

Projet d'optimisation : Etape 4

Toure Hosseine Habraham Youssef

Raulin Vivien

L'objectif de ce projet était de mettre en œuvre les notions appréhendées pendant le cours d'optimisation combinatoire avec un cas concrets d'étude.

Notre problème portait sur l'affectation des encarts à différentes revues dans le but d'obtenir le meilleur profit possible mais en respectant leurs contraintes d'accueil (disponibilité nominale et capacité de dépassement).

Pour ce faire, nous avons supposé que chaque encart était compatible avec au moins une revue.

Nous avons pour cela utilisé un générateur aléatoire qui nous permet de générer aléatoirement des données à chaque fois que l'on test nos implémentations. Ce générateur aléatoire s'affiche juste au-dessus des résultats lorsque l'on lance le code et affichent les données qu'il a généré.

- La première étape de ce projet fût de faire une formalisation mathématique de ce problème en termes de Programmation Linéaire en Nombre Entiers (PLNE).

Avec cette modélisation mathématique du problème, nous avons pu déterminer 2 variables de décision qui indiquent si l'encart j est affecté à la revue k et s'il y des dépassements cumulés en nombre de zones déclenché sur la revue k en plus de plusieurs contraintes imposées.

- La deuxième étape de ce projet fut de concevoir des méthodes heuristiques et méta-heuristiques nous permettant de résoudre notre problème rapidement mais pas de façon optimale. En effet, nous avons proposé des heuristiques dans le but d'utiliser des méthodes approchées pour trouver des « bonne solutions » à notre problème sans chercher à avoir l'optimum du problème. Les solutions que l'on obtient correspondent donc à des bornes inférieures de notre problème.

Nous avons pour cela tenté de concevoir une heuristique gloutonne, une méta-heuristique du recuit-simulé, une autre méta-heuristique de recherche tabou et enfin une dernière méta-heuristique de l'algorithme génétique.

Nous avons décidé d'implémenter notre heuristique gloutonne et notre méta-heuristique du recuit simulé car cela nous semblait être les méthodes les plus abordables en termes de programmation ; de plus l'algorithme génétique est une

méta-heuristique réputée comme étant difficile à implémenter du fait de sa grande complexité.

Pour notre heuristique gloutonne, nous obtenons des résultats plutôt satisfaisants. En effet, le profit global obtenu est très souvent autour des 10k euros. En lançant notre programme avec à chaque fois des valeurs de revenus et d'encombrement différents nous avons observé un profit qui variait la majorité des cas entre 5k euros et 20k euros. Nous sommes donc satisfaits de ce résultat.

Cependant, le dépassement étant très souvent de 0 nous ne sommes pas certains que notre heuristique tient parfaitement compte des coûts de dépassement.

Nous avons ensuite implémenté notre méthode du recuit simulé. Nous avons eu plusieurs difficultés notamment pour bien faire la jonction entre le principe de température et d'intensification avec nos contraintes mais avons quand même réussi à obtenir des résultats pour cette méthode même si nous avons conscience qu'ils ne sont pas parfaits.

En effet, les résultats obtenus pour le recuit simulé sont globalement moins intéressants que pour notre heuristique gloutonne. En effet, le profit obtenu est très variant et peut même être quelquefois négatif car nous avons suivant les données parfois des pertes liées au dépassement qui sont très élevée pouvant dépasser les 100k euros. Ces valeurs très élevées liée au dépassement nous laissent supposer que nous avons très probablement mal raisonné au niveau du code pour coder nos coûts de dépassements.

- La troisième étape de ce projet fût de trouver cette fois-ci des bornes supérieures de l'optimum. Pour cela, nous avons déterminé des techniques pour dans un premier temps trouver des bornes supérieures triviales en relaxant les contraintes qui nous gênaient le plus pour résoudre notre problème. Cela nous a permis de grandement simplifier le problème pour le rendre polynomiale.

Ensuite, nous avons déterminé des bornes supérieures lagrangienne. Pour cela, nous avons dualisé les contraintes qui nous gênaient le plus avec des multiplicateurs de Lagranges. Pour la phase 4 de ce projet, nous avons essayé de coder notre borne lagrangienne UB3 (cf : en commentaire dans le code source java, ligne 335) en implémentant nos 2 sous-problèmes lagrangiens avec l'algorithme du sous-gradient afin de résoudre notre problème dual. Malheureusement, nous n'avons pas réussi à obtenir de solution, de ce fait nous n'avons aucun résultat pour notre borne supérieure et donc aucune valeur du profit.

- Implémentation de l'heuristique gloutonne, de la méta-heuristique du recuit simulé et de la borne supérieure lagrangienne UB3.

