université de BORDEAUX

Collège Sciences et Technologies
UF Mathématiques et Interactions

Technicians and Interventions Scheduling for Telecommunications

Martin Deboute
Gonzague Beutter
Paul Brebinaud

CMI Optim L3

2020 – 2021

Challenge Algorithmique

Table des matières

1 Introduction **4**
1.1 Description et formalisation du problème 4

2 Modèle mathématique **5**
2.1 Modélisation du problème 5
2.2 PLNE 6

3 Algorithmes de résolution **7**
3.1 Heuristique 7
3.1.1 Arborescence des classes (résumé très succinct) : 7
3.1.2 Algorithme : 8
3.1.3 Description de la méthode *Chance* : 8

4 Expérimentations et résultats **9**

5 Conclusion **9**

Engagement de non plagiat

Nous, Martin Debouté, Gonzague Beutter, Paul Brebinaud, déclarons être pleinement conscients que le plagiat de documents ou d'une partie d'un document publiés sur toutes formes de support, y compris l'internet, constitue une violation des droits d'auteur ainsi qu'une fraude caractérisée.

En conséquence, nous nous engageons à citer toutes les sources que nous avons utilisées pour produire et écrire ce rapport.

Fait à Talence le 19 juin 2021

Signatures

Martin Debouté
Gonzague Beutter
Paul Brebinaud

Résumé

Cet article propose une démarche de résolution d'un problème de recherche opérationnelle sur la planification d'interventions de techniciens sur des infrastructures de télécommunications. Ce problème est tiré du challenge ROADEF 2007 organisé par l'entreprise France Télécom (aujourd'hui devenu Orange). Ce travail est tutoré par notre professeur M. Aurélien Froger maître de conférence à l'université de Bordeaux.

1 Introduction

Cette section permet d'introduire le sujet et de formaliser le problème.

1.1 Description et formalisation du problème

Le Technicians and Interventions Scheduling for Telecommunications Problem abrégé *TISTP* est défini dans P.-F. Dutot (2006). On nous donne un ensemble d'interventions $\mathcal{I} = \{1, ..., n\}$ et un ensemble de techniciens $\mathcal{T} = \{1, ..., m\}$. Chaque technicien maîtrise plusieurs domaines de compétence. Il existe $q \in \mathcal{D}$ domaines de compétences et le niveau de compétence d'un technicien dans un domaine donné est décrit par un entier de 0 à $p \in \mathcal{L}$, où 0 signifie que le technicien n'a aucune compétence dans le domaine associé. Nous pouvons exprimer les compétences de chaque technicien par un vecteur de compétences $s(t, d)$, q-dimensionnel, qui indique le niveau de compétence du technicien t dans le n-ième domaine de compétences d.

Les interventions varient en difficulté et certaines nécessitent plus d'un technicien. Le nombre de techniciens requis et la difficulté d'une intervention i sont décrits par une matrice d'exigences de compétences de dimension $p \times q$. Les colonnes de la matrice correspondent aux domaines de compétence et les lignes correspondent aux niveaux de compétence. Ainsi chaque entrée de la matrice $r(i, d, l)$ indique le nombre de techniciens avec un niveau d'au moins l dans le domaine d qui sont nécessaires pour effectuer l'intervention i. De même, nous représentons les compétences du technicien t sous forme d'une matrice $v(t, d, l)$. Notez que l'exigence d'un niveau de compétence élevé se répercute sur les niveaux de compétence inférieurs.

Les techniciens sont regroupés en équipes afin d'effectuer les interventions et une équipe peut être surqualifiée pour l'intervention i. Une équipe doit rester ensemble un jour donné, mais peut être dissoute le lendemain. Une intervention doit toujours être achevée en une seule journée et l'exécution d'une intervention ne peut être ni interrompue ni fractionnée sur plusieurs jours. Les techniciens peuvent être indisponibles certains jours en raison de vacances ou d'une formation. Nous désignerons l'ensemble des techniciens disponibles le jour k par l'ensemble $\mathcal{T}_k$, aussi $\mathcal{T}$ est l'ensemble des techniciens, $\mathcal{K}$ l'ensemble des jours et $\widehat{\mathcal{K}}_t$ l'ensemble des jours durant lequel le technicien $t \in \mathcal{T}$ ne travaille pas pour diverses raisons : formation, vacances, maladie, ... ($\widehat{\mathcal{K}}_t \subset \mathcal{K}$).

Outre la matrice des besoins en compétences, chaque intervention $i \in \mathcal{I}$ est caractérisée par une durée p_i, $i \in \mathcal{I}$, un coût de sous-traitance c_i de l'intervention $i \in \mathcal{I}$, un ensemble $Pred(i)$ d'interventions prédécesseurs. $Pred(i)$ est la liste des interventions devant être terminées avant de commencer l'intervention $i \in \mathcal{I}$ ($Pred(i) \subset \mathcal{I}$). La durée p_i indique combien de temps il faut pour exécuter l'intervention $i \in \mathcal{I}$. Cette durée est constante et reste la même quel que soit le nombre de techniciens affectés à l'intervention.

La priorité w_p d'une intervention $i \in \mathcal{I}$ ($w_p \in \{1, 2, 3\}$) est utilisée dans la fonction objectif du problème. Plus précisément, nous minimisons la fonction $w_1 C_1 + w_2 C_2 + w_3 C_3 + C$ où C_1, C_2 et C_3 représentent la date de fin la plus tardive des interventions de priorité 1, 2 et 3 (respectivement). C représente la date de fin de la plus tardive des interventions (indépendamment de sa priorité). Les valeurs des w_p sont définis dans P.-F. Dutot (2006) comme $w_1 = 28$, $w_2 = 14$, $w_3 = 4$, $w_4 = 1$. Un ensemble d'interventions $\mathcal{A}$ peut être sous traité si $\sum_{i \in \mathcal{A}} c_i \leqslant B$ et $Pred(i) \notin \mathcal{A}, \forall i \in \mathcal{A}$, où B est le budget maximal autorisé pour la sous-traitance. De plus, la sous-traitance des interventions n'est pas pénalisée dans l'objectif. Si une intervention i est sous- traitée, alors tous les interventions succédant i doivent être sous-traitées.

2 Modèle mathématique

2.1 Modélisation du problème

Cette section présente un modèle mathématique pour définir précisément le problème. Nous introduisons des notations supplémentaires. Sous consigne de notre professeur nous avons modélisé ce problème sous forme de programme linéaire en nombres entiers.

Soient $\mathcal{I}_p$ l'ensemble des interventions de priorité p, $\mathcal{I}_Pred$ l'ensemble des interventions avec prédécesseurs. Le sujet ne nous donne pas de limite supérieure imposée sur le nombre de jours utilisés pour planifier toutes les tâches, mais en utilisant une limite supérieure u sur la valeur de la fonction objectif, on peut en déduire une limite sur le nombre de jours : puisqu'une solution optimale peut utiliser au plus $\lceil \frac{u}{120} \rceil$ jours (on le déduit du sujet puisqu'il est précisé que le temps est découpé en unités de 120). On définit $\mathcal{K} := \left\{1, ..., \lceil \frac{u}{120} \rceil \right\}$. Dans ce qui suit M désigne un grand nombre et on fixe $M := 120|K|$, cela est suffisant.

Notre modèle utilise des variables binaires, $x_{tkr} \in \{0, 1\}$ qui est égal à 1 si et seulement si le technicien t est affecté à l'équipe r au jour k, et $y_{ikr} \in \{0, 1\}$ égal à 1 si et seulement si l'intervention i est affectée à l'équipe r le jour k. Nous utilisons également des variables continues $d_i \geqslant 0$ pour l'heure de début de l'intervention i et $f_p \geqslant 0$ pour l'heure de fin de la dernière intervention de priorité p pour $p \in \{1, 2, 3\}$ et la dernière intervention globale pour $p = 4$. La variable binaire $z_i \in \{0, 1\}$ prend la valeur 1 si et seulement si l'intervention i est sous-traitée. Enfin, la variable binaire $h_{ii'} \in \{0, 1\}$, $i, i' \in \mathcal{I}$, $i \neq i'$ prend la valeur 1 si et seulement si l'intervention i se termine avant le début de l'intervention i'. Celle-ci est utilisée pour garantir que deux tâches exécutées par la même équipe le même jour ne se chevauchent pas.

En utilisant cette notation, le *TISTP* peut être modélisé comme ceci (*c.f* page suivante) :

2.2 PLNE

$$\min \sum_{p=1}^{4} w_p f_p \quad (1)$$

$$sc\ f_p \geqslant d_i + p_i\ \forall p \in \{1,2,3\},\ \forall i \in \mathcal{I}_p \quad (2)$$

$$f_4 \geqslant d_i + p_i\ \forall i \in \mathcal{I} \quad (3)$$

$$\sum_{i \in \mathcal{I}} c_i z_i \leqslant B \quad (4)$$

$$|Pred(i)| z_i \leqslant \sum_{i' \in Pred(i)} z_{i'}\ \forall i \in \mathcal{I}_P red \quad (5)$$

$$\sum_{r=1}^{m} x_{tkr} \leqslant 1\ \forall k \in \mathcal{K},\ \forall t \in \mathcal{T}_k \quad (6)$$

$$\sum_{r=1}^{m} x_{tkr} = 0\ \forall k \in \mathcal{K},\ \forall t \in \mathcal{T} \backslash \mathcal{T}_k \quad (7)$$

$$z_i + \sum_{k \in \mathcal{K}} \sum_{r=1}^{m} y_{ikr} = 1\ \forall i \in \mathcal{I} \quad (8)$$

$$y_{ikr} r(i,d,l) \leqslant \sum_{t \in \mathcal{T}_k} v(t,d,l) x_{tkr}\ \forall i \in \mathcal{I},\ \forall k \in \mathcal{K},\ \forall r \in \{1,...,m\},\ \forall d \in \{1,...,q\},\ \forall l \in \{0,...,p\} \quad (9)$$

$$d_i + p_i \leqslant d_{i'} + M z_i\ \forall i \in \mathcal{I}_P red,\ \forall i' \in Pred(i) \quad (10)$$

$$120(k-1) \sum_{r=1}^{m} y_{ikr} \leqslant d_i\ \forall i \in \mathcal{I},\ \forall k \in \mathcal{K} \quad (11)$$

$$120k \sum_{r=1}^{m} y_{ikr} + M(1 - \sum_{r=1}^{m} y_{ikr}) \geqslant d_i + p_i\ \forall i \in \mathcal{I},\ \forall k \in \mathcal{K} \quad (12)$$

$$d_i + p_i - (1 - h_{ii'})M \leqslant d_{i'}\ \forall i, i' \in \mathcal{I},\ i \neq i' \quad (13)$$

$$y_{ikr} + y_{i'kr} - h_{ii'} - h_{i'i} \leqslant 1\ \forall i, i' \in \mathcal{I},\ i \neq i',\ \forall k \in \mathcal{K},\ \forall r \in \{1,...,m\} \quad (14)$$

$$x_{tkr} \in \{0,1\}\ \ t \in \mathcal{T},\ \forall k \in \mathcal{K},\ \forall r \in \{1,...,m\} \quad (15)$$

$$y_{ikr} \in \{0,1\}\ \ i \in \mathcal{I},\ \forall k \in \mathcal{K},\ \forall r \in \{1,...,m\} \quad (16)$$

$$z_i \in \{0,1\}\ \ i \in \mathcal{I} \quad (17)$$

$$h_{ii'} \in \{0,1\}\ \ i, i' \in \mathcal{I},\ i \neq i' \quad (18)$$

$$f_p \geqslant 0\ \forall p \in \{1,2,3,4\} \quad (19)$$

$$d_i \geqslant 0\ i \in \mathcal{I} \quad (20)$$

L'objectif (1) consiste à minimiser une fonction qui dépend des dates de fin des interventions. Les inégalités (2) et (3) définissent les temps de fin utilisés dans l'objectif, et l'inégalité (4) impose le budget de sous-traitance. L'inégalité (5) garantie que si une tâche est sous-traitée, toutes les tâches qui lui succèdent sont aussi sous-traitées. L'inégalité (6) garantie qu'un technicien est utilisé au maximum dans une équipe par jour, et l'inégalité (7) fait que des techniciens indisponibles ne sont pas utilisés. L'inégalité (8) garantie que chaque tâche est exécutée ou sous-traitée, tandis que l'inégalité (9) indique que chaque tâche est exécutée par une équipe possédant les compétences appropriées (si elle n'est pas sous-traitée), et l'inégalité (10) imposent les contraintes de précédence. Les inégalités (11) et (12) fixent des limites inférieures et supérieures sur l'heure de début de chaque tâche en fonction du jour où la tâche est exécutée. L'inégalité (13) définie correctement les variables $h_{ii'}$ par rapport aux heures de début des interventions i et i', tandis que la contrainte (14) garantie que les interventions i et i' ne se chevauchent pas si elles sont exécutées par la même équipe le même jour, en effet : si deux interventions distinctes i et i' sont exécutées par la même équipe r le même jour k, alors $y_{ikr} + y_{i'kr} = 2$ et soit $h_{ii'}$ soit $h_{i'i}$ doit être égal à 1 pour que l'inégalité soit satisfaite. Les contraintes (15) - (20) définissent les domaines des variables de décision.

Pour l'inégalité (5) et l'inégalité (10) nous nous sommes inspiré de la formalisation de J.-F. Cordeau (2008) dans la paragraphe 2 de la section 2, page 4.

3 Algorithmes de résolution

Cette section décrit une heuristique de construction pour le *TISTP*. Cette heuristique peut rapidement fournir une solution réalisable au problème. Le *TISTP* est divisé en 3 sous-problèmes : le choix des interventions à sous-traiter, la constitution des équipes et l'attribution des interventions aux équipes.

Remarque : une solution triviale serait d'affecter une intervention par jour à chaque nouvelle équipe.

3.1 Heuristique

3.1.1 Arborescence des classes (résumé très succinct) :

Une classe *Team* contenant son numéro, une liste d'intervention que l'équipe effectue, une liste de techniciens affectés à l'équipe, un tableau des niveaux dans chaque domaine de l'équipe, et un *boolean* indiquant si l'équipe est pleine (c'est à dire qu'aucune autre intervention ne peut être ajoutée).

Une classe *Day* contenant, le numéro du jour, la liste des techniciens disponibles, la liste des techniciens non disponibles, et la liste des équipes du jour.

Une classe *Solution* qui contient une liste de jour, la liste des interventions faites, la liste des interventions sous-traitées, et le coût total de sous-traitance.

Une classe *Tech* contenant différentes fonctions pour comparer les niveaux dans chaque domaines nécessaires pour une intervention avec ceux d'une équipe, d'un technicien, ou encore d'une équipe plus un technicien, etc...

Une classe *buildTTSPSol* qui permet de créer une instance *TTSPSolution* avec une instance *Solution*.

Une classe *MethodNotUsed* qui regroupe des méthodes non utilisées dans l'algorithme finale puisque moins efficace, mais implémentées et fonctionnelles.

Une *Javadoc* a été crée pour toute les méthodes utilisées pour la réalisation de l'heuristique.

3.1.2 Algorithme :

1. On créer un objet *Solution* (classe *Solution*)

2. On ajoute un jour, jour d=1

3. Pour chaque priorité n, on appelle la méthode *loop* : la méthode *loop* appelle une méthode de tri *buildArray*. *BuildArray* retourne 2 listes d'interventions : une première comprenant dans l'ordre les interventions de priorités n sans prédécesseurs et toutes les interventions devant être effectuées avant les interventions de priorités n. La 2ème liste contient les interventions de priorités n avec prédécesseurs

4. Pour chacune des 2 listes composées d'interventions i, on fait :

   (a) On regarde si i peut être sous-traitée (et donc si tous les interventions i_2 appartenant à $Pred(i)$ peuvent l'être également, si oui, on sous-traite i et tout i_2 appartenant à $Pred(i)$. Toutes les interventions sous-traitées sont retirées de la liste des interventions à assigner

   (b) Tant que la liste d'interventions à traiter n'est pas vide ou que on ne l'a pas réalisée n fois la boucle :
      i. Si on à fait m fois la boucle et si la liste n'est pas vide : on ajoute un jour : $d++$
      ii. On fait k fois les étapes suivantes :
         A. Si l'instance que l'on cherche à résoudre à plus de 10 interventions :
         - On appelle la méthode *completeTeamDday* pour chaque équipe de chaque jour, qui appelle la méthode *Chance* (décrite ci-dessous)
         - On appelle une autre méthode de tri sur la liste des interventions. Elle retourne la liste des interventions triées selon le temps d'exécution de chaque intervention (dans un ordre décroissant)
         B. On appelle la fonction *Chance* (décrite ci-dessous) sur le dernier jour.

5. On créer une *TTSPSolution* avec la *Solution* construite à l'aide de la classe *buildTTSPSol*

3.1.3 Description de la méthode *Chance* :

6. On créer une nouvelle équipe t lors du jour d
7. On ajoute tous les techniciens *tech* disponible le jour d à cette équipe, et on les retire de la liste des techniciens disponibles le jour d
8. On parcourt la liste des interventions i :
   (a) Tant que l'équipe t n'est pas pleine : si l'équipe est suffisante pour faire l'intervention i, c'est à dire qu'elle a des compétences suffisantes dans tous les domaines (ce que l'on vérifie grâce aux méthodes de la classe *Tech*) : on ajoute l'intervention i à l'équipe t
9. On parcourt la liste des techniciens *tech* de l'équipe t :
   (a) Si les compétences de l'équipe t moins le technicien *tech* sont suffisantes pour effectuer toutes les interventions i assignées à l'équipe t : on retire le technicien *tech* de l'équipe t, et on l'ajoute à la liste des techniciens disponibles le jour d
10. On retourne la nouvelle liste des interventions non traitées

4 Expérimentations et résultats

Solutions

	Optimale	Notre heuristique	Distance	Temps	Optimale	Notre heuristique
Instance	Dataset A	Dataset A	En pourcentage %	En seconde	Dataset B	Dataset B
1	2340	3210	37	<1	33900	59640
2	4755	5940	25	<1	15870	31200
3	11880	22620	90	<1	16005	30960
4	13452	18000	34	<1	23775	68055
5	28845	43800	52	<1	88680	152160
6	18795	33240	77	<1	26955	58230
7	30540	44640	46	1	31620	49560
8	16920	31080	84	1	33030	49560
9	27348	40935	50	1	28080	42480
10	38296	52860	38	<1	34680	53760
		Moyenne :	52			Moyenne :
		Moyenne générale :	69			

5 Conclusion

Conclusion sur notre heuristique : notre heuristique n'est pas des plus avancées, nous sommes en moyenne à 69% de l'optimal, mais elle a le mérite de fonctionner pour toutes les instances proposées, même les plus grosses, dans un temps relativement court. Le problème posé étant un problème NP-difficile, il est inconcevable de créer des solutions avec une résolution exacte dans des temps raisonnables pour des instances concrètes (de taille élevées).

Références

J.-F. Cordeau G. Laporte, F. Pasin S. Ropke (avr. 2008). « Scheduling Technicians and Tasks in a Telecommunications Company ». In : *citeseerx*.

P.-F. Dutot A. Laugier, A.-M. Bustos (2006). *Technicians and Interventions Scheduling for Telecommunications*. France Telecom RD.