

# Travail d'Étude et de Recherche

## Aide à l'optimisation de rendez-vous de type « business speed-dating »

Alexandre Medi

Andreea Radulescu

Johan Voland

*Université de Nantes*

10 mai 2011

## Remerciements

En préambule de ce rapport, nous souhaitons remercier toutes les personnes qui nous ont aidés durant ce premier travail d'étude et de recherche.

Tout d'abord, nous souhaitons remercier Monsieur **Denis Poncelet**, professeur à l'ONIRIS et au GEPEA de Nantes, qui est à l'origine de ce sujet de recherche.

Nous remercions ensuite notre professeur encadrant, Monsieur **Xavier Gandibleux**, professeur d'informatique au LINA (Laboratoire d'Informatique de Nantes Atlantique), pour nous avoir proposé ce sujet et nous avoir aiguillé tout au long de ce travail de recherche.

Nous souhaitons aussi remercier Monsieur **Alexandre Goldsztejn**, professeur d'informatique au LINA, pour son temps passé en tant que membre du jury.

Enfin, nous remercions l'ensemble des enseignants du département informatique du LINA pour nous avoir transféré leur savoir durant cette première année de formation en recherche opérationnelle.

# Table des matières

<b>1</b>	<b>Présentation du problème</b>	<b>4</b>
1.1	Contexte . . . . .	4
1.2	Formalisation du processus décisionnel . . . . .	4
1.2.1	Décideur . . . . .	4
1.2.2	Processus d'aide à la décision . . . . .	4
1.2.3	Exemple . . . . .	5
1.3	Pertinence de l'étude . . . . .	6
<b>2</b>	<b>Analyse de l'existant</b>	<b>7</b>
2.1	Données accessibles . . . . .	7
2.1.1	Précédentes bourses . . . . .	7
2.1.2	Information du décideur . . . . .	7
2.1.3	Interaction du décideur . . . . .	7
2.2	Optimisation en place . . . . .	7
2.2.1	Données . . . . .	8
2.2.2	Variables de décision . . . . .	8
2.2.3	Fonction économique . . . . .	8
2.2.4	Contraintes . . . . .	8
2.3	Solution logicielle existante . . . . .	9
2.3.1	Méthode de résolution . . . . .	9
2.3.2	Stockage informatique . . . . .	9
2.3.3	Implémentation . . . . .	9
2.4	Critique . . . . .	9
<b>3</b>	<b>Positionnement scientifique</b>	<b>11</b>
3.1	Classe de problème d'optimisation . . . . .	11
3.2	Besoins exprimés et options existantes . . . . .	11
3.3	Littérature existante . . . . .	12
3.4	Littérature connexe . . . . .	14
3.5	Synthese . . . . .	15
<b>4</b>	<b>Contributions</b>	<b>17</b>
4.1	Simplification de l'utilisation . . . . .	17
4.1.1	Importation des données . . . . .	17
4.1.2	Définition des données supplémentaires . . . . .	17
4.1.3	Choix de la méthode de résolution . . . . .	18
4.1.4	Affichage du résultat . . . . .	19
4.2	Résolution exacte . . . . .	19
4.2.1	Version précédente . . . . .	19
4.2.2	Solveur utilisé . . . . .	19
4.2.3	Le modèle . . . . .	19
4.2.4	Résultats et analyse . . . . .	21
4.3	Résolution approchée avec Tabou . . . . .	24
4.3.1	Structure de données . . . . .	24
4.3.2	Le voisinage . . . . .	25
4.3.3	Résultats et analyse . . . . .	25
4.4	Comparaison CPLEX/Tabou . . . . .	27
<b>5</b>	<b>Conclusion et perspectives</b>	<b>29</b>
5.1	Conclusion . . . . .	29
5.2	Perspectives . . . . .	29

# 1 Présentation du problème

## 1.1 Contexte

Une bourse d'échanges de technologie est un événement qui réunit des experts durant plusieurs jours dans le but de favoriser les discussions entre des personnes souhaitant se rencontrer, et ainsi de réaliser des échanges de connaissances. Chaque participant dispose d'une liste contenant l'ensemble des personnes inscrites à la manifestation et exprime des vœux sur les individus qu'il souhaite rencontrer. Les réunions individuelles durent de 30 à 40 minutes et rassemblent deux participants autour d'une table.

Comme la participation à ce type d'événement est payante, l'organisation d'une bourse d'échanges a pour but de satisfaire au mieux les choix des participants et d'accueillir de nombreuses personnes.

Le but de ce projet est d'améliorer la planification des rendez-vous sur les plages horaires prévues durant l'événement. Le programme des réunions doit prendre en compte plusieurs facteurs : l'ensemble des vœux exprimés par les participants doit être satisfait au mieux, chaque participant doit avoir un nombre minimal de réunions assurées et planifiées de manière équilibrée durant les deux jours. De plus, comme tout changement de vœux de dernière minute doit être pris en compte au mieux, le programme doit être flexible et pouvoir intégrer facilement des modifications.

## 1.2 Formalisation du processus décisionnel

### 1.2.1 Décideur

Le problème a été posé par M. Denis Poncelet, organisateur des Bourses d'échanges de technologie. D'une façon plus générale, le logiciel est destiné aux organisateurs de Bourses d'échanges de type "Business Speed Dating".

### 1.2.2 Processus d'aide à la décision

La planification des rendez-vous dans le cadre d'une bourse d'échanges a pour objectif de satisfaire au mieux les souhaits des participants.

La qualité d'un planning est évaluée en fonction de plusieurs critères comme le nombre minimal de rendez-vous assurés pour chaque participant ou la façon dont ils sont répartis sur l'ensemble des deux jours. Ainsi, la solution rendue par le logiciel doit prendre en compte au mieux l'ensemble des vœux exprimés par les participants, elle doit garantir que chaque participant ait un nombre minimal de réunions et que la planification des rendez-vous soit faite de manière équilibrée durant les jours de l'événement.

De plus, pour limiter les déplacements entre les salles de deux réunions consécutives, le planning doit faciliter la rotation des personnes entre les tables. Afin d'éviter une surcharge de l'emploi du temps pour une personne, la solution doit privilégier le début d'une journée pour la planification des rendez-vous et inclure des pauses après deux réunions consécutives.

Les ressources à mobiliser (tables, salles) devant être utilisées au plus juste besoin, la solution rendue par le logiciel doit équilibrer le nombre de rendez-vous sur l'ensemble des périodes disponibles. Pour que tout changement de vœux de dernière minute soit pris en compte au mieux, le programme de la bourse d'échange doit être flexible et pouvoir intégrer facilement les modifications ultérieures.

Les précédentes bourses d'échanges ont regroupé entre 80 et 140 participants et se sont déroulées sur deux jours. L'organisation a été faite à l'aide du logiciel "Meet-me" développé par une équipe d'étudiants de l'Ecole des Mines de Nantes qui, à l'aide du solveur GLPK sur lequel il est basé, planifie des rendez-vous sur 10 à 12 plages horaires en tenant compte des préférences des participants.

### Spécifications techniques

Le décideur utilisant un ordinateur sous Mac OS X, il est donc important que le logiciel soit compatible avec ce système d'exploitation. De plus, il doit pouvoir s'exécuter sur un ordinateur ayant une configuration matérielle standard et rendre une solution dans un temps acceptable.

#### 1.2.3 Exemple

Dans l'exemple ci-dessous, nous avons représenté les préférences de 6 participants.

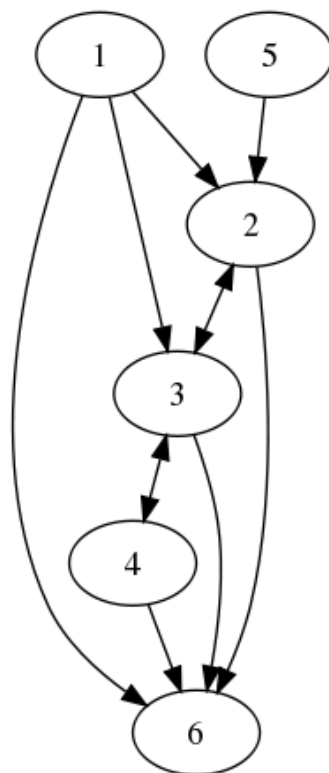


FIGURE 1 – Sens des préférences

Sur la figure 1, nous avons les préférences suivantes :

- 1 veut rencontrer 2, 3 et 6
- 2 veut rencontrer 3 et 6
- 3 veut rencontrer 2, 4 et 6
- 4 veut rencontrer 3 et 6
- 5 veut rencontrer 2
- 6 ne souhaite rencontrer personne

Chaque participant devant indiquer un ordre de préférence, celui-ci est valué en partant de 10 pour son premier souhait, 9 pour le suivant, etc.

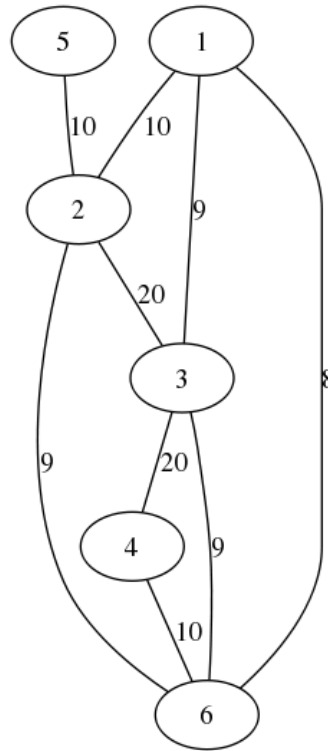


FIGURE 2 – Valeurs des préférences

Sur la figure 2, les poids correspondent aux sommes des préférences cumulées entre les participants.

### 1.3 Pertinence de l'étude

L'objectif principal lors de l'organisation d'une manifestation de type bourse d'échanges de technologie étant de favoriser les discussions entre participants souhaitant se rencontrer, l'organisateur a donc la charge de planifier au mieux les rendez-vous en tenant compte non seulement des préférences individuelles de chaque participant, mais aussi d'autres contraintes, telles que la durée de la manifestation, le nombre de salles ou encore le nombre de tables disponibles.

La planification des réunions a été réalisée manuellement avec un taux de satisfaction raisonnable, pour une manifestation regroupant 60 à 100 personnes, son élaboration mobilisant deux personnes sur une journée. Mais pour un nombre de participants en augmentation, l'élaboration manuelle de la solution devient longue et difficile. En outre, tout changement de vœux de dernière minute, inscription tardive ou absence d'un participant est difficilement pris en compte.

L'étude a donc pour but d'implémenter un modèle d'optimisation pour résoudre le problème de planification et de concevoir un logiciel d'aide à l'organisation d'une bourse d'échanges de technologie. Ce logiciel permettrait à un organisateur d'une telle manifestation de planifier en peu de temps un nombre maximum de rendez-vous souhaités par les participants et donc de satisfaire au mieux les préférences de ces derniers.

## 2 Analyse de l'existant

### 2.1 Données accessibles

#### 2.1.1 Précédentes bourses

Les précédentes bourses organisées ont rassemblé de 80 à 140 personnes (académiques et industriels) durant deux jours. Une réunion entre deux personnes se déroule autour d'une table pendant une période de 30 à 40 minutes, induisant un nombre de 10 à 12 plages horaires sur la durée de la manifestation.

Les participants à une bourse d'échanges s'inscrivent via le site web de l'événement. Lors de leur connexion au site, ils accèdent aux différents renseignements, comme la liste de tous les participants. A partir de cela, ils peuvent sélectionner les personnes qu'ils désirent rencontrer pendant la bourse d'échanges et donner une préférence à chaque rendez-vous souhaité.

Pour l'organisation des rendez-vous, le logiciel "Meet-me" planifie les réunions en tenant compte des préférences de chaque participant.

#### 2.1.2 Information du décideur

Lorsque les informations saisies par les participants ont été traitées, le solveur passe à la résolution du problème. Lorsque celui-ci a trouvé une solution optimale, le logiciel "Meet-me" affiche via son interface graphique les réunions planifiées. Ainsi, à la fin de la résolution, l'utilisateur est informé du planning trouvé par le logiciel et peut consulter les rendez-vous planifiés pour chaque période et pour chaque participant.

#### 2.1.3 Intéraction du décideur

Après l'affichage de la solution trouvée par le logiciel, le décideur dispose de l'ensemble de la liste des rencontres planifiées. Ces rencontres sont présentées sous forme de tableau de rendez-vous sur laquelle des modifications de dernière minute peuvent être effectuées.

Le décideur peut donc intégrer manuellement de nouveaux rendez-vous et améliorer le planning s'il le souhaite. Il peut par exemple compléter manuellement des préférences non satisfaites par le logiciel, supprimer des rendez-vous en cas d'absence d'un participant, ou encore déplacer l'horaire d'une rencontre.

### 2.2 Optimisation en place

Pour établir le programme des réunions, la résolution doit prendre en compte l'ensemble des voeux exprimés par les participants, que chaque personne ait au plus une réunion par plage horaire et un nombre minimal de réunions assurées, réparties de manière équilibrée sur les 2 jours de l'événement. Ainsi, le problème peut être considéré comme multi-objectif : maximiser les souhaits des participants, maximiser le nombre minimum de réunions suivies par une personne et équilibrer le nombre de réunions par personne et par période, ainsi que minimiser la distance à parcourir entre les tables de 2 réunions consécutives.

Cependant, la formalisation du problème a été réalisée sous la forme d'un problème d'optimisation linéaire en variables binaires en se limitant à l'objectif de synthèse entre la maximisation des voeux des participants et la minimisation du nombre de tables nécessaires. Les autres objectifs sont traduits en terme de contraintes.

### 2.2.1 Données

Les données que manipule le problème sont les suivantes :

- $Per$  : Ensemble des périodes  $1..PeriodeNb$
- $Ses$  : Ensemble des sessions. Une session représente un ensemble de périodes consécutives.
- $Par$  : Ensemble des participants  $1..ParticipantNb$
- $T$  : Nombre de tables disponibles  $1..ParticipantNb/2$

### 2.2.2 Variables de décision

La variable  $X_{i,j,p}$  détermine si la réunion entre les participants  $i$  et  $j$  est effectuée pendant la période  $p$ .

$$X_{i,j,p} = (0,1) \text{ avec } \begin{cases} = 1 & \text{si la rencontre entre les participants } i \text{ et } j \text{ a été planifiée sur le créneau } p \\ = 0 & \text{sinon.} \end{cases}$$

La valeur de la variable de décision  $Cmax$  donne le nombre de tables réellement nécessaires à l'organisation des rencontres.

$$Cmax \in 1..T$$

### 2.2.3 Fonction économique

La fonction économique représente une synthèse entre l'objectif de maximiser les souhaits des participants et la minimisation du nombre de tables nécessaires.

En demandant un rangement (ordre complet) sur les rencontres, une importance (poids) est affectée à chaque rencontre souhaitée. La fonction économique correspond alors à la somme des préférences cumulées des rendez-vous retenus.

$$Max \sum_{(i,j)} Pref_{i,j} \cdot X_{i,j} - Cmax$$

### 2.2.4 Contraintes

- unicité de la planification d'une réunion entre deux participants (contrainte de set packing)

$$\forall (i,j) \in R, \sum_{p \in Per} X_{i,j,p} \leq 1$$

- un participant ne peut être affecté qu'à une seule réunion par période (contrainte de set packing)

$$\forall p \in Per, \forall q \in Par, \sum_{(i,j) \in R_q} X_{i,j,p} \leq 1$$

- nombre de tables disponibles qui garantit qu'au moins 1/3 des participants est libre sur une période (contrainte de Knapsack)

$$\forall p \in Per, \sum_{(i,j) \in R} X_{i,j,p} \leq T$$

- au plus 2 réunions consécutives pour une personne dans une même session

$$\forall q \in Par, \forall p \in Per_{-2}, \sum_{(i,j) \in R_q} \sum_{w \in [0..2]} X_{i,j,p+w} \leq 2$$

- équilibre du nombre de réunions par période (contrainte min-max sur les périodes)

$$\forall p \in Per, \sum_{(i,j) \in Par} X_{i,j,p} \leq Cmax$$



- répartition équilibrée des réunions sur les 2 jours

$$\forall q \in Par, \sum_{(i,j) \in R_q} \left( \sum_{p1 \in Day1} X_{i,j,p1} - \sum_{p2 \in Day2} X_{i,j,p2} \right) \leq D$$

- nombre minimal de réunions assurées pour chaque participant

$$\forall q \in Par, \sum_{(i,j) \in R_q} \sum_{p \in Per} X_{i,j,p} \geq Min_q$$

## 2.3 Solution logicielle existante

Une première version d'un prototype d'optimisation appelée "Meetme" a été élaborée pour résoudre ce problème et a été utilisée avec succès pour organiser les précédentes bourses d'échanges de technologie.

### 2.3.1 Méthode de résolution

En se limitant à l'objectif de maximisation des préférences des participants, le problème est assimilable à un problème NP-difficile dit de "Set Packing" pour lequel il existe des algorithmes approchés efficaces. Mais la prise en compte des autres objectifs introduit des contraintes parasites vis-à-vis de la structure de set packing qui ont conduit à utiliser un solveur MILP, en l'occurrence le solveur GLPK.

### 2.3.2 Stockage informatique

L'inscription des participants à une bourse d'échanges est faite via un site web. Les informations personnelles, ainsi que les vœux des participants sont sauvegardés dans une base de données. Ces données sont ensuite converties au format XML afin d'être traitées par le système d'aide à la décision. Les données nécessaires à la phase d'optimisation sont ensuite construites pour être délivrées au solveur GLPK sous un format lp (.mod). Les résultats en sortie du solveur sont de nouveau converties au format XML pour pouvoir être insérés dans la base de données et être utilisées via le site web.

### 2.3.3 Implémentation

Hormis le solveur GLPK, le système d'aide a été implémenté en Java et intègre une partie développée en Python.

## 2.4 Critique

Le logiciel "Meet-me" répond aux besoins de l'utilisateur pour organiser une bourse d'échanges, cependant des améliorations peuvent être apportées. Le transfert des données nécessite beaucoup d'étapes : les vœux des participants qui ont été introduits via le site web sont enregistrés dans une base de données, ils sont ensuite récupérés sous un format XML pour être délivrés au solveur sous un format lp. Les résultats en sortie du solveur se présentent sous le format XML pour être ensuite réinsérés dans la base de données et être utilisables via le site web.

Une autre amélioration peut être faite au niveau du temps de la résolution. Du fait que la résolution exacte avec le modèle complété en GLPK prend un temps significatif, le logiciel lance une résolution d'abord avec les 3 premiers souhaits de chaque participants, ensuite avec les 6 et les 9 premières préférences.

Le logiciel "Meet-me" ne prend pas en compte les contraintes sur le nombre minimal de réunions pour une personne et sur la minimisation des distances à parcourir entre deux réunions. De plus, il sera souhaitable de favoriser la planification des réunions vers le début d'une journée pour laisser du temps libre aux participants à la fin de l'événement.

## 3 Positionnement scientifique

### 3.1 Classe de problème d'optimisation

L'organisation des bourses d'échanges ressemble aux événements de type "Business Speed Dating" où les participants ont l'opportunité de rencontrer beaucoup de personnes en peu de temps ([15]).

Le problème de planification de bourses d'échanges s'inscrit dans la liste des problèmes d'ordonnement et de planification (Scheduling problems) qui impliquent la résolution d'un ordonnancement optimal sous différentes contraintes. La classification des problèmes proposée par CSPLib [19] inscrit un nombre important de problèmes différents dans la classe des problèmes d'ordonnement. Ainsi, à côté des problèmes classiques de Job-shop, Flow-shop et Open-shop, on retrouve des problèmes qui traitent la planification des rencontres sportives (Social golfer problem, ACC basketball schedule, Darts tournaments) ou l'organisation des horaires dans une université (Balanced academic curriculum problem).

Le problème d'organisation des bourses d'échanges peut être vu comme une synthèse entre le problème de planification de rendez-vous (Meeting scheduling problem) et le problème de planification des cours académiques dans une université (Balanced academic curriculum problem).

Dans un problème de planification des réunions, on doit planifier un ensemble de  $n$  rendez-vous entre  $m$  personnes de façon à satisfaire au mieux les contraintes propres à chaque personne. Chaque réunion, affectée à un sous-ensemble de  $m$  personnes, est fixé à une date, emplacement et durée données. Ainsi, chaque personne a un ensemble de réunions où elle doit être présente sans être en retard.

Le problème des horaires universitaires vise à concevoir un programme équilibré pour chaque période tout en respectant les règles administratives et académiques. Les cours doivent être affectés en respectant le nombre maximal de périodes académiques et certains cours peuvent avoir d'autres cours comme pré-requis. Le planning doit contenir un nombre minimal de matières afin de pouvoir obtenir la totalité des crédits nécessaires, mais il doit aussi respecter une charge maximale pour éviter la surcharge. D'autres versions du problème prennent en compte les ressources d'université et les préférences des étudiants pour certains cours.

### 3.2 Besoins exprimés et options existantes

**Organisateurs** Les rendez-vous de type Business Speed Dating peuvent créer des opportunités d'affaires, d'échanges entre les entreprises ou organisations, et de réaliser de nouvelles relations. Ainsi, de nombreuses organisations proposent des événements de type bourse d'échanges d'affaires où l'on peut réaliser beaucoup de rencontres en très peu de temps.

Parmi celles-là, on peut citer les exemples suivants :

La Chambre de Commerce et d'Industrie de la Rochelle qui proposent « Les Rendez-vous d'affaires » [20] où les participants peuvent s'inscrire via le site web de l'événement et obtenir jusqu'à 12 rendez-vous cibles en une journée. Pour participer à un événement, il faut effectuer jusqu'à 20 souhaits de rencontre, classés par ordre de priorité et ensuite, un logiciel spécifique va construire l'agenda de chaque participant avec des rendez-vous d'une durée de 25 minutes.

Un autre organisateur est la Confédération Générale des Petites et Moyennes Entreprises qui propose un salon de l'entreprise [21] avec des rendez-vous d'affaires d'une durée de 20 minutes. Comme dans le cas précédent, les participants s'inscrivent via le site web et sélectionnent 15 souhaits de rendez-vous, mais l'organisateur ne précise rien sur le nombre de réunions auquel on peut s'attendre.

**Logiciels** En ce qui concerne les logiciels existants, il n'existe pas encore de projets open-source capables de répondre aux besoins des entreprises. Mais on peut retrouver des logiciels propriétaires de gestion qui proposent des modules d'optimisation de planification.

L'un de ces logiciels est Visual Andfast [22] qui permet, entre autres, la planification automatique de rencontres de type speed-dating, une optimisation de rendez-vous professionnels d'affaires et une visualisation graphique sous forme de planning des pointages horaires et affaires.

Il existe aussi des logiciels qui sont dédiés seulement à l'organisation des bourses d'échanges comme Contact25 : B2B (Business to business) Promoter Package [24] ou encore Extreme Networking [23]. Ce dernier propose même une démonstration Flash du logiciel. Ainsi on peut voir qu'avec ce logiciel, on peut créer un événement en introduisant les paramètres nécessaires comme le nombre de participants, le nombre de périodes, la durée des rendez-vous et des pauses. On peut ensuite voir le planning créé comme dans la figure 3.

### 3.3 Littérature existante

En ce qui concerne la littérature existante sur ce type de problème, on ne trouve pas beaucoup de publications qui traitent l'optimisation des événements de type Business Speed-Dating. Dans l'article écrit par Robert A. Russell et Timothy L. Urban [1], on trouve des méthodes de résolutions efficaces pour le problème de Power-Networking proposé par la Chambre de Commerce de Tulsa (Tulsa Metro Chamber), Oklahoma.

Dans le problème de type Power-networking les participants veulent rencontrer le plus de personnes en une courte période de temps, la principale différence avec le problème posé par M. Poncelet étant que les participants n'ont pas la possibilité d'exprimer leurs préférences pour les rendez-vous. L'objectif est donc la maximisation des uniques contacts réalisés par les participants au cours des réunions. Plus précisément, l'objectif principal est de maximiser le nombre minimum de contacts de chaque personne, avec un second objectif étant de minimiser le nombre de participants qui réalisent le plus petit nombre de contacts.

L'article propose des modèles de programmation linéaire et par contraintes pour optimiser les problèmes de petite taille, et une méthode heuristique pour les problèmes à grande échelle comme pour l'organisation de la Chambre de Commerce avec plus de 200 participants à chaque événement. Ce problème n'étant pas déjà traité auparavant, les auteurs s'inspirent du problème d'organisation des tournées de golf (social golfer problem). Dans le problème des tournées de golf, le but est de planifier des rencontres entre  $m$  groupes de  $n$  golfeurs pendant  $p$  semaines de sorte qu'aucun golfeur joue avec un autre joueur plus d'une fois. L'objectif est de maximiser le nombre de semaines de l'événement sportif.

Dans les modèles proposés dans cet article, les fonctions objectifs minimisent le nombre maximal de fois où un participant est assis à une table avec une chaise libre et le nombre de personnes qui sont assises un nombre maximal de fois à une table avec une chaise libre. Les contraintes imposent :

- L'unicité d'une rencontre entre deux personnes
- Tous les participants ont une réunion pendant chaque période
- Le planning ne doit pas dépasser le nombre de tables
- A chaque table il y a au plus une chaise vide

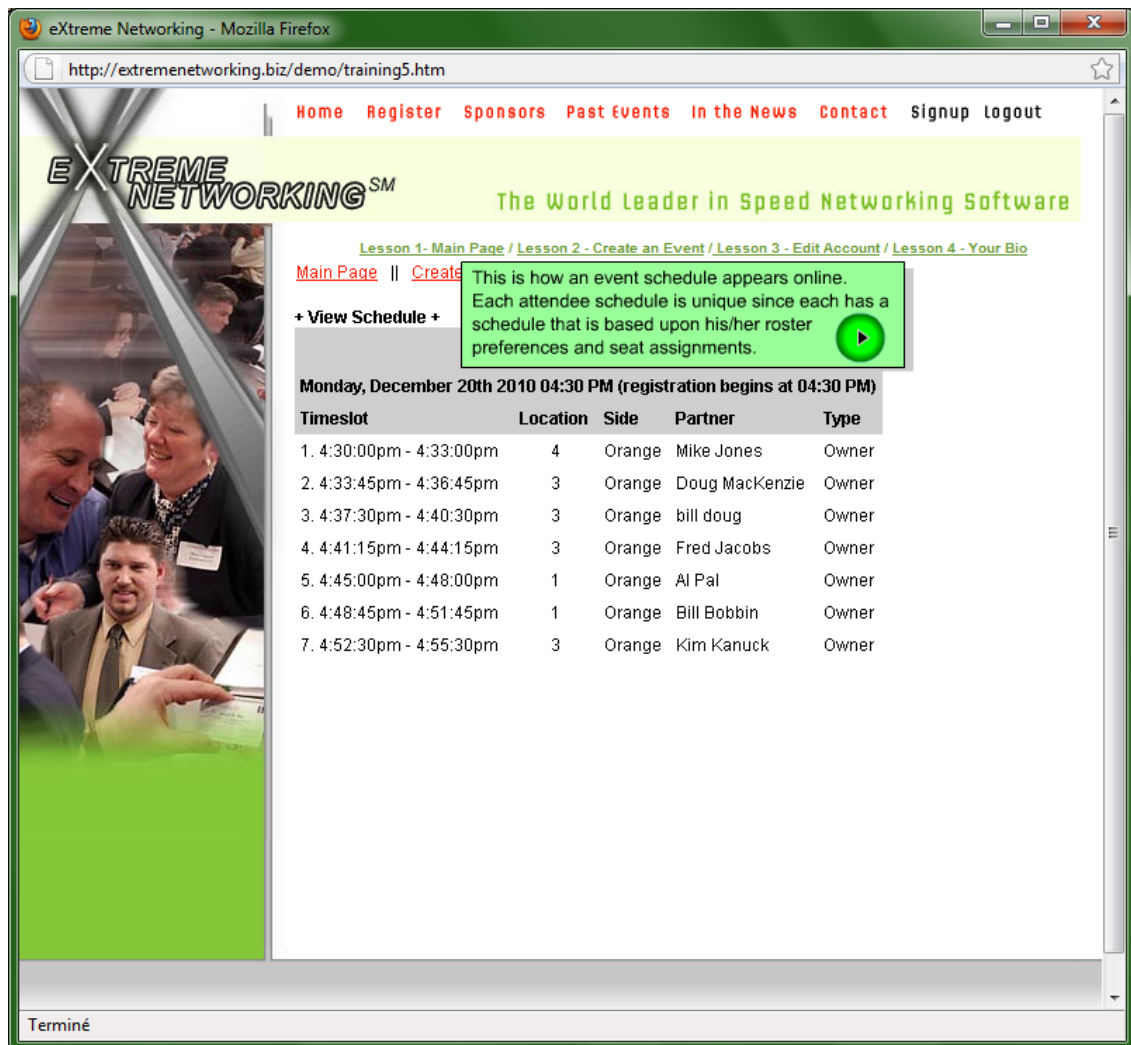


FIGURE 3 – Le logiciel Extreme networking

– L’hôte reste à la même table pour toute la durée de l’événement.

Comme le modèle avec des variables binaires et entières peut être utilisé seulement pour les problèmes de petite taille, les auteurs proposent aussi un modèle de programmation par contraintes, plus rapide. Les deux modèles prennent comme paramètres le nombre de personnes, le nombre de chaises autour d’une table et le nombre de périodes.

En ce qui concerne l’heuristique proposée pour résoudre des problèmes de grande taille avec  $P$  participants, l’article suggère l’algorithme de résolution du problème d’organisation de tournées de golf pour  $S * T$  personnes ( $S$  étant le nombre de chaises autour d’une table et  $T$ , le nombre de tables disponibles). Ensuite on enlève  $S * T - P$  personnes à chaque rotation à condition de ne pas effacer deux participants qui sont assises à la même table.

Les tests des résolutions ont été réalisés sur trois instances entre 172 et 227 participants avec 10 chaises autour de chaque table et 5 périodes. Pour les problèmes de petite taille, les instances contiennent entre 33 et 65 participants avec de 7 à 10 tables et entre 5 et 7 chaises par table pendant 5 périodes.

La résolution du modèle de programmation avec des variables binaires et entières a été réalisé avec CPLEX 11.1 et rend la solution après plus de 30 heures pour les petites instances et ne donne pas de réponse après 48 heures pour les problèmes de grande taille. Le modèle de programmation par contrainte avec ILOG CP Optimizer 2.1 rend la solution pour les grandes instances après plus de 40 heures. En ce qui concerne la résolution approchée, le temps de résolution est inférieur à une minute pour toutes les instances utilisées lors des différents tests et obtient des solutions qui sont proches de l'optimum.

### 3.4 Littérature connexe

Comme le problème d'organisation des bourses d'échanges de technologies n'a pas encore été abordé, nous allons passer en revue la littérature existante pour la recherche liée à ce type de problème et les méthodes utilisées pour les résoudre. Comme nous l'avons mentionné auparavant, l'optimisation des bourses d'échanges peut être vue comme une synthèse entre le problème de planification de rendez-vous (Meeting scheduling problem) et le problème de planification des cours académiques dans une université (Balanced academic curriculum problem) due aux nombreuses contraintes communes.

**L'optimisation des horaires dans une université** L'article écrit par D. de Werra [10] présente les différents types de problèmes pour l'organisation des horaires universitaires (class-teacher timetabling problems) avec leur complexité et décrit l'algorithme Tabou qui résout avec succès les problèmes d'optimisation combinatoire et plus spécialement les problèmes de planification. Les nombreux types de problèmes qui concernent l'optimisation des horaires académiques diffèrent entre elles par le nombre et le type de contraintes prises en compte. Pour un modèle classique :

- l'emploi du temps est divisé en  $p$  périodes
- un cours implique un groupe d'étudiants
- un professeur/groupe d'étudiants ne peut pas avoir plus d'un cours par période.

Les autres types de résolutions prennent en compte aussi la disponibilité des étudiants/professeurs et s'appliquent au types de problèmes où l'emploi du temps est déjà partiellement construit et ne doit pas être changé. Ensuite on peut rajouter des contraintes qui permettent de compacter la planification des emplois du temps. Tous ces types de problèmes sont résolus par l'algorithme de coloration des graphes décrit dans les articles [8] et [7].

En ce qui concerne la résolution des problèmes d'organisation des horaires universitaires avec des contraintes nombreuses nous avons sélectionné trois articles qui proposent trois méthodes différentes ayant des bons résultats. Ainsi l'article [12] propose un modèle de programmation linéaire et un modèle de programmation par contraintes pour la résolution exacte, et un algorithme de type tabou pour la résolution approchée. L'article écrit par Yen-Zen Wang[2] en 2003 décrit la résolution approchée avec un algorithme génétique et la publication [13] propose un algorithme multi-objectif hybride pour résoudre le problème de planification des examens.

Le problème résolu dans l'article [12] représente l'organisation d'emplois du temps à l'Ecole des Mines de Nantes pour les étudiants de troisième année. Les étudiants doivent choisir une spécialité et un ensemble de cours. Ainsi le problème qui doit minimiser les conflits entre les différents cours a plusieurs contraintes comme :

- deux cours appartenant à une même spécialité ne doivent pas être planifiés en même temps
- les pré-requis pour chaque cours doivent être planifiés avant
- le nombre de cours panifiés en même temps ne doit pas dépasser le nombre de salles libres
- le planning doit respecter les disponibilités des professeurs et des étudiants
- l'horaire doit être facilement modifiable.

Les instances pour ce problème comportent 120 étudiants, 30 cours différents, 7 périodes. La méthode de résolution propose 10 solutions en 10 minutes environ et l'algorithme tabou améliore les solutions initiales avec 10% de conflits en moins.

L'article [2] propose une résolution approchée avec un algorithme génétique, la solution rendue satisfaisant les contraintes suivantes :

- un professeur ne peut pas avoir plusieurs cours en même temps
- il y a un seul professeur qui intervient dans un cours
- l'emploi du temps doit contenir des pauses pour les activités extra-scolaires
- dans une salle il doit y avoir un seul cours à une période donnée
- les préférences pour les cours des professeurs et des étudiants doivent être respectés le mieux possible.

Dans les instances utilisées pour les tests il y a 8h de cours par jour, 10 groupes d'étudiants et 26 professeurs. La dimension de la population de l'algorithme génétique est égale à 30 et après 4000 générations, la solution satisfait les contraintes essentielles.

Enfin, l'article [13] présente un algorithme hybride qui utilise des opérateurs de recherche locale à la place des opérateurs génétiques traditionnels de recombinaison. La solution doit minimiser la période des examens et en même temps offrir un temps maximal aux étudiants entre les examens. Chaque étudiant a un seul examen par période et doit respecter la capacité de la salle où l'examen est planifié. L'algorithme a été testé avec 15 instances ayant entre 611 et 21266 étudiants, de 81 à 800 examens planifiés et retourne une solution non dominée après plus de 25 minutes pour les instances de petite taille et peut arriver à plus de 13 heures pour les grandes instances.

**La planification des rendez-vous** Le problème d'ordonnement de réunion est souvent décrit comme un problème de satisfaction de contraintes distribuées. Le Meeting Scheduling Problem est un problème de négociations réparties entre les différents utilisateurs. Par conséquent, les agents sont associés aux utilisateurs et non pas aux réunions. Les réunions sont les variables auxquelles on doit affecter des intervalles de temps et elles sont partagées par tous les agents qui assistent à la même réunion. Les articles [14], [5], [3] et [16] proposent des systèmes multi-agents qui résolvent le problème de planification des rendez-vous avec différentes contraintes respectées.

L'article [14] publié en 2002 étudie la relation entre la maximisation des préférences des agents, la maximisation de la confidentialité des préférences et la minimisation du temps de résolution avec un système multi-agent qui prend en compte les préférences des utilisateurs et les contraintes de distance entre deux réunions consécutives.

Le système multi-agents proposé dans l'article [5] définit la solution optimale comme la solution qui a le plus haut niveau moyen de préférence parmi tous les choix possibles. Le modèle tente d'imiter la vie réelle et ne rend pas les préférences d'un individu publiques. Dans cet article, les auteurs présentent une méthode d'estimation des préférences qui permet de trouver des solutions optimales aux problèmes des négociations sans avoir besoin de connaître les modèles de préférence exactes de tous les participants à la réunion. Les préférences sont construites à partir des réponses de chaque participant lors de la négociation.

L'article [3] propose une nouvelle approche évolutive à base d'agents pour tout problème dynamique de type meeting scheduling. Dans cette approche, seulement les préférences des utilisateurs sont relaxées tout en maintenant la consistance du problème. La méthode peut atteindre efficacement la solution optimale en utilisant uniquement le minimum possible de communications asynchrones. Ce but est atteint avec un nombre de messages échangés minimal tout en essayant de préserver au maximum la vie privée des utilisateurs concernés.

### 3.5 Synthèse

Même s'il existe de nombreux logiciels sur le marché qui proposent des résolutions efficaces au problème d'optimisation d'organisation de bourses d'échanges de technologie, à cause de différents verrous scientifiques il n'existe pas beaucoup de publications qui présentent les recherches effectuées dans ce domaine.

En revanche, il est possible de trouver de nombreuses recherches effectuées sur des problèmes connexes comme l'organisation des horaires universitaires ou la planification des réunions. Nous avons pu voir dans les paragraphes antérieurs que ces deux problèmes ont beaucoup de points communs avec le Business Speed Dating. Ainsi on retrouve des objectifs comme la maximisation des préférences exprimées par les utilisateurs, des contraintes sur le degré de chargement du planning/emploi du temps, l'unicité des rendez-vous/cours sur une période pour chaque participant et sur la capacité des salles où les réunions/cours auront lieu.



## 4 Contributions

Dans ce chapitre nous allons détailler les modifications apportées au modèle existant, les méthodes de résolution exacte et approchée, ainsi que la solution logicielle proposée.

### 4.1 Simplification de l'utilisation

La version précédente de MeetMe nécessite d'effectuer manuellement une dizaine d'étapes depuis la récupération des données (identités, préférences, ...) des participants jusqu'à l'enregistrement des rendez-vous générés pour l'affichage final.

Nous avons donc décidé d'optimiser ce processus pour le rendre plus facile d'utilisation. Pour cela, nous avons créé une interface graphique conviviale à l'aide de la librairie Qt 4.7. Cette interface rassemble toutes les étapes nécessaires à la génération de la solution à l'aide de simples clics.

#### 4.1.1 Importation des données

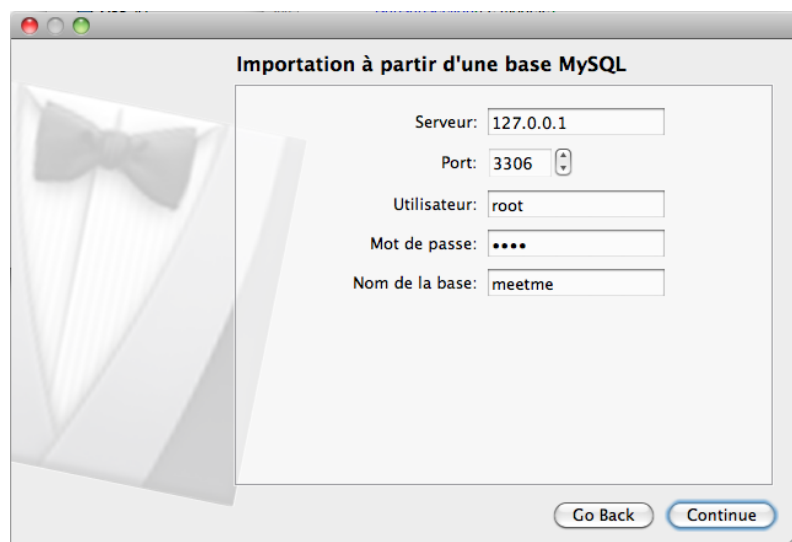


FIGURE 4 – Informations de connexion à la base de données MeetMe

Cette importation automatisée grâce aux outils de connexion fournis par la librairie Qt permet de récupérer les informations sur les participants ainsi que leurs préférences. Pour récupérer ces données, l'organisateur doit simplement entrer les informations permettant de se connecter à sa base de données MySQL, comme indiqué sur la figure 4.

#### 4.1.2 Définition des données supplémentaires

L'utilisateur doit aussi fournir des informations (cf figure 5) qui ne se trouvent pas dans la base de données, à savoir :

- Le nombre de journées
- Le nombre de périodes par journée
- Le nombre de tables disponibles
- Le nombre maximal de rendez-vous successifs
- Le nombre minimal de rendez-vous par personne
- Le nombre maximal de rendez-vous par personne

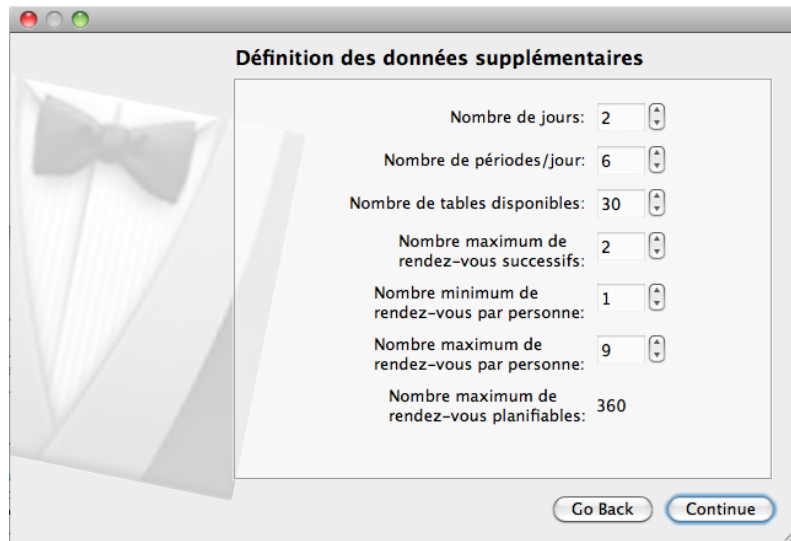


FIGURE 5 – Informations complémentaires

#### 4.1.3 Choix de la méthode de résolution

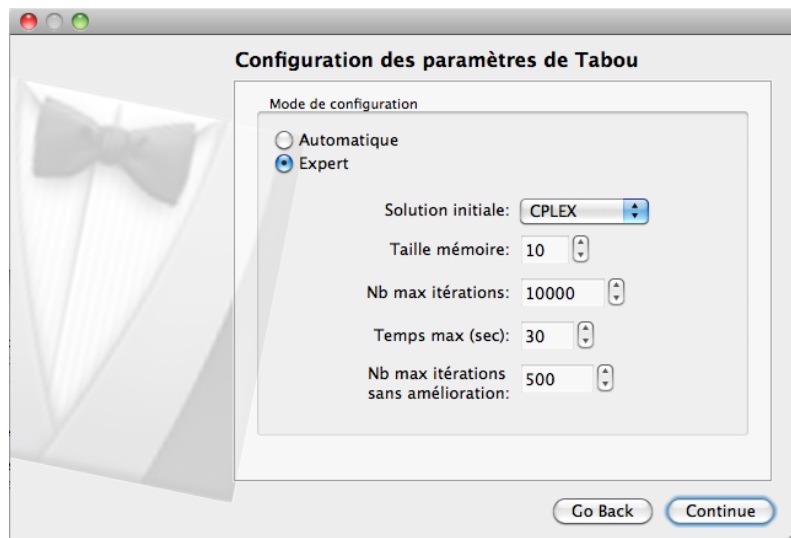


FIGURE 6 – Paramètre de la résolution Tabou

Le décideur peut ensuite choisir entre 2 méthodes de résolution (exacte OU approchée). S'il choisit la résolution exacte, cette dernière démarre dès que l'utilisateur passe à l'étape suivante. Sinon, dans le cas de la résolution approchée, des paramètres supplémentaires (cf figure 6) lui sont demandés pour savoir comment il souhaite résoudre le problème (sauf dans le cas de la configuration automatique).

Lorsque l'utilisateur choisit la résolution approchée, des informations sont affichées lors de la résolution alors que pour la résolution exacte, l'affichage se fait uniquement dans la console.

#### 4.1.4 Affichage du résultat

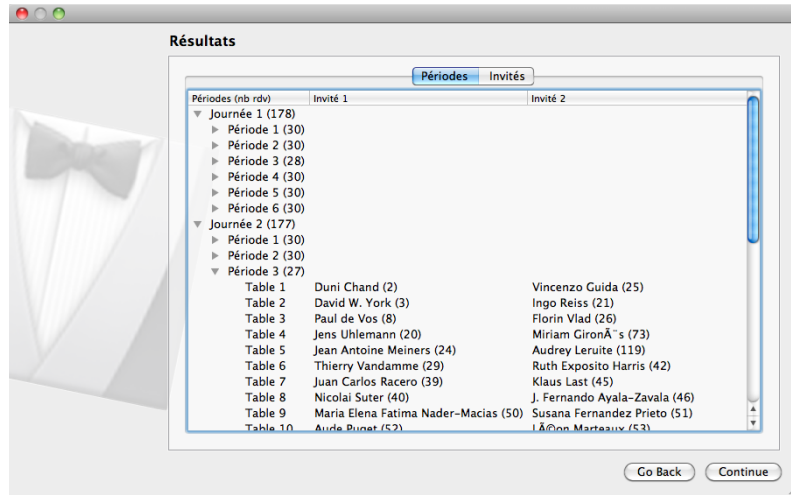


FIGURE 7 – Affichage des résultats

L'affichage du résultat peut se faire de 2 manières différentes : soit par journée, soit par participant. Dans les 2 cas, l'affichage se fait de manière arborescente avec l'indication du nombre de rendez-vous affectés à chaque entité, comme on peut le voir sur la figure 7.

## 4.2 Résolution exacte

### 4.2.1 Version précédente

Dans la version précédente de MeetMe, les développeurs ont préféré utiliser le solveur GLPK grâce à sa gratuité. Cependant, ils ont remarqué que le solveur CPLEX est plus rapide pour résoudre ce type de problème.

### 4.2.2 Solveur utilisé

Nous avons donc décidé d'utiliser le solveur CPLEX v12.2, grâce à une licence gratuite pour l'éducation. Pour intégrer au mieux le solveur dans l'application, nous l'avons utilisé en mode librairie. Ainsi, les données sont fournies directement au solveur via une structure adaptée et réciproquement pour le résultat.

### 4.2.3 Le modèle

Voici le modèle que nous avons implémenté sur CPLEX.

Définition des données d'entrée :

- $PartNb$  : nombre de participants
- $JoursNb$  : nombre de jours
- $PerNb$  : nombre de périodes par jour
- $TabNb$  : nombre de tables
- $pref_{i,j} \in [0..96]$  : indique le niveau de préférences cumulées de  $i$  envers  $j$  et  $j$  envers  $i$ .

Définition des variables :

- $rdv_{i,j,d,p} \in [0..1]$  : égale 1 si le rendez-vous est planifié entre les participants  $i$  et  $j$  durant la période  $p$  de la journée  $d$ ; 0 sinon.

Fonction objectif :

$$\max \sum_{i=1}^{PartNb} \sum_{j=1}^{PartNb} \sum_{d=1}^{JoursNb} \sum_{p=1}^{PerNb} (pref_{i,j} + C * (PerNb - p + (JoursNb - d) * \frac{PerNb}{3})) * rdv_{i,j,d,p}$$

$C$  : coefficient de normalisation pour la répartition des rendez-vous.

Pour ce problème, nous devons traiter 2 objectifs :

- Maximiser les préférences
- Mettre les rendez-vous de préférence vers le début

La maximisation des préférences se traduit par la multiplication de la variable binaire par la valeur correspondant aux préférences cumulées.

Pour mettre les rendez-vous de préférence vers le début, nous avons décidé d'ajouter un coefficient à chaque variable binaire, correspondant à son rang. Ainsi, chaque jour peut se décomposer en trois groupements de périodes comme ceci :

Journée 1	Journée 2
Groupe 1	Groupe 4
Groupe 2	Groupe 5
Groupe 3	Groupe 6

Suivant ce schéma, nous avons fait de telle sorte que le groupe 1 soit privilégié dans tous les cas, puis que les groupes 2 et 4 se trouvent au même rang moins bons que 1 mais meilleurs que 3 et 5, pareil pour les groupes 3 et 5 et enfin le groupe 6 qui reste le dernier groupe.

Les contraintes :

- Une personne ne peut avoir qu'un seul rendez-vous par période :

$$\forall d \in [1..DayNb], \forall p \in [1..PerNb], \forall i \in [1..PartNb], \sum_{j=0}^{PartNb} rdv_{i,j,d,p} \leq 1$$

- 2 personnes ne peuvent se rencontrer qu'une seule fois sur la totalité des période :

$$\forall i, j \in [1..PartNb]^2, \sum_{d=0}^{DayNb} \sum_{p=0}^{PerNb} rdv_{i,j,d,p} \leq 1$$

- Le nombre de rendez-vous ne doit pas excéder le nombre de tables :

$$\forall d \in [1..DayNb], \forall p \in [1..PerNb], \sum_{i=1}^{PartNb} \sum_{j=1}^{PartNb} rdv_{i,j,d,p} \leq TabNb$$

- Une personne doit avoir un nombre minimal  $N_1$  de rendez-vous :

$$\forall i \in [1..PartNb], \sum_{j=1}^{PartNb} \sum_{d=1}^{DayNb} \sum_{p=1}^{PerNb} rdv_{i,j,d,p} \geq N_1$$

- Une personne peut avoir un nombre maximal  $N_2$  de rendez-vous :

$$\forall i \in [1..PartNb], \sum_{j=1}^{PartNb} \sum_{d=1}^{DayNb} \sum_{p=1}^{PerNb} rdv_{i,j,d,p} \leq N_2$$

- Une personne ne peut pas avoir plus de 2 rendez-vous consécutifs :

$$\forall i \in [1..DayNb], \forall j \in [1..PartNb], \forall d \in [1..DayNb], \forall p \in [1..PerNb - 3],$$

$$rdv_{i,j,d,p} + rdv_{i,j,d,p+1} + rdv_{i,j,d,p+2} \leq 2$$

#### 4.2.4 Résultats et analyse

Dans cette partie, nous allons présenter les résultats obtenus avec l’implémentation du modèle sous CPLEX. Pour voir l’influence des différents paramètres sur le temps de résolution, nous avons effectué des tests en modifiant ces paramètres un à un.

**Configuration** Les tests ont été effectués sur un MacBook Pro équipé d’un processeur Intel core i7 quadricœur cadencé à 2,2 GHz accompagné de 4 Gio de mémoire RAM cadencée à 1333 MHz.

**Instances de tests** Pour effectuer les tests, nous avons utilisé 2 jeux de données réelles fournis par le décideur. La première instance, nommée « isam », est composée de 125 personnes alors que la deuxième, nommée « sanan », en a 140. Les deux instances ont exactement le même nombre de préférences.

**Variation du nombre de sessions** Ici, nous avons étudié l’impact du nombre de sessions sur le temps de résolution.

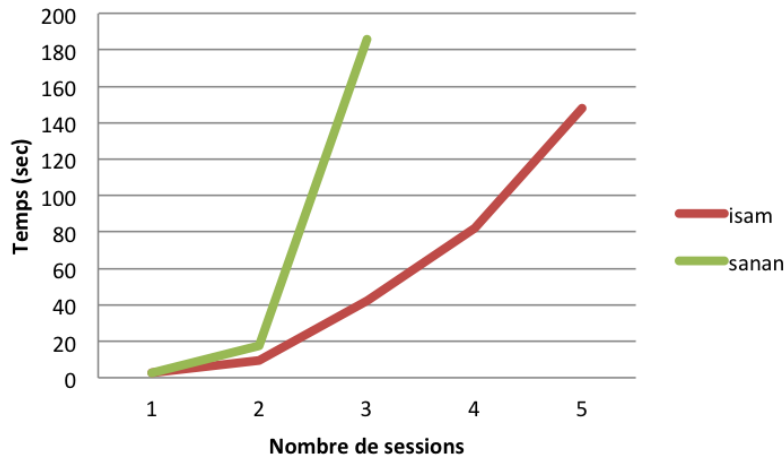


FIGURE 8 – Variation du nombre de sessions

Nous pouvons remarquer, d’après la figure 8, que le temps de résolution augmente exponentiellement avec le nombre de sessions. Nous pouvons aussi observer que le nombre de personnes ne fait qu’amplifier ce phénomène. Pour l’instance contenant 140 participants le solveur nécessite plus de 5 minutes pour effectuer la planification avec 4 sessions ou plus. Ceci est donc considéré comme trop long.

**Variation du nombre de périodes par session** Dans ce paragraphe, nous avons étudié l’impact du nombre de périodes par session sur le temps de résolution (Figure 9). Ces tests sont similaires à la partie précédente, car ils modifient tous les deux le nombre total de périodes.

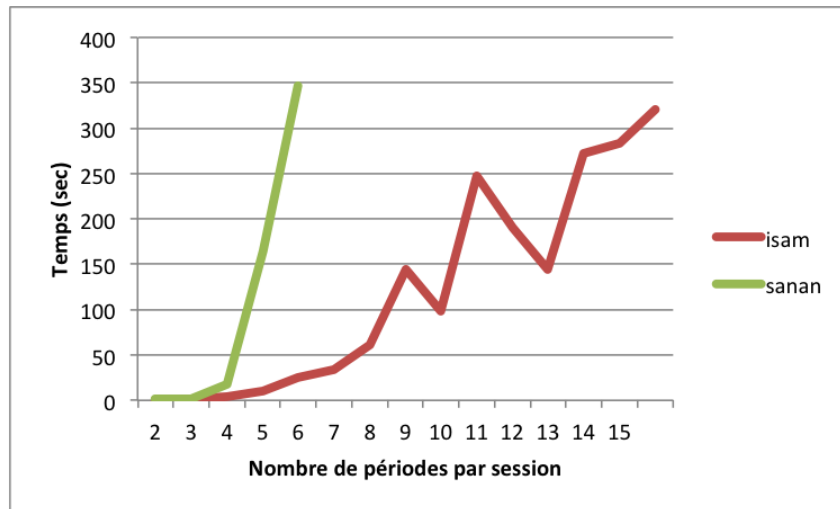


FIGURE 9 – Variation du nombre de périodes par session

Cependant, nous pouvons voir que sur l'instance « isam », le comportement non uniforme. Nous avons donc effectué ces tests plusieurs fois pour vérifier si ceci venait d'une perturbation sur le processeur, mais ils donnaient tous les mêmes conclusions. Nous en avons déduit que ce phénomène pouvait être dû à des contraintes qui ont des impacts différents suivant la répartition des périodes durant les sessions. Ainsi, nous pouvons remarquer des temps de réponse courts pour les événements allant jusqu'à 8 périodes par session.

En revanche, pour l'instance « sanan », le temps de réponse devient très rapidement trop long (supérieur à 5 minutes).

Nous pouvons en déduire que le nombre de périodes par session ne fait qu'augmenter le temps de réponse. Encore une fois le nombre de personnes influence beaucoup le temps de réponse de la méthode exacte sous CPLEX.

**Variation du nombre de tables** Dans cette partie nous avons analysé l'impact du nombre de tables existantes sur le temps de résolution (Figure 10). Les abscisses correspondent au pourcentage de tables par rapport au nombre de participants. Nous sommes partis de 25% car, en dessous de cette valeur, le problème devient difficile à résoudre. Aussi, nous nous sommes limités à 50% car cette valeur correspond au nombre maximal de tables utilisées simultanément dans le cas où tout le monde est en rendez-vous en même temps.

Nous pouvons voir que plus il y a de tables, et plus le temps de résolution est court. Ceci est dû au fait que le problème est de moins en moins contraint. On peut aussi observer que, pour l'instance « sanan », à compter d'un certain pourcentage de tables, le temps de réponse commence à ré-augmenter du fait que ceci propose trop de possibilités de rendez-vous.

**Variation du nombre maximal de rendez-vous successifs** Dans ce paragraphe, nous avons regardé l'influence du nombre maximal de rendez-vous consécutifs sur le temps de résolution (Figure 11).

Nous pouvons observer que le temps de réponse pour l'instance « isam » est très court alors contrairement à celui de « sanan ». On peut donc en conclure que ce facteur influence le temps de réponse lorsqu'il est associé à un grand nombre de personnes.

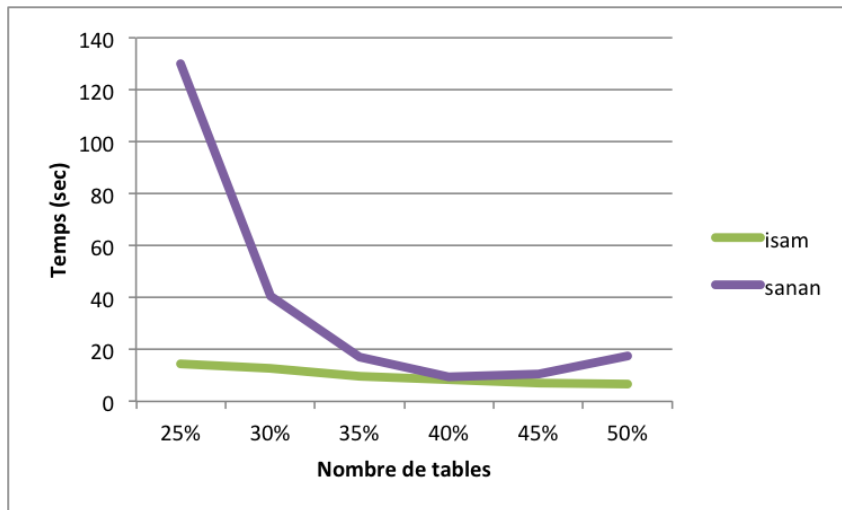


FIGURE 10 – Variation du nombre de tables

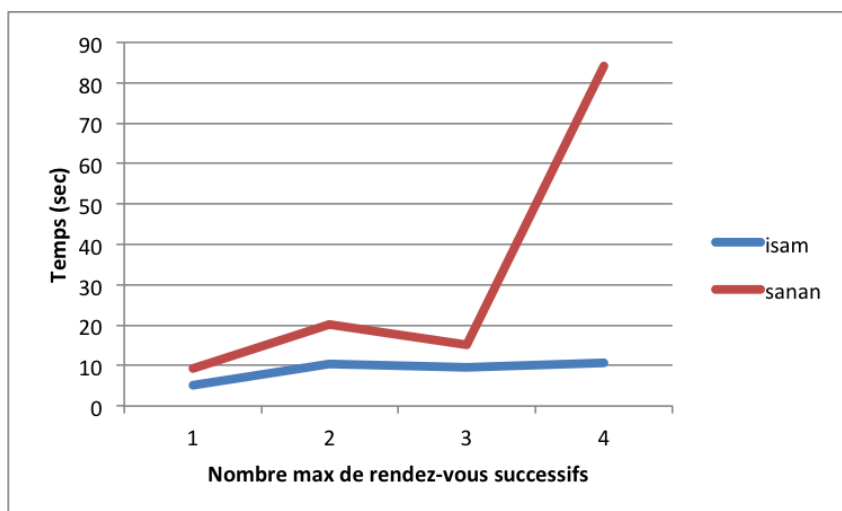


FIGURE 11 – Variation du nombre maximum de rendez-vous consécutifs

### 4.3 Résolution approchée avec Tabou

Etant donné que le temps de résolution avec une méthode exacte est assez long, il a donc fallu que l'on choisissent une méthode permettant de résoudre le problème plus rapidement, au détriment de la qualité de la solution. De plus, le décideur souhaite avant tout obtenir une bonne solution en peu de temps même si elle n'est pas optimale, lui permettant ainsi de pouvoir réorganiser manuellement certains rendez-vous en fonction des préférences fournies après la première planification.

Notre choix s'est donc tourné vers la méthode Tabou car elle permet d'arriver très rapidement à une solution assez proche de l'optimum.

#### 4.3.1 Structure de données

Pour avoir la plus grande rapidité, nous avons décidé d'utiliser plusieurs tableaux permettant d'indexer tout ce dont la méthode tabou a besoin. Bien évidemment, on peut assister à une redondance de données, mais qui est largement négligeable par rapport au temps qu'elle fait gagner. En effet, le fait d'indexer les données permet d'avoir un accès immédiat (complexité en  $O(1)$ ), aussi bien en lecture qu'en écriture, au lieu d'une complexité linéaire (qui augmente donc en fonction de la taille des données). Cependant, cette indexation a un coût en taille mémoire. Avec les ordinateurs actuels, ce problème est négligeable du moment que la taille des données n'est pas en complexité exponentielle.

Voici les différents tableaux utilisés :

- **Le tableau principal** : contenant les préférences, les planifications effectuées ainsi que les nombres de rendez-vous pour chacune des personnes (notés  $nb_i$ ). Ce tableau permet d'avoir un accès direct vers l'ensemble des informations qu'il contient.

		Personnes				
		1	2	3	4	5
Personnes	1	$nb_1$				
	2		$nb_2$			
	3			$nb_3$		
	4				$nb_4$	
	5					$nb_5$

Rdv planifiés

Préférences

- **Le tableau des périodes** : contenant, pour chaque personne, la liste (indexée par période) des personnes qu'elle va rencontrer. Ce tableau permet de savoir de manière directe si une personne a un rendez-vous planifié à une période donnée.

		Périodes							
		1	2	3	4	5	6	7	8
Personnes	1			2					
	2			1					
	3								
	4								
	5								

Personnes rencontrées

L'exemple ci dessus montre un exemple pour un rendez-vous planifié entre les participants 1 et 2 durant la période 3.

- **Le tableau des tables libres** : contenant le nombre de tables libres pour chaque périodes. Permettant ainsi de limité le nombre de rendez-vous au nombre de tables disponibles, en



décrémentant la case appropriée lors de l'ajout d'un rendez-vous.

Périodes							
1	2	3	4	5	6	7	8
Nombre de tables dispo pour chaque période							

- **La mémoire tabou** : pour la liste tabou, nous avons choisit d'insérer le numéro de l'itération à partir de laquelle le rendez-vous qui vient d'être créé ou modifié pourra de nouveau être modifié.

Mémoire Tabou	Personnes				
	1	2	3	4	5
Personnes	1	Numéro de			
	2	l'itération			
	3	Vide			
	4				
	5				

#### 4.3.2 Le voisinage

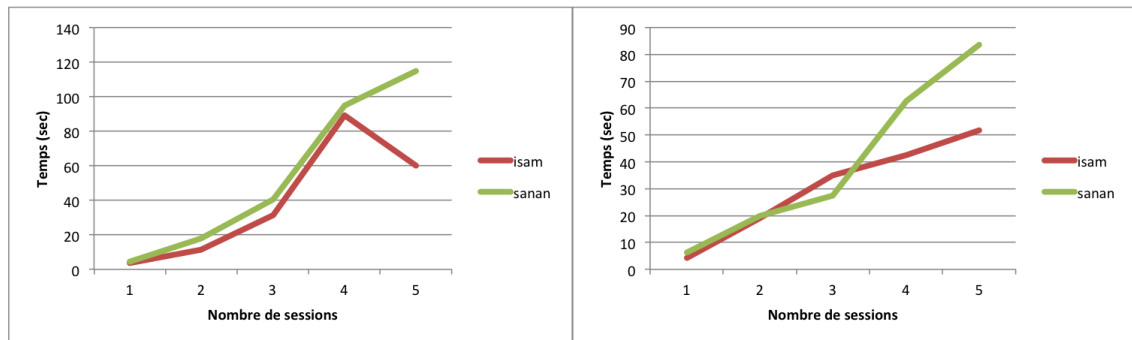
Pour pouvoir fonctionner, la méthode tabou doit parcourir un certain voisinage, dans lequel elle recherche le meilleur voisin qu'elle applique. Nous avons décidé de définir 3 voisinages distincts : *Pour améliorer la lisibilité, les rendez-vous seront noté comme ceci : (Participant1, Participant2, Période)*

- Le **swap** : Ce type de voisinage va permettre de tester toutes les combinaisons qu'il est possible de faire entre les 4 participants et 2 périodes impliqués dans 2 rendez-vous distincts existants. Soit deux rendez-vous  $(A, B, P1)$  et  $(C, D, P2)$ , le voisinage correspondant est donc celui-ci :
  - $(A, C, P1), (B, D, P2)$
  - $(A, D, P1), (B, C, P2)$
  - $(B, C, P1), (A, D, P2)$
  - $(B, D, P1), (A, C, P2)$
- L'**échange** : Ce type de voisinage permet d'échanger un et un seul participant d'un rendez-vous existant. Prenons le rendez-vous noté  $(A, B, P)$ . Le voisinage va donc chercher s'il existe une personne C pour laquelle le remplacement du rendez-vous  $(A, B, P)$  par  $(A, C, P)$  améliore la fonction objectif.
- L'**ajout** : Ce type de voisinage permet simplement d'ajouter un nouveau rendez-vous, uniquement dans le cas où le swap et l'échange n'ont pas réussi à améliorer la fonction objectif.

#### 4.3.3 Résultats et analyse

**Variation du nombre de sessions** Dans cette partie, nous avons étudié l'impact du nombre de sessions sur le temps de résolution (Figure 12).

Nous pouvons voir que, contrairement à la résolution avec CPLEX, le nombre de sessions influence le temps de résolution de manière linéaire. Ce qui permet de pouvoir acquérir une solution dans un temps plus acceptable (alors que la résolution CPLEX dépassait quelque fois les 5 minutes).

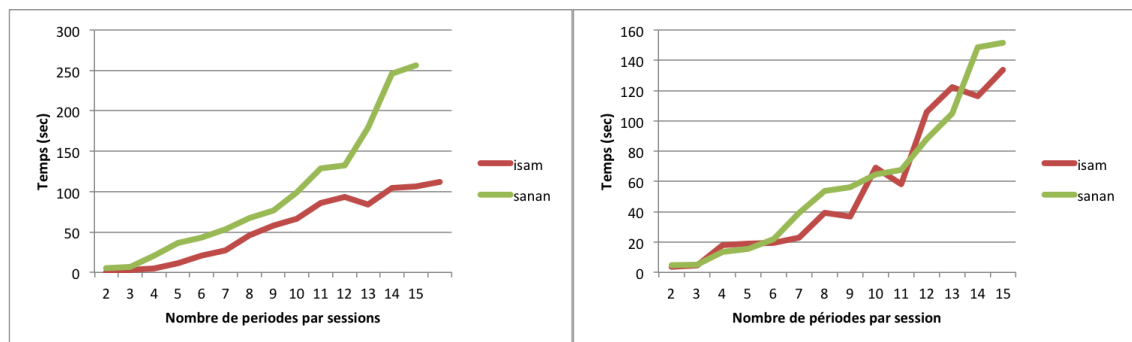


(a) Solution initiale CPLEX

(b) Solution initiale Gloutonne

FIGURE 12 – Variation du nombre de sessions

**Variation du nombre de périodes par session** Ici, nous avons observé l'influence du nombre de périodes par session sur le temps de réponse (Figure 13).



(a) Solution initiale CPLEX

(b) Solution initiale Gloutonne

FIGURE 13 – Variation du nombre de périodes par session

Les graphiques de la figure 13 montrent que ce paramètre augmente le temps de réponse de manière raisonnable.

**Variation du nombre de tables** Dans ce paragraphe, nous avons analysé l'impact du nombre de tables sur le temps de calcul (Figure 14).

Il s'avère que le temps de réponse est maximal lorsque l'on a un nombre de tables égal à 35% du nombre de personnes. Ceci est assez contraire au résultat obtenu avec la résolution CPLEX, où le temps minimal de réponse est enregistré quand le nombre de tables est égal à environ 40% du nombre de personnes.

**Variation du nombre maximal de rendez-vous successifs** Pour ce dernier paragraphe, nous exposons les résultats obtenus lors de la variation du nombre de rendez-vous successifs (Figure 15).

On peut voir que le nombre maximal de rendez-vous successifs influence le résultat de manière complètement différente.

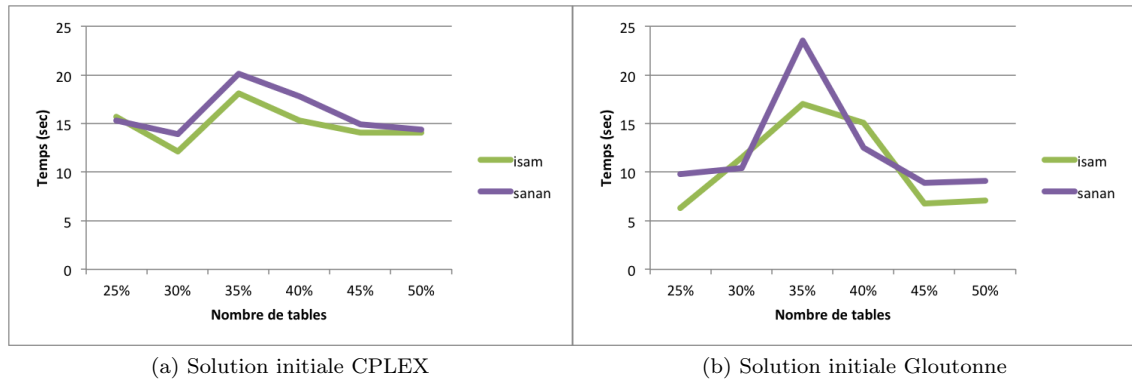


FIGURE 14 – Variation du nombre de tables

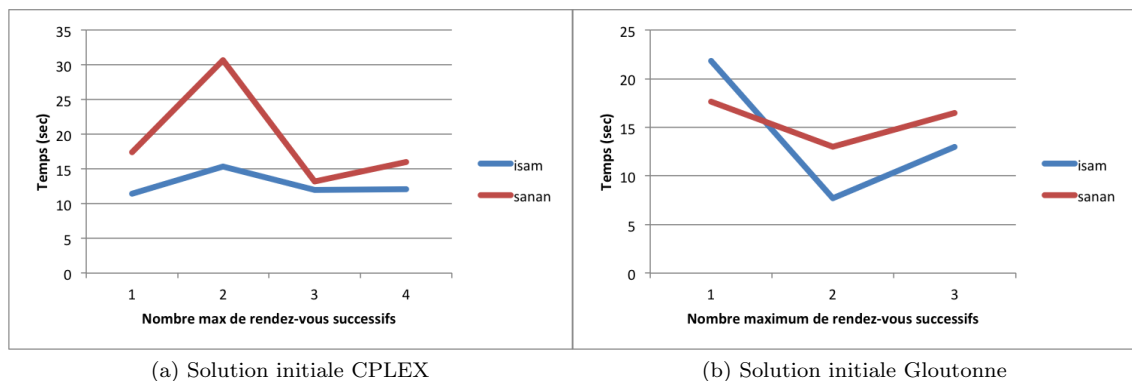


FIGURE 15 – Variation du nombre max de rendez-vous successifs

#### 4.4 Comparaison CPLEX/Tabou

Grâce aux résultats obtenus, on peut voir que les solutions fournies par la méthode tabou sont moins sensibles au nombre de personnes. En effet, les courbes de l'instance « isam » et « sanan » sont, en général, assez proches. Nous voulions effectuer le même genre de test pour voir l'impact du nombre de personnes ou encore celui du nombre de préférences émises sur le temps de résolution. Cependant, nous ne disposons que de deux instances peu différentes. Une génération aléatoire ne reflèterait pas suffisamment la réalité et donnerai donc des résultats erronés.

Pour pouvoir analyser la qualité des solutions fournies par la méthode tabou, nous avons décidé d'analyser les valeurs de la fonction objectif et les temps d'exécution. Nous avons donc fait une résolution avec CPLEX pour obtenir la valeur optimale du problème pour pouvoir positionner les résultats de la méthode tabou. Ainsi, nous avons pu observer que les solutions de tabou sont très proches de la valeur optimale (moins de 5% de dégradation) pour des gains de temps d'environ 90%.

Un échantillon des tests en fourni dans la figure 16. Les informations entre parenthèses correspondent au type de solution initiale (C : CPLEX, G : Glouton) suivit du nombre d'itérations maximal sans amélioration.

Nous pouvons voir sur cet échantillon, que les solutions de tabou avec la génération initiale gloutonne sont dominées par la résolution de tabou avec la génération initiale CPLEX et 500

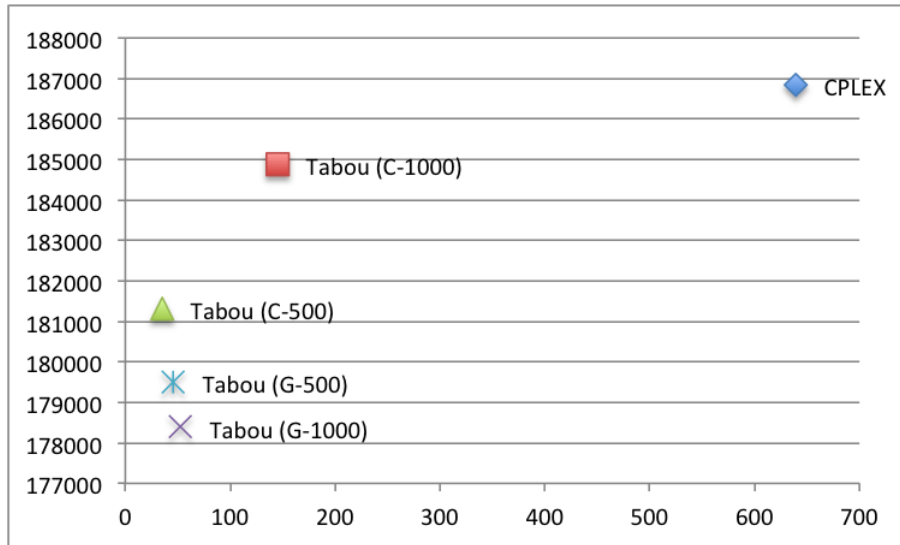


FIGURE 16 – Qualité des solutions obtenues

itérations sans amélioration. De manière générale, nous avons remarqué que le fait d'utiliser la génération initiale provenant de CPLEX donne des meilleurs résultats qu'avec la génération initiale gloutonne.

## 5 Conclusion et perspectives

### 5.1 Conclusion

Notre sujet de TER nous a permis d'étudier le problème de la planification de rendez-vous. Ce type de problème est rencontré dans plusieurs applications réelles telles que les rendez-vous de type speed-dating, des rendez-vous organisés dans le cadre d'un salon de l'emploi ou encore des rendez-vous planifiés dans des bourses d'échanges de technologie. Si la planification de tels rendez-vous est possible manuellement, elle peut s'avérer longue et difficile avec un nombre important de participants.

Malgré le nombre d'applications dans lesquelles ce problème est rencontré, il a encore fait l'objet de peu de travaux. Nous avons présenté dans notre état de l'art les principaux modèles et outils pertinents traitant de ce problème ainsi que des problèmes connexes.

Pour la résolution de ce problème, nous avons proposé deux méthodes de résolutions : une méthode de résolution exacte, adaptée à des instances de petites tailles, basée sur un modèle d'optimisation existant et utilisant le solveur CPLEX, et une méthode de résolution approchée basée sur une méta-heuristique (recherche tabou), adaptée à des instances de plus grandes tailles.

Ce projet, qui nous aura permis d'avoir une première approche de la recherche et de nous confronter à un problème d'optimisation concret avec des données réelles, reste ouvert à de nombreuses perspectives de développement. Même si le logiciel permet actuellement de planifier efficacement des rendez-vous pour des instances de taille conséquente, il demande tout de même à être affiné, complété et généralisé.

### 5.2 Perspectives

Nous présentons ici quelques perspectives d'amélioration du logiciel "MeetMe". Ci dessous un listing de points potentiels à développer :

Optimisation séparée des objectifs : le problème étant intrinsèquement multi-objectifs, nous l'avons modélisé en traduisant certains objectifs sous forme de contraintes et en utilisant un unique objectif de synthèse. Cependant, il pourrait être intéressant d'étudier l'optimisation séparée de ces différents objectifs.

Aléas de dernière minute : le logiciel doit pouvoir adapter les plannings des participants en cas d'absence ou d'inscription tardive de l'un d'entre eux sans toutefois modifier l'ensemble des plannings déjà établis.

Rotation entre les tables : le nombre de déplacements entre deux sessions peut être minimisé pour faciliter les rotations entre tables et salles lors d'une manifestation.

Enfin, nous avons remarqué que le fait de mettre un maximum de rendez-vous au début des journées et le fait d'équilibrer les rendez-vous sur les périodes étaient peu compatibles. Il faudrait donc trouver un compromis permettant de pouvoir faire cohabiter ces deux objectifs ou alors de n'en garder qu'un seul.

## Références

- [1] Robert A. Russell and Timothy L. Urban. Multicriteria models for planning power-networking events. *European Journal of Operational Research*, 207 :83–91, April 2010.
- [2] Yen-Zen Wang. Using genetic algorithm methods to solve course scheduling problems. *Expert Systems with Applications*, 25 :39–50, 2003.
- [3] Ahlem BenHassine and Tu Bao Ho. An agent-based approach to solve dynamic meeting scheduling problems with preferences. *Engineering Application of Artificial Intelligence*, 20 :857–873, January 2007.
- [4] Thomas R. Rohleder and Kenneth J. Klassen. Using client-variance information to improve dynamic appointment scheduling performance. *OMEGA*, 28 :293–302, June 2000.
- [5] Hon Wai Andy Chun and Rebecca Y.M. Wong. Optimizing agent-based meeting scheduling through preference estimation. *Engineering Applications of Artificial Intelligence*, 16 :727–743, September 2003.
- [6] A. Tamilarasi and T. Anantha kumar. An enhanced genetic algorithm with simulated annealing for job-shop scheduling. *International Journal of Engineering Science and Technology*, 2 :144–151, 2010.
- [7] D. J. A. Welsh and M. B. Powell. An upper bound for chromatic number of a graph and its application to timetabling problems. *Oxford Journals*.
- [8] D. C. Wood. A technique for colouring a graph applicable to large scale timetabling problems.
- [9] Gunther Skele Jurgen Dorn, Mario Girsch and Wolfgang Slany. Comparison of iterative improvement techniques for schedule optimization. *European Journal Of Operational Research*, 94 :349–361, June 1996.
- [10] D. de Werra. The combinatorics of timetabling. *European Journal of Operational Research*, 96 :504–513, March 1997.
- [11] Xavier Gandibleux Xavier Delorme and Joaquin Rodriguez. Grasp for set packing problems. *European Journal of Operational Research*, 153 :564–580, January 2004.
- [12] Narendra Jussien Hadrien Cambazard, Fabien Demazeau and Philippe David. Interactively solving school timetabling problems using extensions of constraint programming. *Patate Conference*.
- [13] Tony Wong Pascal Côté and Robert Sabourin. Application of a hybrid multi-objective evolutionary algorithm to the uncapacitated exam proximity problem. *Patate Conference*.
- [14] F. Rossi M.S. Franzin, E.C. Freuder and R. Wallace. Multiagent meeting scheduling with preferences : efficiency, privacy loss and solution quality. *AAAI*, 2002.
- [15] Gwen Moran. Need for speed : Make more contacts with speed networking. *Entrepreneur*, 2004.
- [16] Ismel Brito and Pedro Meseguer. Distributed meeting scheduling. 2003.
- [17] D. Poncelet X. Gandibleux, B. Pajot. Modèle en variables 01 et système d’aide pour l’optimisation de l’organisation de bourses d’échanges de technologies. 2006.
- [18] Xavier Gandibleux et Brice Pajot. Modélisation du problème de planification de bourses d’échanges. 2005.
- [19] I. Gent and T. Walsh. Csplib : a problem library for constraints. <http://www.csplib.org>.
- [20] Les rendez-vous d’affaires. <http://www.rendezvousaffaires.fr/index.html>.
- [21] Varup :salon de l’entreprise. <http://www.varup.com/fr/rendez-vous-affaires.htm>.
- [22] Dovax : Visual andfast. [http://www.dovax.fr/index.php?page=produits&&top=0&&pagebas=visualandfast\\_modules](http://www.dovax.fr/index.php?page=produits&&top=0&&pagebas=visualandfast_modules).
- [23] Extreme networking. <http://extremenetworking.biz/demo/training7.htm>.
- [24] Contact25 : Speed networking events. <http://www.contact25.com/v4/>.

