

Ordonnancement de rendez-vous de type speed-dating

Le Roux, Agnès
Mél : agnes.le-roux@mines-nantes.fr

Résumé : Dans cet article, nous présentons des travaux consistant à ordonnancer les rendez-vous d'une soirée de speed-dating. À cette occasion, des hommes et des femmes se rencontrent mutuellement lors de créneaux courts de durée unitaire (classiquement 7 min). L'organisation de la soirée est rendue plus difficile du fait de participants arrivant en retard et/ou de rencontres interdites entre participants se connaissant déjà. L'objectif est de positionner les rencontres de manière à limiter les attentes des participants. Du point de vue théorique, nous avons démontré la difficulté de ce problème. Il n'existe a priori pas de méthode de résolution s'exécutant en temps polynomial. Nous proposons ensuite un modèle linéaire en nombres entiers pour ce problème que nous testons sur un jeu de 300 instances générées aléatoirement. Enfin, les résultats de ces tests sont présentés.

Mots clés : *Optimisation, ordonnancement, complexité, programmation linéaire en nombres entiers, speed-dating.*

1 Introduction et définition du problème

Nous nous intéressons au problème d'ordonnancement de rencontres au cours de soirées de speed-dating. Il s'agit d'un mode de rencontres rapides amicales ou amoureuses entre une population d'hommes et de femmes. Idéalement, lors de ces soirées, sept hommes et sept femmes se rencontrent sur des créneaux horaires de sept minutes. Toutes les rencontres qui s'effectuent sur le même créneau commencent et finissent au même moment. À la fin de chaque rencontre, les hommes se déplacent de manière à rencontrer une nouvelle personne, alors que les femmes restent assises. L'organisation de soirées de speed-dating peut paraître anodine. En effet, si autant d'hommes que de femmes participent à une soirée et que tous sont à l'heure, la planification des rencontres est facile à organiser : il suffit d'effectuer une permutation circulaire des hommes à la fin de chaque créneau. Dans la pratique, il arrive souvent que les deux populations soient de tailles différentes ou que certaines rencontres soient interdites car les participants se sont déjà rencontrés lors d'une soirée précédente. De plus, certains participants arrivent en retard. Tous ces événements perturbateurs engendrent des attentes pour les participants ne pouvant ainsi pas enchaîner toutes leurs rencontres. Nous souhaitons minimiser le plus grand nombre de créneaux d'attente des participants. Nous nous intéressons à la version statique du problème dans laquelle toutes les données (retards des participants, rencontres ayant déjà eu lieu dans des soirées précédentes) sont connues à l'avance.

Exemple 1.1. À titre d'exemple, nous proposons de considérer une soirée à laquelle participent trois femmes F_1 , F_2 et F_3 et trois hommes M_1 , M_2 et M_3 . Le tableau 1 représente un ordonnancement possible de cette soirée. La femme F_2 arrive à la date 1, la femme F_3 arrive à la date 2 et l'homme M_3 arrive à la date 1. La rencontre entre F_2 et M_1 est interdite, c'est pour cela qu'elle n'apparaît pas dans la liste de rencontres à effectuer par ces deux participants. Une case grisée indique que le (la) participant(e) n'est pas encore arrivé(e) ou qu'il (elle) est parti(e) après avoir effectué l'ensemble de ses rencontres. Le nombre maximum de créneaux d'attente ici est deux, c'est le nombre de créneaux d'attente des hommes M_2 et M_3 . Cette solution n'est pas optimale car on peut trouver un autre ordonnancement avec un nombre maximum d'attente de 1. Le lecteur est invité à trouver un tel ordonnancement.

Dans la partie 2, nous présentons différents problèmes de la littérature se rapprochant du problème de speed-dating. Dans la partie 3, nous traitons de la difficulté de ce problème du point de vue théorique. La partie 4 est dédiée à la modélisation que nous avons faite du problème. Dans la partie 5, nous présentons les résultats expérimentaux que nous avons obtenu avant de conclure et présenter nos perspectives de travail.

Participants	Rencontres	0	1	2	3	4	5	Nb attentes
F_1	$\{M_1, M_2, M_3\}$	M_1	M_2	Att.	M_3			1
F_2	$\{M_2, M_3\}$		Att.	M_3	M_2			1
F_3	$\{M_1, M_2, M_3\}$			M_1	Att.	M_2	M_3	1
M_1	$\{F_1, F_3\}$	F_1	Att.	F_3				1
M_2	$\{F_1, F_2, F_3\}$	Att.	F_1	Att.	F_2	F_3		2
M_3	$\{F_1, F_2, F_3\}$		Att.	F_2	F_1	Att.	F_3	2

TABLE 1 – Ordonnancement des rencontres d'un événement de speed-dating

2 État de l'art

Le problème de speed-dating est un problème récent. À notre connaissance, les seuls travaux connus sont ceux de Lapègue et al. [1] qui proposent un modèle de programmation par contraintes pour résoudre le cas dynamique. Cependant, notre problème se rapproche de problèmes classiques de la littérature.

Le cas particulier de speed-dating dans lequel aucune rencontre n'est interdite et une population arrive à l'heure se rapproche du problème d'open shop à temps d'exécution unitaire dont l'objectif est de minimiser le plus grand retard algébrique des tâches [2]. Les femmes, les hommes et les rencontres représentent respectivement les machines, les tâches et les opérations. Les résultats énoncés dans [2] permettent de déterminer une méthode de résolution "rapide" pour le sous-problème de speed-dating dans lequel les femmes sont à l'heure et aucune rencontre n'est interdite.

Le problème de speed-dating est également similaire au problème d'emploi du temps (TT) défini dans [3], dans lequel les hommes, les femmes et les rencontres représentent respectivement les classes, les professeurs et les cours à planifier. Les retards des participants représentent les indisponibilités des classes ou des professeurs. Cependant, dans notre cas, les indisponibilités sont regroupées en début du planning. De plus l'objectif étudié dans l'article est de minimiser la date finale de l'ensemble des cours.

On peut également considérer le problème de speed-dating comme un problème d'ordonnancement par fournées avec incompatibilités entre les tâches et capacité infinie des fournées [4]. Les rencontres représentent les tâches et deux tâches sont compatibles si les deux rencontres peuvent avoir lieu en même temps. Autrement dit, deux rencontres ne sont pas compatibles si elles partagent un même homme ou une même femme. Cependant, l'objectif généralement considéré pour les problèmes d'ordonnancement par fournées est de minimiser la date de fin de l'ensemble des tâches, de plus les tâches n'ont pas de retard et peuvent être planifiées dès le début de l'ordonnancement. À notre connaissance, il n'existe pas de travaux dans ce domaine étudiant notre objectif.

3 Complexité du problème

Lorsqu'on étudie un nouveau problème d'optimisation, on souhaite connaître son degré de difficulté par rapport aux problèmes connus de la littérature. La théorie de la complexité définit différentes classes de problèmes selon leur difficulté.

On dit en général qu'un problème est "facile" ou polynomial si on connaît une méthode de résolution (algorithme) dont le nombre d'étapes est borné par un polynôme en la taille de l'instance. Nous avons démontré que le problème de speed-dating est polynomial si aucune rencontre n'est interdite et si toutes les femmes sont à l'heure. La taille de notre problème est définie par le nombre d'hommes (noté n) et le nombre de femmes (noté p) et nous avons trouvé un algorithme qui résout ce problème en un nombre d'étapes égal à $k \times n^{5/2} p^{3/2} \log(np)$ où k est une constante.

Il existe une autre classe de complexité classique dans la littérature notée \mathcal{NP} . Il s'agit de l'ensemble des problèmes se vérifiant en temps polynomial. C'est à dire que si on donne une solution de ce problème, on peut vérifier en temps polynomial que celle-ci est valide.

La classe \mathcal{NP} -difficile est la classe des problèmes plus "difficiles" que tous les autres problèmes \mathcal{NP} . En

particulier, on ne connaît pas d'algorithme qui puisse les résoudre en temps polynomial¹. Pour démontrer qu'un problème est \mathcal{NP} -difficile, on démontre qu'un problème \mathcal{NP} -difficile classique de la littérature est un "cas particulier" de celui étudié. Nous avons démontré que le problème de speed-dating est \mathcal{NP} -difficile dans le cas où les participants peuvent être en retard [5].

4 Modélisation du problème

Nous proposons un programme linéaire en nombres entiers (PLNE) permettant de modéliser notre problème. Pour cela, nous définissons les variables suivantes :

- $x_{ij}^k = \begin{cases} 1 & \text{si la rencontre entre la femme } F_i \text{ et l'homme } M_j \text{ a lieu à la date } k, \\ 0 & \text{sinon.} \end{cases}$

- W_{\max} est le plus grand nombre de créneaux d'attente des participants.

On souhaite minimiser W_{\max} . On modélise l'ensemble de nos contraintes de manière linéaire à l'aide des variables x_{ij}^k et W_{\max} . On note respectivement les ensembles de femmes, hommes et créneaux \mathcal{F} , \mathcal{M} et \mathcal{S} . Nous listons maintenant l'ensemble des contraintes de notre problème. Toutes ces contraintes s'écrivent de manière linéaire à l'aide des variables définies. Par souci de clarté, nous ne donnons pas toutes les formules mathématiques.

- Aucune rencontre interdite ne peut avoir lieu et toutes les rencontres obligatoires ont lieu exactement une fois dans la soirée.

$$\forall i \in \mathcal{F}, \forall j \in \mathcal{M}, \quad \sum_{k \geq r_{ij}} x_{ij}^k = O_{ij}$$

où $O_{ij} = 1$ si et seulement si la femme F_i et l'homme M_j doivent se rencontrer et r_{ij} est la première date à laquelle les deux participants sont disponibles.

- Un participant ne peut faire qu'une rencontre pendant un créneau.

$$\forall i \in \mathcal{F}, \forall k \in \mathcal{S}, \quad \sum_{j \in \mathcal{M}} x_{ij}^k \leq 1.$$

$$\forall j \in \mathcal{M}, \forall k \in \mathcal{S}, \quad \sum_{i \in \mathcal{F}} x_{ij}^k \leq 1.$$

- Une rencontre ne peut pas avoir lieu tant que les deux participants ne sont pas arrivés.

- L'ordonnancement des rencontres induit un nombre de créneaux d'attente pour chaque participant. Une contrainte liant les variables x_{ij}^k et la variable W_{\max} permet de déterminer le plus grand nombre de créneaux d'attente des participants.

En général, la résolution d'un PLNE est \mathcal{NP} -difficile. On cherche donc à améliorer ce modèle.

Une première manière de procéder consiste à fournir au modèle des bornes pour le nombre maximum de créneaux d'attentes des participants. Pour une instance donnée, on calcule facilement une borne supérieure en déterminant un planning réalisable pour la soirée. Pour cela, on affecte créneau par créneau les rencontres entre les participants disponibles et en priorité à ceux qui ont le moins de rencontres déjà affectées. La solution obtenue n'est pas nécessairement optimale mais fournit une borne supérieure pour W_{\max} . Nous avons développé également une technique permettant déterminer une borne inférieure de W_{\max} . Par exemple, dans l'exemple 1.1, on remarque que pendant le premier créneau, il ne peut y avoir qu'une rencontre car une seule femme est arrivée. L'un des hommes M_1 ou M_2 aura nécessairement au moins un créneau d'attente dans son planning. Une borne inférieure ici est donc 1.

On peut également améliorer le modèle en ajoutant des contraintes (appelées inégalités valides). Dans l'exemple 1.1, on remarque notamment que la femme F_2 doit effectuer sa dernière rencontre pendant le créneau 2 pour n'avoir aucune attente. De même, l'homme M_2 doit également effectuer sa dernière rencontre au créneau 2. On en déduit que si la rencontre entre ces deux participants à lieu à la date $2+k$ (pour $k \in \mathbb{N}$), alors ces deux participants auront k créneaux d'attente et $W_{\max} \geq k$. On peut écrire cette contrainte de manière linéaire :

$$\forall k \in \mathbb{N}, \quad k \times x_{2,2}^{2+k} \leq W_{\max}. \quad (1)$$

On peut ajouter de la même manière une inégalité valide pour chaque rencontre non interdite.

1. Trouver un algorithme polynomial résolvant un problème \mathcal{NP} -difficile est un des problèmes du millénaire proposés par le Clay Mathematics Institute : claymath.org. Si on trouve un algorithme polynomial résolvant un problème \mathcal{NP} -difficile, alors il existe un algorithme polynomial résolvant tous les problèmes \mathcal{NP} .

5 Tests numériques

Afin de tester notre modèle, nous avons généré aléatoirement 300 instances de 14 à 70 participants à partir de paramètres fournis par un organisateur de soirées de speed-dating. Une instance est caractérisée par son nombre d’hommes et de femmes, les dates d’arrivée de chacun des participants et les rencontres interdites. Pour tester l’efficacité de ce modèle, nous avons utilisé le solveur de PLNE : Cplex 12.4 qui est un solveur générique utilisant une méthode (classique) de branch-&-cut. Un temps maximum de résolution est fixé à 30 min pour chaque instance.

On obtient les résultats présentés dans le tableau 2. La première configuration (\emptyset) présente les résultats obtenus avec le modèle sans contraintes additionnelles. On obtient de meilleurs résultats en fournissant au solveur une borne inférieure et une borne supérieure de l’objectif W_{\max} (configuration LUB). Enfin, le solveur réussit à trouver 296 solutions optimales sur 300 par l’ajout des bornes et d’inégalités valides (configuration LUB+IV).

Taille	\emptyset		LUB		LUB+IV	
14	60/60	< 0,1	60/60	< 0,1	60/60	< 0,1
24	60/60	1,3	60/60	0,3	60/60	0,3
40	60/60	171,6	60/60	4,4	60/60	4,5
50	53/60	>1240	59/60	>80	60/60	23,8
70	0/60		28/60	>1200	56/60	>510

TABLE 2 – Proportion d’instances résolues en moins de 30 min et temps moyens de résolution (en s).

6 Conclusion et perspectives

Dans ces travaux de thèse, nous avons proposé la définition d’un nouveau problème d’ordonnancement qui s’intéresse à l’organisation de soirées de speed-dating dans le but de minimiser les attentes des participants. Nous avons démontré que ce problème est \mathcal{NP} -difficile ce qui rend son étude particulièrement intéressante. Nous avons proposé un modèle linéaire amélioré à l’aide de bonnes bornes et d’inégalités valides efficaces. Le modèle montre cependant ses limites pour des instances de grande taille.

Afin d’améliorer les résultats obtenus sur nos instances, nous travaillons sur une méthode de génération de colonnes. Par la suite, nous envisageons d’élargir ce problème en ajoutant des contraintes de déplacements des participants et de borner les attentes consécutives. D’autre part, nous prévoyons d’élargir notre étude à des problèmes connexes tels que le job-dating ou les bourses d’échange de technologie.

Références

- [1] T. Lapègue, O. Bellenguez-Morineau, and C. Guéret. Scheduling of a speed-dating event. In *OR 2011*, p.99–100, 2011.
- [2] H. Kellerer, T. Tautenhahn, and G. Woeginger. Note : Open-shop scheduling with release dates to minimize maximum lateness. *Naval Research Logistics (NRL)*, 42(1) :141–145, 1995.
- [3] S. Even, A. Itai, and A. Shamir. On the complexity of timetable and multicommodity flow problems. *SIAM Journal on Computing*, 5(4) :691–703, 1976.
- [4] M. Boudhar and G. Finke. Scheduling on a batch machine with job compatibilities. *Belgian Journal of Operations Research*, 40 :69–80, 2000.
- [5] A. Le Roux, O. Bellenguez-Morineau, C. Guéret and D. Prot. Complexity of speed-dating scheduling. (Soumis)