
  SINON
     $p = \frac{\exp(-\Delta f)}{T}$ 
     $r :=$  solution aléatoire entre [0, 1]
    SI  $r \leq p$ 
      Retourne Vrai
    SINON
      Retourne Faux
  FIN SI
FIN SI
FIN Fonction
```

Un voisin qui améliore ($\Delta < 0$) ou à coût égal ($\Delta = 0$) est toujours accepté.

Une dégradation faible est acceptée avec une probabilité plus grande qu'une dégradation plus importante.

Paramètre de température

Dans la fonction $\text{CriterreMetropolis}(\Delta f, T)$, le paramètre T (température) est un réel positif.

La température permet de contrôler l'acceptation des dégradations :

- Si T est grand, les dégradations sont acceptées avec une probabilité plus grande.
- A la limite, quand T tend vers l'infini, tout voisin est systématiquement accepté.
- Inversement, pour $T=0$, une dégradation n'est jamais acceptée.

La fonction qui spécifie l'évolution de la température est appelé le schéma de refroidissement.

Recuit simulé

$x :=$ solution aléatoire

$f(x) :=$ valeur de la fonction

$f_{min} := f(x)$

$x_{min} := x$

$T :=$ initialiser température (assez élevée)

REPETER

 REPETER

 générer un voisin $s(x) \in$ voisinage $S(x)$

$\Delta f = f(s(x)) - f(x)$

 SI(CritereMetropolis ($\Delta f, T$))

$f(x) = f(s(x))$

$x = s(x)$

 SI $f(x) < f_{min}$

$f_{min} := f(x)$

$x_{min} := x$

 FIN DE SI

 FIN DE SI

 JUSQU'A condition d'arrêt sur palier (équilibre thermodynamique)

$T :=$ décroître température

JUSQU'A conditions d'arrêt satisfaites

Réglage des paramètres du recuit simulé

Dans le cas du recuit simulé classique (avec refroidissement), le réglage des paramètres n'est pas évident.

Pour régler les paramètres, il peut être utile d'observer le pourcentage d'acceptation (acceptance rate) au cours de la recherche = nombre de mouvements réellement exécutés / nombre d'itérations

Température initiale : Si la température initiale est trop élevée, le début de la recherche ne sert à rien.

Critère d'arrêt : Il est inutile de poursuivre si le pourcentage d'acceptation devient très faible, la fonction d'évaluation cesse d'évoluer

Algorithme de recuit simulé

$x :=$ solution aléatoire

$f(x) :=$ valeur de la fonction

$f_{min} := f(x)$

$x_{min} := x$

$T :=$ initialiser température (assez élevée)

REPETER

$nb_moves := 0$

 Pour $i := 1$ à nb_iter_palier

 générer un voisin $s(x) \in$ voisinage $S(x)$

$\Delta f = f(s(x)) - f(x)$

 SI(CritereMetropolis ($\Delta f, T$))

$f(x) = f(s(x))$

$x = s(x)$

$nb_moves := nb_moves + 1$

 SI $f(x) < f_{min}$

$f_{min} := f(x)$

$x_{min} := x$

 FIN DE SI

 FIN DE SI

$acceptance_rate := nb_moves / i$

FIN POUR

$T :=$ décroître température

JUSQU'A conditions d'arrêt satisfaites

Schéma de refroidissement

La fonction qui spécifie l'évolution de la température est appelé le schéma de refroidissement (cooling schedule).

Dans le recuit simulé standard la température décroît par paliers.

- On peut aussi utiliser d'autres schémas de refroidissement :
- On peut faire décroître la température à chaque itération.
- On utilise parfois une température constante (algorithme de Metropolis).
- On peut utiliser des schémas plus complexes, dans lesquels la température peut parfois remonter.

Contrôle de l'algorithme

L'efficacité du recuit simulé dépend fortement du choix de ses paramètres de contrôle, dont le réglage reste très empirique.

Les principaux paramètres de contrôle sont les suivants :

- la valeur initiale de la température,
- la fonction de décroissance de la température,
- le critère de changement de palier de température,
- les critères d'arrêt.

la valeur initiale de la température

- Une des méthodes est basée sur l'observation de la variation moyenne de la fonction f .
- A partir d'une solution initiale x_0 on génère, par transformations élémentaires aléatoires, un certain nombre de solutions x'_0 (environ 50 à 100)
telles que $f(x'_0) > f(x_0)$,
- et on calcule la variation moyenne $\langle |\Delta f| \rangle_{init}$.
- Une température initiale T_{init} est calculée de façon à accepter au départ une certaine proportion P_{nit} de mouvements dégradant la fonction f .
- Pour une température initiale "moyenne", la valeur de P_{init} est de 0.5.
- La valeur de T_{init} est déduite de la formule suivante :

$$P_{init} = \exp \frac{- \langle \Delta f \rangle_{init}}{T_{init}}$$

la fonction de décroissance de la température

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

$$T_k = \frac{\mu}{\text{Log}(1+k)}$$

où k est le nombre de paliers de température effectués, et μ une constante positive.

En pratique, on adopte souvent une décroissance géométrique $T_{k+1} = \alpha T_k$, avec $(0 < \alpha < 1)$, car la loi précédente induit un temps de calcul prohibitif.

Le critère de changement de palier de température, Les critères d'arrêt.

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

Longueur d'un palier

Un paramètre borne le nombre d'essais (itérations) par palier

Un autre paramètre borne le nombre changements (mouvements) pour le début de la recherche

Critère d'arrêt implantation de Johnson et al

Un compteur sert à déterminer si l'algorithme stagne. Le compteur est fixé à zéro au début La recherche s'arrête quand le compteur atteint un certain seuil.

A la fin d'un palier, le compteur est

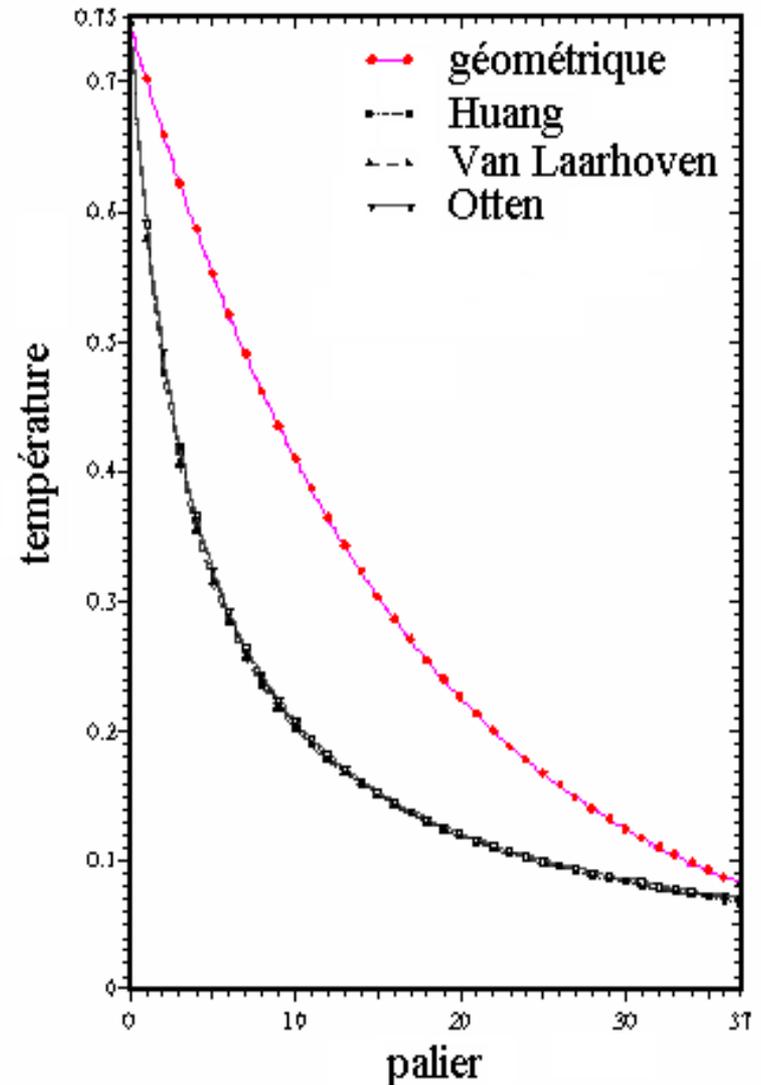
- incrémenté si le pourcentage d'acceptation est inférieur à un seuil.
- remis à zéro si la qualité de la meilleure solution a évolué au cours du palier

Décroissance adaptative de la température

- Etude récente, théorique et expérimentale

- Principaux résultats :

• plusieurs **lois adaptatives classiques**, ayant des origines et des expressions mathématiques bien différentes, sont, en pratique, **équivalentes** :



- on montre que ce sont des **cas particuliers de la loi** générique :

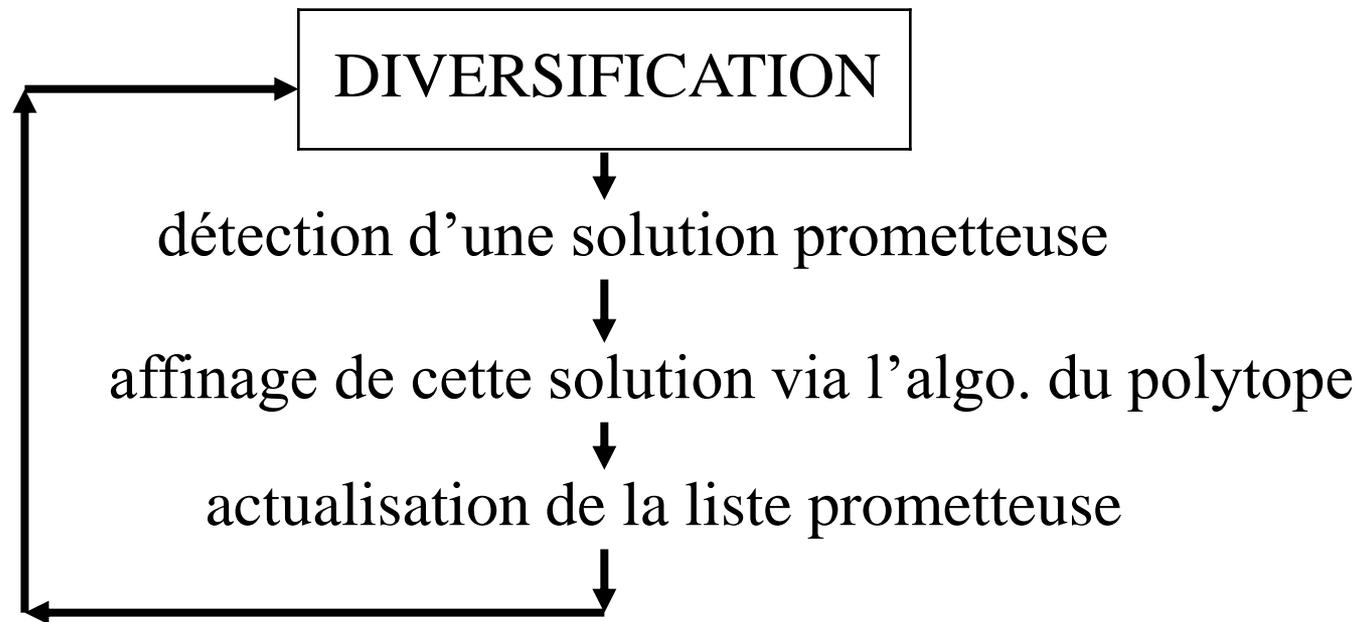
$$T_{k+1} = T_k \cdot \left[1 - T_k \cdot \Delta(T_k) / \sigma^2(T_k) \right]$$

où : $\sigma^2(T_k) = \langle f_{T_k}^2 \rangle - \langle f_{T_k} \rangle^2$

$\Delta(T_k)$: dépend de la loi adaptative choisie

- le **choix le plus simple** : $\Delta(T_k) = C^{te}$ ne correspond à aucune loi classique
- nous avons montré que ce choix peut être formateur, face à un problème nouveau

- Les phases de diversification et d'intensification sont alternées :



Pseudocode de l'algorithme hybride recuit simulé-polytope (ESASS)

x := solution aléatoire

f(x) := valeur de la fonction

f_{min} := **f(x)**

x_{min} := **x**

DIVERSIFICATION

T := initialiser la température (assez élevée)

REPETER

REPETER

générer une perturbation **s(x)** ∈ voisinage **S(x)**

appliquer la règle de Metropolis

SI **f(x)** < **f_{min}**

f_{min} := **f(x)**

x_{min} := **x**

FIN DE SI

JUSQU'A équilibre thermodynamique atteint

T := décroître la température

JUSQU'A conditions d'arrêt de la diversification satisfaites

INTENSIFICATION

réduction du domaine de recherche

construction du polytope initial

REPETER

mouvements géométriques

JUSQU'A condition d'arrêt satisfaite