

Métaheuristique pour l'optimisation tp1

denizsungurtekin

September 2020

1 Introduction

Le but de cet exercice est d'étudier les modèles NK-Landscape et leur façon de maximiser une fonction. Nous étudierons plus particulièrement la méthode Déterministe de Hill Climbing et la méthode Probabiliste de Hill-Climbing pour maximiser le score ($F(x)$) d'une séquence de bits générée aléatoirement selon des fonctions f_0 , f_1 et f_2 données. Dans un premier temps nous expliquerons le fonctionnement de ces méthodes, leur rugosité selon le paramètre K , leur espace de recherche puis nous analyserons les résultats obtenus.

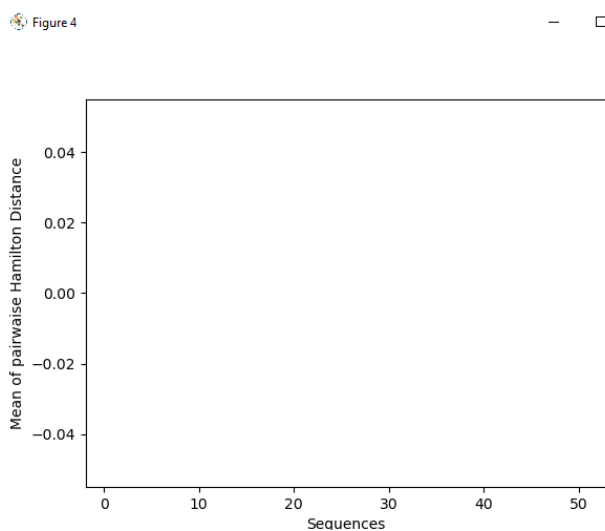
2 Méthodologie

Premièrement, il est nécessaire de définir nos trois fonctions f_0 , f_1 et f_2 qui détermineront la valeur de F selon le K choisi. K donne le nombre de bits qui sera évalué par f_0 si $K = 0$, f_1 si $K = 1$ et f_2 si $K = 2$. La relation bits-score est donc donnée par chacune de ces fonctions. Une fois ces fonctions définies, il nous est possible d'obtenir F en sommant l'output de ces fonctions sur une séquence de bits. Puisque nous voulons un score maximal, on utilise les méthodes Déterministe et Probabiliste de Hill-Climbing. Les deux versions sont assez similaires puisqu'elles consistent à chercher parmi les voisins d'une séquence de bits générée aléatoirement le meilleur score jusqu'à qu'aucun des voisins du noeud courant ne soit meilleur que lui-même. C'est donc uniquement dans la manière dont nous évalueront nos noeuds que les méthodes diffèrent. On dit qu'une séquence de bits est voisine d'une autre si elles diffèrent uniquement d'un seul bit, par conséquent une séquence de N bits possédera N voisins. Puisque pour l'algorithme Déterministe, nous avons obtenu en moyenne 10 étapes de recherche (nombre d'itération), l'espace de recherche est de 21×10 ($N = 21$). En ce qui concerne l'espace de recherche probabiliste, le nombre d'itération étant 10 fois plus élevé (spécifié dans l'énoncé), on a un espace de recherche de 21×100 . Cependant, le résultat final est trouvé bien avant par l'algorithme.

3 Résultats

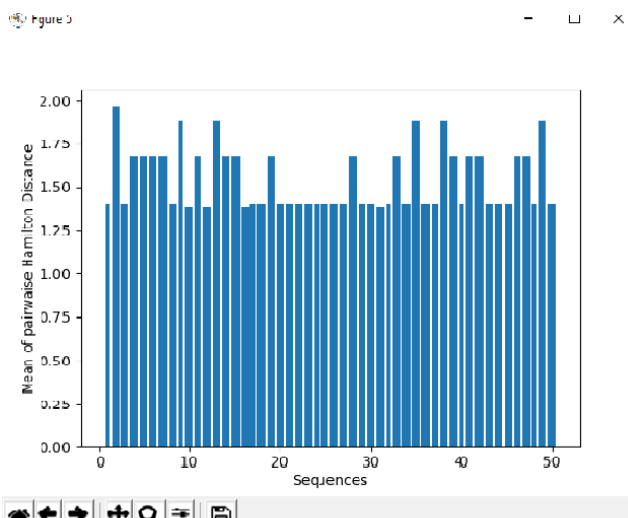
Pour chacune des valeurs de K , nous avons lancé 50 fois les deux méthodes d'optimisation afin d'étudier leur stabilité, voici un histogramme décrivant la distance de Hamming moyenne par pair pour chacun des 50 résultats obtenus.

Algorithme déterministe, $K=0$:



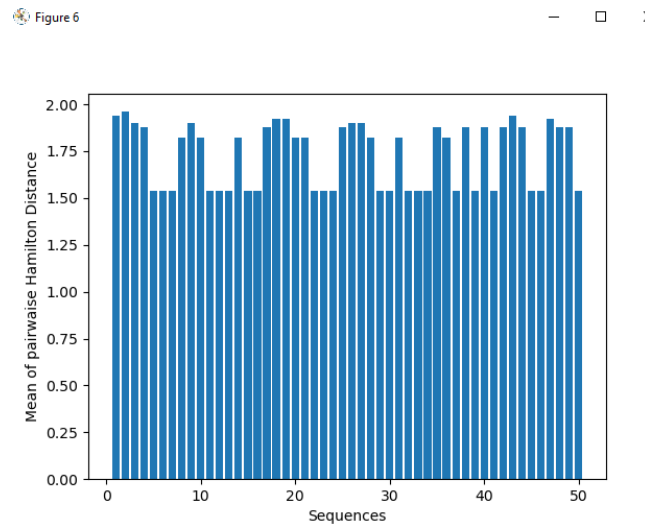
Toutes les solutions sont triviales (Une suite de 0), puisque $K = 0$ on choisit toujours le bit possédant le plus grand score.

$K=1$:



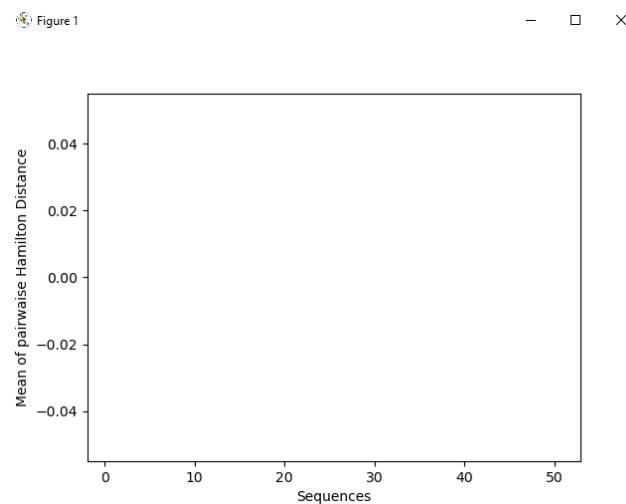
Ici, on remarque que la grande majorité des séquences possèdent en moyenne 1,4 bits de différence.

K=2:



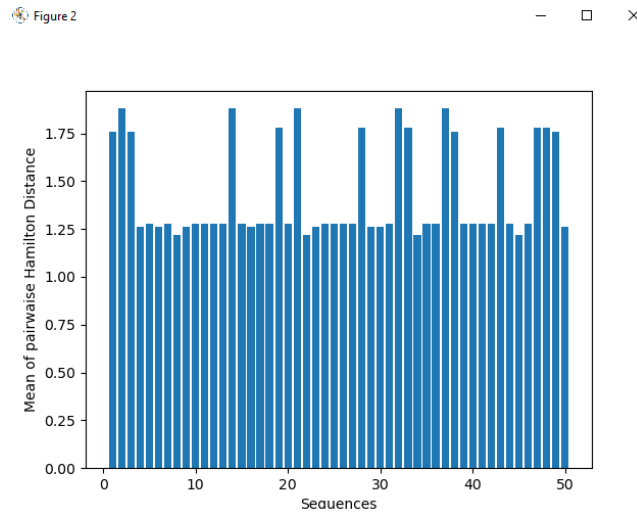
Ici, on remarque que la grande majorité des séquences possèdent en moyenne 1,5 bits de différence.

Algorithme probabiliste, K=0:



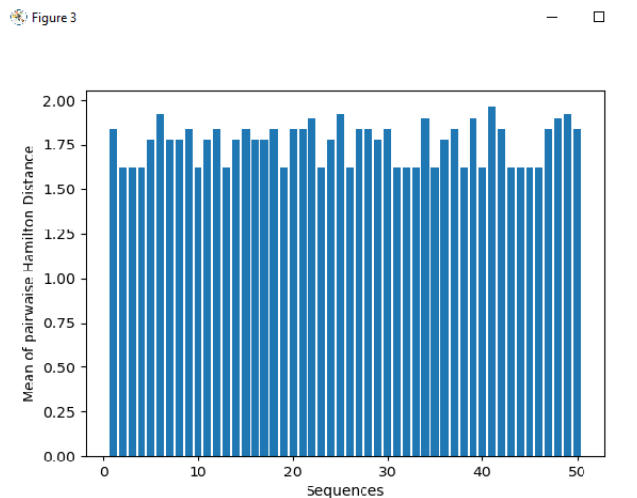
Même raisonnement que pour la méthode déterministe.

K=1:



Ici, on remarque que la grande majorité des séquences possèdent en moyenne 1,25 bits de différence.

K=2:



Ici, on remarque que la grande majorité des séquences possèdent en moyenne 1,6 bits de différence.

Globalement les deux méthodes possèdent donc, à notre échelle, la même stabilité. Cependant, on remarque donc que plus K augmente moins les solutions sont "rigides". Puisqu'il existe beaucoup plus de combinaisons possibles (dû aux nombre de bits qui détermine le score), le choix de la meilleur solution n'est pas toujours identique, ceci démontre que la solution de l'optimisation est un élément de l'espace de recherche.