

Méthode *simulated annealing* (recuit simulé)

- C'est une amélioration de l'algorithme *hill-climbing* pour **minimiser le risque d'être piégé dans des maxima/minima locaux**
 - ◆ au lieu de regarder le meilleur voisin immédiat du nœud courant, **avec une certaine probabilité on va regarder un moins bon voisin immédiat**
 - » on espère ainsi s'échapper des optima locaux
 - ◆ au début de la recherche, la **probabilité de prendre un moins bon voisin** est plus élevée et **diminue graduellement**
- Le nombre d'itérations et la diminution des probabilités sont définis à l'aide d'un schéma (*schedule*) de « températures », en ordre décroissant
 - ◆ ex.: schéma de 100 itérations [2^0 , 2^{-1} , 2^{-2} , ... , 2^{-99}]
 - ◆ la meilleure définition du schéma va varier d'un problème à l'autre

Algorithme *simulated annealing*

Algorithme SIMULATED-ANNEALING(*noeudInitial*, *schema*) // *cette variante maximise*

1. déclarer deux nœuds : n, n'
2. déclarer : $t, T, \Delta E$,
3. $n = \text{noeudInitial}$
4. pour $t = 1 \dots \text{taille}(\text{schema})$
 5. $T = \text{schema}[t]$
 6. $n' = \text{successeur de } n \text{ choisi au hasard}$
 7. $\Delta E = F(n') - F(n)$ // *si on minimisait, $\Delta E = F(n) - F(n')$*
 8. si $\Delta E > 0$ alors assigner $n = n'$ // *amélioration p/r à n*
 9. sinon assigner $n = n'$ seulement avec probabilité de $e^{\Delta E / T}$
6. retourner n

plus T est petit,
plus $e^{\Delta E / T}$ est petite



D'autres améliorations: *tabu search*

- L'algorithme *Simulated annealing* minimise le risque d'être piégé dans des optima locaux
 - ◆ par contre, il n'élimine pas **la possibilité d'osciller indéfiniment** en revenant à un nœud antérieurement visité
- **Idée:** on pourrait **enregistrer les nœuds visités**
 - ◆ on revient à A^* et approches similaires!
 - ◆ mais c'est impraticable si l'espace d'états est trop grand
- L'algorithme ***tabu search*** (recherche taboue) enregistre seulement les **k derniers nœuds visités**
 - ◆ l'**ensemble tabou** est l'ensemble contenant les k nœuds
 - ◆ le paramètre k est choisi empiriquement
 - ◆ cela n'élimine pas les oscillations, mais les réduit
 - ◆ il existe en fait plusieurs autres façon de construire l'ensemble taboue...

D'autres améliorations: *beam search*

- **Idée:** plutôt que maintenir un seul nœud solution n , en pourrait maintenir un ensemble de k nœuds différents
 1. on commence avec un ensemble de k nœuds choisis aléatoirement
 2. à chaque itération, tous les successeurs des k nœuds sont générés
 3. on choisit les k meilleurs parmi ces nœuds et on recommence
- Cet algorithme est appelé ***local beam search*** (exploration locale par faisceau)
 - ◆ à ne pas confondre avec *tabu search*
- Variante ***stochastic beam search***
 - ◆ plutôt que prendre les k meilleurs, on assigne une probabilité de choisir chaque nœud, même s'il n'est pas parmi les k meilleurs (comme dans *simulated annealing*)