Algorithme du Recuit Simulé

PRÉSENTÉ PAR: LARISSA DENAKPO

larissa.denakpo@imsp-uac.org

Introduction

- Optimisation multi-objectif: discipline où on cherche à optimiser simultanément plusieurs objectifs d'un même problème
- Pour y arriver: emploie plusieurs méthodes
- Certaines se basent sur les méta-heuristiques
- Exemple de méta-heuristique: Recuit Simulé

Plan

- Historique
- Recuit thermique
- Application du recuit thermique à l'optimisation
- Algorithme du recuit simulé
- Recuit simulé multi-objectif
  - Méthode P.A.S.A
  - Méthode M.O.S.A
- Application
- Conclusion

Historique

- Basé sur l'algorithme de Metropolis en 1953: comportement de la matière dans le processus du recuit utilisé en métallurgie, but d'atteindre un état d'équilibre où l'énergie est minimale

- Conçu par trois chercheurs de la société IBM, S. Kirkpatrick, C. D. Gelatt et M. P. Vecchi en 1983

Recuit thermique(1)

- Processus métallurgique

- Permet à un métal de retrouver une structure proche du cristal parfait (plus souple, plus flexible, moins d'irrégularités).

- l'état cristallin correspond au minimum d'énergie de la structure atomique du métal

Recuit thermique(2)

- le métal est porté à une température élevée: permet aux atomes de se déplacer
- le métal est refroidi: les atomes se positionnent de façon à minimiser leur énergie
- Le refroidissement reste lent de manière à ce que les atomes aient le temps de s'ordonner régulièrement

Application du recuit thermique à l'optimisation(1)

- Idée: simuler numériquement une opération de recuit thermique

- Principe:
  - Système composé de N éléments
  - A chaque configuration ou état du système est associée une fonction à optimiser

Application du recuit thermique à l'optimisation(2)

- On recherche la configuration qui minimise la fonction
- Par analogie avec le processus physique, la fonction à minimiser deviendra l'énergie E du système.
- On introduit également un paramètre fictif, la température T du système.

Algorithme du recuit simulé(1)

- On part d'un état initial pris au hasard dans l'espace des états possibles
- A cet état correspond une énergie initiale

$$E = E_0$$

- Au départ de l'itération, on fixe une température initiale élevée

$$T = T_0$$

Algorithme du recuit simulé(2)

- On choisi au hasard un voisin de l'état actuelle qui change l'énergie du système d'une quantité ΔE
- Si l'énergie diminue, l'état voisin devient le nouvel état courant
- Si l'énergie augmente, on effectue le changement avec une probabilité

$$p = e^{-\frac{\Delta E}{T}} \text{ (critère de Metropolis)}$$

Algorithme du recuit simulé(3)

- La température décroit lentement au fur et à mesure des itérations
- L'itération se poursuit tant que l'énergie du système diminue.
- On arrête lorsque les diminutions de température restent inefficaces ou lorsque la température atteint un seuil minimal
- La température peut diminuer toutes les L itérations ou de façon continue

Pseudo-code

- *init* T *(température initiale)*
- *init* x *(point de départ)*
- *init* T_{min} *(température minimale)*

- *while(not(end))*

  $y = Voisin\ (x)$

  $\Delta E = E(y) - E(x)$

  if $\Delta E < 0$ then $x = y$

  else if $alea(0,1) < e^{\frac{-\Delta E}{T}}$ then $x = y$

  $T = a(T)$

  if $T < T_{min}$ then end(while)

  Repeat (while)

Recuit simulé multi-objectif

- (P): $\min f(x) = (f_1(x), \ldots, f_N(x))$
  $sc: \quad x \in D$
- On dira que y domine x si:
  - y est au moins aussi bon que x dans tous les objectifs et,
  - y est strictement meilleur que x dans au moins un objectif.
- On notera $y \ll x$ si y domine x

Méthode P.A.S.A (*Pareto Archived Simulated Annealing*)(1)

- Utilise une fonction d'agrégation des fonctions objectifs

- Couplé à un système d'archivage de solutions non dominées

Méthode P.A.S.A (*Pareto Archived Simulated Annealing*)(2)

- On suppose que les fonctions objectifs sont positives
- Pour se ramener à un problème de minimisation mono-objectif, on peut donc utiliser la fonction d'agrégation:

$$G(x) = \sum_{i=1}^{N} \ln(f_i(x))$$

Méthode P.A.S.A (*Pareto Archived Simulated Annealing*)(3)

- La variation relative moyenne des fonctions objectif entre le point courant x et le point à tester x':

$$\Delta G = G(x') - G(x) = \sum_{i=1}^{N} \ln\left(\frac{f_i(x')}{f_i(x)}\right)$$

- On suit ensuite la même procédure que le recuit simulé original

Méthode P.A.S.A (*Pareto Archived Simulated Annealing*)(4)

- si $\Delta G > 0$, la nouvelle solution x' détériore, en moyenne relative, l'ensemble des fonctions objectif
- si $\Delta G < 0$, la nouvelle solution x' améliore, en moyenne relative, l'ensemble des fonctions objectif

Méthode P.A.S.A (*Pareto Archived Simulated Annealing*)(5)

- Pour archiver les solutions non dominées, on utilise une “archive” A de taille variable
  - Si $\exists\ y \in A /\ \ y \ll x'$ alors x' n’est pas archivée
  - Si $\exists\ y \in A /\ \ x' \ll y$ alors x' remplace y dans l’archive
  - Si $\forall\ y \in A /\ \ y$ ne domine pas x' alors x' est archivée et on retire les solutions dominées par x'

Méthode P.A.S.A (*Pareto Archived Simulated Annealing*)(6)

- Pour que la recherche se fasse sur toute la surface de compromis, il est nécessaire de relancer régulièrement la recherche à partir d'un point choisi au hasard, au sein de l'archive.
- Cette étape du choix de l'individu est appelée *return to base*.

Pseudo-Code

- *init T (température initiale)*
- *init x (point de départ)*
- *Init $A = \{x\}$*
- *init T_{min} (température minimale)*
- *while(not(end))*

  For i = 1:L

  $y = Voisin\ (x)$

  $\Delta G = G(y) - G(x)$

  if $\Delta G < 0$ then $x = y$

  else if $alea(0,1) < e^{\frac{-\Delta G}{T}}$ then $x = y$

  end for

Application du Principe de non-dominance pour l'archivage de x dans A

Return to base : lorsque c'est nécessaire, $x = a$ (a choisi aléatoirement dans A)

$T = a(T)$

if $T < T_{min}$ then end(while)

Repeat (while)

La méthode M.O.S.A (*Multiple Objective Simulated Annealing*)(1)

- Utilise le recuit simulé pour rechercher la surface de compromis
- On défini une suite de fonctions donnant la probabilité d'acceptation d'une mauvaise solution, pour chaque fonction objectif :

$$\pi_k = \begin{cases} e^{-\frac{\Delta f_k}{T_n}} & si\ \Delta f_k > 0 \\ 1 & si\ \Delta f_k \leq 0 \end{cases}$$

La méthode M.O.S.A (*Multiple Objective Simulated Annealing*)(2)

- T_n la température à l'itération *n*,
- f_k *la* $k^{\text{ième}}$ fonction objectif,
- x solution obtenue à l'itération *n*,
- y point voisin de x considéré,
- $\Delta f_k \ = f_k(y\ ) - f_k(x)$

La méthode M.O.S.A (*Multiple Objective Simulated Annealing*)(3)

- Après le calcul de chaque probabilité on les agrège
- Soit on prend le produit de ces probabilités:

$$t(\Pi, \lambda) = \prod_{k=1}^{N} (\pi_k)^{\lambda_k}$$

avec

Π l'ensemble des $\pi_k, k = \{1, \cdots, N\}$

λ l'ensemble des $\lambda_k, k = \{1, \cdots, N\}$

La méthode M.O.S.A (*Multiple Objective Simulated Annealing*)(4)

- Soit on prend la plus petite probabilité :

$$t(\Pi, \lambda) = \min_{k \in 1 \dots N} (\pi_k)^{\lambda_k}$$

- λ_k *est* un coefficient de pondération relatif à une fonction objectif. Il permet de prendre en compte un certain ordre entre les différents objectifs.

La méthode M.O.S.A (*Multiple Objective Simulated Annealing*)(5)

- On agrège, maintenant, les fonctions objectifs de la façon suivante

$$F(x) = \sum_{k=1}^{N} \lambda_k f_k(x)$$

- Condition d'acceptation d'une solution:

  On calcul toujours ΔF et les conditions d'acceptations restent les mêmes que précédemment avec comme probabilité $t(\Pi, \lambda)$

La méthode M.O.S.A (*Multiple Objective Simulated Annealing*)(6)

- Pour chaque ensemble λ de coefficients de pondération considérés, un ensemble de solutions est calculé.
- Chaque λ va "remplir" une partie de la surface de compromis.
- Les solutions sont alors réunies, puis les individus dominés sont éliminés

Application (méthode PASA)

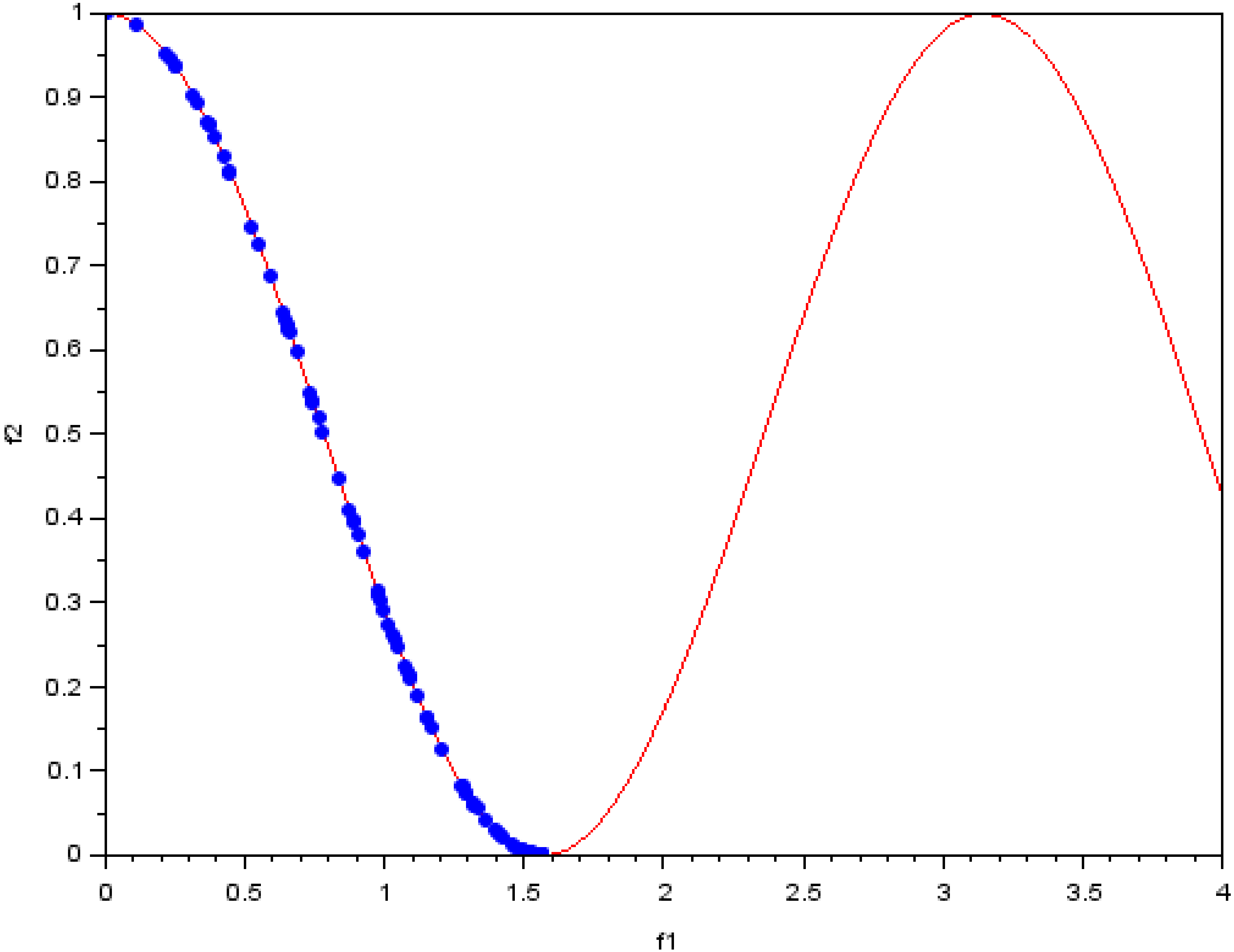
$$\min \begin{array}{cc} f_1(x) = x, & f_2(x) = (\cos x)^2 \\ s.c & x \in]0,4] \end{array}$$

$T_0 = 10$

$T_{min} = 10^{-10} T_0$

$x_0 = 3$

$L = 50$



f2
f1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
0.5
1
1.5
2
2.5
3
3.5
4

Conclusion(1)

- Le recuit simulé permet de résoudre des problèmes très complexes
- Ne s'appuie sur aucunes propriétés intrinsèques aux modèles
- Le résultat dépend fortement de la méthode utilisée pour générer un nouvel état aléatoire

Conclusion(2)

- La méthode MOSA prend plus en considération l'aspect multi-objectif du problème d'optimisation que la méthode PASA car dans cette méthode, on commence par calculer des probabilités par fonction objectif en leur attribuant des coefficients

Merci pour votre attention