

Projet de Graphes et Optimisation Combinatoires

DACHY Corentin, HUYLENBROECK Florent, JOSSE Thomas

20 décembre 2018

Table des Matières

1 Méthodes Débattues

- GRASP
- Recherche Tabou

2 Méthode implémentée

- Fourmis pour les QAPs
- La colonie de fourmi hybride
- Fonctionnement
- Implémentation

3 méthodes débattues :

- GRASP
- Recherche Tabou
- Colonie de fourmis

2 Choiesies :

- Recherche Tabou
- Colonie de fourmis

1 Implémentée :

- Colonie de fourmis hybride

- Idée Générale
 - ① Glouton avec Proba
 - ② Recherche locale
- Pourquoi ?
 - ① Point de départ pour autres algos
 - ② Abandonné après début colonie de fourmis hybride

- Idée Générale

- 1 Voisinage = toute permutation de 2 éléments.
(Ex : Pour $[1,2,3]$ le voisinage serait : $[[2,1,3],[3,2,1],[1,3,2]]$)
- 2 Deux types de recherche tabou : par mouvement et par solution
- 3 Longueur de la liste Tabou : n (+x) mais décroissant
- 4 La recherche effectue N itérations en mouvement, ensuite, on effectue une recherche par solution jusqu'à ce que l'on n'améliore plus notre solution.

- Gardée pour l'implémentation finale du projet (Voir slide 7)

Afin d'appliquer la colonie de fourmis aux problèmes d'affectation quadratiques, il faut voir la matrice des phéromones non plus comme "La probabilité de passer de i à j " mais plutôt "La désirabilité que $s_i = j$ dans notre solution s ".

Les fourmis ne construisent plus une solution à partir d'un point de départ, mais explorent le voisinage d'une solution de départ.

L'hybridation apportée à cette colonie de fourmi consiste en deux points :

- Les solutions initiales sont affinées par une recherche locale (*fast-start*) ou par une recherche tabou (*slow-start*)
- Chaque fourmi, lors d'une itération, applique elle-même une recherche locale à sa solution modifiée.

Et fonctionne selon deux modes :

- Une phase d'intensification : les fourmis gardent toujours la meilleure de leurs solutions.
- Une phase hors intensification : les fourmis gardent toujours la solution générée par leurs permutations.

Une phase plus rare de diversification vient parfois s'ajouter.

Lors d'une itération, une fourmi k applique à sa solution s^k :

- Un certain nombre ($N/3$) de permutations basées sur la matrice des phéromones $\rightarrow s'^k$.
- Une recherche locale $\rightarrow s''^k$.
- Si intensification : $s''^k \leftarrow \max(s^k, s''^k)$

Ensuite, après évaporation des phéromones (par un facteur $\alpha = 0.9$) seule la meilleure fourmi dépose les siennes, ce qui rend la recherche assez intense.

Pour contrer cela, si un certain nombre d'itérations ($N/2$) se sont déroulées sans améliorer la meilleure solution s^* , une remise à zéro des solutions tenues par les fourmis est effectuée, en ne gardant que la meilleure solution s^* dans une fourmi. Une recherche tabou est ensuite effectuée à la place d'une recherche locale pour la première nouvelle itération.

Le nombre de fourmis est fixé à 10.

En ce qui concerne l'implémentation, les fourmis nous semblaient avantageuses pour une raison, la possibilité de lancer chaque fourmi dans un thread. Les étapes lors de l'implémentation ont été :

- Premier jet en python3 (Colonie de fourmi + recherche locale).
- Traduction en C++.
- Ajout du *multi-threading*.
- Ajout de la recherche tabou lors de l'initialisation.

Chaque étape ayant mené à une amélioration de la vitesse ou de la qualité de la solution trouvée.