INF6102 – Métaheuristiques

Travaux pratiques n°4 – Covering Array

*Kévin Baumann – 1647505  
Florian Korsakissok - 1628087*

# I – Présentation du problème

Le problème traité est celui du « Covering Array », ou autrement dit d’une matrice de couverture. La résolution d’une instance d’un tel problème dépend de deux paramètres notés v et k. L’objectif est de remplir une matrice contenant k colonnes avec un minimum de lignes, à l’aide d’entiers compris entre 0 et v-1, de telle sorte que pour chaque paire de symboles, et pour chaque paire de colonnes, au moins une ligne de la matrice contienne cette paire de symboles sur cette paire de colonnes.

Plus formellement, le problème peut être défini comme suit :

* Données d’entrée : v, k
* Sortie de l’algorithme : une matrice M à coefficients dans [0 ; v-1] de taille N\*k
* Contraintes : pour tout i, j dans [0 ; k-1], pour tout a, b dans [0 ; v-1], il existe une ligne l telle que M[l][i] = a et M[l][j] = b
* Objectif : minimiser N

N est la grandeur à minimiser : on veut se retrouver avec la matrice la plus petite possible, c’est-à-dire écrire aussi peu de lignes que possible dans le résultat. De ce fait, N peut être vu comme un coût dans le cadre de ce problème.

Dans l’implémentation de l’algorithme proposé, on utilisera souvent la notion de « contrainte élémentaire ». Une contrainte élémentaire est un quadruplet (i,j,a,b), et on dira que cette contrainte est satisfaite si il existe dans la matrice en cours de construction une ligne l telle que M[l][i] = a et M[l][j] = b.

# II – Opérateurs de variation

## A – Description de l’opérateur de mutation

Blabla

## B – Description de l’opérateur de croisement de base

Blabla

## C – Description de l’opérateur de croisement spécialisé

Blabla

## D – Evolution de la diversité de la population

Blabla

# III – Description de l’algorithme évolutionniste

## A – Pseudo-code

Blabla

## B – Caractéristiques

Blabla

## C – Liste des paramètres

Blabla

# IV – Résultats expérimentaux

## A – Présentation des données utilisées

Les données utilisées à des fins de test sont constituées de 7 exemplaires, chacun correspondant à un couple (v,k) différent. Les cas traités par la suite sont les suivants :

* v = 2, k = 4 - au mieux, avec l’algorithme tabou, on a pu avoir N = 5.
* v = 3, k = 20 - au mieux, avec l’algorithme tabou, on a pu avoir N = 17.
* v = 3, k = 60 - au mieux, avec l’algorithme tabou, on a pu avoir N = 22.
* v = 5, k = 10 - au mieux, avec l’algorithme tabou, on a pu avoir N = 37.
* v = 5, k = 15 - au mieux, avec l’algorithme tabou, on a pu avoir N = 43.
* v = 8, k = 10 - au mieux, avec l’algorithme tabou, on a pu avoir N = 94.
* v = 8, k = 15 - au mieux, avec l’algorithme tabou, on a pu avoir N = 109.

## B – Spécifications de la machine de test

Coefficient trouvé par dfmax : 5.40

Coefficient d'ajustement : 8.6/5.4 = 1.59

|  |  |
| --- | --- |
| **Système d’exploitation** | Ubuntu 13.10 64 bits |
| **Mémoire vive** | 7.7 GiB |
| **Processeur** | Intel® CoreTM i7-3610QM CPU @ 2.30GHz × 8 |

## C – Présentation des tests préliminaires

Blabla

## D – Résultats obtenus

Blabla

## E – Commentaires

Blabla