TP1 - NK-Landscape Models

Introduction

Le problème considéré est celui de la recherche de maximum dans un "paysage rugueux" (i.e avec un grand nombre de sommet et de vallée). La présence de nombreux sommet rend difficile de trouver le maximum global, car il y a de forte change quand l'algorithme trouve un maximum, ce dernier ne soit qu'un minimum local.

Plus spécifiquement, le cas considéré ici est le cas d'une chaine binaire avec une fonction de "fitness" local définie sur des blocs de 1 à 3 bits. Le but étant de trouver la "fitness" globale maximale pour une chaine de bits de longueur N donnée.

Metaheuristic

Les approches considérées pour résoudre ce problème sont:

1. le grimpeur strict; dans ce cas, c'est toujours parmis toutes les variantes possibles $V(x)$ partant d'un état donné x, celle avec le meilleure score F qui est considérée comme devant être celle à prendre. Ce processus est répéter jusqu'à ce qu'un maximum soit atteint.

$$\max_{x^{'} \in V(x)} F(x^{'})$$

   L'avantage de cette méthode, c'est qu'elle est rapide pour trouver un maximum. Par contre, le désavantage est que si le maximum atteint est un maximum local, il n'y a aucune chance de trouver le maximum global.

2. le grimpeur probabiliste, dans ce cas, à chaque étape, le prochain pas est choisi avec une probabilité P dépendant de la "fitness" de chacune des possibilités.

$$P(x^{'}) = \frac{F(x^{'})}{\sum_{y \in V(x)} F(y)}, \text{ pour } x^{'} \in V(x)$$

   L'avantage de cette méthode est qu'il n'y a aucun risque de rester bloqué sur un maximum qui n'est pas local. Le désavantage est qu'il faut énormément d'itération pour réussir à parcourir tout l'espace et trouver le maximum global.

3. le grimpeur probabiliste avec aspiration, ce cas est une combinaison des deux algorithmes précédents. Ici, le nouvel état $x^{'}$ est choisi en utilisant la méthode du grimpeur strict s'il y a un état avec une "fitness" plus grande que la "fitness" actuel. Dans le cas contraire, c'est l'algorithme du grimpeur probabiliste qui est utilisé.

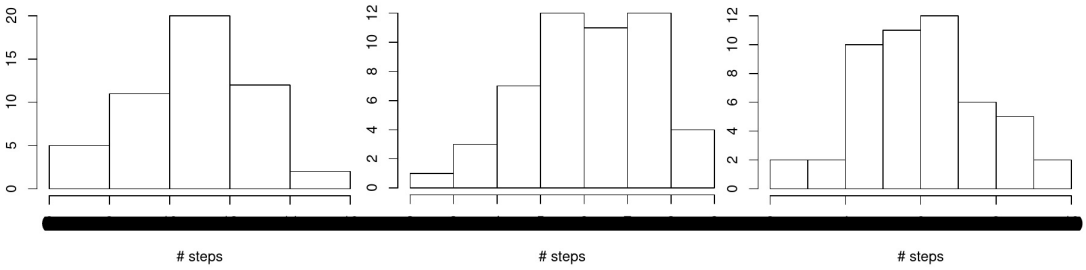
L'avantage de cette méthode est qu'elle permet d'atteindre rapidement un maximum, mais qu'elle ne reste pas bloquée sur ce maximum, mais peux en sortir, ce qui augmente la probabilité de trouver le maximum global.

Expérience

Détermination du nombre d'itération probabiliste

Afin de pouvoir définir le nombre d'itération nécessaire pour les algorithmes probabilistes, il est nécessaire de voir le nombre d'étape qui sont nécessaire à l'algorithme 1). Pour ce faire, les cas avec $K = 0$, $K = 1$ et $K = 2$ sont considérés.

Les résultats des simulations (Figure. 1) montrent qu'il y a une grande variation entre le cas $K = 0$ et les autre cas (Table 1). Cela s'explique par le fait que le cas $K = 0$ le maximum ne contient qu'une seule sort de bit et donc en moyenne, il faut 11.5 opérations pour mettre tout les bits à la même parité. Ce n'est évidement plus le cas avec $K > 0$.



(a) K=0 (b) K=1 (c) K=2

Figure 1: Distribution du nombre d'étape pour atteindre le maximum avec l'algo 1

Table 1: Quantiles des distributions du nombre d'étape pour atteindre le maximum avec l'algo 1

	K0	K1	K2
Min.	[redacted]		
1st Qu.	[redacted]		
Median	[redacted]		
Mean	[redacted]		
3rd Qu.	[redacted]		
Max.	[redacted]		

Pour les simulations avec un algorithme probabiliste, une recherche sur 70 itérations sera considérée, car le cas $K = 0$ peut-être vu comme un cas dégénéré.

Cas $K = 0$

Du point de vue de la fitness, la valeur trouvée est toujours la même (Table 2) (Figure 2) pour l'algo 1 et 3. Comme expliqué ci-dessus, la raison est, pour l'algo 1, que tous les bits finissent par valoir la même valeur qui maximise la fitness globale. [redacted] Par contre, [redacted] n'arrive pas à atteindre le maximum. La distance de Hamming (Figure 3) montre que les états obtenus par l'algo 2 sont très différents entre eux (8 bits de différents). [redacted]

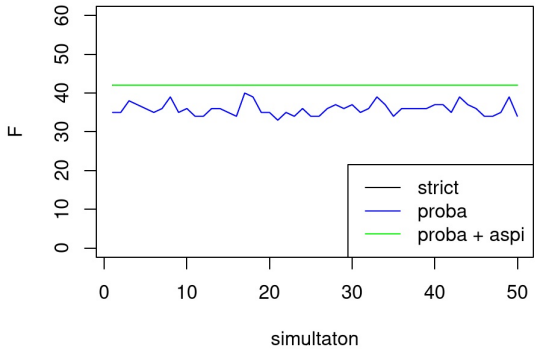
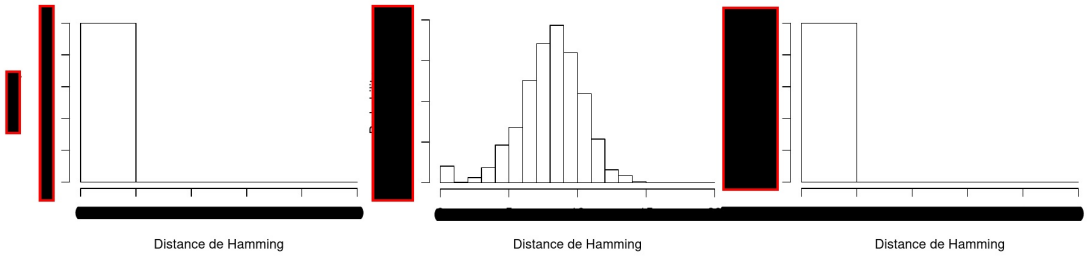


Figure 2: Fitness obtenue avec chaque algorithme pour K=0

Table 2: Quantiles des distributions de la fitness pour K=0 selon les algos

	Strict	Prob	Prob_aspi
Min.	[redacted]	[redacted]	[redacted]
1st Qu.	[redacted]	[redacted]	[redacted]
Median	[redacted]	[redacted]	[redacted]
Mean	[redacted]	[redacted]	[redacted]
3rd Qu.	[redacted]	[redacted]	[redacted]
Max.	[redacted]	[redacted]	[redacted]



(a) Strict (b) Proba (c) Proba+aspi

Figure 3: Distribution des distances de Hamming pour K=0

Cas $K = 1$

Pour ce cas, l'algo 3 permet d'obtenir la fitness maximal dans plus de la moitié des simulations (Table 3), alors que l'algo 1 n'y arrive en général pas. Cela fait bien apparaitre l'avantage de la méthode [redacted]

La distribution des distances de Hamming (Figure 5) montre qu'il y a une différence entre l'algo 1 et l'algo 3. Cela s'explique par la forme des maximums atteint par les différentes simulations. Dans le cas de l'algo 1, il y a [redacted] qui sont souvent atteint, ce qui explique la forte proportion de distance [redacted] Dans le cas de l'algo 3, les états avec la fitness maximal ne diffèrent entre eux que du shit de quelque bit, c'est pourquoi la distribution est beaucoup plus plate pour des distances faibles.

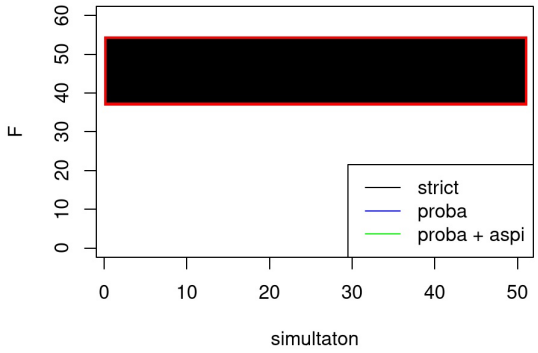


Figure 4: Fitness obtenue avec chaque algorithme pour K=1

Table 3: Quantiles des distributions de la fitness pour K=1 selon les algos

	Strict	Prob	Prob_aspi
Min.			
1st Qu.			
Median			
Mean			
3rd Qu.			
Max.			

(a) Strict (b) Proba (c) Proba+aspi

Figure 5: Distribution des distances de Hamming pour K=1

Cas $K = 2$

Dans ce cas, l'algo 3 se montre supérieur à l'algo 1 qui n'arrive pas à atteindre la valeur maximale (Table 4). l'augmentation de la rugosité, les approches stricte pur et probabiliste

Concernant la distance de Hamming, ce qui était observé avec $K = 1$ s'accentue encore ici (Figure 7). Le pic dans la distribution de l'algo 3, s'explique par la différence entre les solutions avec une fitness de 26 et celle de 25 qui ont 14 bits d'écart entre elle. La solution optimale étant:

et les sub-optimales:

[redacted] et [redacted]

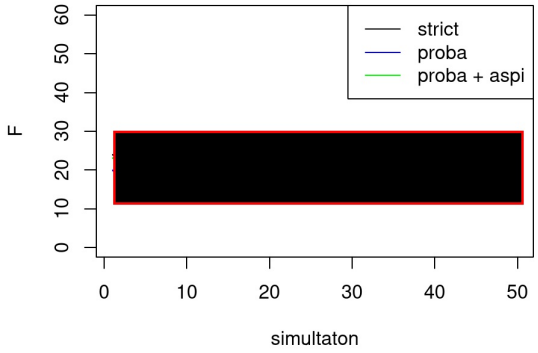


Figure 6: Fitness obtenue avec chaque algorithme pour K=2

Table 4: Quantiles des distributions de la fitness pour K=2 selon les algos

	Strict	Prob	Prob_aspi
Min.	[redacted]		
1st Qu.			
Median			
Mean			
3rd Qu.			
Max.			

(a) Strict (b) Proba (c) Proba+aspi

Figure 7: Distribution des distances de Hamming pour K=2

Discussion

Il apparaît clairement que [redacted] résultats(Figures 2, 4, 6), car dans [redacted] (algo 1). Une approche [redacted] l'obtenir de bon résultat, par contre, [redacted] es résultats obtenus [redacted] eux de l'algo 1.

L'approche de [redacted], dès que le nombre de maximaux locaux devient important, car il n'est alors pas possible de trouver autre chose que [redacted] vers lequel l'algorithme [redacted]