Devoir Maison : Relaxation Lagrangienne

Lin Hirwa Shema, Martin Debouté

May 6, 2022

1 Sac à dos disjonctif

On s'intéresse au problème du sac à dos disjonctif. On dispose de n objets indicés de 1 à n. On note $I = 1, ..., n$. Chaque objet a une taille c_i, et un profit p_i. La taille du sac à dos est C. On dispose en plus d'un graph (I, E) qui représente des conflits entre objets. Si $(i, j) \in E$ alors on ne peut pas sélectionner i et j dans une solution. Le problème peut se modéliser de la manière suivante.

$$
\begin{array}{lllr}
\max & \sum_{i \in I} p_i x_i & & (1) \\
\text{s.c.} & \sum_{i \in I} c_i x_i \leq C & & (2) \\
& x_i + x_j \leq 1 & \forall (i, j) \in E & (3) \\
& x_i \in \{0, 1\} & i \in I & (4)
\end{array}
$$

2 Relaxation lagrangienne

2.1 Relaxation de la contrainte de sac à dos

Soit $u \in \mathbb{R}$ notre multiplicateur lagrangien fixé pour la contrainte de sac à dos. Notre modèle, ayant la contrainte (2) relâchée, devient:

$$
\begin{array}{lllr}
\min & \sum_{i \in I} -p_i x_i + u(\sum_{i \in I} c_i x_i - C) & & (1') \\
\text{s.c.} & x_i + x_j \leq 1 & \forall (i, j) \in E & (3) \\
& x_i \in \{0, 1\} & i = i \in I & (4)
\end{array}
$$

2.2 Relaxation des contraintes de disjonction

Soit $v_{ij} \in \mathbb{R}, (i, j) \in E$ nos multiplicateurs lagrangiens fixés pour les contraintes de disjonction. En relâchant les contraintes (3), on obtient le modèle:

$$
\begin{array}{lllr}
\min & \sum_{i \in I} (-p_i x_i + \sum_{j:(i,j) \in E} v_{ij}(x_i + x_j - 1)) & & (1'') \\
\text{s.c.} & \sum_{i \in I} c_i x_i \leq C & & (2) \\
& x_i \in \{0, 1\} & i \in I & (4)
\end{array}
$$

3 Programmation Dynamique

Étape 1:

On créer un tableau T avec $(n+1)$ nombre de lignes et $(w+1)$ nombre de colonnes que l'on remplis de zéros.

Étape 2:

On commence à remplir le tableau de haut en bas, de gauche à droite avec la formule de récurrence :

$T(i, j) = max\{T(i-1, j), p_i + T(i-1, j–c_i)\}$

Ici, $T(i, j)$ = valeur maximale des éléments sélectionnés si nous pouvons prendre les éléments 1 à i et avoir les restrictions de poids de j. Cette étape conduit à remplir entièrement le tableau. Ensuite, la valeur de la dernière case représente la valeur maximale possible du sac à dos.

Étape 3:

Pour identifier les éléments qui doivent être mis dans le sac à dos pour obtenir ce profit maximum, considérez la dernière colonne du tableau.

Commencez à scanner les entrées de bas en haut.

Lorsque vous rencontrez une entrée dont la valeur n'est pas la même que la valeur stockée dans l'entrée immédiatement au-dessus, marquez l'indice de la ligne de cette entrée.

Une fois toutes les entrées scannées, les indices marqués représentent les éléments qui doivent être placés dans le sac à dos.

Complexité temporelle:

Chaque entrée du tableau nécessite un temps constant O(1) pour son calcul. Il faut $O(nw)$ temps pour remplir $(n+1)(w+1)$ entrées de table. Il faut $O(n)$ temps pour retrouver la solution puisque le processus parcours les n lignes. Ainsi, le temps global $O(nw)$ est pris pour résoudre le problème du sac à dos en utilisant la programmation dynamique.