

Programmer un Système Complexe :

Le problème de la synchronisation des fusiliers

Master 1 I2L
2013 / 2014

1 Description

Le problème de synchronisation d'une ligne de fusiliers (firing squad problem) est un problème qui consiste à trouver la règle locale d'un automate cellulaire uni-dimensionnelle où un état, appelé feu, doit apparaître pour la première fois sur toutes les cellules en même temps. Le nombre d'état de cet automate n'est pas limité à deux.

Plus précisément, il consiste à trouver la table des règles locales d'un automate cellulaire (AC) unidimensionnel, tel que, partant d'une configuration où toutes les cellules sont dans un même état de repos à l'exception d'une unique cellule dans l'état dit "général", l'AC évolue dans une configuration où toutes les cellules sont dans un même état (état dit de "feu"), état qui n'est jamais apparu auparavant. La table des règles locales doit être identique quelque soit la taille de la configuration. Le voisinage est de rayon 1. Le temps de synchronisation est le nombre d'itérations avant l'apparition de l'état feu. Le temps minimal qui permet de résoudre le problème quelque soit la longueur n de la configuration est $2n - 2$.

Ce problème a été défini en 1957 par J. Myhill, et une première solution a été publiée par Edward Moore en 1962 ; Jacques Mazoyer en 1982 en a proposé une solution impliquant six états en temps minimal. Il a été démontré également qu'il n'existe pas de solution pour un AC à quatre états. Pour une revue sur ce problème, vous pouvez consulter les travaux J.B. Yunes.

L'existence d'un automate à 5 états répondant exactement à ce problème de synchronisation reste à ce jour un problème ouvert.

Vous trouverez une description plus précise dans le mémoire de TER de A. Fulconis, H. Benouali, O. Bouhleb, P. Casanova.

2 But

Tenter de résoudre ce problème à 5 états avec d'un algorithme de recherche locale.

3 Représentation du problème

Voir mémoire de TER.

4 Travail Demandé

Le code minimal permettant d'exécuter un automate cellulaire et évaluer sa performance se trouve dans code.zip.

a - Donner la taille de l'espace de recherche et la taille du voisinage.

d - Coder la métaheuristique Hill-Climbing First-improvement.

g - Réaliser un grand nombre d'exécutions de votre métaheuristique.

d - Coder une recherche "Iterated Local Search".

nota. Prendre grand soin des cas d'égalité.

g - Comparer statistiquement les deux métaheuristicues par l'exécution d'un nombre suffisant de chacune d'elle. Vous présenterez au moins pour chaque métaheuristique, la valeur moyenne, l'écart-type et le maximum de la performance finale selon les valeurs du paramètres que vous avez étudié.

h - Analyser les résultats de vos expériences.

Et si vous voulez (et le temps!), apportez des améliorations à l'algorithme...

Une archive contenant l'ensemble des documents sera à envoyer par email. Votre rapport écrit devra être au format pdf.

Grosse récompense promise à ceux qui trouvent une même règle de synchronisation pour toutes les longueurs entre 2 et 21.