

Programmation par contraintes pour l'ordonnancement d'un bloc opératoire

ARNAULD HANSET¹, DAVID DUVIVIER², OLIVIER ROUX¹, NADINE MESKENS¹

¹ LSM & FUCAM

151, Chaussée de Binche, 7000 Mons, Belgique
{arnauld.hanset, olivier.roux, nadine.meskens}@fucam.ac.be

² LABORATOIRE D'INFORMATIQUE DU LITTORAL

50 rue Ferdinand Buisson, BP 719, F62228 Calais Cedex
david.duvivier@lil.univ-littoral.fr

Résumé – L'ordonnancement du bloc opératoire est devenu un enjeu de taille dans la bonne gestion d'un hôpital. Aussi n'est-il pas étonnant de constater que la littérature du domaine regorge d'approches basées sur des modèles mathématiques et des métaheuristiques. Cependant, nous recherchons ici un modèle qui tienne compte d'un maximum de contraintes sous-jacentes au problème d'ordonnancement du bloc opératoire. Plus précisément, il s'agit de contraintes humaines (préférences du personnel) et matérielles (disponibilité de matériel renouvelable et non-renouvelable) qui réduisent l'espace des solutions réalisables. Le jeu de données utilisé pour la réalisation de notre ordonnancement est issu d'un hôpital belge. Notre approche se veut générique et pourra être utilisée comme élément de base d'une solution d'ordonnancement s'adaptant facilement aux contraintes de chaque hôpital.

Abstract – The scheduling of operating theater is critical for an efficient management of each hospital. So it is not surprising to notice that the literature of the domain abounds in approaches based on mathematical models and metaheuristics. However, in this paper our aim is to build a model which is able to take a maximum of real life constraints into account when considering this scheduling problem. More precisely, it comprises human aspects (preferences, skills) and material constraints (availability of renewable and not renewable resources). The data set related to the experimental part of this article came from a belgian hospital. However our approach wants to be generic and to form an efficient building-bloc for an effective scheduling solution which can integrate several components corresponding to the constraints of the target hospital(s).

Mots clés – bloc opératoire, ordonnancement, programmation par contraintes

Keywords – operating rooms, scheduling, constraint programming

1 INTRODUCTION

L'optimisation de la gestion du bloc opératoire est plus que jamais d'actualité. En effet, le bloc opératoire est au cœur de l'hôpital et c'est à son rythme que battent les autres services ainsi que le personnel hospitalier. Si la gestion du bloc s'effectue de manière optimale, le personnel hospitalier ne devra pas effectuer d'heures supplémentaires et ne sera moins soumis au stress. C'est la qualité du travail qui est donc récompensée et la qualité des soins des patients qui s'en trouve améliorée. Il est difficile de chiffrer toutes les répercussions que peut entraîner un ordonnancement mal agencé. Les gestionnaires hospitaliers semblent s'accorder pour dire que quinze pourcents seraient une assez bonne estimation du coût du bloc opératoire par rapport à l'ensemble de l'hôpital [Clergue, 1999 et Macario *et al.*, 1995].

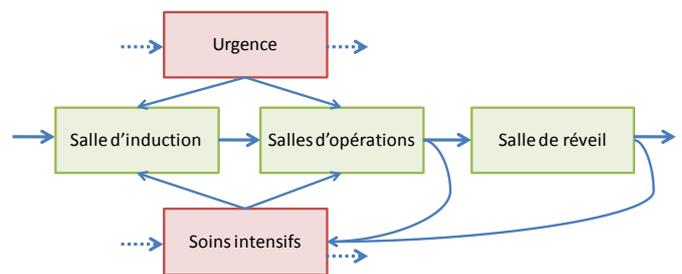


Figure 1. Flux du bloc opératoire

Le bloc opératoire se compose de salle d'induction, de salles d'opérations et d'une salle de réveil que le patient rejoint normalement après son opération avant de remonter dans son lit d'hospitalisation. Notons aussi qu'à tout moment un patient peut provenir des soins intensifs ou des urgences pour rejoindre la salle d'induction ou une salle d'opérations (Figure 1). La gestion du bloc, si elle veut être efficace, doit tenir compte d'une multitude de contraintes. Tout d'abord, il y a les contraintes humaines : un minimum d'effectif doit être présent durant l'opération, les heures de travail du personnel doivent être respectées (que ce soit celles des chirurgiens, des

anesthésistes, des infirmières et des brancardiers...). Au-delà de ces contraintes, nous pouvons aussi mentionner la nécessité de tenir compte de contraintes de préférences de travail en équipe, soit par pool de service soit par duo ou trio (chirurgien-anesthésiste-infirmière) ou encore par plages horaires (par exemple, un chirurgien ne travaillant que certaines demi-journées). Ensuite, surviennent les contraintes matérielles telles que le nombre de salles et leur polyvalence, la disponibilité de matériel spécifique...

Les contraintes à intégrer sont très nombreuses et leur nombre croît proportionnellement au nombre de personnels hospitaliers, de salles et d'opérations. Ce type de problème d'optimisation très fortement contraint entre parfaitement dans le cadre d'application de la programmation par contraintes.

Nous considérons, dans le cadre de cet article, l'ordonnancement des opérations sur une journée. Ce problème consiste à décider de l'ordre de passage des interventions ayant été affectées à une journée par le niveau de planification dans les salles d'opérations en prenant en compte la disponibilité des chirurgiens. Dans un premier temps, nous ne traitons pas l'ensemble des contraintes mais considérons les plus importantes pour obtenir un programme opératoire sur une journée. Nous émettons une série d'hypothèses simplificatrices pour faciliter l'écriture du cœur de notre modèle. Ce modèle se formalise sous la forme d'un problème de type « open-scheduling ». De plus, nous ne prenons en compte ni la disponibilité des lits de réveil, ni les contraintes matérielles qui restreignent la disponibilité des salles.

Bien conscient que ce modèle ne constitue que la première phase de la modélisation pour l'ordonnancement d'un bloc opératoire générique par une approche de programmation par contraintes, nous voulons, ici, tester l'adéquation de cette approche au problème considéré et par après comparer à d'autres approches telle que la méthode tabou [Hanset *et al.*, 2007].

La suite de cet article se compose de quatre parties : dans la première partie, nous présentons un état de l'art synthétique, suivi d'une description de la programmation par contraintes, des particularités de cette méthode et d'une méthode de résolution. Ensuite, nous donnons la description du modèle et les notations utilisées. Puis dans une quatrième partie, les résultats des expérimentations numériques effectuées sur le jeu de données réelles seront présentés. Cet article se termine par les conclusions et perspectives.

2 ETAT DE L'ART

Dans un bloc opératoire, le patient peut suivre différentes trajectoires. Ces dernières définissent les ressources nécessaires. La trajectoire la plus courante est constituée de la séquence suivante : salle d'induction, salle d'opérations, salle de réveil, lit du patient. Toutefois, dans la littérature, certains auteurs s'intéressent plus particulièrement au fait qu'à tout moment un patient peut provenir des soins intensifs ou des urgences pour rejoindre la salle d'induction ou une salle d'opérations (Figure 1). De la même manière, un patient peut quitter la salle d'opérations ou la salle de réveil pour rejoindre l'unité de soins intensifs.

Les processus opératoires se décomposent en trois phases :

1. La phase pré-opératoire durant laquelle le patient subit des consultations chirurgicales et anesthésiques. Elle s'étend de la prise en charge du patient jusqu'à la veille de l'intervention.
2. La phase per-opératoire proprement dite qui s'étend de la préparation psychique du patient, avant l'intervention, jusqu'à ce qu'il se réveille et quitte la salle réveil. Cette phase a lieu le jour de l'intervention. Ce jour-là, les patients sont d'abord anesthésiés et transférés par les brancardiers aux salles d'opérations, où les équipes chirurgicales vont les opérer. Après avoir été opérés, ils sont transférés à la salle de réveil et y restent jusqu'au moment où les anesthésistes les autorisent à retourner dans leurs services d'hospitalisation ou dans les secteurs de soins intensifs et de réanimation. Cette phase est la partie la plus importante du processus opératoire.
3. La phase post-opératoire : après son réveil, le patient est transféré vers son service d'hospitalisation. Cette phase couvre l'ensemble des soins nécessaires suite à l'intervention. Dans le cas où l'état du patient serait jugé critique, il sera plutôt conduit vers les soins intensifs et de réanimation.

Toutes ces étapes nécessitent des ressources matérielles et humaines. Chaque ressource prise en compte dans un modèle de gestion de bloc opératoire, entraîne un accroissement de sa complexité. C'est pourquoi la majeure partie des problèmes décrits dans la littérature se focalise sur un aspect de la problématique et pose certaines hypothèses conduisant à diminuer la complexité du problème. Citons quelques-unes des contraintes les plus courantes : durée d'ouverture des salles d'opérations, disponibilité des chirurgiens, des anesthésistes, des infirmières, des équipements chirurgicaux, des lits de réanimation, des lits de réveil, nombre de lits de réveil, polyvalence des salles d'opérations, qualification du personnel.

La gestion du bloc opératoire est rendue d'autant plus difficile du fait de perturbations diverses [Hammami, 2007]. Premièrement, les aléas affectent les durées des activités, la trajectoire du patient [Augusto, 2007], le nombre et le type d'interventions. Deuxièmement, les impondérables principalement liés aux modifications de dernières minutes apportées par les chirurgiens, les complications liées à l'état du patient, l'arrivée d'urgences, les pannes de matériels. Dans les hôpitaux belges ciblés pour notre étude, les gestionnaires gèrent ces aléas et impondérables en adaptant manuellement les programmes établis préalablement.

Les ressources les plus souvent prises en compte dans les problèmes de planification et d'ordonnancement concernent les chirurgiens et les salles d'opérations (nombre et durée d'ouverture). Les programmes opératoires proposés tiennent compte des plages horaires attribuées aux chirurgiens (ou à leur(s) spécialité(s)) ainsi que du nombre d'interventions quotidiennes autorisées pour chaque chirurgien. Citons [Guinet et Chaabane 2003] ; [Chaabane *et al.*, 2007] ; [Marcon et Dexter, 2007] ; [Hammami *et al.*, 2007] ; [Dexter *et al.*, 2002] ; [Fei *et al.*, 2007, 2008].

Belien et Demeulemeester [Belien et Demeulemeester 2007b] limitent la disponibilité journalière du nombre de salles d'opérations. Jebali *et al.* [Jebali *et al.*, 2006] quant à eux prennent en compte, dans la résolution d'un problème

d'ordonnement, la disponibilité de certains équipements spécifiques (le type d'équipement requis dans les salles pour certaines opérations).

La contrainte relative à la disponibilité d'un lit de réveil pour qu'un patient puisse quitter la salle d'opérations est prise en compte dans la résolution du problème de planification – ordonnancement de Fei *et al.* [Fei *et al.*, 2006]. Dans pareille situation, le patient peut commencer sa phase de réveil dans la salle d'opérations, ce qui allonge le temps d'occupation de la salle d'opérations. Rares sont les auteurs qui prennent aussi en compte la disponibilité des brancardiers [Augusto *et al.*, 2007] ; [Perdomo *et al.*, 2006] ; [Saadani et Guinet 2008].

Roland *et al.* [Roland *et al.*, 2006, 2007] ainsi que Chaabane *et al.* [Chaabane *et al.*, 2007] étudient la planification et l'ordonnement du bloc opératoire sur un cas réel. Ils incluent dans leur modèle des ressources renouvelables (infirmières, anesthésistes, etc.) en quantités finies et des ressources non-renouvelables attribuées à chaque opération (médicaments, certains matériaux chirurgicaux, plateaux médicaux stériles (le temps nécessaire à la remise à disponibilité du plateau est supérieur à une journée)). Les ressources non-renouvelables sont des ressources consommables en quantités finies sur une journée (*i.e.* notre horizon d'ordonnement).

Dans toutes ces études, trop souvent, les anesthésistes et le personnel infirmier sont supposés être en nombre suffisant. En dehors de quelques cas où cela se justifie, cette hypothèse est bien trop simplificatrice. De même que le chirurgien ne peut être qu'à un seul endroit à la fois, l'anesthésiste lui aussi ne peut être présent que dans une salle d'opérations à la fois. Rares sont les auteurs qui en font état [Chaabane *et al.*, 2007]. Roland *et al.* [Roland *et al.*, 2007] envisagent toutefois la possibilité que les infirmières ou les anesthésistes puissent quitter la salle d'opérations avant la fin de l'opération pour en commencer une autre.

L'affectation des infirmières est un problème en soi car elle nécessite la prise en compte de la charge de travail [Belien *et al.*, 2007a]. Même sans en tenir compte, la confection des plannings pour les infirmières est tout aussi complexe car bon nombre de contraintes sont liées à leur seule fonction.

Le problème de réalisation de plannings pour les infirmières n'est pas spécifique au bloc opératoire. Effectivement, la majeure partie des infirmières d'un hôpital travaille dans d'autres services ou d'autres départements. Mais toutefois les contraintes sont semblables pour le bloc opératoire. Ces contraintes sont : la réglementation légale, les objectifs de l'hôpital, les préférences des infirmières, les qualifications de celles-ci, la taille d'une équipe ou d'un service, le nombre de jours consécutifs de travail, le nombre de nuits consécutives, éviter les coûts supplémentaires de sur-utilisation, établir des gardes de nuits et des week-ends, assurer une formation continue, pourvoir à l'absentéisme, gérer les demandes de congés...

Dans la littérature, nous constatons l'existence de quatre problèmes distincts liés à la problématique d'ordonnement du bloc opératoire : affectation de ressources, planification, ordonnancement, et des problèmes d'emploi du temps (construction de planning et time-tabling). Il s'avère toutefois

que pour obtenir un ordonnancement efficace, il est nécessaire de tenir compte de ces problématiques interdépendantes.

Dans leur article, Azaiez et Sharif [Azaiez et Sharif, 2005] visent plus particulièrement les objectifs suivants : éviter les jours de travail isolés, tenir compte de jours de préférence, obtenir le même nombre d'heures de garde pour chaque infirmière, avoir plus de jours que de nuits de travail, interdire les séquences nuit-jour en raison du non respect du repos journalier. Jaumard *et al.* [Jaumard *et al.*, 1998] tiennent compte de différents niveaux de qualification du personnel et d'une possible rotation de celui-ci aux différents postes pour assurer aux infirmières le maintien de leur qualification par une pratique suffisante dans chaque discipline.

Le lecteur intéressé par plus de détails pourra trouver un état de l'art plus complet dans [Hanset *et al.* 2008]. A ce stade, nous nous intéressons à la problématique d'ordonnement en gardant à l'esprit que notre modèle sera étendu avec l'une ou l'autre problématique susnommée.

En dehors du domaine médical, il existe plusieurs recherches présentant divers modèles mathématiques et algorithmes pour la planification et l'affectation du personnel. Au vu de toutes les contraintes susnommées, nous nous sommes dirigés vers la programmation par contraintes. La construction de planning est une application de la programmation par contrainte. Pour mieux comprendre les avantages de la programmation par contraintes dans notre contexte, expliquons avant tout les quelques principes de base de celle-ci.

3 PROGRAMMATION PAR CONTRAINTES

3.1 Principes de la modélisation

Plusieurs principes sont à la base de la programmation par contraintes [Krzysztof R. 2003]. Ceux-ci nous permettent de décrire et de justifier notre démarche.

Principe 1: La programmation par contraintes est à la fois la formulation d'un problème de satisfaction de contraintes et sa résolution par réduction de domaines spécifiques ou par des méthodes appropriées.

Ce principe nous éclaire sur plusieurs faits. Premièrement, bien conscient qu'un problème puisse être résolu de différentes manières, ici en associant à un problème une manière de le résoudre, nous définissons un programme par contraintes. Deuxièmement, un problème est défini par le(s) domaine(s) que parcourent ses variables et en scindant un domaine, nous obtenons un autre problème.

Principe 2 : Plusieurs méthodes de résolution par contraintes peuvent être naturellement expliquées par un travail basé sur des règles. La résolution par contraintes consiste alors à trouver un ensemble de règles qui spécifie le comportement du problème et par conséquent sa résolution. Ce point de vue établit les connexions entre la programmation basée sur des règles et la programmation par contraintes.

Cela explique notre décision de décrire les solveurs par contraintes au moyen de règles. Des règles vont guider la recherche pour l'obtention d'une solution réalisable.

Principe 3: les algorithmes de propagation de contraintes peuvent être naturellement expliqués comme des algorithmes itératifs où, à chaque étape, une variable est instanciée.

Cette vue nous permet de clarifier la nature des algorithmes de propagation de contraintes. Ainsi, ils nous fournissent de manière naturelle une méthode d'implémentation des solveurs par contraintes, du fait qu'une règle du solveur n'est autre qu'une itération supplémentaire de l'algorithme d'instanciation.

Principe 4: les techniques de recherche peuvent être vues conceptuellement comme des algorithmes de recherche au sein des arbres.

Nous sommes face à une génération de règles qui peuvent être décrites sous la forme d'un arbre. Nous devons alors mettre au point des techniques de recherche au sein de ces arbres pour optimiser la recherche d'une ou plusieurs solution(s).

3.2 Principes de résolution

Un problème de satisfaction de contraintes est entièrement décrit par l'énoncé du problème et par ses variables (les domaines pour chacune des variables).

Pour résoudre un problème de satisfaction de contraintes (PSC) nous le transformons plusieurs fois de manière spécifique. Chacune des transformations engendre un(des) nouveau(x) problème(s) plus petit(s). Chacun de ces nouveaux problèmes va nous rapprocher d'une solution, respectivement, toutes les solutions. Une fois résolu, le PSC « final » (*i.e.* une feuille dans l'arbre des contraintes) mène à une solution dans le cas où tous les domaines sont réduits à des singletons, ou à plusieurs solutions dans le cas où les domaines ont plus d'un élément. Les transformations du PSC doivent s'opérer de sorte que l'équivalence (au sens approprié de la programmation par contraintes [Krzysztof R. 2003]) entre deux problèmes soit conservée.

Plusieurs transformations existent, la plus répandue est le découpage (split). Un découpage est obtenu en scindant le domaine d'une variable et donc *a priori* les contraintes associées. Cette transformation remplace notre PSC courant par deux ou plusieurs PSC qui diffèrent du PSC courant par le domaine d'une variable qui a été découpé ou instancié à une valeur.

Nous supposons que la variable x_i soit instanciée à la valeur a_k où k est un indice parcourant tout le domaine de la variable sélectionnée. Alternativement, cette règle est répétée et paramétrée par k . Elle nous conduit au remplacement d'un PSC par k PSCs.

La procédure de découpage engendre deux ou plusieurs nouveaux PSCs. Plusieurs questions surviennent : Quelles variables faut-il sélectionner ? Quelles valeurs sont choisies ou quelles contraintes sont découpées ? Plusieurs heuristiques existent. Par exemple, sélectionner la variable qui est la plus contraignante ou, pour les domaines réels, les diviser en leur milieu.

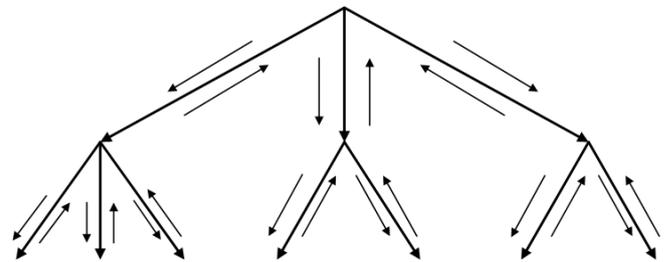


Figure 2. Recherche par « Backtracking » (Krzysztof, 2003)

Pour un arbre fini donné, les deux techniques les plus connues pour parcourir celui-ci sont la recherche en profondeur d'abord ou « backtracking » (Figure 2) pour la recherche d'une solution réalisable et, quand une solution optimale est recherchée, le « branch and bound » (Figure 3). Si nous sommes intéressés par la recherche d'une solution réalisable, la recherche en profondeur d'abord stoppe dès qu'une feuille est rencontrée, ce qui résout le PSC. La recherche par « branch and bound » prend en compte la valeur d'une fonction objective à la recherche d'une solution optimale vis-à-vis de cette fonction objective. Celle-ci est utilisée, chaque fois que la méthode descend dans les branches de l'arbre, pour déterminer les branches qui seront explorées et celles qui ne le seront pas si elles ne comportent pas de solutions meilleures que la solution courante. Une fois l'arbre parcouru, nous trouvons une solution optimale (ou nous avons la preuve de l'absence de solution).

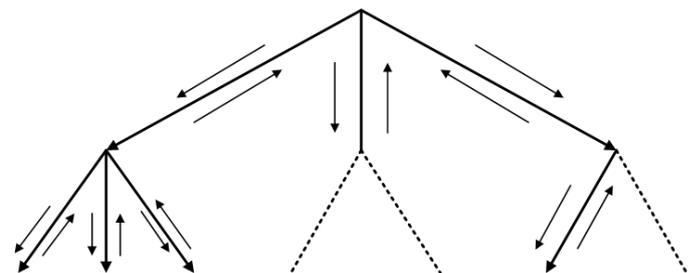


Figure 3. Recherche par « Branch and bound » (Krzysztof, 2003)

Les avantages qui nous ont conduits à la programmation par contraintes sont, premièrement, que la représentation des PSCs est proche des problèmes originaux (description intuitive des contraintes à respecter), deuxièmement elle simplifie la phase de programmation et facilite sa maintenance.

Nous allons maintenant décrire les hypothèses qui nous ont menées à l'élaboration de notre modèle mathématique.

4 MODELE MATHEMATIQUE

4.1 Hypothèse réductrice

Nous nous intéressons dans cet article à assigner des opérations dans un bloc opératoire. L'ordonnement des salles d'opérations est allégé de plusieurs contraintes matérielles et humaines pour définir un modèle de base en programmation par contraintes, modèle que nous pourrons ensuite enrichir itérativement à mesure de l'intégration de nouvelles contraintes. Nous devons émettre des hypothèses simplificatrices pour décrire un premier modèle qui ne tienne compte que des éléments nécessaires pour établir un ordonnancement sur une journée.

Ces hypothèses sont les suivantes :

- toutes les opérations sont affectées à un chirurgien à l'avance ;
- toutes les ressources matérielles et humaines sont disponibles en nombre suffisant exceptés les chirurgiens ;
- toutes les salles d'opérations sont disponibles simultanément en début de journée dès huit heures et sont disponibles pour huit heures consécutives (en temps normal) ;
- tous les patients sont prêts à être opérés, nous ne tenons pas compte d'un temps de préparation de ceux-ci, ni d'éventuels trajets vers les étages ;
- dans ce modèle, nous ne prenons pas en compte la capacité restreinte en lits de réveil, nous comptons développer notre modèle dans cette voie ;
- les cas des urgences ne sont pas pris en compte, ni les aléas ;
- une fois qu'une opération est commencée, elle n'est pas interrompue ;
- les temps de préparation de salles, qui consistent en temps de nettoyage et en réapprovisionnement du matériel nécessaire, sont intégrés dans les temps opératoires ;
- nous ne tenons pas compte des ressources matérielles telles que les plateaux médicaux stériles qui constituent le matériel de base des opérations et nécessitent un long traitement de stérilisation entre deux usages.

4.2 Modèle mathématique

Nous avons découpé la journée en T intervalles de temps de dix minutes. La granularité temporelle est assez proche de la réalité, sans augmenter considérablement la taille de l'espace de recherche. Il y a un ensemble de R salles d'opérations. Chacune des salles se verra donc attribuer T intervalles de temps, qui correspondront à des périodes disponibles pour les chirurgiens.

Aussi, pour un même intervalle de temps, nous définissons l'occupation des salles au moyen de R variables booléennes. Nous devons placer une à une les interventions, chacune étant décrite par un certain nombre d'intervalles qu'elles occuperont dans une salle (nous supposons que chaque opération o dure $d(o)$ intervalles de temps). Au sein d'une salle r , la position de l'opération sera renseignée par la présence de $d(o)$ variables booléennes successives indiquant l'exécution de l'opération o dans cette salle. Nous notons que seules les variables booléennes relatives à l'opération pour la salle en question et pour les intervalles de temps occupés par l'intervention vaudront un et que partout ailleurs elles vaudront zéro. De cette manière, nous travaillons avec trois dimensions (R salles, O opérations et T intervalles de temps) de variables booléennes. Cette matrice est représentée par $OTR(o,t,r)$.

Le premier ensemble de contraintes que nous pouvons exprimer facilement sont que deux opérations ne peuvent pas avoir lieu en même temps dans une même salle d'opérations :

$$\sum_{o \in O} OTR(o,t,r) \leq 1 \quad \forall r \in R, \forall t \in T \quad (1)$$

De plus, nous avons une correspondance exacte entre chaque opération et son chirurgien. Nous avons un ensemble de S chirurgiens. Nous connaissons donc l'ensemble de toutes les opérations d'un chirurgien. Chacun ne pouvant opérer deux opérations en même temps dans des salles d'opérations différentes. Nous pouvons écrire :

$$\sum_{r \in R} \sum_{o \in S} OTR(o,t,r) \leq 1 \quad \forall t \in T, \forall s \in S \quad (2)$$

Ces deux ensembles de contraintes ((1) et (2)) impliquent par déduction que deux chirurgiens ne peuvent pas opérer dans le même intervalle de temps dans la même salle d'opérations.

Nous devons exprimer le fait que chaque opération o doit être affectée à une salle r et une seule et se déroule sur $d(o)$ intervalles de temps consécutifs :

$$\sum_{r \in R} \sum_{j=1}^{(T-d(o)+1)(j+d(o)-1)} \left[\frac{OTR(o,i,r)}{d(o)} \right] = 1, \quad \forall o \in O, j \in T \quad (3)$$

De cette manière, nous exprimons le fait que le quotient soit entier et ne prenne la valeur « 1 » qu'une seule fois (zéro dans les autres cas).

Dans le but de rajouter facilement des contraintes sur les chirurgiens, nous développons par extension de $OTR(o,t,r)$ la matrice $STR(s,t,r)$. La matrice $STR(s_i,t,r)$ exprime que le chirurgien i opère à certains moments dans les salles d'opérations. Nous devons écrire l'ensemble de contraintes pour établir la correspondance entre la matrice OTR et la matrice STR :

$$STR(s,t,r) = \sum_{o \in S} OTR(o,t,r) \quad \forall r \in R, \forall t \in T, \forall s \in S \quad (4)$$

Depuis (1), (2) et (4), nous pouvons exprimer facilement et de façon redondante à l'aide de la matrice STR qu'un chirurgien ne peut pas opérer en même temps dans deux salles différentes via les contraintes (5). Nous présentons ces contraintes par soucis d'exhaustivité, mais en raison des redondances susnommées, ces dernières ne sont pas implémentées dans notre modèle.

$$\sum_{r \in R} STR(s,t,r) \leq 1 \quad \forall t \in T, \forall s \in S \quad (5)$$

Bien que notre solveur soit paramétré en mode « backtracking » à la recherche d'une solution réalisable, nous utilisons une fonction objectif afin d'évaluer les solutions obtenues. La fonction objectif utilisée représente l'heure de fin de la dernière opération (makespan) se déroulant dans le bloc opératoire. Celle-ci nous permet de chiffrer l'efficacité de nos solutions.

$$\max(t \cdot OTR(o,t,r)) \quad \forall r \in R, \forall t \in T, \forall o \in O \quad (6)$$

Dans la partie suivante, nous mettons en application les différentes équations développées dans ce modèle. Le jeu de données utilisé par la suite est un jeu qui a déjà fait l'objet de tests préalables et pour lequel nous savons qu'au moins une

solution réalisable existe respectant les heures d'ouvertures du bloc opératoire.

5 RESULTATS

Le modèle développé ci-dessus a été programmé en Java (JDK1.6) sur un processeur Pentium IV (3,2 GHz), 1,5 Go de RAM sous Windows XP.

La librairie CHOCO (2.0.0.2) [Rochart *et al.*, 2008] utilisée pour notre programme est une librairie développée par l'Ecole des Mines de Nantes dans le but d'écrire des problèmes sous contraintes. Cette librairie a l'avantage d'être un logiciel libre.

La caractéristique principale du langage est que sa programmation doit se faire de manière dynamique indépendamment de la taille des variables et donc du domaine sur lequel elle est définie. Notons que si les opérateurs classiques (arithmétiques et logiques) sont présents, l'expression des contraintes est particulière en ce sens qu'il faut être exhaustif sur tout le domaine de la variable.

Le jeu de données utilisé pour notre expérimentation est issu du monde réel. Une collaboration avec un hôpital belge nous a permis de récolter des données opératoires sur une année complète. Ces données nous renseignent sur les temps d'entrée en salle, de début de chirurgie, de fin de chirurgie, de sortie de salle, de la pathologie du patient, du (ou des) chirurgien(s) qui a (ont) pratiqué(s) l'opération, et de l'anesthésiste. Pour l'instant, comme décrit dans le modèle, nous ne nous intéressons qu'au temps en salle et au chirurgien qui a opéré. Dans ce jeu de données, nous considérons six salles d'opérations.

Utilisé en mode « backtracking », les résultats fournis par le solveur correspondent à nos attentes : rapides et efficaces. Les opérations sont assignées aux salles d'opérations sans chevauchement au sein d'une même salle. Les chirurgiens opèrent les cas qui leurs sont assignés et n'opèrent pas plus d'une opération à la fois.

L'aspect « backtracking » est visible lorsque nous analysons une solution, car les variables qui indiquent qu'une opération a lieu dans une salle (traduit par plusieurs 1 consécutifs parmi les 0 dans la matrice $OTR(o,t,r)$) sont les dernières variables dans l'énumération de cette solution. C'est-à-dire que la procédure de « backtracking » a bien commencé par instancier les dernières valeurs et est revenue en arrière dans l'arbre qui lui était fourni pour instancier les variables les plus proches.

Dans notre cas, les premières opérations sont assignées en fin de journées dans la sixième salle. Nous avons constaté qu'en inversant l'ordre des colonnes de notre matrice, cela nous permet de replacer celle-ci en début de journée. Etant donné que nous sommes sûrs de placer toutes les opérations, nous pouvons imaginer que la fonction économique se dégrade au fur et à mesure que nous occupons plus de salle et/ou que nous occupons trop peu celles-ci.

Pour l'instant, afin de valider rapidement les contraintes constituant notre modèle, nous nous limitons à la génération d'une solution réalisable. C'est-à-dire que nous parcourons notre arbre de contraintes en profondeur d'abord (mode « backtracking ») à la recherche de la première solution réalisable. Dans une seconde étape, pour explorer l'arbre des contraintes en mode « branch-and-bound » nous devons lui

adjoindre la fonction objectif qui guide le parcours dans l'arbre. Notre but étant d'insérer un maximum de contraintes dans notre modèle, nous sommes parfaitement conscients du fait que l'espace de solutions réalisables se restreindra au fur et à mesure de leur intégration dans le modèle, complexifiant d'autant la recherche. Cependant, si le temps de recherche d'une solution optimale devient rédhibitoire, nous avons toujours la possibilité d'utiliser le solveur en mode « backtracking » à la recherche d'une solution réalisable (mais non nécessairement optimale).

Compte tenu des résultats obtenus et de la rapidité du solveur, nous envisageons d'ajouter des contraintes supplémentaires telles que des contraintes de temps de set-up ainsi que des périodes dans lesquelles les opérations sont interdites (pour des opérations de maintenance par exemple). De la même manière, un « block-scheduling » avec des plages propres aux chirurgiens, reviendrait à interdire certaines heures d'ouverture.

6 CONCLUSION ET PERSPECTIVES

Dans cet article, nous avons développé un nouveau modèle qui décrit l'activité du bloc opératoire grâce à la programmation par contraintes. Ce modèle constitue la première phase de notre travail. Le but de nos travaux étant d'intégrer un maximum de contraintes, et de pouvoir par la suite comparer nos résultats aux travaux déjà réalisés.

Il est à noter que les préférences des chirurgiens sont souvent les contraintes prises en compte en premier lieu, car elles reflètent un travail en équipe, que ce soit des équipes constituées de duo chirurgien-anesthésiste ou chirurgien-infirmier. De plus, la question du planning hebdomadaire des infirmières est tout aussi importante. Les qualifications de celles-ci, les jours et nuits de gardes, le respect des pauses et des heures de travail supplémentaires accumulées sont autant de contraintes dont il faut tenir compte si nous voulons établir un modèle complet qui peut optimiser le fonctionnement du bloc opératoire.

Néanmoins, notre objectif n'est pas de traiter le problème de dimensionnement des salles, ni du personnel qu'il soit chirurgien, anesthésiste, infirmier ou auxiliaire.

Les résultats obtenus sont encourageants. Les préférences des chirurgiens via un « block-scheduling » pourraient être intégrées en interdisant certaines heures pour chaque chirurgien. Les perspectives sont multiples, parmi lesquelles :

- améliorer notre fonction économique pour nous diriger vers l'utilisation d'un « branch and bound ».
- intégrer dans notre modèle tout d'abord la gestion d'une salle de réveil constituée de plusieurs lits de réveil. Il est nécessaire de modéliser la salle de réveil car elle constitue souvent un goulot d'étranglement pour les patients en sortie de salle d'opérations.
- énumérer un maximum de contraintes humaines et matérielles, conformément à la finalité de notre modèle. Nous devons par la suite prendre en compte celles-ci dans notre modèle, qu'il s'agisse de contraintes liées à des ressources renouvelables ou non renouvelables.
- prendre en compte les préférences des chirurgiens et du planning hebdomadaire des infirmières dans la suite de notre application.

7 REFERENCES

- Azaiez, M.N., Al Sharif, S.S., (2005) A 0-1 goal programming model for nurse scheduling. *Computers and Operations Research*, 32, pp. 491-507.
- Augusto, V., Xie, X., Perdomo, V., (2007) Operating Theatre Scheduling With Limited Recovery Beds And Patient Recovery In Operating Rooms. *International conference on Industrial Engineering and System Management - IESM'07*, Beijing, Chine, 30 Mai – 2 Juin.
- Belien, J., Demeulemeester, E., (2007 a) A branch-and-price approach for integrating nurse and surgery scheduling. *European Journal of Operational Research*. 189(3), 16 Septembre 2007, pp. 652-668.
- Beliën, J., Demeulemeester, E. (2007 b) Building cyclic master surgery schedules with leveled resulting bed occupancy. *European Journal of Operational Research*, 176(2), 16 Janvier 2007, pp. 1185-1204.
- Chaabane, S., Meskens, N., Guinet, A., Laurent, M. (2007) Comparaison des performances d'une méthode de planification appliquée à plusieurs politiques de programmation opératoire. *Logistique et Management*, 2007, 15(1), pp. 17-26.
- Clergue, F., (1999), Gestion du bloc opératoire : Pourquoi une telle préoccupation ? *Informations cliniques en Anesthésie-Réanimation*, pp. 93-95.
- Dexter, F., Traub, R.D., (2002) How to schedule elective surgical cases into specific operating rooms to maximize the efficiency of use of operating room time. *Anesthesia and Analgesia*, 94, pp. 933-942.
- Fei, H., Duvivier, D., Meskens, N., (2006) Ordonnement journalier dans un bloc opératoire dans le cadre d'une stratégie "open scheduling", *Actes de la 3^{ème} conférence francophone en Gestion et Ingénierie des Systèmes Hospitaliers - GISEH'06*, pp. 615-622., Luxembourg, Gd-Duché de Luxembourg, 14-16 Septembre.
- Fei H, Meskens, N., Chu, C. (2007) An operating theatre planning and scheduling problem in the case of an 'open scheduling' strategy. *International conference on Industrial Engineering and System Management - IESM'07*, Beijing, Chine, 30 Mai – 2 Juin.
- Fei, H., Chu, C., Meskens, N., Artiba, A. (2008) Solving surgical cases assignment problem by a branch-and-price approach. *Int. J. Production Economics*, 112(1), pp. 96-108, Mars.
- Guinet, A., Chaabane, S., (2003) Operating theatre planning. *Int. J. Production Economics*, 85, pp. 69-81.
- Hanset, A., Duvivier, D., Meskens, N., (2007) Ordonnement des interventions chirurgicales par une recherche Tabou : Exécutions courtes vs longues. *Logistique et Transport LT'07*, Sousse, Tunisie, Septembre.
- Hanset, A., Meskens, N., Roux, O., Duvivier, D., (2008) Prise en compte des ressources humaines et matérielles dans la gestion du bloc opératoire : état de l'art. *Gestion Industriel des Systèmes Hospitaliers - GISEH'08*, Lausanne, Suisse, Septembre.
- Hammami, S., Ladet, P., Atidel, B.H.A., (2007) *Une programmation opératoire robuste*. *Logistique & Management*, Mai 2007, 15(1).
- Jaumard, B., Semet, F., Vorvor, T., (1998) A generalized linear programming model for nurse scheduling. *European Journal of Operational Research*, 107, pp. 1-18.
- Jebali, A., Hadj Alouane A. B., Ladet, P., (2006) Operating rooms scheduling. *Int. J. Production Economics*, 99, pp. 52-62.
- Krzysztof, R.A. (2003) Principles of constraint programming, Edition Cambridge University Press,
- Macario, A., Vitez T.S., Dunn, B., Mcdonald, T., (1995) Where are the costs in perioperative care? *Anesthesiology* 83 (6) pp. 1138-1144.
- Marcon, E., Dexter, F., (2007) An Observational Study of Surgeons' Sequencing of Cases and Its Impact on Postanesthesia Care Unit and Holding Area Staffing Requirements at Hospitals, *International anesthesia research society*, Juillet 2007, 105(1), pp. 119-126
- Perdomo, V., Augusto, V., Xiaolan, X., (2006) Operating Theatre Scheduling Using Lagrangian Relaxation. *Proceedings of International conference on Service Systems and Service Management - SSSM'06*, pp. 1234-1239, Troyes, France, 25-27 Octobre.
- Rochart, G., Jussien, N., Lorca, X., (2008) The CHOCO constraint programming solver, *CPAIOR'08 workshop on Open-Source Software for Integer and Constraint Programming (OSSICP'08)*, Paris, France, juin.
- Roland, B., Di Martinelly, C., Riane F., (2006) Operating Theatre Optimization : A resource-Constrained Based Solving Approach. *Proceedings of International conference on Service Systems and Service Management - SSSM'06*, pp. 443-448, Troyes, France, 25-27 Octobre.
- Roland, B., Di Martinelly, C., Riane, F., (2007) Scheduling Operating Theatre under Human Resources Constraints. *International Conference on Industrial Engineering and Systems Management - IESM'07*, Beijing, Chine, 30 Mai – 2 Juin.
- Saadani, N.H., Guinet, A., (2008), Heuristique pour l'ordonnement des interventions dans un bloc opératoire, *Gestion et Ingénierie des Systèmes Hospitaliers GISEH'08*, 8 pages, Actes CDROM, EPFL (Ecole Polytechnique Fédérale de Lausanne), Suisse, 4-6 Septembre