Méthode *hill-climbing*

- Entrées :
  - nœud initial
  - fonction objectif à optimiser (notée *F(n)* dans les algorithmes)
  - fonction générant des nœuds successeurs (voisins)

- Méthode :
  - le nœud courant est initialisé au nœud initial
  - itérativement, le nœud courant est comparé à ses successeurs (voisins) immédiats
    - **le meilleur voisin *n'*** ayant la plus grande valeur *F*(*n'*) et tel que *F*(*n'*) > *F*(*n*) devient le nœud courant
    - **si un tel voisin n'existe pas, on arrête** et on retourne le nœud courant comme solution

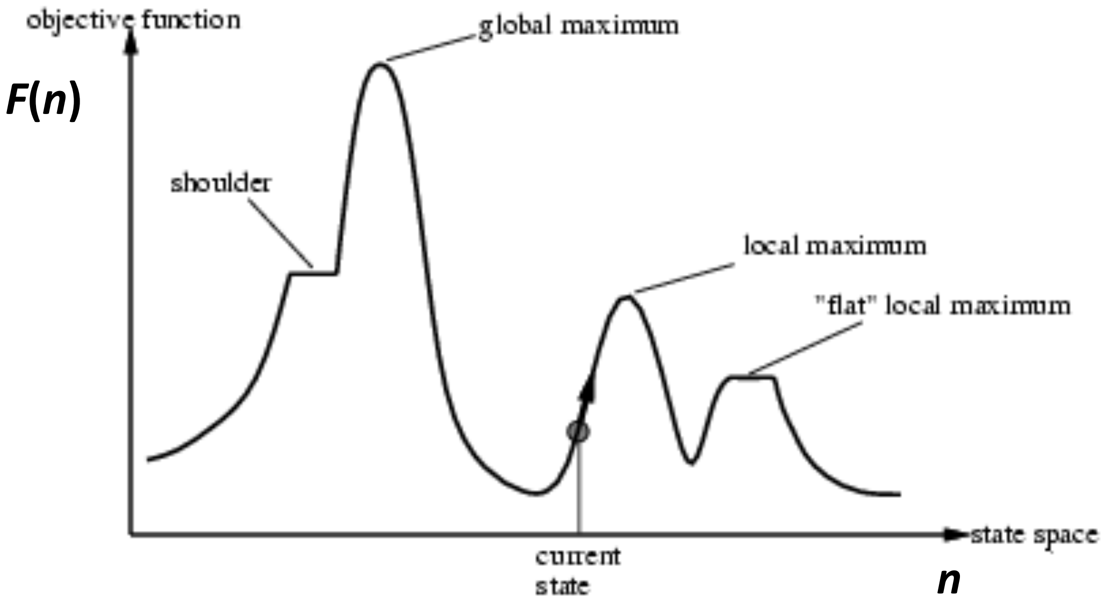
Hugo Larochelle et Froduald Kabanza

Algorithme *hill-climbing*

Algorithme HILL-CLIMBING(*noeudInitial*) // *cette variante maximise*

1. déclarer deux nœuds : *n*, *n'*
2. *n = noeudInitial*
3. tant que (1) // *la condition de sortie (exit) est déterminée dans la boucle*
   4. *n'* = noeud successeur de *n* ayant la plus grande valeur *F(n')*
   5. si *F*(*n'*) ≤ *F*(*n*) // *si on minimisait, le test serait F(n') ≥ F(n)*
      6. retourner *n* // *on n'arrive pas à améliorer p/r à F(n)*
   5. *n = n'*

Illustration de l'algorithme *hill-climbing*



Imaginez ce que vous feriez pour arriver au (trouver le) sommet d'une colline donnée, en plein brouillard et soufrant d'amnésie.

Exemple de simulation de *hill-climbing*

- Soit la fonction objectif suivante, définie pour les entiers de 1 à 16 :

n =	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
F(*n*) =	4	6	15	5	3	2	4	5	6	7	8	10	9	8	7	3

- Quelle valeur de *n* trouverait la méthode *hill-climbing* si la valeur initiale de *n* était 6 et que les successeurs (voisins) utilisés étaient *n-1* (seulement si *n>1*) et *n+1* (seulement si *n<16*)?

Exemple de simulation de *hill-climbing*

- Soit la fonction objectif suivante, définie pour les entiers de 1 à 16 :

?

n =	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
$F(n)$ =	4	6	15	5	3	2	4	5	6	7	8	10	9	8	7	3

- Quelle valeur de n trouverait la méthode *hill-climbing* si la valeur initiale de n était 6 et que les successeurs (voisins) utilisés étaient $n-1$ (seulement si $n>1$) et $n+1$ (seulement si $n<16$)?
- **Réponse**:
  - suite des valeurs de n parcourues: 6

Exemple de simulation de *hill-climbing*

- Soit la fonction objectif suivante, définie pour les entiers de 1 à 16 :

n =	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
F(*n*) =	4	6	15	5	3	2	4	5	6	7	8	10	9	8	7	3

- Quelle valeur de *n* trouverait la méthode *hill-climbing* si la valeur initiale de *n* était 6 et que les successeurs (voisins) utilisés étaient *n-1* (seulement si *n>1*) et *n+1* (seulement si *n<16*)?
- **Réponse**:
  - suite des valeurs de *n* parcourues: 6 → 7

Exemple de simulation de *hill-climbing*

- Soit la fonction objectif suivante, définie pour les entiers de 1 à 16 :

?

n =	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
F(*n*) =	4	6	15	5	3	2	4	5	6	7	8	10	9	8	7	3

- Quelle valeur de *n* trouverait la méthode *hill-climbing* si la valeur initiale de *n* était 6 et que les successeurs (voisins) utilisés étaient *n-1* (seulement si *n>1*) et *n+1* (seulement si *n<16*)?
- **Réponse**:
  - suite des valeurs de *n* parcourues: 6 → 7→ 8

Exemple de simulation de *hill-climbing*

- Soit la fonction objectif suivante, définie pour les entiers de 1 à 16 :

n =	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
F(*n*) =	4	6	15	5	3	2	4	5	6	7	8	10	9	8	7	3

- Quelle valeur de *n* trouverait la méthode *hill-climbing* si la valeur initiale de *n* était 6 et que les successeurs (voisins) utilisés étaient *n-1* (seulement si *n>1*) et *n+1* (seulement si *n<16*)?
- **Réponse**:
  - suite des valeurs de *n* parcourues: 6 → 7→ 8→ 9→ 10→ 11→ 12
  - *hill-climbing* termine et retourne *n*=12

Exemple : *N-Queens*

- Problème : Placer *N* reines sur un échiquier de taille *N*× *N* de sorte que deux reines ne s'attaquent pas mutuellement :
  - c-à-d., jamais deux reines sur la même diagonale, ligne ou colonne

Hugo Larochelle et Froduald Kabanza

Hill-Climbing avec 8 reines

- *n* : configuration de l'échiquier avec *N* reines
- *F*(*n*) : nombre de paires de reines
qui s'attaquent mutuellement
directement ou indirectement
dans la configuration *n*
- On veut le **minimiser**

- *F*(*n*) pour l'état (nœud) affiché : 17
- Encadré : les meilleurs successeurs,
si on bouge une reine dans sa colonne

Hill-Climbing avec 8 reines

- Un exemple de minimum local avec $F(n)$=1