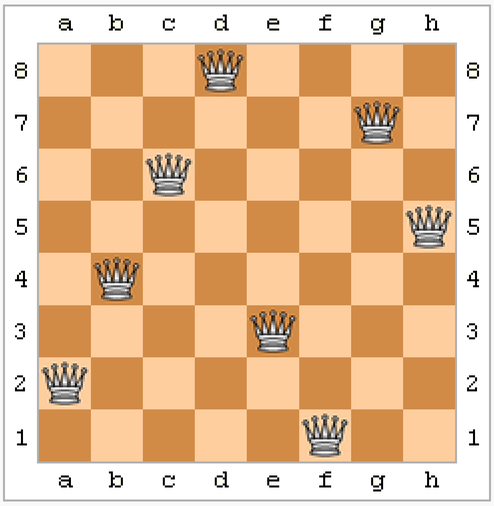
# Le problème des *n* dames pour illustrer les métaheuristiques

par Didier Müller, Lycée cantonal de Porrentruy

Le but du *problème des huit dames* est de placer huit dames d'un jeu d'échecs sur un échiquier de 8×8 cases sans que les dames se menacent mutuellement, conformément aux règles du jeu d'échecs. Par conséquent, deux dames ne devront jamais partager la même rangée, colonne, ou diagonale.

Le problème des *n* dames est une généralisation de ce problème classique : on considère un échiquier *n*x*n* au lieu d'un échiquier 8x8. Bien que ce ne soit pas à proprement parlé un problème d'optimisation, il présente de nombreux avantages :

- il est visuel et facile à comprendre ;
- on peut coder la position des dames très simplement : pour chaque colonne, on note sur quelle ligne se trouve la dame. Par exemple, la position ci-contre sera notée [2, 4, 6, 8, 3, 1, 7, 5] ;
- on passe très facilement d'une configuration à une configuration voisine : il suffit d'échanger deux colonnes. Une position voisine (mais qui ne satisfait pas forcément les contraintes) de celle ci-contre est par exemple [2, 1, 6, 8, 3, 4, 7, 5].



Exemple de solution

On peut le traiter comme un problème d'optimisation si l'on considère qu'il faut minimiser le nombre de conflits (on parlera de *conflit* quand deux dames se menacent mutuellement). Il s'agira ici de placer *n* dames sur l'échiquier *n*x*n*, sans aucun conflit, en partant d'une solution avec une seule dame par ligne et par colonne (par exemple toutes les dames sur la diagonale) et en échangeant deux colonnes. On ne cherchera pas toutes les solutions possibles : une seule nous suffira. Notons que dans un problème d'optimisation classique, il n'y a en général qu'une seule meilleure solution. Ici il y en a plusieurs.

Il est à noter qu'il existe un algorithme permettant de trouver une solution quel que soit *n* supérieur à 3.

1. soit $r = n \bmod 12$
2. écrire les nombres pairs de 2 à *n*
3. si $r = 3$ ou $r = 9$ mettre le 2 à la fin de la liste
4. écrire ensuite les nombres impairs de 1 à *n,* mais si $r=8$ permuter les nombres impairs 2 par 2 (i.e. 3, 1, 7, 5, 11, 9, ...)
5. si $r = 2$, permuter les places de 1 et 3, puis mettre 5 à la fin de la liste
6. si $r = 3$ ou $r = 9$, mettre 1 puis 3 en fin de liste

Ainsi, pour $n$=8 on obtient la position [2, 4, 6, 8, 3, 1, 7, 5]. C'est la position qui est illustrée ci-dessus.
Pour $n$=14, on aura la solution [2, 4, 6, 8, 10, 12, 14, 3, 1, 7, 9, 11, 13, 5].
Pour $n$=15, on aura la solution [4, 6, 8, 10, 12, 14, 2, 5, 7, 9, 11, 13, 15, 1, 3].

L'intérêt de l'exercice n'est donc pas de trouver une solution, mais d'illustrer sur ce problème classique comment se comportent deux des métaheuristiques les plus connues : la *recherche avec tabous* et *le recuit simulé*.

## Première approche : descente de plus grande pente

À partir d'une position donnée, essayer toutes les permutations de deux colonnes et échanger les deux colonnes qui permettent de diminuer le plus le nombre de conflits.
Répéter le processus jusqu'à avoir 0 conflit (dans l'idéal) ou un blocage.

### Résultats avec la descente de plus grande pente

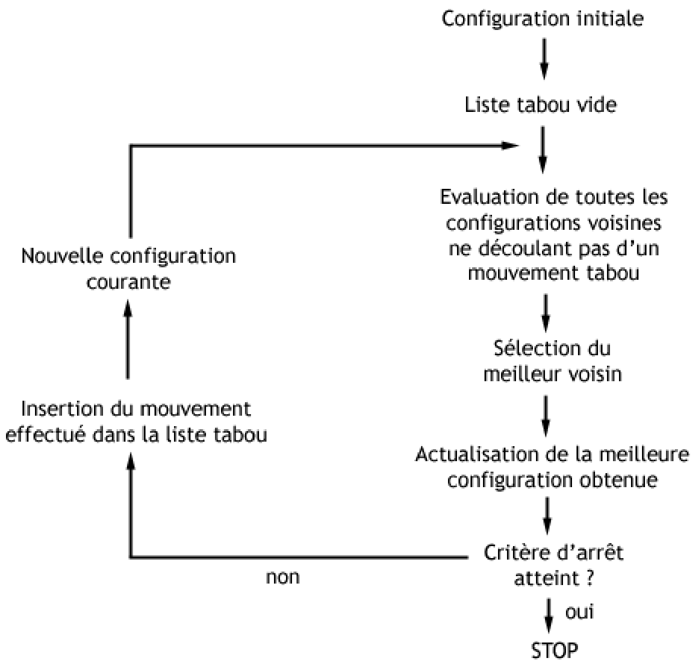
| Nombre de dames (*n*) | 20 | 40 | 60 | 80 | 100 |
|---|---|---|---|---|---|
| Temps ou nombre de conflits restants | 2 conflits | 13 sec. | 1 conflit | 418 sec. | 1308 sec. |

**Commentaires**

- Selon la position initiale et le nombre de dames, il arrive que l'algorithme se bloque dans un minimum local. C'est arrivé ici avec 20 et 60 dames placées au départ sur la diagonale.
- Avec 100 dames, on passe d'une situation avec 4950 conflits à une configuration sans conflit en 82 échanges de colonnes.
- Les temps sont seulement indicatifs et dépendent évidemment de l'ordinateur utilisé. Toutes les expériences ont été faites sur le même ordinateur et dans les mêmes conditions.

## Deuxième approche : recherche avec tabous

La *méthode taboue* est une métaheuristique d'optimisation présentée par Fred Glover en 1986. On trouve souvent l'appellation « recherche avec tabous » en français.
La méthode taboue consiste, à partir d'une position donnée, à explorer le voisinage et à choisir la position dans ce voisinage qui minimise la fonction objectif (comme dans la descente de plus grande pente). Il est essentiel de noter que cette opération peut conduire à dégrader la valeur de la fonction : c'est le cas lorsque tous les points du voisinage ont une valeur plus élevée. Le risque est qu'à l'étape suivante, on retombe dans le minimum local auquel on vient d'échapper. C'est pourquoi il faut que l'heuristique ait de la mémoire : le mécanisme consiste à interdire (d'où le nom de « tabou ») de revenir sur les dernières positions explorées.



Les positions déjà explorées sont conservées dans une file FIFO (First In First Out), souvent appelée liste des tabous, d'une taille donnée. Cette file doit conserver des positions complètes, mais cela ne pose pas de problèmes avec les *n* dames. Pour nos expériences, nous avons utilisé une liste de 10 mouvements tabous.

### Résultats avec la méthode taboue

| Nombre de dames (*n*) | 20 | 40 | 60 | 80 | 100 |
|---|---|---|---|---|---|
| Temps ou nombre de conflits restants | 1 sec. | 15 sec. | 114 sec. | 422 sec. | 1756 sec. |

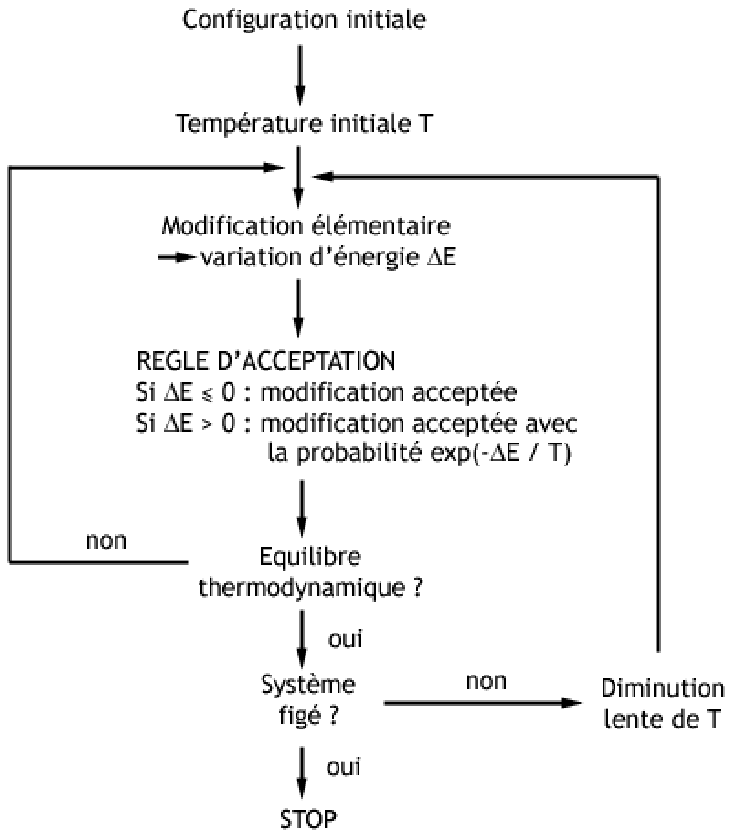
**Commentaires**

- Le programme ne se bloque plus dans un minimum local.
- La solution finale est toujours la même et dépend de la position initiale.
- Avec 100 dames, on passe d'une situation avec 4950 conflits à une configuration sans conflit en 95 échanges de deux colonnes.

## Troisième approche : recuit simulé

Le recuit simulé (*Simulated Annealing* en anglais) est une métaheuristique inspirée d'un processus utilisé en métallurgie. Ce processus alterne des cycles de refroidissement lent et de réchauffage (recuit) qui tendent à minimiser l'énergie du matériau. Elle est aujourd'hui utilisée en optimisation pour trouver les extrema d'une fonction.

Elle a été mise au point par trois chercheurs de la société IBM, S. Kirkpatrick, C.D. Gelatt et M.P. Vecchi en 1983, et indépendamment par V. Cerny en 1985.



Ce qu'on appelle « modification élémentaire » dans ce schéma consistera simplement à échanger deux colonnes choisies aléatoirement.

L'énergie sera dans notre problème le nombre de conflits.

Le réglage des paramètres est ici plus délicat. En particulier, comment faut-il faire baisser la température ? Trop vite, on risque de se bloquer dans un minimum local. Trop lentement, le temps de calcul augmentera, sans garantie de trouver une solution.

La solution retenue a été de faire baisser la température par palier de longueur 10, avec une température initiale $T$=100, avec $T_{k+1} = 0.6 \cdot T_k$, $k$ représentant un numéro de palier. C'est très inhabituel, car on prend généralement un coefficient proche de 1.

## Résultats avec le recuit simulé

| Nombre de dames (*n*) | 20 | 40 | 60 | 80 | 100 |
|---|---|---|---|---|---|
| Temps moyen en cas de succès* | 0.3 sec. | 1.5 sec | 4 sec. | 9 sec. | 23 sec. |
| Nombre de succès sur 10 essais | 7 | 10 | 10 | 10 | 10 |

*Il s'agit d'une moyenne sur 10 essais, puisque le hasard joue ici un rôle important. Les temps des essais où l'on n'a pas trouvé de solution n'ont pas été pris en compte.

### Commentaires

- Étant donné l'usage du hasard, on ne sait pas quelle solution finale sera trouvée. Ce ne sera pas toujours la même, contrairement à la recherche avec tabous.
- Curieusement, le recuit simulé marche le moins bien quand il y a peu de dames (7 succès seulement avec 20 dames).
- Il est par contre redoutable en temps de calcul, puisqu'il ne lui faut que 23 secondes pour trouver une solution avec 100 dames, alors que la méthode taboue en mettait 1756.

## Conclusion

J'ai conçu cette leçon sur les métaheuristiques dans le cadre de l'OC informatique, mais elle peut aussi être utilisées en maths appliquées. Ce sont les élèves qui ont cherché (et trouvé) les paramètres les plus efficaces, en particulier pour le recuit simulé.
J'ai aussi essayé de trouver une solution à l'aide d'un algorithme génétique, mais, pour l'instant, les résultats sont très décevants : cela ne marche quasiment jamais si l'on a plus de 16 dames. Cependant, les *n* dames permettent d'expliquer concrètement comment fonctionne cette métaheuristique.

## Références


- **Le problème des *n* dames**, http://www.apprendre-en-ligne.net/OCinfo/algo/ndames.html ; le mot de passe pour accéder aux programmes est « xxd »

- Wikipedia : **Eight Queens Problem**, http://en.wikipedia.org/wiki/Eight_queens_puzzle