

Modèles mathématiques pour minimiser le temps d'attente des patients en ambulatoire

Simon Moulard, Yannick Kergosien, Pierre Desport

▶ **To cite this version:**

Simon Moulard, Yannick Kergosien, Pierre Desport. Modèles mathématiques pour minimiser le temps d'attente des patients en ambulatoire. 11e Conférence Francophone en Gestion et Ingénierie des Systèmes Hospitaliers (GISEH 2022), Jul 2022, Saint Étienne, France. hal-03737517

HAL Id: hal-03737517
https://hal.science/hal-03737517

Submitted on 25 Jul 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Modèles mathématiques pour minimiser le temps d'attente des patients en ambulatoire

Moulard Simon [1], Kergosien Yannick [1], Desport Pierre [1]
[1] Université de Tours, LIFAT EA 6300, CNRS, ROOT EMR CNRS 7002, Tours, France ;
<prénom.nom>@univ-tours.fr

Résumé. La prise en charge des patients en ambulatoire plutôt qu'en hospitalisation complète permet de réduire les coûts hospitaliers. Cette prise en charge entraîne une complexification de la planification quotidienne des activités de soin et demande la mise en place d'une organisation efficace. Dans ce papier, nous nous intéressons à un problème de planification journalier d'activités de soin en ambulatoire avec pour objectif de minimiser le temps d'attente des patients. Pour résoudre ce problème, nous proposons deux types de modèles mathématiques l'un basé sur la programmation linéaire, et l'autre sur la programmation par contraintes. Des expérimentations sont réalisées pour comparer les deux types de modèle et leurs variantes sur des données inspirées d'un cas réel de l'APHP.

Mots clés : modèles mathématiques, planification, ambulatoire, parcours patient.

Introduction

Depuis plusieurs années, les instances nationales mènent une politique visant à augmenter le nombre d'actes médicaux réalisés en ambulatoire dans les CHU et CHRU[1]. La prise en charge en ambulatoire offre une réduction de la charge de travail, des coûts d'hospitalisation et des risques sanitaires liés aux infections nosocomiales par rapport aux hospitalisations complètes en limitant la durée de séjour à l'hôpital. Ce type de prise en charge implique une densification des activités à réaliser sur une journée et nécessite une organisation coordonnée et efficace pour maintenir une bonne qualité de service et des bonnes conditions de travail. Cela se traduit par la résolution d'un problème combinatoire de coordination de ressources et de planification des activités de soins pour des patients qui ne sont présents qu'une journée. La conception et l'emploi d'outils d'aide à la décision pour résoudre ce problème permettraient d'alléger la charge de travail des gestionnaires et cadres orchestrant ces activités.

Cet article vise à étudier et proposer une solution au problème d'ordonnancement des activités de soin dans un service ambulatoire. La liste des patients et des activités à réaliser est supposée connue et établie au préalable par les gestionnaires. Il s'agit alors d'ordonnancer les différentes activités à réaliser sur une journée en coordonnant les différentes ressources. Cette étude étant dans une phase préliminaire, nous nous intéressons dans cet article à une résolution optimale de ce problème via des modèles mathématiques pour pouvoir ensuite développer de nouveaux algorithmes à faible temps de calcul et évaluer leurs performances grâces aux résultats optimaux obtenus.

1 https://www.hcsp.fr/explore.cgi/avisrapportsdomaine?clefr=1078.

Ce problème de planification intègre de multiples contraintes inhérentes au milieu hospitalier. L'admission des patients n'a lieu que pendant une certaine période. Les patients suivent différents parcours constitués d'activités de soin, qui sont soumises à des contraintes de précédences partielles et des délais minimaux et maximaux entre des activités consécutives. La réalisation d'une activité nécessite l'allocation d'un ensemble de ressources, c'est-à-dire des équipements, des salles et des soignants avec des qualifications particulières. Ces ressources sont en nombre limité, sont utilisées à différentes fins dans différents parcours, et peuvent parfois être indisponibles. L'objectif est de minimiser la somme des temps d'attente des patients de leur admission à leur départ. Pour résoudre ce problème, nous proposons des modèles mathématiques ainsi que de comparer leurs performances dans le cadre d'une étude préliminaire. Les modèles mathématiques présentés se déclinent en deux types : le premier type de modèle est un problème linéaire dont certaines variables sont entières (MILP), le second type utilise la programmation par contraintes, et est décliné en trois variantes. Par soucis de lisibilité et de taille de l'article, nous ne listons pas l'ensemble des relations mathématiques modélisant les contraintes. L'objectif principal de cette étude est de comparer ces deux approches mathématiques afin d'en déterminer la plus efficace pour ce type de problème.

Les modèles proposés peuvent être utilisés dans un contexte opérationnel ou tactique. Dans un contexte opérationnel, pour simplement ordonnancer les actes médicaux d'une journée. Dans un contexte tactique en préparant des ordonnancements pour une multitude de combinaisons de parcours dont le nombre serait fixé par la politique de l'établissement, afin d'obtenir des pré-plannings fixes et de décider le jour pendant lequel chaque nouveau patient sera traité.

Le plan de l'article est le suivant. La section 1 présente un état de l'art sur la recherche opérationnelle en santé et sur les problèmes d'ordonnancement avec des contraintes de ressources. La section 2 détaille la modélisation du problème et présente deux approches de résolution. La section 3 décrit les expérimentations réalisées. La section 4 conclut et donne des directions pour les recherches futures.

1 État de l'art

De nombreux articles traitent le problème d'ordonnancement de séries d'actes médicaux à pratiquer dans un contexte hospitalier. Plusieurs parmi eux supposent un horizon de planification de plusieurs jours. [Apergi *et al.*, 2020] proposent un modèle d'optimisation pour minimiser le nombre de visites de patients en cardiologie et [Petrovic *et al.*, 2011] un algorithme génétique pour minimiser le temps d'accès au traitement pour des patients en cancérologie. Dans notre étude, nous considérons une planification d'une seule journée pendant laquelle des patients doivent réaliser des séries complètes d'actes médicaux variées. Dans ce même contexte, [Du *et al.* 2013] propose un algorithme génétique minimisant la date d'achèvement de la dernière activité d'une journée avec l'hypothèse forte qu'une activité ne nécessite qu'une seule ressource. Dans cette étude, nous considérons que les activités peuvent nécessiter plusieurs ressources à la fois et le critère à optimiser est la minimisation du temps d'attente des patients. Ce même critère est considéré par [Zhao *et al.*, 2018] qui planifient dans un contexte de réhabilitation plusieurs activités par patient dans une seule journée à l'aide d'un algorithme génétique. Cependant, les activités qu'ils considèrent ont peu ou pas de relations de précédence entre elles. Pour plus de détails, nous proposons au lecteur de se référer à l'excellent état de l'art de [Marynissen et Demeulemeester, 2019].

Parmi les problèmes combinatoires étudiés dans la littérature en recherche opérationnelle, le problème de planification de soins que nous étudions possède des similitudes avec le problème d'ordonnancement de projet

avec ressources limitées (RCPSP) dont un récent état de l'art est disponible dans [Hartmann et Briskorn, 2022]. Ce problème classique a pour but d'ordonnancer l'exécution d'un ensemble de tâches d'un projet afin de le terminer au plus vite. Chaque tâche nécessite un certain nombre de ressources pendant leur exécution, et les ressources disponibles à chaque instant sont limitées. Les tâches peuvent avoir des relations de précédence entre elles. De nombreuses variantes du RCPSP classique ont été étudiées, notamment deux : une qui considère plusieurs projets à la fois, et l'autre qui octroie aux ressources une ou plusieurs compétences, ce qui ajoute des décisions sur l'allocation des ressources en plus des décisions sur les dates d'exécution des tâches. Une description de la variante multi-projet est proposée par [Browning et Yassine, 2010]. L'une des premières mentions de la variante multi-compétences est faite par [Néron, 2002]. Le problème que nous traitons est similaire à une combinaison de ces deux variantes : chaque parcours patient est assimilé à un projet, et les ressources de l'hôpital ont chacune une ou plusieurs compétences. Plusieurs articles combinent ces deux variantes, mais la plupart considèrent d'autres objectifs non-temporels comme une minimisation des coûts [Hosseinian et Baradaran, 2021]. D'autres introduisent également des niveaux de compétences qui font varier la durée des activités [Heimerl et Kolisch, 2010], ce qui diffère de notre problème puisque la durée des actes médicaux n'est pas compressible. Beaucoup d'articles ne traitent qu'une des deux variantes. [Browning et Yassine, 2010] s'intéresse à la variante multi-projet uniquement et compare les performances d'heuristiques pour minimiser la somme des dates d'achèvement des projets. Cet objectif se rapproche d'un temps d'attente total, à condition que l'heure d'arrivée du patient à l'hôpital soit une donnée du problème. [Almeida *et al.*, 2016] considère la variante multi-compétences et compare également les performances d'heuristiques pour minimiser la date d'achèvement d'un unique projet. Cependant l'une des différences de ce type de problème avec le problème de planification de parcours de soin des patients est que les différentes activités de projets peuvent avoir lieu en parallèle alors que le patient ne peut suivre qu'une activité à la fois. À notre connaissance, il n'existe donc pas pour le moment d'études scientifiques traitant et résolvant le problème étudié dans cet article : la planification de parcours patients dans une même journée en considérant les aspects multi-activités, multi-ressources et multi-compétences.

2 Définition du problème

Le problème consiste à planifier sur une journée les activités d'un ensemble connu de patients au sein d'un hôpital de jour. Ces patients viennent sur rendez-vous, aucun ne vient en urgence, et la date du rendez-vous est généralement déterminée plusieurs jours voire semaines à l'avance. Le traitement de chaque patient et les activités qu'il subit sont définis par le parcours de soin qu'il doit suivre, fixé avant la planification. Le parcours complet du patient doit être planifié sur une seule journée et l'heure d'arrivée du patient à l'hôpital doit être planifiée dans une fenêtre de temps.

Un parcours est composé de plusieurs activités qui sont soumises à des contraintes de précédences partielles. Par exemple, les examens nécessitant que le patient soit à jeun ont lieu avant la collation. Nous supposerons que la durée de chaque activité est connue. Deux activités d'un même patient ne peuvent pas avoir lieu en même temps et certaines sont même contraintes par des délais minimum ou maximum entre elles à respecter. Les activités utilisent également des ressources limitées, qui sont les salles, les équipements et le personnel soignant. Ces ressources ne sont disponibles que pendant une certaine période afin de prendre en compte les horaires de travail du personnel médical ou encore des périodes d'indisponibilité de ressources matérielles lorsqu'elles sont réservées pour des activités extérieures au service ambulatoire. Certaines salles peuvent être utilisées à différentes fins, et chaque membre du personnel soignant a une ou plusieurs qualifications. Plus généralement,

il est dit qu'une ressource possède des compétences. Pour avoir lieu, une activité requiert une quantité définie de ressources de chaque compétence (chaque ressource ne pouvant apporter qu'une compétence). Le problème de planification des activités est par conséquent couplé à un problème d'allocation des ressources.

L'objectif du problème est de déterminer l'heure d'admission de chaque patient (qui lui sera donné suite à la planification), les heures de début et fin de chaque activité et les affectations des ressources aux activités, ce qui permettra de donner un planning à chaque ressource, tout en minimisant le temps d'attente total de tous les patients. La figure 1 représente un exemple de parcours patient constitué de 10 activités avec les contraintes de précédences, par exemple le bilan biologique précède la collation. Il est indiqué également la durée et les ressources nécessaires. Par exemple, l'activité Calorimétrie dure 30 min et nécessite un box calorimètre et d'une IDE obésité. Un box équipé d'un calorimètre pourra être utilisé en tant que simple box de prélèvement, et une IDE obésité pourra être allouée à une activité ne nécessitant qu'une IDE sans spécialisation.

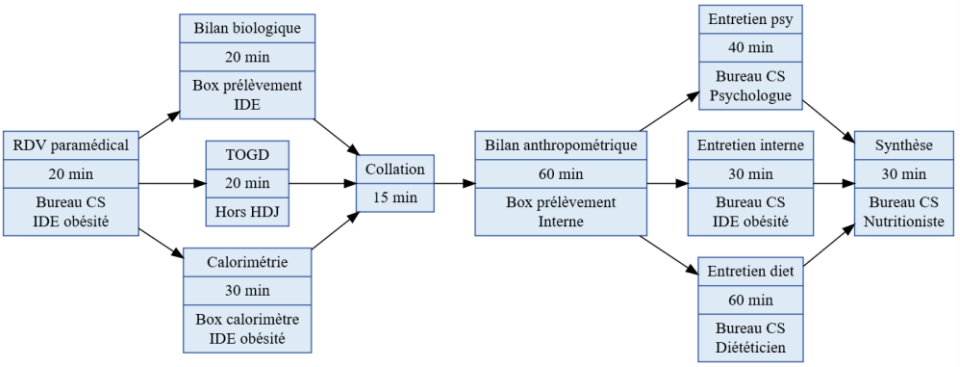


Figure 1 : Parcours Obésité sévère - Post-op

2.1 Notations et modélisation du problème

Nous considérons un ensemble de patients I, d'activités J, de compétences S et de ressources R. Les activités J_i de chaque patient i forment une partition de J. Chaque activité j dure p_j unités de temps, et requiert certaines compétences S_j. Une activité j a besoin exactement de n_{js} ressources possédant la compétence s. Une ressource r possède la compétence s si r est dans R_s. Certaines activités ont des relations de précédence. P est l'ensemble des couples d'activités (j,k) tels que j doit précéder k. De même, il existe des couples d'activités (j,k) ayant des délais minimum δ^-_{jk} et maximum δ^+_{jk} à respecter entre la fin de j et le début de k. L'ensemble des délais constitue Δ, et permet de modéliser par exemple une période de repos d'un patient. Chaque ressource r a une période de disponibilité qui a un début b_r et une fin e_r, pour représenter par exemple les horaires du personnel.

Tous nos modèles utilisent des variables de décisions identiques. L'heure d'admission d'un patient i sera notée t_i, son départ de l'hôpital f_i, et son temps d'attente w_i, qui est la différence entre f_i et la somme de t_i et des durées de toutes les activités du patient i. L'objectif est la minimisation de la somme des temps d'attente des patients. Les variables de décisions relatives aux dates de début ou de fin de chaque activité, et à l'allocation des ressources sont introduites avant chaque modèle ou variante qui les utilisent.

2.2 Modèle de programmation linéaire en nombre entiers

Le premier type de modèle se base sur une modélisation classique en recherche opérationnelle qui est la programmation linéaire en nombre entiers (MILP). Pour modéliser le problème sous cette forme, nous introduisons la variable c_j qui définit l'heure où une activité *j* se termine, ainsi que trois ensembles de variables de décisions booléennes. Le premier ensemble, y_{jrs}, permet de savoir pour chaque activité quelles ressources sont allouées, et avec quelles compétences ces ressources contribuent. y_{jrs} vaut 1 si la ressource *r* est utilisée par l'activité *j* pour sa compétence *s*, 0 sinon. La satisfaction des besoins des activités est garantie par la contrainte (1). La contrainte (2) empêche une ressource de contribuer plusieurs fois à une activité.

$$\sum_{r \in R_s} y_{jrs} = n_{js} \qquad \forall j \in J, \forall s \in S_j \qquad (1)$$

$$\sum_{s \in S} y_{jrs} \leq 1 \qquad \forall j \in J, \forall r \in R \qquad (2)$$

Les y_{jrs} sont aussi utilisées pour empêcher une activité d'utiliser une ressource en dehors de ses horaires de disponibilité. Les deux autres ensembles de variables de décisions sont utilisés dans des contraintes pour empêcher le chevauchement des activités lorsqu'elles concernent le même patient ou qu'elles utilisent la même ressource : pour chaque paire d'activités *(j,k)* d'un même patient, x_{jk} vaut 1 si *j* précède *k* et 0 sinon ; pour chaque ressource *r* et chaque paire d'activités *(j,k)* pouvant utiliser *r*, z_{rjk} vaut 1 si *j* et *k* utilisent *r* et *j* précède *k*, et 0 sinon.

2.3 Programmation par contraintes

Le deuxième type de modèle est basé sur la programmation par contraintes (PPC). "Activité" étant le nom d'un type d'objet dans le cadre de la PPC, les activités *j* seront mentionnées en tant qu'actes dans toute cette sous-section. Nous proposons trois modèles de PPC qui ne diffèrent que sur la façon de garantir la satisfaction des besoins des actes mais qui ont la même base : chaque acte *j* est représenté par un objet A_j nommé activité, qui contient le début, la fin et la durée de l'acte *j*. Les contraintes de précédence et de délais minimaux et maximaux entre activités consécutives n'utilisent que les objets A_j. Le non-chevauchement des activités concernant un même patient est garanti avec des contraintes utilisant la fonction *noOverlap*. Pour les ressources, des activités optionnelles B_{jrs} sont déclarées. B_{jrs} est présente seulement si la ressource *r* est allouée à l'acte *j* et contribue à *j* avec sa compétence *s*. Les activités B_{jrs} présentes sont synchronisées avec leurs activités respectives A_j, c'est-à-dire qu'elles commencent et se terminent en même temps. Le non-chevauchement des activités B_{jrs} présentes pour une même ressource *r* est garanti par la fonction *noOverlap*, et d'autres contraintes empêchent les activités B_{jrs} d'avoir lieu en dehors des horaires de disponibilité de leur ressource *r*.

À partir de cette base commune, nous proposons trois façons de garantir la satisfaction des besoins des actes. Dans la première variante (CP1), cette garantie est apportée par les contraintes (3) et (4) qui sont similaires aux contraintes (1) et (2) du MILP, où *presenceOf*(B_{jrs}) joue le rôle de la variable y_{jrs} proposée dans le MILP (*presenceOf*(B_{jrs}) vaut 1 si *r* est allouée à *j* et est utilisée pour sa compétence *s*, 0 sinon).

$$\sum_{r \in R_s} \mathrm{presenceOf}(B_{jrs}) = n_{js} \qquad \forall j \in J, \forall s \in S_j \qquad (3)$$

$$\sum_{s \in S} \mathrm{presenceOf}(B_{jrs}) \leq 1 \qquad \forall j \in J, \forall r \in R \qquad (4)$$

Dans la deuxième variante (CP2), les contraintes (3) et (4) sont remplacées par la contrainte (5) utilisant fonction *alternative*. Cela permet à la fois de synchroniser les B_{jrs} présents avec leurs A_j respectifs, mais également d'en sélectionner exactement le nombre requis pour être présents.

$$\mathrm{alternative}(A_j, \{B_{jrs} / r \in R_s\}, n_{js}) \qquad \forall j \in J, \forall s \in S_j \qquad (5)$$

Dans la troisième variante (CP3), on fait l'hypothèse de ressources disjointes, c'est-à-dire qu'aucune ressource

ne possède deux compétences demandées par la même activité. Dans ce cas, la compétence utilisée de la ressource pour l'activité à laquelle elle contribue se déduit par la connaissance de l'allocation de la ressource à l'activité. On peut donc réduire le nombre de variables de décision en n'indexant plus les activités optionnelles B_{jrs} selon s. Cela donne un ensemble d'intervalles B_{jr}. On utilise la fonction *alternative* de la deuxième variante, ce qui donne la contrainte (6). L'hypothèse faite pour CP3 est vérifiée pour tous les parcours patients qui ont été mis à notre disposition.

$$\text{alternative}(A_j, \{B_{jr}/r \in R_s\}, n_{js}) \qquad \forall j \in J, \forall s \in S_j \qquad (6)$$

3 Expérimentations numériques

Ces deux types de modèle s'appuyant sur des techniques de résolution bien différentes, nous les avons comparés ainsi que leurs variantes sur des instances générées aléatoirement mais en s'inspirant de parcours patients réels. La génération des instances et la procédure associée sont décrites dans la sous-section 3.1. Les résultats obtenus par les différents modèles sont analysés dans la sous-section 3.2.

3.1 Génération d'instances

Pour générer les instances de tests, nous nous sommes assurés d'obtenir des instances variées, non-triviales à résoudre et représentatives de cas réels. L'ensemble des parcours patients et des ressources utilisés proviennent de données réelles fournies par l'APHP. La génération d'instances utilise deux paramètres : le nombre de patients et α le facteur de disponibilité des compétences. Chaque patient se voit attribuer un parcours aléatoirement. Cela définit l'ensemble des activités à planifier et permet d'en déduire la demande globale en temps-ressource pour chaque compétence. L'ensemble des ressources disponibles est constitué en ajoutant des ressources une à une jusqu'à ce que l'offre dans chaque compétence, c'est-à-dire la quantité de temps-ressource pouvant être fournie, dépasse la demande de cette compétence en temps-ressource multipliée par α.

Nous avons généré deux lots d'instances. Le premier lot d'instances utilise trois petits parcours patients de trois à cinq activités. Le nombre de patients prend ses valeurs dans le domaine {5, 10, 15, 20} et α dans {1, 1.2, 1.4, 1.6, 1.8}. Pour chaque combinaison de ces deux paramètres, la procédure décrite au paragraphe précédent a été appliquée pour produire cinq instances, ce qui donne un total de cent instances dans le premier lot. Le deuxième lot d'instances utilise six parcours patients plus volumineux (de six à onze activités par parcours) liés à l'obésité, afin de limiter la diversité des compétences demandées. Les domaines des paramètres sont les mêmes, à part α qui prend ses valeurs dans {1.4, 1.6, 1.8}, car un grand nombre de patients et un α faible donnent de trop nombreuses instances infaisables. Le second lot compte donc soixante instances.

3.2 Résultats et comparaison des modèles

Les instances ont été résolues sur des ordinateurs équipés de processeurs Intel Core i7-7700 de fréquence 3,60 Ghz avec des limites de temps d'exécution de cinq minutes et une heure. Chaque instance a été résolue avec chaque modèle en utilisant la version 20.1 du solveur mathématique CPLEX.

Le tableau 1 présente pour chaque modèle le nombre d'instances pour lesquelles la solution optimale a été trouvée (#opt), une solution a été trouvée (#ub), et la meilleure solution a été trouvée par rapport l'ensemble des modèles (#best) avec un temps limite d'exécution de cinq minutes. La colonne Fo contient la moyenne des temps d'attente pour les instances où le modèle a trouvé au moins une solution. La colonne CPU indique le temps

moyen en seconde pris pour trouver la solution optimale lorsque le modèle la trouve. La colonne #act contient le nombre moyen d'activités par instance. Le tableau 2 présente de la même manière que le tableau 1, les résultats obtenus par les différents modèles avec un temps limite d'exécution d'une heure. Les tableaux 3 et 4 présentent les expérimentations pour le second lot d'instances sous le même format que les tableaux 1 et 2.

		MILP					CP1					CP2					CP3				
#patients	#act	#opt	CPU	#ub	Fo	#best	#opt	CPU	#ub	Fo	#best	#opt	CPU	#ub	Fo	#best	#opt	CPU	#ub	Fo	#best
5	21.6	**25**	**2.4**	**25**	**9.0**	**25**	**25**	9.3	**25**	**9.0**	**25**	**25**	8.2	**25**	**9.0**	**25**	**25**	8.1	**25**	**9.0**	**25**
10	43.4	8	44.9	**25**	84.6	14	9	9.8	**25**	82.2	12	10	12.0	**25**	**76.8**	**17**	**11**	12.8	**25**	83.8	13
15	66.2	-	-	23	374	0	4	111.2	**25**	**153**	**13**	**6**	31.6	**25**	154	12	5	62.6	**25**	163	11
20	85.4	-	-	12	943	0	1	286	**25**	269	3	4	155	**25**	**193**	**14**	-	-	**25**	204	9

Tableau 1 : Résultats obtenus sur le premier lot d'instances avec une limite de temps d'exécution de 5 min.

		MILP					CP1					CP2					CP3				
#patients	#act	#opt	CPU	#ub	Fo	#best	#opt	CPU	#ub	Fo	#best	#opt	CPU	#ub	Fo	#best	#opt	CPU	#ub	Fo	#best
5	21.6	**25**	**2.4**	**25**	**9.0**	**25**	**25**	9.4	**25**	**9.0**	**25**	**25**	8.2	**25**	**9.0**	**25**	**25**	8.1	**25**	**9.0**	**25**
10	43.4	10	110	**25**	**71.0**	14	**11**	402	**25**	79.4	14	**11**	47.0	**25**	**72.4**	14	**11**	**12.8**	**25**	73.2	**15**
15	66.2	3	2970	23	184	3	**8**	714	**25**	**113**	11	6	31.6	**25**	134	11	**8**	**232**	**25**	128	**18**
20	85.4	-	-	23	530	0	4	591	**25**	200	6	**9**	765	**25**	**169**	**15**	5	941	**25**	175	13

Tableau 2 : Résultats obtenus sur le premier lot d'instances avec une limite de temps d'exécution de 1h.

Le MILP obtient les meilleurs résultats en moyenne pour les petites instances de cinq patients. Pour les instances plus grandes (quinze et vingt patients), ce modèle trouve peu de solutions optimales ou des solutions aussi proches de l'optimale que les autres modèles. Le modèle MILP rencontre également des difficultés pour trouver des solutions faisables en moins de cinq minutes. Contrairement au MILP, les trois modèles basés sur la programmation par contraintes sont capables de trouver des solutions faisables pour presque chaque instance en moins de cinq minutes, ce qui donne la possibilité de les utiliser pour produire des solutions initiales au sein de méthodes approchées. Ils permettent également de trouver plus de solutions optimales.

Parmi les trois modèles basés sur la programmation par contraintes, CP2 et CP3 obtiennent globalement de meilleurs résultats que CP1, ce qui est flagrant sur le second lot d'instances où CP1 ne trouve pas de solutions faisables à toutes les instances en moins d'une heure. Les performances de CP2 et CP3 sur les instances du premier lot sont assez proches. CP2 obtient globalement de meilleurs résultats que CP3 en cinq minutes, mais CP3 trouve le plus de meilleures solutions en une heure. Cet écart en faveur de CP3 est notamment visible pour les instances du second lot possédant plus de 15 patients, pour le nombre de meilleurs solutions trouvées mais également pour la moyenne des temps d'attente trouvés. Le principal désavantage de CP3 est que ce modèle nécessite que les ressources soient disjointes, contrairement à CP2 qui ne nécessite pas la satisfaction de cette contrainte.

		MILP					CP1					CP2					CP3				
#patients	#act	#opt	CPU	#ub	Fo	#best	#opt	CPU	#ub	Fo	#best	#opt	CPU	#ub	Fo	#best	#opt	CPU	#ub	Fo	#best
5	49.0	11	41.0	**15**	7.5	12	13	3.5	**15**	1.3	14	**14**	2.6	**15**	**1.0**	**15**	13	2.1	**15**	1.3	14
10	97.9	-	-	-	-	-	3	14.4	**15**	50.6	5	**7**	**36.1**	**15**	**31.7**	**12**	**7**	61.7	**15**	38.0	10
15	142.7	-	-	-	-	-	1	58.5	11	220	2	4	190	**15**	118	**8**	3	126	**15**	**93.0**	**8**
20	190.2	-	-	-	-	-	-	-	3	343	0	1	64.7	**12**	249	**8**	1	78.9	11	253	5

Tableau 3 : Résultats obtenus sur le second lot d'instances avec une limite de temps d'exécution de 5 min.

		MILP					CP1					CP2					CP3				
#patients	#act	#opt	CPU	#ub	Fo	#best	#opt	CPU	#ub	Fo	#best	#opt	CPU	#ub	Fo	#best	#opt	CPU	#ub	Fo	#best
5	49.0	12	68.1	**15**	1.4	13	13	3.4	**15**	1.3	14	**15**	**199**	**15**	**1.0**	**15**	**15**	212	**15**	**1.0**	**15**
10	97.9	-	-	11	334	0	7	250	**15**	34.0	7	8	457	**15**	**17.3**	**12**	**9**	269	**15**	26.3	9
15	142.7	-	-	-	-	-	5	1106	11	53.1	8	**8**	598	**15**	70.0	9	7	604	**15**	**56.7**	**10**
20	190.2	-	-	-	-	-	-	-	5	45	0	**5**	1029	**15**	106	8	3	624	**15**	**73.7**	**10**

Tableau 4 : Résultats obtenus sur le second lot d'instances avec une limite de temps d'exécution d'1h.

Les tableaux 3 et 4 confirment les premières conclusions tirées des tableaux 1 et 2 : les modèles par programmation par contraintes sont plus performants que le modèle par programmation linéaire, et CP2 semble le meilleur des modèles par programmation par contraintes. Nous pouvons noter également que malgré la taille des instances (jusqu'à 20 patients et 190 activités en moyenne), le modèle CP2 trouve des solutions réalisables en moins d'une heure. Cependant, à partir de 10 patients et une moyenne de 98 activités, moins de la moitié des instances est résolue à l'optimal en moins de cinq minutes.

Conclusion

Cet article étudie un problème de planification d'actes médicaux en ambulatoire. Ce problème consiste à planifier sur une journée un ensemble d'actes rassemblés en différents parcours patients, et à leur allouer des ressources multi-compétences et en nombre limité, afin de minimiser la somme des temps d'attente des patients. Dans le cadre d'une étude préliminaire nous proposons deux types de modèle mathématique. Des expérimentations sur cent soixante instances inspirées de cas réels permettent de comparer les approches et démontrent l'efficacité des méthodes basées sur la programmation par contraintes.

Parmi les perspectives de recherche, une direction envisagée est la conception de méthodes approchées afin de résoudre rapidement et régulièrement des problèmes à taille réelle (au moins une cinquantaine de patients dans notre cas d'étude). Ces méthodes nécessitant un faible temps de calcul pourraient être ensuite intégrées dans un outil d'aide à la décision pour réaliser de la planification des parcours en temps réel qui recalculerait de nouvelles solutions à chaque événement ou perturbation (retards de patient, annulation, etc.). Les approches développées de ce papier pourront être utilisées à des fins de comparaison.

Références

Almeida, B. F., I. Correia et F. Saldanha-da-Gama (2016). Priority-based heuristics for the multi-skill resource constrained project scheduling problem. *Expert Systems with Applications*, 57, 91-103.

Apergi, L. A., J. S. Baras, B. L. Golden et K. E. Wood (2020). An optimization model for multi-appointment scheduling in an outpatient cardiology setting. *Operations Research for Health Care*, 26, 100267.

Browning, T. R. et A. A. Yassine (2010). Resource-constrained multi-project scheduling: Priority rule performance revisited. *International Journal of Production Economics*, 126(2), 212-228.

Chern, C. C., P. S. Chien et S. Y. Chen (2008). A heuristic algorithm for the hospital health examination scheduling problem. *European Journal of Operational Research*, 186(3), 1137-1157.

Du, G., Z. Jiang, Z. Yao et X. Diao (2013). Clinical pathways scheduling using hybrid genetic algorithm. *Journal of medical systems*, 37(3), 1-17.

Hartmann, S. et D. Briskorn (2022). An updated survey of variants and extensions of the resource-constrained project scheduling problem. *European Journal of Operational Research*, 297(1), 1-14.

Heimerl, C. et R. Kolisch (2010). Scheduling and staffing multiple projects with a multi-skilled workforce. *OR spectrum*, 32(2), 343-368.

Hosseinian, A. H. et V. Baradaran (2021). A two-phase approach for solving the multi-skill resource-constrained multi-project scheduling problem: a case study in construction industry. *Engineering, Construction and Architectural Management*, 2021.

Marynissen, J. et E. Demeulemeester (2019). Literature review on multi-appointment scheduling problems in hospitals. *European Journal of Operational Research*, 272(2), 407-419.

Néron, E. (2002). Lower bounds for the multi-skill project scheduling problem. In : *Proceeding of the eighth international workshop on project management and scheduling*, pp. 247-277.

Petrovic, D., M. Morshed et S. Petrovic (2011). Multi-objective genetic algorithms for scheduling of radiotherapy treatments for categorised cancer patients. *Expert Systems with Applications*, 38(6), 6994-7002.

Zhao, L., C. F. Chien et M. Gen (2018). A bi-objective genetic algorithm for intelligent rehabilitation scheduling considering therapy precedence constraints. *Journal of Intelligent Manufacturing*, 29(5), 973-988.