



Le recuit simulé

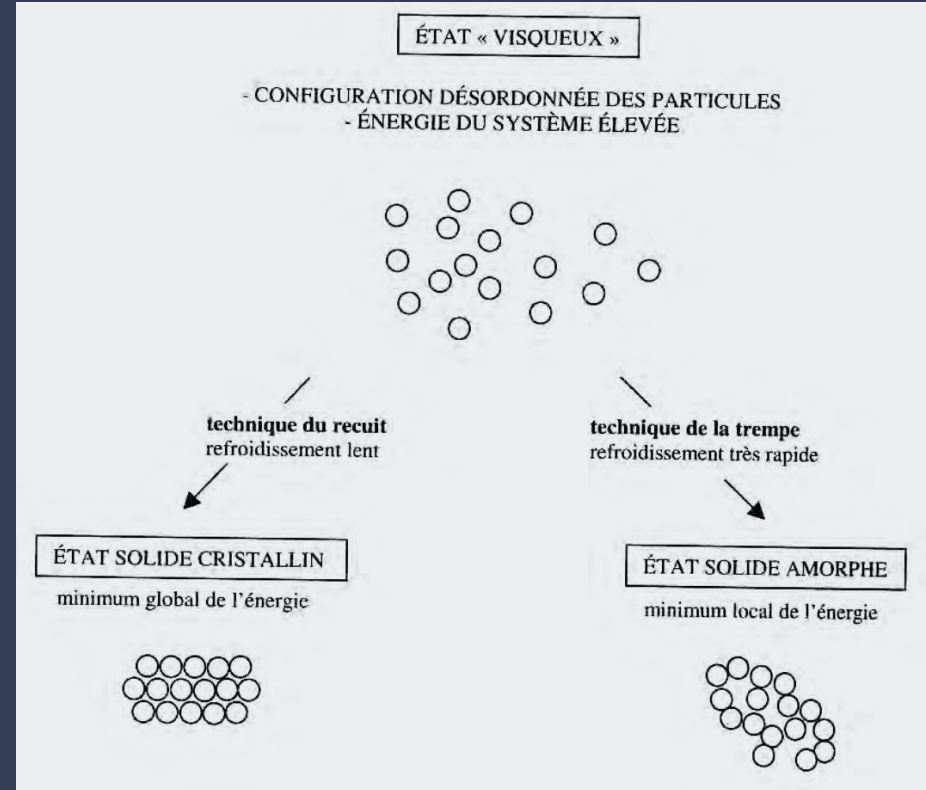
I. Historique

- Inventé par Kirkpatrick, Gelatt et Vecchi en 1983 (IBM)
- Inspiré d'un processus métallurgique : le recuit
- Recuit (*annealing*) : cycle de chauffage, maintien en température puis refroidissement contrôlé permettant de modifier les caractéristiques d'un métal



II. Analogie physique

Refroidissement lent → temps suffisamment longs pour que les atomes se redistribuent au fur et à mesure que leur mobilité diminue
(1.)





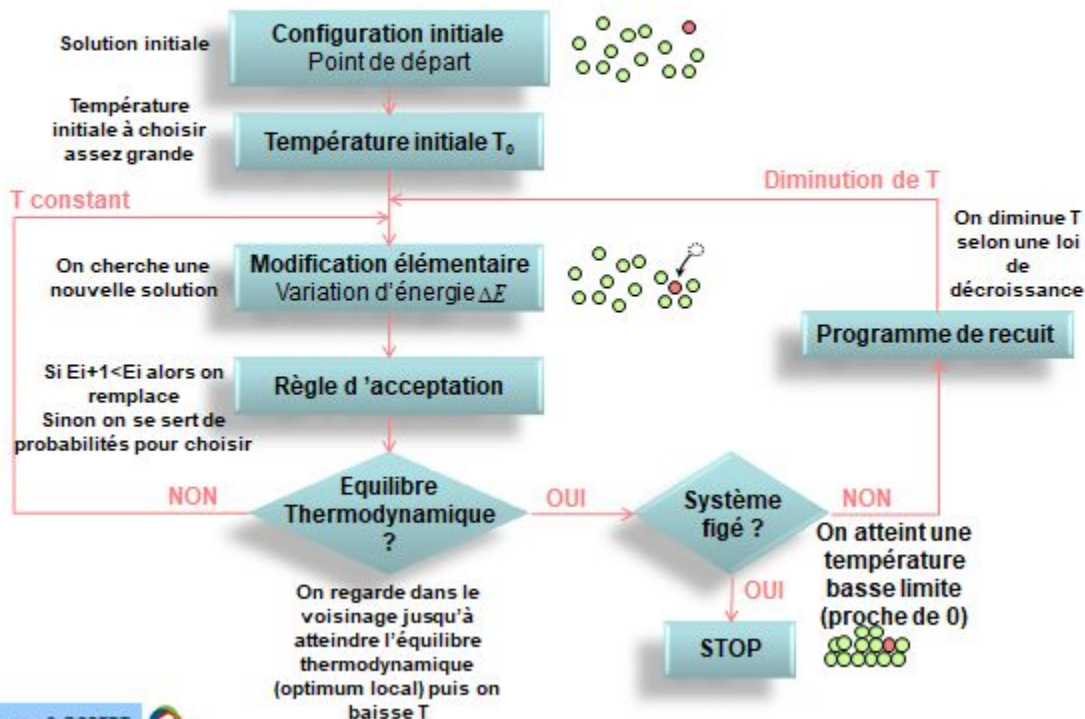
II. Analogie physique

Problème d'optimisation	Problème physique
Fonction objectif	Energie (libre)
Paramètres du problème	Coordonnées des particules
Configuration optimale	États de basse énergie

III. Principe de la méthode



Organigramme de l'algorithme RS



III. Principe de la méthode

Règles d'acceptation de modification de la solution courante

La règle d'acceptation va permettre de choisir entre la solution actuelle x_i et une nouvelle valeur $x_{(i+1)}$ qu'on aura choisie.

On utilise la règle de l'algorithme de Metropolis-Hastings (2.) :

- 1) Si $f(x_{(i+1)}) \leq f(x_i)$: on prend $x_{(i+1)}$ à la place de x_i car c'est une meilleure solution
- 2) Si $f(x_{(i+1)}) > f(x_i)$: on prend $x_{(i+1)}$ avec une probabilité $\propto e^{(-\Delta E/T)}$ où $\Delta E = E_{(i+1)} - E_i = f(x_{(i+1)}) - f(x_i)$ et T est la température actuelle

Au début de la méthode lorsque T est grand il y aura donc plus de chances de se déplacer vers une valeur lointaine en terme d'énergie, et plus T baisse moins ces déplacements seront probables.

$$T \nearrow \longrightarrow e^{(-\Delta E/T)} \nearrow$$

Amélioration possible : garder en mémoire la meilleure solution trouvée



III. Principe de la méthode

Génération des nouvelles solutions

Le but est de choisir une nouvelle valeur $x(i+1)$ aléatoire pour remplacer x_i , il y a plusieurs méthodes :

1) On modifie x_i avec un pas d dépendant de la taille n du problème : $x(i+1) = x_i + a \cdot d$ avec $-1 < a < 1$ et $d = \frac{1}{4}$ du domaine

2) On choisit une valeur aléatoirement dans un intervalle autour de x_i dépendant de la taille N du problème : $N/3$ par exemple

3) On code les valeurs en binaire, on inverse aléatoirement deux chiffres du code binaire de l'élément actuel

III. Principe de la méthode

Configuration initiale

- La température : il faut débiter avec une température élevée qui permet d'explorer de nombreuses solutions en s'échappant des minimums locaux et elle fixe la probabilité d'accepter ou de refuser les solutions défavorables à l'optimisation de la fonction f . Une température trop petite engendre le risque d'être bloqué sur un minimum local.

Exemple : évaluer un ΔE moyen (100 itérations), on a : $T_0 = - \langle \Delta E \rangle / \log(\theta_0)$
avec θ la probabilité d'acceptation, $\theta=0.5$



- La solution initiale : choix aléatoire ou solution trouvée à partir d'une heuristique (descente du gradient ou algorithme glouton)



III. Principe de la méthode

Equilibre thermodynamique

On choisit un paramètre pour définir l'équilibre thermodynamique, en général on choisit un nombre limite d'itérations dans l'algorithme de Metropolis pour la même température.

On peut compter toutes les itérations ou uniquement celles où il y a un changement.

Exemples (5.) :

- Multiple de la taille du problème, compliqué si trop de données, 100N si on compte toutes les itérations et 12N si on ne compte que les changements.
- $(N/2)^2$

III. Principe de la méthode

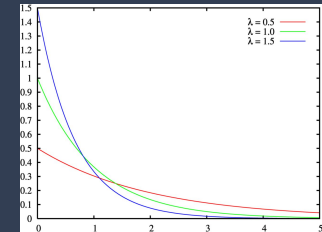
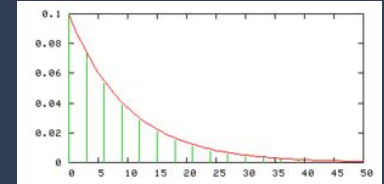
Décroissance de la température

On souhaite un refroidissement lent il est donc capital de bien choisir cette loi

- Décroissance par paliers : on diminue T lorsque l'équilibre thermodynamique est atteint
- Décroissance sans paliers : On diminue la température à chaque itération de l'algorithme de Metropolis-Hastings

Type de lois de décroissance :

- Loi géométrique : $T = K \cdot T_0$ où $0 < K < 1$, en général K très proche de 1, $K=0,99$
- Loi exponentielle : $T = e^{(-T_0/\tau)}$ avec τ suffisamment grand pour décroître lentement (3.)
- Loi discrète : $T = T_0 - dT_0$ où d est très petit, $d=0.01$



III. Principe de la méthode

L'arrêt de l'algorithme

L'algorithme s'arrête lorsqu'on atteint une température proche de zéro que l'on définit ou lorsque l'on effectue aucune amélioration de la solution courante sur un palier de température.

$T_{lim} \approx 0$

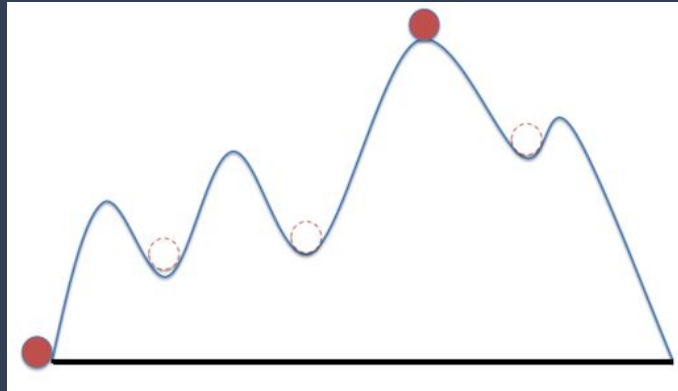
x constant pour un palier de température

Pour définir cette température limite, on peut choisir d'atteindre un certain pourcentage de T_0 : $T_{lim} = 10^{-6}T_0$ par exemple.

III. Principe de la méthode

Convergence de l'algorithme

La théorie démontre que la méthode converge presque sûrement (i.e. avec une probabilité 1) vers un minimum global si la température ne décroît pas plus vite que $C/\log(t)$, où C est une constante dépendant de la profondeur des « puits d'énergie ». (5.)





IV. Extension de la méthode

Recuit simulé continu

Méthode initialement utilisée pour l'optimisation combinatoire → valeurs discrètes

En continu → plus de longueur minimale de pas imposée → pas devient un paramètre de contrôle

Pas: ni trop petit (piège dans une région limitée avec de faibles améliorations), ni trop grand (nouvelles solutions rarement acceptées, proche d'une exploration purement aléatoire).

But: adapter le pas en fonction de l'espace.



IV. Extension de la méthode

Recuit simulé continu

Procédure:

Initialisation d'un pas en fonction du domaine de la fonction ($\frac{1}{4}$ ou $\frac{1}{3}$ usuellement).

Définition d'une loi de calcul du mouvement: $x'_i = x_i \pm y \cdot STEP_i$
avec y un nombre aléatoire tiré entre 0 et 1.

Définition d'une loi d'ajustement du pas pour s'adapter à la configuration (ex: taux d'acceptation)

Définition de la fréquence d'ajustement (ex: tous les changements de température)

Définition des variables concernées par le mouvement (N/3 au hasard)



IV. Extension de la méthode

Recuit simulé pour optimisation multiobjectif:

Méthodes existantes: MOSA (Multiple Objective Simulated Annealing) et PASA (Pareto Archived Simulated Annealing) (7.)

Principe:

- PASA: utilisation d'une fonction d'agrégation pour se ramener à une fonction mono objectif.

Application de l'algorithme du recuit simulé.

A chaque mouvement validé, on teste si la solution appartient au front de Pareto. Si oui, elle est stockée dans une archive et remplace les solution qu'elle domine.

Cette archive contient donc l'ensemble des solutions non dominées. (On peut initialiser l'archive avec une première optimisation multiobjectifs)



IV. Extension de la méthode

Recuit simulé pour optimisation multiobjectif:

- MOSA: Définition des probabilités d'acceptation pour chaque fonction objectif.
Agrégation de ces probabilités (produit ou minimum des probabilités pondérées par un coefficient propre à la fonction objectif)
Sélection de la solution en appliquant le recuit simulé sur une somme pondérée des fonctions objectifs



V. Avantages et inconvénients de la méthode

Avantages:

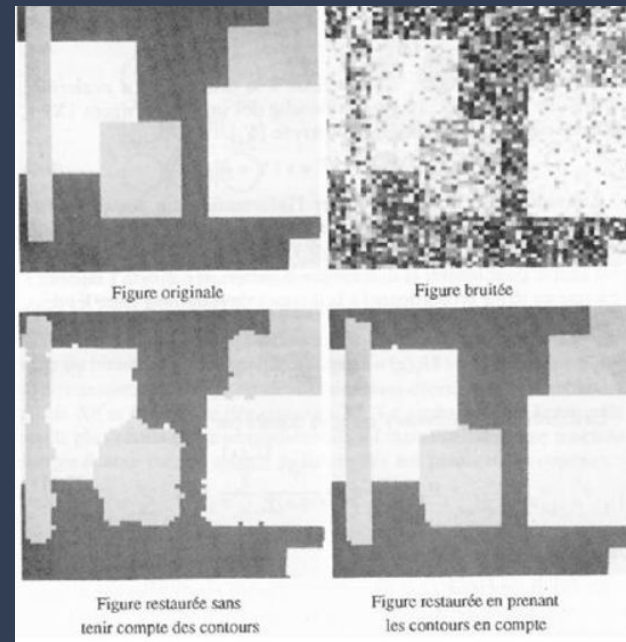
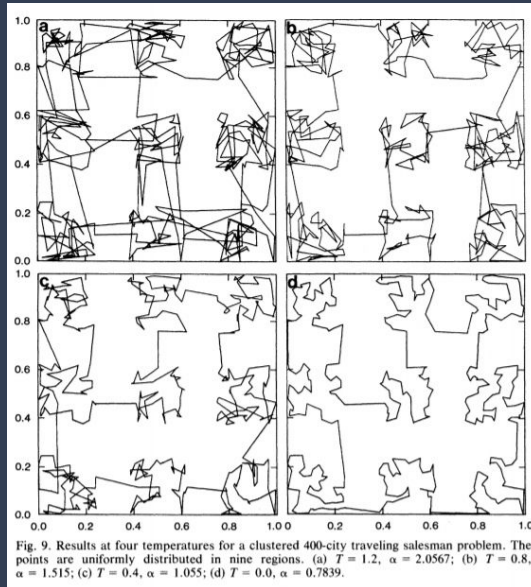
- Permet d'obtenir un minimum global
- Méthode polyvalente (ne s'appuie pas sur les paramètres intrinsèques du modèle) valable en optimisation continue ou combinatoire
- Simplicité d'implémentation
- Forte probabilité de convergence

Inconvénients:

- Compromis entre qualité et temps de calcul (par le paramètre T_0)
- Paramètres délicats à régler → influent sur le taux d'acceptation → nécessite de l'expérience
- Si l'algorithme est piégé dans minimum global à basse température: impossible d'en sortir
- dégradation des performances pour les problèmes avec peu de minimas locaux par rapport aux autres algorithmes

VI. Applications

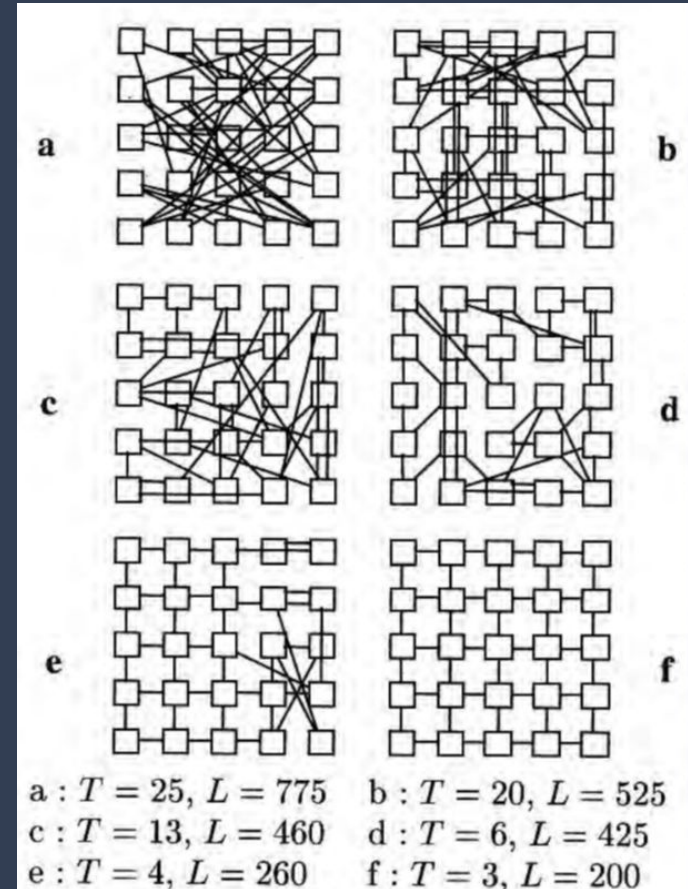
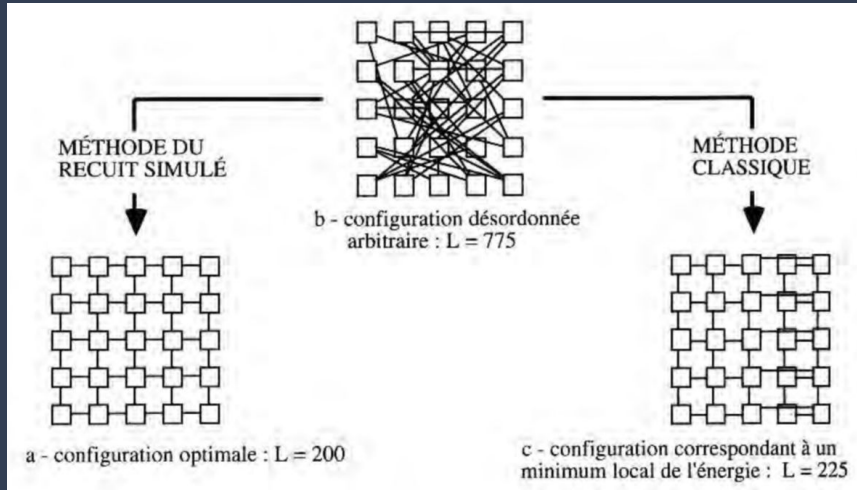
- Traitement d'image : Reconstituer une image nette à partir d'une image bruitée (*maximum de vraisemblance a posteriori*) (2.)



- Problème du voyageur de commerce : Chemin plus court à moindre coût (2.)

VI. Applications

- Placement de composants électroniques : Minimiser la longueur totale des connexions (2.)





VI. Applications

Autres applications (1.) :

- Organisation du réseau informatique du Loto (relier des machines de jeu à des ordinateurs)
- Optimisation de collecte d'ordure à Grenoble
- Problèmes d'emploi du temps : Implantation optimale des jours de repos dans un planning hospitalier)
- Architecture : répartir des activités d'une compagnie d'assurance dans des pièces d'un immeuble afin d'optimiser le rendement



Bibliographie

1. Méthode du recuit simulé - Jean Philippe Préaux
https://old.i2m.univ-amu.fr/~preaux/PDF/pdf_proteges/OptimisationCombinatoire/Metaheuristiques2.pdf
2. Le recuit simulé
<http://www.lps.ens.fr/~weisbuch/livre/b9.html>
3. Recuit simulé
https://perso.liris.cnrs.fr/pierre-edouard.portier/teaching_2015_2016/ia/sima/sima.html
4. L'algorithme du recuit simulé
<http://www.tangentex.com/RecuitSimule.htm#Par2>
5. Le recuit simulé
<https://www.slideshare.net/achrafmanaa3/chapitre-2-le-recuit-simul>
6. Méthode d'optimisation : le recuit simulé
http://hebergement.u-psud.fr/hamadache/simulated_annealing.pdf
7. Méthodes PASA et MOSA
<https://books.google.fr/books?id=haV2w-vahGYC&pg=PA108&lpg=PA108&dq=recuit+simul%C3%A9+par+eto+PASA&source=bl&ots=-hShUJnfSq&sig=ACfU3U2R-ECFlwvFfpHDJ5CQRTHrQFVjbq&hl=fr&sa=X&ved=2ahUKEwiTqYeQ9u3hAhVDxoUKHZs8DD4Q6AEwAXoECAkQAQ#v=onepage&q&f=false>