



# **Le RECUIT SIMULÉ**

**A. BENGHENI(2019/2020)**

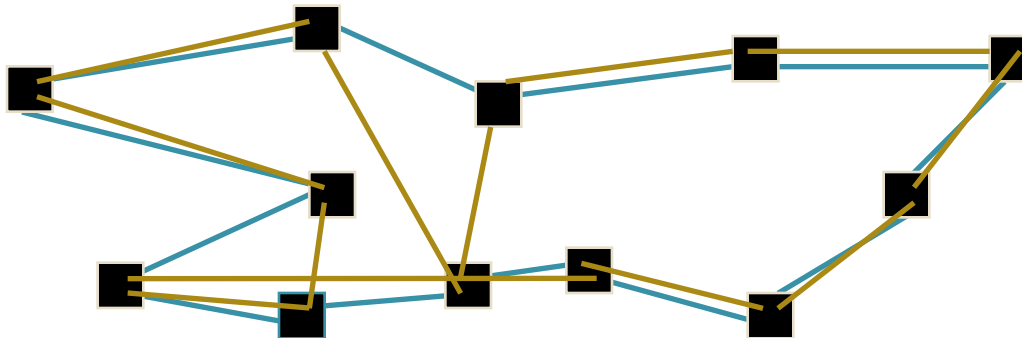
*Université IBN Khaldoun - TIARET – Faculté MI –  
Département informatique*

# Plan

- Introduction
- Origine du recuit simulé
- Principe de l'algorithme du recuit simulé
- Réglage des paramètres
- Avantages et inconvénients
- Applications
- Conclusion

# Introduction

⇒ Des problèmes informatiques complexes  
ex: Voyageur de commerce



- trouver une solution optimum à un problème.
- basé sur le hasard

# Origine du recuit simulé

- **Le recuit simulé** (Kirkpatrick 83)  
tire ses origines de la thermodynamique et de la métallurgie
- **Le recuit** : Processus physique qui consiste à élever au maximum la température d'un métal jusqu'à fusion. Les particules se déplacent d'une manière aléatoire dans ce liquide. Cette phase est immédiatement suivie d'un refroidissement effectué en baissant la température graduellement jusqu'à atteindre une énergie minimale. On obtient une structure cristallisée qui est un état stable.
- **La trempe** : refroidissement rapide. Le métal ne dispose pas de temps nécessaire pour atteindre son équilibre thermique pour chaque valeur de température. Le métal résultant contiendra beaucoup de défauts.

# L'analogie

- Refroidissez lentement un solide, de sorte que toutes les particules s'arrangent dans l'état d'énergie solide
- À chaque température attendre jusqu' à ce que le solide atteint son équilibre thermique.
- Probabilité d'être dans un état avec une énergie  $E$ :

$$P_r\{E = E\} = 1/Z(T) * \exp(-E/K_B T)$$

$E$  : énergie

$T$  : Température

$k_B$  : constante de Boltzmann

$Z(T)$  : facteur de normalisation

# Algorithme de Métropolis-Hasting: simulation du recuit

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

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

$k_B$  : constante de Boltzmann

$T$  : Température

# Application du recuit simulé aux POC (problèmes d'optimisation combinatoire)

- La fonction objectif (coût) du problème, étant analogue à l'énergie du métal. Elle est minimisée par l'introduction d'une température ( paramètre de contrôle de l'algorithme).
- La température doit conduire vers :
  - l'état optimal si elle est baissée de façon lente et bien contrôlée (technique du recuit)
  - un minimum local si elle est baissée brutalement (trempe)

# Analogie en optimisation

- $f(x) = E$ : niveau d'énergie
- $x$  = état du système
- Paramètre de contrôle = température

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

avec :

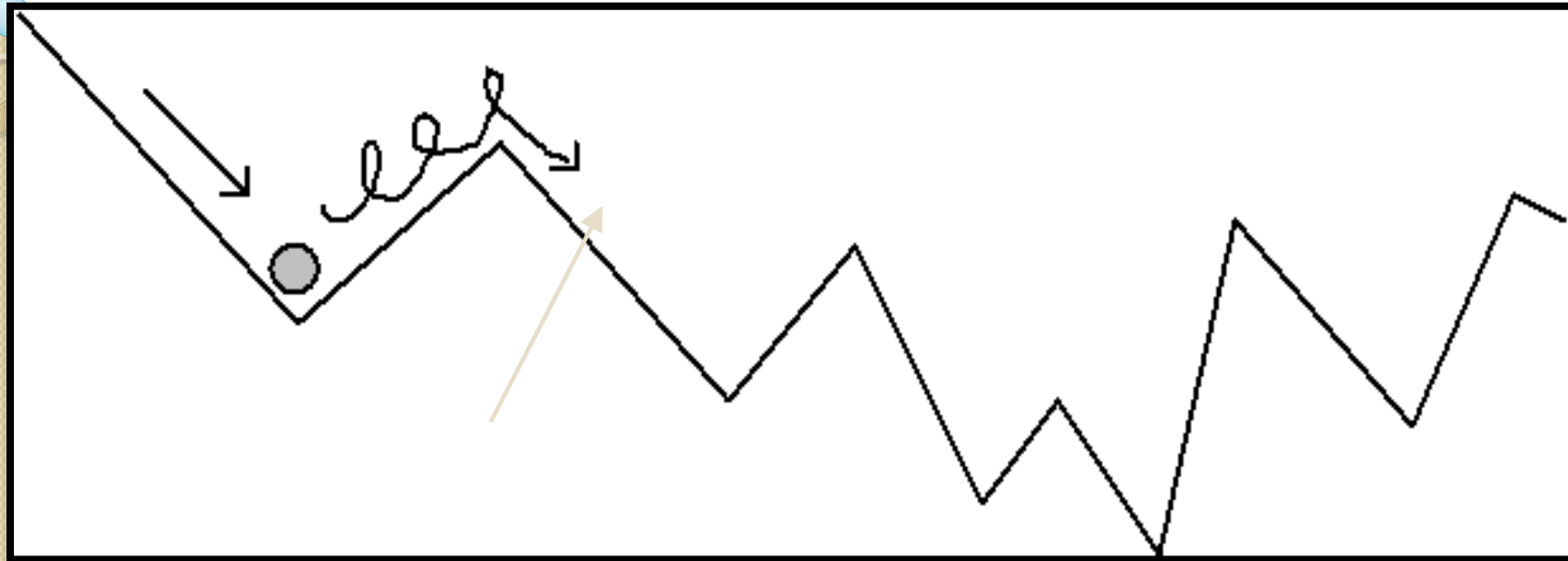
$C_k$  : paramètre de contrôle



# D'autres noms du recuit simulé (simulated annealing)

- Monte Carlo Annealing
- Statistical Cooling
- Probabilistic Hill Climbing
- Stochastic Relaxation
- Probabilistic Exchange Algorithm

# Principe du recuit



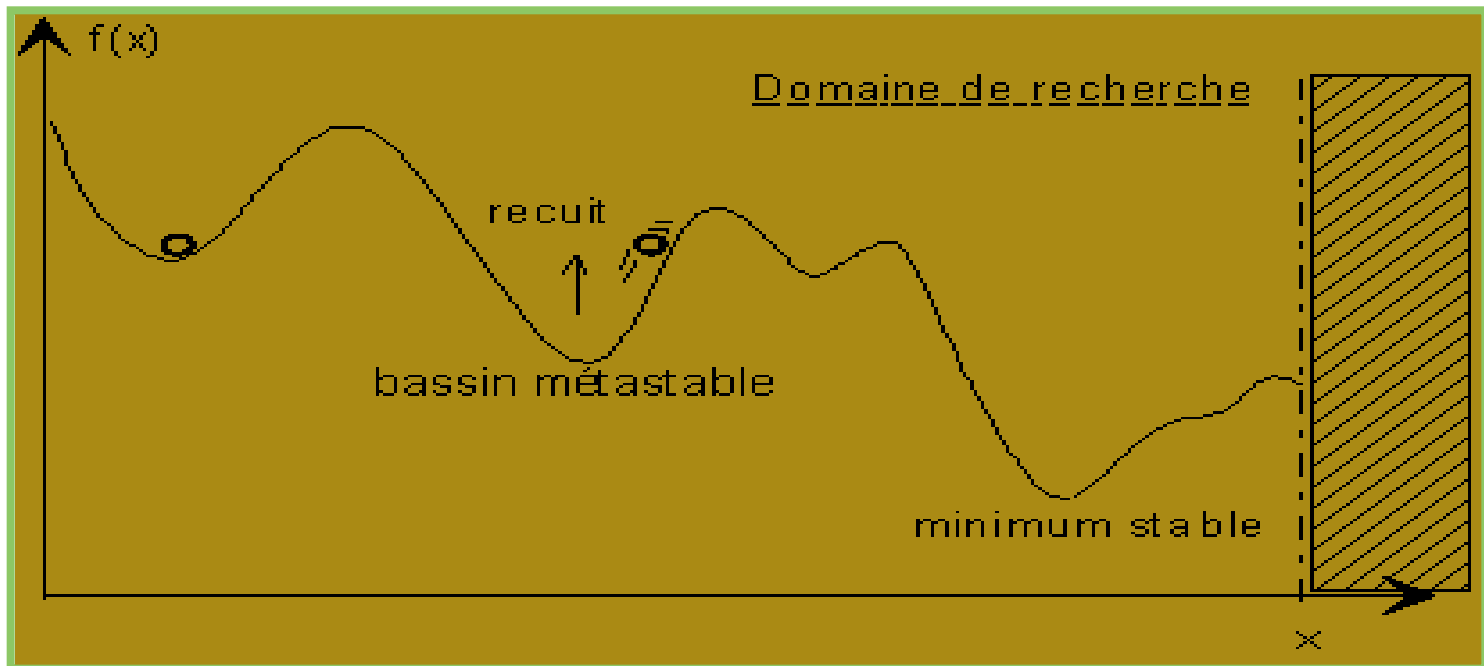
~~Minimum local~~  $\Rightarrow$  Recuit simulé

**méthode Générale pour résoudre les problèmes  
d'optimisation combinatoire.**

# Principe du recuit

$$P = \exp(-\Delta E / T), \quad r < P, \quad r \in [0,1]$$

T: Température



1. Recherche d'un minimum à une température fixé
2. Diminution par étape de la température

# Principe du recuit

- Commencer avec une configuration initiale
- Rechercher à plusieurs reprises la solution et sélectionner un point voisin comme candidat.
- Évaluer une certaine fonction de coût (objectif) et accepter le candidat si il présente la meilleur solution sinon sélectionner un autre candidat
- Arrêter si la qualité est suffisamment haute, si aucune amélioration ne peut être trouvée, ou après un certain temps d'arrêt.

# La méthode recuit simulé

## Le recuit simulé :

- C'est un processus qui baisse lentement sa température jusqu'à ce que le système se fige et que plus aucun changement ne soit observé,
- Le recuit simulé s'appuie sur l'algorithme de Métropolis-hasting

# La méthode recuit simulé

- Le principe :

1. Choisir une solution  $s \in S$  ainsi qu'une température initiale  $T$ ;
2. Tantqu'aucun critère d'arrêt n'est satisfait faire
3. Choisir aléatoirement  $s' \in N(s)$  ( $s'$  solution voisine de  $s$ );
4. Générer un nombre réel aléatoire  $r$  dans  $[0,1]$ ;
5. Si  $r < P(T, s, s')$  alors poser  $s := s'$ ;
6. Mettre à jour  $T$ ;
7. Fin tantque

Avec  $T$ : Température,  $s$ : solution initiale,  $s'$ : solution voisine de  $s$

- La fonction  $P(T, s, s') = \frac{e^{\frac{f(s)-f(s')}{T}}}{\sum e^{\frac{f(s)-f(s')}{T}}}$  à la distribution de Boltzman avec:
- si  $f(s') < f(s)$  alors  $e^{\frac{f(s)-f(s')}{T}} > 1 \Rightarrow :r < P(T, s, s')$  et on accepte  $s'$
- si  $T$  a une très grande valeur et  $f(s') > f(s)$  alors  $e^{\frac{f(s)-f(s')}{T}} \cong 1 \Rightarrow$  :on peut accepte  $s'$  (malgré que  $s'$  est une solution mauvaise que  $s$ )
- si  $T$  a une très petite valeur et  $f(s') > f(s)$  alors  $e^{\frac{f(s)-f(s')}{T}} \cong 0 \Rightarrow$  :on va donc probablement refuser le  $s'$

Avec :  $f$  : c'est la fonction objectif (l'énergie  $E$  du métal)

# Réglage des paramètres

Pour appliquer le recuit simulé, on devrait déterminer quelques **paramètres choisis expérimentalement**:

- La **température** initiale
- La **configuration** (solution) initiale
- La **règle de transformation** permettant de passer d'une configuration à une autre.
- La **durée de relaxation** (ie : le nombre d'itérations de la transformation par palier de température ).
- La **loi de décroissance** de la température.
- Le(s) **critère(s) d'arrêt**.



# Réglage des paramètres

Il n'y a pas **de règles générales** pour choisir les valeurs des Paramètres.

- Par **expérience**:

- T0 assez grand pour que la probabilité d'accepter de mauvaises solutions soit importante.
- Facteur de décroissance de la température: 0.95 (**effet sur le temps d'exécution**)
- Nombre d'itérations dans **un palier**
- **Température d'arrêt**: telle que plus aucune détérioration significative du résultat ne puisse plus être acceptée.

# Réglage des paramètres

- Diminution de la température

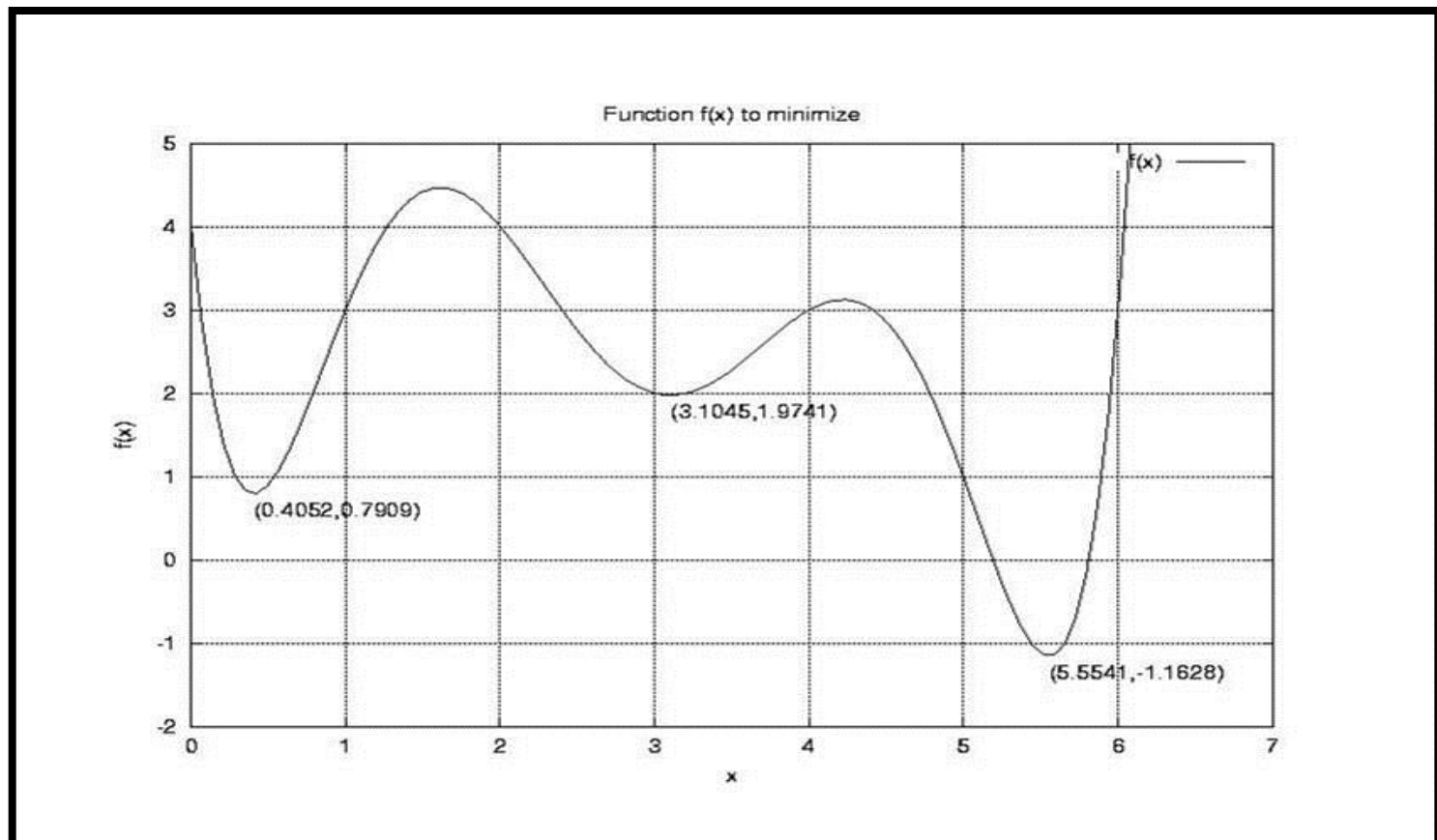
Type	Fonctions	Paramètres
Linéaire	$T_{K+1} = \alpha \times T_K$	$\alpha$
Discrète	$T_{K+1} = T_K - \Delta T$	$\Delta T$
Exponentielle	$T_{K+1} = T_K \times \exp\left(\frac{-\lambda.T_K}{\sigma_K}\right)$	$\sigma_k$ : l'écart-type des coûts ( sous Tk)

- Température départ / arrêt : nb de paliers

# Quelques Applications

## 1. Recherche de minimum

$$f(x) = 4 - 19.167x + 36.39167x^2 + 25.2917x^3 + 8.041667x^4 - 1.19167x^5 + 0.06667x^6$$



# 1. Recherche de minimum

- Conditions initiales:  $T_0$ ,  $T_f$  et  $X_c$
- Fonction de réduction de  $T^\circ$  :  $\alpha(T)$
- Répéter tant que  $T > T_f$

- Répéter  $k$  fois

$$k = 100$$

- Choisir au hasard un voisin  $X$  de  $X_c$

$$\{X - 0.25, X + 0.25\}$$

- $\Delta = f(X) - f(X_c)$

- Si  $\Delta < 0$  alors  $X_c = X$

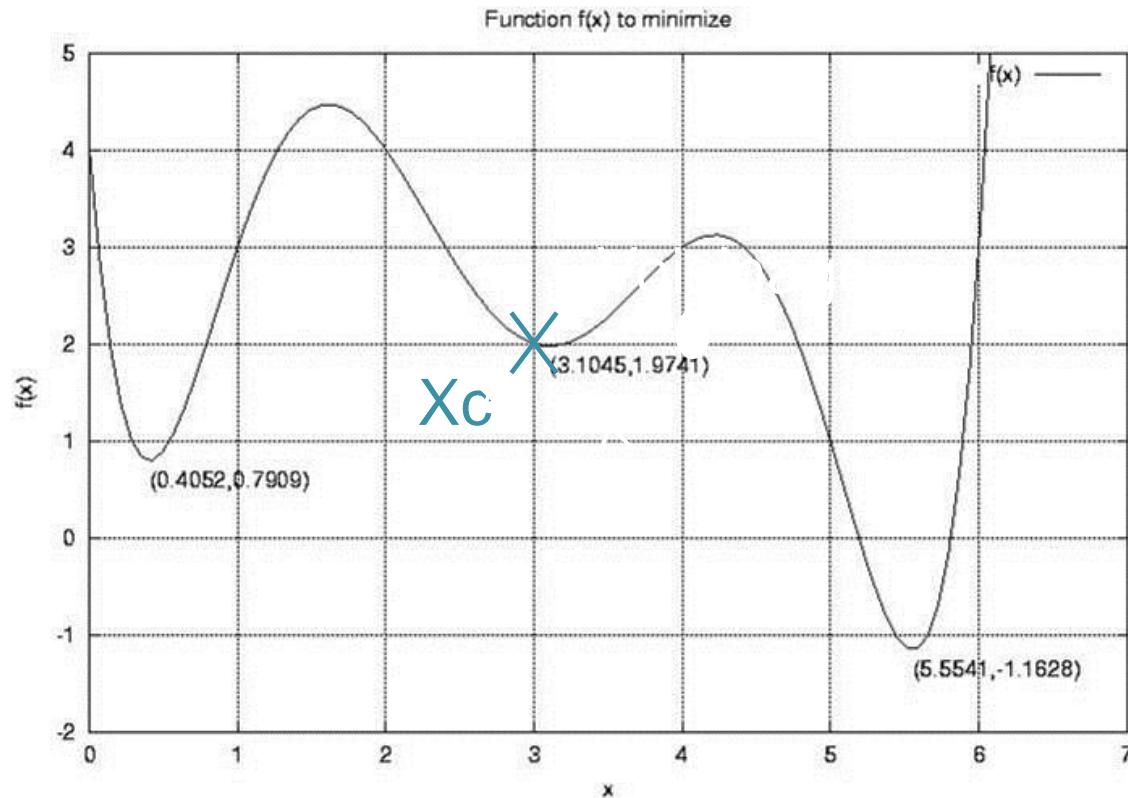
- Sinon

- Choisir  $r$  aléatoirement entre 0 et 1

- Si  $r < \exp(-\Delta/T)$  alors  $X_c = X$

- $T = \alpha(T)$

# 1. Recherche de minimum



# Avantages et inconvénients de la méthode recuit simulé

## Avantages:

- Le recuit simulé présente l'avantage d'offrir des solutions de bonne qualité
- Méthode générale et facile à programmer
- Souplesse d'emploi: De nouvelles contraintes peuvent être facilement incorporées.

## Inconvénients:

- Nombre important de paramètres
- Temps de calcul: excessif dans certaines applications
- **Traitement d'une solution plusieurs fois**

# Quelques applications du recuit simulé

- Traitement d'images
- Problèmes d'ordonnancement
- Conception des circuits électroniques (Problème de placement et de routage)
- Organisation du réseau informatique
- Collecte des ordures ménagères
- Problème du voyageur de commerce (10000 villes).

# Conclusion

- ✓ Recuit simulé : algorithme adaptable à différents type de problème
- ✓ retenir l'idée : accepter une « détérioration » pour accéder à la meilleure solution
  - l'un des principaux de la méthode recuit simulé est l'étape 3 où s'est choisi aléatoirement  $s'$  dans  $N[s]$
  - La méthode qui n'a pas ce défaut c'est la recherche tabou (l'objectif du prochain cours)





Exemple d'application

Voir le fichier d'annexe