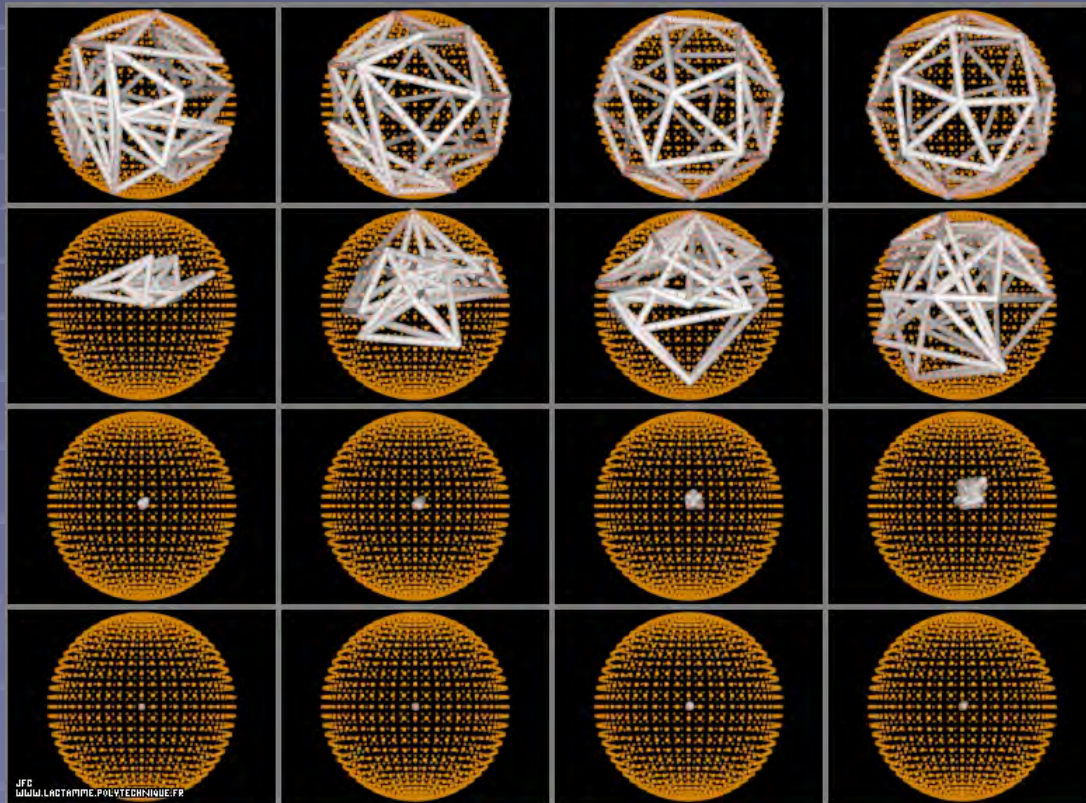


Méthode du recuit simulé

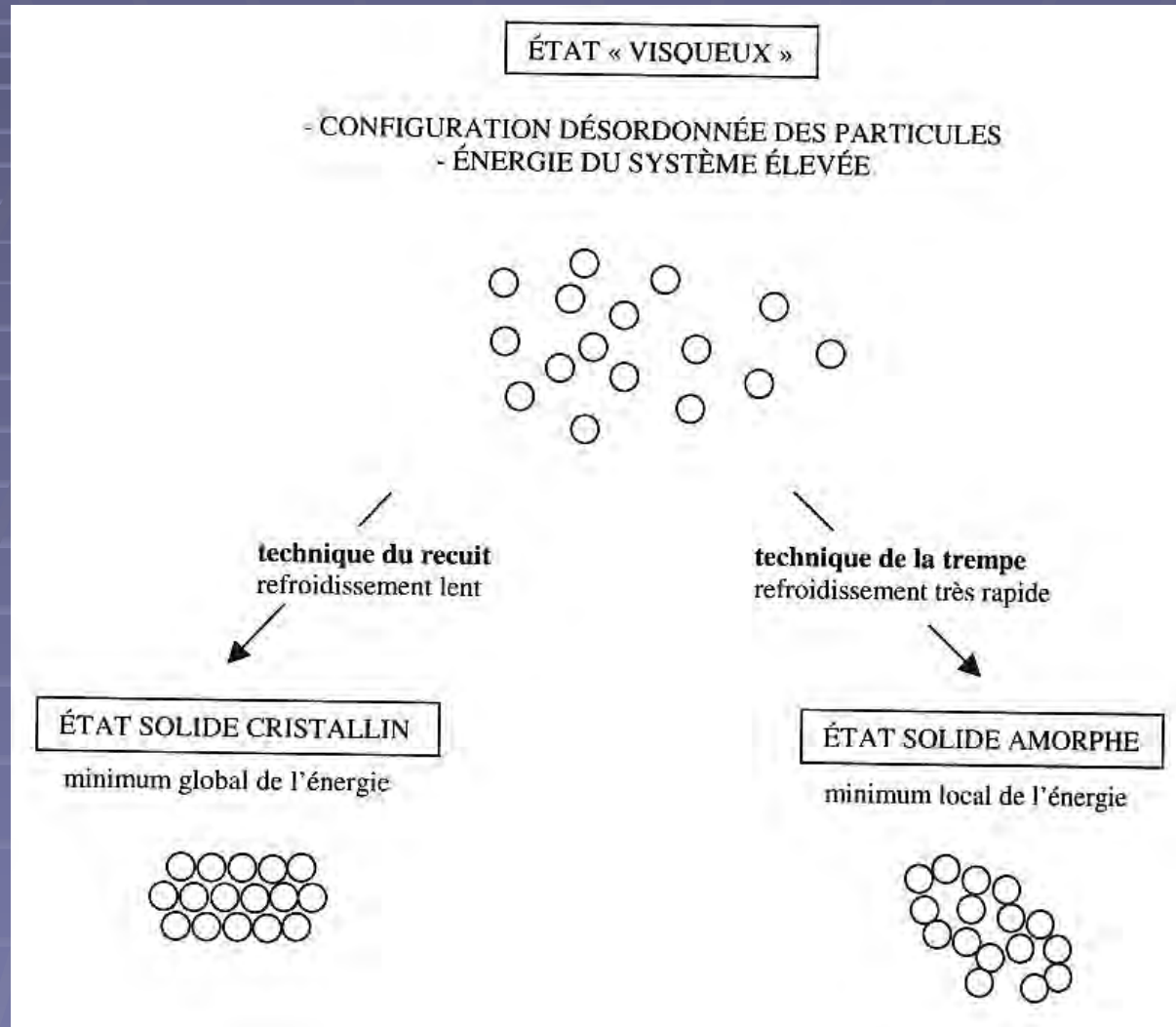


Méthode du recuit simulé

- Inventée par les physiciens Kirkpatrick, Gellat, Vecchi en 1983 (IBM).
- A permis de résoudre de façon quasi-optimale des PVC de 5000 sommets !
- Basé sur les méthodes de simulation de Metropolis (processus stochastique) en mécanique statistique.
- S'inspire du recuit des métaux en métallurgie. (En anglais 'annealing'.)

- La méthode du recuit simulé a prouvé son efficacité dans des domaines aussi divers que :
 - conception des circuits électroniques,
 - traitement des images,
 - collecte des ordures ménagères,
 - organisation du réseau informatique du loto.
 - etc...
- Trop gourmande ou inadaptée pour résoudre certains problèmes d'optimisation, mieux traités par des heuristiques spécifiques.

Analogie avec la technique du recuit en metallurgie

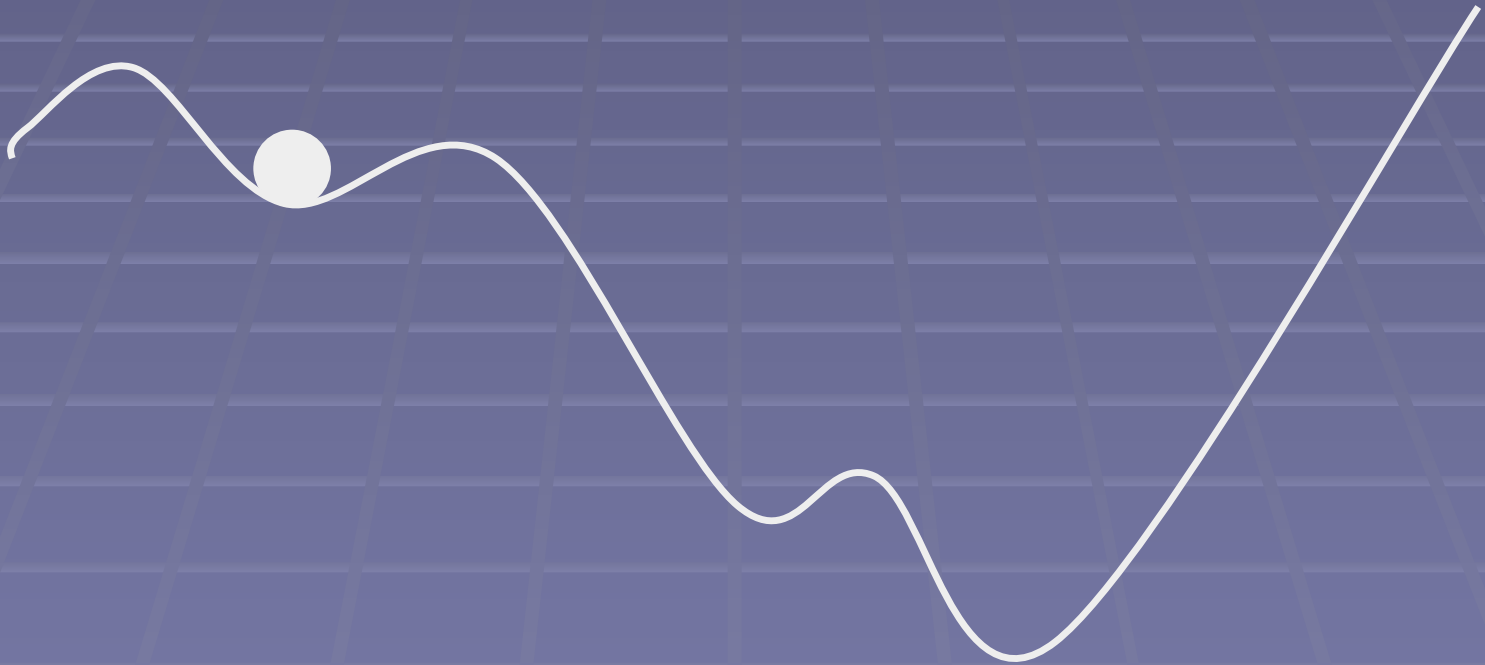


Analogie avec le lâcher d'une balle élastique



Analogie avec le lâcher d'une balle élastique

Lâcher d'une pierre



min local de l'énergie potentielle

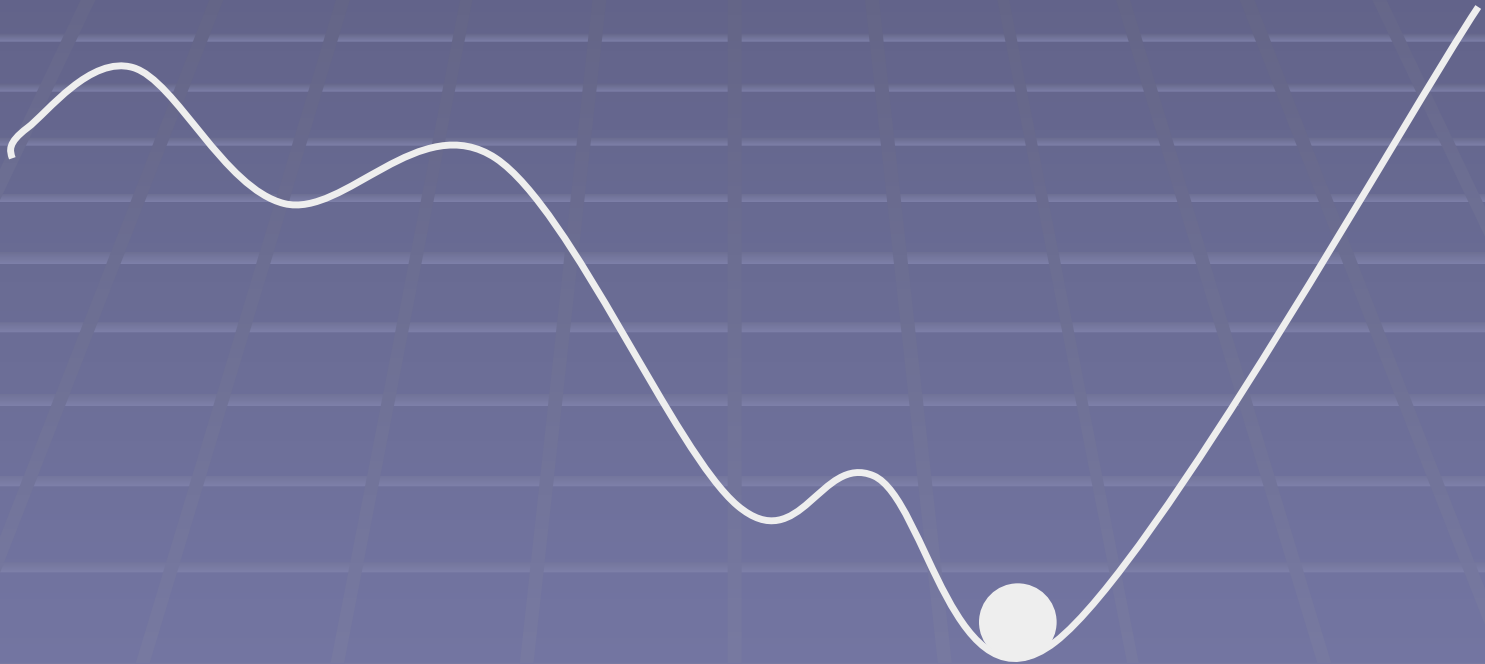
Analogie avec le lâcher d'une balle élastique

Lâcher d'une balle élastique



Analogie avec le lâcher d'une balle élastique

Lâcher d'une balle élastique



min global de l'énergie potentielle

Analogie entre un problème physique et un problème d'optimisation

Problème d'optimisation	Problème physique
Fonction objectif	Énergie libre
Paramètres du problème	Coordonnées des particules
Trouver une configuration optimale	Trouver les états de basse énergie

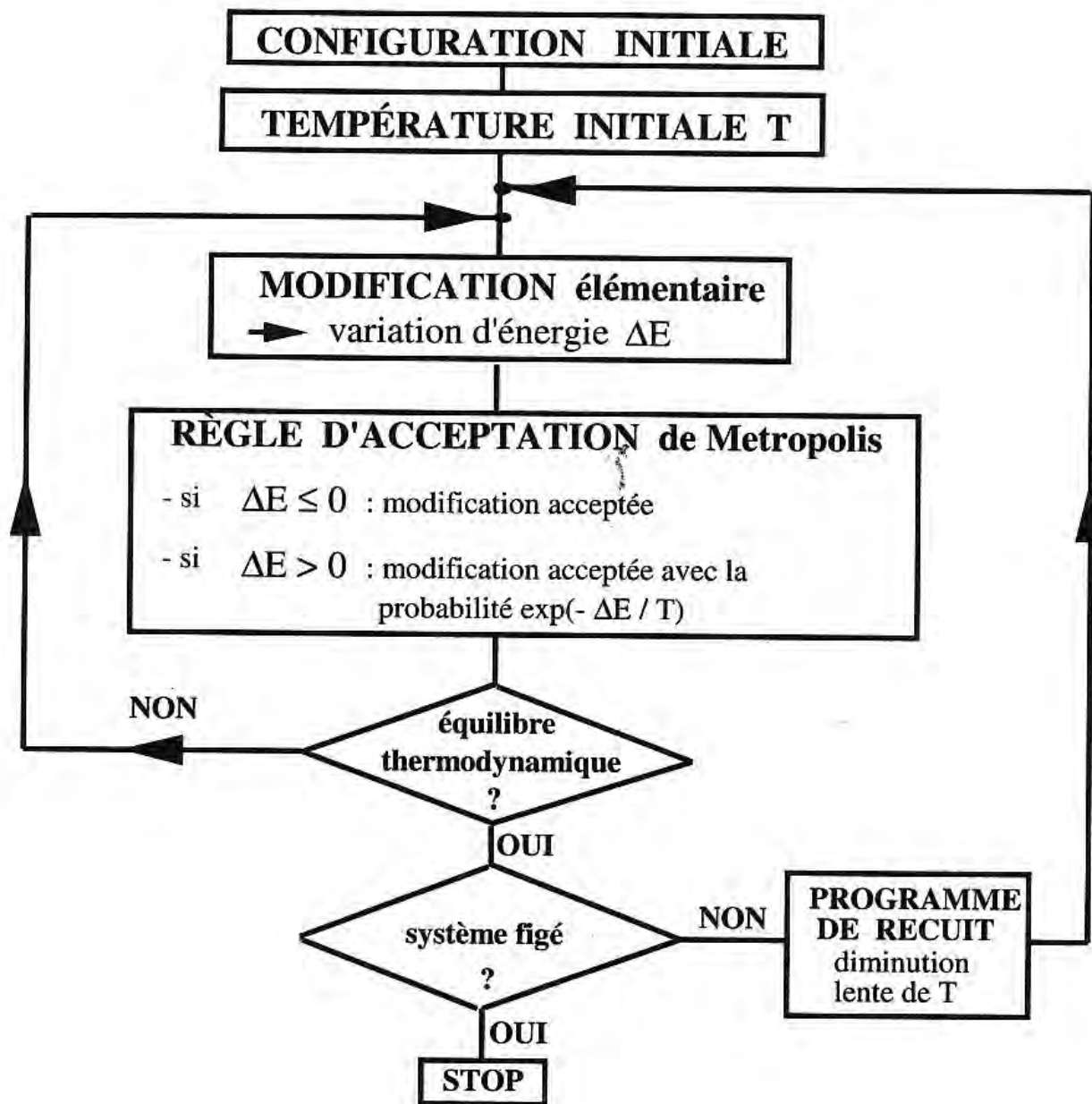
Principes du recuit simulé

L'algorithme s'appuie sur deux résultats de physique statistique :

- Lorsque l'équilibre thermodynamique est atteint à une température T , la probabilité pour un système physique de posséder une énergie E est proportionnelle au facteur de Boltzmann : $\exp(-E/KT)$ (K =constante de Boltzmann).
- Pour simuler l'évolution d'un système physique vers son équilibre thermodynamique à une température T , on utilise l'algorithme de Metropolis :
- Partant d'une configuration initiale du système on lui fait subir une modification élémentaire. Si celle-ci diminue l'énergie E elle est acceptée. Sinon, si $\Delta E > 0$ elle est tout de même acceptée avec une probabilité $\exp(-\Delta E/T)$. (à basse température seuls de très faibles augmentation d'énergie sont acceptées).
- La suite des configurations converge vers un état d'équilibre thermodynamique à la température T .
- On diminue « légèrement » la température et on re-converge vers un équilibre thermodynamique. Si la température est abaissée trop vite la convergence est ralentie.
- Lorsque la température est proche d'un zéro, on est proche d'un minimum de l'énergie.

Algorithme du recuit simulé

- On se donne un réel arbitraire $T > 0$: la température.
- Des transformations définissant le voisinage $V(s)$ de s .
- On tire au sort une transformation $s \rightarrow s'$ dans $V(s)$.
- Soit $\Delta f = f(s') - f(s)$ la variation de coût.
- Si $\Delta f \leq 0$ on effectue la transformation : s devient s' .
- Si $\Delta f > 0$ on accepte la transformation s devient s' avec la probabilité $\exp(-\Delta f/T)$. (i.e. on tire au sort un nombre x dans $[0,1]$ avec une loi uniforme et on accepte si $x \leq \exp(-\Delta f/T)$).
- On diminue légèrement la température. Par exemple $T = 0.999 T$. (en général on la diminue par palier).
- On recommence tant que T n'a pas atteint un seuil fixé, proche de 0.



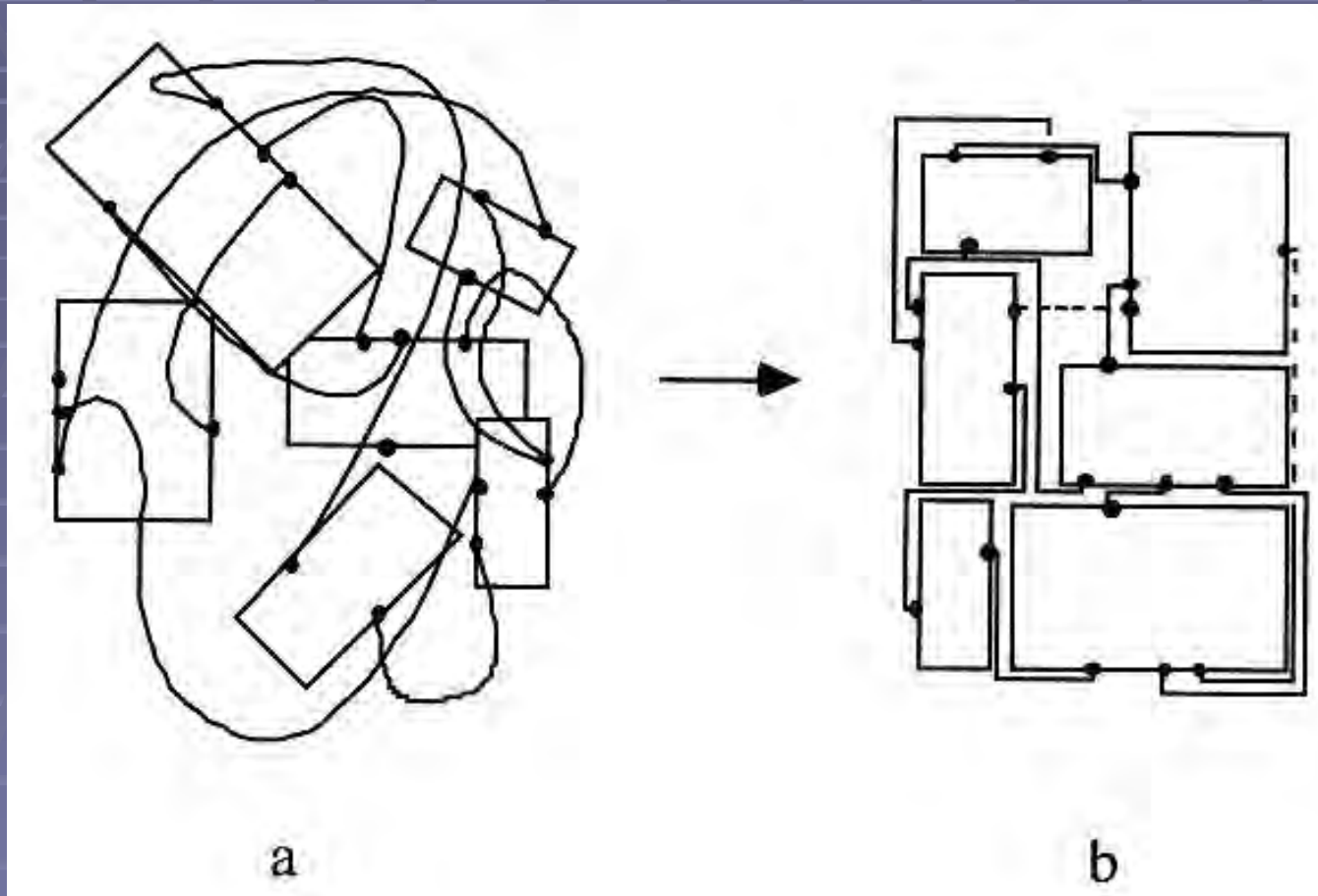
Avantages du recuit simulé

- Facile à implémenter.
- D'une grande souplesse. Se prête aussi bien à l'optimisation combinatoire que continue.
- A donné d'excellents résultats pour des problèmes de grande taille.
- 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 « puis d'énergie ».
- C'est un très bon algorithme itératif pour l'optimisation de fonctions continues non convexes (sinon il y a mieux, c.f. cours d'optimisation continue).

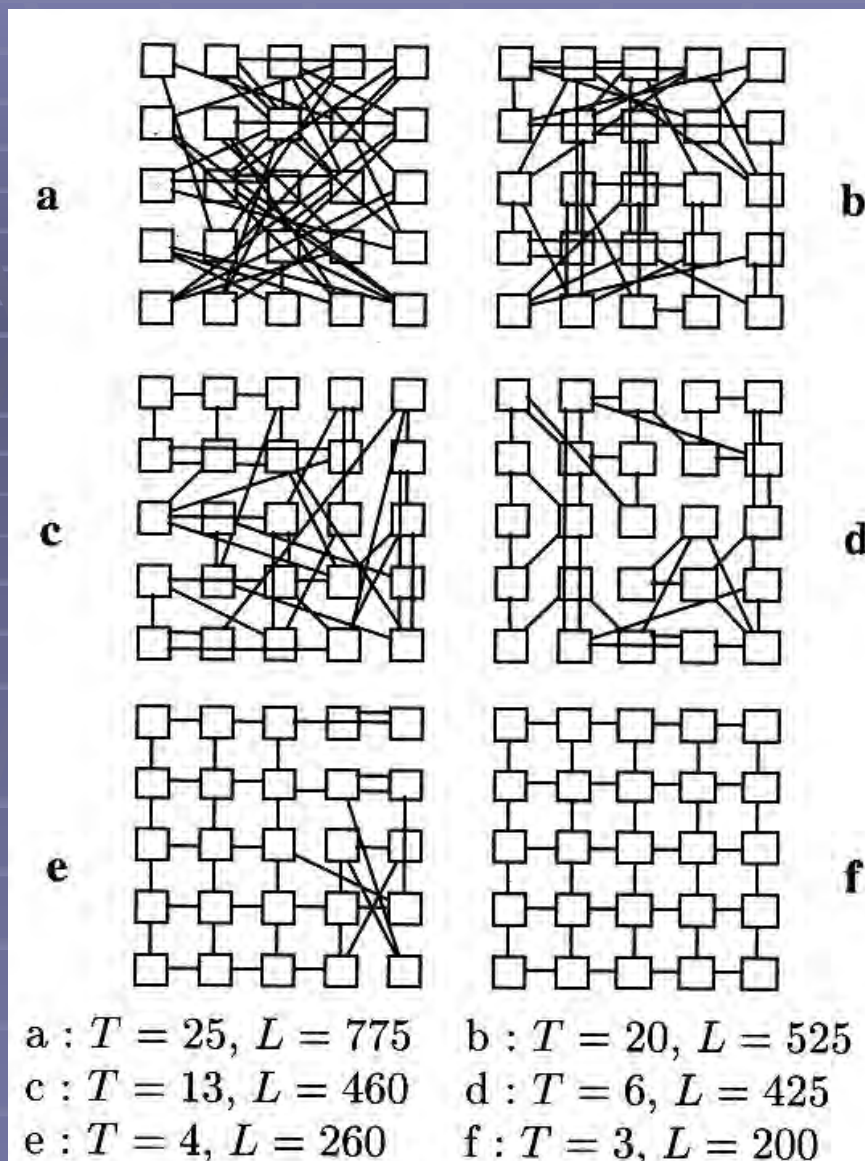
Inconvénients de la méthode du recuit simulé

- Des paramètres délicats à régler (loi de décroissance de température, température initiale, paliers), qui nécessitent un certain savoir-faire et de nombreux essais.
- Les temps de calcul peuvent devenir importants ; ce qui peut nécessiter une parallélisation des calculs.
- Un choix simple et souvent efficace pour le paramètre le plus sensible : la loi de décroissance de température, est $T = kT$ pour k très proche de 1, $T = 0.999$ par exemple.

Exemple : application au placement de composants électronique

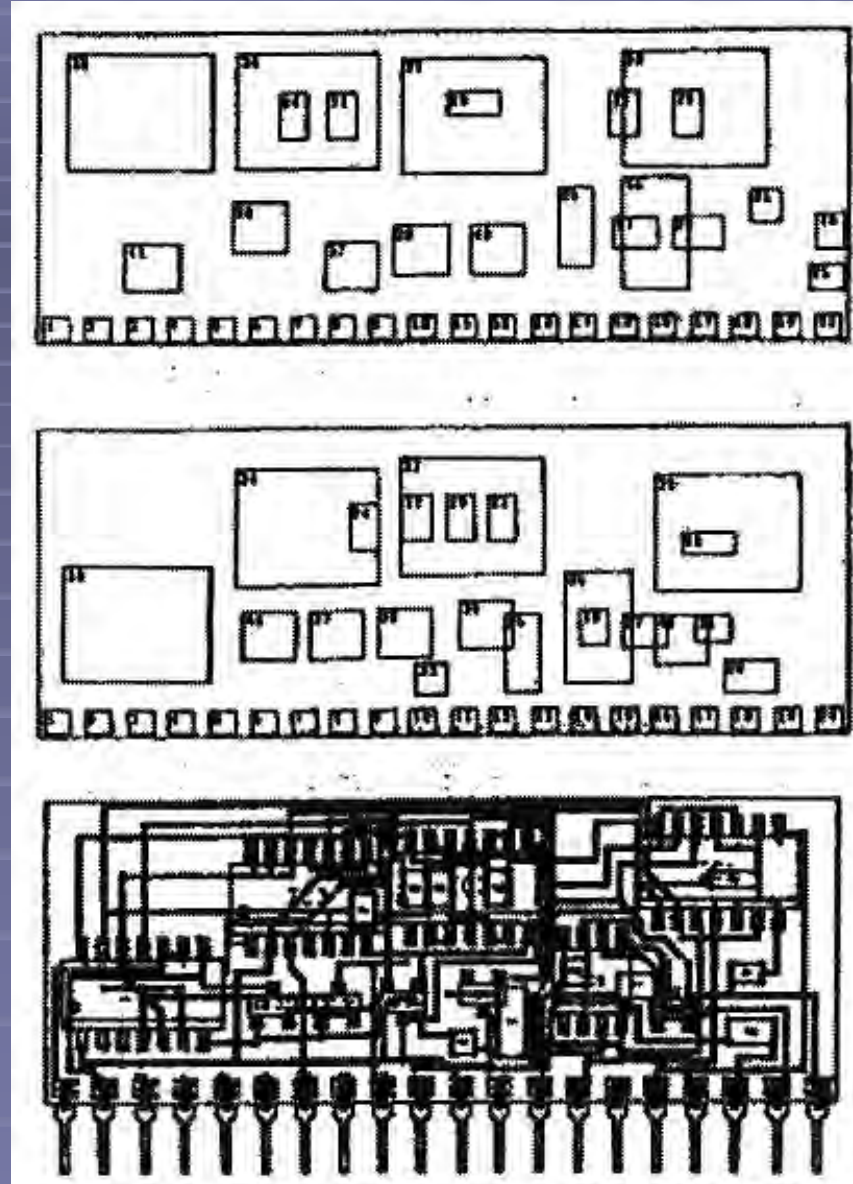


Placement de composants électroniques de façon à minimiser la longueur totale des connexions.

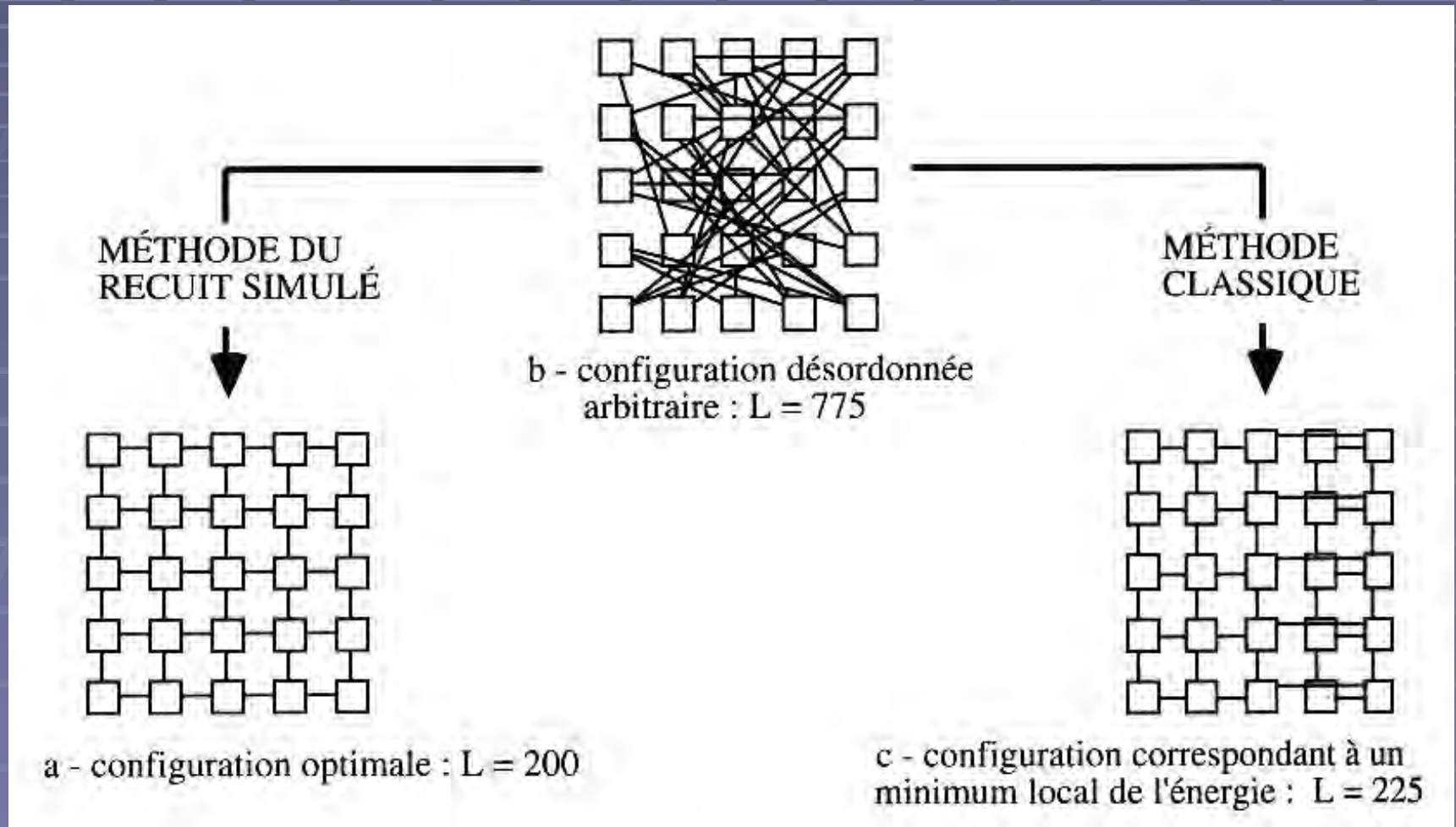


Evolution du système par recuit simulé à diverses températures.

- En haut : placement manuel de 41 composants :
 $L=9532$.
- Au milieu : placement optimisé par recuit :
 $L=7861$.
- En bas : routage utilisant le placement optimisé par recuit.



Comparaison du recuit simulé et d'une heuristique de recherche locale :



Exemple pour le PVC

- Le voisinage d'une solution est construit par 2-opt.
- on part d'un mauvais cycle $(1, 2, \dots, N, 1)$ car un bon cycle serait détruit par la haute température du début.
- On s'arrange pour que la pire augmentation $D_{\max}$ soit acceptée avec probabilité $2/3$. cela donne pour température initiale $T_0 = D_{\max} / \log(2/3)$.
- On prend pour décroissance de la température $T = 0.999T$.
- Pas de palier : décroissance de la température à chaque itération.
- On peut utiliser une technique de « cycle de réchauffement » qui accélère la convergence : On effectue N itérations en tout, mais plusieurs implémentations à partir de températures $T_0, T_0/2, T_0/4$, etc...

- Sur les graphes de 200 et 800 noeuds déjà vus on trouve d'assez mauvais résultats (200 sommets) :

Heuristique	Non-euclidiens		euclidiens	
	Coût	Durée	Coût	Durée
2-opt	3144	17	7935	37
Tabou	3025	337	7893	335
Recuit	5823	31	8638	34

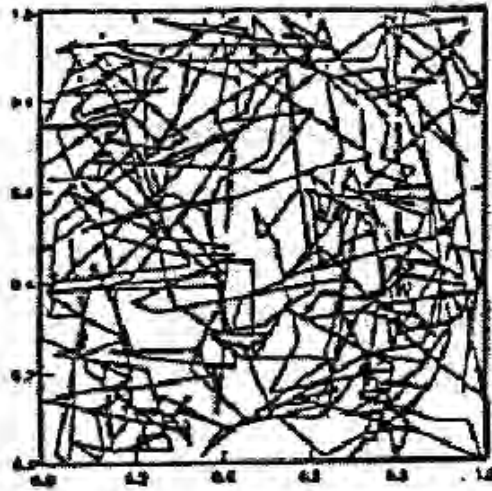
- Sur les graphes de 200 et 800 noeuds déjà vus on trouve d'assez mauvais résultats (200 sommets) :

Heuristique	Non-euclidiens		euclidiens	
	Coût	Durée	Coût	Durée
2-opt	3144	17	7935	37
Tabou	3025	337	7893	335
Recuit	5823	31	8638	34

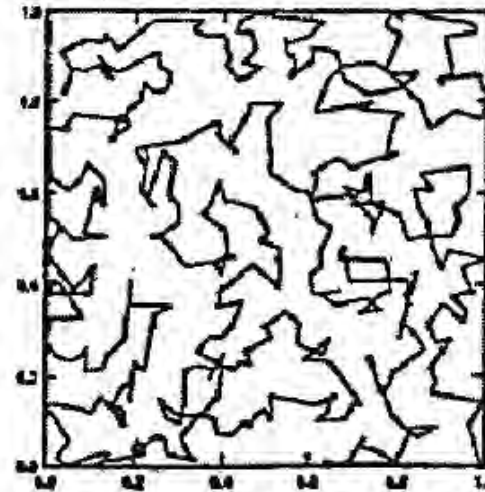
En prenant $T=0.999T$, on surpasse 2-opt.

La meilleure reste la méthode Tabou.

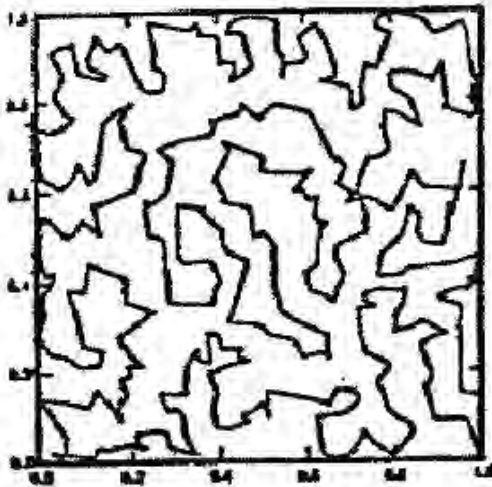
$T=2.0$
 $L=3.6447$



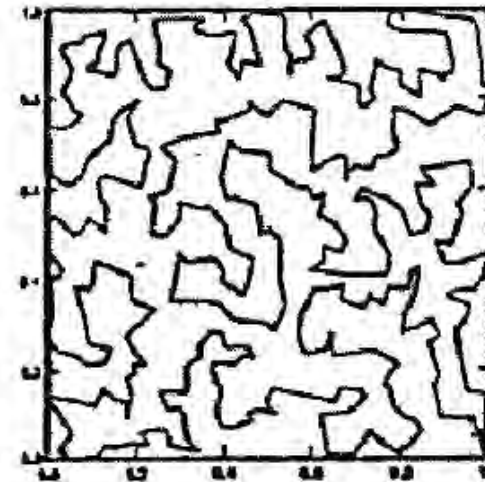
$T=0.4$
 $L=1.1605$



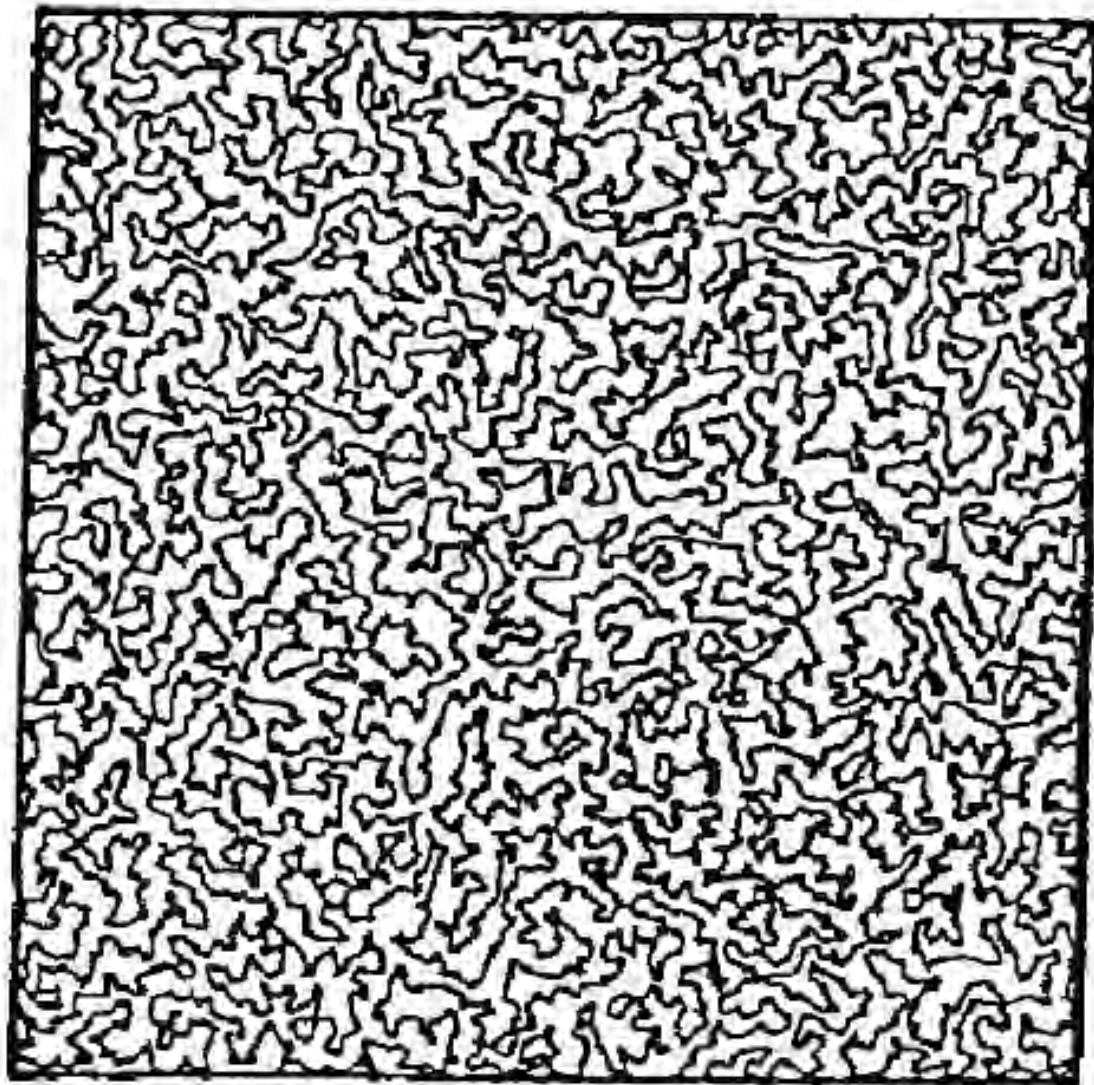
$T=0.2$
 $L=0.9845$



$T=0.01$
 $L=0.9559$



Solution au PVC euclidien pour 400 villes, par un recuit simulé, à 4 paliers de température.



Solution obtenu par recuit à un PVC euclidien de

100 villes.
Jean-Philippe
Préaux

[http://www.i2m.univ-
amu.fr/~preaux](http://www.i2m.univ-amu.fr/~preaux)

Implémentation pour la coloration

- On prend les mêmes paramètres que dans l'implémentation du PVC.
- Les colorations sont codées par des ordres sur les sommets via l'heuristique séquentielle.
- Le voisinage est obtenu en appliquant une transposition sur la solution donnée.
- Comme précédemment on part d'une solution mauvaise (obtenue par FFS).
- Ici le comportement du recuit est très bon. sensiblement meilleur que la recherche tabou.

- Sur des graphes aléatoires de 30 sommets :

Méthode	#moyen couleurs	#optima	Durée moyenne
Rech. locale	7.64	28	3
Tabou (#T=20, N=2000)	7.24	48	84
Recuit ($T=0.99T$)	7.22	49	101

Quelques réussites notables du recuit

- **Traitement d'image** : Reconstituer une image nette à partir d'une image bruitée.
- Une approche théorique 'bayésienne', à l'aide des « champs de markov » détermine l'image ayant le maximum de vraisemblance à posteriori.
- Se ramène à la minimisation d'une fonction objectif ayant pour paramètres les intensités des pixels.
- Le recuit fait évoluer l'image en modifiant l'intensité des pixels à tour de rôle dans un ordre préétabli.
- temps de calcul important, mais l'image peut se traiter localement : d'où une parallélisation du calcul.
- A obtenu de très bons résultats dans les tâches essentielles de ce domaine : restauration d'images brouillées, segmentation d'image, identification de scènes,...
- d'autres applications pour déterminer la structure géologique des sols à partir de résultats sismiques.

Autres succès

- Organisation du réseau informatique du loto (relier une dizaine de milliers de machines de jeu à des ordinateurs centraux).
- optimisation de la collecte des ordures à grenoble.
- problèmes d'emploi du temps (implantation optimale des jours de repos dans un planning hospitalier).
- optimisation en architecture (projet de construction d'un immeuble de 17 étages destiné à une compagnie d'assurance : répartir les activités dans chaque pièce de façon à optimiser le rendement de 2000 employés).