Université Paris Cité

Département D'INFORMATIQUE

Un CSP pour représenter le problème d'allocation de ressources

Salah BOUCHAMA

Master 2 IAD

15 février 2023

1 Description du problème

Je travaille à temps partiel dans une entreprise qui fournis ces services de livraison au géant du e-commerce Amazon. Le gérant de cette entreprise de livraison souhaite automatiser la planification de ses employés sur les jours de la semaine. Plus précisément, il doit désigner la liste des employés qui vont travailler en chaque jour de la semaine, en prenant en compte la disponibilité de chacun d'entre eux et le nombre d'employés demandé par Amazon, pour chaque jour de la semaine.

Cette affectation (employé - jours travaillés) doit se faire d'une manière la plus équitable possible. En effet, la répartition du travail doit se faire uniformément parmi tous les employés en fonction de leurs disponibilités, c'est à dire qu'on ne doit pas solliciter un employé x plus souvent qu'un autre employé y, or que les deux travailleurs ont des disponibilités similaires.

Un employé qui travaille sur deux ou plusieurs jours consécutifs va utiliser le même camion pour ses livraisons, c'est pour cela que idéalement, il est préférable qu'un employé soit affecté sur des jours consécutifs, pour éviter qu'il ai à remettre le camion utilisé pour ses livraisons, à l'un de ses collègues planifié à travailler le lendemain.

2 Modélisation du problème

En considérant que chaque employé est une ressource, et que chaque jour de la semaine est une tâche à accomplir, on peut traiter le problème de planification des employés de l'entreprise sur les sept jours de la semaine, comme un problème d'allocation de ressources.

2.1 Le modèle proposé

2.1.1 Les données du problème

Nous allons utiliser les notations suivantes pour décrire les données du problème. Soient :

- $E = \{1, 2, \ldots, m\}$, l'ensemble des m employés de l'entreprise ;
- $J = \{1, 2, \ldots, n\}$, l'ensemble des jours d'activité de l'entreprise ;
- $NE = \{ne(j), j \in J\}$, où ne_j est le nombre d'employés demandé par amazon pour un jour $j \in J$;
- $Disp = \{disp(i, j),\ i,\ j \in J\}$, où $disp(i, j) = 1$ si l'employé i est disponible pour travailler le jour j, 0 sinon.

2.1.2 Les variables de décision

Pour le CSP associé à ce problème d'allocation de ressources, les variables de décisions correspondent aux variables d'affectation des employés selon les jours de la semaine.

Nous allons les noter par :

$$x = \{x_{ij},\ \forall i \in E,\ \forall j \in J\}$$

l'ensemble des variables de décisions, où $x_{ij} = 1$ si l'employé i travaille le jour j, 0 sinon.

Finalement, le modèle CSP en construction est à $m \times n$ variables de décision. Chaque variable est de domaine $D_{ij} = \{0, 1\},\ \forall i \in E,\ \forall j \in J$

2.1.3 Les contraintes du problème

Les variables de décision définies précédemment sont soumises aux contraintes suivantes :

1. Le nombre d'employés travaillant en une journée donnée ne doit pas dépasser le nombre d'employé nécessaires pour cette journée :

$$\sum_{i \in E} x_{ij} \times disp(i, j) = ne(j), \quad \forall j \in J \tag{1}$$

2. Le nombre d'employés sollicité en une journée $j \in J$ ne doit pas dépasser le nombre total d'employés disponible en cette journée :

$$\sum_{i \in E} x_{ij} \leq \sum_{i \in E} disp(i, j), \quad \forall j \in J \tag{2}$$

3. Le nombre de jours qu'un employé $i \in E$ est sollicité ne doit pas dépasser le nombre total de ses jours de disponibilité :

$$\sum_{j \in J} x_{ij} \leq \sum_{j \in J} disp(i, j), \quad \forall i \in E \tag{3}$$

4. Les variables de décision sont des variables en 0 et 1 :

$$x_{ij} \in \{0, 1\}, \quad \forall i \in E,\ \forall j \in J \tag{4}$$

5. De plus, pour représentez l'équitabilité dans la répartition du travail sur les employeurs, nous allons imposer une contrainte qui stipule que pour des employés ayant le total de nombres de jours de disponibilité égaux à un seuil $S > 0$ près, le nombre totale de jours de travail pour ces employés doit être le même au paramètre S d'écart. Formellement, on va exprimer cela par la relation :

$$\left| \sum_{j \in J} x_{ij} - \sum_{j \in J} x_{\tilde{i}j} \right| \leq S, \quad \forall (i, \tilde{i}) \in E^2,\ i \neq \tilde{i},\ avec \left| \sum_{j \in J} disp(i, j) - \sum_{j \in J} disp(\tilde{i}, j) \right| \leq S \tag{5}$$

6. Pour un employé donné $i \in E$, si le nombre de jour où il est disponible durant la semaine est inférieur à une valeur ϕ, alors il doit travailler en chaque jour où il est disponible. Dans le cas contraire, il devra travailler au moins ϕ jours par semaine. Notons que ϕ correspond à la répartition uniforme de la demande hebdomadaire en nombre de travailleurs exigés par Amazon sur le nombre total d'employés. ϕ est calculé par la formule :

$$\phi = \sum_{j \in J} ne(j)/m \tag{6}$$

Formellement, ces deux contraintes s'expriment par :

$$\forall i \in E, \left(\sum_{j \in J} disp(i,j) \leq \phi \right) \implies \left(\sum_{j \in J} x_{ij} = \sum_{j \in J} disp(i,j) \right). \tag{7}$$

$$\forall i \in E, \left(\sum_{j \in J} disp(i,j) > \phi \right) \implies \left(\sum_{j \in J} x_{ij} \geq \phi \right). \tag{8}$$

3 Résolution par un solveur CSP

Pour la résolution de ce problème, j'ai choisi d'utiliser le solveur ACE (AbsCon Essence) qui a montré des résultats très intéressants dans la compétition de 2022. J'ai ainsi utilisé la librairie PyCSP3 de python pour modéliser le problème et pour faire appele au solveur. Dans l'implémentation des variable, j'ai décidé représenter les $m \times n$ variable par un tableau de taille $m \times n$.

Lors de l'essaye de ce modelé sur des données de différents cas réel, Les solutions proposer était très satisfaisante. De plus, la partie ou ont a dit qu'il est préférable qu'un employé soit affecté sur des jours consécutifs, le system de recherche du solver ACE permet d'avoir la plus part du temps de bon résultats de ce côté. Cependant, dans la situation ou le nombre d'employé disponible le jour j est inférieur au nombre d'employé demander par Amazon dans le même jour alors Il n'existe pas de solution. Pour remédier a cela, j'ai choisi d'utiliser une méthode de résolution incrémental où dans ce genre de cas, la contrainte (1) va être supprimer et remplacer par celle-ci :

$$\sum_{i \in E} x_{ij} \leq \sum_{i \in E} disp(i,j), \quad \forall j \in J \tag{9}$$

Ce qui permet d'avoir toujours une solution acceptable qui ne touche pas au contrainte très sensible comme la disponibilité des employés.