RÉSUMÉ DE LA SPÉCIFICATION DE LA RECHERCHE TABU POUR LE PROBLÈME DU SAC À DOS À PLUSIEURS CONTRAINTES

16 novembre 2020

1. Le problème

le problème est celui du sac à dos à multiple contraintes, qui est un problème d'optimization combinatoire. Celui ci est une variante dans laquel on a plus que juste un poids maximal a ne pas dépasser mais plusieurs contraintes que dois satisfaire toute solution possible (combinaison d'objets) pour etre considéré comme admissible. il peut etre formulé formellement comme suit :

$$Max \sum_{j=1}^{n} c_j x_j$$

$$sous \quad contraintes :$$

$$\sum_{j=1}^{n} a_{ij} x_j \leq b_i, \forall i \in \{1, 2, .., m\}$$

$$x \in \{0, 1\}$$

n est le nombre d'articles et m est le nombre de contraintes du sac à dos. Les valeurs c_j peuvent être interprétée comme la valeur de l'inclusion des différents éléments, les valeurs a_{ij} peuvent être interprétées comme des mesures de poids ou de volume pour chaque article, et les valeurs b_i sont des limites pour chaque contrainte. Les valeurs x_i représentent les éléments, qui reçoivent la valeur 1 s'ils sont inclus dans le solution, 0 sinon. Le problème est alors de trouver la meilleure combinaison d'articles pour inclure dans la solution, sans violer aucune des contraintes. Il y a 2^n différentes solutions au problème, certaines d'entre elles sont irréalisables car elles violent une ou plusieurs des contraintes. Le problème est connu pour être NP-difficile et beaucoup de travail a été fait pour trouver de bons algorithmes pour le résoudre.

2. La Solution

Pour résoudre ce problème il est question de définir un algorithme basé sur la Recherche Tabu, une méta heuristique qui permet de faire des meileurs choix de solution basé sur l'information des anciennes recherches stocker temporairement dans une file de taille limité, notament la fréquence et l'ancienneté dans notre cas.On va essayer ici de faire des oscillations (déplacements) au niveau de la limite entre les solutions admissibles et inadmissibles, stockant a chaque fois la nouvelles meileure solution (si on en a trouvé), car on c'est que l'optimum se trouve quelque part dans ces environs et que étant à une solution donné on peut arriver à la solution optimale avec au trop n déplacements reste a trouver lesquels.

3. Description de la solution

La solution comporte 3 phases principales :

3.1 La solution initiale :

La solution initiale est vide (que des 0s). Ceci permet de commencer avec une solution admissible et d'après le document de reférence combiné avec l'exploration basé sur les frequences des variables c'est avéré plus fiables que des solutions généré aléatoirement combiné avec la notion de redémarrage de l'algorithme pour explorer d'autres régions de l'espace des solutions.

3.2 La Recherche :

La Recherche comporte deux phases principales : l'évaluation et le déplacement, l'évaluation va attribué une note a chaque variable définissant la probabilité de chacun d'etre dépalcé (inversé), celle du déplacement consiste juste a inversé des variable données en prenant garde aux évènements critiques que l'on vera.

3.2.1 L'évaluation :

Ce fait en deux parties qui séront combiné par la suite : l'évaluation des contraintes puis celles des pénalités.

3.2.1.1 Les contraintes :

D'abord on camcume le reste comme suit :

$$b_i = b_i - \sum_{j=1}^{n} (a_{ij} : for \quad j \quad with \quad x_j = 1) \quad \forall i \in \{1, 2, .., m\}$$

Un reste positif signie une consomation incomplète hors un négatif dénote un débordement, si il est zero alors le b_i a été entièrement consomé.De là on peut calculé le coéfficient de chaque contrainte comme :

$$w_i = 1/b_i, si \quad b_i > 0$$

$$w_i = 2 + |b_i|, si \quad b_i \leq 0$$

Le choix de la variable a dépalcé pour la phase contructive est obtenu comme suit :

$$max(c_j/s_j : x_j = 0)$$

et pour la phase destructive :

$$min(c_j/s_j : x_j = 1)$$

où s_j est défini comme :

$$s_j = \sum_{1}^{n} w_i a_{ij}$$

3.2.1.2 Les pénalités :

Les penalités sont appliqué sur la base le la fréquence (d'apprision dans les solutions trouvés) et l'ancienneté. Ils sont garant de l'exploration en évitant d'avoir a chaque fois les même variables qui ont engendré les meilleurs solutions passés dans les nouvelles solutions contruites. La pénalité d'ancienneté est calculé comme suit :

$$PenAncien = N_j * PenR$$

ou N_j est le nombre de fois que la variable x_j a apparu (égal a 1) dans une meilleur solution et $PenR$ est la somme maximale des sommes des colonnes de la matrices des contraintes (la matrice des contraintes est une définit à partir des contraintes avec chaque ligne représentant une contrainte et chaque colonne une variable, à chaquecase on a donc le coéfficient de cette variable dans la contrainte donné).
La pénalité de fréquence elle est obtenu comme suit :

$$PenF = PenR/(C * curIter)$$

$PenR$ est le même qu'en haut, $curIter$ est la position de l'iteration courante et C une contante fixé a 10000.
Ces pénalités sont donc combiné avec les contraintes lors du choix de la variablea dépalcé.

3.2.2 Le déplacement :

Le déplacement consiste juste a inversé la valeur des variable, ceci autant de fois que l'amplitude le décide (l'amplitude ici est un nombre qui définit le nombre de déplacement a faire a chaque itération, dans le document de reférence il défini comme on nombre aléatoire entre 1 et 6 choisi à chaque itération). Le déplacement ce fait en deux phase : la phase contructive et destructive.Lors de la phase contructive les variable a 0 sont sectioné selon la note et inversé a 1 autant de fois que l'amplitude le désigne.Lors de cette phase il peut arrivé que l'inversement d'une variable crée une solution inadmissible (on a franchi la frontière entre les solutions admissibles et inadmissibles), ceci est appellé un évènement critique. Dans ce cas avant que l'inversement n'arrive on essaye de trouver une nouvelle meileure solution en inversant s'autres variables qui maintiennent la solution admissible, si on la trouve elle est ajouté à la liste Tabu et les fréquences de variables concernés sont mises à jour.La longeur de la liste Tabu ("Tabu Tenure") est fixé à 10 dans le document de reférence.
Dans la phase destructive les variables à 0 sont mis à 1 selon l'amplitude.

3.3 La Fin :

L'algorithme ne s'arrete que quand il a atteint le nombre maximal d'itération fixé a 100000 dans le document de reférence, mais en pratique nous pourions prendre un nombre plus pétit su a notre puissance de calcul réduite.