



HAL
open science

Planification de ressources et ordonnancement de tâches dans la chaîne inter et intra logistique hospitalière

Lahcene Mezouari

► **To cite this version:**

Lahcene Mezouari. Planification de ressources et ordonnancement de tâches dans la chaîne inter et intra logistique hospitalière. Recherche opérationnelle [math.OC]. Université de Technologie de Compiègne, 2023. Français. NNT : 2023COMP2758 . tel-04692299

HAL Id: tel-04692299

<https://theses.hal.science/tel-04692299v1>

Submitted on 9 Sep 2024

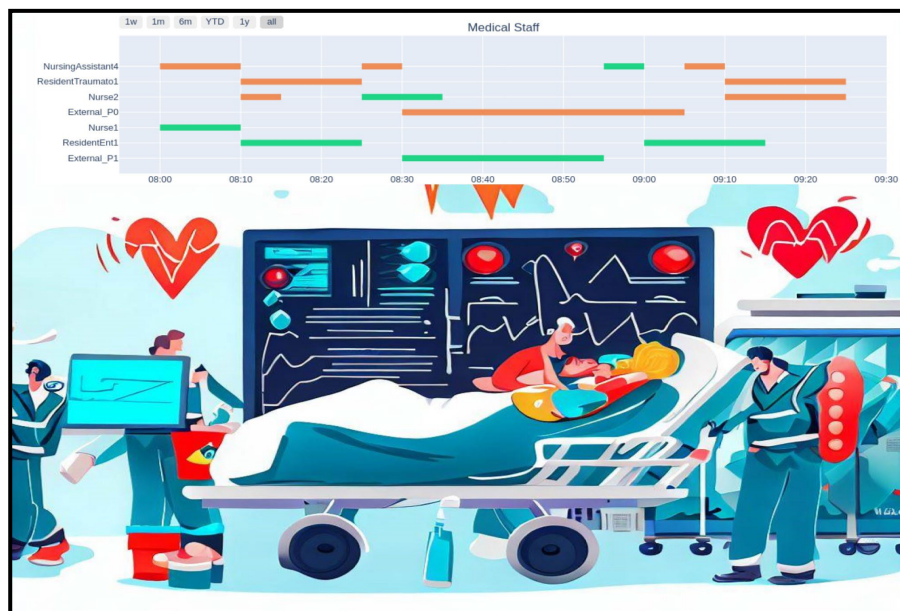
HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

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.

Par **Lahcene MEZOUARI**

*Planification de ressources et
ordonnancement de tâches dans la chaîne
inter et intra logistique hospitalière*

Thèse présentée
pour l'obtention du grade
de Docteur de l'UTC



Soutenue le 11 mai 2023

Spécialité : Informatique : Unité de recherche Heudyasic
(UMR-7253)

D2758



THESE

PRÉSENTÉE EN VUE D'OBTENIR LE GRADE DE
DOCTEUR

Par

Lahcene MEZOUARI

Planification de ressources et ordonnancement de tâches dans la chaîne inter et intra logistique hospitalière

Spécialité : Informatique

Soutenue le 11 mai 2023 devant le jury composé :

- **Mme Laetitia JOURDAN**, professeur, membre rapporteur
CRISAL, Université de Lille, Villeneuve d'Ascq
- **M. Nikolay TCHERNEV**, professeur, membre rapporteur
LIMOS, Université Clermont Auvergne, Aubière
- **M. Hamid ALLAOUI**, professeur, membre examinateur
Faculté de sciences appliquées, Université d'Artois, Béthune
- **M. Jean-Marie RENARD**, maître de conférences, membre examinateur
Faculté de médecine, centre d'études et de recherche en informatique, Lille
- **Mme Joanna DAABOUL**, maître de conférences, membre examinateur
Roberval, Université de Technologie de Compiègne
- **M. Aziz MOUKRIM**, professeur, directeur de thèse
Heudiasyc, Université de Technologie de Compiègne
- **M. Jean-Paul BOUFFLET**, maître de conférences, co-encadrant de thèse
Heudiasyc, Université de Technologie de Compiègne

Remerciements

Au terme de cette aventure passionnante, il est essentiel pour moi d'exprimer ma gratitude envers toutes les personnes qui ont contribué de manière significative à la réalisation de cette thèse. Ces années de recherche ont été un véritable parcours jalonné de découvertes, de défis et de moments précieux, et je tiens à adresser mes remerciements les plus sincères à tous ceux qui m'ont accompagné tout au long de ce voyage.

Avant toute chose, je tiens à témoigner de ma sincère gratitude envers mes directeurs de thèse, Aziz et Jean-Paul. Votre encadrement, vos compétences pointues et votre patience ont joué un rôle déterminant dans la réussite de ce projet.

Par la suite, je désire adresser mes remerciements à Laetitia JOURDAN et Nikolay TCHERNEV pour avoir investi leur temps et leur expertise dans l'évaluation minutieuse de mes travaux, malgré des délais particulièrement serrés. Votre patience et votre bienveillance ont grandement contribué à ma préparation en vue de la soutenance. J'exprime également ma profonde gratitude envers Joanna DAABOUL, Hamid ALLAOUI, Jean-Marie RENARD d'avoir accepté de faire partie du jury de ma thèse. Les échanges fructueux que nous avons eus lors de la soutenance ont enrichi cette étape cruciale de mon parcours. Je tiens également à exprimer ma reconnaissance envers le personnel du service d'urgence du CHU Jeanne de Flandres de Lille, qui a joué un rôle essentiel dans mes visites au service. Je souhaite également exprimer ma profonde reconnaissance envers tous les membres du laboratoire Heudiasyc.

Le soutien de ma famille a été une lumière constante tout au long de ce voyage. À mes parents, vos expressions d'amour, vos encouragements inlassables et les sacrifices que vous avez consentis ont modelé la personne que je suis aujourd'hui. Mes sœurs et frères, à chacun de vous, je veux exprimer toute ma gratitude. Votre impact transcende les mots, et je suis profondément reconnaissant d'avoir des frères et sœurs aussi exceptionnels que vous.

Résumé

Les hôpitaux doivent prendre en compte la logistique pour améliorer les liens entre les systèmes de santé (entre les établissements et entre les divisions au sein d'un hôpital) afin de promouvoir la cohérence et l'efficacité. La logistique hospitalière implique une bonne coordination des services de prise en charge des patients. Les patients en situation d'urgence se rendent généralement en premier lieu au service des urgences. Une visite aux urgences peut révéler qu'un patient a besoin d'une intervention chirurgicale à programmer plus tard. Cela crée des patients en chirurgie électorale. Dans le cadre de l'ANR OIILH, nous avons proposé des solutions pour le problème d'ordonnancement des tâches de parcours de soins de patients. Nous avons également étudié le problème de planification d'interventions chirurgicales. Pour le problème des urgences, nous avons considéré en premier lieu le cas statique. Nous avons réalisé un générateur d'instances basé sur les données réelles du service d'urgence du CHU Jeanne de Flandres de Lille. Nous avons implémenté une approche heuristique constructive permettant de gérer les ressources en suivant deux règles métier utilisées par les praticiens. Nous avons proposé une modélisation mathématique du problème. Le modèle a été implémenté et utilisé pour obtenir des solutions de références pour les instances générées. Nous avons élaboré une approche heuristique itérative basée sur la destruction et la construction d'une solution. Les résultats montrent que des solutions de meilleure qualité par rapport à l'heuristique constructive peuvent être obtenues en des temps raisonnables. Des solutions optimales peuvent être trouvées pour les petites instances. Ensuite, nous avons élaboré une méta-heuristique *Adaptive Large Neighborhood Search* (ALNS) plus efficace. Les résultats expérimentaux montrent l'efficacité de cette approche avec des temps de calcul faibles qui sont adéquats par rapport à la réactivité requise dans le cas dynamique du problème. Nous avons proposé une approche de planification prédictive et réactive pour le cas dynamique. Nous construisons, modifions et améliorons le planning en utilisant les méthodes de résolution du cas statique. Pour le service de chirurgie, des interventions chirurgicales non urgentes et non immédiates doivent être programmées à l'avance. Nous avons développé deux approches de résolution pour ce problème en aval du service des urgences. Nous avons proposé une heuristique en deux phases et une approche ALNS qui donnent des résultats strictement meilleurs que ceux de la littérature.

Mots-clés : logistique hospitalière, service des urgences, planification des interventions chirurgicales, planification dynamique, ordonnancement, heuristique adaptative, métaheuristique.

À la mémoire de mes chers grands-parents, dont le départ de ce monde est survenu avant que vous ne puissiez partager avec moi la joie d'atteindre le titre de docteur.

TABLE DES MATIÈRES

Résumé	iii
1 Introduction générale	7
2 Domaine de l'étude	13
2.1 Introduction	13
2.2 Problèmes d'optimisation dans la logistique hospitalière	14
2.3 Les problèmes d'optimisation dans les services des urgences (SU)	16
2.3.1 Triage et gestions de flux des patients	17
2.3.2 Allocation des ressources	19
2.3.3 Affectation de personnel	20
2.3.4 Planifications des tâches	21
2.4 Les problèmes d'optimisation dynamique	22
2.5 Les problèmes de planification des intervention chirurgicales	26
2.6 Conclusion	32
3 Problème de planification dans le service des urgences	33
3.1 Introduction	33
3.2 Définition du problème	34
3.2.1 Structure générale du service des urgences adultes du CHU de Lille	34
3.2.2 Structure du circuit court	36
3.2.3 Structure du circuit long	37
3.2.4 Tâches et contraintes entre tâches	37
3.2.5 Ressources et lieux	40
3.2.6 Calcul de performance et objectifs	41
3.3 Modélisation mathématique circuit court	41
3.4 Modélisation mathématique circuit long	50
3.5 Approches de résolution cas statique	54
3.5.1 Heuristique constructive basée sur les règles métiers (PRH)	54
3.5.2 IDCH	55
3.5.3 Méta-heuristique ALNS	56
3.6 Approche de résolution cas dynamique	64
3.7 Résultats	66
3.7.1 Benchmark et protocole expérimental	66
3.7.2 Paramétrage de IDCH	67
3.7.3 Paramétrage de ALNS acceptance base <i>Record-to-record</i>	68
3.7.4 Paramétrage de ALNS acceptance base recuit simulé	73
3.7.5 Évaluation des composantes de l'ALNS	78

3.7.6	Comparaison des deux versions de l'ALNS avec IDCH et ILP	80
3.7.7	Algorithme de planification dynamique	82
3.8	Conclusion et perspectives	84
4	Problème de planification des intervention chirurgicales	87
4.1	Introduction	87
4.2	Définition du problème et modélisation mathématique	88
4.3	Approches de résolutions	94
4.3.1	Heuristique 2PSC-EM	94
4.3.2	Métaheuristique ALNS	96
4.4	Résultats	106
4.4.1	Benchmark et protocole expérimental	106
4.4.2	Comparaison de l'heuristique 2PSC-EM avec la littérature	107
4.4.3	Paramétrage de ALNS	108
4.4.4	Évaluation des composantes de l'ALNS	112
4.4.5	Comparaison de l'ALNS avec la littérature	114
4.5	Conclusion et perspectives	115
5	Conclusion générale perspectives	117
A	Parcours des patients	121
A.1	Parcours Divers/Consultation Générale	121
A.1.1	Sans examens	121
A.1.2	Avec examen biologique uniquement	122
A.1.3	Avec autres examens, mais sans examen biologique	123
A.1.4	Avec examens (biologique + autres)	124
A.2	Parcours ORL	125
A.2.1	Sans examens	125
A.2.2	Avec examen biologique uniquement	126
A.2.3	Avec autres examens, mais sans examen biologique	127
A.2.4	Avec examens (biologique + autres)	128
A.3	Parcours Stomatologie	129
A.3.1	Sans examens	129
A.3.2	Avec examen biologique uniquement	130
A.3.3	Avec autres examens, mais sans examen biologique	131
A.3.4	Avec examens (biologique + autres)	132
A.4	Parcours Traumatologie	133
A.4.1	Sans examens	133
A.4.2	Sans plâtre	134
A.4.3	Avec plâtre	136
A.5	Parcours Traumatologie avec suture	138
A.5.1	Sans examens	138
A.5.2	Sans plâtre	139
A.5.3	Avec plâtre	141

LISTE DES FIGURES

2.1	Planification entièrement réactive (Bukkur et al., 2018)	24
2.2	Planification pro-active robuste (Bukkur et al., 2018)	25
2.3	Planification prédictive-réactive (Bukkur et al., 2018)	26
3.1	Structure du SUA de Lille	34
3.2	Parcours patient dans le SUA	35
3.3	Exemple de graphe représentant un parcours patient	38
3.4	Exemple de trois tâches de soin commençant en même temps à t suivies de leurs tâches d'attente. Une autre tâche pour le patient j peut débuter après les tâches de soin et leurs attentes.	39
3.5	Pour un patient j , séquencement des tâches $i_1 \prec i_2 \prec i_3 \prec i_4$ de durées respectives $p_{i_1j}, a_{i_2j}, p_{i_3j}, a_{i_4j}$.	40
3.6	Illustration de l'algorithme dynamique proposé	66
3.7	Paramétrage de β pour IDCH, effet de β sur le RPE.	69
3.8	Paramétrage de β pour IDCH, effet de β sur le CPU.	69
3.9	Réglage de Δ_{Dev} , effet de sur le RPE.	71
3.10	Réglage de $Iter_D$, effet de δ sur le RPE.	71
3.11	Réglage de σ_2 et σ_3 par rapport à σ_1 , effet de θ et ϕ sur le RPE.	72
3.12	Effet de p_u sur le RPE.	72
3.13	Effet de l sur le RPE.	73
3.14	Réglage de $Iter_{max}$, effet de γ sur le RPE.	73
3.15	Réglage de T_0 , effet de T_0 sur le RPE.	75
3.16	Réglage de T_0 , effet de sur le CPU.	75
3.17	Réglage de α , effet de α sur le RPE.	75
3.18	Réglage de α , effet de α sur le CPU.	75
3.19	Réglage de σ_2 et σ_3 par rapport à σ_1 , effet de θ et ϕ sur le RPE.	76
3.20	Effet de p_u sur le RPE.	76
3.21	Effet de l sur le RPE.	77
3.22	Réglage de $Iter_{max}$ effet de γ sur le RPE.	77
3.23	Les temps de mise à jour de planning dans le cas dynamique	84
4.1	Réglage de D_{limit} , effet de α sur RPE.	110
4.2	Réglage de Δ_{Dev} , effet de sur l'RPE.	110
4.3	Réglage de $Iter_D$, effet de δ sur l'RPE.	110
4.4	Réglage de σ_2 et σ_3 par rapport à σ_1 , effet de θ et ϕ sur le RPE.	110
4.5	Effet de p_u sur l'RPE.	111
4.6	Effet de l sur l'RPE.	111
A.1	Divers sans examens : Graphe de tâches	121

A.2 Divers avec uniquement examen biologique : Graphe de tâches 122

A.3 Divers avec examens sans examen biologique : Graphe de tâches 123

A.4 Divers avec examens dont examen biologique : Graphe de tâches 124

A.5 ORL sans examens : Graphe de tâches 125

A.6 ORL avec uniquement examen biologique : Graphe de tâches 126

A.7 ORL avec examens sans examen biologique : Graphe de tâches 127

A.8 ORL avec examens dont examen biologique : Graphe de tâches 128

A.9 Stomatologie sans examens : Graphe de tâches 129

A.10 Stomatologie avec uniquement examen biologique : Graphe de tâches . . . 130

A.11 Stomatologie avec examens sans examen biologique : Graphe de tâches . . 131

A.12 Stomatologie avec examens dont examen biologique : Graphe de tâches . . 132

A.13 Traumatologie sans examens : Graphe de tâches 134

A.14 Traumatologie avec examens sans examen biologique sans plâtre : Graphe
de tâches 134

A.15 Traumatologie avec examens dont examen biologique sans plâtre : Graphe
de tâches 135

A.16 Traumatologie avec examens sans examen biologique : Graphe de tâches . 136

A.17 Traumatologie avec examens dont examen biologique : Graphe de tâches . 137

A.18 Traumatologie avec suture sans examens : Graphe de tâches 139

A.19 Traumatologie avec suture avec examens sans examen biologique sans
plâtre : Graphe de tâches 140

A.20 Traumatologie avec suture avec examens dont examen biologique sans
plâtre : Graphe de tâches 142

A.21 Traumatologie avec suture avec examens sans examen biologique : Graphe
de tâches 144

A.22 Traumatologie avec suture avec examens dont examen biologique :
Graphe de tâches 146

LISTE DES TABLEAUX

2.1	La classification par niveau de décision	14
2.2	Études sur les problèmes de programmation des interventions chirurgicales.	31
3.1	Délais maximaux de prise en charge selon le code CCMU	36
3.2	Lieux du CC	36
3.3	Personnel médical du CC	37
3.4	Personnel médical du CL	37
3.5	Description des tâches de la figure (A.17)	38
3.6	Caractéristiques des instances générées	68
3.7	Caractéristiques des instances générées	69
3.8	Paramètres θ , ϕ , δ et γ à régler.	70
3.9	Paramètres α , θ , ϕ , δ et γ à régler.	74
3.10	Impact des opérateurs de destruction DAP, DO, DCW, DR, DA et DL pour ALNS avec recuit simulé.	79
3.11	Impact des opérateurs de destruction DAP, DO, DCW, DR, DA et DL pour ALNS avec <i>record-to-record</i>	80
3.12	Impact des opérateurs de construction LA, LO, LCW et impact du mécanisme d'adaptation pour ALNS avec recuit simulé.	81
3.13	Impact des opérateurs de construction LA, LO, LCW et impact du mécanisme d'adaptation pour ALNS avec <i>record-to-record</i>	82
3.14	Comparaison des deux version de l'ALNS avec IDCH et ILP	83
3.15	Comparaison entre les résultats de l'algorithme dynamique et l'approche statique avec et sans la phase d'amélioration	84
4.1	heuristique 2PSC-EM, 100 exécutions et trois exécutions, comparée à ACI.	107
4.2	Paramètres α , θ , ϕ , δ et γ à régler.	109
4.3	Impact des opérateurs de destruction RR, WR, RDR et WDR.	112
4.4	Impact des opérateurs de réparation INP, NMD et AC, et impact du mécanisme d'adaptation.	113
4.5	ALNS par rapport à ACI	114
A.1	Divers sans examens : description des tâches	122
A.2	Divers avec uniquement examen biologique : Description des tâches	123
A.3	Divers avec examens sans examen biologique : Description des tâches	124
A.4	Divers avec examens dont examen biologique : Description des tâches	125
A.5	ORL sans examens : Description des tâches	126
A.6	ORL avec uniquement examen biologique : Description des tâches	127
A.7	ORL avec examens sans examen biologique : Description des tâches	128
A.8	ORL avec examens dont examen biologique : Description des tâches	129

A.9 Stomatologie sans examens : Description des tâches 130

A.10 Stomatologie avec uniquement examen biologique : Description des tâches 131

A.11 Stomatologie avec examens sans examen biologique : Description des tâches 132

A.12 Stomatologie avec examens dont examen biologique : Description des tâches 133

A.13 Traumatologie sans examens : Description des tâches 134

A.14 Traumatologie avec examens sans examen biologique sans plâtre : Description des tâches 135

A.15 Traumatologie avec examens dont examen biologique sans plâtre : Description des tâches 136

A.16 Traumatologie avec examens sans examen biologique : Description des tâches 137

A.17 Traumatologie avec examens dont examen biologique : Description des tâches 138

A.18 Traumatologie avec suture sans examens : Description des tâches 139

A.19 Traumatologie avec suture avec examens sans examen biologique sans plâtre : Description des tâches 141

A.20 Traumatologie avec suture avec examens dont examen biologique sans plâtre : Description des tâches 143

A.21 Traumatologie avec suture avec examens sans examen biologique : Description des tâches 145

A.22 Traumatologie avec suture avec examens dont examen biologique : Description des tâches 147

INTRODUCTION GÉNÉRALE

CONTEXTE

Le Haut Conseil de la santé publique définit le système de santé comme l'ensemble des moyens et des activités dont la fonction est d'assurer les soins (promotion, prévention, réparation, rééducation et réinsertion).

L'OMS a développé un indicateur global de performance en matière de santé mondiale en 2000. Une nation dispose d'un bon système de soins si elle peut garantir globalement à ses citoyens un niveau de santé élevé. Il s'agit d'assurer une répartition relativement égale de ce niveau dans la population, d'avoir un système de soins réactif qui garantit que les individus sont traités avec respect, rapidement et efficacement, et d'avoir une répartition équitable des coûts associés à ces soins.

Assurer un niveau élevé de cet indicateur est un défi compte tenu de demandes croissantes et de contraintes budgétaires difficiles. L'accès à des services de santé de qualité est devenu un enjeu majeur [Xie et al., 2015].

Les hôpitaux doivent penser en terme de logistique pour améliorer les connexions entre les systèmes de santé (entre les établissements et entre les services d'un même établissement) afin d'augmenter la cohérence et l'efficacité du système de santé actuel. La logistique dans un établissement hospitalier nécessite la coordination efficace des services qui gèrent des flux différents de patients.

Fondamentalement, deux types de patients à l'origine de flux de natures différentes sont à considérer. Premièrement, les patients dont la venue est non programmée (patient dit en situation d'urgence) et ceux dont la venue est programmée (patient dit électif).

Généralement, le premier contact des patients en situation d'urgence avec un hôpital est le service des urgences. Il représente un pourcentage important des besoins en soins non programmés de l'hôpital. La mission principale des services des urgences adultes (SUA) est d'assurer la prise en charge des venues non programmées, dans un établissement de santé public ou privé, vingt-quatre heures sur vingt-quatre, tous les jours de l'année, de toute personne sans sélection, se présentant en situation d'urgence, y compris psychiatrique. Une autre mission importante des SUA est de classer les patients en fonction de leur état de santé et de leur pathologie, ce qui facilitera par la suite leur prise en charge ultérieure, éventuellement en tant que patient électif dans d'autres services.

Les patients se présentent généralement aux urgences avec un niveau d'anxiété élevé car ils arrivent sans savoir quelle est leur pathologie. L'un des défis du service des

urgences est de déterminer rapidement la pathologie. La mise en tension des SUA est causée par l'augmentation du nombre de patients, par l'arrivée aléatoire des patients ainsi que par la difficulté de connaître à l'avance toutes les tâches de soin à prodiguer à un patient [LaCalle et al., 2010]. Les patients qui subissent des temps d'attente élevés aux urgences ont un plus grand risque de décès, un séjour hospitalier plus long et sont moins satisfaits de leurs soins. En l'état, les ressources en personnel de santé (e.g. personnel infirmier, médecins) mais aussi les ressources fixes (e.g. lits) ne sont pas en adéquation si le fonctionnement global n'est pas amélioré.

Ces dernières années, le nombre d'études portant sur les performances des systèmes des services des urgences a augmenté. Conscients de la problématique, les scientifiques étudient des approches susceptibles d'apporter des améliorations.

Les analyses sur l'efficacité des SUA constituent une partie abondante de la littérature. Deux types d'analyses sont généralement utilisés, elles ont pour objectif de déterminer les ressources en tension pour établir des plans de dimensionnement prévisionnel des ressources ou de gestion des patients. Le premier utilise des outils de simulation à partir d'un modèle d'un SUA, et explore des scénarios qui induisent des situations de tensions. Il est utilisé pour anticiper la surcharge du système en étudiant/analysant son comportement. Le deuxième exploite les données passées, si elles peuvent être recueillies, pour analyser les performances historiques du service. Il aide à prendre des décisions sur l'affectation des ressources et du personnel. Les deux types peuvent permettre aussi d'évaluer l'efficacité de stratégies d'allocation de ressources ou de gestion des priorités à accorder aux différents degrés d'urgence des patients.

Ces études sont utiles pour aider à anticiper le dimensionnement amont en ressources mais ne traitent pas du fonctionnement quotidien. Il s'agit aussi de mettre en œuvre des outils d'aide à la décision pour le fonctionnement quotidien d'un SUA sachant que les ressources sont fixées. Compte tenu de l'état actuel du système et des ressources disponibles, l'aide à la décision en temps réel vise à offrir une aide pour l'ordonnancement des tâches de soin qui sont induites par le flux de patients arrivant au SUA. Relativement peu d'études dans la littérature traitent de l'ordonnancement des tâches de soin.

Dans le cadre de l'ANR OIILH, nous avons étudié et proposé des solutions pour le problème d'ordonnancement des tâches de parcours de soins des patients dans le service des urgences du CHU Jeanne de Flandres de Lille. Il s'agit de prendre en considération les contraintes de qualité de service, la disponibilité des ressources à chaque instant, les capacités des lieux ainsi que l'arrivée non programmée des patients.

Nous avons étudié les différentes organisations et procédures de traitement des patients selon les pays ([Kiriş et al., 2010; Luscombe et al., 2016; Lee et al., 2020]). Dans ces études, nous avons constaté que les règles de fonctionnement des SUA diffèrent d'un pays à un autre, mais aussi qu'au sein d'un système de santé d'un pays les SUA ne fonctionnent pas exactement de la même façon. Le fonctionnement quotidien d'un SUA dépend de l'organisation de chaque SUA qui est lui-même immergé dans l'organisation générale de l'hôpital. Par conséquent, aucune solution proposée ne peut être utilisée directement et le fonctionnement du service doit être observé en premier et ensuite modélisé.

Pour modéliser le fonctionnement du service des urgences du CHU Jeanne de Flandres de Lille, nous avons réalisé plusieurs séjours au SUA guidés par deux médecins urgentistes.

Le SUA est structuré en deux circuits, le circuit court (CC) dédié aux plus faibles degré d'urgence dans la journée et le circuit long (CL) qui fonctionne 24h/24h. Nous avons étudié l'organisation du SUA et observé les parcours des patients, les tâches des personnels et l'allocation des ressources matérielles.

CONTRIBUTIONS DE LA THÈSE

L'objectif est de réduire les durées de séjour aux urgences en utilisant au mieux les ressources disponibles (lieux et ressources médicales), en tenant compte de la priorité de chaque patient et de son parcours. Les contributions de cette thèse sont organisées en trois parties :

Approches de résolution dans le cas statique :

Dans le cas statique du problème, nous supposons que le nombre des patients et leurs parcours dans le circuit sont connus au début de la planification. Les durées des tâches de traitements et de déplacement sont connues et sont non préemptives. L'étude du cas statique est essentielle, elle permet de modéliser le problème et d'avoir des solutions de références qui serviront de base de comparaison pour le cas dynamique.

Pour simuler les parcours patients dans le SUA, nous déterminons l'enchaînement des tâches internes et externes. Nous avons modélisé une vingtaine des parcours patients les plus fréquents et estimé la durée des tâches de soins. Nous avons développé un générateur d'instances. Il est générique et peut être utilisé pour étudier le circuit d'urgences de jour (circuit court, CC) et le circuit d'urgences qui fonctionne 24h/24h (circuit long, CL).

À partir des observations sur le terrain, nous avons constaté que certaines règles métiers sont utilisées pour gérer les patients, leur parcours au sein du SUA et à l'extérieur du SUA. Nous avons implémenté l'approche *Practitioners' Rules Heuristic* (PRH) en intégrant deux règles métiers, elle construit de premières solutions réalisables en respectant les contraintes de ressources.

Nous avons proposé un modèle mathématique basé sur la Programmation Linéaire en Nombres Entiers (PLNE) pour le circuit court. La résolution exacte du PLNE nous a permis d'avoir des solutions de référence optimales sur de petites instances et des solutions réalisables sur des instances plus importantes. Cependant pour certaines instances le solveur ne trouve pas de solution réalisable en des temps raisonnables.

Les approches heuristiques offrent de nouvelles combinaisons de principes de résolution de problèmes qui peuvent utiliser des informations spécifiques au problème et être adaptées pour produire des solutions de haute qualité dans un délai adapté à une mise en œuvre en temps réel. Nous avons élaboré deux approches heuristiques.

La première utilise des méthodes de voisinage basées sur la destruction/construction de la solution et la deuxième est une méta-heuristique ALNS (*Adaptive Large Neighborhood Search*). L'approche heuristique *Iterative Destruction Construction Heuristic* (IDCH)

que nous avons élaborée utilise des méthodes de voisinage basées sur la destruction/construction partielle ou complète d'une solution en cours, la solution initiale étant fournie par l'heuristique PRH. Les résultats montrent que les solutions optimales sur les petites instances sont atteintes et que nous obtenons des solutions de bonne qualité dans des temps raisonnables. Un article est soumis dans une conférence internationale [Mezouari et al., 2023].

L'approche ALNS a été élaborée pour obtenir des résultats de meilleure qualité que ceux obtenus avec IDCH dans des temps de calcul plus courts grâce à un paramétrage adapté, mais aussi pour pouvoir être en adéquation avec le contexte dynamique des tâches de soins dans le SUA.

Approches de résolution dans le cas dynamique :

Dans le cas dynamique, les détails concernant le patient et les tâches de soins à effectuer ne sont pas connus avant l'arrivée effective du patient. Le temps réel nécessaire à l'accomplissement d'une tâche de soins est inconnu jusqu'à ce que la prise en charge soit terminée. Nous proposons une approche d'ordonnancement prédictive/réactive pour intégrer l'environnement dynamique du SUA induit par les événements. Nous avons identifié différents événements qui nécessitent d'adapter le planning/ordonnancement des tâches comme (arrivée d'un nouveau patient, sortie d'un patient, durée d'une tâche réalisée différente de celle prévue, modification du parcours d'un patient après une consultation médicale). Dans l'ordonnancement prédictif-réactif, les deux phases principales sont la prédiction et la réaction. La phase prédictive consiste à construire un planning/ordonnancement sur la base des données connues ou prévues. La phase réactive consiste à recalculer le planning/ordonnancement à partir de la date d'observation de l'événement à considérer tout en modifiant le moins possible les affectations des ressources afin de ne pas perturber le prévisionnel pour l'équipe médicale.

L'approche ALNS que nous avons élaborée dans le cas statique a été conçue pour avoir des temps de calcul courts et recalculer un planning/ordonnancement à partir d'un instant t quelconque.

Avoir des temps de calcul courts est nécessaire pour être compatible avec le fonctionnement réactif du SUA. La destruction partielle d'une solution à partir d'un instant t permet la mise à jour des données en fonction des événements possibles présentés ci-dessus. Nous avons conçu et élaboré l'ALNS pour pouvoir l'utiliser à la fois dans la génération du planning initial (prédictif) et pour la replanification (réactif). L'algorithme dynamique répond à tout changement de situations qui nécessite une nouvelle décision.

Aval du SUA, planification d'interventions chirurgicales :

Le SUA est générateur de flux de patients pour les autres services de l'hôpital. Lors d'un passage aux urgences, il peut être détecté qu'un patient nécessite une intervention chirurgicale dans les mois/semaines qui suivent son passage aux urgences. De fait, cela génère un flux de patients électifs pour les opérations chirurgicales. Parallèlement à la modélisation du SUA, nous avons élaboré une approche de résolution pour cette problématique en aval du SUA à partir de données disponibles dans la littérature ([Riise et al., 2016]).

Le service de chirurgie représente une activité essentielle pour la majorité des hôpitaux, générant environ deux tiers des revenus de l'hôpital (Denton et al., 2007). Les interventions chirurgicales représentent 52% de tous les motifs d'entrées à l'hôpital selon (Gupta, 2009).

Il s'agit de planifier un nombre maximum d'interventions chirurgicales tout en tenant compte des ressources limitées et des contraintes de disponibilité.

Le problème de planification d'interventions chirurgicales est dit "problème d'admission" dans la littérature. Les interventions chirurgicales de centaines de patients électifs doivent être programmées plusieurs mois/semaines avant la date de l'opération. Le niveau d'urgence est pris en compte pour chaque patient (date au plus tôt et date au plus tard pour l'intervention).

L'approche méthodologique est similaire à celle utilisée pour traiter la planification/l'ordonnancement de tâches de soin de SUA.

Nous avons proposé une heuristique 2PSC-CHI en deux phases basée sur les idées des heuristiques constructives utilisées dans la littérature pour traiter des problèmes de type RCPSP (*Resource-Constrained Project Scheduling*) et l'utilisation d'un opérateur de voisinage local (Arezki et al., 2020). Comme les autres approches proposées dans la littérature, cette heuristique ne permettait pas de planifier toutes les interventions chirurgicales.

Nous avons procédé à l'élaboration d'une approche ALNS. Les expériences numériques menées sur des instances de référence de la littérature ont montré que l'ALNS obtient des résultats strictement meilleurs que ceux de la littérature. Pour toutes les instances du jeu test de référence, elle réussit à placer toutes les interventions. Les résultats sont publiés dans la revue internationale *Engineering Applications of Artificial Intelligence* (Mezouari et al., 2022).

Nos travaux permettent une interaction entre le SUA et l'aval. À partir du moment où une future intervention est détectée au niveau du SUA, le traitement du problème du planning d'admission pourrait être utilisé comme outil d'aide à la décision. En considérant le patient comme patient électif, il serait possible de déterminer si, compte tenu du planning actuel de chirurgie, il est oui ou non possible de le prendre en charge ou s'il faut orienter le patient vers un autre hôpital.

PRÉSENTATION DE LA THÈSE

Le reste de manuscrit est structuré comme suit.

Le chapitre 2 présente une étude globale sur les problèmes d'optimisation dans la logistique hospitalière. Nous présentons d'abord quelques problèmes d'optimisation qui se posent au sein d'un établissement hospitalier sans vouloir être exhaustif. Nous présentons ensuite une synthèse de la littérature relative aux problèmes d'optimisation dans les services des urgences (SU). Nous présentons aussi les différents types de problèmes de planification dynamique étudiés dans la littérature. Nous présentons enfin les problèmes de planification d'interventions chirurgicales.

Le chapitre 3 est consacré à notre étude sur le problème de planification dans le service d'urgence adulte du CHU Jeanne de Flandres de Lille. Nous présentons la

structure générale du service avec les deux circuits court et long et les différentes caractéristiques du problème. Nous présentons ensuite la formulation mathématique du problème. Nous présentons aussi les approches de résolution proposées pour le cas statique et dynamique et les différents résultats numériques.

Le chapitre 4 est consacré à notre étude sur le problème de planification des interventions chirurgicales. Dans ce chapitre, nous présentons la définition du problème et les approches de résolution proposées. Nous présentons aussi les différents résultats numériques.

Dans chapitre 5, nous terminons par une conclusion générale et des perspectives de nos travaux de recherches.

DOMAINE DE L'ÉTUDE

SOMMAIRE

2.1	Introduction	13
2.2	Problèmes d'optimisation dans la logistique hospitalière	14
2.3	Les problèmes d'optimisation dans les services des urgences (SU)	16
2.3.1	Triage et gestions de flux des patients	17
2.3.2	Allocation des ressources	19
2.3.3	Affectation de personnel	20
2.3.4	Planifications des tâches	21
2.4	Les problèmes d'optimisation dynamique	22
2.5	Les problèmes de planification des intervention chirurgicales	26
2.6	Conclusion	32

2.1 INTRODUCTION

La prestation de soins de santé est réalisable par un ensemble d'infrastructures, de ressources et d'activités qui sont mobilisées pour fournir des soins et des services. L'hôpital ne peut pas fonctionner en îlot dans ses activités quotidiennes. La coordination et le partage des activités avec d'autres institutions et acteurs du secteur de la santé sont à encourager pour atteindre deux objectifs : des soins complets et continus pour le patient et l'optimisation des coûts. L'ensemble des infrastructures, des ressources et des actions mobilisées pour fournir des services répondant aux besoins de la population constitue la base des préoccupations en matière de santé. L'hôpital doit harmoniser et partager son travail avec d'autres structures et intervenants du domaine de la santé. La qualité des soins passe également par l'amélioration de l'efficacité des installations et la mise en œuvre de systèmes de gestion efficaces.

Nous présentons d'abord dans la section 2.2 quelques problèmes d'optimisation qui se posent au sein d'un établissement hospitalier sans vouloir être exhaustif. L'objectif est de montrer la variété des problématiques et les interactions entre elles.

Nous présentons dans la section 2.3 une synthèse de la littérature relative aux problèmes d'optimisation dans les services des urgences (SU). Les principaux points présentés concernent des études sur les problèmes d'optimisation au sein du service des

urgences et les différentes méthodes qui ont été explorées pour améliorer le système. Nous présentons les différents types de problèmes de planification dynamique étudiées dans la littérature car nous devons envisager l'aspect dynamique de la planification des tâches de soins dans le service des urgences allons tenir compte de la dynamique du SUA de Lille.

Nous présentons enfin dans la section 2.5 les problèmes de planification d'interventions chirurgicales. En effet, le service des urgences est une source de flux de patients pour les interventions chirurgicale. Il y a interaction entre services et un lien amont/aval entre le service des urgences et l'organisation des opérations chirurgicales.

2.2 PROBLÈMES D'OPTIMISATION DANS LA LOGISTIQUE HOSPITALIÈRE

Ces dernières années, les études sur le fonctionnement des systèmes de soins de santé se sont multipliées. Nous pouvons classer les problèmes d'optimisation par rapport à trois niveaux de décision. Le Tableau 2.1 donne les trois niveaux et les horizons temporels associés l'horizon de chaque niveau.

Niveau de décision	Horizon de décision	Exemple de problème traité
stratégique	plusieurs années, année	taille des effectifs, composition du personnel
tactique	année, mois, semaines	plannings à moyen terme/planning cyclique
opérationnel	semaine, jours	planning journalier, affectation des ressources aux tâches de traitement pour chaque patient

Tableau 2.1: La classification par niveau de décision

PLANIFICATION DE PERSONNEL INFIRMIER

La planification doit tenir compte des différents niveaux d'expertise requis pour des fonctions telles que celles d'infirmier anesthésiste, d'infirmier de bloc opératoire, d'infirmier puéricultrice et de cadre de santé. De nombreuses éventualités doivent être prises en compte. La composition du personnel infirmier d'une unité de médecine interne peut être différente de celle d'une unité chirurgicale ou encore d'un service des urgences, ce qui peut impacter le processus de planification. La littérature sur ce sujet, connu sous le nom de *nurse rostering*, est abondante. Le défi consiste à concevoir un emploi du temps pour toutes les infirmiers d'un service tout en respectant diverses exigences (heures du service, horaires de travail des infirmiers, compétences spécifiques de certains infirmier, contingent minimum).

PLANIFICATION AU SEIN DE SERVICES D'IMAGERIE

Un examen dans un service d'imagerie est une procédure qui nécessite l'utilisation d'équipements spécifiques et de ressources humaines qui dépendent de la particularité

du patient. Le type d'équipement utilisé peut être transporté ou non. Deux examens équivalents n'ont pas toujours la même priorité. Les examens doivent être classés comme programmés ou non programmés. Le problème de planification des examens consiste à établir un planning avec des examens prévus à l'avance mais aussi à insérer les examens non prévus en prenant en considération les degrés d'urgence. En effet, un flux de patients provient des urgences. La disponibilité des équipements est liée à la disponibilité du personnel qui les entretient et qui les fait fonctionner. Elle est évidemment liée à celle du personnel qui assure les soins directs (les examens) aux patients et leur transport. Il s'agit d'organiser les examens en fonction du temps de disponibilité, du lieu et du personnel disponible.

PLANIFICATION DE TOURNÉES

Il est essentiel de renforcer le cadre organisationnel des soins médicaux à domicile et de l'hospitalisation à domicile. Les enjeux sont considérables et il est urgent d'améliorer l'efficacité de ces systèmes de soins, bien moins coûteux que l'hospitalisation. L'un des enjeux de la gestion du parcours du patient dans ces structures est de coordonner les déplacements et les présences des acteurs qui doivent tous se trouver au même endroit au même moment pour certains soins (fenêtre de temps). La planification des actes de soins ou d'aide à la vie quotidienne dans les tournées de visites fait l'objet des défis rencontrés. Le transport de médicaments, d'échantillons ou de documents entre les établissements (itinéraires des véhicules), les soins à domicile (itinéraires du personnel) et de nombreuses autres difficultés liées à la santé dépendent tous d'une planification de tournées. L'importance de déterminer les itinéraires les plus rapides et les plus rentables ne fait que croître au fur et à mesure que le réseau s'étend. Les caractéristiques classiques du problème de routage sont présentes, telles que la limitation du temps et la nécessité de gérer une flotte de véhicules, mais il existe également de nouvelles contraintes introduites par le domaine de la santé, telles que la nécessité de prendre en compte les préoccupations relatives aux compétences du personnel lors de la conception d'itinéraires qui les incluent.

PLANIFICATION POUR SERVICES DES URGENCES

Un patient peut arriver à l'improviste ou être orienté vers le service des urgences par un médecin ou un autre établissement. Il est possible d'ajuster le nombre de personnes travaillant dans un service des urgences donné en tenant compte des événements futurs qui peuvent être prédits avec un certain degré de précision (week-ends particuliers, événements festifs connus à l'avance). En cas d'événements importants et inattendus, l'hôpital doit être prêt à mobiliser rapidement des ressources supplémentaires, soit en puisant dans ses propres ressources, soit en intégrant d'autres ressources dans son fonctionnement. Différents types de problèmes sont rencontrés dans la littérature :

- *Staffing problems* concernent les décisions de dimensionnement des effectifs de professionnels de santé dans les services des urgences pour assurer une qualité de

service donnée sous contraintes budgétaires. La majorité des publications sur les *Staffing problems* évaluent des décisions à long terme ;

- *Rostering problems* concernent l'affectation de ressources à des intervalles de travail (shift). Un niveau de soins requis doit être couvert durant chaque période de la planification ;
- *Task Scheduling problems* concernent l'allocation des ressources (personnels et/ou matériels) aux patients pour réaliser des tâches de soins spécifiques.

DIMENSIONNEMENT ET PLANNING DE BLOCS OPÉRATOIRES

L'objectif du dimensionnement est de trouver le bon équilibre entre la charge de travail et la quantité de ressources disponibles (humaines et matérielles). Dans le cas du dimensionnement de blocs opératoires, la modélisation en salles d'opérations doit tenir compte des compétences spécifiques des médecins et des infirmiers qui y travaillent mais aussi du matériel qui s'y trouve, les blocs ne sont pas équivalents. La gestion des blocs est une part importante du budget total d'un hôpital tant au niveau des coûts que des recettes. Le taux d'utilisation d'un bloc influe sur le coût.

Dans le cadre de la planification d'une intervention chirurgicale, il est nécessaire de s'assurer de la disponibilité du personnel médical. Un chirurgien, un anesthésiste et une équipe d'infirmiers spécialisés sont nécessaires pour chaque intervention chirurgicale. Pour un niveau de décision opérationnel, il est essentiel d'inclure d'autres éléments cruciaux, tels que la préparation (stérilisation) et la maintenance du bloc par des professionnels qualifiés. Bien que les unités de soins intensifs (USI) puissent être subdivisées par spécialité, la disponibilité d'une salle de surveillance interventionnelle (SSI) et/ou de lits d'USI peut constituer aussi une ressource cruciale pour le bloc opératoire. Deux approches sont possibles pour élaborer les plannings une fois que le dimensionnement a été effectué. La première consiste à concevoir des groupes d'opérations chirurgicales pour chaque salle sur une période donnée sans tenir compte des équipes, puis, dans un deuxième temps, à former les équipes et à déterminer la séquence optimale pour les groupes d'interventions dans chaque salle. La deuxième approche consiste à organiser les processus à réaliser par chaque équipe dans l'ordre dans lequel ils seront réalisés, d'abord par salle et ensuite sur un horizon temporel. Chaque intervention chirurgicale est un acte unique réalisé par une équipe sur un patient particulier. Il n'y a pas de durée strictement fixée pour les interventions. Néanmoins une estimation raisonnable peut être faite par type d'opération. C'est particulièrement le cas pour les opérations courantes/standards réalisées sur les patients électifs. Le bloc opératoire doit souvent faire face à des situations d'urgence, ce qui signifie que des salles, des opérations et des équipes risquent de devoir être transférées à la dernière minute. Pour que le système de salle d'opération puisse faire face à des événements inattendus, il doit être en mesure de réorganiser et/ou de planifier des périodes de réserve.

2.3 LES PROBLÈMES D'OPTIMISATION DANS LES SERVICES DES URGENCES (SU)

Les services des urgences sont un élément clé du système de santé. Ils permettent d'accéder rapidement aux soins pour évaluer, stabiliser et traiter les patients gravement malades ou blessés.

Les services des urgences, en France ou ailleurs, font face à un certain nombre de problèmes. La congestion des urgences est un phénomène observé au niveau mondial qui a des répercussions régionales (Jayaprakash et al., 2009). Elle est source de plusieurs problèmes de fonctionnement, tels que des patients traités dans les couloirs, une durée de séjour (*Length of Stay*, LOS) et des temps d'attente excessifs, des patients qui quittent le service sans traitement, ou même des erreurs médicales qui peuvent introduire une augmentation de la mortalité.

Selon la définition donnée par l'American College of Emergency Physicians (ACEP) en 2002, la congestion est une situation dans laquelle le besoin constaté de services des urgences dépasse les ressources disponibles. Cette situation se produit lorsqu'il y a plus de patients que de places de traitement disponibles et que le temps d'attente dépasse une période raisonnable. La congestion se caractérise par le fait que les patients sont surveillés dans des zones autres que les zones de traitement (e.g. les couloirs) en attendant des lits de traitement ou des lits d'hospitalisation.

La Fédération française des Observatoires Régionaux des Urgences (FEDORU) a défini la congestion aux SU comme une inadéquation entre la demande (flux entrant de patients et besoins d'hospitalisation), les moyens disponibles au niveau des urgences (personnel, plateau technique et structure des urgences) et de l'aval des urgences (lits d'hospitalisation disponibles et fluidité du transfert vers d'autres services).

Des études descriptives dans la littérature médicale définissent la congestion (Hwang et al., 2004) et en examinent les causes et les effets (Richardson et al., 2009; Hoot et al., 2008; Asplin et al., 2003). Elles proposent des modèles pour mesurer la congestion (Hwang et al., 2011; Higginson, 2012).

L'intérêt pour ce phénomène se reflète dans le nombre de publications sur la congestion des urgences parues au cours de la dernière décennie. Nous présentons des travaux relatifs au problème d'optimisation de la chaîne logistique au sein de services d'urgences. Ces problèmes ont de nombreuses caractéristiques spécifiques qui dépendent des cas pratiques abordés par l'étude, ainsi que la localisation de l'étude dans la chaîne logistique entre amont, intra, et aval. Nous présentons quatre types de problème.

2.3.1 *Triage et gestions de flux des patients*

Des soins d'urgence de qualité nécessitent une gestion du flux des patients. Il s'agit de coordonner les ressources, le personnel et les systèmes pour voir et traiter les patients rapidement. La gestion du flux des patients est complexe et implique une connaissance approfondie du service des urgences et des demandes des patients. [Forero et al., 2010] estiment qu'une augmentation de 20 à 30% de la mortalité est associée au blocage de l'accès et à la congestion.

La gestion du flux des patients garantit des soins aux patients en temps opportun. Cela implique que les patients soient vus dans l'ordre, par le bon praticien et avec les bons tests et traitements. Elle implique également une gestion efficace du personnel et des ressources du service des urgences. Le blocage de l'accès et la durée prolongée du séjour aux urgences sont également des facteurs prédictifs de l'augmentation de la durée du séjour des patients hospitalisés (Liew et al., 2003; Richardson, 2002). La réduction de la durée du séjour à l'urgence est primordiale pour atténuer les effets associés au blocage de l'accès et à la congestion.

Le triage est le point de départ de la gestion du flux des patients. Il s'agit d'évaluer l'état du patient et de choisir le bon traitement. Cela permet de s'assurer que les patients sont vus et traités dans l'ordre d'arrivée. Il consiste à trier les patients en fonction de la gravité de leur affection et de l'urgence de leur besoin de soins médicaux.

Triage :

[Green et al., 2006] ont présenté le système de triage d'un service des urgences comme une illustration typique d'une file d'attente de priorité.

[Xu et al., 2014] ont présenté une stratégie permettant de regrouper les patients en fonction des procédures médicales dont ils ont besoin. Ils utilisent des méthodes hiérarchiques et l'algorithme k-means.

[Afilal et al., 2016] ont proposé une nouvelle classification pratique des patients des urgences qui prend en compte les classifications CCMU (Classification Clinique des Malades aux Urgences) et GEMSA (Groupes d'Etude Multicentrique des Services d'Accueil). L'objectif de cette nouvelle classification est de regrouper les patients qui demandent des ressources similaires et/ou présentent un comportement similaire dans le service des urgences.

[Zhang et al., 2019a] ont étudié une technique de triage modifiée qui identifie les patients admissibles au triage en fonction de leur ressenti de leurs temps d'attente, de leurs pathologies et de la gravité de leur maladie.

Gestions de flux des patients :

Le gestionnaire du flux des patients coordonne les soins après le triage. Cela implique de consulter le bon praticien, de prescrire les bons tests et traitements et de faire sortir le patient rapidement. Le gestionnaire du flux des patients doit également assurer le suivi des traitements et l'orientation des patients.

La théorie des files d'attente est une méthode régulièrement utilisée pour aborder ce genre de problèmes.

[Au et al., 2009] et [Hoot et al., 2009] utilisent la théorie des files d'attente et la simulation pour construire des modèles en vue de prédire l'occurrence d'une situation de congestion dans une fenêtre de 6 heures. Les auteurs produisent un modèle de multi-régression lissée pour prévoir les admissions aux urgences. L'analyse prévisionnelle est intégrée dans un tableau de bord en ligne qui peut être utilisé en temps réel par les responsables des urgences. La capacité de ces modèles de prévision est d'envisager l'avenir immédiat. L'objectif est de déterminer si un point de crise sera atteint dans

les prochaines heures, ce qui permet aux responsables des urgences d'envisager une stratégie d'adaptation en temps réel.

[Seshaiah et al., 2011] combinent la théorie des files d'attente avec la simulation pour examiner la congestion entre les services des urgences et les services d'hospitalisation. [Wang et al., 2011] ont présenté un modèle de théorie des files d'attente pour le flux des patients et l'allocation des lits dans le cadre des soins aigus. Les auteurs considèrent trois flux d'admission : les urgences, les admissions électives et les transferts d'un autre établissement de soins de santé. Les arrivées des patients sont modélisés à l'aide d'un processus de Poisson non stationnaire. Le résultat détermine la meilleure allocation d'un nombre fixe de lits entre les différents services d'hospitalisation.

[Duguay et al., 2007] ont réalisé une étude avec une simulation à événements discrets d'un service des urgences canadien afin de déterminer des recommandations pour réduire le temps d'attente des patients et d'améliorer la performance des services et le taux de passage.

[Salmon et al., 2018] ont réalisé une étude de la littérature sur l'utilisation de la simulation à événements discrets dans les services des urgences.

2.3.2 Allocation des ressources

Les problèmes d'allocation des ressources concernent essentiellement les décisions concernant la taille appropriée de l'effectif dans les services des urgences pour assurer une qualité de service donnée sous des contraintes budgétaire. La majorité des publications sur le problème évaluent des décisions à long terme. Il peuvent être vu comme des problèmes de dimensionnement.

L'allocation des ressources implique une réflexion approfondie sur les ressources dont dispose le service des urgences et sur la manière dont ces ressources sont utilisées pour répondre aux besoins. Il s'agit de la répartition du personnel, des outils et des fournitures.

Les ressources doivent être réparties de manière à optimiser l'efficacité et assurer la sécurité des patients. Le service des urgences doit disposer d'un personnel en nombre suffisant, des outils et des fournitures nécessaires, pour assurer des soins rapides et efficaces. Les parcours de soin des patients, notamment leur âge, leurs antécédents médicaux et la gravité de leur affection, doivent également être prises en compte.

[Marchesi et al., 2020] ont traité le problème de l'allocation et de l'affectation des médecins dans un service des urgences. Ils ont cherché à déterminer le nombre de médecins requis pour chaque équipe, pour chaque jour de l'horizon de planification. Le problème d'allocation des ressources et le problème d'affectation sont conjoints. Les auteurs ont proposé un modèle de programmation stochastique généralisé. L'approche de résolution est combinée avec un modèle de simulation à événements discrets tenant compte des incertitudes liées à l'arrivée des patients. Les tests ont été menés sur des données réelles provenant des urgences de deux hôpitaux.

[Chouba et al., 2022] ont présenté un système d'aide à la décision à moyen terme qui traite la planification et l'allocation des ressources dans les urgences du centre hospitalier de Troyes, en France. Ils ont établi 145 scénarios. Deux méthodes d'analyse multicritères

sont utilisées pour classer et sélectionner les meilleures alternatives. Les auteurs ont développé un outil d'évaluation des performances basé sur la simulation à événements discrets. L'objectif de l'étude est de minimiser le temps d'attente et le séjour des patients hospitalisés en considérant le coût des ressources.

[Chen et al., 2016] ont proposé de résoudre les problèmes d'allocation des ressources médicales aux urgences en utilisant un modèle d'optimisation stochastique multi-objectifs. Pour mieux évaluer les valeurs de performance attendues de chaque solution produite, un modèle de simulation à événements discrets du flux des urgences est construit. En suivant l'évolution de l'ensemble de Pareto à mesure que la recherche se poursuit, les auteurs ont évalué la convergence de l'algorithme proposé.

[Izady et al., 2012] ont utilisé une approche similaire qui itère entre un modèle de théorie des files d'attente et une simulation sur des périodes successives pour concevoir des niveaux de personnel appropriés.

[Sinreich et al., 2012] ont combiné la simulation et la programmation linéaire (PL). L'objectif est de réduire les temps d'attente des patients grâce à un rééquilibrage des ressources.

[Testik et al., 2012] ont utilisé une série de files d'attente pour modéliser les arrivées. Les variations quotidiennes et les horaires des arrivées sont identifiées à l'aide d'une méthode de classification et de régression en deux étapes. Le modèle est utilisé pour estimer les niveaux de personnel nécessaires pour atteindre les durées d'attente et de traitement cibles.

[Green et al., 2006] ont modélisé les services d'urgence comme une file d'attente, des périodes indépendantes qui suivent la loi de poisson sont utilisées. Leur modèle calcule les niveaux du personnel qui répondent aux variations de la demande avec un modèle de simulation à événements discrets afin de valider les résultats.

2.3.3 Affectation de personnel

Nous pouvons définir ce problème comme un problème d'affectation de ressources à des quarts de travail (*shift*). Un niveau de soins requis doit être couvert dans chaque période de la planification. Le problème a généralement deux types de contraintes :

- les contraintes dures : concernent la charge de travail et les règles qui régissent les conditions de travail du personnel. Elle sont généralement imposées par la législation et la direction de l'hôpital ;
- les contraintes souples : mesurées à l'aide de variables de décision et des pénalités associées sont ajoutées à la fonction objectif.

L'augmentation du nombre d'hospitalisations a pesé sur des ressources déjà limitées. Les hôpitaux ont du mal à répondre à la demande et à maintenir des effectifs suffisants dans les services des urgences. Ces problèmes sont abordés dans la littérature et des revues de la littérature (*surveys*) donnent une synthèse des problématiques et approches de résolution ([Cheang et al., 2003],[Ernst et al., 2004], et [Burke et al., 2004]). [Burke

et al., 2009] ont proposé une modélisation basée sur la programmation linéaire en nombres entiers (PLNE). Ils utilisent cependant un algorithme génétique hybride pour la résolution.

[Ferreira et al., 2012] ont abordé le problème pour les personnels médicaux et les infirmières pour un hôpital en utilisant une formulation de problème de satisfaction de contraintes et un algorithme de *backtracking*.

[Bard et al., 2005] ont utilisé la planification réactive pour modifier quotidiennement les listes des infirmières afin de tenir compte des absences, des urgences et des déséquilibres de la demande. Une formulation mathématique est utilisée pour produire une liste de base de six semaines avec un horizon mobile. Les auteurs ont replanifié les effectifs sur un horizon court (prochaines 24h) lorsque de nouvelles informations sont disponibles concernant les déséquilibres entre l'offre et la demande. La replanification réactive est réalisée à l'aide d'une heuristique basée sur l'algorithme de *branch and price*.

Les systèmes de soins de santé fonctionnent dans un environnement dynamique et imprécis. Des événements imprévus peuvent entraîner des perturbations complexes des horaires. La plupart des chercheurs partent du principe qu'il y a toujours une équipe de remplacement pour les personnels absents. Cependant, il peut être coûteux de toujours disposer d'une équipe de remplacement.

2.3.4 Planifications des tâches

Une utilisation plus efficace des ressources pourrait permettre d'améliorer le fonctionnement des services des urgences. Des salles peuvent être utilisées de façon inappropriée, ce qui prolonge les temps d'attente pour les autres patients. Les patients peuvent subir des retards en raison de décisions de planification et d'utilisation inefficace de ressources.

Les décisions de planification sont difficiles dans les services des urgences en raison du flux important de patients malades ou blessés. Ils arrivent de manière imprévisible ce qui nécessite une attention dans un environnement au rythme rapide et en permanente modification.

La planification des tâches dans les services des urgences est un problème complexe en raison de la nature dynamique de l'environnement. L'urgence est un environnement sous tension où les tâches doivent être accomplies rapidement et avec précision. Le personnel des urgences doit être en mesure de répondre aux besoins variés des patients et de prioriser les tâches en conséquence. De plus, le personnel doit être en mesure de gérer plusieurs tâches simultanément et d'ajuster son travail aux besoins.

[Kiriş et al., 2010] ont présenté un modèle basé sur une formulation de type machine parallèle avec des arrivées de patients stochastiques et des temps de traitement déterministes. Une heuristique constructive est développée pour affecter les patients aux médecins en fonction de l'heure d'arrivée, de la priorité du patient, des durées des tâches et de la charge du médecin. L'assignation des patients aux médecins au fur et à mesure que des nouveaux arrivants sont enregistrés au triage est mise à jour dynamiquement.

[Daknou et al., 2010; Ajmi et al., 2019] ont proposé une approche multi-agents pour la programmation des patients en utilisant la planification dynamique avec des temps de traitement stochastiques. Le problème d'ordonnancement est résolu par un algorithme génétique.

[Azadeh et al., 2014] ont analysé la programmation des examens des patients dans les laboratoires des urgences en utilisant des critères de triage. Dans son étude, la planification des patients a été modélisée comme un problème open shop flexible avec un modèle linéaire en variables mixtes. L'ordonnancement a été résolu en utilisant un algorithme génétique (AG).

[Luscombe et al., 2016] ont établi deux catégories d'affectation : des lits aux patients et puis des ressources aux tâches, respectivement. Ils ont développé une méta-heuristique basée sur recuit simulé et utilisent un graphe disjonctif des tâches et des règles de priorités afin de minimiser le temps total de soins des patients.

[Lee et al., 2020] ont proposé un algorithme d'apprentissage de renforcement profond, ils ont proposé un modèle mathématique et une modélisation par une chaîne de Markov. Pour chaque ressource ils ont défini une liste des patients qui nécessitent l'utilisation de cette ressource. L'algorithme d'apprentissage détermine la meilleure méthode de choix des patients pour chaque liste parmi plusieurs méthodes définies.

Nous avons constaté que peu de travaux de la littérature traitent spécifiquement de la planification des tâches de soins au sein d'un service des urgences.

En synthèse, l'objectif est de minimiser le temps de séjour (*Length of Stay*, LOS) du patient dans le service des urgences en prenant en considération la gravité de ce dernier, en affectant un patient à la ressource médicale appropriée au bon moment. Chaque patient est classé par la méthode de triage et un niveau de gravité lui est affecté. La valeur du temps d'attente pour la première consultation (*Door to Doctor Time*, DTDT) médicale peut être différente selon le niveau de gravité du patient.

2.4 LES PROBLÈMES D'OPTIMISATION DYNAMIQUE

Les problèmes de planification dynamique des tâches de soins dans les services des urgences constituent un défi important pour les professionnels de la santé. Les services des urgences doivent être préparés à gérer rapidement et efficacement une grande variété de situations. Il est difficile de prévoir le nombre de patients qui demanderont des soins à un moment donné, en raison de la nature imprévisible des soins d'urgence.

Le système de planification peut modifier les affectations du personnel pour répondre à des besoins variables. Le personnel doit être planifié pour répondre au mieux aux demandes du service à un moment donné. Le problème d'ordonnancement dynamique nécessite de procéder à des affectations et ré-affectations de ressources à différents moments. Il s'agit d'une question délicate qui nécessite une réflexion approfondie sur les ressources disponibles, les tâches à accomplir et les délais à respecter.

Dans la situation dynamique, les décisions de planification doivent être prises sans nécessairement disposer de toutes les données nécessaires pour résoudre le problème. Plusieurs facteurs peuvent contribuer à ce manque de données :

- les tâches se présentent individuellement, par lots ou de façon continue. Lorsqu'on prend des décisions de planification, il faut le faire sans avoir d'information sur les tâches futures ;
- avant et pendant l'exécution d'une tâche, le temps exact qu'il faudra pour la terminer est inconnu. Le temps exact n'est connu que lorsque la tâche est terminée ;
- certaines ressources peuvent devenir non disponibles.

Nous présentons quatre types de problèmes de planification dynamique (Ouelhadj et al., 2009).

2.4.0.1 *Planification entièrement réactive*

Avec une planification entièrement réactive, il n'y a pas de planning prédéterminé et toutes les décisions sont prises en temps réel. Lorsqu'une ressource devient disponible pour traiter des tâches, une règle de répartition détermine quelle tâche dans la file d'attente a la plus haute priorité et sera traitée ensuite (Ouelhadj et al., 2009). La Figure 2.1 décrit les grandes étapes de cette approche de planification.

[Chen et al., 2019] ont étudié le problème de planification multi-projets stochastiques avec contraintes de ressources. Dans cette étude, l'arrivée d'un nouveau projet est inconnue et les durées des tâches de chaque projet sont stochastiques. Les auteurs ont établi 20 règles de priorité pour la sélection des tâches à traiter. Les auteurs ont proposé une méthode hybride heuristique pour résoudre le problème de manière temporelle en divisant l'ensemble du processus de planification en multi-états résolus séparément par les règles de priorité.

[Heger et al., 2016] ont étudié le problème de planification dans les ateliers flow shops avec des arrivées dynamiques des tâches. Les auteurs ont proposé une méthode de résolution basée sur trois règles de priorités pondérées. Les poids sont ajustés dynamiquement.

De nombreuses études ont évalué et comparé les règles de priorité et constaté que leurs performances dépendent de nombreux paramètres, notamment les exigences de performance, la configuration du système et la charge de travail (Durasević et al., 2018; Priore et al., 2014). Les règles de priorité ont deux principales limitations : il n'existe pas de règle de priorité qui soit meilleure que toutes les autres pour tous les critères donnés, et une règle qui donne de très bons résultats pour un critère de performance peut en donner de très mauvais pour un autre.

2.4.0.2 *Planification pro-active robuste*

Cette approche de la planification repose sur la construction de plannings prédictifs. Les principaux facteurs contribuant aux perturbations futures sont étudiés et la construction a pour but d'en tenir compte. Les perturbations sont quantifiées en comparant les taux d'achèvement réels avec ceux prévus initialement, et les problèmes sont résolus en ajustant les temps alloués à certaines tâches (Ouelhadj et al., 2009). Le principal

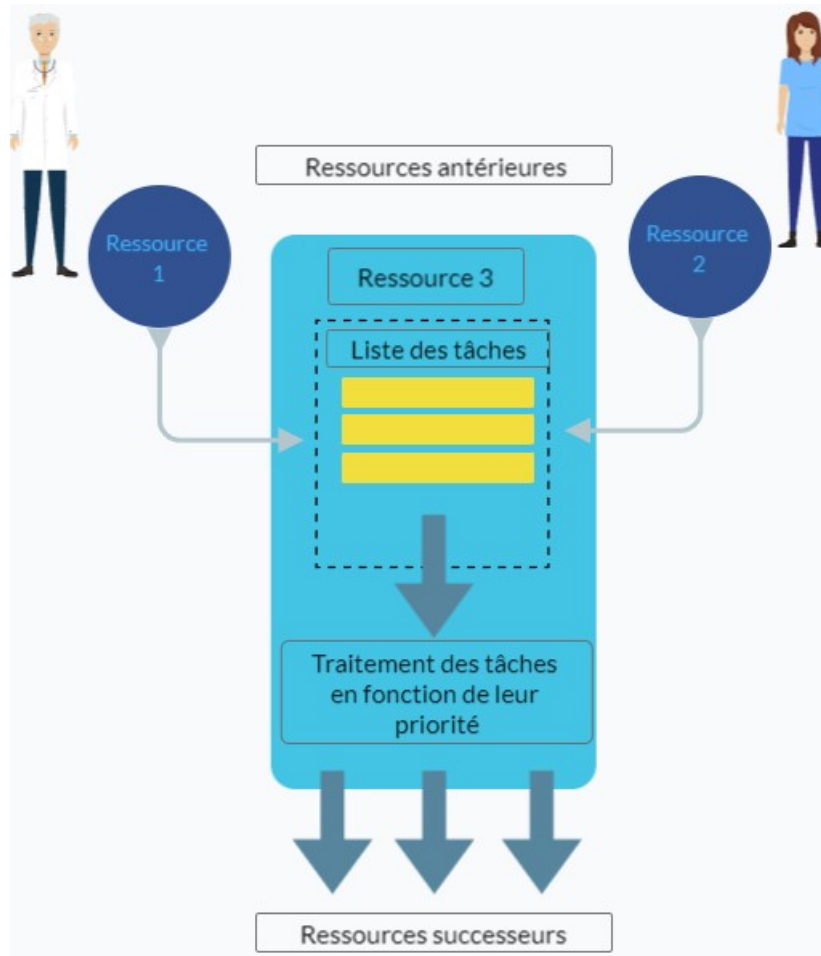


Fig 2.1: Planification entièrement réactive (Bukkur et al., 2018)

désavantage de cette approche est qu'il faut prévoir les perturbations ce qui n'est pas toujours possible. Pour élaborer des méthodes de planification robustes, il faut quantifier la résilience des plannings [Minguillon et al., 2019]. La Figure 2.2 décrit globalement cette approche de planification.

[Mahmoudzadeh et al., 2020] ont étudié le problème de planification des patients avec minimisations du temps d'attente globale pour tous les patients. Les patients sont classés par ordre de priorité en fonction de leur niveau de criticité. Les auteurs ont proposé un modèle d'optimisation déterministe pour l'ordonnancement en supposant que les arrivées des patients sont connues à l'avance pour l'horizon complet. Ils ont considéré une approche robuste avec une valeur nominale pour chaque paramètre de demande et une déviation possible de cette valeur. Les ensembles d'incertitudes sont construits sur la base de données historiques des arrivées de patients de chaque priorité.

[Nouiri et al., 2017] ont étudié le problème de planification des tâches en se basant sur un modèle d'atelier job shop flexible tout en tenant compte des pannes de machines. Pour créer un planning prédictif fiable et stable face à d'éventuelles défaillances imprévues, une approche d'optimisation par essaims de particules en deux phases a été présentée par les auteurs. La première optimise la fonction objectif principale dans

l'hypothèse de l'absence d'interruptions sous des paramètres déterministes. L'objectif de réduction de la durée d'exécution (makespan) guide l'évolution de l'essaim. La deuxième phase affine les solutions sur la base de la mesure de robustesse et de stabilité bi-objectif en tenant compte des pannes aléatoires des machines.

Pour construire des algorithmes d'ordonnancement robuste, il est nécessaire de développer des mesures quantitatives qui fournissent une bonne approximation de la robustesse du planning. Les plannings de base sont des plans de réalisation de planification qui ont été créés à l'avance. En utilisant ces méthodes d'augmentation de la robustesse, l'idée est d'élaborer des plans qui sont plus résistants aux changements. Jusqu'à présent, les études sur la robustesse des plannings ont généralement utilisé soit des mesures directes, obtenues à partir de performances réelles, soit des techniques heuristiques, qui font appel à des mesures de substitution simples (Goren et al., 2007).

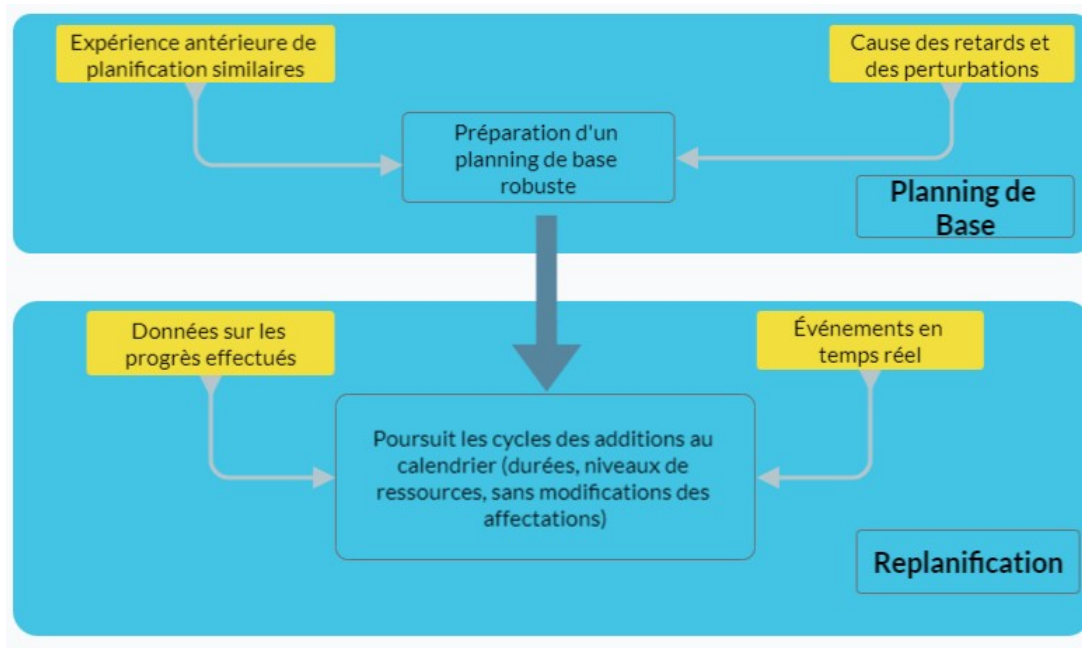


Fig 2.2: Planification pro-active robuste (Bukkur et al., 2018)

2.4.0.3 Planification prédictive-réactive

La méthode de planification prédictive-réactive est l'approche de planification dynamique la plus utilisée dans les systèmes de production.

Comme illustré sur la Figure 2.3, l'idée principale de la planification prédictive-réactive est de construire d'abord un planning prédictif simple, puis de le modifier (replanifier) en fonction des événements réels qui se produisent. Il faut mettre en place une politique et une stratégie de replanification qui définissent le moment, le type d'événements qui déclencheront le modification.

[Daknou et al., 2010; Ajmi et al., 2019] ont proposé une méthode de planification pour un service des urgences. Il s'agit d'une approche multi-agents. Elle est dynamique avec des temps de traitement stochastiques, et réalisée par un algorithme génétique. Un

agent planificateur gère les événements de replanification. La déviation du planning n'est pas mesurée dans l'étude.

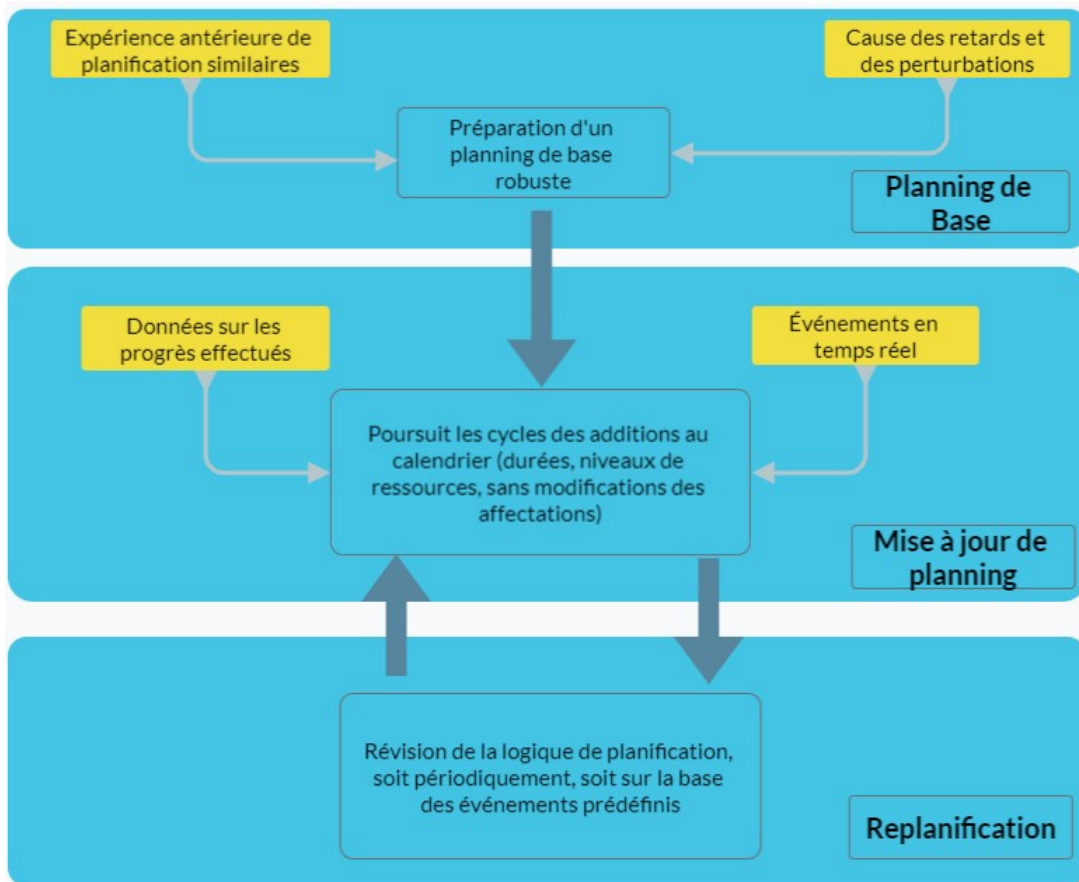


Fig 2.3: Planification prédictive-réactive (Bukkur et al., 2018)

2.4.0.4 Planification prédictive-réactive robuste

L'objectif principal de l'ordonnancement prédictif-réactif robuste est de minimiser les effets négatifs de la perturbation d'un planning initial. Pour créer un planning robuste, une solution est de replanifier en tenant compte de la qualité et de la déviation relative à un planning initial (la stabilité est recherchée). La mesure de la stabilité quantifie les effets indésirables de l'ajustement du planning en mesurant la divergence par rapport au planning prévisionnel (Ouelhadj et al., 2009). La replanification permet d'éviter une déviation significative des performances par rapport au planning initial, de maintenir la stabilité du planning et d'éviter l'impact sur toute tâche de planification dépendante (Moratori et al., 2012).

[Soudi et al., 2019] ont formulé la planification et l'ordonnancement hebdomadaires des opérations chirurgicales comme un problème de planification hybride de type flow shop, en tenant compte des restrictions de capacité des lits de salles et des salles d'opération multiples. Les auteurs ont étudié l'arrivée des patients en situation

d'urgence dans le service et son impact sur le planning électif. Afin de générer un planning de base stable dans la phase prédictive, un modèle de programmation sous contrainte en probabilité (CCP) est développé. Ensuite, un modèle de programmation réactive est proposé. Il vise à préserver la stabilité de la planification initiale dans la phase réactive.

2.5 LES PROBLÈMES DE PLANIFICATION DES INTERVENTION CHIRURGICALES

La planification des interventions chirurgicales a été largement étudiée, et les synthèses de la littérature (Rahimi et al., 2020; Samudra et al., 2016; Erdogan et al., 2011; May et al., 2011; Cardoen et al., 2010; Magerlein et al., 1978) donnent un aperçu de la grande variété des problèmes rencontrés.

L'organisation hospitalière, les préoccupations médicales du patient, les ressources chirurgicales et l'horizon à considérer sont autant d'éléments qui contribuent à la variété des problèmes rencontrés dans la planification d'interventions chirurgicale. Les chirurgiens peuvent être rémunérés à l'acte ou employés par l'hôpital, ou les deux. Dans certains cas, l'hôpital peut attribuer des fenêtres de temps pour les salles d'opération à un service (par exemple, un groupe de chirurgiens orthopédistes) et laisser aux chirurgiens eux-mêmes le choix de répartir ces fenêtres de temps. Dans d'autres cas, les chirurgiens programment leurs propres patients dans des salles d'opération qu'ils réservent quelques semaines à l'avance. Dans d'autres cas encore, à condition que plusieurs chirurgiens aient les compétences chirurgicales pour effectuer une intervention qui peut être réalisée dans certaines salles d'opération, les salles et les chirurgiens sont considérés comme des ressources renouvelables partagées à gérer efficacement pour l'organisation hospitalière. Ces cas peuvent également être hybridés comme c'est le cas pour les organisations publiques/privées qui sont largement rencontrées.

Les problèmes peuvent être classés selon plusieurs axes, tels que les caractéristiques des patients ou la longueur de l'horizon de planification (mois, semaines, jours). Généralement deux grandes catégories de patients sont considérées : les patients électifs ou les patients non électifs. Pour les premiers, les interventions peuvent être planifiées des mois à l'avance en considérant les principaux besoins en ressources médicales sur le long terme, il s'agit du problème dit d'admission dans la littérature. Pour les seconds, les interventions sont soit plus urgentes, soit imprévues puisqu'un diagnostic récent pour un patient peut impliquer une intervention à programmer à court terme (plusieurs semaines) ou immédiatement après un accident.

Dans certaines organisations hospitalières, les services des urgences ont leurs propres salles d'opération ; si ce n'est pas le cas, il faut envisager le partage des salles d'opération entre les deux catégories de patients. Une façon typique de gérer ces deux catégories de patients est de réserver ou de dédier des salles d'opération soit pour les patients électifs, soit pour les patients non électifs, mais cela peut dépendre de l'horizon considéré.

La planification d'admission des patients peut ne prendre en compte que les principaux besoins en ressources médicales pour une intervention, l'objectif étant d'attribuer une date pour l'intervention chirurgicale pour chaque patient, en tenant généralement compte de l'horizon à long terme (mois). Ensuite, un plan pour une ou deux semaines est

établi sur la base, d'une part, du planning existant d'admission des patients et, d'autre part, des interventions plus urgentes des nouveaux patients qui ne sont connues que quelques semaines à l'avance. Enfin, un plan est établi chaque jour pour le jour suivant. Les plannings hebdomadaires et journaliers sont généralement soumis à une reprogrammation dynamique qui doit tenir compte des politiques d'annulation ou de report des patients électifs afin de perturber le moins possible la planification des admissions des patients.

Dans cette étude, nous considérons le cas déterministe, pour les approches non déterministes nous invitons le lecteur à se référer aux travaux récents tels que présentés dans [Zhou et al., 2021]. Dans cette section, nous nous intéressons aux études concernant la planification des interventions chirurgicales par rapport à l'horizon de planification, allant de l'horizon à court terme d'une journée à l'horizon à moyen terme de plusieurs semaines. Les problèmes à court et moyen terme ont de nombreuses caractéristiques spécifiques qui dépendent des cas pratiques traités par l'étude, ainsi que des aspects qui peuvent être considérés dans une perspective à plus long terme.

Problèmes de planification des interventions quotidiennes

[Park et al., 2021] ont proposé un modèle mathématique mixte en nombres entiers et une heuristique basée sur le regroupement des interventions pour un même chirurgien pour la planification des interventions quotidiennes d'un hôpital universitaire coréen. Chaque patient est déjà assigné à un chirurgien. Certaines salles d'opération sont réservées pour des interventions chirurgicales urgentes, en dehors de la portée du plan quotidien. Les préférences des chirurgiens pour les salles d'opération ainsi que les contraintes de la chirurgie coopérative, où plusieurs chirurgiens effectuent une opération simultanément ou séquentiellement, sont considérées. L'objectif est de minimiser le nombre total d'heures supplémentaires, le nombre de salles d'opération utilisées, le nombre de chirurgies non attribuées aux salles préférées et le nombre de chirurgies attribuées aux salles d'opération défavorables.

[Wang et al., 2015] ont proposé deux approches pour résoudre le problème de planification des interventions chirurgicales quotidiennes d'un hôpital universitaire en Belgique. Les interventions à planifier par jour sont connues. Les auteurs ont pris en compte les contraintes de disponibilité des salles d'opération et des chirurgiens, ainsi que le nombre limité de lits de réveil. Certaines interventions doivent commencer plus tôt que d'autres, en fonction des priorités liées au cas du patient (par exemple, les enfants, les diabétiques). La première approche utilise un modèle mathématique mixte en nombres entiers, tandis que la seconde fait appel à la programmation par contraintes. L'objectif principal est de minimiser le *makespan* des interventions planifiées.

[Xiang et al., 2015] ont proposé une approche de solution basée sur l'optimisation par colonies de fourmis (ACO) pour résoudre le problème de planification des interventions chirurgicales quotidiennes du centre médical MD Anderson à Houston, Texas (USA). Les interventions chirurgicales à planifier sur une base quotidienne sont connues. L'objectif est de minimiser le *makespan* en tenant compte de la limitation des ressources (par exemple, le chirurgien, l'anesthésiste, l'infirmière, les lits) nécessaires à la réalisation d'une opération, ainsi que des étapes de pré-chirurgie et de post-anesthésie. [Xiang, 2017] a proposé une extension de l'approche ACO en considérant deux objectifs

supplémentaires : minimiser la variation du temps de travail des ressources et minimiser le temps de travail supplémentaire total de toutes les ressources

[Sier et al., 1997] ont proposé une solution pour un problème d'intervention chirurgicale quotidienne basée sur un algorithme de recuit simulé. Une formulation non linéaire en nombres entiers mixtes pour le problème étudié a été proposée. La fonction objectif est une somme pondérée de toutes les contraintes du problème. La faisabilité de la solution n'est pas garantie, l'approche vise à minimiser les violations des contraintes.

[Cardoen et al., 2009] ont abordé le problème de la planification des interventions du centre de soins chirurgicaux journaliers de l'hôpital universitaire, UZ Leuven (Belgique). Une phase de post-anesthésie à deux niveaux ainsi que des transferts et des capacités limitées doivent être gérés : le premier pour gérer la phase critique de réveil, et le second pour gérer le séjour du patient jusqu'à ce que le chirurgien donne l'autorisation de sortir. Les objectifs sont de réaliser les interventions chirurgicales des enfants tôt le matin, de planifier des interventions en début de journée pour les patients hautement prioritaires, de gérer la distance parcourue par les patients jusqu'au centre de soins de jour lors de la planification des interventions, de limiter le séjour en salle de réveil après la fermeture du centre de soins de jour et de minimiser le nombre maximal de lits utilisés dans les deux unités de soins post-anesthésiques. Les auteurs ont appliqué une approche de génération de colonnes, et ont résolu le problème de tarification par la programmation dynamique. Trois stratégies de branchement pour la méthode de *Branch-and-Price* ont été proposées et leur impact sur la qualité de la solution a été analysé.

Les problèmes de planification des interventions chirurgicales quotidiennes présentent des caractéristiques communes qui peuvent être observées, quelle que soit l'organisation qui elle est spécifique. La liste des interventions électives des patients à planifier est connue et toutes les interventions sont planifiées. Le makespan ou les heures supplémentaires sont souvent considérés comme une partie explicite de la fonction objectif, car il est important de ne pas dépasser les heures de travail normales. Étant donné que toutes les interventions doivent être planifiées, l'utilisation des ressources critiques pour établir un plan quotidien doit être envisagée car chaque ressource a une capacité limitée et constante.

Problèmes de planification des interventions hebdomadaires

[Roshanaei et al., 2021] ont proposé un modèle de programmation en nombres entiers mixtes basé sur des séquences et un modèle de programmation par contraintes pour résoudre un problème initialement proposé dans [Hashemi Doulabi et al., 2016] pour planifier des interventions chirurgicales sur un horizon d'une semaine. Les patients sont gérés à l'aide de deux ensembles (patients obligatoires et facultatifs). L'affectation des patients aux jours, l'affectation des couples (patient, jour) aux salles d'opération, l'affectation des triplets (patient, jour, salle d'opération) aux chirurgiens et, enfin, la planification des heures de début de ces opérations sont prises en compte à tour de rôle. L'objectif est de maximiser le temps total des interventions planifiées. Étant donné que ce problème peut être structuré de manière hiérarchique, les auteurs ont étudié la décomposition de Benders.

[Akbarzadeh et al., 2020] ont considéré un problème de planification d'une intervention chirurgicale sur plusieurs jours avec la réaffectation des infirmières. Des expériences

ont été menées sur des données générées à partir de données réelles de l'hôpital Sina (Téhéran, Iran). L'approche de solution heuristique proposée construit d'abord une solution LP optimale obtenue via un algorithme de génération de colonnes, puis une heuristique de recherche en profondeur est appliquée pour conduire les solutions fractionnaires à l'intégralité. L'objectif est de maximiser l'utilisation des salles d'opération en tenant compte des besoins en infirmières.

[Oliveira et al., 2020] ont étudié la planification des patients sur deux semaines en tenant compte de leur priorité et des sessions disponibles (salles d'opération et chirurgiens) pour un département d'urologie d'un hôpital universitaire de la ville de Québec. Un modèle MIP est utilisé sur des instances générées aléatoirement à partir de données réelles pour étudier l'impact de plusieurs politiques de gestion des patients en liste d'attente (quatre catégories de patients sont considérées par niveau d'urgence).

[Ballestín et al., 2019] ont étudié la planification et la replanification de patients non urgents sur deux semaines. Un plan provisoire est d'abord construit, suivi d'un plan définitif qui est calculé quelques jours avant le début de l'horizon de planification afin de prendre en compte de nouvelles informations (changement dans la liste d'attente ou dans le statut des patients). Les objectifs sont de minimiser le nombre de patients en retard et de maximiser le taux d'utilisation des salles d'opération. Une formulation en modèle linéaire est utilisée pour effectuer des simulations sur des instances générées aléatoirement à partir de données réelles d'un hôpital en Espagne. Les auteurs ont étudié le compromis entre le degré de changement autorisé dans le plan de travail provisoire et les avantages par rapport aux objectifs.

[Zhang et al., 2019b] a considéré un problème de planification hebdomadaire des interventions chirurgicales électives pour une spécialité unique. Les salles d'opération et l'unité de soins intensifs chirurgicaux ont une capacité limitée. Cependant, les chirurgiens ne sont pas pris en compte dans cette étude. Une approche à deux niveaux a été proposée. En appliquant un processus de décision de Markov, le premier niveau sélectionne les interventions à partir d'une liste de patients en attente. Le deuxième niveau est basé sur une approche de programmation dynamique affecte les interventions sélectionnées aux salles.

[Castro et al., 2015] ont abordé le problème de la planification sur une semaine des interventions chirurgicales électives de différentes spécialités à choisir parmi une grande liste d'interventions chirurgicales électives sur la base de trois niveaux de priorité des patients. Chaque chirurgien a une limite maximale de temps de travail hebdomadaire. Une approche de décomposition en deux étapes qui fait appel à la programmation disjonctive généralisée (GDP) est proposée. La première étape génère un modèle de planification qui ne tient pas compte des contraintes de disponibilité des chirurgiens. La deuxième étape génère un modèle de planification, qui vise à déterminer l'heure de début des interventions sélectionnées en assignant les chirurgiens aux salles d'opération, et en tenant compte des types d'interventions et de la disponibilité des chirurgiens. Certaines affectations de intervention/chirurgien/salle ne peuvent pas être réalisées en raison de conflits non pris en compte dans le modèle de planification, et certaines interventions peuvent donc ne pas être planifiées. Les affectations de salles et de jours ne changent pas, mais certaines interventions chirurgicales planifiées de moindre priorité peuvent être replacées sur une liste d'attente. L'approche proposée est testée sur des données

réelles provenant d'un hôpital central et universitaire de Lisbonne (Portugal). L'objectif est de maximiser le temps total de chirurgie.

Les problèmes de planification des interventions chirurgicales hebdomadaires présentent des caractéristiques communes qui peuvent être observées, quelle que soit l'organisation spécifique. Les principales ressources à gérer sont les chirurgiens et les salles d'opération. Dans le cadre d'une organisation hospitalière et compte tenu de l'importance relative des parties prenantes, les contraintes sur les allocations des chirurgiens et des salles d'opération doivent être gérées de manière adéquate. Toutes les interventions ne peuvent pas être planifiées car la liste d'attente dépasse la limite de l'horizon considéré de quelques jours, une semaine ou quelques semaines.

Les modèles de programmation mathématique sont couramment utilisés pour résoudre les problèmes de planification des interventions chirurgicales sur une journée ou une semaine. La programmation mathématique offre un cadre qui permet d'exprimer la grande variété des contraintes dépendantes du contexte rencontrées. Cependant, lorsque les problèmes augmentent en taille ou en complexité/nombre de contraintes différentes, les solveurs peuvent rencontrer des difficultés pour obtenir des solutions dans des délais raisonnables. Des approches heuristiques basées sur des formulations mathématiques, sur des heuristiques dédiées ou sur des méta-heuristiques sont utilisées pour obtenir de bonnes solutions dans des temps raisonnables.

Quelles que soient les spécificités de l'organisation des salles d'opération, les caractéristiques communes sont la liste des patients, chacun ayant une priorité, et les ressources principales sont les chirurgiens et les salles d'opération avec leurs disponibilités. Plus l'horizon augmente, plus l'importance de planifier tous les patients augmente en tenant compte de leur priorité en termes d'accès à la chirurgie. Face à une liste importante de patients pour lesquels des interventions doivent être programmées, chacun ayant ses propres caractéristiques de santé, il s'agit pour un hôpital de savoir s'il est possible d'avoir un plan d'admission où toutes les interventions peuvent être programmées. Il peut être utile de déterminer non seulement un jour pour l'opération, mais aussi une heure, ce qui permet de prendre en compte le plus tôt possible des contraintes telles que la planification de certaines interventions le matin. Les chirurgiens et les salles d'opération peuvent être considérés sur un horizon à long terme à condition de connaître les contraintes de disponibilité.

Pour un aperçu plus détaillé de la littérature sur les problèmes de planification de la chirurgie, nous invitons le lecteur à se référer à [Rahimi et al., 2020], [Samudra et al., 2016], [Erdogan et al., 2011], [May et al., 2011], [Cardoen et al., 2010], [Magerlein et al., 1978] qui font le point sur les variantes de ces problèmes et sur les approches de solutions.

2.6 CONCLUSION

Dans ce chapitre nous avons présenté différents problèmes d'optimisations aux quels doit faire face le système de santé.

No.	Littérature	Caractéristiques du problème				Méthode d'optimisation
		ORs	Surg	Obj	Data	
1	[Park et al., 2021]	✓	✓	SO	BRD	EA: MIP, H
2	[Roshanaei et al., 2021]	✓	✓	SO	TD	EA: MIP, EA: CP
3	[Akbarzadeh et al., 2020]	✓		SO	BRD	H
4	[Oliveira et al., 2020]	✓	✓	SO	BRD	EA: MIP
5	[Ballestín et al., 2019]	✓	✓	SO	BRD	EA: MIP, H: S
6	[Zhang et al., 2019b]	✓		SO	TD	H: ADP
7	[Xiang, 2017]	✓	✓	MO	TD	MH: ACO
8	[Hashemi Doulabi et al., 2016]	✓	✓	SO	TD	EA: CG
9	[Riise et al., 2016]	✓	✓	SO	BRD	H
10	[Wang et al., 2015]	✓	✓	SO	BRD	EA: MIP, EA: CP
11	[Xiang et al., 2015]	✓	✓	SO	TD	MH: ACO
12	[Castro et al., 2015]	✓	✓	SO	RD	H: GDP
13	[Sier et al., 1997]	✓		SO	BRD	H
14	notre étude	✓	✓	SO	BRD	MH: ALNS, H

Note :

- ORs(Operating Rooms), Surg(Surgeons), Obj(Objective).
- Obj: SO(Single-Objective), MO(Multi-Objective).
- Data: RD(Real Data), BRD(Based on Real Data), TD(Theoretic Data).
- Optimization method: H(Heuristics), MH(Metaheuristics), EA(Exact Algorithm).
- CP(Constraint Programming), MIP(Mixed Integer Program).
- ACO(Ant Colony Optimization), GDP(Generalized Disjunctive Programming).
- S(Simulation), ADP(Approximate Dynamic Programming).

Tableau 2.2: Études sur les problèmes de programmation des interventions chirurgicales.

Nous avons présenté des travaux qui concernent les services des urgences : le triage et la gestion du flux de patients, l'allocation des ressources, la planification/affectation de personnel et enfin la planification des tâches de soin.

La planification des tâches de soin au sein d'un service des urgences est un processus dynamique. Nous avons donc présenté des travaux qui concernent l'optimisation dynamique.

Finalement, nous avons présenté des travaux qui concernent la planification d'interventions chirurgicales car les SU sont sources de flux de patients à opérer.

CHAPITRE 3

PROBLÈME DE PLANIFICATION DANS LE SERVICE DES URGENCES

SOMMAIRE

3.1	Introduction	33
3.2	Définition du problème	34
3.2.1	Structure générale du service des urgences adultes du CHU de Lille	34
3.2.2	Structure du circuit court	36
3.2.3	Structure du circuit long	37
3.2.4	Tâches et contraintes entre tâches	37
3.2.5	Ressources et lieux	40
3.2.6	Calcul de performance et objectifs	41
3.3	Modélisation mathématique circuit court	41
3.4	Modélisation mathématique circuit long	50
3.5	Approches de résolution cas statique	54
3.5.1	Heuristique constructive basée sur les règles métiers (PRH)	54
3.5.2	IDCH	55
3.5.3	Méta-heuristique ALNS	56
3.6	Approche de résolution cas dynamique	64
3.7	Résultats	66
3.7.1	Benchmark et protocole expérimental	66
3.7.2	Paramétrage de IDCH	67
3.7.3	Paramétrage de ALNS acceptance base <i>Record-to-record</i>	68
3.7.4	Paramétrage de ALNS acceptance base recuit simulé	73
3.7.5	Évaluation des composantes de l'ALNS	78
3.7.6	Comparaison des deux versions de l'ALNS avec IDCH et ILP	80
3.7.7	Algorithme de planification dynamique	82
3.8	Conclusion et perspectives	84

3.1 INTRODUCTION

Les études sur le fonctionnement des systèmes de santé se sont multipliées ces dernières années. Il faut maintenir une qualité de service élevée, bien que la tendance soit à la réduction des ressources pour des raisons de coûts. L'un des secteurs médicaux impacté est celui des services des urgences. Les services des urgences adultes (SUA) assurent la prise en charge des venues non programmées, dans un établissement de santé public ou privé, vingt-quatre heures sur vingt-quatre, tous les jours de l'année, de toute personne sans sélection, se présentant en situation d'urgence, y compris psychiatrique. Le manque de ressources des services d'urgence pour faire face aux changements rapides observés dans la demande entraîne des niveaux élevés de congestion dans les salles d'attente des services des urgences. Les manques de ressources concernent les professionnels de santé (les infirmières, les médecins), mais aussi des ressources fixes, comme la capacité en lits. La congestion est causée par l'arrivée aléatoire des patients ainsi que par l'impossibilité de connaître à l'avance toutes les tâches de soin à prodiguer à un patient [LaCalle et al., 2010].

Un des moyens pour améliorer le fonctionnement des SUA est de mieux utiliser les ressources disponibles, il s'agit d'optimiser l'usage des ressources. Nous nous intéressons au problème d'ordonnancement des tâches de parcours de soins des patients dans le service des urgences du CHU Jeanne de Flandres de Lille. Nous avons pris en considération des contraintes de qualité de service, la disponibilité des ressources à chaque instant, les capacités des lieux ainsi que l'arrivée non programmée des patients dans les SUA.

Nous avons observé des différences dans la procédure de traitement des patients selon les pays [Kiriş et al., 2010; Luscombe et al., 2016; Lee et al., 2020], ce qui rend la mise en comparaison difficile. Pour modéliser le parcours des patients dans le service des urgences du CHU Jeanne de Flandres de Lille, nous avons réalisé plusieurs séjours au SUA guidés par deux médecins urgentistes. Nous avons étudié l'organisation du SUA et observé les parcours des patients, les tâches des personnels et l'allocation des ressources matérielles.

3.2 DÉFINITION DU PROBLÈME

Dans cette section, nous présentons la structure générale de service d'urgence adultes du CHU de Lille avec les deux circuits court et long. Nous présentons aussi les différentes caractéristiques du problème.

3.2.1 *Structure générale du service des urgences adultes du CHU de Lille*

Dans un Service des Urgences Adultes (SUA), la prise en charge d'un patient induit un ensemble de tâches de soins. Elles peuvent être effectuées de façon séquentielle et/ou parallèle. Elles requièrent les compétences d'un ou plusieurs membres du personnel médical et un lieu pour la réalisation. Le service des urgences adultes du CHU de Lille comprend deux circuits. Le circuit court (CC) fonctionne à la journée, il prend en charge

les patients les moins urgents de 08h00 à 21h00. Le circuit long (CL) fonctionne 24h/24h et 7 jours/7, il prend en charge les patients les plus urgents dans la journée et tous les patients qui se présentent aux urgences la nuit.

Pour modéliser les parcours des patients dans le service des urgences du CHU Jeanne de Flandres de Lille, il nous a fallu étudier l'organisation du service ainsi que le parcours des patients. Nous avons effectué plusieurs séjours au SUA du CHU de Lille avec deux médecins urgentistes pour observer les différentes tâches du personnel et l'allocation des ressources matérielles. Cette expérience immersive a été très bénéfique.

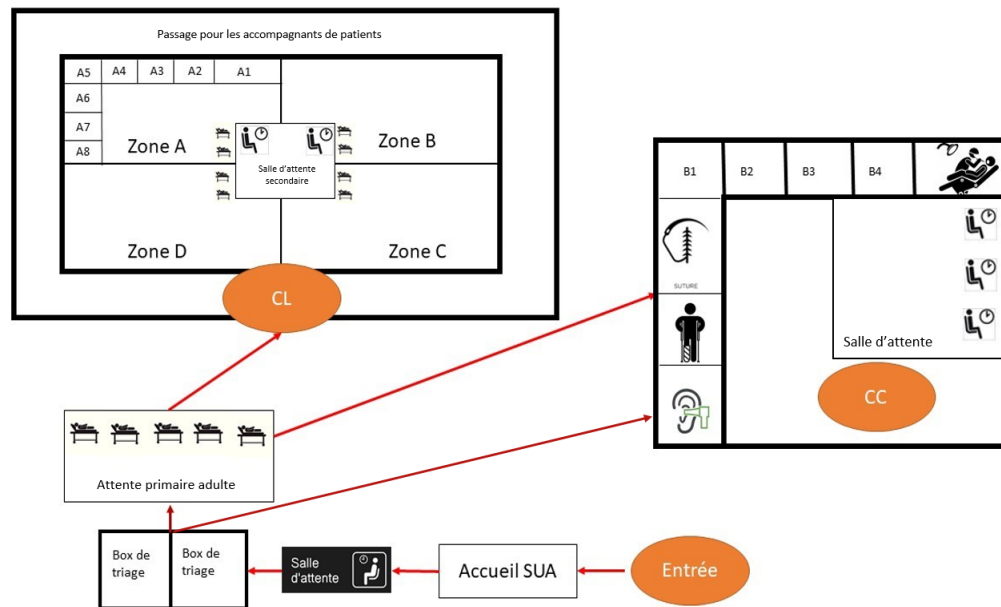


Fig 3.1: Structure du SUA de Lille

La Figure 3.1 décrit de manière détaillée la structure du service des urgences adultes (SUA) du CHU de Lille qui comprend :

- une entrée ;
- un accueil administratif ;
- une salle d'attente avant le triage ;
- deux box pour le triage ;
- une salle d'attente primaire pour les patients sur brancards ;
- un circuit court (CC) : qui comporte 4 boîtes de consultation, un box de suture, un box de plâtre, un box d'ORL, un box de stomatologie et une salle d'attente ;
- un circuit long (CL) : constitué de 4 zones (A, B, C, D). Chaque zone comporte 7 boîtes pour les patients sur brancards, un box pour les patients valides, une salle d'attente secondaire dédiée pour chaque zone et une salle d'attente secondaire commune pour toutes ces quatre zones.

Conceptuellement, le CL peut se voir comme la modélisation de quatre CC (zones A à D dans la Figure (3.1)) avec en plus une salle d'attente partagée et des box différemment dédiés .

La structuration de chaque étape du processus est nécessaire pour révéler les transitions, les périodes d'attente et les divergences qui peuvent entraver la qualité des soins dispensés aux patients. Il est essentiel de se concentrer d'abord sur la compréhension des processus, et non sur la recherche de solutions.

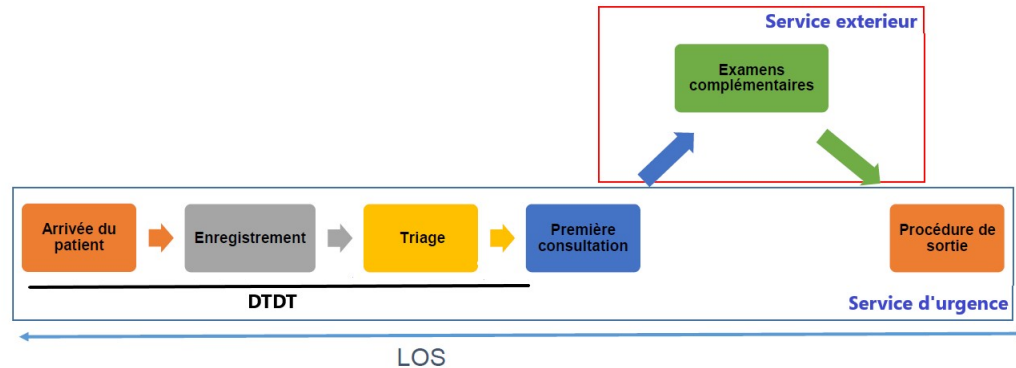


Fig 3.2: Parcours patient dans le SUA

La Figure 3.2 résume le parcours du patient, depuis son arrivée au SUA jusqu'à sa sortie. Le patient peut arriver à pied, par le SAMU ou par ambulances. Si le patient est en situation critique, la prise en charge par un médecin est immédiate. La prise en charge des patients en situation non-critique sera effectuée après une période d'attente dépendant de l'état d'encombrement du SUA et de la gravité du patient. Excepté pour les patients pris en charge par le SAMU, le premier soignant rencontré est l'infirmière organisatrice de l'accueil (IOA). L'IOA va rapidement, à l'aide d'un examen et d'outils de classification et de triage, évaluer la « charge en soins » nécessaire pour chaque patient afin de déterminer la priorité de prise en charge et l'orientation du patient au sein du service.

Le degré d'urgence est déterminé par le triage et se traduit par un code de la classification clinique des malades aux urgences (CCMU) entre 1 et 5. Ces codes CCMU déterminent entre autres le délai de prise en charge maximal pour le patient. Ces délais sont donnés dans le Tableau 3.1.

3.2.2 Structure du circuit court

Nous nous intéressons au problème de planification des patients au sein du CC. Nous considérons que l'étape de triage est déjà réalisée. Les mentions d'arrivée d'un patient correspondront à l'entrée du patient au sein du CC et non à l'arrivée au SUA.

Pour chaque patient arrivant dans le CC, plusieurs tâches sont à réaliser, l'ensemble des tâches et leurs séquençement sera désigné sous le terme parcours patient. Chaque tâche a besoin d'un lieu et/ou d'une ou plusieurs ressource(s) pour être menée à bien.

Code	Délais de prise en charge
1	Médecin < 240 min
2	Médecin < 120 min
3	Médecin < 90 min
4	Infirmier < 1min Médecin < 20 min
5	Infirmier < 1 min Médecin < 1 min

Tableau 3.1: Délais maximaux de prise en charge selon le code CCMU

L'attribution d'un lieu permet de respecter les capacités des salles et de suivre la localisation du patient. Les lieux et les ressources correspondent respectivement aux salles et au personnel médical du CC. La composition des lieux et du personnel médical du CC sont décrites par les Tableaux 3.2 et 3.3.

Nom	Nombre	Capacité
Salle d'attente	1	3
Box de consultation	4	1
Box de dentisterie	1	1
Box de suture	1	1
Box de plâtre	1	1
Box ORL	1	1

Tableau 3.2: Lieux du CC

Nom	Nombre
Médecin urgentiste	1
Interne en médecine	1
Interne en traumatologie	1
Interne dentaire	1
Interne ORL	1
Infirmier	2
Aide-soignants	4

Tableau 3.3: Personnel médical du CC

Nous considérons aussi de façon générique un lieu dit "externe" qui correspond aux lieux des services hors du SUA (e.g. le département de radiologie). Puisque le SUA n'a pas le contrôle sur ces lieux en dehors du SUA, nous considérons un lieu externe générique de capacité infinie. De manière similaire, nous considérons une ressource externe de capacité infinie (i.e. toujours disponible), associée aux services hors du périmètre du SUA.

3.2.3 *Structure du circuit long*

Pour chaque patient arrivant dans le CL, plusieurs tâches sont à réaliser que l'on désignera sous le terme parcours patient. Chaque tâche a besoin d'un lieu et/ou d'une ou plusieurs ressource(s) pour être menée à bien. Le circuit est composé de 4 zones (A, B, C, D). Chaque zone comporte 7 box pour les patients sur brancards et un box pour les patients valide. Pour chaque zone, il existe une salle d'attente pour les patients sur brancards avec une capacité très limitée. Les patients valide utilisent une salle d'attente commune pour les quatre zones. La composition du personnel médical du CL est décrite par le Tableau 3.4

Nom	Nombre
Médecin urgentiste	2
Interne en médecine	3
Externe en médecine	2
Infirmier	4
Aide-soignants	4
Hôtesses	2

Tableau 3.4: Personnel médical du CL

3.2.4 *Tâches et contraintes entre tâches*

Indépendamment de l'organisation en CC et CL, le parcours patient au sein de SUA peut être représenté sous la forme d'un graphe orienté acyclique comme dans l'exemple représenté dans la Figure (A.17). Les noeuds du graphe représentent les tâches que le patient devra effectuer lors de son séjour au SUA. Les arcs orientés représentent les relations de précédences entre les tâches. Les tâches de l'exemple fourni dans la Figure (A.17) sont décrites dans le Tableau A.15.

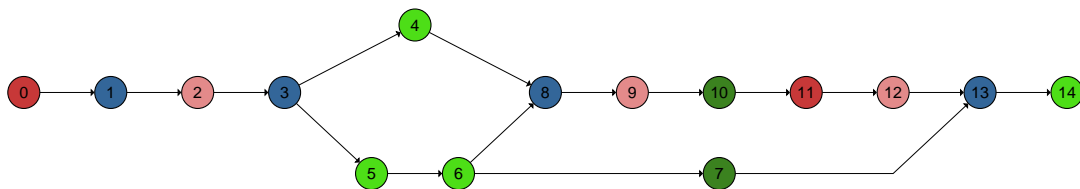


Fig 3.3: Exemple de graphe représentant un parcours patient

Il existe trois types de tâches : attente, déplacement et soin. Une tâche, de soin ou de déplacement, commencée ne peut être interrompue (non préemptible). Entre deux tâches, de soin ou de déplacement, un patient peut-être mis en attente. Les tâches d'attente correspondent à l'attente qu'effectue le patient dans un espace dédié (i.e. la salle d'attente, ou box) ou à l'extérieur du CC. Elles nécessitent un lieu mais pas de ressource pour être effectuées.

Tableau 3.5: Description des tâches de la figure (A.17)

N°	Nom	Type	Lieu	Ressource	Durée min
0	Attente avant entrée CC	Attente	Externe	∅	0
1	Entrée CC	Déplacement	Transport	Infirmière Aide-soignante	5
2	Attente avant consultation	Attente	Salle d'attente	∅	0
3	Déplacement vers consultation	Déplacement	Transport	Infirmière Aide-soignante	5
4	Consultation médicale	Traitement	Box	Médecin urgentiste Interne Traumatologie	15
5	Prise de signes vitaux	Traitement	Box	Infirmière	5
6	Prise de sang	Traitement	Box	Infirmière	5
7	Analyse biologique	Traitement	∅	Externe	30
8	Déplacement vers lieu d'attente	Déplacement	Déplacement	Infirmière Aide-soignante	5
9	Attente avant examen(s)	Attente	Salle d'attente	∅	0
10	Examens	Traitement	Externe	Externe	50
11	Attente avant retour CC	Attente	Salle d'attente	∅	0
12	Attente avant évaluation	Attente	Salle d'attente	∅	0
13	Déplacement vers évaluation	Déplacement	Transport	Infirmière Aide-soignante	5
14	Évaluation médicale	Traitement	Box	Médecin urgentiste Interne médecine	15

Les tâches de déplacement correspondent à l'accompagnement d'un patient par le personnel médical d'une salle à une autre. Elles nécessitent une ressource et un lieu fictif de capacité infinie qui peut être vu comme l'espace entre les différentes salles.

Les tâches de traitement correspondent à un soin prodigué sur un patient dans une salle allouée. Elles ont besoin d'un lieu et d'une ressource.

Enfin, il existe des tâches qui ne nécessitent pas la présence du patient mais qui sont nécessaires (e.g. analyse biologique d'un échantillon prélevé sur le patient). Puisque le patient n'est pas présent, ces tâches nécessitent seulement une ressource.

Les tâches de traitement sont soumises à plusieurs contraintes. Il est possible dans la réalité qu'un soin soit réalisé par plusieurs ressources. Une seule tâche ne peut représenter ce soin car seule une ressource peut y être affectée. On décompose le soin en autant de tâches que de ressources nécessaires. Ces tâches ainsi créées sont alors soumises à une contrainte stipulant que leur début de réalisation par leur ressource respective doit être le même. Les tâches 4 et 5 "consultation médicale" et "prise de signes vitaux" de le Tableau A.15 en sont un exemple.

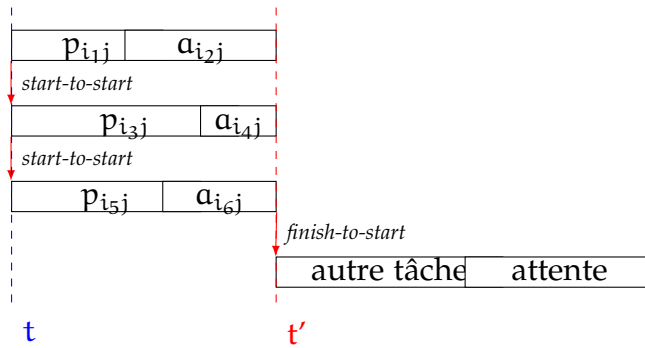


Fig 3.4: Exemple de trois tâches de soin commençant en même temps à t suivies de leurs tâches d’attente. Une autre tâche pour le patient j peut débuter après les tâches de soin et leurs attentes.

La Figure (3.4) donne un exemple pour un soin pour un patient j qui nécessite trois tâches débutant au même instant t qui requièrent des ressources différentes. Ce soin est décomposé en trois tâches avec leurs attentes qui sont suivies ensuite par une *autre tâche* et de son attente toujours pour le patient j . Cette *autre tâche* ne peut débuter qu’à l’instant t' , ce qui correspond à la fin des attentes des trois tâches qui la précèdent et à la disponibilité de la ressource nécessaire à la réalisation de cette autre tâche.

Il est aussi possible que des soins soient réalisés de manière séquentielle dans la réalité. Il n’est pas envisageable de passer d’une salle à une autre entre ces soins. Ainsi les tâches de traitement correspondantes sont soumises à une contrainte stipulant que le lieu doit être le même pour toutes les tâches de soins. Les tâches 4, 5 et 6 "consultation médicale", "prise de signes vitaux" et "prise de sang" de le Tableau A.15 en sont un exemple. La Figure (3.5) donne un exemple pour un séquençement de quatre tâches.

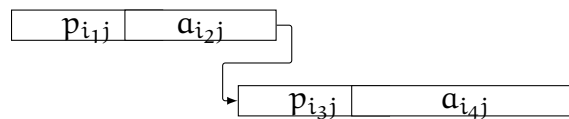


Fig 3.5: Pour un patient j , séquençement des tâches $i_1 \prec i_2 \prec i_3 \prec i_4$ de durées respectives $p_{i_1 j}, a_{i_2 j}, p_{i_3 j}, a_{i_4 j}$.

La durée exacte de toutes les tâches est de base inconnue dans ce problème. Les tâches disposent cependant d’une durée minimale : nulle pour les tâches d’attente et égale à un temps de réalisation estimé pour les autres.

3.2.5 Ressources et lieux

Les *ressources* peuvent être internes au SUA, elles sont de type personnel (médecin, infirmier, aide-soignant, etc.) mais aussi de type matériel (ECG sur chariot). Les *ressources* peuvent être externes au SUA, nous ne pouvons détailler précisément ces ressources car le SUA n'a pas de contrôle sur ces ressources (e.g. scanner et personnel médical qui réalise le scanner).

Nous faisons l'hypothèse que les ressources non renouvelables (consommables) sont infinies sur l'horizon T de temps d'un service (e.g. médicaments), et ce, aussi bien au sein du SUA que dans les services externes au SUA. Nous considérons l'utilisation des ressources renouvelables qui correspondent à des personnels et matériels spécifiques du SUA. Nous considérons aussi les ressources renouvelables externes au SUA sur lesquelles nous n'avons pas de contrôle. Nous faisons l'hypothèse que les ressources sont structurables par type.

Il y a plusieurs types de ressources. Au sein du SUA, il peut s'agir de matériels disponibles en quantité limitée qui peuvent être utilisés dans différents lieux (e.g. matériel ECG transportable). Il peut s'agir de matériels spécifiques qui ne sont disponibles que dans un (ou plusieurs) lieu(x) mais qui ne peuvent changer de lieu. Une tâche de soin qui requiert ce type de matériel devra être réalisée dans un des lieux possibles.

Nous considérons les personnels soignants (médecins, infirmier, aide-soignant, etc.) en tant que ressources avec un type qui correspond à des compétences (skills) pour exécuter des actes médicaux (tâches de soin). Nous faisons l'hypothèse que les personnels médicaux ne sont pas toujours disponibles sur l'horizon T . En effet, il faut pouvoir tenir compte des temps de pause. Nous faisons l'hypothèse qu'une tâche de soin requiert *au moins* une ressource (d'un type donné), elle peut aussi avoir besoin de plusieurs exemplaires d'une même ressource (e.g. plusieurs personnes d'une même compétence). Il peut s'agir de ressources internes du SUA ou de ressources externes au SUA. Par conséquent, une tâche de soin peut ne mobiliser aucune ressource du SUA. Par contre, une tâche de soin peut requérir plusieurs types de ressource du SUA simultanément. Une tâche de déplacement requiert exactement une ressource, qu'elle soit interne au SUA ou externe.

Le SUA est composé de lieux de traitement qui peuvent être interne au SUA ou externe au SUA. Nous distinguons donc les lieux internes au SUA et les lieux externes. Chaque lieu a une capacité qui correspond au nombre maximum de patients à ne pas dépasser.

Nous n'avons pas le contrôle des lieux externes, néanmoins, par exemple, prescrire trop d'exams de radiologie en même temps peut saturer ces ressources externes et induire une augmentation des temps d'attente. Pour la modélisation nous faisons l'hypothèse que les lieux externes sont de capacité infinie. Les lieux *internes* au SUA avec une capacité supérieure à 1 sont décomposés en plusieurs lieux de capacité 1. Nous faisons l'hypothèse que toute tâche de soin ou d'attente nécessite un lieu pour sa réalisation.

3.2.6 Calcul de performance et objectifs

Les temps d'attente susmentionnés doivent être minimisés. Pour cela on minimise le **Length Of Stay (LOS)** qui est la somme des temps de séjour de tous les patients. On doit également veiller à ce que les patients soient pris en charge correctement selon leur degré d'urgence. Pour ce faire, on minimise également le **Door-To-Doctor-Time (DTDT)** qui est la somme des attentes entre l'arrivée d'un patient au SUA et sa première consultation médicale, le tout pénalisé par le degré d'urgence du patient.

3.3 MODÉLISATION MATHÉMATIQUE CIRCUIT COURT

Cette section présente le modèle ILP (*Integer Linear Programming*) développé pour modéliser le problème de planification dans le circuit court. Dans le cas statique du problème, Nous supposons que le nombre des patients et leurs parcours dans le circuit sont connus au début de planification. Les durées des tâches de traitements et de déplacement sont connues et elles sont non préemptives. Nous présentons d'abord les données et paramètres relatifs au problème. Ensuite, nous présentons la formulation mathématique du problème.

Données et paramètres

Les patients, les ressources, les lieux ainsi que certaines considérations médicales et pratiques, correspondent aux données et paramètres suivants :

T , l'horizon de planification, ou la durée de la période de planification ;

J , l'ensemble de patients, indices j , nous désignons par $|J|$ sa taille ;

a_j , l'heure d'arrivée du patient j ;

d_j , la première tâche du patient j qui nécessite un médecin, première consultation médicale ;

ω_j , la pénalité de priorité qui se réfère au degré d'urgence du patient j (dépendent du code CCMU) ;

φ_j , la dernière tâche dans le parcours du patient j ;

I_j^{in} , l'ensemble des tâches réalisées dans le SUA (des tâches internes) du patient j ;

I_j^{ou} , l'ensemble des tâches réalisées hors du SUA à la demande du SUA (tâches externes) du patient j ;

I_j , l'ensemble des tâches du patient j , indice i , $I_j = I_j^{in} \cup I_j^{ou}$;

β_{ij} , date au plus tôt de la tâche i du patient j ;

γ_{ij} , date au plus tard de la tâche i du patient j ;

W_j^{in} , l'ensemble des tâches d'attente internes du patient j (attentes au sein du SUA) ;

W_j^{ou} , l'ensemble des tâches d'attente externes du patient j (attentes en dehors du SUA) ;

W_j , l'ensemble des tâches d'attente du patient j , $W_j = W_j^{in} \cup W_j^{ou}$;

C_j^{in} , l'ensemble des tâches de soins réalisées au sein du SUA (des tâches de soins internes) du patient j ;

C_j^{ou} , l'ensemble des tâches de soins réalisées en dehors du SUA (des tâches de soins externes) du patient j ;

C_j , l'ensemble des tâches de soins du patient j , $C_j = C_j^{in} \cup C_j^{ou}$;

D_j^{in} , l'ensemble des tâches de déplacement internes du patient j ;

D_j^{ou} , l'ensemble des tâches de déplacement externes du patient j ;

D_j , l'ensemble des tâches de déplacement du patient j , $D_j = D_j^{in} \cup D_j^{ou}$;

E_j , l'ensemble des tâches qui ne nécessitent pas la présence du patient j (e.g. analyse des échantillons) ;

L , l'ensemble des lieux, indice l ;

L_{ij} , l'ensemble des lieux où la tâche i du patient j peut être effectuée ;

G , l'ensemble des types de ressources, indice g ;

R_g , le nombre de ressources de type g ;

g_{ext} , le type correspondant aux ressources externes ;

G_{ij} , l'ensemble des types de ressources qui peuvent effectuer la tâche i du patient j ;

Γ_{ij}^{-1} , l'ensemble des tâches de précédence de la tâche i du patient j ;

S_{ij} , l'ensemble des tâches successeurs de la tâche i du patient j ;

Λ_{ij} , ensemble des tâches qui doivent être effectuées au même lieu que la tâche i du patient j ;

Ω_{ij} , l'ensemble des tâches qui doivent commencer en même temps que la tâche i du patient j .

Les variables de décision principales sont les suivantes :

$$x_{ijlt} = \begin{cases} 1 & \text{si le patient } j \text{ entre dans le lieu } l \\ & \text{pour effectuer la tâche } i \text{ à } t \\ 0 & \text{sinon} \end{cases}$$

$$z_{ijgrt} = \begin{cases} 1 & \text{si la ressource } r \text{ de type } g \text{ est utilisée} \\ & \text{pour effectuer la tâche } i \text{ du patient } j \text{ à } t, \\ & \text{début de la tâche du patient } j \\ 0 & \text{sinon} \end{cases}$$

À l'exclusion des tâches d'attente, chaque tâche est traitée pendant le temps d'occupation de la place qui lui est allouée et nécessite une ressource pour être exécutée. Le début du temps d'occupation est exprimé par les variables de décision x_{ijlt} . Le parcours d'un patient au sein du SUA est modélisé comme une succession de lieux dans lesquels des tâches sont effectuées, ces tâches peuvent requérir des ressources pour être réalisées. Les variables x_{ijlt} permettent de modéliser l'arrivée d'un patient j dans un lieu l afin que la tâche i puisse être effectuée. Il est important de faire remarquer qu'il s'agit de déplacements formels et pas nécessairement de déplacements physiques du patient. Par exemple, deux tâches i et i' qui se succèdent peuvent être effectuées dans le même lieu l sans le déplacement physique du patient dans un autre lieu. Le patient quitte formellement le lieu l à l'instant t après la réalisation de la tâche i et entre formellement au même instant t dans le même lieu l pour la réalisation de la tâche i' . Le modèle utilise les variables z_{ijgrt} pour l'allocation des ressources.

Les variables de décision secondaires sont les suivantes :

$$u_{iji'j'l} = \begin{cases} 1 & \text{si le lieu } l \text{ est utilisé pour effectuer la tâche } i \\ & \text{du patient } j \text{ avant la tâche } i' \text{ du patient } j' \\ 0 & \text{sinon} \end{cases}$$

$$\delta_{ij} = \begin{cases} 1 & \text{si la durée de la tâche } i \text{ du patient } j \\ & \text{dans un lieu est supérieure à zéro} \\ 0 & \text{sinon} \end{cases}$$

Nous avons supposé que les lieux interne du SUA sont de capacité unitaire (box et salle d'attente CC décomposée en un ensemble de salle de capacité unitaire). Un lieu ne peut être utilisé que par un unique patient à la fois. Nous devons gérer l'ordre de passage des patients dans les lieux interne du SUA. Les variables $u_{iji'j'l}$ sont utilisées pour gérer la disjonction des usages des lieux. elles permettent de traiter le séquençement des patients dans un lieu de traitement. La durée d'une tâche d'attente est inconnue, elle peut être nulle. Nous utilisons les variables δ_{ij} pour récupérer information sur la durée nulle d'une tâche d'attente.

La formulation ILP est la suivante :

Min

$$\sum_{j \in J} \left(\sum_{t=1}^T \sum_{l \in L_{\varphi_j}} t \cdot x_{\varphi_j lt} - \sum_{t=1}^T \sum_{l \in L_{1j}} t \cdot x_{1j lt} \right) \quad (3.1a)$$

$$+ \sum_{j \in J} \omega_j \left(\sum_{t=1}^T \sum_{g \in G_{d,j}} \sum_{r=1}^{R_g} t \cdot z_{d,j,r,t} - a_j \right) \quad (3.1b)$$

Sous les contraintes :

$$x_{ijlt}, z_{ijgrt}, u_{iji'/l}, \delta_{ij} \in \{0, 1\} \quad (3.2)$$

$$\sum_{t=\beta_{ij}}^{\gamma_{ij}} \sum_{l \in L_{ij}} (t \cdot x_{ijlt}) = a_j \quad \forall j \in J \quad (3.3)$$

$$\sum_{t=\beta_{ij}}^{\gamma_{ij}} \sum_{l \in L_{ij}} x_{ijlt} = 1 \quad \forall j \in J, \forall i \in I_j \quad (3.4)$$

$$\sum_{t=\beta_{ij}}^{\gamma_{ij}} \sum_{g \in G_{ij}} \sum_{r=1}^{R_g} z_{ijgrt} = 1 \quad \forall j \in J, \forall i \in D_j \cup C_j \cup E_j \quad (3.5)$$

$$\sum_{j \in J} \sum_{i \in D_j \cup C_j \cup E_j} \sum_{\tau=\max\{\beta_{ij}, t-p_{ij}+1\}}^{\min\{\gamma_{ij}, t\}} z_{ijgrt} \leq 1$$

$$\forall g \in G \setminus \{g_{ext}\}, \forall r \in \{1, \dots, R_g\}, \forall t \in \{1, \dots, T\} \quad (3.6)$$

$$\sum_{t=\beta_{ij}}^{\gamma_{ij}} \sum_{l \in L_{ij}} (l \cdot x_{ijlt}) - \sum_{t=\beta_{i'/j}}^{\gamma_{i'/j}} \sum_{l \in L_{i'/j}} (l \cdot x_{i'/jlt}) = 0$$

$$\forall j \in J, \forall i \in I_j, \forall i' \in \Lambda_{ij} \quad (3.7)$$

$$\sum_{t=\beta_{ij}}^{\gamma_{ij}} \sum_{g \in G_{ij}} \sum_{r=1}^{R_g} (t \cdot z_{ijgrt}) - \sum_{t=\beta_{i'/j}}^{\gamma_{i'/j}} \sum_{g \in G_{i'/j}} \sum_{r=1}^{R_g} (t \cdot z_{i'/jgrt}) = 0$$

$$\forall j \in J, \forall i \in C_j, \forall i' \in \Omega_{ij} \quad (3.8)$$

$$\sum_{t=\beta_{ij}}^{\gamma_{ij}} \sum_{l \in L_{ij}} (t \cdot x_{ijlt}) - \sum_{t=\beta_{i'/j}}^{\gamma_{i'/j}} \sum_{l \in L_{i'/j}} (t \cdot x_{i'/jlt}) = 0$$

$$\forall j \in J, \forall i \in I_j, \forall i' \in \Omega_{ij} \quad (3.9)$$

$$\sum_{t=\beta_{ij}}^{\gamma_{ij}} \sum_{l \in L_{ij}} (t \cdot x_{ijlt}) - \sum_{t=\beta_{i'j}}^{\gamma_{i'j}} \sum_{l \in L_{i'j}} (t \cdot x_{i'jlt}) \geq 0$$

$$\forall j \in J, \forall i \in I_j, \forall i' \in \Gamma_{ij}^{-1} \cap W_j \quad (3.10)$$

$$\sum_{t=\beta_{ij}}^{\gamma_{ij}} \sum_{l \in L_{ij}} (t \cdot x_{ijlt}) - \sum_{t=\beta_{i'j}}^{\gamma_{i'j}} \sum_{g \in G_{i'j}} \sum_{r=1}^{R_g} (t \cdot z_{i'jgrt}) \geq p_{i'j}$$

$$\forall j, \forall i \in I_j, \forall i' \in \Gamma_{ij}^{-1} \cap (C_j^{\text{in}} \cup E_j) \quad (3.11)$$

$$\sum_{t=\beta_{ij}}^{\gamma_{ij}} \sum_{l \in L_{ij}} (t \cdot x_{ijlt}) - \sum_{t=\beta_{i'j}}^{\gamma_{i'j}} \sum_{g \in G_{i'j}} \sum_{r=1}^{R_g} (t \cdot z_{i'jgrt}) = p_{i'j}$$

$$\forall j \in J, \forall i \in I_j, \forall i' \in \Gamma_{ij}^{-1} \cap (D_j \cup C_j^{\text{ou}}) \quad (3.12)$$

$$\sum_{t=\beta_{ij}}^{\gamma_{ij}} \sum_{g \in G_{ij}} \sum_{r=1}^{R_g} (t \cdot z_{ijgrt}) - \sum_{t=\beta_{i'j}}^{\gamma_{i'j}} \sum_{g \in G_{i'j}} \sum_{r=1}^{R_g} (t \cdot z_{i'jgrt}) \geq p_{i'j}$$

$$\forall j \in J, \forall i \in E_j, \forall i' \in \Gamma_{i'j}^{-1} \cap (C_j \cup D_j \cup E_j) \quad (3.13)$$

$$\sum_{t=\beta_{ij}}^{\gamma_{ij}} \sum_{g \in G_{ij}} \sum_{r=1}^{R_g} (t \cdot z_{ijgrt}) - \sum_{t=\beta_{ij}}^{\gamma_{ij}} \sum_{l \in L_{ij}} (t \cdot x_{ijlt}) \geq 0$$

$$\forall j \in J, \forall i \in C_j^{\text{in}} \quad (3.14)$$

$$\sum_{t=\beta_{ij}}^{\gamma_{ij}} \sum_{g \in G_{ij}} \sum_{r=1}^{R_g} (t \cdot z_{ijgrt}) - \sum_{t=\beta_{ij}}^{\gamma_{ij}} \sum_{l \in L_{ij}} (t \cdot x_{ijlt}) = 0$$

$$\forall j \in J, \forall i \in D_j \cup C_j^{\text{ou}} \quad (3.15)$$

$$\sum_{t=\beta_{sj}}^{\gamma_{sj}} \sum_{l \in L_{sj}} (t \cdot x_{sjlt}) - \sum_{t=\beta_{ij}}^{\gamma_{ij}} \sum_{l \in L_{ij}} (t \cdot x_{ijlt}) \geq \delta_{ij}$$

$$\forall j \in J, \forall i \in I_j, s \in S_{ij} \cap I_j \quad (3.16)$$

$$\sum_{t=\beta_{sj}}^{\gamma_{sj}} \sum_{l \in L_{sj}} (t \cdot x_{sjlt}) - \sum_{t=\beta_{ij}}^{\gamma_{ij}} \sum_{l \in L_{ij}} (t \cdot x_{ijlt}) \leq \delta_{ij} \cdot \Delta_{ij}$$

$$\forall j \in J, \forall i \in I_j, s \in S_{ij} \cap I_j \quad (3.17)$$

$$u_{iji'j'l} + u_{i'j'ijl} \geq \sum_{t=\beta_{ij}}^{\gamma_{ij}} x_{ijlt} + \sum_{t=\beta_{i'j'}}^{\gamma_{i'j'}} x_{i'j'lt} + \delta_{ij} + \delta_{i'j'} - 3$$

$$\forall j \in J, \forall i \in I_j^{\text{in}}, \forall j' \in J \setminus \{j\}, \forall i' \in I_{j'}^{\text{in}}, \forall l \in L_{ij} \cap L_{i'j'} \quad (3.18)$$

$$u_{iji'j'l} + u_{i'j'ijl} \leq 1$$

$$\forall j \in J, \forall i \in I_j^{\text{in}}, \forall j' \in J \setminus \{j\}, \forall i' \in I_{j'}^{\text{in}}, \forall l \in L_{ij} \cap L_{i'j'} \quad (3.19)$$

$$\sum_{t=\beta_{i'j'}}^{\gamma_{i'j'}} (t \cdot x_{i'j'lt}) - \sum_{t=\beta_{sj}}^{\gamma_{sj}} \sum_{l \in L_{sj}} t \cdot x_{sjlt} \geq (u_{iji'j'l} - 1) \cdot M_{ij}$$

$$\forall j \in J, \forall i \in I_j^{\text{in}}, \forall j' \in J \setminus \{j\}, \forall i' \in I_{j'}^{\text{in}},$$

$$\forall l' \in L_{ij} \cap L_{i'j'}, s \in S_{ij} \cap I_j \quad (3.20)$$

La fonction objectif se compose de deux termes. Le premier est le LOS (3.1a) qui représente la somme des temps de séjour entre tous les patients et le second est le DTDT (3.1b) qui est le temps d'attente pénalisé pour tous les patients entre leur arrivée a_j et leur première consultation médicale d_j avec un médecin. Les poids ω_j dépendent du code CCMU du patient.

Usage des capacités des ressources

Les contraintes (3.2) imposent que les variables booléennes soit zéro ou un. Les contraintes (3.3) garantissent que chaque patient j se trouve dans un lieu à l'heure d'arrivée a_j ce qui permet de toujours avoir la localisation du patient car cette tâche peut être une tâche d'attente. Les contraintes (3.4) garantissent que chaque tâche est traitée dans un seul et unique lieu. Ces contraintes concernent aussi bien les tâches de soin que les tâches d'attente, et, les lieux internes et les lieux externes. Les contraintes (3.5)

garantissent qu'une seule ressource est utilisée par les tâches de soins et de déplacement. Dans ce cas nous avons une unique variable $z_{ijgrt} = 1$, ce qui indique que la ressource r de type g nécessaire est affectée à la réalisation de la tâche, et cela pour tout l'horizon. Les contraintes (3.4) et (3.5) garantissent que chaque tâche n'est exécutée qu'une seule fois dans le parcours d'un patient.

Disponibilité des ressource

Il faut assurer que la ressource r de type g est disponible à l'instant t pour qu'une tâche puisse l'utiliser. Une unique variable $z_{ijgrt} = 1$ si à l'instant t débute l'utilisation de la ressource r de type g pour la tâche i du patient j . Mais, pour cela cette ressource doit rester non utilisée par les autres tâches qui ont besoin et qui peuvent aussi utiliser cette ressource pour être réalisées. Par conséquent, si on veut utiliser la ressource à t , il faut vérifier que à l'instant t : il n'y a qu'une seule tâche de soin qui débute l'utilisation de la ressource r de type g . Il faut aussi vérifier qu'aucune autre tâche qui peut utiliser la ressource r de type g n'a débuté avant t et qu'elle ne soit pas encore achevée. Si ces deux conditions ne sont pas remplies l'utilisation de la ressource r n'est pas encore possible. À l'exception des ressources externes, les contraintes (3.6) garantissent qu'une ressource allouée à une tâche i d'un patient j ne peut être allouée à un autre patient pendant le temps de traitement de la tâche i .

Continuité de l'utilisation des lieux

Les contraintes (3.7) garantissent qu'un même lieu est utilisé par les tâches qui en ont besoin. Généralement, ces tâches doivent débiter au même instant t . Nous rappelons que si un soin dans le SUA pour un patient j nécessite plusieurs ressources alors il est décomposé en autant de tâches de soin que de ressources. Dans le SUA, à une tâche correspond une ressource. Cependant, ces tâches doivent impérativement débiter au même instant t . Pour un patient j et pour chaque tâche de soin i , nous devons assurer que l'utilisation des ressources nécessaires aux tâches de soin $i' \in \Omega_{ij}$ doivent débiter au même instant que l'utilisation de la ressource nécessaire à la tâche de soin i .

Synchronisation des tâches

Les contraintes (3.8) permettent d'assurer que le début t de l'utilisation d'une ressource r de type g pour la tâche i ($z_{ij(gr)t} = 1$) est le même pour chaque ressource nécessaire aux tâches i' . Les contraintes (3.8) et (3.9) garantissent que les moments t de début des tâches simultanées sont respectées.

Précédences entre les tâches

Un patient j peut entrer dans un lieu pour une tâche i lorsque toutes les tâches de précédence Γ_{ij}^{-1} de la tâche i sont terminées. Contraintes (3.10)-(3.12) assurent les contraintes de précédence en fonction de chaque type de tâches qui précède la tâche i (attente, soins, déplacement). Les contraintes de précédences ne sont pas de même nature selon les types de tâches. Les contraintes 3.10 imposent que si la tâche qui précède la tâche i est une tâche d'attente alors le changement de lieu pour la tâche suivante peut débiter à n'importe quel moment après l'arrivée dans le lieu d'attente. Les contraintes 3.11 imposent que si la tâche qui précède la tâches i est une tâche de traitement alors le changement de lieu pour la tâche suivante peut débiter à n'importe quel moment

après la fin de la réalisation de durée $p_{i'j}$ de la tâche de traitement par la ressource. On autorise ici une attente dans le lieu de traitement avant la tâche suivante. Les contraintes 3.12 imposent que si un prédécesseur est une tâche de déplacement alors le changement de lieu pour la tâche suivante doit débiter à la fin de la réalisation de la tâche de déplacement par la ressource. On n'autorise pas d'attente dans le lieu de déplacement car le patient doit se trouver dans un lieu autre que le lieu fictif de déplacement à la fin de la réalisation de cette tâche.

Pour les tâches d'un patient qui ne requiert pas sa présence (e.g. analyse d'un échantillon de sang) le début de leur réalisation est lui aussi contraint par leurs prédécesseurs. Les contraintes 3.13 exigent qu'une tâche sans patient peut commencer sa réalisation dès la fin de la réalisation de des tâches précédentes par sa ressource.

Synchronisation tâches et lieux

Les contraintes (3.14) garantissent que une tâche de soins peut commencer une fois que le patient se trouve dans l'un des lieux appropriés pour la tâche i . Pour les tâches de déplacement, les contraintes (3.15) garantissent que le patient est immédiatement pris en charge par la ressource. Les contraintes (3.16)-(3.17) garantissent que δ_{ij} est égal à zéro si la tâche i a une durée nulle.

Usage des capacités des lieux et disjonction d'usage

Il faut assurer qu'un lieu du SUA n'est utilisé que par un patient à la fois. Nous rappelons que les lieux sont considérés de capacité unitaire. Si un lieu physique réel du SUA est de capacité $k_l > 1$ alors il est considéré dans le modèle comme k_l lieux différents de capacité 1. Un patient j et un patient j' ne peuvent être présents en même temps dans un lieu l de capacité unitaire. L'un doit donc précéder l'autre pour les tâches de soin ou d'attente dans ce lieu s'il est nécessaire pour une tâche. Pour une tâche i d'un patient j et une tâche i' d'un patient j' , si ces tâches peuvent être effectuées dans le lieu l , nous avons introduit la variable de décision booléenne $u_{(ij)(i'j')l}$ qui vaut 1 si la tâche i du patient j précède la tâche i' du patient j' dans le lieu l , zéro si c'est inverse. Les contraintes (3.18)-(3.20) assurent les contraintes de disjonction.

Les contraintes (3.18)-(3.20) utilisent la notation big-M pour modéliser le séquençement des patients qui utilise le même lieu de traitement. La paire de contraintes big-M est nécessaire parce que la durée de séjour dans un lieu est inconnue. Cette paire de contraintes modélise une relation logique *if-then* entre l'heure d'entrée et l'heure de sortie de deux patients affectés au même lieu. Si deux patients $j, j' \in J$ utilisent le même lieu $l \in L$ alors la variable binaire $u_{(ij)(i'j')l}$ est utilisée pour déterminer l'ordre des deux patients.

Discussion sur le choix des valeurs des paramètres M_{ij} et Δ_{ij}

Les contraintes 3.17 et 3.20 utilisent respectivement les nombres Δ_{ij} et M_{ij} dans leur formulation. Ces paramètres permettent de fixer les valeurs des variables lorsqu'une variable binaire prend une certaine valeur tout en leur permettant de rester libres dans les autres cas. Il est préférable de choisir une petite valeur pour Δ_{ij} et M_{ij} parmi celles possibles car une valeur trop grande risque de poser problème pour la résolution du modèle. Plusieurs choix sont possibles.

Nous pouvons donner la valeur T à Δ_{ij} et à M_{ij} car elles encadrent une différence de temps nécessairement inférieure à l'horizon du modèle. Nous proposons une meilleure solution, elle dépend de chaque tâche.

D_{ij} doit être supérieure à la durée d'occupation maximale du lieu de la tâche i du patient j . Elle est donc égale à la date au plus tard de cette tâche moins sa date au plus tôt :

$$D_{ij} = \gamma_{ij} - \beta_{ij}$$

Pour M_{ij} , plusieurs cas sont à considérer :

- la tâche i' se déroule dans le lieu l'

la valeur de M_{ij} doit être supérieure à l'écart entre le début d'occupation du lieu de la tâche i' du patient j' et début d'occupation du lieu d'une tâche successeur s de i nécessitant le patient j . M_{ij} doit donc être supérieure à la date au plus tard d'une tâche successeur s de du patient j moins la date au plus tôt de la tâche i' du patient j' .

- la tâche i' ne se déroule pas dans le lieu l'

la valeur de M_{ij} doit être supérieure à l'entrée dans un lieu d'une tâche successeur s de i nécessitant le patient j . En effet $\sum_{t=\beta_{i'j'}}^{\gamma_{i'j'}} (t \cdot x_{i'j'l't}) = 0$.

Ces deux cas nous amènent à la formulation suivante de M_{ij} :

$$M_{ij} = \gamma_{sj} \quad s \in S_{ij} \cap I_j$$

3.4 MODÉLISATION MATHÉMATIQUE CIRCUIT LONG

L'extension au CL peut se voir comme la modélisation de 4 CC (zones A à D dans la Figure (3.1)) avec une salle d'attente partagée. Pour modéliser le circuit long, nous gardons les mêmes données et paramètres du modèle mathématique du circuit court. Pour une tâche i d'un patient j , nous désignons l'ensemble des lieux où la tâche i peut être exécutée par L_{ij} , l'ensemble des lieux où i peut être exécutée dans une zone spécifique k par L_{ijk} , l'ensemble des types de ressources qui peuvent exécuter i par G_{ij} , l'ensemble des types de ressources qui peuvent exécuter la tâche i dans une zone spécifique k par G_{ijk} . Nous désignons par W_j^C l'ensemble des tâches d'attente qui ont besoin de la salle d'attente commune. Au variables du modèle du circuit court, nous ajoutons les variable y_{kj} pour gérer les affectations des patients aux zones.

$$y_{kj} = \begin{cases} 1 & \text{si le patient } j \text{ est affecté à la zone } k \\ 0 & \text{sinon} \end{cases}$$

Nous ajoutons aussi les contraintes 3.21, 3.4 et 3.22b. Les contraintes (3.21) garantissent que chaque patient est affecté à une zone unique. Les contraintes (3.4) et 3.22b garantissent que chaque tâche est traitée à un seul lieu dans la zone assignée au patient.

$$\sum_k y_{kj} = 1 \quad \forall k, \forall j \in J \quad (3.21)$$

$$\sum_{t=\beta_{ij}}^{\gamma_{ij}} \sum_{l \in L_{ijk}} x_{ijlt} = y_{kj} \quad \forall k, \forall j \in J, \forall i \in I_j^{\text{in}} \setminus (W^C \cup D_j) \quad (3.22a)$$

$$\sum_{t=\beta_{ij}}^{\gamma_{ij}} \sum_{l \in L_{ij}} x_{ijlt} = 1 \quad \forall j \in J, \forall i \in I_j^{\text{out}} \cup W^C \cup D_j \quad (3.22b)$$

Dans le circuit long, chaque zone a sa propre équipe médicale. Nous remplaçons les contraintes 3.5 de la modélisation du circuit court par :

$$\sum_{t=\beta_{ij}}^{\gamma_{ij}} \sum_{g \in G_{ijk}} \sum_{r=1}^{R_g} z_{ijgrt} = 1 \quad \forall j \in J, \forall i \in D_j \cup C_j \cup E_j$$

Ces contraintes garantissent qu'une seule ressource d'une seule zone est utilisée par les tâches de soins et de déplacement.

La formulation ILP est la suivante :

Min

$$\sum_{j \in J} \left(\sum_{t=1}^T \sum_{l \in L_{\varphi_j}} t \cdot x_{\varphi_j lt} - \sum_{t=1}^T \sum_{l \in L_{1j}} t \cdot x_{1j lt} \right) \quad (3.23a)$$

$$+ \sum_{j \in J} \omega_j \left(\sum_{t=1}^T \sum_{g \in G_{d_j}} \sum_{r=1}^{R_g} t \cdot z_{d_j r t} - a_j \right) \quad (3.23b)$$

s.t.

$$y_{kj}, x_{ijlt}, z_{ijgrt}, u_{iji'/l}, \delta_{ij} \in \{0, 1\} \quad (3.24)$$

$$\sum_{t=\beta_{1j}}^{\gamma_{1j}} \sum_{l \in L_{1j}} (t \cdot x_{1j lt}) = a_j \quad \forall j \in J \quad (3.25)$$

$$\sum_k y_{kj} = 1 \quad \forall k, \forall j \in J \quad (3.26)$$

$$\sum_{t=\beta_{ij}}^{\gamma_{ij}} \sum_{l \in L_{ijk}} x_{ijlt} = y_{kj} \quad \forall k, \forall j \in J, \forall i \in I_j^{\text{in}} \setminus (W^C \cup D_j) \quad (3.27a)$$

$$\sum_{t=\beta_{ij}}^{\gamma_{ij}} \sum_{l \in L_{ij}} x_{ijlt} = 1 \quad \forall j \in J, \forall i \in I_j^{\text{out}} \cup W^C \cup D_j \quad (3.27b)$$

$$\sum_{t=\beta_{ij}}^{\gamma_{ij}} \sum_{g \in G_{ijk}} \sum_{r=1}^{R_g} z_{ijgrt} = 1 \quad \forall j \in J, \forall i \in D_j \cup C_j \cup E_j \quad (3.28)$$

$$\sum_{j \in J} \sum_{i \in D_j \cup C_j \cup E_j} \sum_{\tau=\max\{\beta_{ij}, t-p_{ij}+1\}}^{\min\{\gamma_{ij}, t\}} z_{ijgr\tau} \leq 1$$

$$\forall g \in G \setminus \{g_{\text{ext}}\}, \forall r \in \{1, \dots, R_g\}, \forall t \in \{1, \dots, T\} \quad (3.29)$$

$$\sum_{t=\beta_{ij}}^{\gamma_{ij}} \sum_{l \in L_{ij}} (l \cdot x_{ijlt}) - \sum_{t=\beta_{i'j}}^{\gamma_{i'j}} \sum_{l \in L_{i'j}} (l \cdot x_{i'jlt}) = 0$$

$$\forall j \in J, \forall i \in I_j, \forall i' \in \Lambda_{ij} \quad (3.30)$$

$$\sum_{t=\beta_{ij}}^{\gamma_{ij}} \sum_{g \in G_{ij}} \sum_{r=1}^{R_g} (t \cdot z_{ijgrt}) - \sum_{t=\beta_{i'j}}^{\gamma_{i'j}} \sum_{g \in G_{i'j}} \sum_{r=1}^{R_g} (t \cdot z_{i'jgrt}) = 0$$

$$\forall j \in J, \forall i \in C_j, \forall i' \in \Omega_{ij} \quad (3.31)$$

$$\sum_{t=\beta_{ij}}^{\gamma_{ij}} \sum_{l \in L_{ij}} (t \cdot x_{ijlt}) - \sum_{t=\beta_{i'j}}^{\gamma_{i'j}} \sum_{l \in L_{i'j}} (t \cdot x_{i'jlt}) = 0$$

$$\forall j \in J, \forall i \in I_j, \forall i' \in \Omega_{ij} \quad (3.32a)$$

$$\sum_{l \in L_{ij}} (t \cdot x_{ijlt}) - \sum_{l \in L_{i'j}} (t \cdot x_{i'jlt}) = 0$$

$$\forall t \in \{\max(\beta_{ij}, \beta_{i'j}), \dots, \min(\gamma_{ij}, \gamma_{i'j})\},$$

$$\forall j \in J, \forall i \in I_j, \forall i' \in \Omega_{ij} \quad (3.32b)$$

$$\sum_{t=\beta_{ij}}^{\gamma_{ij}} \sum_{l \in L_{ij}} (t \cdot x_{ijlt}) - \sum_{t=\beta_{i'j}}^{\gamma_{i'j}} \sum_{l \in L_{i'j}} (t \cdot x_{i'jlt}) \geq 0$$

$$\forall j \in J, \forall i \in I_j, \forall i' \in \Gamma_{ij}^{-1} \cap W_j \quad (3.33)$$

$$\sum_{t=\beta_{ij}}^{\gamma_{ij}} \sum_{l \in L_{ij}} (t \cdot x_{ijlt}) - \sum_{t=\beta_{i'j}}^{\gamma_{i'j}} \sum_{g \in G_{i'j}} \sum_{r=1}^{R_g} (t \cdot z_{i'jgrt}) \geq p_{i'j}$$

$$\forall j, \forall i \in I_j, \forall i' \in \Gamma_{ij}^{-1} \cap (C_j^{\text{in}} \cup E_j) \quad (3.34)$$

$$\sum_{t=\beta_{ij}}^{\gamma_{ij}} \sum_{l \in L_{ij}} (t \cdot x_{ijlt}) - \sum_{t=\beta_{i'j}}^{\gamma_{i'j}} \sum_{g \in G_{i'j}} \sum_{r=1}^{R_g} (t \cdot z_{i'jgrt}) = p_{i'j}$$

$$\forall j \in J, \forall i \in I_j, \forall i' \in \Gamma_{ij}^{-1} \cap (D_j \cup C_j^{\text{ou}}) \quad (3.35)$$

$$\sum_{t=\beta_{ij}}^{\gamma_{ij}} \sum_{g \in G_{ij}} \sum_{r=1}^{R_g} (t \cdot z_{ijgrt}) - \sum_{t=\beta_{i'j}}^{\gamma_{i'j}} \sum_{g \in G_{i'j}} \sum_{r=1}^{R_g} (t \cdot z_{i'jgrt}) \geq p_{i'j}$$

$$\forall j \in J, \forall i \in E_j, \forall i' \in \Gamma_{i'j}^{-1} \cap (C_j \cup D_j \cup E_j) \quad (3.36)$$

$$\sum_{t=\beta_{ij}}^{\gamma_{ij}} \sum_{g \in G_{ij}} \sum_{r=1}^{R_g} (t \cdot z_{ijgrt}) - \sum_{t=\beta_{ij}}^{\gamma_{ij}} \sum_{l \in L_{ij}} (t \cdot x_{ijlt}) \geq 0$$

$$\forall j \in J, \forall i \in C_j^{\text{in}} \quad (3.37)$$

$$\sum_{t=\beta_{ij}}^{\gamma_{ij}} \sum_{g \in G_{ij}} \sum_{r=1}^{R_g} (t \cdot z_{ijgrt}) - \sum_{t=\beta_{ij}}^{\gamma_{ij}} \sum_{l \in L_{ij}} (t \cdot x_{ijlt}) = 0$$

$$\forall j \in J, \forall i \in D_j \cup C_j^{\text{ou}} \quad (3.38)$$

$$\sum_{t=\beta_{sj}}^{\gamma_{sj}} \sum_{l \in L_{sj}} (t \cdot x_{sjlt}) - \sum_{t=\beta_{ij}}^{\gamma_{ij}} \sum_{l \in L_{ij}} (t \cdot x_{ijlt}) \geq \delta_{ij}$$

$$\forall j \in J, \forall i \in I_j, s \in S_{ij} \cap I_j \quad (3.39)$$

$$\sum_{t=\beta_{sj}}^{\gamma_{sj}} \sum_{l \in L_{sj}} (t \cdot x_{sjlt}) - \sum_{t=\beta_{ij}}^{\gamma_{ij}} \sum_{l \in L_{ij}} (t \cdot x_{ijlt}) \leq \delta_{ij} \cdot D_{ij}$$

$$\forall j \in J, \forall i \in I_j, s \in S_{ij} \cap I_j \quad (3.40)$$

$$u_{iji'j'l} + u_{i'j'ijl} \geq \sum_{t=\beta_{ij}}^{\gamma_{ij}} x_{ijlt} + \sum_{t=\beta_{i'j'}}^{\gamma_{i'j'}} x_{i'j'lt} + \delta_{ij} + \delta_{i'j'} - 3$$

$$\forall j \in J, \forall i \in I_j^{in}, \forall j' \in J \setminus \{j\}, \forall i' \in I_{j'}^{in}, \forall l \in L_{ij} \cap L_{i'j'} \quad (3.41)$$

$$u_{iji'j'l} + u_{i'j'ijl} \leq 1$$

$$\forall j \in J, \forall i \in I_j^{in}, \forall j' \in J \setminus \{j\}, \forall i' \in I_{j'}^{in}, \forall l \in L_{ij} \cap L_{i'j'} \quad (3.42)$$

$$\sum_{t=\beta_{i'j'}}^{\gamma_{i'j'}} (t \cdot x_{i'j'lt}) - \sum_{t=\beta_{sj}}^{\gamma_{sj}} \sum_{l \in L_{sj}} t \cdot x_{sjlt} \geq (u_{iji'j'l} - 1) \cdot M_{ij}$$

$$\forall j \in J, \forall i \in I_j^{in}, \forall j' \in J \setminus \{j\}, \forall i' \in I_{j'}^{in},$$

$$\forall l' \in L_{ij} \cap L_{i'j'}, s \in S_{ij} \cap I_j \quad (3.43)$$

3.5 APPROCHES DE RÉOLUTION CAS STATIQUE

Dans le cas statique du problème, on suppose que le nombre de patients et leurs parcours dans le circuit sont connus dès le début du processus de conception. Les durées des traitements et des déplacements sont connues et ne peuvent être anticipées. L'étude de la situation statique est essentielle car elle nous permet de modéliser le problème et de fournir des solutions de référence qui serviront de base comparative pour le cas dynamique. Les approches heuristiques offrent de nouvelles combinaisons de principes de résolution de problèmes qui peuvent utiliser des informations spécifiques au problème et être adaptées pour produire des solutions de haute qualité dans un délai adapté à une mise en œuvre en temps réel. Dans cette section, nous présentons les deux approches heuristiques proposées et aussi une heuristique constructive basée sur les règles métiers.

3.5.1 *Heuristique constructive basée sur les règles métiers (PRH)*

Nous avons passé plusieurs jours au SUA pour observer son fonctionnement et nous avons longuement discuté avec le personnel médical. À partir de ces observations sur le terrain, nous avons constaté que certaines règles métiers sont utilisées pour gérer les patients, leur parcours et les tâches de soins dans ce dernier. Certaines de ces règles sont purement médicales et difficiles à implémenter. Par exemple, nous avons observé que les règles varient selon le type de diagnostic. En effet, si deux patients ont le même niveau d'urgence et la même heure d'arrivée, la façon dont les patients sont traités dépend du type de cas clinique.

Cependant, deux règles de base sont utilisées quel que soit le cas qui permet d'imiter la stratégie globale pour planifier les tâches des patients. Elles sont liées au niveau d'urgence et au temps d'attente écoulé relativement au délai acceptable pour la première évaluation médicale. Les patients sont principalement triés dans une liste selon leur niveau d'urgence. Cette liste est ensuite gérée dynamiquement au fil du temps. Lorsqu'un patient dont le niveau d'urgence est inférieur à celui qui se trouve au début de la liste a un délai d'attente qui se rapproche du délai acceptable (ou plus élevé), le niveau d'urgence associé augmente artificiellement. Cela permet de planifier la première évaluation médicale, puis les autres tâches à effectuer pour ce patient. Cela permet d'éviter de reporter indéfiniment les patients dont le niveau d'urgence est faible, car des patients dont le niveau d'urgence est relativement plus élevé peuvent arriver.

Nous avons appelé *Practitioners' Rules Heuristic (PRH)* la stratégie que nous avons mise en œuvre et qui respecte les contraintes dures présentées dans le modèle ILP. Les solutions trouvées par PRH sont utilisées comme solutions de départ pour les autres approches de résolution. Nous utilisons également ces solutions à titre comparatif.

3.5.2 *IDCH*

Nous proposons une heuristique itérative de destruction/construction *Iterative Destruction Construction Heuristic (IDCH)* pour améliorer une solution initiale. L'Algorithme 1 donne le schéma général de l'approche IDCH proposée. Nous utilisons la solution de PRH comme solution initiale pour IDCH.

L'algorithme proposé alterne deux phases : une phase de destruction et une phase de construction. À partir d'une solution initiale S_{cur} et à chaque itération, nous retirons aléatoirement k patients de la solution courante et les mettrons dans une liste L . Dans la phase de destruction, au maximum D_{max} patients peuvent être retirés à chaque itération globale avec $k \leq D_{max}$. La valeur de D_{max} agit comme un degré de diversification.

Elle est incrémentée après chaque itération globale non améliorante jusqu'à D_{limit} et réinitialisée à D_{init} après chaque itération améliorante de la meilleure solution trouvée. Nous utilisons ce mécanisme adaptatif pour équilibrer entre l'exploration de voisinage plus éloigné de la solution courante et l'intensification de la recherche pour améliorer la solution. L'objectif est de mieux explorer l'espace des recherches pour trouver une meilleure solution.

Algorithme 1 : structure générale de l'IDCH

```

Entrées :  $S_0$  solution initiale
Sorties :  $S_{best}$  la meilleure solution trouvée
Paramètres :  $D_{limit}$  limite pour le degré de diversification
                $D_{init}$  le degré de diversification initial
                $Iter_{max}$  nombre maximal d'itérations sans aucune amélioration
Variabes :  $S_{cur}$  solution courante,  $i$  indice d'itérations
                $D_{max}$  nombre maximum des patients à retirer de la solution
                $k$  nombre d'interventions à retirer sur plusieurs jours
                $S_i$  solution à l'itération  $i$ 

1  $Iter \leftarrow 0$ 
2  $D_{max} \leftarrow D_{init}$ 
3  $S_{best} \leftarrow S_{cur}$ 
4 tant que  $Iter < MaxIter$  faire
5    $k \leftarrow \text{rand}(1, D_{max})$ 
6    $L \leftarrow \text{PatientRandomDestruction}(S_{cur}, k)$ 
7    $\text{PatientPriorityAndSort}(L)$ 
8    $S_{cur} \leftarrow \text{PatientAddInOrder}(S_{cur}, L)$ 
9   si  $\text{Obj}(S_{cur}) < \text{Obj}(S_{best})$  alors
10     $S_{best} \leftarrow S_{cur}$ 
11     $Iter \leftarrow 0$ 
12     $D_{max} \leftarrow D_{init}$ 
13    sinon
14       $D_{max} \leftarrow \min(D_{max} + 1, D_{limit})$ 
15       $Iter \leftarrow Iter + 1$ 
16    fin
17  fin
18 fin
19 retourner  $S_{best}$ 

```

Dans la phase de construction, la procédure `PatientPriorityAndSort(L)` trie L en classant les patients retirés selon l'ordre décroissant de la contribution dans la fonction objectif. Puis, les patients contenus dans la liste triée L sont à nouveau planifiés afin d'obtenir une nouvelle solution réalisable S_{cur} .

Différentes solutions peuvent être obtenues en utilisant différentes listes de priorités. Nous avons choisi de calculer la contribution dans la fonction objectif de chaque patient de L retiré de la solution courante. Les patients sont ordonnés par ordre décroissant de la contribution calculée. Lorsqu'une meilleure solution est trouvée, on met à jour S_{best} , et on remet le compteur d'itération $Iter$ à zéro et D_{max} à sa valeur initiale D_{init} .

L'algorithme s'arrête lorsque $MaxIter$ itérations ont été effectuées sans aucune amélioration.

3.5.3 Méta-heuristique ALNS

Nous présentons la structure générale de notre méthode Adaptive Large Neighborhood Search (ALNS) appliqué au problèmes des urgences dans la section 3.5.3.1, puis nous détaillons les différents mécanismes de l'approche proposée. Pour une recherche de voisinage à grande échelle, des solutions similaires sont obtenues en modifiant la solution originale par l'application de plusieurs opérateurs de recherche locale. Nous explorons le voisinage d'une solution actuelle en appliquant d'abord un opérateur de destruction pour libérer les ressources. Ensuite, nous replanifions les tâches détruites en appliquant un opérateur de réparation dans le but d'améliorer la qualité de la solution

de départ. Nous présentons les six opérateurs de destruction et les quatre opérateurs de réparation que nous étudions dans les sections 3.5.3.2 et 3.5.3.3. Le degré de destruction doit être ajusté car il affecte la diversification. Un mécanisme de diversification adaptative est utilisé pour gérer le degré de destruction de certains opérateurs de destruction. Un couple d'opérateurs de destruction et de réparation joue le rôle d'un voisinage pour la recherche locale.

Le succès des opérateurs peut varier en fonction de l'instance du problème. Un choix adaptatif conduit généralement à de meilleurs résultats plutôt que de fixer le choix des opérateurs pour toute la durée de l'algorithme. Le mécanisme adaptatif présenté dans la section 3.5.3.4 est basé sur un principe de roulette que nous utilisons pour gérer le succès des opérateurs de destruction et le succès des opérateurs de réparation. Chaque opérateur a un poids qui représente sa part dans une roulette. Les poids sont mis à jour en fonction des performances, les opérateurs efficaces sont utilisés plus souvent que les opérateurs moins efficaces.

L'ALNS utilise ces opérateurs de destruction/réparation pour explorer le voisinage d'une solution. Cependant, cela peut ne pas être suffisant pour permettre au processus de recherche d'éviter ou d'échapper aux optima locaux. La méthode d'acceptation de l'ALNS a pour but de décider soit de continuer à explorer le voisinage d'une solution actuelle, soit de sélectionner une solution nouvellement créée. Une solution de moindre qualité peut être acceptée dans le but d'échapper aux optima locaux. Nous présentons dans la section 3.5.3.5 deux stratégies d'acceptation adaptatives.

L'ALNS que nous proposons utilise des mécanismes adaptatifs pour les procédures de destruction, de construction et d'acceptation. Plusieurs paramètres doivent être réglés pour obtenir une bonne efficacité de ces mécanismes adaptatifs.

3.5.3.1 Structure générale de l'ALNS

L'algorithme 2 donne la structure générale de l'ALNS que nous décrivons ici avant de fournir des informations plus détaillées.

Étant donné une solution initiale S , l'algorithme alterne les phases de destruction et de réparation sur une solution courante S_{cur} pour obtenir une solution S_i à chaque itération i (lignes 5-10). Lorsque la solution S_i remplit les critères d'acceptation, elle est retenue comme la nouvelle solution courante S_{cur} , et, le nombre d'itérations i est remis à zéro sinon il est incrémenté (lignes 11-15). La meilleure solution trouvée jusqu'à présent au cours des itérations S_{best} est conservée (lignes 16-18). À condition de respecter un nombre maximal d'itérations Iter_{max} sans aucune amélioration, l'algorithme ALNS s'arrête et renvoie alors S_{best} .

Pour les opérateurs de destruction orientés patients, un nombre aléatoire de k patients limité par D_{max} est retiré complètement ou partiellement (ligne 7). Un mécanisme de diversification adaptative gère D_{max} (lignes 20-24), ce qui permet de diversifier la recherche autour de la meilleure solution trouvée jusqu'à présent, avec l'intention de trouver une solution encore meilleure (voir section 3.5.3.2). L'opérateur de destruction est effectué en utilisant $\text{Destruction}(S_{\text{cur}}, k, O_{\text{des}})$ (ligne 8). Cela libère des ressources et des lieux. La liste des patients retirés de la solution est \mathcal{L}_{des} .

Algorithme 2 : Adaptive Large Neighborhood Search (ALNS) algorithm

```

Entrées :  $S_0$  solution initiale
Sorties :  $S_{best}$  la meilleure solution trouvée
Paramètres :  $D_{limit}$  limite pour le degré de diversification
                $Iter_{max}$  nombre maximal d'itérations sans aucune amélioration
Variabes :  $S_{cur}$  solution courante,  $i$  nombre d'itérations
                $D_{max}$  nombre maximum de patient à retirer
                $O_{des}$  l'opérateur de destruction sélectionné,  $O_{rep}$  l'opérateur de réparation sélectionné
                $k$  nombre de patients à retirer
                $\mathcal{L}_{des}$  liste des patients retirés
                $S_i$  solution à l'itération  $i$ 
1  $S_{best}, S_{cur} \leftarrow S_0$  // Initialiser la meilleure solution et la solution courante
2  $i \leftarrow 0$ 
3  $D_{max} \leftarrow 3$ 
4 tant que  $i < Iter_{max}$  faire
5    $O_{des} \leftarrow SelectDestructionOperator()$  // Sélectionner un opérateur de destruction
6    $k \leftarrow 1$ 
7   si  $\neg(ResourceDestroy(O_{des}) \wedge PlaceDestroy(O_{des}))$  alors  $k \leftarrow rand(1, D_{max})$ 
   // Retirer  $k$  patients, ou tous les patients qui utilisent une ressource choisie ou un lieu
8    $(S_i, \mathcal{L}_{des}) \leftarrow Destruction(S_{cur}, k, O_{des})$ 
9    $O_{rep} \leftarrow SelectRepairOperator()$  // Sélectionner un opérateur de réparation
10   $S_i \leftarrow Construction(S_i, \mathcal{L}_{des}, O_{rep})$ 
   // Procédure d'acceptation sur  $S_i$ , solution en cours d'exécution
11  si  $Accept(S_{best}, S_i)$  alors
12     $S_{cur} \leftarrow S_i$  //  $S_i$  est acceptée comme la nouvelle solution actuelle
13     $i \leftarrow 0$ 
14  fin
15  sinon
16     $i \leftarrow i + 1$ 
17  fin
18  si  $Obj(S_i) < Obj(S_{best})$  alors
19     $S_{best} \leftarrow S_i$  // Une nouvelle meilleure solution est trouvée
20     $D_{max} \leftarrow 3$ 
21  fin
22  sinon
   // Mise à jour du degré maximal de diversification adaptative
23    si  $\neg(ResourceDestroy(O_{des}) \wedge PlaceDestroy(O_{des}))$  alors  $D_{max} \leftarrow \min(D_{max} + 1, D_{limit})$ 
24  fin
   // Mise à jour des poids des deux roulettes (Destruction/Réparation)
25   $AdaptiveWeightsAdjustment(S_{best}, S_{cur}, S_i)$ 
26 fin
27 retourner  $S_{best}$ 

```

L'opérateur de réparation O_{rep} est sélectionné (ligne 9) parmi un ensemble d'opérateurs de réparation (voir la section 3.5.3.3). Ensuite, il est utilisé dans $Construction(S_i, \mathcal{L}_{des}, O_{rep})$ (ligne 10) pour planifier tous les patients retirés (\mathcal{L}_{des}). Étant donné une solution en cours S_i , l'ordre dans lequel les patients \mathcal{L}_{des} sont traités a un impact sur les durées d'attentes de chaque patient. Les opérateurs de réparation que nous proposons sont basés sur la façon dont les patients de \mathcal{L}_{des} sont triés.

Afin d'échapper aux optimums locaux, nous permettons l'acceptation d'une solution non améliorante (ligne 11). Le problème principal à résoudre est d'éviter de sélectionner une solution de trop faible qualité, conduisant à une perte de temps de traitement lors de l'exploration de son voisinage dans l'espoir d'obtenir une solution de meilleure qualité. Nous avons choisi d'implémenter deux stratégies d'acceptation adaptatives, que nous détaillons dans la section 3.5.3.5.

Il est important de permettre la sélection de bons opérateurs de destruction/construction au fil des itérations afin de pouvoir améliorer la solution courante. Le mécanisme

adaptatif que nous mettons en œuvre est détaillé dans la section 3.5.3.4. Les poids des deux roulettes pour la sélection des opérateurs O_{des} et O_{rep} sont mis à jour en utilisant $AdaptiveWeightsAdjustment(S_{best}, S_{cur}, S_i)$.

3.5.3.2 Opérateurs de destruction

Nous avons étudié six opérateurs de destruction, quatre sont orientés patient et deux autres sont orientés ressource/lieu. Tous ces opérateurs utilisent la solution courante S_{cur} comme entrée, et les opérateurs patient suppriment un certain nombre de patients complètement ou partiellement tandis que les opérateurs ressource/lieu suppriment tous les patients planifiés qui utilisent une ressource ou un lieu choisi. Étant donné un opérateur de destruction O_{des} , la fonction $Destruction(S_{cur}, k, O_{des})$ (voir Algorithme 2) renvoie S_i , la solution en cours avec les patients retirés. Les opérateurs de destruction que nous avons étudiés sont :

DA Destruction Aléatoire (patients) ;

DAP Destruction Aléatoire Partiellement (patients) ;

DO Destruction à partir d'une liste triée par ordre décroissant de contribution dans la fonction Objectif (patients) ;

DCW Destruction à partir d'une liste triée par ordre décroissant de durée d'attente multipliée par le poids CCMU (patients) ;

DR Destruction de tous les patients qui utilisent la Ressource la plus chargée (ressource/lieu) ;

DL Destruction de tous les patients qui utilisent le lieu le plus chargé (ressource/lieu).

DA et **DAP** retirent k patients de manière aléatoire de la solution et les insèrent l'une après l'autre dans \mathcal{L}_{des} . **DA** retire toutes les tâches des k patients choisis. **DAP** retire un nombre aléatoire des tâches des k patients choisis. **DO** évalue d'abord puis trie les patients par ordre décroissant de coût. Ensuite, les k patients dont le coût de planification est le plus élevé sont supprimés. Le coût de planification d'un patient p est évalué par la différence de coût, en supposant le retrait du patient dans la solution. Ceci vise à diminuer le coût. **DCW** évalue d'abord puis trie les patients par ordre décroissant de durée d'attente multipliée par le poids CCMU. Ensuite, les k patients dont l'évaluation est la plus élevée sont supprimés. **DR** choisit la ressource la plus chargée. Ensuite, tous les patients planifiés qui utilisent cette ressource sont supprimés de la solution. **DL** choisit le lieu le plus chargé. Ensuite, tous les patients planifiés qui utilisent ce lieu sont supprimés de la solution.

Tous ces opérateurs de destruction visent à libérer des ressources et des lieux, ce qui permet aux opérateurs de réparation de planifier autant d'interventions qu'ils le peuvent. **DAP** et **DA** sont des opérateurs de destruction purement aléatoires. Ils permettent de diversifier la recherche ([Hemmati et al., 2017]). **DR** et **DL** fonctionnent tous deux au prix d'un temps de traitement plus important mais cela permet de réduire le coût global $Obj(S_i)$ à l'aide des opérateurs de réparation.

À chaque itération, la fonction `SelectDestructionOperator()` (voir Algorithme 2) sélectionne un opérateur de destruction O_{des} qui est choisi parmi les six décrits ci-dessus à l'aide d'un mécanisme adaptatif basé sur un algorithme de sélection par roulette (voir Section 3.5.3.4).

Le degré de destruction k affecte la diversification et par conséquent la qualité des solutions. Il peut être fixe, choisi au hasard dans un intervalle dépendant de la taille de certaines caractéristiques de l'instance, ou varier progressivement au moment de l'exécution selon une stratégie. Nous avons choisi d'implémenter une stratégie adaptative de gestion de k pour les opérateurs de destruction orientés patient.

Dans l'algorithme 2, la valeur k est choisie au hasard comme $k \leftarrow \text{rand}(1, D_{max})$. Dans la mesure où un opérateur de destruction orienté patient a été choisi, k est limité par D_{max} . Nous définissons D_{max} comme le degré de diversification. Cette valeur est initialisée à 3 et ensuite incrémentée après chaque itération non améliorante comme $D_{max} \leftarrow \min(D_{max} + 1, D_{limit})$, lorsque la valeur de D_{limit} n'est pas atteinte. Dès qu'une amélioration est trouvée, nous réinitialisons D_{max} à 3 afin d'explorer entièrement le voisinage de la nouvelle meilleure solution S_{best} .

Le choix de la valeur de D_{limit} est important car cela permet de diversifier la recherche autour d'une solution pour trouver une meilleure solution. Cependant, une valeur trop grande pourra générer une dépense inutile de temps de traitement car une meilleure solution pourrait être trouvée dans une autre partie de l'espace de recherche à partir d'une solution de moindre qualité qui pourrait être sélectionnée grâce au mécanisme d'acceptation (voir la section 3.5.3.5). Au fil des itérations de l'ALNS, le mécanisme adaptatif permet de choisir parmi les opérateurs de destruction pour explorer le voisinage d'une solution.

3.5.3.3 Opérateurs de réparation

Dans la littérature, l'algorithme *Best Insertion Algorithm* (BIA) est souvent utilisé pour la réparation de solutions. Globalement, l'idée est d'insérer des éléments dans une solution courante à un endroit choisi comme le meilleur selon des critères spécifiques (Amarouche et al., 2020; Ben-Said et al., 2019).

En nous appuyant sur le concept général, nous avons proposé un algorithme dédié BIPA, comme le montre l'algorithme 3. Étant donné une solution S_i à l'itération i , nous calculons l'ensemble de 6-uplet $S_{i,p,t6}$, indexé par $t6$. Un uplet peut être utilisé pour planifier la tâche i du patient p . Un uplet $t6$ contient un lieu, une ressource, une heure d'entrée dans le lieu, une heure de sortie du lieu, une heure de début de traitement et une heure de fin de traitement. Le BIPA sélectionne le premier uplet réalisable pour programmer i , Ensuite, les ressources et la disponibilité des lieux sont mises à jour.

L'algorithme utilise la solution S_i et une liste ordonnée/priorisée des patients à planifier, désignée par \mathcal{L}_{des} . La liste \mathcal{L}_{des} contient les patients qui ont été retirés complètement ou partiellement en appliquant l'un des six opérateurs de destruction. Nous définissons la liste $\mathcal{L}_{des}^{(p)}$ pour chaque patient à planifier p . Nous utilisons cette liste pour stocker les tâches détruites de parcours de patient p dans l'étape de destruction de l'ALNS. La disposition des patients dans la liste \mathcal{L}_{des} a un impact sur les durées d'attentes pour

Algorithme 3 : Best Insertion Patient Algorithm (BIPA)

Entrées : S_i solution en cours à l'itération i
 \mathcal{L}_{des} liste ordonnée/priorisée des patients à planifier,
 $\mathcal{L}_{des}^{(p)}$ liste des tâches à planifier du patient p
Sorties : S_i solution réparée à l'itération i
Variables : $S_{i,p,t6}$ pour la tâche i du patient p ,
ensemble de 6-uplets
(lieu, ressource, date d'entrée, date de sortie, date début, date fin)
pouvant être utilisés pour la planification

```

1 tant que ( $\mathcal{L}_{des} \neq \text{nil}$ ) faire
    // Le patient suivant de la liste ordonnée des patients à planifier
2    $p \leftarrow \mathcal{L}_{des}.\text{pop\_front}()$ 
3   tant que ( $\mathcal{L}_{des}^{(p)} \neq \text{nil}$ ) faire
4      $S_{i,p,t6} \leftarrow \emptyset$ 
        // Sélectionner les insertions possibles pour la tâche  $i$  du patient  $p$ 
        // Insertion au plus tôt
5     Planifier la tâche  $i$  du patient  $p$  dans la solution  $S_i$  au plus tôt possible
6     Mettre à jour les disponibilités des lieux et des ressources
7   fin
8 fin
9 retourner  $S_i$ 

```

chaque patient à planifier. Nous proposons et examinons quatre manières différentes de construire l'ordre des patients dans \mathcal{L}_{des} .

Étant donné une liste de priorités \mathcal{L}_{des} , nous essayons d'insérer les patients l'un après l'autre dans la solution S_i . Pour chaque patient p , nous suivons le graphe orienté du parcours de ce patient pour planifier les tâches.

L'algorithme d'insertion BIPA adapté et la liste ordonnée/priorisée \mathcal{L}_{des} jouent ensemble le rôle d'un opérateur de réparation. On peut avoir différentes solutions en utilisant différentes listes de priorités. Nous proposons trois façons d'ordonner/prioriser les patients dans \mathcal{L}_{des} :

LO Liste ordre croissant de contribution dans la fonction Objectif ;

LCW Liste ordre croissant de durée d'attente multipliée par le poids CCMU ;

LA Liste Aléatoire des patients.

L'opérateur **LO** vise à donner la priorité aux patients qui ont un coût élevé dans la fonction objectif, en supposant que nous aurons la possibilité de réduire ces coûts. L'opérateur **LCW** prend en considération les durée d'attentes avec les degrés d'urgence des patients. L'objectif de cet opérateur est d'essayer de réduire les temps d'attentes des patients plus urgents en premier. L'opérateur **LA** est une construction aléatoire dans le but d'explorer de l'espace de recherche. Au fil des itérations de l'ALNS, le mécanisme adaptatif permet de choisir parmi les trois opérateurs de réparation pour explorer le voisinage d'une solution.

3.5.3.4 Mécanisme adaptatif de destruction/réparation

L'idée est de favoriser les choix futurs d'opérateurs performants en modifiant les poids utilisés dans un algorithme de roulette qui sélectionne les opérateurs. Les poids de la roulette utilisée pour les opérateurs de destruction et les poids de l'autre roulette

utilisée pour les opérateurs de réparation sont mis à jour en utilisant la procédure AdaptiveWeightsAdjustment en fonction des performances obtenues.

L'algorithme ALNS proposé utilise quatre opérateurs de destruction et trois de réparation. Nous désignons O comme étant soit un opérateur de destruction, soit un opérateur de réparation. Le succès des opérateurs peut varier en fonction de l'instance du problème. L'ajustement des poids $w(O)$ de la roulette est nécessaire pour augmenter la probabilité que les opérateurs efficaces soient utilisés plus souvent que les opérateurs moins efficaces. L'ajustement dynamique est le seul moyen d'assurer une réévaluation permanente des poids des opérateurs.

Deux approches peuvent être utilisées pour mettre à jour les poids $w(O)$: à chaque itération ou périodiquement après un certain nombre d'itérations. L'avantage de la première approche est que les poids sont à jour à tout moment mais au détriment du temps de traitement puisque les poids et les probabilités des opérateurs pour la prochaine itération doivent être calculés à chaque itération. Au contraire, en mettant périodiquement à jour les poids après un certain nombre d'itérations p_u , on gagne en temps de traitement. Cependant, cela implique de déterminer une bonne valeur pour p_u . Une valeur élevée permet de gagner du temps de traitement mais ne permet pas au mécanisme adaptatif de jouer pleinement son rôle, ce qui peut aboutir à des solutions de mauvaise qualité. Pour gagner du temps de traitement, nous avons choisi la deuxième approche. Le paramétrage de p_u est donc nécessaire pour obtenir un bon compromis entre le gain de temps de calcul et la qualité des solutions.

Le mécanisme adaptatif est le même pour la sélection de l'opérateur de destruction ou de réparation. Le nombre de fois où l'opérateur O a été utilisé pendant p_u itérations est noté $u(O)$. Le succès de l'opérateur O est noté $s(O)$, fixé à zéro au début de la période de p_u itérations (à déterminer). Étant donné la solution S_i à l'itération i , la solution actuelle S_{cur} , et la meilleure solution trouvée jusqu'à présent S_{best} , la valeur de $s(O)$ est augmentée à chaque itération comme :

$$s(O) + \sigma_i \quad \text{où } \sigma_i = \begin{cases} \sigma_1 & \text{si } \text{Obj}(S_i) \leq \text{Obj}(S_{best}) \\ \sigma_2 & \text{si } \text{Obj}(S_i) \leq \text{Obj}(S_{cur}) \\ \sigma_3 & \text{sinon} \end{cases}$$

Dans le premier cas, l'opérateur O améliore ou égale la meilleure solution trouvée jusqu'à présent, tandis que dans le second cas, il améliore ou égale la solution courante. Dans le dernier cas, nous récompensons l'opérateur lorsqu'une nouvelle solution est plus mauvaise mais acceptée selon la stratégie d'acceptation (voir section 3.5.3.5). Nous avons $\sigma_1 > \sigma_2 > \sigma_3$. L'opérateur est ainsi favorisé lorsque la qualité relative ou absolue de la solution augmente. Ces paramètres doivent être réglés car ils ont un impact sur la qualité de la solution. Nous avons choisi de formuler σ_2 comme $\sigma_2 = \theta \cdot \sigma_1$ et σ_3 comme $\sigma_3 = \phi \cdot \sigma_2$ où $\theta, \phi \in [0, 1]$. À condition qu'une valeur initiale soit fixée pour σ_1 , les paramètres de σ_2, σ_3 peuvent être réglés en faisant varier le couple de paramètres (θ, ϕ, p_u) . Les poids pour les itérations à venir sont calculés comme suit :

$$w(O) = \begin{cases} (1-l)w(O) + l \frac{s(O)}{u(O)}, & \text{si } u(O) > 0 \\ (1-l)w(O), & \text{si } u(O) = 0 \end{cases}$$

Le facteur de réaction $l \in [0, 1]$ est un paramètre qui contrôle l'influence du succès récent d'un opérateur sur son poids.

Les six opérateurs de destruction et les trois opérateurs de réparation utilisés permettent d'effectuer des changements importants d'une solution à ses voisines, ce qui peut empêcher le processus de recherche d'être bloqué dans des optima locaux. Le succès des opérateurs peut varier en fonction de l'instance du problème. Le mécanisme adaptatif joue un rôle important en récompensant les opérateurs efficaces tout au long des itérations. Pour rendre le mécanisme adaptatif efficace, les paramètres θ, ϕ, ρ doivent être réglés.

3.5.3.5 Stratégie d'acceptation adaptative

Nous avons choisi d'implémenter deux stratégies d'acceptation adaptatives selon des schémas probabiliste ou déterministe.

La première stratégie est basée sur l'algorithme du recuit simulé. Il est proposé à l'origine par [Laarhoven et al., 1987]. La stratégie vise à simuler les effets du recuit physique. L'objectif est d'obtenir une configuration stable pour les atomes d'un échantillon métallique. Ce processus se déroule en deux étapes. Dans la première étape, un métal est chauffé jusqu'au point de fusion. Dans la deuxième partie, le métal refroidit lentement, permettant aux atomes de former la structure solide. L'heuristique imite la deuxième étape, qui est le recuit contrôlé. Comme un métal qui recuit, la température de départ T_0 fournie diminue progressivement au fil du temps. La stratégie d'acceptation est plus susceptible d'accepter des solutions non améliorantes au début.

À chaque itération, la solution S_i est acceptée si $\text{Obj}(S_i) \leq \text{Obj}(S_{\text{best}})$, sinon elle est acceptée avec une probabilité de $\exp(\frac{\text{Obj}(S_{\text{best}}) - \text{Obj}(S_i)}{T_i})$ où T_i est la température à l'itération i . La température diminue à chaque itération par un facteur α , généralement $0.88 \leq \alpha \leq 0.99$. L'idée est de laisser l'algorithme explorer l'espace de recherche pendant un certain temps tout en acceptant une solution avec une qualité un peu moins bonne que celle qu'il utilise actuellement, puis d'orienter la recherche vers des solutions qui ont de meilleures probabilités d'acceptation par la suite car $(\text{Obj}(S_{\text{best}}) - \text{Obj}(S_i))$ va tendre vers zéro.

La deuxième stratégie d'acceptation adaptative est basée sur l'algorithme *record-to-record travel*. Il est proposé à l'origine par [Dueck, 1993], il s'agit d'une variante déterministe du recuit simulé. Une solution de faible qualité est acceptée en fonction de la différence entre le coût $\text{Obj}(S_i)$ de la solution à l'itération i et le coût de la meilleure solution $\text{Obj}(S_{\text{best}})$ par rapport à une valeur appelée *record-deviation*. C'est une stratégie basée sur un seuil. Ce mécanisme simple mais efficace a l'avantage de ne dépendre que d'un seul paramètre, qui est la valeur de l'écart record, noté ici Dev (voir [Dueck, 1993]). Le paramètre de *record-deviation* Dev joue un rôle central dans le contrôle de l'acceptation des solutions de faible qualité. Dans le schéma général de l'algorithme présenté dans [Dueck, 1993], le paramètre de *record-deviation* est fixe et utilisé à l'origine

comme $Dev \leq Obj(S_i) - Obj(S_{best})$. Nous avons choisi de l'implémenter par rapport à un pourcentage comme $Obj(S_i) \leq (1 + Dev) \cdot Obj(S_{best})$ avec $Dev \in [0, 1]$. Néanmoins, plus ce paramètre est grand, plus nous explorons inutilement des solutions de faible qualité parce que le mécanisme *record-to-record* est activé trop tôt. Pour avoir la possibilité d'explorer d'abord le voisinage d'une solution courante en utilisant les opérateurs de destruction/réparation, nous proposons une stratégie d'acceptation adaptative que nous gérons dans la procédure $Accept(S_{best}, S_i)$ (voir Algorithme 2). Ainsi, la valeur de Dev n'est pas fixe, elle peut augmenter ou être remise à zéro tout au long du processus d'exploration de l'espace de recherche. Cependant, l'augmentation de Dev peut se faire au détriment du mécanisme de diversification adaptative qui gère la valeur de D_{max} , le nombre maximum d'interventions qui peuvent être retirées de S_i , lorsqu'une solution de moindre qualité est rencontrée. Dans l'algorithme 6, nous mettons à jour $D_{max} \leftarrow \min(D_{max} + 1, D_{limit})$ lorsque $Obj(S_i) \geq Obj(S_{best})$ pour certains opérateurs de destruction. Un bon compromis entre ces deux mécanismes doit être trouvé pour prioriser les recherches locales.

Dans la procédure $Accept(S_{best}, S_i)$, nous mettons en œuvre un mécanisme adaptatif qui augmente la valeur de Dev d'une valeur constante Δ_{Dev} (à déterminer) lorsqu'aucune amélioration n'est rencontrée tout au long d'un nombre défini d'itérations $Iter_D$ (à déterminer). Le paramètre *record-deviation* est égal à zéro au début ($Dev = 0$). Lorsque le nombre défini d'itérations $Iter_D$ sans amélioration a été effectué, Dev est augmenté de la valeur de Δ_{Dev} pour permettre au mécanisme de diversification de jouer son rôle. Les solutions de faible qualité sont plus souvent acceptées, ce qui permet à la recherche de sortir des optima locaux. La valeur Dev augmente de Δ_{Dev} tant que les opérateurs de destruction/réparation ne parviennent pas à trouver une meilleure solution. Nous remettons le paramètre Dev à zéro si une amélioration locale ou globale se produit.

Par conséquent, les deux paramètres à régler pour rendre cette stratégie d'acceptation adaptative efficace sont l'incrément de *record-deviation* Δ_{Dev} et le nombre d'itérations sans amélioration $Iter_D$ qui induit l'incrément de Dev .

3.6 APPROCHE DE RÉOLUTION CAS DYNAMIQUE

Dans la partie statique du problème (voir section 3.3 et section 3.5), nous avons supposé que toutes les informations sur les patients (arrivée, parcours, durée des tâches) sont connues au début de planification. Les disponibilités des ressources et des lieux sont aussi connues au début de la planification et ne changent pas.

Dans le cas dynamique, les informations concernant le patient et les tâches ne sont pas disponibles avant son arrivée. Après le triage, à l'aide des premières observations et du motif de venue, un parcours de soins est affecté au patient, mais le parcours peut changer durant le séjour du patient. Une durée de tâche de soin est estimée, la durée réelle de cette tâche n'est connue que lorsque le traitement est terminé.

Dans un monde idéal où l'on disposerait d'informations parfaites sur l'arrivée des patients, il serait possible d'utiliser directement les approches de résolution proposées dans le cas statique. Nous proposons une méthode de planification prédictive-réactive

(voir section 2.4.0.3) pour résoudre le problème dynamique qui lui doit tenir compte aussi d'évènements.

L'ordonnancement prédictif-réactif se compose de deux étapes :

- générer un planning sans tenir compte des événements futurs ;
- mettre à jour le planning quand des événements surviennent et aussi pour mettre à jour les durées réelles des tâches.

Il y a deux préoccupations qui doivent être abordées :

Quand un nouveau planning est-il nécessaire ?

Un nouveau planning doit être établi chaque fois qu'un événement imprévu se produit pour la mise à jour et la prise en compte des temps de traitement réels. Nous avons déterminé les événements suivant dans notre environnement de planification :

1. L'arrivée un nouveau patient ;
2. La sortie d'un patient ;
3. L'indisponibilité d'une ressource ;
4. Une tâche prévue n'est pas encore terminée, durée réelle > durée prévue ;
5. Une tâche prévue est déjà terminée, durée réelle < durée prévue ;

Pour l'évènement 1, les tâches pour le nouveau patient sont à planifier avec leurs durées prévues. Pour tous les évènements il faut mettre à jour le planning. Par exemple des tâches initialement planifiées ont réellement nécessité plus de temps, d'autres ont nécessité moins de temps. La mise à jour des durées est nécessaire, puis il faut effectuer une replanification.

Comment établir un nouveau planning ?

Le calcul d'un nouveau planning consiste à, soit le calcul d'une nouvelle solution avec des nouvelles données générées par l'évènement, soit la réparation du planning actuel non réalisable. Le pire scénario est de devoir recalculer un nouveau planning complet à partir de zéro chaque fois qu'un événement imprévu se produit. En outre, dans la plupart des cas, le temps nécessaire au calcul reste assez élevé. Nous avons choisi d'utiliser les méthodes proposées pour la résolution de problème du cas statique. Nous avons donc adapté ces méthodes pour trouver le planning initial et aussi pour la partie replanification. Lorsqu'une tâche est entrée dans le système, son temps de traitement réel reste inconnu tant qu'elle n'est pas achevée, son temps de traitement réel n'est connu que lorsque son traitement est terminé.

L'algorithme 4 décrit les différentes étapes de l'approche dynamique proposée. Nous avons un axe de temps entre 0 et H fin de l'horizon de planification. Au début de l'horizon, nous construisons une solution initiale avec une des méthodes de construction de ALNS avec les patients qui arrivent à $t = 0$. Nous utilisons ALNS pour améliorer la solution. Le temps d'exécution de l'algorithme ALNS est limité par Temps_{\max} à définir

Algorithme 4 : Algorithme dynamique

Entrées : Temps_{max} temps max, H l'horizon
Sorties : S_H solution jusqu'à la fin d'horizon de la planification
Variables : S_t solution à l'instance t

```

1 début
2   t ← 0
3   Construire une solution initiale S0
4   Amélioration avec ALNS(S0, [0, H], Tempsmax)
5   tant que t < H faire
6     si événement(t) alors
7       Détruire la solution St partiellement ou complètement avec l'intervalle [t, H]
8       Mettre à jour les informations réelles connues au moment de l'évènement
9       Construire la solution St avec un des voisinages de ALNS
10      Amélioration de la solution avec ALNS(St, [t, H], Tempsmax)
11    fin
12  retourner SH
13 fin
14 fin

```

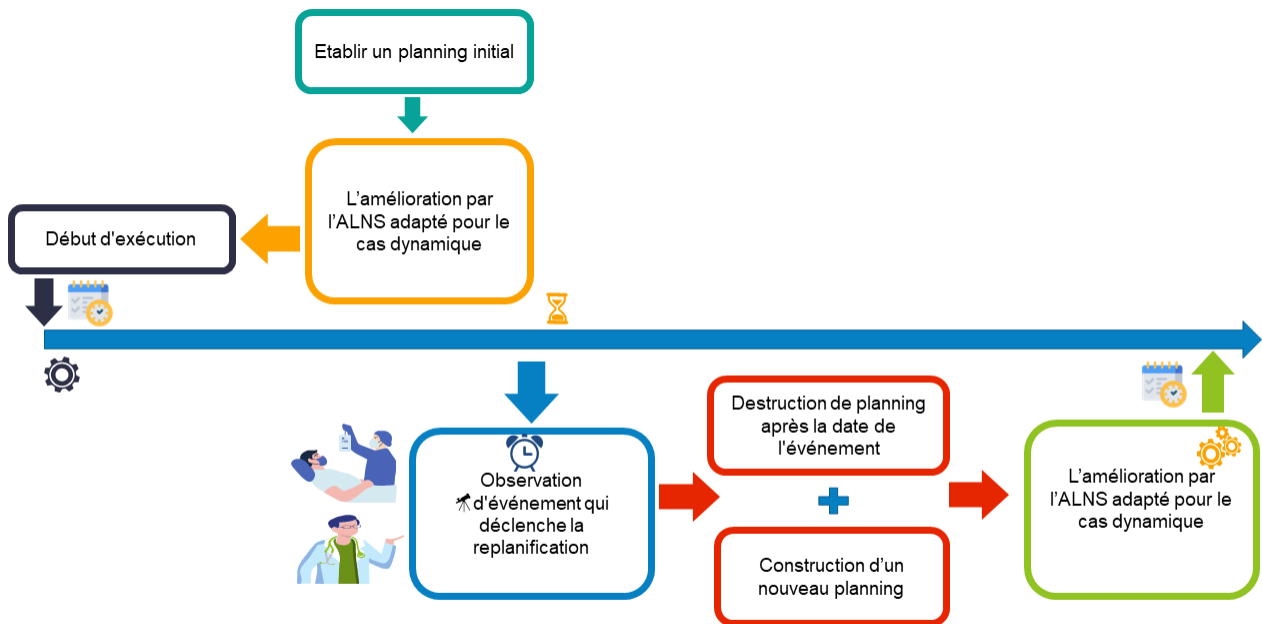


Fig 3.6: Illustration de l'algorithme dynamique proposé

avec les praticiens du SUA. Un temps trop important n'est pas concevable pour assurer la réactivité.

L'algorithme dynamique réagit à chaque occurrence d'un événement à un instant t et déclenche la replanification (ligne 4). Si la replanification est déclenchée, nous détruisons la solution S_t partiellement ou complètement. Nous ne détruisons que les tâches entre l'intervalle $[t, H]$ avec t date de l'évènement et H fin de l'horizon de planification. Nous mettons à jour les informations générées par l'évènement. Nous construisons après une solution initiale à améliorer. Nous utilisons l'algorithme ALNS pour améliorer la solution. La Figure 3.6 illustre les différentes étapes de l'algorithme dynamique.

3.7 RÉSULTATS

Dans nos expériences, nos objectifs étaient : (i) de fournir un résumé et des indications sur la façon de procéder au paramétrage de l'IDCH et de l'ALNS ; (ii) d'évaluer l'efficacité des opérateurs de destruction/réparation et du mécanisme adaptatif pour la destruction/réparation, analyser les résultats afin de montrer si un composant contribue à l'amélioration de la qualité des solutions en considérant également le temps de traitement ; (iii) d'évaluer la qualité des solutions obtenues par IDCH et ALNS sur des instances de référence ; (iv) d'évaluer la qualité des solutions obtenues par l'approche dynamique sur des instances de référence.

Les tests ont été réalisés en utilisant C++ et la Standard Template Library (STL), compilés avec GCC sous Linux, sur une machine avec un Intel(R) Xeon(R) Gold 6138 @ 2.00 GHz.

3.7.1 *Benchmark et protocole expérimental*

Nous avons eu à notre disposition des données provenant des urgences du CHU Jeanne de Flandres à Lille. Ces données couvraient une période de quatre ans entre 2016 et 2020. Elles contiennent certaines informations sur les arrivées des patients, leurs conditions et leurs déplacements pendant leur séjour aux urgences.

Cependant, les données quotidiennes sont incorrectement ou mal remplies. Il n'a pas été possible de déterminer si les mêmes activités étaient effectuées dans la même succession aux mêmes lieux. Les temps indiqués à un lieu donné incluent le temps d'attente et ne peuvent être considérés comme fiables. Il n'est pas clairement indiqué quels membres du personnel médical sont responsables de la prestation de ces soins. Nous avons donc décidé de générer des instances sur la base de données plus fiables et en échangeant avec le personnel médical. Ce qui nous a permis d'établir des parcours de traitement (parcours patients) les plus courants.

Lorsqu'un patient arrive aux urgences, le parcours de traitement pour ce patient n'est pas encore déterminé dans son intégralité. Chaque patient arrive avec un motif (e.g. motif traumatologique) et repart avec un diagnostic établi. Il reçoit les soins et les examens qui lui ont été prescrits à la suite de la consultation initiale avec le patient. Nous avons construit des graphes de tâches pour les parcours les plus courants aux SUA (voir annexe A). Bien que certains patients se présentent aux urgences pour la même raison, les tâches assignées après l'évaluation initiale peuvent différer.

Pour générer une instance avec un ensemble de parcours patients, nous établissons les arrivées de patients selon un processus de Poisson non stationnaire et les parcours des patients sont choisis aléatoirement.

Nous avons généré 27 instances qui varient en nombre de patients avec des arrivées entre 8h00 et 9h00. Ces instances servent de base de test pour les solutions proposées dans le cas statique. Des informations détaillées concernant ces instances sont présentées dans le Tableau 3.6. Ces instances sont utilisées pour évaluer les approches proposées dans le cas statique.

Pour le l'évaluation d'algorithme dynamique, nous avons généré 24 instances qui varient en nombre de patients entre 20 et 120 avec des arrivées entre 8h00 et 21h00. Des informations détaillées concernant ces instances sont présentées dans le Tableau 3.7.

Instance	Patients	Nombre de tâches de soin
a_01	4	46
a_02	4	47
a_03	4	55
a_04	5	82
a_05	5	37
a_06	5	51
a_07	6	53
a_08	6	97
a_09	6	71
a_10	7	69
a_11	7	91
a_12	7	45
a_13	8	81
a_14	8	74
a_15	8	82
a_16	9	77
a_17	9	92
a_18	9	101
a_19	10	87
a_20	10	155
a_21	10	76
a_22	11	132
a_23	11	100
a_24	11	110
a_25	12	169
a_26	12	133
a_27	12	88

Tableau 3.6: Caractéristiques des instances générées

3.7.2 Paramétrage de IDCH

L'approche de résolution IDCH que nous proposons a un seul paramètre $MaxIter$, le nombre maximal d'itérations sans aucune amélioration. Ce paramètre influence la qualité de la solution mais aussi le temps d'exécution. Pour prendre en compte les caractéristiques de chaque instance, nous choisissons de formuler $MaxIter$ comme $\beta \cdot \frac{\sum_j |I_j|}{\max(|L|, \sum_g |R_g|)}$, $\beta \in \mathbb{N}$, avec I_j l'ensemble des tâches du patient j , L l'ensemble des lieux et R_g l'ensemble des ressources de type g . Pour une instance donnée, le ratio augmente

Instance	Patients	Nombre de tâches de soin
D ₁	20	229
D ₂	20	238
D ₃	30	336
D ₄	30	288
D ₅	40	429
D ₆	40	419
D ₇	50	602
D ₈	50	606
D ₉	60	606
D ₁₀	60	650
D ₁₁	60	673
D ₁₂	70	782
D ₁₃	70	713
D ₁₄	80	918
D ₁₅	80	914
D ₁₆	90	1022
D ₁₇	90	1022
D ₁₈	100	1029
D ₁₉	100	1172
D ₂₀	110	1370
D ₂₁	110	1160
D ₂₂	120	1329
D ₂₃	120	1400
D ₂₄	120	1420

Tableau 3.7: Caractéristiques des instances générées

avec le nombre total de tâches et diminue avec le nombre de places ou le nombre total de ressources pour les tâches de soins. Pour calibrer le paramètre, nous avons effectué des expériences préliminaires sur un sous-ensemble de 10 instances choisies aléatoirement parmi les instances de référence. Comme IDCH est une méthode de recherche aléatoire, les expériences ont été répétées cinq fois avec une graine aléatoire différente.

Pour ajuster les paramètres, nous évaluons les résultats en utilisant le pourcentage d'erreur relative (RPE). Nous définissons RPE comme suit : $RPE = 100 \times \frac{Z_{best} - Z_{max}}{Z_{best}}$ où Z_{best} désigne le résultat de l'ILP, et Z_{max} désigne le meilleur résultat obtenu parmi les cinq exécutions effectuées. Ensuite, nous calculons le RPE moyen pour les 10 instances choisies pour le paramétrage.

Les résultats sont présentés dans la Figure 3.7 et la Figure 3.8. Comme le montrent les deux figures, le RPE moyen diminue lorsque la valeur β augmente, et le temps de traitement augmente avec la valeur β . Le RPE moyen se stabilise après $\beta = 130$. Nous avons choisi $\beta = 130$, qui est un bon compromis entre la qualité de la solution et le temps de traitement inférieur à 60s.

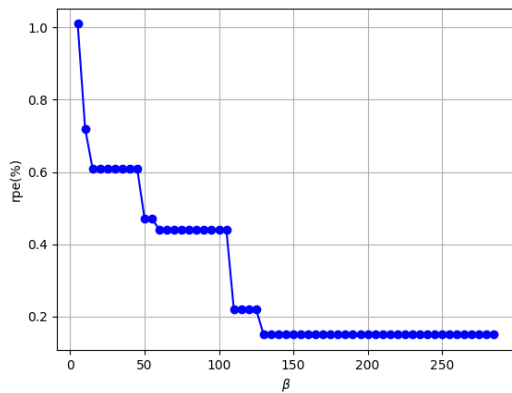


Fig 3.7: Paramétrage de β pour IDCH, effet de β sur le RPE.

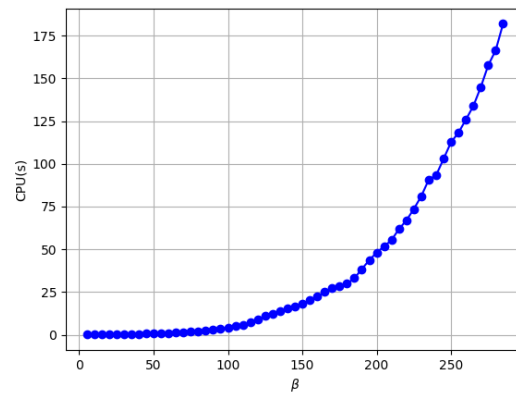


Fig 3.8: Paramétrage de β pour IDCH, effet de β sur le CPU.

3.7.3 Paramétrage de ALNS acceptance base *Record-to-record*

Nous présentons les expériences que nous avons menées afin de déterminer les valeurs des paramètres qui ont été utilisées pour tester notre algorithme ALNS. Comme décrit dans la section 3.5.3.1, l'ALNS avec la stratégie d'acceptation *Record-to-record* que nous proposons possède trois ensembles de paramètres :

- Δ_{Dev} et $Iter_D$, le taux d'augmentation de l'écart par rapport au record et le nombre d'itérations sans amélioration avant augmentation de l'écart ;
- $\sigma_1, \sigma_2, \sigma_3, p_u$, et l , les trois premiers sont les scores ajoutés après l'utilisation d'un opérateur de destruction ou de réparation, p_u est le nombre d'itérations avant l'ajustement des poids des opérateurs, et l est le facteur de réaction ;
- $Iter_{max}$, le nombre maximal d'itérations sans amélioration.

Pour D_{limit} , le nombre maximal de patients retirés par les opérateurs de destruction, nous avons choisis d'avoir la possibilité à détruire toute la solution donc D_{limit} est le nombre total des patient dans le service. Certains de ces paramètres sont calculés au cours de l'algorithme ALNS en fonction de certaines caractéristiques de l'instance traitée ou sont relatifs à d'autres paramètres.

La première colonne du Tableau 3.8 indique les paramètres tels que présentés dans la section 3.5.3.1. La deuxième colonne décrit comment ils sont calculés et les paramètres associés qui doivent être réglés.

Nous avons exécuté des expériences préliminaires sur 10 exemples choisis aléatoirement parmi les instances de référence afin de calibrer tous les paramètres. Le code a été exécuté cinq fois, chaque fois avec une graine aléatoire différente, car ALNS est une méthode de recherche aléatoire.

Pour ajuster les paramètres, nous évaluons les résultats à l'aide du pourcentage d'erreur relatif (RPE). Nous définissons le RPE comme suit : $RPE = 100 \times \frac{Z_{best} - Z_{max}}{Z_{best}}$

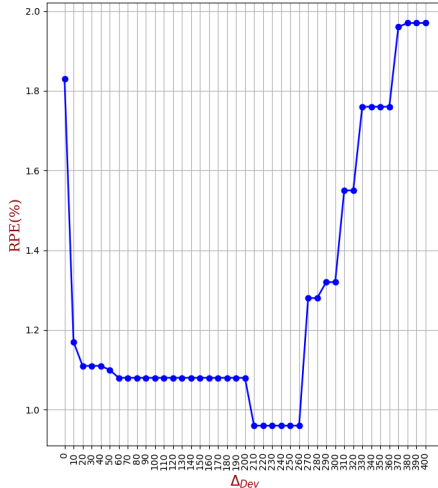
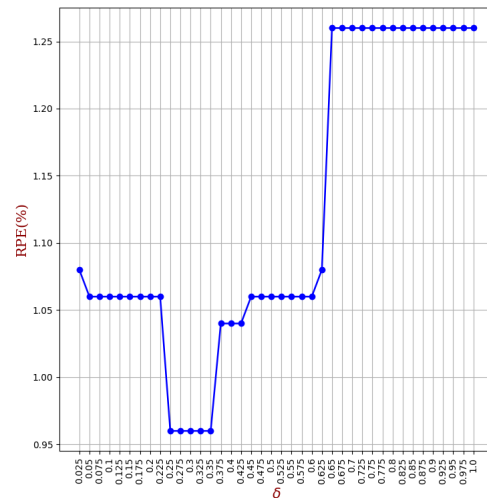
Calculé comme, et paramètres à régler	
$\sigma_1, \sigma_2, \sigma_3$	σ_1 fixe, ensuite $\sigma_2 = \theta \cdot \sigma_1$ et $\sigma_3 = \phi \cdot \sigma_2$ paramètres (θ, ϕ) à régler
$Iter_D$	$\delta \cdot Iter_{max}$, paramètre δ à régler
$Iter_{max}$	$\gamma \cdot \frac{\sum_j I_j }{\max(L , \sum_g R_g)}$, paramètre γ à régler

Tableau 3.8: Paramètres θ , ϕ , δ et γ à régler.

où Z_{best} désigne le meilleur résultat obtenu sur l'ensemble des exécutions effectuées pour une instance, et Z_{max} désigne le meilleur résultat que nous avons obtenu parmi les cinq exécutions réalisées. Ensuite, la moyenne des valeurs RPE est calculée pour les cinq instances choisies de manière aléatoire, et cette moyenne est indiquée en pourcentage dans les figures que nous présentons et que nous commentons ci-dessous.

Nous commençons par régler Δ_{Dev} , puis nous réglons chaque paramètre, l'un après l'autre, en considérant les trois ensembles de paramètres présentés. Les valeurs initiales que nous utilisons pour régler le premier Δ_{Dev} sont les suivantes $\delta = 0.2$, $\sigma_1 = 50$, $\theta = 0.5$, $\phi = 0.2$, $p_u = 70$, $l = 0.7$ et $\gamma = 100$. Ensuite, pour régler un paramètre, nous conservons le meilleur paramétrage des autres paramètres trouvés avant de procéder à son réglage.

Stratégie d'acceptation adaptative basée sur le principe *Record-to-record* ; réglage de Δ_{Dev} et $Iter_D$.

Fig 3.9: Réglage de Δ_{Dev} , effet de sur le RPE.Fig 3.10: Réglage de $Iter_D$, effet de δ sur le RPE.

Pour calibrer la valeur de Δ_{Dev} , nous l'avons fait varier entre 0% et 400% avec des incréments de 10%. Dans la Figure 3.9, nous montrons que lorsque Δ_{Dev} varie entre zéro et 200%, le RPE diminue. Nous atteignons ensuite un plateau sur la courbe pour 6 valeurs qui donnent la plus petite valeur de RPE. Ensuite, les valeurs moyennes de RPE augmentent lorsque Δ_{Dev} augmente. Nous fixons la valeur de $\Delta_{Dev} = 210\%$ pour la

déviations par rapport au record. Il s'agit de la plus petite valeur pour laquelle la valeur moyenne de RPE est minimale.

Nous procédons ensuite au paramétrage de δ pour $Iter_D$, le nombre d'itérations sans amélioration avant l'augmentation de la déviation. Dans la Figure 3.10, 5 bonnes valeurs de δ peuvent être choisies ; la première est 0.25. Si $Iter_D$ est petit, la déviation autorisée augmente rapidement et l'algorithme accepte trop des solutions de mauvaise qualité. En revanche, si $Iter_D$ est grand, l'algorithme sera piégé dans un optimum local. Nous avons choisi $\delta = 0.25$.

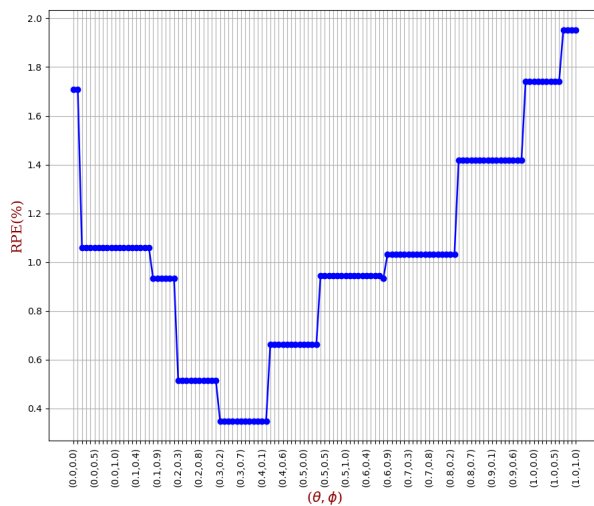


Fig 3.11: Réglage de σ_2 et σ_3 par rapport à σ_1 , effet de θ et ϕ sur le RPE.

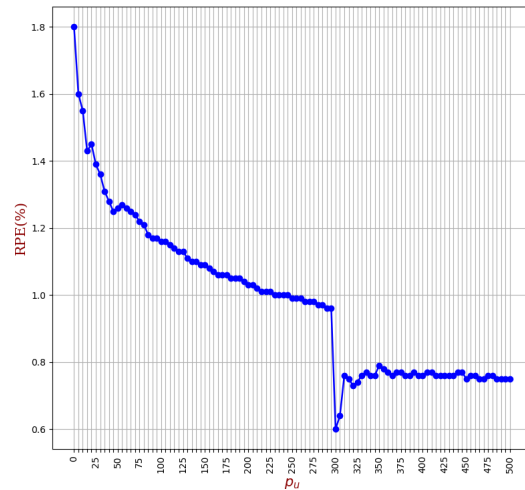


Fig 3.12: Effet de p_u sur le RPE.

Paramètres d’ajustement du poids adaptatif ; réglage de $\sigma_1, \sigma_2, \sigma_3, p_u$ et l .

Nous commençons par régler les paramètres d’ajustement des poids $\sigma_1, \sigma_2, \sigma_3$, les scores ajoutés après l’utilisation d’un opérateur de destruction ou de réparation. Étant donné que $\sigma_1 > \sigma_2 > \sigma_3$ selon la récompense habituelle des opérateurs par rapport à la qualité de la solution précédente S_i , de la solution courante S_{cur} et de la meilleure

solution trouvée jusqu'à présent, S_{best} , nous avons choisi de fixer d'abord σ_1 à 50 et nous calculons $\sigma_2 = \theta \cdot \sigma_1$ et $\sigma_3 = \phi \cdot \sigma_2$ tels que $\theta, \phi \in [0, 1]$.

La Figure 3.11 montre la courbe que nous obtenons. Nous avons choisi $\theta = 0.3$ et $\phi = 0.2$, la première valeur du couple (θ, ϕ) qui donne les meilleures performances par rapport au RPE moyen.

Pour paramétrer p_u , le nombre d'itérations avant d'ajuster les poids des opérateurs, nous le faisons varier entre 0 et 500 itérations avec un pas de 5. La Figure 3.12 montre clairement que le RPE est meilleur lorsque $p_u = 300$, et nous avons choisi de fixer p_u à 300.

Pour trouver la meilleure valeur de l , le facteur de réaction qui contrôle l'influence du succès récent d'un opérateur sur son poids, nous le faisons varier entre 0 et 1 avec un incrément de 0.1. Les valeurs moyennes de RPE présentées sur la Figure 3.13 révèlent que de bonnes solutions sont obtenues lorsque l est compris entre 0.6 et 1. Les itérations précédentes semblent avoir une petite influence sur le calcul du poids. Nous avons choisi de fixer l à 0.6.

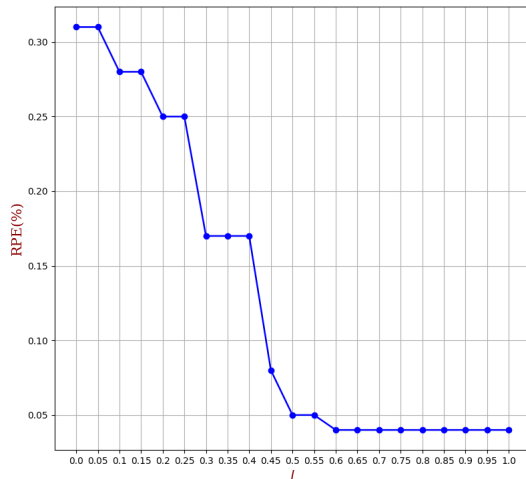


Fig 3.13: Effet de l sur le RPE.

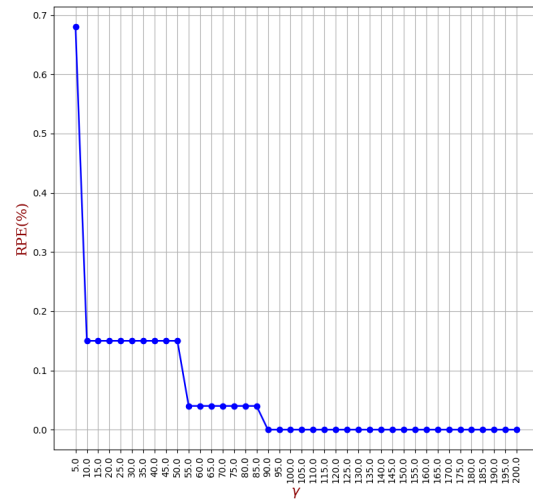


Fig 3.14: Réglage de $Iter_{max}$, effet de γ sur le RPE.

Nombre maximum d'itérations sans amélioration ; réglage de $Iter_{max}$

Enfin, nous procédons au paramétrage de $Iter_{max}$, le nombre maximal d'itérations sans amélioration. Le temps de traitement nécessaire pour trouver une solution de bonne qualité varie en fonction des instances et ce paramètre a un impact important. Nous avons choisi de formuler $Iter_{max}$ comme $\gamma \cdot \frac{\sum_j |I_j|}{\max(|L|, \sum_g |R_g|)}$ où $\gamma \in \mathbb{N}$ avec I_j l'ensemble des tâches du patient j , L l'ensemble des lieux et R_g l'ensemble des ressources de type g . Pour une instance donnée, le ratio augmente avec le nombre total de tâches et diminue avec le nombre de places ou le nombre de types de ressources pour les tâches de soins.

Dans la Figure 3.22, le RPE moyen diminue à mesure que γ augmente, puis il se stabilise à partir de $\gamma = 90$. Le temps de traitement augmente avec la valeur de γ . Plus les valeurs de γ sont grandes, plus les temps de traitement sont importants, ce qui est

inefficace. Nous avons choisi $\gamma = 90$, qui est un bon compromis entre la qualité de la solution et le temps de traitement.

Les résultats du paramétrage final sont les suivants : $\Delta_{Dev} = 210\%$, $\delta = 0.25$, $\sigma_1 = 50$, $\theta = 0.3$, $\phi = 0.2$, $p_u = 300$, $l = 0.60$ et $\gamma = 90$. Les valeurs des paramètres sont utilisées dans la suite pour nos expériences sur toutes les instances de référence. Elles ont été choisies pour obtenir un bon compromis entre la qualité de la solution et le temps de traitement.

3.7.4 Paramétrage de ALNS acceptance base recuit simulé

Nous présentons les expériences que nous avons menées pour trouver les valeurs des paramètres qui ont été utilisées pour tester notre approche ALNS avec le recuit simulé comme stratégie d'acceptation des solutions non améliorantes. L'ALNS que nous proposons possède quatre ensembles de paramètres :

- T_0 et α , la température initiale et le facteur de refroidissement ;
- $\sigma_1, \sigma_2, \sigma_3, p_u$, et l , les trois premiers sont les scores ajoutés après l'utilisation d'un opérateur de destruction ou de réparation, p_u est le nombre d'itérations avant l'ajustement des poids des opérateurs, et l est le facteur de réaction ;
- $Iter_{max}$, le nombre maximal d'itérations sans amélioration.

Certains de ces paramètres sont déterminés par l'algorithme ALNS en fonction de certaines caractéristiques de l'instance traitée, tandis que d'autres sont déterminés en fonction des uns des autres ou d'autres paramètres. Pour ces paramètres, la première colonne du Tableau 3.9 indique les paramètres tels que présentés dans la section 3.5.3.1. La deuxième colonne décrit comment ils sont calculés et les paramètres associés qui doivent être réglés.

	Calculé comme, et paramètres à régler
$\sigma_1, \sigma_2, \sigma_3$	σ_1 fixe, ensuite $\sigma_2 = \theta \cdot \sigma_1$ et $\sigma_3 = \phi \cdot \sigma_2$ paramètres (θ, ϕ) à régler
$Iter_{max}$	$\gamma \cdot \frac{\sum_j I_j }{\max(L , \sum_g R_g)}$, paramètre γ à régler

Tableau 3.9: Paramètres $\alpha, \theta, \phi, \delta$ et γ à régler.

Nous avons réalisé des expériences exploratoires sur un sous-ensemble de 10 instances choisis au hasard parmi les instances de référence afin de calibrer tous les paramètres. Ces instances ont été choisies dans l'ensemble des instances de référence. Compte tenu du fait que l'ALNS est une méthode de recherche aléatoire, les tests ont été effectués cinq fois avec une nouvelle graine aléatoire à chaque fois.

Nous commençons par régler le premier T_0 , puis nous réglons chaque paramètre, l'un après l'autre, en considérant les trois ensembles de paramètres présentés. Les valeurs initiales que nous utilisons pour régler T_0 sont les suivantes $\sigma_1 = 50$, $\alpha = 0.99$, $\theta = 0.5$,

$\phi = 0.2$, $p_u = 70$, $l = 0.7$ et $\gamma = 120$. Ensuite, pour régler un paramètre, nous conservons le meilleur paramétrage des autres paramètres trouvés avant de procéder à son réglage.

Stratégie d'acceptation adaptative basée sur le recuit simulé ; réglage de T_0 et α .

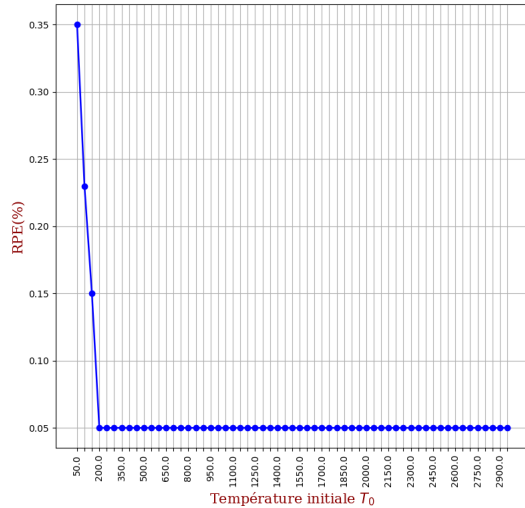


Fig 3.15: Réglage de T_0 , effet de T_0 sur le RPE.

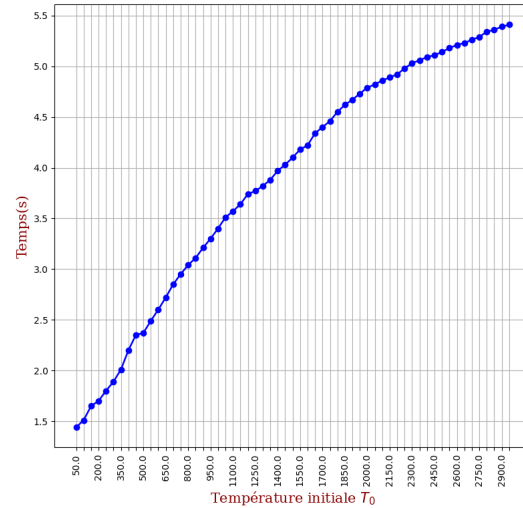


Fig 3.16: Réglage de T_0 , effet de T_0 sur le CPU.

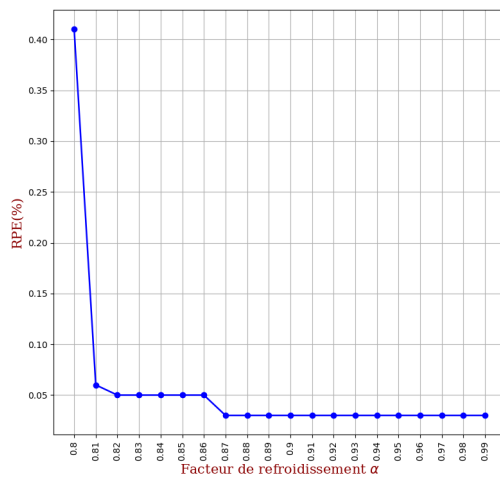


Fig 3.17: Réglage de α , effet de α sur le RPE.

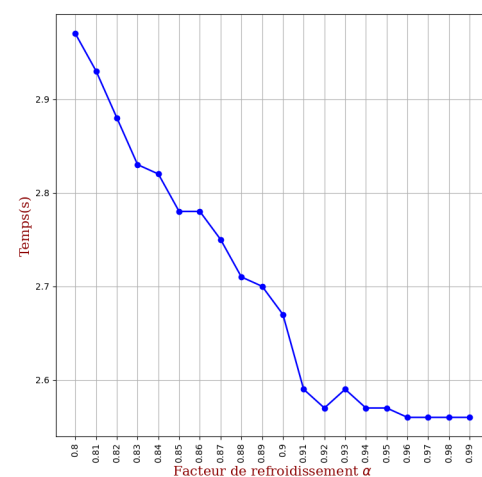


Fig 3.18: Réglage de α , effet de α sur le CPU.

Pour calibrer la valeur de T_0 , nous l'avons fait varier entre 50 et 3000 avec des incréments de 50. Dans la Figure 3.15, nous montrons que lorsque T_0 varie entre 50 et 200, le RPE diminue rapidement. Nous atteignons ensuite un plateau sur la courbe pour le reste des valeurs qui donnent la plus petite valeur de RPE. Le réglage de

la température initiale est important car il affecte directement le nombre de mauvaises solutions qui sont acceptées. L'algorithme s'appuie sur l'acceptation de nombreuses mauvaises solutions initiales afin d'échapper aux optima locaux dans l'espace de recherche. Si ce paramètre est trop faible, il se bloquera, alors que si ce paramètre est trop élevé, il est possible que il effectue une marche aléatoire dans l'espace de recherche pendant une partie significative de la recherche ce qui induit un temps CPU plus élevé. La Figure 3.16 montre ce phénomène, le temps CPU augmente avec augmentation de T_0 . Nous fixons la valeur de $T_0 = 200$.

Nous procédons ensuite au paramétrage de α , le facteur de refroidissement. L'algorithme utilise ce paramètre pour diminuer la température au fil du temps, de sorte que la probabilité d'accepter une solution plus mauvaise diminue avec le temps.

Dans la Figure 3.17, plusieurs bonnes valeurs de α peuvent être choisies entre 0.87 et 0.99. Dans la Figure 3.18, le temps CPU diminue avec l'augmentation de α et il se stabilise à partir de $\alpha = 0.96$. Nous avons choisi $\alpha = 0.96$.

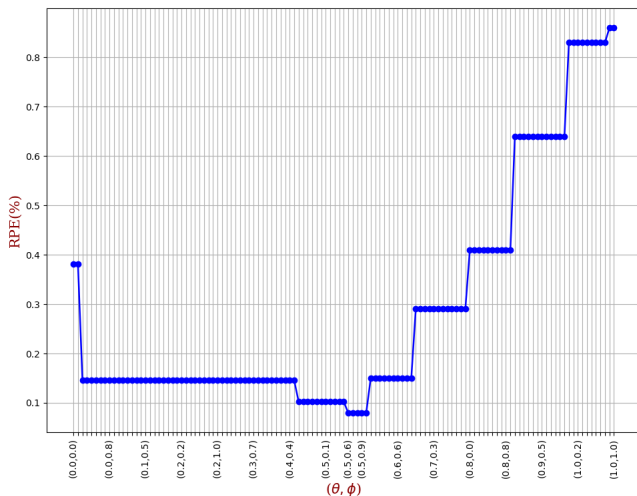


Fig 3.19: Réglage de σ_2 et σ_3 par rapport à σ_1 , effet de θ et ϕ sur le RPE.

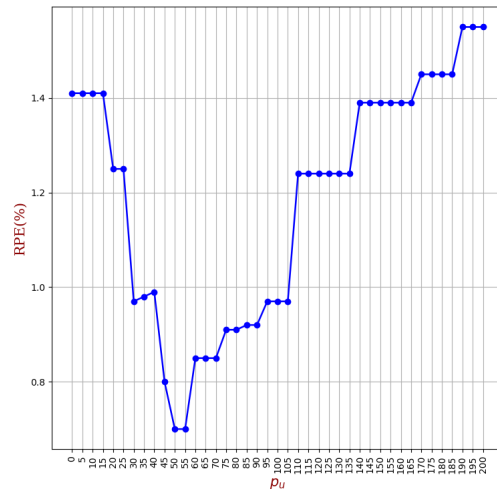
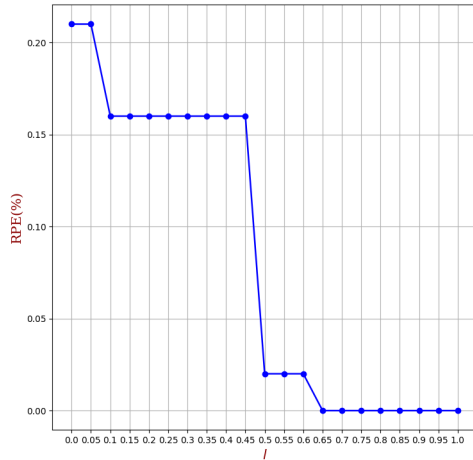
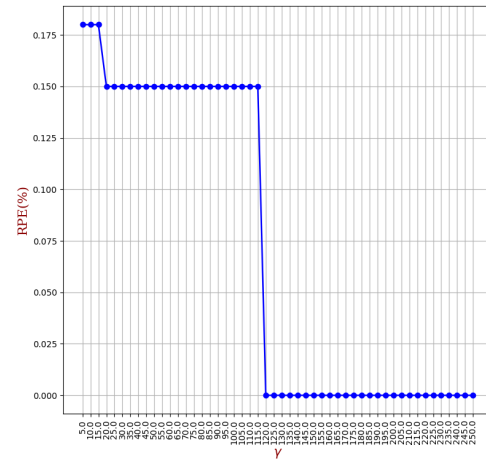


Fig 3.20: Effet de p_u sur le RPE.

Fig 3.21: Effet de l sur le RPE.Fig 3.22: Réglage de $Iter_{max}$ effet de γ sur le RPE.

Paramètres d'ajustement du poids adaptatif ; réglage de $\sigma_1, \sigma_2, \sigma_3, p_u$ et l .

Nous commençons par régler les paramètres d'ajustement des poids $\sigma_1, \sigma_2, \sigma_3$, les scores ajoutés après l'utilisation d'un opérateur de destruction ou de réparation. Étant donné que $\sigma_1 > \sigma_2 > \sigma_3$ selon la récompense habituelle des opérateurs par rapport à la qualité de la solution précédente S_i , de la solution courante S_{cur} , et de la meilleure solution trouvée jusqu'à présent, S_{best} , nous avons choisi de fixer d'abord σ_1 à 50 et nous calculons $\sigma_2 = \theta \cdot \sigma_1$ et $\sigma_3 = \phi \cdot \sigma_2$ tels que $\theta, \phi \in [0, 1]$.

La Figure 3.19 montre la courbe que nous obtenons. Nous avons choisi $\theta = 0.4$ et $\phi = 0.6$, la première valeur du couple (θ, ϕ) donnant les meilleures performances par rapport au RPE.

Pour paramétrer p_u , le nombre d'itérations avant d'ajuster les poids des opérateurs, nous le faisons varier entre 0 et 200 itérations avec un pas de 5. La Figure 3.20 montre que le RPE est meilleur lorsque p_u est compris entre 50 et 55, et nous avons choisi de fixer p_u à 50.

Pour trouver la meilleure valeur de l , le facteur de réaction qui contrôle l'influence du succès récent d'un opérateur sur son poids, nous le faisons varier entre 0 et 1 avec un incrément de 0.1. Les valeurs moyennes de RPE présentées sur la Figure 3.21 révèlent que de bonnes solutions sont obtenues lorsque l est compris entre 0.65 et 1. Les itérations précédentes semblent avoir une petite influence sur le calcul du poids. Nous avons choisi de fixer l à 0.65.

Nombre maximum d'itérations sans amélioration ; réglage de $Iter_{max}$

Enfin, nous procédons au paramétrage de $Iter_{max}$, le nombre maximal d'itérations sans amélioration. Le temps de traitement nécessaire pour trouver une solution de bonne qualité varie en fonction des instances et ce paramètre a un impact important. Nous avons choisi de formuler $Iter_{max}$ comme $\gamma \cdot \frac{\sum_j |I_j|}{\max(|L|, \sum_g |R_g|)}$ où $\gamma \in \mathbb{N}$ avec I_j l'ensemble des tâches du patient j , L l'ensemble des lieux et R_g l'ensemble des ressources de type g

Dans la Figure 3.22, le RPE moyen diminue à mesure que γ augmente, puis il se stabilise à partir de $\gamma = 120$. Le temps de traitement augmente avec la valeur de γ . Plus les valeurs de γ sont grandes, plus les temps de traitement sont importants, ce qui est inefficace. Nous avons choisi $\gamma = 120$, qui est un bon compromis entre la qualité de la solution et le temps de traitement et qui reste généralement inférieur à une seconde avec cette valeur.

Les résultats du paramétrage final sont les suivants : $T_0 = 200$, $\alpha = 0.96$, $\sigma_1 = 50$, $\theta = 0.4$, $\phi = 0.6$, $p_u = 50$, $l = 0.65$ et $\gamma = 120$. Les valeurs des paramètres sont utilisées dans la suite pour nos expériences sur toutes les instances de référence. Elles ont été choisies pour obtenir un bon compromis entre la qualité de la solution et le temps de traitement.

3.7.5 Évaluation des composantes de l'ALNS

En désactivant individuellement les opérateurs de destruction, de réparation et le processus adaptatif de destruction et de réparation, nous pouvons évaluer leur efficacité relative. Cela nous permet de discuter des résultats et de montrer, en tenant compte du temps de traitement, si les différentes parties contribuent ou non à l'obtention de bonnes solutions.

Dans le Tableau 3.10, nous montrons les résultats pour les quatre opérateurs de destruction pour la version ALNS-SA avec recuit simulé comme stratégie d'acceptation de solution non améliorante (voir section 3.5.3.2). Sous chaque rubrique, nous indiquons les valeurs minimales (colonne *Min* pour *Obj(S)*) et les temps de traitement moyens en secondes (colonne *t(s)*) que nous obtenons en 10 exécutions. À des fins de comparaison, nous présentons sous le titre *ALNS-SA* les valeurs que nous obtenons en exécutant l'ALNS avec recuit simulé avec tous les composants activés (également dans le Tableau 3.12). Dans le Tableau 3.10 pour ALNS-SA nous obtenons des valeurs minimales différentes pour 10 instances.

L'ALNS-SA sans **Destruction Aléatoire** (la colonne *Sans DA*) obtient des résultats inférieurs à ceux de l'ALNS en utilisant tous les composants. Le **DA** opérateur de destruction est nécessaire. Les patients sont choisies au aléatoirement ce qui permet d'éviter de rester bloqué dans un optimum local.

L'ALNS-SA sans **Destruction Aléatoire Partielle** (la colonne *Sans DAP*) obtient des résultats inférieurs à ceux d'ALNS en utilisant tous les composants. **DAP** et **DA** sont des opérateurs de destruction purement aléatoires. Ils permettent de diversifier la recherche.

L'ALNS-SA sans **Destruction** de tous les patients qui utilisent la **Ressource** la plus chargé (la colonne *Sans DR*) et **Destruction** de tous les patients qui utilisent le lieu le plus chargé (la colonne *Sans DL*) obtient des résultats inférieurs à ceux d'ALNS en utilisant tous les composants. ces deux opérateurs de destruction visent à libérer des ressources et des lieux.

L'ALNS-SA sans **Destruction** à partir d'une liste triée par ordre décroissant de contribution dans la fonction **Objectif** (la colonne *Sans DO*) et **Destruction** à partir d'une liste triée par ordre décroissant de durée d'attente multipliée par le poids CCMU (la colonne

Sans DCW) obtient des résultats inférieurs à ceux de l'ALNS-SA en utilisant tous les composants. Les deux opérateurs de destruction sont nécessaires car ils visent à minimiser la fonction objectif directement.

	ALNS-SA		Sans DAP		Sans DO		Sans DCW		Sans DR		Sans DA		Sans DL	
	Min	t(s)	Min	t(s)	Min	t(s)	Min	t(s)	Min	t(s)	Min	t(s)	Min	t(s)
a_01	1110	0.35	1110.0	0.33	1110.0	0.29	1110.0	0.35	1110.0	0.32	1110.0	0.3	1110.0	0.38
a_02	1140	0.35	1140.0	0.3	1140.0	0.29	1140.0	0.32	1140.0	0.38	1140.0	0.3	1140.0	0.4
a_03	1160	0.7	1160.0	0.75	1160.0	0.78	1160.0	0.76	1160.0	0.78	1160.0	0.77	1160.0	0.78
a_04	1525	1.61	1525.0	1.78	1525.0	1.94	1525.0	0.81	1525.0	2.15	1525.0	1.84	1525.0	4.45
a_05	2765	0.82	2945.0	0.54	2765.0	0.43	2765.0	0.41	2765.0	0.47	2765.0	0.26	2765.0	0.39
a_06	1300	1.41	1300.0	1.62	1300.0	1.45	1300.0	1.42	1300.0	1.58	1300.0	1.62	1300.0	1.64
a_07	1805	0.65	1805.0	0.48	1805.0	0.3	1805.0	0.36	1805.0	0.47	1805.0	0.38	1805.0	0.46
a_08	2120	20.99	2125.0	9.49	2120.0	4.66	2120.0	5.96	2140.0	12.27	2120.0	5.83	2145.0	16.06
a_09	2750	4.61	2815.0	1.4	2770.0	1.5	2755.0	0.45	2775.0	0.63	2835.0	1.16	2780.0	1.37
a_10	1805	0.8	1805.0	0.39	1805.0	0.42	1805.0	0.37	1805.0	0.59	1805.0	0.48	1805.0	0.61
a_11	1850	8.41	1850.0	1.86	1850.0	1.61	1850.0	1.53	1850.0	1.83	1850.0	1.4	1850.0	1.72
a_12	1600	0.4	1600.0	0.33	1600.0	0.31	1600.0	0.36	1600.0	0.34	1600.0	0.35	1600.0	0.39
a_13	2090	1.44	2090.0	0.94	2090.0	0.68	2090.0	0.65	2090.0	1.07	2090.0	0.63	2090.0	1.0
a_14	2530	0.89	2530.0	0.43	2530.0	0.37	2530.0	0.55	2530.0	0.49	2530.0	0.43	2530.0	0.64
a_15	1940	1.9	1950.0	1.33	1940.0	1.04	1940.0	0.65	1940.0	1.2	1940.0	1.02	1940.0	0.99
a_16	4860	2.15	4860.0	0.85	4860.0	1.08	4860.0	1.01	4860.0	1.13	4860.0	1.19	4860.0	1.33
a_17	3205	3.88	3205.0	1.29	3205.0	1.25	3205.0	1.49	3205.0	1.47	3205.0	1.46	3205.0	2.06
a_18	4045	7.63	4065.0	0.71	4055.0	1.6	4055.0	1.76	4055.0	2.71	4065.0	1.3	4055.0	1.77
a_19	2835	1.38	2835.0	0.87	2835.0	0.89	2835.0	0.78	2835.0	1.0	2835.0	0.79	2835.0	1.07
a_20	7325	129.1	7550.0	113.86	7435.0	367.13	7325.0	255.22	7325.0	286.83	7570.0	645.09	7325.0	479.26
a_21	4085	1.3	4085.0	1.1	4085.0	1.15	4085.0	0.94	4085.0	1.1	4085.0	0.97	4085.0	1.16
a_22	3750	13.18	3785.0	4.17	3765.0	2.17	3745.0	2.07	3755.0	3.0	3765.0	2.1	3750.0	3.58
a_23	5955	9.35	5970.0	1.75	5955.0	1.97	5955.0	2.06	5955.0	2.05	5955.0	1.65	5955.0	2.16
a_24	4325	5.84	4325.0	2.73	4330.0	2.56	4325.0	1.27	4325.0	0.99	4330.0	1.77	4325.0	2.63
a_25	5030	115.08	5040.0	20.69	5030.0	32.79	5110.0	48.46	5030.0	45.21	5030.0	34.51	5030.0	41.59
a_26	3970	15.62	3995.0	2.02	3970.0	2.99	3970.0	3.12	3970.0	3.21	3970.0	3.75	3970.0	3.22
a_27	4310	2.22	4310.0	1.32	4310.0	0.89	4310.0	0.93	4310.0	1.24	4310.0	0.85	4310.0	1.19

Tableau 3.10: Impact des opérateurs de destruction DAP, DO, DCW, DR, DA et DL pour ALNS avec recuit simulé.

Dans le Tableau 3.11, nous montrons les résultats pour les quatre opérateurs de destruction pour la version ALNS-RE avec *record-to-record* comme stratégie d'acceptation de solution non améliorante (voir section 3.5.3.2). Sous chaque rubrique, nous indiquons aussi les valeurs minimales (colonne *Min* pour $Obj(S)$) et les temps de traitement moyens en secondes (colonne *t(s)*) que nous obtenons en 10 exécutions. À des fins de comparaison, nous présentons sous le titre *ALNS-RE* les valeurs que nous obtenons en exécutant l'ALNS avec *record-to-record* avec tous les composants activés (également dans le Tableau 3.13).

Dans le Tableau 3.11 pour ALNS-RE nous obtenons des valeurs minimales différentes de la fonction objectif pour 15 instances. Pour l'ALNS-RE, la Destruction Aléatoire (DA) et Destruction Aléatoire Partielle (DAP) permettent aussi la diversification et sont nécessaires. Les résultats sont meilleurs pour 15 instances. En ce qui concerne les opérateurs DR et DL qui se basent sur les ressources et les lieux, leurs résultats sont moins tranchés quand il sont utilisés ou non dans la version ALNS-RE. Dans certains

cas leur retrait permet des améliorations mais dans d'autres cas l'utilisation de ces deux opérateurs aussi permet des améliorations.

	ALNS-RE		Sans DAP		Sans DO		Sans DCW		Sans DR		Sans DA		Sans DL	
	Min	t(s)	Min	t(s)	Min	t(s)	Min	t(s)	Min	t(s)	Min	t(s)	Min	t(s)
a.01	1110	0.3	1110.0	0.42	1110.0	0.32	1110.0	0.31	1110.0	0.39	1110.0	0.35	1110.0	0.44
a.02	1140	0.33	1140.0	0.42	1140.0	0.36	1140.0	0.34	1140.0	0.38	1140.0	0.36	1140.0	0.44
a.03	1160	0.73	1160.0	1.0	1160.0	0.72	1160.0	0.6	1160.0	0.82	1160.0	0.64	1160.0	0.84
a.04	1525	1.4	1525.0	1.95	1525.0	1.66	1525.0	1.55	1525.0	1.85	1525.0	1.57	1525.0	1.69
a.05	2765	0.38	2945.0	0.83	2765.0	0.75	2765.0	0.73	2765.0	0.87	2765.0	0.75	2765.0	1.04
a.06	1300	1.51	1300.0	1.57	1300.0	1.27	1300.0	1.21	1300.0	1.61	1300.0	1.48	1300.0	1.51
a.07	1805	0.45	1805.0	0.76	1805.0	0.62	1805.0	0.56	1805.0	0.67	1805.0	0.61	1805.0	0.73
a.08	2165	7.98	2120.0	21.45	2125.0	20.78	2120.0	18.32	2120.0	24.99	2120.0	21.16	2120.0	21.66
a.09	2775	1.08	2810.0	4.55	2750.0	3.44	2750.0	3.29	2750.0	4.65	2755.0	4.12	2750.0	4.97
a.10	1805	0.34	1805.0	1.17	1805.0	0.8	1805.0	0.75	1805.0	0.94	1805.0	0.82	1805.0	0.99
a.11	1850	1.43	1855.0	7.32	1970.0	6.05	1865.0	6.02	1855.0	10.84	1855.0	4.24	1855.0	6.68
a.12	1600	0.35	1610.0	0.45	1610.0	0.43	1610.0	0.38	1630.0	0.48	1610.0	0.39	1620.0	0.47
a.13	2090	0.79	2190.0	1.65	2090.0	1.23	2290.0	1.18	2090.0	1.56	2090.0	1.25	2090.0	1.44
a.14	2530	0.48	2535.0	1.37	2530.0	0.86	2540.0	0.9	2530.0	1.21	2530.0	0.95	2530.0	1.28
a.15	1940	0.95	1950.0	2.03	1940.0	2.08	1940.0	1.44	1940.0	2.36	1940.0	2.08	1940.0	2.36
a.16	4860	0.82	4860.0	2.73	4860.0	1.96	4860.0	1.99	4860.0	2.69	4860.0	2.21	4860.0	3.05
a.17	3205	1.58	3205.0	4.65	3205.0	3.61	3205.0	3.12	3205.0	4.02	3205.0	3.59	3245.0	3.97
a.18	4045	1.62	4065.0	11.14	4055.0	10.81	4065.0	9.64	4060.0	11.39	4065.0	9.74	4065.0	12.06
a.19	2835	0.81	2835.0	1.75	2835.0	1.67	2840.0	1.25	2835.0	1.74	2835.0	1.32	2835.0	1.56
a.20	7370	368.17	7340.0	1333.15	7325.0	1465.9	7320.0	1391.96	7320.0	1583.37	7295.0	1379.46	7595.0	1285.66
a.21	4085	1.01	4085.0	1.58	4085.0	1.15	4085.0	1.25	4085.0	1.57	4085.0	1.27	4085.0	1.6
a.22	3745	1.54	3785.0	16.89	3750.0	14.19	3755.0	12.75	3745.0	14.29	3750.0	12.21	3750.0	18.95
a.23	5960	2.17	5955.0	13.14	5955.0	10.39	5955.0	8.1	5955.0	13.01	5955.0	14.56	5955.0	12.79
a.24	4325	1.44	4325.0	10.01	4325.0	8.28	4325.0	6.74	4325.0	8.75	4325.0	8.07	4125.0	8.3
a.25	5040	42.06	5040.0	87.02	5030.0	97.27	5030.0	101.33	5035.0	95.65	5030.0	92.43	5230.0	108.01
a.26	3970	3.8	3995.0	17.04	3970.0	14.65	3990.0	11.13	3970.0	12.66	3970.0	11.33	3970.0	15.64
a.27	4310	1.24	4310.0	2.79	4310.0	2.66	4310.0	2.26	4310.0	2.4	4310.0	2.47	4310.0	2.54

Tableau 3.11: Impact des opérateurs de destruction DAP, DO, DCW, DR, DA et DL pour ALNS avec *record-to-record*.

Dans les deux Tableaux 3.12 et 3.13, nous montrons les résultats pour les trois opérateurs de réparation (voir Section 3.5.3.3) et pour le mécanisme adaptatif (voir section 3.5.3.4) qui gère le choix des opérateurs de destruction et de réparation.

Globalement, pour les trois opérateurs de réparation, la Liste Aléatoire des patients, la Liste ordre croissant de contribution dans la fonction Objectif, la Liste ordre croissant de durée d'attente multipliée par le poids CCMU nous observons d'abord qu'aucun n'a réussi à atteindre les résultats de l'approche ALNS-SA ou ALNS-RE en utilisant tous ces composants pour toutes les instances. Nous observons aussi que les temps de traitement sont généralement plus importants.

Comme on peut le voir dans la colonne *Sans Adaptif*, les temps de traitement sont plus importants. Pour que l'ALNS fournisse des résultats de bonne qualité en temps raisonnable, il faut un système adaptatif qui contrôle la sélection des opérateurs de destruction et de réparation.

	ALNS-SA		Sans LA		Sans LO		Sans LCW		Sans adaptatif	
	Min	t(s)	Min	t(s)	Min	t(s)	Min	t(s)	Min	t(s)
a.01	1110	0.35	1110.0	0.59	1110.0	0.57	1110.0	0.57	1110.0	0.35
a.02	1140	0.35	1140.0	0.69	1140.0	0.64	1140.0	0.59	1140.0	0.31
a.03	1160	0.7	1160.0	1.53	1160.0	1.61	1160.0	1.64	1160.0	0.58
a.04	1525	1.61	1525.0	5.37	1525.0	6.2	1525.0	9.23	1525.0	1.19
a.05	2765	0.82	2765.0	0.6	2765.0	0.93	2765.0	0.54	2765.0	10.63
a.06	1300	1.41	1300.0	3.92	1300.0	3.98	1300.0	3.88	1300.0	10.89
a.07	1805	0.65	1805.0	0.62	1805.0	0.57	1805.0	0.78	1805.0	20.48
a.08	2120	20.99	2120.0	26.44	2125.0	13.56	2120.0	18.43	2120.0	52.21
a.09	2750	4.61	2780.0	1.8	2790.0	2.09	2765.0	2.59	2750.0	25.2
a.10	1805	0.8	1805.0	1.03	1805.0	1.2	1805.0	1.04	1805.0	40.74
a.11	1850	8.41	1850.0	2.61	1850.0	4.31	1850.0	3.23	1850.0	71.25
a.12	1600	0.4	1600.0	0.72	1600.0	0.66	1600.0	0.62	1600.0	40.35
a.13	2090	1.44	2090.0	1.49	2090.0	1.91	2090.0	1.68	2090.0	10.94
a.14	2530	0.89	2530.0	0.74	2530.0	1.22	2530.0	0.73	2530.0	20.74
a.15	1940	1.9	1940.0	2.14	1940.0	2.59	1940.0	1.76	1940.0	51.43
a.16	4860	2.15	4860.0	1.44	4860.0	1.81	4860.0	2.37	4860.0	41.62
a.17	3205	3.88	3205.0	3.55	3205.0	3.07	3205.0	3.46	3205.0	22.75
a.18	4045	7.63	4065.0	2.9	4065.0	2.24	4065.0	3.28	4055.0	53.41
a.19	2835	1.38	2835.0	1.44	2835.0	1.75	2835.0	1.48	2835.0	121.14
a.20	7325	129.1	7325.0	1256.13	7325.0	710.26	7320.0	1039.48	7575.0	615.41
a.21	4085	1.3	4085.0	2.51	4085.0	2.39	4085.0	1.66	4085.0	50.99
a.22	3750	13.18	3765.0	10.25	3755.0	8.19	3745.0	11.05	3750.0	52.4
a.23	5955	9.35	5955.0	4.76	5960.0	5.41	5970.0	3.51	5955.0	42.94
a.24	4325	5.84	4350.0	2.55	4325.0	2.89	4325.0	3.49	4325.0	73.91
a.25	5030	115.08	5095.0	76.06	5050.0	101.49	5030.0	104.65	5030.0	53.89
a.26	3970	15.62	3970.0	6.59	3970.0	7.63	3970.0	8.14	3970.0	51.38
a.27	4310	2.22	4310.0	2.94	4310.0	1.96	4310.0	3.33	4310.0	81.54

Tableau 3.12: Impact des opérateurs de construction LA, LO, LCW et impact du mécanisme d'adaptation pour ALNS avec recuit simulé

3.7.6 Comparaison des deux versions de l'ALNS avec IDCH et ILP

Dans le Tableau 3.14, pour chaque instance la colonne *PRH* indique le résultat de l'heuristique constructive qui imite les règles des praticiens.

Sous la rubrique *ILP*, les colonnes *Obj*, *t(s)* et *g(%)* affichent les performances de l'ILP. Elles indiquent la valeur de la fonction objectif, le temps de calcul et l'écart obtenus par le solveur CPLEX.

Sous la rubrique *IDCH*, nous indiquons les valeurs *Min*, *Max* et *Avg* que nous avons obtenues au cours des 10 exécutions. La colonne *t(s)* indique les temps de traitement moyens en secondes. Nous utilisons $(\text{Min} - \text{Obj})/\text{Obj}$ indiqué dans les titres des colonnes comme *Ig(%)* pour comparer les résultats entre ILP et IDCH.

Sous la rubrique *ALNS-RE*, nous indiquons les valeurs *Min*, *Max* et *Avg* que nous avons obtenues au cours des 10 exécutions de ALNS avec *record-to-record* comme stratégie d'acceptation de solution non améliorante. La colonne *t(s)* indique les temps de traitement moyens en secondes. Nous utilisons $(\text{Min} - \text{Obj})/\text{Obj}$ indiqué dans les titres des colonnes comme *REg(%)* pour comparer les résultats entre ALNS *record-to-record* avec et IDCH. Sous la rubrique *ALNS-SA*, nous indiquons aussi les valeurs *Min*, *Max* et *Avg* que nous avons obtenues au cours des 10 exécutions de ALNS avec recuit simulé comme

	ALNS-RE		Sans LA		Sans LO		Sans LCW		Sans adaptatif	
	Min	t(s)	Min	t(s)	Min	t(s)	Min	t(s)	Min	t(s)
a_01	1110.0	0.3	1110.0	0.59	1110.0	0.57	1110.0	0.57	1110.0	0.35
a_02	1140.0	0.33	1140.0	0.69	1140.0	0.64	1140.0	0.59	1140.0	0.31
a_03	1160.0	0.73	1160.0	1.53	1160.0	1.61	1160.0	1.64	1160.0	0.58
a_04	1525.0	1.4	1525.0	5.37	1525.0	6.2	1525.0	9.23	1525.0	12.19
a_05	2765.0	0.38	2765.0	0.6	2765.0	0.93	2765.0	0.54	2765.0	10.63
a_06	1300.0	1.51	1300.0	3.92	1300.0	3.98	1300.0	3.88	1300.0	10.89
a_07	1805.0	0.45	1805.0	0.62	1805.0	0.57	1805.0	0.78	1805.0	20.48
a_08	2120.0	7.98	2120.0	26.44	2125.0	13.56	2120.0	18.43	2120.0	12.21
a_09	2750.0	1.08	2780.0	1.8	2790.0	2.09	2765.0	2.59	2750.0	20.2
a_10	1805.0	0.34	1805.0	1.03	1805.0	1.2	1805.0	1.04	1805.0	30.74
a_11	1850.0	1.43	1850.0	2.61	1850.0	4.31	1850.0	3.23	1850.0	31.25
a_12	1600.0	0.35	1600.0	0.72	1600.0	0.66	1600.0	0.62	1600.0	20.35
a_13	2090.0	0.79	2090.0	1.49	2090.0	1.91	2090.0	1.68	2090.0	40.94
a_14	2530.0	0.48	2530.0	0.74	2530.0	1.22	2530.0	0.73	2530.0	50.74
a_15	1940.0	0.95	1940.0	2.14	1940.0	2.59	1940.0	1.76	1940.0	81.43
a_16	4860.0	0.82	4860.0	1.44	4860.0	1.81	4860.0	2.37	4860.0	51.62
a_17	3205.0	1.58	3205.0	3.55	3205.0	3.07	3205.0	3.46	3205.0	62.75
a_18	4045.0	1.62	4065.0	2.9	4065.0	2.24	4065.0	3.28	4055.0	73.41
a_19	2835.0	0.81	2835.0	1.44	2835.0	1.75	2835.0	1.48	2835.0	41.14
a_20	7325.0	368.17	7325.0	1256.13	7325.0	710.26	7320.0	1039.48	7575.0	615.41
a_21	4085.0	1.01	4085.0	2.51	4085.0	2.39	4085.0	1.66	4085.0	20.99
a_22	3745.0	1.54	3765.0	10.25	3755.0	8.19	3745.0	11.05	3750.0	85.4
a_23	5955.0	2.17	5955.0	4.76	5960.0	5.41	5970.0	3.51	5955.0	34.94
a_24	4325.0	1.44	4350.0	2.55	4325.0	2.89	4325.0	3.49	4325.0	53.91
a_25	5030.0	42.06	5095.0	76.06	5050.0	101.49	5030.0	104.65	5030.0	53.89
a_26	3970.0	3.8	3970.0	6.59	3970.0	7.63	3970.0	8.14	3970.0	65.38
a_27	4310.0	1.24	4310.0	2.94	4310.0	1.96	4310.0	3.33	4310.0	41.54

Tableau 3.13: Impact des opérateurs de construction LA, LO, LCW et impact du mécanisme d'adaptation pour ALNS avec *record-to-record*

stratégie d'acceptation de solution non améliorante. La colonne $t(s)$ indique les temps de traitement moyens en secondes. Nous utilisons $(\text{Min} - \text{Obj})/\text{Obj}$ indiqué dans les titres des colonnes comme $S\text{Ag}(\%)$ pour comparer les résultats.

Dans le délai de 1800s, l'approche ILP trouve des solutions optimales pour 20 instances, trouve des solutions réalisables pour 3 instances et échoue à obtenir une solution pour 3 autres. Le temps de calcul de 1800 s est incompatible avec les besoins dans le cas dynamique.

L'approche IDCH a trouvé une solution réalisable pour chaque instance. Les deux versions de ALNS améliorent strictement et trouvent des meilleurs résultats en comparant avec IDCH et ILP. Elles sont aussi en général plus rapide que IDCH. Nous pouvons remarquer que ALNS avec recuit simulé comme stratégie d'acceptation trouve des solutions meilleurs que ALNS avec *record-to-record* pour 6 instances. Nous avons choisis d'utiliser ALNS avec recuit simulé comme phase d'amélioration dans l'algorithme dynamique.

PRH	ILP				IDCH					ALNS-RE				ALNS-SA					
	Obj	t(s)	g(%)	Min	Max	Avg	Ig(%)	t(s)	Min	Max	Avg	REIg(%)	t(s)	Min	Max	Avg	SAIg(%)	t(s)	
a.01*	1210	1110	4	0.00	1110	1110	1110	0.00	0.07	1110	1110	1110	0.00	0.3	1110	1110	1110	0.00	0.35
a.02*	1195	1140	5	0.00	1140	1140	1140	0.00	0.08	1140	1140	1140	0.00	0.33	1140	1140	1140	0.00	0.35
a.03*	1365	1160	17	0.00	1160	1160	1160	0.00	0.15	1160	1160	1160	0.00	0.73	1160	1160	1160	0.00	0.7
a.04*	1630	1525	106	0.00	1525	1560	1544	0.00	25.95	1525	1545	1527	0.00	1.4	1525	1525	1525	0.00	1.61
a.05*	4660	2765	6	0.00	2765	2945	2845	0.00	0.16	2765	2945	2793.5	0.00	0.38	2765	2945	2873	0.00	0.82
a.06*	1500	1300	11	0.00	1300	1300	1300	0.00	0.27	1300	1300	1300	0.00	1.51	1300	1300	1300	0.00	1.41
a.07*	2175	1805	17	0.00	1805	1805	1805	0.00	0.12	1805	1805	1805	0.00	0.45	1805	1805	1805	0.00	0.65
a.08	2505	2615	1800	27.18	2175	2430	2291	-16.83	18.57	2165	2285	2231	-17.21	7.98	2120	2130	2121	-18.93	20.99
a.09	4370	2750	343	0.00	2845	3685	3290	3.45	2.31	2775	3725	3048	0.91	1.08	2750	3205	2849.5	0.00	4.61
a.10*	1925	1805	33	0.00	1805	1920	1815	0.00	57.81	1805	1805	1805	0.00	0.34	1805	1805	1805	0.00	0.8
a.11*	1955	1850	16.41	0.00	1850	1850	1850	0.00	0.28	1850	1850	1850	0.00	1.43	1850	1850	1850	0.00	8.41
a.12*	2020	1600	7	0.00	1600	1600	1600	0.00	0.07	1600	1600	1600	0.00	0.35	1600	1600	1600	0.00	0.4
a.13*	2300	2090	51	0.00	2090	2090	2090	0.00	0.25	2090	2090	2090	0.00	0.79	2090	2090	2090	0.00	1.44
a.14*	2755	2530	44	0.00	2530	2740	2570	0.00	21.57	2530	2640	2541.5	0.00	0.48	2530	2640	2541	0.00	0.89
a.15*	2380	1940	183.38	0.00	1940	1950	1945	0.00	0.51	1940	1950	1944.5	0.00	0.95	1940	1950	1948	0.00	1.9
a.16	6500	4395	288	0.00	4860	5680	5235	10.58	17.27	4860	5445	5037	10.58	0.82	4860	4860	4860	10.58	2.15
a.17*	4100	3205	526.08	0.00	3205	3205	3205	0.00	0.81	3205	3205	3205	0.00	1.58	3205	3205	3205	0.00	3.88
a.18	4880	4045	973	0.00	4160	4880	4361	2.84	42.47	4045	4300	4118	0.00	1.62	4045	4045	4045	0.00	7.63
a.19*	3625	2835	379.80	0.00	2835	2835	2835	0.00	0.36	2835	2835	2835	0.00	0.81	2835	2835	2835	0.00	1.38
a.20	20560	-	1800	-	7715	8030	7855	-	23.59	7370	8035	7685	-	368.17	7325	7570	7460	-	129.1
a.21*	4385	4085	184.22	0.00	4085	4085	4085	0.00	0.28	4085	4085	4085	0.00	1.01	4085	4085	4085	0.00	1.3
a.22	6260	-	1800	-	3800	4080	3911	-	2.49	3745	3910	3827.5	-	1.54	3750	3820	3768	-	13.18
a.23	7275	7360	1800	1915.76	6020	6350	6170	-17.25	1.79	5960	6115	6017.5	-19.02	2.17	5955	6105	5991.5	-19.09	9.35
a.24	7190	5190	1800	1666.67	4335	4595	4485	-16.47	1.37	4325	4595	4373.5	-16.67	1.44	4325	4590	4360	-16.67	5.84
a.25	12960	-	1800	-	5095	5215	5170	-	22.81	5040	5165	5108.5	-	42.06	5030	5110	5063	-	115.08
a.26	5450	-	1800	-	3990	4050	4030	-	2.61	3970	4010	3980	-	3.8	3970	3990	3975.5	-	15.62
a.27*	5405	4310	125.95	0.00	4310	4515	4380	0.00	0.41	4310	4315	4310.5	0.00	1.24	4310	4310	4310	0.00	2.22

Tableau 3.14: Comparaison des deux version de l'ALNS avec IDCH et ILP

3.7.7 Algorithme de planification dynamique

L'algorithme de planification dynamique proposé est évalué sur un ensemble d'instances générées (voir Tableau 3.7). Une première série de tests vise à évaluer la qualité de la solution proposée par rapport à la limite supérieure calculée par ALNS dans le cas statique. Le second est une étude de temps moyen pour les mises à jour dynamiques du planning. La mise à jour dynamique du planning reflète le temps que l'algorithme passe dans la phase d'amélioration méta-heuristique.

Dans le Tableau 3.15, pour chaque instance sous la rubrique *ALNS-Statique*, nous indiquons les résultats obtenus par ALNS dans le cas statique. La colonne *Obj* indique la valeur de la fonction objectif et les colonnes *t(s)* indique le temps de calcul en secondes. Sous la rubrique *Dyn*, nous indiquons la valeur de la fonction objectif *Obj* de la solution trouvée par l'approche dynamique mais sans la phase d'amélioration par ALNS. À des fins de comparaison, nous utilisons les valeurs $(Obj(Dyn) - Obj(ALNS-Statique)) / Obj(ALNS-Statique)$ désignées dans l'en-tête de colonne par *gapDS(%)*.

Sous la rubrique *ALNS-Dynamique*, nous indiquons la valeur de la fonction objectif *Obj* de la solution trouvée par l'approche dynamique avec la phase d'amélioration par ALNS-SA. Nous utilisons les valeurs $(Obj(ALNS-Dynamique) -$

	ALNS-Statique		Dyn		ALNS-Dynamique	
	Obj	t(s)	Obj	gapDS(%)	Obj	gapDS(%)
D1	7060	21,39	7560	7,08	7060	0,00
D2	6075	129,16	6660	9,63	6075	0,00
D3	6300	70,92	6390	1,27	6310	0,16
D4	5045	32,21	5115	1,39	5045	0,00
D5	5300	34,3	5410	2,08	5300	0,00
D6	9060	206,211	19440	114,57	10465	15,51
D7	8050	121	11195	39,07	8050	0,00
D8	11125	433,47	19880	78,70	11640	4,63
D9	8900	118,28	9900	11,24	9050	1,69
D10	9195	155,75	10375	12,83	9665	5,11
D11	12725	234,86	15005	17,92	13990	9,94
D12	14560	239,12	18500	27,06	14880	2,20
D13	12660	329,49	15680	23,85	13405	5,88
D14	14250	214,87	18710	31,30	15015	5,37
D15	13555	231	18750	38,33	15120	11,55
D16	24105	4734,13	33225	37,83	25405	5,39
D17	44030	368,08	49755	13,00	46865	6,44
D18	17460	452,87	44240	153,38	17985	3,01
D19	26405	514,27	56362	113,45	28705	8,71
D20	33100	578,14	63412	91,58	36660	10,76
D21	23755	652,85	29840	25,62	26500	11,56
D22	78235	624,12	103658	32,50	85420	9,18
D23	72105	657,14	80485	11,62	78500	8,87
D24	319855	678,54	434035	35,70	368845	15,32

Tableau 3.15: Comparaison entre les résultats de l’algorithme dynamique et l’approche statique avec et sans la phase d’amélioration

$\text{Obj}(\text{ALNS-Statique})/\text{Obj}(\text{ALNS-Statique})$ désignées dans l’en-tête de colonne par $\text{gapDS}(\%)$.

La comparaison entre Dyn et ALNS-SA dans le cas statique montre un écart important qui peut arriver à 153%. Ce résultat est prévisible, puisque l’algorithme Dyn n’utilise qu’une méthode de construction pour construire les plannings dynamiques et ne fait appel à aucune phase d’amélioration méta-heuristique. L’algorithme ALNS-Dynamique utilise ALNS avec recuit simulé comme stratégie d’acceptation dans la phase d’amélioration et présente un écart beaucoup plus faible. L’écart est 0.0% pour 5 instances.

Le temps nécessaire à la mise à jour d’un planning dynamique représente le laps de temps vécu par un utilisateur humain dans le cadre d’une mise en œuvre en temps réel. La mise à jour du planning dynamique reflète le temps que l’algorithme passe dans la phase d’amélioration de la méta-heuristique. La Figure 3.23 montre que les mises à jour dynamiques prennent moins de 120s dans la plupart des instances, ce qui est raisonnable en vue d’une mise en œuvre en situation réelle au sein d’un SUA. Il existe deux instances dans lesquels les mises à jour dynamiques prennent plus de 120s. Ces instances présentent un taux élevé d’arrivée de patients.

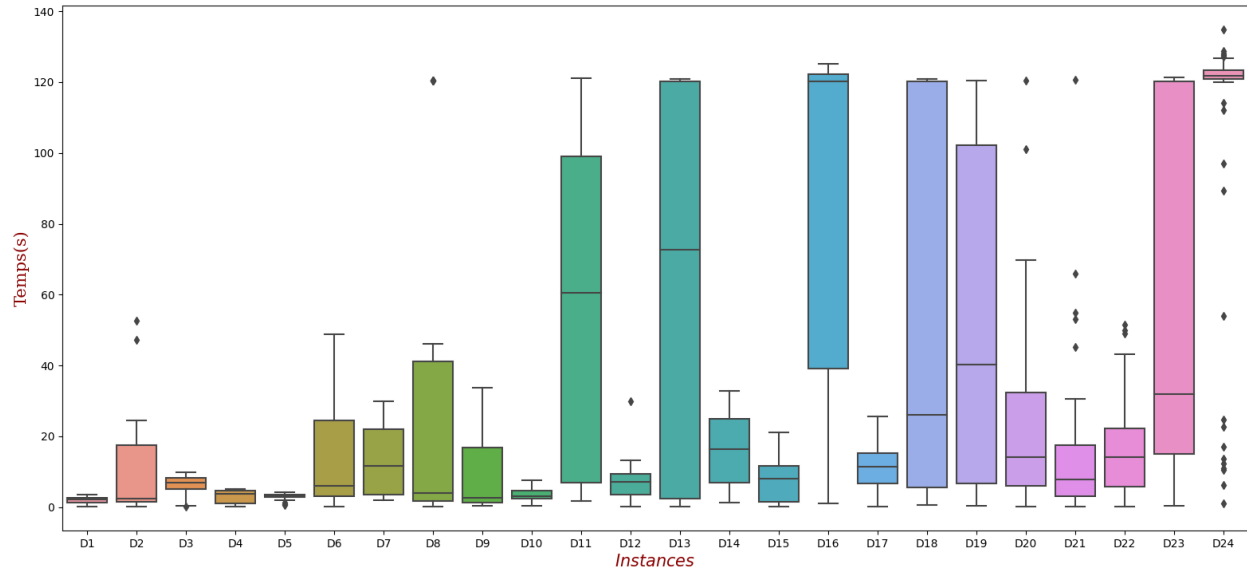


Fig 3.23: Les temps de mise à jour de planning dans le cas dynamique

3.8 CONCLUSION ET PERSPECTIVES

Dans ce chapitre, nous avons étudié le problème de l'ordonnancement des tâches de soins dans le service des urgences du CHU Jeanne de Flandres de Lille. Différents types de tâches et différents types de ressources ont été pris en compte dans notre étude. Nous avons présenté une description générale de structure de service. Nous avons aussi présenté le problème étudié en détaillant les différentes contraintes et caractéristiques.

Dans le cas statique du problème, nous avons implémenté l'approche *Practitioners' Rules Heuristic* (PRH) en intégrant deux règles métiers, elles construisent de premières solutions réalisables en respectant les contraintes de ressources.

Nous avons proposé un modèle mathématique basé sur la Programmation Linéaire en Nombres Entiers (PLNE) pour le circuit court. La résolution exacte du PLNE nous a permis d'avoir des solutions de référence optimales sur de petites instances et des solutions réalisables sur des instances plus importantes. Ce qui nous a permis d'obtenir de comparaison pour les approches heuristiques.

Nous avons élaboré une approche heuristique itérative basée sur la destruction et la construction de solutions (IDCH). Dans des temps raisonnables, les résultats indiquent que des solutions de meilleure qualité que l'heuristique constructive PRH peuvent être produites. Des solutions optimales pour certaines instances sont obtenues.

Nous avons ensuite développé une méta-heuristique plus efficace *Adaptive Large Neighborhood Search* (ALNS). Nous avons proposé plusieurs voisinages de destruction et de construction et expérimenté deux processus adaptatifs d'acceptation de solutions non améliorantes (probabiliste et déterministe). Les expériences pour le réglage des paramètres de l'approche ALNS ont été menées pour obtenir un bon compromis entre la qualité de la solution et le temps de traitement. L'évaluation de l'efficacité de chaque composant de l'approche ALNS montre que tous les composants sont nécessaires pour obtenir de bonnes solutions dans de bons délais de traitement. Les résultats

expérimentaux montrent que cette méthode est efficace, avec des temps de calcul courts qui répondent au besoin de réactivité nécessaire dans le cas du problème dynamique.

Le cas dynamique du problème a également été étudié. Lorsque les patients sont arrivés, il est possible de déterminer leurs besoins spécifiques et les alternatives de traitement les plus appropriées. Nous avons proposé une approche prédictive et réactive pour la planification sur la base des approches proposées pour le cas statique. La technique proposée consiste à générer un planning sans tenir compte de la possibilité de perturbations dynamiques, puis à modifier le planning en fonction des événements qui se produisent. En utilisant les méthodes de résolution du cas statique, nous concevons, modifions et optimisons le planning dynamiquement.

Dans le cas statique toutes les informations sont stables et connues. Dans le cas dynamique les durées des tâches de soin réalisées peuvent changer et l'arrivée des patients induit une replanification. Sur la base d'un jeu de données plus conséquent, nous avons calculé la solution en considérant toutes les informations connues pour avoir des valeurs de références. Nous avons ensuite considéré l'aspect dynamique sur la base de scénarios pour lesquels les durées des tâches réalisées changent et en considérant l'arrivée des patients.

Les résultats expérimentaux montrent que l'approche de résolution dynamique proposée obtient de bons résultats dans des temps de calcul compatible avec les contraintes de réactivité d'un SUA.

PROBLÈME DE PLANIFICATION DES INTERVENTION CHIRURGICALES

SOMMAIRE

4.1	Introduction	87
4.2	Définition du problème et modélisation mathématique	88
4.3	Approches de résolutions	94
4.3.1	Heuristique 2PSC-EM	94
4.3.2	Métaheuristique ALNS	96
4.4	Résultats	106
4.4.1	Benchmark et protocole expérimental	106
4.4.2	Comparaison de l’heuristique 2PSC-EM avec la littérature	107
4.4.3	Paramétrage de ALNS	108
4.4.4	Évaluation des composantes de l’ALNS	112
4.4.5	Comparaison de l’ALNS avec la littérature	114
4.5	Conclusion et perspectives	115

4.1 INTRODUCTION

Les systèmes de soins de santé sont confrontés à des demandes croissantes dans un contexte de contraintes budgétaires difficiles. L'accès à des services de santé de qualité est devenu un problème majeur. (Xie et al., 2015). Le service de chirurgie représente une activité essentielle pour la majorité des hôpitaux, générant environ deux tiers des revenus de l'hôpital. (Denton et al., 2007). Les interventions chirurgicales représentent 52% de tous les motifs d'entrées à l'hôpital selon (Gupta, 2009). Compte tenu des contraintes budgétaires, l'augmentation du nombre de chirurgiens et de salles d'opération, en plus de toutes les autres ressources nécessaire, n'est pas la solution à rechercher en premier lieu, mais plutôt l'amélioration de l'utilisation des ressources existantes lorsque cela est possible (Fischer et al., 2020). L'un des moyens d'y parvenir est de construire de meilleurs programmes de chirurgie dans le but de planifier un nombre maximal

d'interventions chirurgicales tout en tenant compte des ressources limitées et des contraintes de disponibilité.

Dans ce chapitre, nous abordons le problème de la planification dit des admissions pour lequel les interventions chirurgicales de centaines de patients électifs doivent être programmées plusieurs mois avant la date de l'opération. Nous proposons une heuristique aléatoire 2PSC-EM à deux phases basée sur les idées des heuristiques constructives de RCPSP. Nous étudions une approche de solution basée sur l'ALNS dans le but de planifier toutes les interventions des patients.

Le chapitre est organisé comme suit.

La section 4.2 présente le problème d'admission que nous abordons. Les contraintes, les paramètres et la fonction objectif relatifs aux patients sont détaillés. La section 4.3.1 présente l'heuristique à deux phases 2PSC-EM que nous proposons. Notre approche ALNS est décrite dans la section 4.3.2 et plusieurs composants sont détaillés. Les expériences numériques sont rapportées et commentées dans la section 4.4. La conclusion et les perspectives se trouvent dans la section 4.5.

4.2 DÉFINITION DU PROBLÈME ET MODÉLISATION MATHÉMATIQUE

Les hôpitaux doivent créer les plannings d'admission pour les interventions chirurgicales des patients des mois avant la date de leur intervention en tenant compte de leurs ressources les plus critiques : les chirurgiens et les salles d'opération.

Dans [Riise et al., 2016], les auteurs ont présenté le modèle GOSSP (problème d'ordonnancement opérationnel généralisé de la chirurgie) comme une extension du MR-CPSP/Max. multi-projets, une généralisation du problème classique d'ordonnancement de projet sous contrainte (RCPSP), connu pour être NP-difficile au sens fort. Le modèle proposé par les auteurs permet de considérer des paramètres relatifs aux patients (niveau de priorité, date de disponibilité, date d'échéance, date d'échéance ferme) et aux disponibilités des ressources, ainsi qu'à des préférences visant à programmer certaines interventions tôt le matin. Un chirurgien peut préférer par exemple d'effectuer des interventions chirurgicales complexes et longues tôt le matin. Pour certains types de patients beaucoup plus à risque ou pour les enfants la chirurgie doit être programmée le matin. Trois situations de planification ont été modélisées : la planification de l'admission, le problème hebdomadaire et le problème quotidien. À mesure que le jour de l'opération approche, davantage de ressources doivent être gérées et le modèle est capable d'intégrer les informations liées à la situation de planification. Les auteurs ont proposé des ensembles de données accessibles au public qui peuvent être utilisés pour comparer les méthodes de recherche. Dans un souci de généralité, les auteurs ont proposé un algorithme de construction et d'amélioration adaptative (ACI) pour traiter toutes ces situations de planification sans aucun réglage hors ligne des paramètres. À notre connaissance, aucun autre résultat n'a été publié sur ces jeux de données. Pour un aperçu plus détaillé, nous invitons le lecteur à se référer à [Riise et al., 2016].

Nous nous concentrons sur le problème de planification dit des admissions suivant : planifier toutes les interventions pour un ensemble de patients afin d'obtenir une date de début (un jour et une heure) pour chaque intervention sur un horizon de plusieurs

mois. Il se peut que certaines interventions ne puissent pas être planifiées, mais il faut éviter cela autant que possible. Nous considérons les patients électifs, hospitalisés ou non, pour lesquels une intervention chirurgicale doit être planifiée sans tenir compte des événements qui peuvent survenir dans des horizons de planifications à plus court terme.

L'organisation du service de chirurgie que nous considérons ici est basée sur une gestion partagée des salles d'opération et des chirurgiens qui sont considérés comme des ressources renouvelables disponibles dans certains intervalles de temps, et, où chaque ressource a une capacité limitée, constante. Une intervention peut être réalisée selon plusieurs modes puisque plusieurs couples de salles d'opération et de chirurgiens peuvent être utilisés. Un mode a une durée fixe qui peut dépendre, par exemple, de l'acte chirurgical spécifique au chirurgien et à la salle d'opération.

Les considérations médicales et pratiques induisent plusieurs types de contraintes temporelles. Chaque patient a une date de début au plus tôt pour l'intervention, une date d'échéance définie pour refléter une priorité médicale et une date limite pour l'intervention. La date de début de l'intervention chirurgicale du patient p est strictement inférieure à la date limite.

La disponibilité de chaque ressource est donnée par des intervalles de temps. Ils correspondent aux disponibilités des chirurgiens ou aux heures d'ouverture des blocs opératoires dans les jours et semaines de l'horizon de planification.

Nous présentons d'abord les données et paramètres relatifs aux patients (niveau de priorité, date de disponibilité, date d'échéance, date limite), les disponibilités des ressources, ainsi que les préférences visant à planifier certaines interventions tôt le matin. Ensuite, nous présentons la formulation mathématique du problème à partir de [Riise et al., 2012] et [Riise et al., 2016]. Dans ce qui suit, nous utilisons la plupart des notations introduites dans les documents [Riise et al., 2012] et [Riise et al., 2016].

Données et paramètres

Les patients, ainsi que certaines questions médicales et pratiques connexes, sont examinés à l'aide des données et paramètres suivants:

H , l'horizon de planification, ou la durée de la période de planification ;

z , le nombre d'unités de temps dans une journée ;

P , ensemble de patients électifs pour lesquels une intervention chirurgicale doit être planifiée, indices p , nous désignons par $|P|$ sa taille ;

X_p , date de début au plus tôt pour l'intervention du patient p ;

D_p , date d'échéance ; le patient p doit de préférence être programmé avant ;

H_p , date limite ; le patient p doit être programmé avant ; sinon, l'intervention n'est pas programmée ;

β_p , pour un patient, l'importance relative de programmer l'intervention chirurgicale du patient ;

$P^{em} \subset P$, sous-ensemble de patients pour lesquels l'intervention doit être planifiée tôt dans la matinée (*early in the morning*), on désigne par $|P^{em}|$ sa taille ;

τ , pour tous les patients $p \in P^{em}$, l'heure du matin à partir de laquelle l'intervention peut être planifiée ;

γ_p , pour un patient $p \in P^{em}$, l'importance relative de planifier le patient tôt le matin ;

R_s , ensemble des chirurgiens, indices s ;

R_r , ensemble des salles d'opération, indices r ;

R , ensemble de toutes les ressources, $R = R_s \cup R_r$;

R^p , ensemble des ressources affectées aux interventions chirurgicales du patient p , par le choix du mode ;

K_i , ensemble des intervalles de disponibilité successifs et disjoints associés à la ressource i pour (chirurgien ou salle) ;

$K_i^p \subseteq K_i$, ensemble des intervalles pour lesquels la ressource i est disponible pour les interventions chirurgicales du patient p ;

M , ensemble des modes $m = (r_s, r_r) \in R_s \times R_r$, r_s indice d'un chirurgien et r_r indice d'une salle ; ce sont toutes les combinaisons de chirurgiens et de salles d'opération ;

M_p , ensemble des modes réalisables m pour l'intervention chirurgicale du patient p ;

φ^k , heure de début fixe de l'intervalle de ressources k ;

σ^k , heure de fin au plus tard de l'intervalle de ressources k ;

u_i^m , si le mode m utilise la ressource i alors $u_i^m = 1$;

ϑ_p^m , durée de l'intervention chirurgicale pour le patient p avec l'utilisation du mode m .

Chaque patient p a une date de début au plus tôt X_p pour l'intervention et une date d'échéance D_p définie pour refléter une priorité médicale. Chaque patient p a une date limite $H_p \geq D_p$. La date de début de l'intervention doit être strictement inférieure à cette date d'échéance, sinon l'intervention chirurgicale n'est pas planifiée.

Les paramètres X_p , D_p et H_p sont des nombres entiers de jours par rapport au début de l'horizon de planification. Pour un patient p , la valeur β_p est liée au degré de priorité du patient.

Pour certains patients particuliers $p \in P^{em}$, l'intervention chirurgicale devrait être planifiée de préférence tôt dans la matinée ; le paramètre τ est l'heure du matin à partir de laquelle l'intervention chirurgicale peut être planifiée. Le paramètre γ_p est l'importance relative de programmer l'intervention chirurgicale du patient plus tôt le matin.

Pour être planifiée, une intervention chirurgicale nécessite un chirurgien et une salle d'opération. Les ensembles R_s , R_r , R et R^p , sont utilisés pour gérer les ressources. Les ensembles K_i sont utiles pour prendre en compte le temps de travail quotidien des équipes ou des chirurgiens en considération, ainsi que les durées de fonctionnement des salles d'opération et les tâches de maintenance de ces dernières.

Pour un patient p , l'ensemble des modes réalisables est M_p , en fonction de l'intervention chirurgicale à réaliser. Les évaluations médicales effectuées lors des consultations préopératoires permettent de sélectionner les chirurgiens non seulement en fonction de la spécialité requise, mais aussi en fonction de l'expérience professionnelle liée à la difficulté du cas clinique.

Habituellement, dans une telle organisation, ce sont les chirurgiens eux-mêmes, organisés par spécialité, qui décident qui peut effectuer l'opération chirurgicale. Par conséquent, la durée ϑ_p^m de l'opération dépend du chirurgien et de la salle d'opération où elle peut être réalisée.

Un mode m doit être choisi pour planifier l'intervention chirurgicale du patient p parmi M_p . La préemption des interventions chirurgicales n'est pas autorisée. Sachant que l'on planifie de nombreux patients, lorsqu'on utilise le mode $m = (r_s, r_r)$ pour un patient p , un intervalle J parmi l'ensemble des intervalles $K_{r_s} \cap K_{r_r}$ est à choisir pour permettre de planifier l'intervention chirurgicale de longueur ϑ_p^m , en tenant compte de toutes les autres interventions chirurgicales déjà programmées.

Pour un patient p , une intervention chirurgicale doit être planifiée, une intervention correspond à un patient p . Pour des raisons de simplicité, nous utilisons les mêmes indices p pour indexer les interventions dans la suite puisqu'il n'y a qu'une intervention par patient.

Formulation mathématique

Nous introduisons la formulation mathématique du problème à partir de [Riise et al., 2012] et [Riise et al., 2016].

Les variables sont les suivantes :

$x_p^m = 1$, si l'intervention chirurgicale du patient p utilise le mode $m \in M_p$, 0 sinon ;

$q_p^k = 1$, si l'intervention chirurgicale du patient p utilise l'intervalle de ressource $k \in K_i$, pour la ressource i , 0 sinon ;

t_p , l'heure de début non négative d'intervention chirurgicale p ;

$z_{pp'} = 1$, si l'intervention chirurgicale du patient p précède l'intervention chirurgicale du patient p' , 0 sinon ;

c_p , temps de réalisation d'interventions chirurgicale du patient p ;

$g_p^i = 1$, si l'intervention chirurgicale du patient p utilise la ressource i dans le mode choisi, 0 sinon ;

$h^p = 1$, si l'intervention chirurgicale du patient p n'est pas planifiée, 0 sinon.

La formulation MIP est la suivante :

Min

$$W_{un} \underbrace{\frac{1}{\sum_{p \in P} \beta_p} \sum_{p \in P} \beta_p h^p}_{O_{un}} \quad (4.1a)$$

$$+ W_{wt} \underbrace{\frac{1}{1 + \lambda|P|} \sum_{p \in P} o_{wt(p)}(1 - h^p)}_{O_{wt}} \quad (4.1b)$$

$$+ W_{em} \underbrace{\frac{1}{z|P^{em}|} \sum_{p \in P^{em}} \gamma_p \Gamma(p)(1 - h^p)}_{O_{em}} \quad (4.1c)$$

Soumis à :

$$x_p^m \in \{0, 1\} \quad \forall p \in P, \forall m \in M_p \quad (4.2)$$

$$z_{pp'} \in \{0, 1\} \quad \forall p, p' \in P \quad (4.3)$$

$$t_p \in \mathbb{R}_+ \quad \forall p \in P \quad (4.4)$$

$$q_p^k \in \{0, 1\} \quad \forall i \in R, \forall k \in K_i^p \quad (4.5)$$

$$c_p \in \mathbb{R}_+ \quad \forall p \in P \quad (4.6)$$

$$g_p^i \in \{0, 1\} \quad \forall i \in R, \forall p \in P \quad (4.7)$$

$$c_p = t_p + \sum_{m \in M_p} \vartheta_p^m x_p^m \quad \forall p \in P \quad (4.8)$$

$$g_p^i = \sum_{m \in M_p} u_i^m x_p^m \quad \forall i \in R, \forall p \in P \quad (4.9)$$

$$\sum_{m \in M_p} x_p^m = 1 - h^p \quad \forall p \in P \quad (4.10)$$

$$\sum_{k \in K_i^p} q_p^k = g_p^i \quad \forall p \in P, \forall i \in R \quad (4.11)$$

$$z_{pp'} + z_{p'p} \leq 1 \quad \forall p, p' \in P \quad (4.12)$$

$$t_{p'} - c_p \geq (z_{pp'} - 1)\mathcal{M} \quad \forall p, p' \in P \quad (4.13)$$

$$z_{pp'} + z_{p'p} \geq g_p^i + g_{p'}^i - 1 \quad \forall p, p' \in P, \forall i \in R \quad (4.14)$$

$$t_p \geq X_p \quad \forall p \in P \quad (4.15)$$

$$c_p \leq H_p \quad \forall p \in P \quad (4.16)$$

$$t_p - \varphi^k q_p^k \geq 0 \quad \forall p \in P, \forall i \in R, \forall k \in K_i^p \quad (4.17)$$

$$c_p - \sigma^k q_p^k - (1 - q_p^k)\mathcal{M} \leq 0 \quad \forall p \in P, \forall i \in R, \forall k \in K_i^p \quad (4.18)$$

Les contraintes (4.2)-(4.7) définissent les variables de decision. Les contraintes (4.8) définissent les temps de réalisation c_p . Contraintes (4.9) garantissent que la ressource i est utilisée dans le mode choisi m pour le patient p . Contraintes (4.10) s'assurer qu'au plus un mode m est choisi et définir h^p en conséquence. Les contraintes (4.11) garantissent qu'un intervalle k est utilisé pour chaque ressource i du mode m pour un patient p . Les contraintes (4.12), (4.13) et (4.14) assurent les contraintes de précédence lorsque deux patients p, p' utilisent une même ressource i .

Contraintes (4.15) et (4.16) assurent que l'intervention chirurgicale soit dans l'intervalle requis $[X_p, H_p]$ pour le patient p . Les contraintes (4.17) et (4.18) garantissent que l'intervalle d'intervention chirurgicale $[t_p, c_p]$ se situe dans l'intervalle $[\varphi^k, \sigma^k]$ de la ressource i . Dans les contraintes (4.13) et (4.18), la valeur du grand M est $\mathcal{M} = H$ (voir [Riise et al., 2012]).

Une solution réalisable S est composée de $\tilde{P} \subseteq P$, un ensemble de patients électifs pour lesquels une intervention chirurgicale est planifiée ($h^p = 0$) et un ensemble $P \setminus \tilde{P}$ de patients pour lesquels une intervention chirurgicale n'est pas planifiée ($h^p = 1$). De la même manière, nous considérons \tilde{P}^{em} et $P^{em} \setminus \tilde{P}^{em}$.

La fonction objectif à minimiser est une somme pondérée de trois termes normalisés, voir (4.1a)-(4.1c).

Étant donné une solution réalisable, le premier terme O_{un} de la fonction objectif vise à minimiser le nombre de patients électifs pour lesquels les interventions chirurgicales sont laissées non planifiées. Le terme est une somme normalisée des valeurs β_p , l'importance relative de programmer l'intervention chirurgicale du patient, pour tous les patients électifs pour lesquels leur intervention chirurgicale ($p \in P \setminus \tilde{P}$) ne peut être programmé. Le deuxième terme de la fonction objectif O_{wt} est une somme normalisée des temps d'attente dont les interventions chirurgicales sont planifiées. Pour un patient p , Le temps d'attente $o_{wt(p)}$ est évaluée en utilisant la fonction suivante :

$$o_{wt(p)} = \begin{cases} \frac{c_p - X_p}{D_p - X_p} & \text{si } c_p \leq D_p \\ 1 + \lambda \frac{c_p - D_p}{H_p - D_p} & \text{sinon.} \end{cases} \quad (4.19)$$

Globalement, pour un patient p , cette fonction linéaire par morceaux augmente lorsque c_p augmente par rapport à la date de début au plus tôt X_p et par rapport à la date d'échéance D_p . Étant donné que nous devons programmer l'intervention avant la date d'échéance D_p , la valeur de la pente devient plus importante λ lorsque $c_p > D_p$. Le but est de sur-pénaliser l'intervention chirurgicale d'un patient p qui est planifiée après la date d'échéance D_p .

Le troisième terme O_{em} est une somme normalisée qui évalue les interventions chirurgicales des patients $p \in P^{em}$ dont les interventions doivent être planifiées tôt dans la matinée. Étant donné que τ est connu pour tous les patients $p \in P^{em}$, la fonction $\Gamma(p)$ calcule la différence de temps entre τ et c_p pour le jour de planification choisi, cette valeur est pondérée par l'importance relative γ_p par patient.

Les termes normalisés sont pondérés par W_{un} , W_{wt} et W_{em} . Nous avons $W_{un} > W_{wt} > W_{em}$, ce qui reflète bien la hiérarchie des considérations médicales : la priorité est d'abord de planifier le plus grand nombre possible d'interventions des patients, ensuite de planifier au plus tôt possible, et enfin de planifier le matin les patients qui ont cette contrainte à respecter.

Nous désignons par $Obj(S)$ l'évaluation d'une solution S que nous calculons comme présenté dans les équations (4.1a)-(4.1c).

4.3 APPROCHES DE RÉOLUTIONS

4.3.1 Heuristique 2PSC-EM

Étant donné que les interventions chirurgicales de l'ensemble P de patients peuvent être considérées comme des projets qui nécessitent des ressources renouvelables et partagées, le problème de planification des admissions est présenté dans [Riise et al., 2016] comme un RCPSM multi-projets/multi-modes avec des décalages temporels minimum et maximum.

Nous proposons une phase de construction avec deux listes deux priorités (2PSC) basée sur les idées des heuristiques constructives de RCPSM (voir [Kolisch et al., 1999; Tormos et al., 2003; Almeida et al., 2016]), suivie d'une phase d'amélioration qui se concentre sur les interventions qui doivent être planifiées tôt dans la matinée. Nous désignons cette heuristique par le terme 2PSC-EM.

La phase de construction 2PSC utilise deux listes de priorités, l'une pour gérer les patients à planifier (P), et la seconde pour gérer les modes d'exécution de chaque patient (M_p). L'idée générale est d'augmenter le temps à chaque itération afin de choisir la prochaine intervention chirurgicale à planifier en prenant en compte toutes les interventions chirurgicales en cours qui sont déjà planifiées (celles-ci sont supposées être toujours en cours) puisqu'elles ne peuvent pas être interrompues (pas de préemption autorisée).

L'algorithme de la phase de construction du 2PSC est présenté dans l'Algorithme 5. L'itération i correspond à l'instant t_i où une intervention chirurgicale peut être planifiée ; elles sont initialisées à $i = 0$ et $t_0 = 0$. Le début de l'horizon de planification est 0 et la fin est t_{end} . Nous désignons par E_p le temps de fin d'intervention p lorsque l'intervention chirurgicale est planifiée pour un patient p de longueur ϑ_p^m en utilisant

Algorithme 5 : La phase de construction 2PSC

Entrées : P ensemble des interventions (et des patients), R_s, R_r ensembles de ressources (chirurgien, salle)
Sorties : S solution, planning calculé, \mathcal{L}_u liste des interventions non planifiées
Variables : t_i le temps où une intervention chirurgicale peut être planifiée; i nombre d'itérations, par rapport au temps t_i ; C_i ensemble d'interventions, réalisées au moment t_i ; A_i ensemble d'interventions, en cours au moment t_i ; R_i ensemble de ressources utilisées par les interventions dans A_i ; \mathcal{L}_i liste des interventions qui peuvent être planifiées au moment t_i ; p nombre d'interventions; $M_{p,i}$ liste des modes pouvant être utilisés au moment t_i pour l'intervention p ; m un mode (r_s, r_r) ; E_p heure de fin de l'intervention p

```

1   $i, t_i \leftarrow 0$ 
2   $S, A_0, C_0 \leftarrow \emptyset$ 
3   $R_0 \leftarrow R_s \cup R_r$ 
4  tant que  $|A_i \cup C_i| < |P| \wedge$  (Les interventions peuvent être planifiées sans dépasser l'horizon) faire
    // e temps de l'intervention en cours avec la fin la plus tôt prévue
5     $t_i \leftarrow \min_{p \in A_i} \{E_p\}$ 
6    Calculer en utilisant le nombre de pas de temps  $i$ :  $C_i, A_i, R_i, \mathcal{L}_i$ 
7    Trier  $\mathcal{L}_i$  dans l'ordre croissant de  $D_p$  // La date d'échéance la plus tôt
8    Mélange de  $\mathcal{L}_i$  par bloc de valeurs égales de  $D_p$ 
9    tant que  $\mathcal{L}_i \neq \text{nil}$  faire
10      $p \leftarrow \mathcal{L}_i.\text{pop\_front}()$  // La première intervention qui peut être planifiée au moment  $t_i$ 
        // Une intervention ayant l'une des plus petites valeurs de  $D_p$  est sélectionnée
11     Calculer  $M_{p,i}$  // Les modes utilisables  $m$  à un moment  $t_i$  pour l'intervention  $p$ 
12     Trier  $M_{p,i}$  dans l'ordre croissant de  $\vartheta_p^m$  // la petite durée d'abord
13      $m \leftarrow M_{p,i}.\text{pop\_front}()$  // mode avec la plus petite durée  $\vartheta_p^m$ 
14     Planifier l'intervention  $p$  à  $t_i$  avec le mode  $m$  dans la solution  $S$ 
15      $E_p \leftarrow t_i + \vartheta_p^m$ 
16     Mise à jour  $R_i, A_i, \mathcal{L}_i$  // Une autre intervention peut être planifiée à  $t_i$ 
17     Mélange de  $\mathcal{L}_i$  par bloc de valeurs égales de  $D_p$ 
18   fin
19 fin
20 Calculer  $\mathcal{L}_u$  // Liste des interventions non planifiée
21 retourner  $S, \mathcal{L}_u$ 

```

un mode m . Nous désignons par C_i l'ensemble des interventions chirurgicales des patients qui ont été achevées au moment t_i , formellement $C_i = \{p \in P, E_p \leq t_i\}$. Nous désignons par A_i l'ensemble des interventions chirurgicales actives des patients au temps t_i , formellement $A_i = \{p \in P, E_p - \vartheta_p^m \leq t_i < E_p\}$; ces interventions chirurgicales sont en cours. Les interventions chirurgicales planifiées jusqu'à t_i appartiennent soit à C_i soit à A_i . Nous désignons R_i comme l'ensemble de ressources de R_s ou R_r utilisé par les interventions actives dans l'ensemble A_i . Nous désignons par \mathcal{L}_i la liste des interventions qui peuvent être planifiées à l'instant t_i . Les interventions dans \mathcal{L}_i sont $\{p \in P \wedge p \notin A_i \wedge p \notin C_i, X_p \leq t_i\}$. Ces listes gèrent les interventions des patients à planifier tout au long de l'algorithme.

Une intervention pour un patient p parmi les \mathcal{L}_i peut éventuellement être réalisée en utilisant le mode m parmi M_p . Cependant, tous ces modes ne sont pas possibles au moment t_i , car les ressources peuvent déjà être utilisées par certaines interventions dans A_i qui ne sont pas encore achevées. Nous désignons par $M_{p,i}$ la liste des modes qui peuvent être utilisés au moment t_i . Les modes dans $M_{p,i}$ sont $\{m = (r_s, r_r) \in M_p, (r_s, r_r \notin R_i) \wedge (t_i + \vartheta_p^m \leq H_p) \wedge ([t_i, t_i + \vartheta_p^m] \in K_{r_s} \cap K_{r_r})\}$.

Dans la première boucle de l'algorithme 5, l'idée générale est de gérer les interventions à planifier en appliquant la règle de la date d'échéance la plus proche. Nous mettons d'abord à jour t_i en sélectionnant l'heure de l'intervention en cours dont la fin est la plus proche dans A_i , calculée à l'itération précédente. Ensuite, les nouveaux C_i, A_i, R_i et

\mathcal{L}_i qui correspondent au nouveau pas de temps i sont calculés. La liste \mathcal{L}_i , les interventions qui peuvent être planifiées au temps t_i , est ensuite triée par ordre croissant de D_p ; la priorité donnée à la date d'échéance la plus proche vise à minimiser le terme O_{wt} . La boucle se termine lorsque toutes les interventions sont planifiées ou lorsque plus aucune intervention ne peut être planifiée sans dépasser t_{end} .

Dans la deuxième boucle de l'algorithme 5, l'idée est de planifier autant d'interventions que possible au moment t_i en appliquant la règle du temps de traitement le plus court. La première intervention p qui peut être planifiée au moment t_i est sélectionnée. Ses modes utilisables $M_{p,i}$ à l'instant t_i sont calculés et ensuite triés par longueur croissante ϑ_p^m . Ensuite, le mode m ayant la plus petite longueur est choisi, l'intervention p est planifiée, et son heure de fin E_p est mise à jour. Les ensembles et la liste S , R_i , A_i et \mathcal{L}_i sont mis à jour. Pour une intervention planifiée p , on fixe $c_p = t_i$ pour l'heure de début de l'intervention. La boucle se termine lorsque plus aucune intervention ne peut être planifiée au temps t_i . Cette règle vise à libérer les ressources le plus rapidement possible.

Nous avons observé que les patients peuvent avoir les mêmes valeurs de date d'échéance D_p . Nous pouvons tirer parti de cette observation en introduisant une randomisation partielle pour obtenir des solutions différentes lors de plusieurs exécutions. Nous mélangeons les blocs consécutifs de \mathcal{L}_i qui correspondent à la même valeur de D_p (voir lignes 9 et 19), les interventions avec des D_p égaux sont choisies après aléatoirement.

L'algorithme 2PSC construit une solution S , donc \tilde{P} est connu. La liste des interventions non planifiées est calculées et retournée \mathcal{L}_u . Cette approche heuristique vise à minimiser le terme O_{wt} , mais peut être améliorée en considérant le terme O_{em} .

Nous proposons une phase d'amélioration EM dans le but d'améliorer le nombre d'interventions des patients de \tilde{P}^{em} qui sont planifiées tôt le matin. Nous essayons d'avancer les interventions $p \in \tilde{P}^{em}$ pour les planifier tôt le matin. Pour chaque patient $p \in \tilde{P}^{em}$, on obtient le jour d où l'intervention chirurgicale est planifiée, puis on obtient la liste de toutes les interventions planifiées de ce jour. Cette liste \mathcal{L}_d est triée par ordre croissant de l'élément temps de c_p . Étant donné que p se trouve en position i , nous essayons d'échanger l'intervention p avec chaque intervention planifiée auparavant. La fonction $\text{Swap}(s, i, j)$ renvoie Vrai lorsqu'il est possible d'échanger les interventions. A condition que les deux modes et les deux longueurs soient les mêmes, l'échange est effectué par cette fonction et S est mis à jour. Cette condition simple présente un avantage majeur, à savoir que nous pouvons obtenir une solution améliorée à moindre effort en évitant le temps de calcul nécessaire pour replanifier une partie de la solution.

Étant donné une solution S obtenue en appliquant EM, l'ensemble \tilde{P} est le même, et pour chaque jour d , l'ensemble des interventions planifiées est toujours le même. Nous rappelons que les paramètres X_p , D_p , H_p et le nombre de jours dans c_p sont des nombres entiers de jours par rapport au début de l'horizon de planification. Par Conséquence, les deux premiers termes de la fonction objectif ne changent pas. L'intervention des modes d'exécution des interventions conduit à une solution équivalente par rapport aux contraintes du problème (consommation et libération des ressources). L'heure de début

de certaines interventions de \tilde{P}^{em} . peut changer, ce qui peut diminuer le troisième terme de la fonction objectif.

La 2PSC-EM que nous proposons est une heuristique rapide dédiée qui calcule un planning conforme aux contraintes d'utilisation des ressources, tout en tenant compte des considérations médicales liées aux patients. En randomisant la liste \mathcal{L}_i , nous pouvons l'exécuter plusieurs fois pour obtenir différentes solutions.

4.3.2 Métaheuristique ALNS

L'heuristique 2PSC-EM peut obtenir une solution de bonne qualité, mais certaines interventions peuvent ne pas être planifiées. Il est important pour la direction de l'hôpital de savoir si les interventions pour tous les patients électifs peuvent être planifiées. L'approche ALNS a été présentée pour la première fois par [Ropke et al., 2006]. Nous fournissons une vue d'ensemble contextuelle qui considère le problème de planification des admissions traité ici. Nous présentons ensuite la structure générale de notre méthode Adaptive Large Neighborhood Search (ALNS) appliquée au problème des urgences dans la section 4.3.2.1.

Pour une recherche de voisinage large, des solutions similaires sont obtenues en modifiant la solution initiale par l'application de plusieurs opérateurs de recherche locale. Étant donné que certaines interventions peuvent ne pas être planifiées, nous explorons le voisinage d'une solution courante en appliquant d'abord un opérateur de destruction pour libérer les ressources.

Ensuite, toutes les interventions qui restent à planifier sont planifiées en appliquant un opérateur de réparation dans le but de planifier toutes les interventions lorsque cela est possible. Nous présentons les quatre opérateurs de destruction et les trois opérateurs de réparation que nous étudions dans les sections 4.3.2.2 et 4.3.2.3. Le degré de destruction doit être ajusté car il affecte la diversification. Un mécanisme de diversification adaptative est utilisé pour gérer le degré de destruction de certains opérateurs de destruction. Un couple d'opérateurs de destruction et de réparation joue le rôle d'un voisinage pour la recherche locale.

Le succès des opérateurs peut varier en fonction de l'instance du problème. Un choix adaptatif conduit généralement à de meilleurs résultats plutôt que de fixer le choix des opérateurs pour toute la durée de l'algorithme. Le mécanisme adaptatif présenté dans la section 4.3.2.4 est basé sur un principe de roulette que nous utilisons pour gérer le succès des opérateurs de destruction et le succès des opérateurs de réparation. Chaque opérateur a un poids qui représente sa part dans une roulette. Les poids sont mis à jour en fonction des performances, les opérateurs efficaces sont utilisés plus souvent que les opérateurs moins efficaces.

L'ALNS utilise ces opérateurs de destruction/réparation pour explorer le voisinage d'une solution. Cependant, cela peut ne pas être suffisant pour permettre au processus de recherche d'éviter ou d'échapper aux optima locaux. La méthode d'acceptation de l'ALNS a pour but de décider soit de continuer à explorer le voisinage d'une solution actuelle, soit de sélectionner une solution nouvellement créée. Une solution de moindre qualité peut être acceptée dans le but d'échapper aux optima locaux. Nous présentons

dans la section 4.3.2.5 une stratégie d'acceptation adaptative basée sur l'algorithme *record-to-record travel*, initialement proposé par [Dueck, 1993].

L'ALNS que nous proposons utilise des mécanismes adaptatifs pour les procédures de destruction, de construction et d'acceptation. Plusieurs paramètres doivent être réglés pour obtenir une bonne efficacité de ces mécanismes adaptatifs.

4.3.2.1 Structure générale de l'ALNS

L'algorithme 6 donne la structure générale de l'ALNS que nous décrivons globalement avant de fournir des informations plus détaillées.

Algorithme 6 : Adaptive Large Neighborhood Search (ALNS) algorithm

Entrées : S_0 solution initiale, \mathcal{L}_u liste des interventions non planifiée
Sorties : S_{best} la meilleure solution trouvée, \mathcal{L}_u liste des interventions non planifiée
Paramètres : D_{limit} limite pour le degré de diversification
 $Iter_{max}$ nombre maximal d'itérations sans aucune amélioration
Variables : S_{cur} solution courante, i nombre d'itérations
 D_{max} nombre maximum d'interventions à retirer sur plusieurs jours
 O_{des} l'opérateur de destruction sélectionné, O_{rep} l'opérateur de réparation sélectionné
 k nombre d'interventions à retirer sur plusieurs jours
 S_i solution en cours

```

1  $S_{best}, S_{cur} \leftarrow S_0$  // Initialiser la meilleure solution et la solution courante
2  $i \leftarrow 0$ 
3  $D_{max} \leftarrow 3$ 
4 tant que  $i < Iter_{max}$  faire
5    $O_{des} \leftarrow SelectDestructionOperator()$  // Sélectionner un opérateur de destruction
6    $k \leftarrow 1$ 
7   si  $\neg DayDestroy(O_{des})$  alors  $k \leftarrow rand(1, D_{max})$ 
8   // Retirer k interventions sur plusieurs jours, ou toutes les interventions d'un jour
9    $(S_i, \mathcal{L}_{des}) \leftarrow Destruction(S_{cur}, k, O_{des})$ 
10   $O_{rep} \leftarrow SelectRepairOperator()$  // Sélectionner un opérateur de réparation
11   $S_i \leftarrow Construction(S_i, \mathcal{L}_{des}, \mathcal{L}_u, O_{rep})$  // Insérer le plus grand nombre possible d'interventions
12  // Procédure d'acceptation sur  $S_i$ , solution en cours d'exécution
13  si  $Accept(S_{best}, S_i)$  alors
14     $S_{cur} \leftarrow S_i$  //  $S_i$  est acceptée comme la nouvelle solution actuelle
15     $i \leftarrow 0$ 
16  fin
17  sinon
18     $i \leftarrow i + 1$ 
19  fin
20  si  $Obj(S_i) < Obj(S_{best})$  alors
21     $S_{best} \leftarrow S_i$  // Une nouvelle meilleure solution est trouvée
22     $D_{max} \leftarrow 3$ 
23  fin
24  sinon
25    // Mise à jour du degré maximal de diversification adaptative
26    si  $\neg DayDestroy(O_{des})$  alors  $D_{max} \leftarrow \min(D_{max} + 1, D_{limit})$ 
27  fin
28  // Mise à jour des poids des deux roulettes (Destruction/Réparation)
29   $AdaptiveWeightsAdjustment(S_{best}, S_{cur}, S_i)$ 
30 fin
31 retourner  $S_{best}, \mathcal{L}_u$ 

```

Étant donné une solution initiale S et la liste \mathcal{L}_u des interventions non planifiées, toutes deux produites par l'heuristique 2PSC-EM (voir la section 4.3.1), l'algorithme alterne les phases de destruction et de réparation sur une solution courante S_{cur} pour obtenir une solution S_i à l'itération i (lignes 5-10).

La solution en cours est désignée par S_i . Cependant, nous ne créons pas un nouveau conteneur pour chaque solution en cours pour chaque itération i . L'indice i est utilisé à des fins d'explication. Lorsque la solution S_i remplit les critères d'acceptation, elle est retenue comme la nouvelle solution courante S_{cur} , et, le nombre d'itérations i est remis à zéro sinon il est incrémenté (lignes 11-15). La meilleure solution trouvée jusqu'à présent au cours des itérations S_{best} est conservée (lignes 16-18). À condition de respecter un nombre maximal d'itérations $Iter_{max}$ sans aucune amélioration, l'algorithme ALNS s'arrête et renvoie alors S_{best} et \mathcal{L}_u .

Pour les opérateurs de destruction orientés intervention/patient ($DayDestroy(O_{des})=false$), un nombre aléatoire de k interventions limité par D_{max} doit être supprimées (ligne 7). Un mécanisme de diversification adaptative gère D_{max} (ligne 20), ce qui permet de diversifier la recherche autour de la meilleure solution trouvée jusqu'à présent, avec l'intention de trouver une meilleure solution (voir section 4.3.2.2). L'opérateur de destruction est effectué en utilisant $Destruction(S_{cur}, k, O_{des})$ (ligne 8). Cela libère des ressources mais une liste supplémentaire \mathcal{L}_{des} d'interventions n'est pas planifiée.

L'opérateur de réparation O_{rep} est sélectionné (ligne 9) parmi un ensemble d'opérateurs de réparation (voir la section 4.3.2.3). Ensuite, il est utilisé dans $Construction(S_i, \mathcal{L}_{des}, \mathcal{L}_u, O_{rep})$ (ligne 10) pour planifier toutes les interventions non planifiées (\mathcal{L}_{des} et \mathcal{L}_u), lorsque cela est possible. Étant donné une solution en cours S_i , l'ordre dans lequel les interventions \mathcal{L}_{des} et \mathcal{L}_u sont traitées a un impact sur le nombre d'interventions qui peuvent être planifiées. Cela dépend des ressources libres dans les intervalles. Les opérateurs de réparation que nous proposons sont basés sur la façon dont les interventions de \mathcal{L}_{des} et \mathcal{L}_u sont triées.

Afin d'échapper aux optimums locaux, nous permettons l'acceptation d'une solution non améliorante (ligne 11). Le problème principal à résoudre est d'éviter de sélectionner une solution de trop faible qualité, conduisant à une perte de temps de traitement lors de l'exploration de son voisinage dans l'espoir d'obtenir une solution de meilleure qualité. Nous avons choisi d'implémenter une stratégie d'acceptation adaptative, que nous détaillons dans la section 4.3.2.5, basée sur l'algorithme *record-to-record travel* proposé dans [Dueck, 1993].

Il est important de permettre la sélection de bons opérateurs de destruction/construction au fil des itérations afin de pouvoir planifier toutes les interventions. Le mécanisme adaptatif que nous mettons en œuvre est détaillé dans la section 4.3.2.4. Les poids des deux roulettes sont mis à jour en utilisant $AdaptiveWeightsAdjustment(S_{best}, S_{cur}, S_i)$.

Le nombre d'itérations sans amélioration $Iter_{max}$ peut être fixé indépendamment de l'instance ou calculé au cours de l'algorithme ALNS en fonction de certaines caractéristiques de l'instance traitée afin de s'adapter à la difficulté de l'instance.

Nous avons choisi de formuler $Iter_{max}$ comme $\gamma \frac{|P \setminus \tilde{P}|}{|R|} + 1 \cdot \frac{|\tilde{P}|}{|R|}$ où $\gamma \in [1, 10]$. Nous rappelons que l'ensemble des interventions planifiées au cours de l'algorithme est \tilde{P} , l'ensemble des patients laissés non planifiées est $P \setminus \tilde{P}$, et l'ensemble des ressources est R . Par conséquent, $Iter_{max}$ augmente principalement lorsque le nombre moyen d'interventions planifiées par ressource $\frac{|\tilde{P}|}{|R|}$ augmente. Cela nous permet de passer plus

de temps de traitement lorsque toutes les interventions sont planifiées afin de mieux améliorer les termes O_{wt} et O_{em} de la fonction objectif.

Cependant, nous devons également avoir un plus grand nombre d'itérations lorsque les interventions ne sont pas planifiées afin de planifier toutes les interventions. Le terme $\gamma^{\frac{|P \setminus \tilde{P}|}{|R|} + 1}$ est égal à la valeur du paramètre γ lorsque toutes les interventions sont planifiées, puisque $|P \setminus \tilde{P}|$ est nul car l'ensemble des interventions non planifiées est vide. Il est plus grand lorsque de nombreuses interventions ne sont pas planifiées. Le paramètre γ doit être réglé pour obtenir un bon compromis entre les solutions de bonne qualité et les temps de traitement.

4.3.2.2 Opérateurs de destruction

Nous avons étudié quatre opérateurs de destruction, deux qui sont orientés intervention/patient et deux autres qui sont orientés jour. Tous ces opérateurs utilisent la solution courante S_{cur} comme entrée, et les opérateurs intervention/patient suppriment un certain nombre d'interventions tandis que les opérateurs jour suppriment toutes les interventions planifiées en un jour. Étant donné un opérateur de destruction O_{des} , la fonction $Destruction(S_{cur}, k, O_{des})$ (voir Algorithme 6) renvoie S_i , la solution en cours avec les interventions supprimées, et \mathcal{L}_{des} la liste des interventions supprimées. Les opérateurs de destruction que nous avons étudiés sont :

DAP Destruction Aléatoire Patient (orienté patient) ;

DCP Destruction Coût Patient (orienté patient) ;

DAJ Destruction Aléatoire d'une Journée (orienté jour) ;

DCJ Destruction de Coût Journée (orienté jour).

DAP retire k interventions chirurgicales de manière aléatoire de la solution et les insère l'une après l'autre dans \mathcal{L}_{des} . Cette opération peut être effectuée en temps constant. **DCP** évalue d'abord puis trie les interventions par ordre décroissant de coût, voir les équations (4.1a)-(4.1c). Ceci peut être fait en $\mathcal{O}(|P| \cdot \ln(|P|))$.

Ensuite, les k interventions chirurgicales dont le coût de planification est le plus élevé sont supprimées. Le coût de planification d'un patient p est évalué par la différence de coût, en supposant le retrait de l'intervention du patient dans la solution. Ceci vise à diminuer le coût. **DAJ** choisit un jour de l'horizon de planification de manière aléatoire. Ensuite, toutes les interventions planifiées pour ce jour sont supprimées de la solution. Au maximum, cette opération peut être réalisée en $\mathcal{O}(|P|)$. **DCJ** évalue d'abord toutes les interventions planifiées pour chaque jour, puis classe les jours par ordre décroissant de coût. Ensuite, toutes les interventions du jour ayant le coût de planification le plus élevé sont supprimées de la solution. Ceci peut être réalisé en $\mathcal{O}(|P| \cdot \ln(|P|))$.

Tous ces opérateurs de destruction visent à libérer des ressources, ce qui permet aux opérateurs de réparation de planifier autant d'interventions qu'ils le peuvent. **DAP** et **DAJ** sont des opérateurs de destruction purement aléatoires. Ils permettent de diversifier le mécanisme de recherche ([Hemmati et al., 2017]).

En se focalisant sur les jours, **DAJ** et **DCJ** permettent de réallouer les ressources au sein d'une même journée, tout en visant à diminuer le coût, notamment en ce qui concerne le troisième terme O_{em} de la fonction objectif (4.1c) relatif aux patients pour lesquels l'intervention doit être planifiée tôt le matin.

Pour **DCP**, nous évaluons la contribution individuelle de chaque intervention planifiée p dans S_{cur} . Cela nous permet de supprimer les k interventions dont le coût est le plus élevé. L'opérateur de destruction **DCJ** évalue également les contributions individuelles. **DCP** et **DCJ** fonctionnent tous deux au prix d'un temps de traitement plus important mais cela permet de réduire le coût global $Obj(S_i)$ à l'aide des opérateurs de réparation. À chaque itération, la fonction `SelectDestructionOperator()` (voir Algorithme 6) sélectionne un opérateur de destruction O_{des} qui est choisi parmi les quatre décrits ci-dessus à l'aide d'un mécanisme adaptatif basé sur un algorithme de sélection par roulette (voir Section 4.3.2.4).

Notez que le paramètre $k = 1$ Lorsqu'un opérateur orienté jour est choisi puisqu'un jour doit être nous avons $k = 1$. Le degré de destruction k affecte la diversification et par conséquent la qualité des solutions. Il peut être fixe, choisi au hasard dans un intervalle dépendant de la taille de certaines caractéristiques de l'instance, ou varier progressivement au moment de l'exécution selon une stratégie. Nous avons choisi d'implémenter une stratégie adaptative de gestion de k pour **DAP** et **DCP** qui utilise le nombre de modes et le nombre d'interventions planifiées que nous obtenons au cours de l'algorithme ALNS.

Dans l'algorithme 6, la valeur k est choisie au hasard comme $k \leftarrow \text{rand}(1, D_{max})$. Dans la mesure où un opérateur de destruction orienté intervention/patient a été choisi, k est limité par D_{max} . Nous définissons D_{max} comme le degré de diversification. Cette valeur est initialisée à 3 et ensuite incrémentée après chaque itération non améliorante comme $D_{max} \leftarrow \min(D_{max} + 1, D_{limit})$, lorsque la valeur de D_{limit} n'est pas atteinte. Dès qu'une amélioration est trouvée, nous réinitialisons D_{max} à 3 afin d'explorer entièrement le voisinage de la nouvelle meilleure solution S_{best} .

Le choix de la valeur de D_{limit} est important car cela permet de diversifier la recherche autour d'une solution pour trouver une meilleure solution. Cependant, une valeur trop grande pourra générer une dépense inutile de temps de traitement car une meilleure solution pourrait être trouvée dans une autre partie de l'espace de recherche à partir d'une solution de moindre qualité qui pourrait être sélectionnée grâce au mécanisme d'acceptation (voir la section 4.3.2.5).

Nous avons choisi de formuler D_{limit} comme $\alpha \frac{|\tilde{P}|}{|M|}$ où \tilde{P} est l'ensemble des interventions (i.e. patients) déjà planifiées et M les modes possibles qui peuvent être utilisés. Tel que formulé, D_{limit} est globalement proportionnel au nombre moyen d'interventions planifiées par mode qui sont utilisées par l'instance. Puisque $|\tilde{P}| \leq |P|$ lorsqu'un certain nombre d'interventions ne sont pas planifiées dans les solutions, cela favorise la stratégie d'acceptation adaptative des solutions de faible qualité afin de trouver des solutions où toutes les interventions sont planifiées. Comme il augmente avec le nombre d'interventions planifiées, il permet alors de mieux explorer le voisinage des solutions.

La taille de \tilde{P} est plus grande que la taille de M , et le temps de traitement augmente avec le nombre d'interventions supprimées. Nous avons choisi de sélectionner $\alpha \in [0, 1]$. Le paramètre α doit être ajusté pour obtenir des solutions de bonne qualité.

Au fil des itérations de l'ALNS, le mécanisme adaptatif permet de choisir parmi les quatre opérateurs de destruction pour explorer le voisinage d'une solution.

4.3.2.3 Opérateurs de réparation

Dans la littérature, l'algorithme *Best Insertion Algorithm* (BIA) est souvent utilisé pour la réparation de solutions. Globalement, l'idée est d'insérer des éléments dans une solution courante à un endroit choisi comme le meilleur selon des critères spécifiques (Amarouche et al., 2020; Ben-Said et al., 2019).

En nous appuyant sur le concept général, nous avons proposé un algorithme dédié, comme le montre l'algorithme 7. Étant donné une solution à l'itération i S_i , nous calculons l'ensemble qui contient les triplets suivants : chirurgien, salle et intervalle de temps qui peuvent être utilisés pour planifier une intervention. Nous obtenons l'ensemble des (s, r, J) libres relatifs à S_i (modes et intervalles). Nous désignons cet ensemble par $\mathcal{S}_{sr(ti)}$, indexé par $sr(ti)$ pour chirurgien (*surgeon*), salle (*room*) et intervalle de temps (*time interval*). Le calcul initial de $\mathcal{S}_{sr(ti)}$ prend $\mathcal{O}(H \cdot |P|)$ itérations, ensuite $\mathcal{S}_{sr(ti)}$ est mis à jour en temps constant au cours de l'algorithme ALNS.

Algorithme 7 : Meilleur insertion patient

Entrées : S_i solution en cours à l'itération i
 \mathcal{L}_{itbs} liste ordonnée/priorisée des interventions à planifier,
 $\mathcal{S}_{sr(ti)}$ ensemble de triplets (chirurgien, salle, intervalle de temps) pouvant être utilisés pour la planification

Sorties : S_i solution réparée à l'itération i
 \mathcal{L}_u liste des interventions non planifiée

Variables : $\mathcal{S}_{p, sr(ti)c}$ pour l'intervention p ,
ensemble de quadruplets (chirurgien, salle, intervalle, coût)

```

1  $\mathcal{L}_u \leftarrow \emptyset$ 
2 tant que ( $\mathcal{L}_{itbs} \neq \text{nil}$ ) faire
   // L'intervention suivante de la liste ordonnée des interventions à planifier
3    $p \leftarrow \mathcal{L}_{itbs}.\text{pop\_front}()$ 
4    $\mathcal{S}_{p, sr(ti)c} \leftarrow \emptyset$ 
   // Sélectionner les modes et les intervalles pour l'intervention  $p$ 
   // Evaluation du coût
5   pour tous  $(s, r, J) \in \mathcal{S}_{sr(ti)}$  faire
6     si  $(m = (s, r) \in M_p) \wedge (\vartheta_p^m \leq \text{size}(J))$  alors
7        $\text{cost}_p \leftarrow \text{DiffCostAdd}(S_i, p, s, r, J)$ 
8        $\mathcal{S}_{p, sr(ti)c} \leftarrow \mathcal{S}_{p, sr(ti)c} \cup \{(s, r, J, \text{cost}_p)\}$ 
9     fin
10  fin
11  si  $\mathcal{S}_{p, sr(ti)c} \neq \emptyset$  alors
12     $(p, s_{\text{best}}) \leftarrow \text{SelectBest}(\mathcal{S}_{p, sr(ti)c})$ 
13    Planifier l'intervention  $p$  dans la solution  $S_i$  en utilisant le mode  $m$  et intervalles  $J$ 
14    Mise à jour de l'ensemble  $\mathcal{S}_{sr(ti)}$ 
15  fin
16  sinon
17     $\mathcal{L}_u.\text{push\_back}(p)$ 
18  fin
19 fin
20 retourner  $S_i, \mathcal{L}_u$ 

```

L'objectif est de planifier autant d'interventions que possible tout en minimisant le coût. L'algorithme utilise la solution S_i et une liste ordonnée/priorisée d'interventions à planifier, désignée par \mathcal{L}_{itbs} . La liste \mathcal{L}_{itbs} contient les interventions non planifiées de \mathcal{L}_u et les interventions de \mathcal{L}_{des} qui ont été retirées en appliquant l'un des quatre opérateurs de destruction. Étant donné une liste initiale de \mathcal{L}_{itbs} , le BIA prend $\mathcal{O}(H \cdot |P|)$. La disposition des interventions dans la liste \mathcal{L}_{itbs} a un impact sur le nombre d'interventions pouvant être planifiées en utilisant l'ensemble $\mathcal{S}_{sr(ti)}$ des ressources disponibles dans les intervalles. Nous proposons et examinons trois manières différentes de construire l'ordre des interventions dans \mathcal{L}_{itbs} .

L'algorithme 7 réinitialise d'abord \mathcal{L}_u pour obtenir la liste des interventions non planifiées pour la prochaine itération de l'ALNS. Étant donné une liste de priorités \mathcal{L}_{itbs} , nous essayons d'insérer les interventions une après l'autre dans la solution en cours S_i . Pour une intervention p , nous déterminons d'abord les modes $m = (s, r) \in M_p$ et les intervalles associés \mathcal{J} parmi l'ensemble des triplets $\mathcal{S}_{sr(ti)}$ qui peuvent être utilisés pour planifier cette intervention p . Nous évaluons ensuite le coût associé, noté $cost_p$, de l'insertion de l'intervention p dans S_i . Nous obtenons donc l'ensemble des quadruplets $\mathcal{S}_{p, sr(ti)c}$ chirurgien s , salle r , intervalle (ti) et coût c . Sous réserve que $\mathcal{S}_{p, sr(ti)c}$ ne soit pas vide, nous sélectionnons le meilleur quadruplet d'insertion avec le coût minimum et nous planifions l'intervention p en utilisant le mode m et l'intervalle \mathcal{J} associés. L'ensemble $\mathcal{S}_{sr(ti)}$ est ensuite mis à jour pour l'itération suivante puisque des ressources ont été utilisées. Si l'ensemble $\mathcal{S}_{p, sr(ti)c}$ est vide, l'intervention p ne peut pas être planifiée et elle est ajoutée à la liste \mathcal{L}_u .

L'algorithme d'insertion adapté et la liste ordonnée/priorisée \mathcal{L}_{itbs} jouent ensemble le rôle d'un opérateur de réparation. On peut avoir différentes solutions en utilisant différentes listes de priorités. Étant donné les deux listes \mathcal{L}_u et \mathcal{L}_{des} , nous ajoutons d'abord la liste \mathcal{L}_{des} derrière la liste \mathcal{L}_u pour obtenir la liste initiale \mathcal{L}_{itbs} . Nous proposons ensuite trois façons d'ordonner/prioriser les interventions dans \mathcal{L}_{itbs} :

INP Interventions Non planifiées en Premier ;

NMD triés par ordre croissant du Nombre de Modes Demandés ;

AC triés dans l'ordre croissant d'une Agrégation des Critères.

Calculer la liste \mathcal{L}_{itbs} prend au maximum $\mathcal{O}(|P| \cdot \ln(|P|))$.

Le premier opérateur de réparation **INP** vise à donner la priorité aux interventions non planifiées restant dans \mathcal{L}_u . L'idée est de libérer des ressources en appliquant un opérateur de destruction pour planifier d'abord les interventions de \mathcal{L}_u , nous planifions ensuite les interventions de \mathcal{L}_{des} .

De manière générale, l'idée pour l'opérateur de réparation **NMD** est de planifier d'abord les interventions avec le moins de modes, en supposant que nous aurons plus de possibilités de planifier les autres au fur et à mesure que les choses avancent lors de la planification des interventions.

Plus généralement, il peut être intéressant de considérer globalement tous les éléments liés à la planification de l'intervention d'un patient. Il ne faut pas perdre de vue que les contraintes médicales d'échéance dure H_p et de date d'échéance D_p doivent

également être prises en considération. Les modes de planification de l'intervention d'un patient ainsi que les ressources peuvent aussi avoir un impact.

Nous proposons d'étudier une stratégie pour le troisième opérateur de réparation **AC** en se basant sur l'agrégation des critères liés au niveau d'urgence du patient (H_p et D_p) et ceux liés aux modes et aux ressources. Soit n_{mode} le nombre de modes différents qui peuvent être utilisés pour un patient et n_{res} le nombre de ressources différentes que nous comptons sur les différents modes. Nous calculons d'abord $H_p \cdot D_p \cdot n_{\text{res}} \cdot n_{\text{mode}}$, puis nous trions $\mathcal{L}_{\text{itbs}}$ dans l'ordre croissant de ces valeurs. Une valeur faible indique que le patient est urgent, que la ressource est critique, et qu'il y a peu de modes d'intervention pour le patient.

L'idée de l'opérateur de réparation **AC** est de planifier d'abord les interventions des patients avec des valeurs faibles, en supposant que nous aurons plus de possibilités de planifier les autres au fur et à mesure de l'avancement de la planification des interventions.

Au fil des itérations de l'ALNS, le mécanisme adaptatif permet de choisir parmi les trois opérateurs de réparation pour explorer le voisinage d'une solution.

4.3.2.4 Mécanisme adaptatif de destruction/réparation

L'idée est de favoriser les choix futurs d'opérateurs performants en modifiant les poids utilisés dans un algorithme de roulette qui sélectionne les opérateurs. Les poids de la roulette utilisée pour les opérateurs de destruction et les poids de l'autre roulette utilisée pour les opérateurs de réparation sont mis à jour en utilisant la procédure *AdaptiveWeightsAdjustment* en fonction des performances obtenues.

L'algorithme ALNS proposé utilise quatre opérateurs de destruction et trois de réparation. Le succès des opérateurs peut varier en fonction de l'instance du problème. L'ajustement des poids de la roulette est nécessaire pour augmenter la probabilité que les opérateurs efficaces soient utilisés plus souvent que les opérateurs moins efficaces. L'ajustement dynamique permet d'assurer une réévaluation permanente des poids des opérateurs.

Deux approches peuvent être utilisées pour mettre à jour les poids : à chaque itération ou périodiquement après un certain nombre d'itérations. L'avantage de la première approche est que les poids sont à jour à tout moment mais au détriment du temps de traitement puisque les poids et les probabilités des opérateurs pour la prochaine itération doivent être calculés à chaque itération. Au contraire, en mettant périodiquement à jour les poids après un certain nombre d'itérations p_u , on gagne en temps de traitement. Cependant, cela implique de déterminer une bonne valeur pour p_u . Une valeur élevée permet de gagner du temps de traitement mais ne permet pas au mécanisme adaptatif de jouer pleinement son rôle, ce qui peut aboutir à des solutions de mauvaise qualité. Pour gagner du temps de traitement, nous avons choisi la deuxième approche. Le paramétrage de p_u est donc nécessaire pour obtenir un bon compromis entre le gain de temps de calcul et la qualité des solutions.

Le mécanisme adaptatif est le même pour la sélection de l'opérateur de destruction ou de réparation. Nous désignons O comme étant soit un opérateur de destruction, soit un opérateur de réparation. Le nombre de fois où l'opérateur O a été utilisé pendant p_u

itérations est noté $u(O)$. Le succès de l'opérateur O est noté $s(O)$, fixé à zéro au début de la période de p_u itérations. Étant donné la solution en cours S_i , la solution actuelle S_{cur} , et la meilleure solution trouvée jusqu'à présent S_{best} , la valeur de $s(O)$ est augmentée à chaque itération comme :

$$s(O) + \sigma_i \quad \text{où } \sigma_i \text{ est soit } \begin{cases} \sigma_1 & \text{si } \text{Obj}(S_i) \leq \text{Obj}(S_{best}) \\ \sigma_2 & \text{si } \text{Obj}(S_i) \leq \text{Obj}(S_{cur}) \\ \sigma_3 & \text{sinon} \end{cases}$$

Dans le premier cas, l'opérateur O améliore ou égale la meilleure solution trouvée jusqu'à présent, tandis que dans le second cas, il améliore ou égale la solution courante. Dans le dernier cas, nous récompensons l'opérateur lorsqu'une nouvelle solution est plus mauvaise mais acceptée selon la stratégie d'acceptation (voir section 4.3.2.5). Nous avons $\sigma_1 > \sigma_2 > \sigma_3$. L'opérateur est ainsi favorisé lorsque la qualité relative ou absolue de la solution augmente. Ces paramètres doivent être réglés car ils ont un impact sur la qualité de la solution. Nous avons choisi de formuler σ_2 comme $\sigma_2 = \theta \cdot \sigma_1$ et σ_3 comme $\sigma_3 = \phi \cdot \sigma_2$ où $\theta, \phi \in [0, 1]$. À condition qu'une valeur initiale soit fixée pour σ_1 , le paramétrage de $\sigma_1, \sigma_2, \sigma_3$ peut être réalisée en faisant varier le couple de paramètres (θ, ϕ) . Les poids pour les itérations à venir sont calculés comme suit :

$$w(O) = \begin{cases} (1-l)w(O) + l \frac{s(O)}{u(O)}, & \text{si } u(O) > 0 \\ (1-l)w(O), & \text{si } u(O) = 0 \end{cases}$$

Le facteur de réaction $l \in [0, 1]$ est un paramètre qui contrôle l'influence du succès récent d'un opérateur sur son poids.

Les quatre opérateurs de destruction et les trois opérateurs de réparation permettent d'effectuer des changements importants d'une solution à ses voisines, ce qui peut empêcher le processus de recherche d'être bloqué dans des optima locaux. Cependant, comme le succès des opérateurs peut varier en fonction de l'instance traitée, le mécanisme adaptatif joue un rôle important en récompensant les opérateurs efficaces tout au long des itérations. Pour rendre le mécanisme adaptatif efficace, les paramètres présentés ci-dessus doivent être réglés.

4.3.2.5 Stratégie d'acceptation adaptative

Nous avons choisi d'implémenter une stratégie d'acceptation adaptative basée sur l'algorithme *record-to-record travel*. Il est proposé à l'origine par [Dueck, 1993], il s'agit d'une variante déterministe du recuit simulé. Une solution de faible qualité est acceptée en fonction de la différence entre le coût de la solution en place $\text{Obj}(S_i)$ et le coût de la meilleure solution $\text{Obj}(S_{best})$ par rapport à une valeur appelée *record-deviation*, qui est une stratégie basée sur un seuil. Ce mécanisme simple mais efficace a l'avantage de ne dépendre que d'un seul paramètre, qui est la valeur de l'écart record, noté ici Dev (voir [Dueck, 1993]).

Le paramètre de *record-deviation* Dev joue un rôle central dans le contrôle de l'acceptation des solutions de faible qualité. Dans le schéma général de l'algorithme

présenté dans [Dueck, 1993], le paramètre de *record-deviation* est fixe et utilisé à l'origine comme $Dev \leq Obj(S_i) - Obj(S_{best})$. Nous avons choisi de l'implémenter par rapport à un pourcentage comme $Obj(S_i) \leq (1 + Dev) \cdot Obj(S_{best})$ car les interventions non planifiées (reportées d'un point de vue chirurgical) contribuent fortement au coût global de la fonction objectif (4.1a)-(4.1c). Il est préférable d'utiliser ce pourcentage plutôt qu'une valeur de différence absolue à des fins de comparaison entre les instances pour lesquelles toutes les interventions sont planifiées et celles pour lesquelles toutes les interventions ne sont pas planifiées.

Néanmoins, plus ce paramètre est grand, plus nous explorons inutilement des solutions de faible qualité parce que le mécanisme *record-to-record* est activé trop tôt. Pour avoir la possibilité d'explorer d'abord le voisinage d'une solution courante en utilisant les opérateurs de destruction/réparation, nous proposons une stratégie d'acceptation adaptative que nous gérons dans la procédure $Accept(S_{best}, S_i)$ (voir Algorithme 6). Ainsi, la valeur de Dev n'est pas fixe, elle peut augmenter ou diminuer tout au long du processus d'exploration de l'espace de recherche.

Cependant, l'augmentation de Dev peut se faire au détriment du mécanisme de diversification adaptative qui augmente également D_{max} , le nombre maximum d'interventions qui peuvent être retirées de S_i , lorsqu'une solution de moindre qualité est rencontrée. Dans l'algorithme 6, nous mettons à jour $D_{max} \leftarrow \min(D_{max} + 1, D_{limit})$ lorsque $Obj(S_i) \geq Obj(S_{best})$ pour certains opérateurs de destruction. Un bon compromis entre ces deux mécanismes doit être trouvé pour prioriser les recherches locales.

Dans la procédure $Accept(S_{best}, S_i)$, nous mettons en œuvre un mécanisme adaptatif qui augmente la valeur de Dev d'une valeur constante Δ_{Dev} lorsqu'aucune amélioration n'est rencontrée tout au long d'un nombre défini d'itérations $Iter_D$. Le paramètre *record-deviation* est égal à zéro au début ($Dev = 0$). Lorsque le nombre défini d'itérations $Iter_D$ sans amélioration a été effectué, Dev est augmenté de la valeur de *record-deviation* Δ_{Dev} pour permettre au mécanisme de diversification de jouer son rôle. Les solutions de faible qualité sont plus souvent acceptées, ce qui permet à la recherche de sortir des optima locaux. La valeur Dev augmente de Δ_{Dev} tant que les opérateurs de destruction/réparation ne parviennent pas à trouver une meilleure solution. Nous remettons le paramètre Dev à zéro si une amélioration locale ou globale se produit.

Par conséquent, les deux paramètres à régler pour rendre cette stratégie d'acceptation adaptative efficace sont l'incrément de *record-deviation* Δ_{Dev} et le nombre d'itérations sans amélioration $Iter_D$ qui induit l'incrément de Dev .

4.4 RÉSULTATS

Dans nos expériences, nos objectifs étaient : (i) de montrer l'efficacité de l'heuristique 2PSC-EM par rapport aux résultats de la littérature ; (ii) de fournir un résumé et des indications sur la façon de procéder au paramétrage de l'ALNS afin qu'il puisse être utilisé dans d'autres situations où la planification des patients électifs doit être effectuée sur une période de plusieurs mois ; (iii) évaluer l'efficacité des opérateurs de destruction/réparation et du mécanisme adaptatif pour la destruction/réparation et analyser

les résultats afin de montrer si un composant contribue à l'amélioration de la qualité des solutions. résultats pour montrer si un composant contribue à l'obtention de bonnes solutions en considérant également le temps de traitement ; et (iv) d'évaluer la qualité des solutions obtenues par l'ALNS sur des instances de référence par rapport aux résultats de la littérature. Les tests ont été réalisés en utilisant C++ et la Standard Template Library (STL), compilés avec GCC sous Linux, sur une machine avec un Intel(R) Xeon(R) Gold 6138 @ 2.00 GHz.

4.4.1 *Benchmark et protocole expérimental*

Dans ce travail, nous nous concentrons sur la construction de la planification des interventions chirurgicales, c'est-à-dire la planification des admissions. Pour l'organisation hospitalière l'objectif principale est de planifier les interventions de tous les patients électifs, qu'ils soient ambulatoires ou hospitalisés. Nous avons testé l'heuristique 2PSC-EM et la méta-heuristique ALNS sur l'ensemble des instances du benchmark proposé par [Riise et al., 2016]. Les auteurs ont conçu un générateur d'instances basé sur les ressources réelles d'un service de chirurgie (Bærum Sykehus, Norvège). Pour des raisons de confidentialité, les patients sont générés. Les données de test sont disponibles en ligne au format XML (SINTEF, 2013).

Chacune des dix instances du test de référence comprend 728 patients à planifier sur un horizon de planification compris entre 65 et 71 jours (environ 3 mois). Dans toutes les instances, il y a sept chirurgiens et quatre salles d'opération.

Dans [Riise et al., 2016], les auteurs ont choisi d'exécuter l'algorithme ACI dix fois pour chaque instance. Une limite de temps de 600 secondes a été fixée pour chaque exécution. Les auteurs ont rapporté les valeurs minimales, maximales et moyennes de la fonction objectif.

Dans [Riise et al., 2016], la meilleure valeur de la fonction objectif, désignée par UB, est également rapportée pour chaque instance. Cette valeur UB correspond à la meilleure valeur obtenue sur toutes des exécutions réalisées par les auteurs. Les temps de traitement pour obtenir les valeurs UB ne sont pas indiqués dans le document. Les auteurs ont indiqué qu'elles sont obtenues en utilisant des temps d'exécution considérablement plus longs que la limite de temps de 600 secondes.

4.4.2 *Comparaison de l'heuristique 2PSC-EM avec la littérature*

Dans le tableau 4.1, pour chaque instance sous l'intitulé ACI, la colonne UB indique la meilleure valeur connue, ou limite supérieure, la colonne Min indique la valeur minimale de la fonction objectif obtenue par ACI (voir [Riise et al., 2016]),

la colonne Min/UB indique le pourcentage $(UB - Min)/UB$, et la colonne $t(s)$ indique le temps de traitement. Étant donné que le poids W_{un} est fixé à 1,000 (voir le terme (4.1a)), les valeurs supérieures à mille indiquent que certaines interventions ne sont pas planifiées. Les résultats qui sont meilleurs que UB sont indiqués en caractères gras.

Comme notre algorithme est également aléatoire comme l'ACI, nous avons choisi d'effectuer initialement 100 exécutions pour déterminer si l'heuristique peut obtenir de

meilleurs résultats. Une exécution de l'heuristique 2PSC-EM prend environ dix secondes. Sous la rubrique *2PSC-EM (100)*, nous indiquons la valeur minimale obtenue et le pourcentage $(UB - Min)/UB$ dans les colonnes *Min* et *MizUB*, respectivement.

Pour huit instances sur dix, l'heuristique 2PSC-EM obtient de meilleurs résultats.

Pour les huit instances pour lesquelles toutes les interventions sont planifiées en utilisant l'approche ACI et l'heuristique 2PSC-EM proposée, nous améliorons l'UB de 16 % et 9 % en moyenne.

	ACI				2PSC-EM (100)			2PSC-EM (3)		
	UB	Min	MizUB%	t(s)	Min	MizUB%	t(s)	Min	MizUB%	t(s)
a_01	21.58	21.85	-1.27	600	21.15	1.97	1319	22.00	-1.94	42.0
a_02	22.83	23.25	-1.83	600	19.72	13.63	1335	20.32	11.01	40.0
a_03	24.52	24.58	-0.26	600	22.55	8.02	918	22.66	7.81	31.2
a_04	21.97	22.28	-1.40	600	20.43	7.02	958	20.62	6.16	29.7
a_05	5113.88	6341.84	-24.01	600	11025.10	-115.59	934	13025.30	-154.71	28.2
a_06	23.55	23.55	0.03	600	22.15	5.97	991	22.68	3.72	31.6
a_07	22.73	23.25	-2.28	600	21.98	3.31	1026	22.44	1.28	30.0
a_08	2086.32	3148.49	-50.91	600	5024.88	-140.85	797	10024.70	-380.50	23.9
a_09	24.83	24.99	-0.64	600	20.75	16.43	926	21.23	14.50	29.2
a_10	24.48	24.74	-1.08	600	20.61	15.83	935	20.92	14.53	25.9

Tableau 4.1: heuristique 2PSC-EM, 100 exécutions et trois exécutions, comparée à ACI.

À l'exception des instances a_05 et a_08 pour lesquelles ni l'ACI ni l'heuristique 2PSC-EM n'ont réussi à planifier toutes les interventions, l'heuristique 2PSC-EM que nous proposons fournit de meilleurs résultats. Ces résultats préliminaires utilisant l'heuristique 2PSC-EM montrent que de meilleurs résultats sont obtenus. Cependant, dans certains cas, des interventions ne sont toujours pas planifiées. Une approche plus efficace doit être étudiée dans le but de planifier toutes les interventions.

L'heuristique 2PSC-EM peut potentiellement fournir de bonnes solutions initiales à l'approche ALNS que nous nous proposons d'étudier. Cependant, nous ne pouvons pas dépenser trop de temps de traitement pour obtenir ces solutions. Nous avons réalisé donc une autre série d'expériences où nous limitons le nombre d'exécutions pour montrer si de bonnes solutions initiales peuvent être obtenues à l'aide de l'heuristique en utilisant un temps limité, tout en ne détériorant pas significativement la qualité des solutions. Comme l'heuristique est une méthode aléatoire, nous avons réalisé plusieurs expériences de k exécutions. Nous avons observé expérimentalement que $k = 3$ exécutions semble être un bon compromis entre la qualité des solutions et le temps de traitement.

Dans la Table 4.1 sous le titre *2PSC-EM (3)*, nous montrons les meilleurs résultats obtenus à l'aide de l'heuristique 2PSC-EM en effectuant trois exécutions de l'heuristique dix fois. Dans sept cas sur dix, l'heuristique 2PSC-EM obtient toujours de meilleurs résultats. Une légère dégradation est observée pour l'instance a_01 par rapport à UB. Pour les huit instances pour lesquelles toutes les interventions peuvent être planifiées, l'heuristique 2PSC-EM proposée améliore encore UB jusqu'à 14 % et 7,1 % en moyenne.

Comme on peut le voir, des résultats légèrement meilleurs peuvent être obtenus en utilisant 100 exécutions de l'heuristique 2PSC-EM par rapport à trois exécutions, mais c'est au détriment du temps de calcul. Nous avons observé expérimentalement que

trois exécutions sont suffisantes pour obtenir des solutions de bonne qualité dans un temps de traitement court.

Nous proposons l'approche ALNS dans le but de planifier toutes les interventions et d'améliorer les résultats de l'heuristique 2PSC-EM. Comme solutions initiales de l'approche ALNS, nous utilisons la meilleure solution calculée en utilisant trois exécutions de l'heuristique 2PSC-EM.

4.4.3 Paramétrage de ALNS

Nous présentons les expériences que nous avons menées pour trouver les valeurs des paramètres qui ont été utilisées pour tester notre approche ALNS. Comme décrit dans la section 4.3.2.1, l'ALNS que nous proposons possède quatre ensembles de paramètres :

- D_{limit} , le nombre maximal de patients retirés par les opérateurs de destruction ;
- Δ_{Dev} et Iter_{D} , le taux d'augmentation de l'écart par rapport au record et le nombre d'itérations sans amélioration avant augmentation de l'écart ;
- $\sigma_1, \sigma_2, \sigma_3, p_u$, et l , les trois premiers sont les scores ajoutés après l'utilisation d'un opérateur de destruction ou de réparation, p_u est le nombre d'itérations avant l'ajustement des poids des opérateurs, et l est le facteur de réaction ;
- Iter_{max} , le nombre maximal d'itérations sans amélioration.

Certains de ces paramètres sont calculés au cours de l'algorithme ALNS en fonction de certaines caractéristiques de l'instance traitée ou sont relatifs à d'autres paramètres. Pour ces paramètres, la première colonne du tableau 4.2 indique les paramètres tels que présentés dans la section 4.3.2.1. La deuxième colonne décrit comment ils sont calculés et les paramètres associés qui doivent être réglés.

	Calculé comme, et paramètres à régler
D_{limit}	$\alpha \cdot \frac{ \bar{P} }{ \bar{M} }$, paramètre α à régler
$\sigma_1, \sigma_2, \sigma_3$	σ_1 fixe, suivant $\sigma_2 = \theta \cdot \sigma_1$ et $\sigma_3 = \phi \cdot \sigma_2$ paramètres (θ, ϕ) à régler
Iter_{D}	$\delta \cdot \text{Iter}_{\text{max}}$, paramètre δ à régler
Iter_{max}	$\gamma \cdot \frac{ \bar{P} \setminus \bar{P}_1 + 1}{ \bar{R} } \cdot \frac{ \bar{P} }{ \bar{R} }$, paramètre γ à régler

Tableau 4.2: Paramètres $\alpha, \theta, \phi, \delta$ et γ à régler.

Pour calibrer tous les paramètres, nous avons réalisé des expériences préliminaires sur un sous-ensemble de cinq instances sélectionnées aléatoirement parmi les huit instances de référence pour lesquelles toutes les interventions peuvent être programmées. Comme ALNS est une méthode de recherche aléatoire, les expériences ont été répétées cinq fois avec une graine aléatoire différente.

Pour ajuster les paramètres, nous évaluons les résultats à l'aide du pourcentage d'erreur relatif (RPE). Nous définissons l'RPE comme suit : $\text{RPE} = 100 \times \frac{Z_{\text{best}} - Z_{\text{max}}}{Z_{\text{best}}}$

où Z_{best} désigne le meilleur résultat obtenu sur l'ensemble des exécutions effectuées pour une instance, et Z_{max} désigne le meilleur résultat que nous avons obtenu parmi les cinq exécutions réalisées. Ensuite, la moyenne des valeurs RPE est calculée pour les cinq instances choisies de manière aléatoire, et cette moyenne est indiquée en pourcentage dans les figures que nous présentons et que nous commentons ci-dessous.

Nous commençons par régler le premier D_{limit} , puis nous réglons chaque paramètre, l'un après l'autre, en considérant les quatre ensembles de paramètres présentés. Les valeurs initiales que nous utilisons pour régler le premier D_{limit} sont les suivantes $\Delta_{\text{Dev}} = 0.2$, $\delta = 0.2$, $\sigma_1 = 50$, $\theta = 0.3$, $\phi = 0.1$, $p_u = 60$, $l = 0.5$ et $\gamma = 10$. Ensuite, pour régler un paramètre, nous conservons le meilleur paramétrage des autres paramètres trouvés avant de procéder à son réglage.

Le nombre maximum de patients à retirer D_{limit}

Nous avons effectué les expériences de paramétrage en faisant varier α de 0,1 à 1,0 avec un pas de 0,05. Sur la Figure 4.1, nous montrons que lorsque α est inférieur à 0,2, le nombre d'interventions retirées de la solution est trop faible, l'exploration du voisinage est insuffisante. Lorsque α est supérieur à 0,35, le nombre d'interventions retirées d'une solution courante augmente. La diversification augmente donc la qualité des solutions se dégrade, comme on peut le constater sur la Figure 4.1.

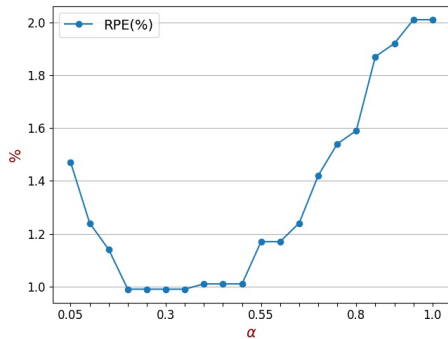


Fig 4.1: Réglage de D_{limit} , effet de α sur RPE.

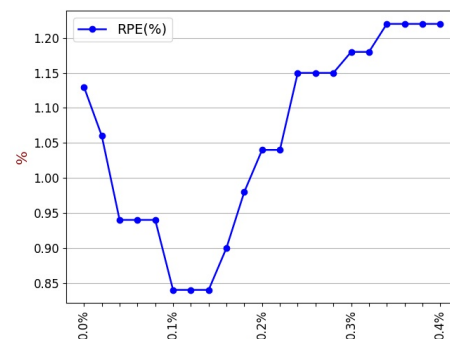


Fig 4.2: Réglage de Δ_{Dev} , effet de Δ_{Dev} sur RPE.

Comme on attendu, le temps de traitement augmente avec la valeur de α , nous ne montrons donc pas de courbe du temps CPU en fonction de α . Plus les valeurs de α sont grandes, plus les temps de traitement sont importants, ce qui est inefficace. Nous avons choisi $\alpha = 0.3$, qui est un bon compromis entre la qualité de la solution et le temps de traitement.

Stratégie d'acceptation adaptative basée sur le principe d'enregistrement à enregistrement ; réglage de Δ_{Dev} et Iter_D .

Pour calibrer la valeur de Δ_{Dev} , nous l'avons fait varier entre 0,0% et 2,5% avec des incréments de 0,02%. Dans la Figure 4.2, nous montrons que lorsque Δ_{Dev} varie entre zéro et 0,1%, le RPE diminue. Nous atteignons ensuite un plateau sur la courbe pour trois valeurs qui donnent la plus petite valeur de RPE. Ensuite, les

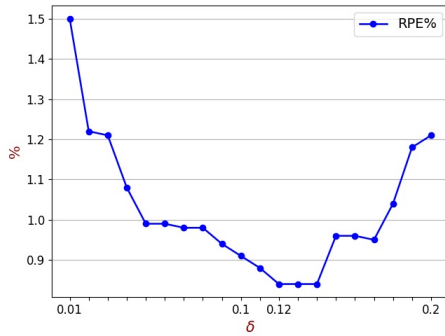


Fig 4.3: Réglage de Iter_D , effet de δ sur l'RPE.

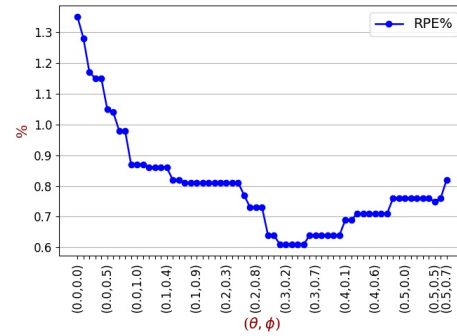


Fig 4.4: Réglage de σ_2 et σ_3 par rapport à σ_1 , effet de θ et ϕ sur le RPE.

valeurs moyennes de RPE augmentent lorsque Δ_{Dev} augmente. Nous observons ce même phénomène pour les valeurs de Δ_{Dev} supérieures à 0,4%. Par conséquent, les valeurs supérieures à 0,4% n'ont pas été représentées sur la Figure 4.2. Nous fixons la valeur de $\Delta_{Dev} = 0,1\%$ pour la déviation par rapport au record. Il s'agit de la plus petite valeur pour laquelle la valeur moyenne de RPE est minimale.

Nous procédons ensuite au paramétrage de δ pour Iter_D , le nombre d'itérations sans amélioration avant l'augmentation de la déviation. Dans la Figure 4.3, trois bonnes valeurs de δ peuvent être choisies ; la première est 0,12. Si Iter_D est petit, la déviation autorisée augmente rapidement et l'algorithme accepte des solutions de mauvaise qualité. En revanche, si Iter_D est grand, la déviation n'augmente pas autant et l'algorithme sera piégé dans un optimum local. Nous avons choisi $\delta = 0.12$.

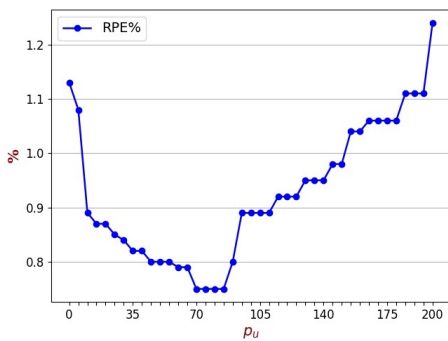


Fig 4.5: Effet de p_u sur l'RPE.

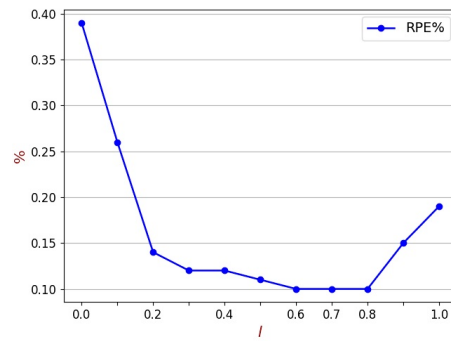


Fig 4.6: Effet de l sur l'RPE.

Paramètres d'ajustement du poids adaptatif ; réglage de $\sigma_1, \sigma_2, \sigma_3, p_u$ et l .

Nous commençons par régler les paramètres d'ajustement des poids $\sigma_1, \sigma_2, \sigma_3$, les scores ajoutés après l'utilisation d'un opérateur de destruction ou de réparation. Étant donné que $\sigma_1 > \sigma_2 > \sigma_3$ selon la récompense habituelle des opérateurs par rapport à

la qualité de la solution précédente S_i , de la solution courante S_{cur} , et de la meilleure solution trouvée jusqu'à présent, S_{best} , nous avons choisi de fixer d'abord σ_1 à 50 et nous calculons $\sigma_2 = \theta \cdot \sigma_1$ et $\sigma_3 = \phi \cdot \sigma_2$ tels que $\theta, \phi \in [0, 1]$.

La figure 4.4 montre le début de la courbe que nous obtenons pour mieux nous concentrer sur les valeurs intéressantes. Nous avons choisi $\theta = 0.3$ et $\phi = 0.2$, la première valeur du couple (θ, ϕ) donnant les meilleures performances par rapport au RPE moyen.

Pour paramétrer p_u , le nombre d'itérations avant d'ajuster les poids des opérateurs, nous le faisons varier entre 0 et 200 itérations avec un pas de 5. La Figure 4.5 montre que le RPE est meilleur lorsque p_u est compris entre 70 et 85, et nous avons choisi de fixer p_u à 70.

Pour trouver la meilleure valeur de l , le facteur de réaction qui contrôle l'influence du succès récent d'un opérateur sur son poids, nous le faisons varier entre 0 et 1 avec un incrément de 0.1. Les valeurs moyennes de RPE présentées sur la Figure 4.6 révèlent que de bonnes solutions sont obtenues lorsque l est compris entre 0.6 et 0.8. Les itérations précédentes semblent avoir une petite influence sur le calcul du poids. Nous avons choisi de fixer l à 0,7.

Nombre maximum d'itérations sans amélioration ; réglage de $Iter_{max}$

Enfin, nous procédons au paramétrage de $Iter_{max}$, le nombre maximal d'itérations sans amélioration. Le temps de traitement nécessaire pour trouver une solution de bonne qualité varie en fonction des instances et ce paramètre a un impact important. Nous avons choisi de formuler $Iter_{max}$ comme $\gamma^{\frac{|P/\tilde{P}|}{|R|} + 1} \cdot \frac{|\tilde{P}|}{|R|}$ où $\alpha \in [1, 10]$, $|\tilde{P}|$ est le nombre d'interventions programmées, $|P/\tilde{P}|$ est le nombre d'interventions non programmées et $|R|$ est le nombre de ressources.

Comme attendu, le RPE moyen diminue à mesure que γ augmente, et le temps de traitement augmente avec la valeur de γ . Par conséquent, dans un souci de compacité, nous ne montrons pas les courbes pour le paramétrage de γ . La forme de la courbe a un comportement asymptotique comme celui de $1/x$. Plus les valeurs de γ sont grandes, plus les temps de traitement sont importants, ce qui est inefficace. Nous avons choisi $\gamma = 11$, qui est un bon compromis entre la qualité de la solution et le temps de traitement et qui reste généralement inférieur à 250 s avec cette valeur.

Les résultats du paramétrage final sont les suivants : $\alpha = 0,3$, $\Delta_{Dev} = 0,1\%$, $\delta = 0,1$, $\sigma_1 = 50$, $\theta = 0,5$, $\phi = 0,2$, $p_u = 70$, $l = 0,7$ et $\gamma = 11$. Les valeurs des paramètres sont utilisées dans la suite pour nos expériences sur toutes les instances de référence. Elles ont été choisies pour obtenir un bon compromis entre la qualité de la solution et le temps de traitement.

4.4.4 Évaluation des composantes de l'ALNS

Nous évaluons l'efficacité des opérateurs de destruction/réparation et du mécanisme adaptatif de destruction/réparation en désactivant chacun d'entre eux un par un. Cela nous permet de discuter les résultats pour montrer si chaque composant contribue à l'obtention de bonnes solutions en considérant également le temps de traitement.

Dans la Table 4.3, nous montrons les résultats pour les quatre opérateurs de destruction (voir section 4.3.2.2). Sous chaque rubrique, nous indiquons les valeurs minimales (colonne *Min*) et les temps de traitement moyens en secondes (colonne *t(s)*) que nous obtenons en 10 exécutions. À des fins de comparaison, nous présentons sous le titre *ALNS* les valeurs que nous obtenons en exécutant l'ALNS avec tous les composants activés (également dans le tableau 4.4).

	ALNS		Sans DAP		Sans DCP		Sans DAJ		Sans DCJ	
	Min	t(s)	Min	t(s)	Min	t(s)	Min	t(s)	Min	t(s)
a_01	20.64	116.25	20.89	85.09	20.69	193.74	20.65	168.87	20.69	183.92
a_02	19.41	119.86	19.59	109.17	19.50	132.04	19.44	120.89	19.50	159.37
a_03	22.13	117.25	22.58	120.99	22.21	196.98	22.22	190.22	22.18	226.77
a_04	20.37	129.81	20.49	94.58	20.46	139.49	20.37	135.38	20.43	264.55
a_05	23.77	1109.92	6027.92	301.24	24.00	4045.17	24.05	4624.39	24.58	3268.68
a_06	22.30	72.59	22.44	53.26	22.34	81.75	22.39	161.33	22.39	100.86
a_07	21.44	93.54	21.56	107.37	21.50	113.31	21.44	123.28	21.45	182.66
a_08	22.59	548.21	3025.79	147.91	22.61	1980.98	22.85	1655.09	22.74	1560.71
a_09	20.52	77.80	20.57	69.03	20.55	95.83	20.57	118.73	20.52	137.31
a_10	20.33	78.33	20.36	122.60	20.35	117.41	21.82	109.47	20.34	131.32

Tableau 4.3: Impact des opérateurs de destruction RR, WR, RDR et WDR.

L'ALNS sans Destruction Aléatoire Patient (la colonne *Sans DAP*) obtient des résultats inférieurs à ceux de l'ALNS en utilisant tous les composants. Le **DAP** opérateur de destruction est nécessaire. Les interventions des patients sont choisies au aléatoirement ce qui permet d'éviter de rester bloqué dans un optimum local. C'est le cas pour les instances a_05 et a_08 pour lesquelles nous ne pouvons pas planifier toutes les interventions si **DAP** est désactivé. Les temps de traitement sans **DAP** sont plus courts mais nous observons expérimentalement que l'ALNS sans **DAP** s'arrête car nous effectuons $Iter_{max}$ itérations sans aucune amélioration.

L'ALNS sans Destruction Coût Patient (la colonne *Sans DCP*) obtient des résultats inférieurs à ceux d'ALNS en utilisant tous les composants. Pour toutes les instances, toutes les interventions peuvent être programmées mais au prix de temps de traitement plus importants. Le **DCP** opérateur de destruction est nécessaire car il vise à minimiser la fonction objectif.

Le Destruction Aléatoire d'une Journée (DAJ) et Destruction de Coût Journée (DCJ) sont des opérateurs de destruction orientés vers la destruction d'une journée entière, ils permettent la réallocation des ressources d'une journée. L'ALNS sans Destruction Aléatoire Journée (la colonne *Sans DAJ*) obtient des résultats inférieurs à ceux de l'ALNS en utilisant tous les composants, à l'exception des instances a_04 et a_07 pour lesquelles nous obtenons les mêmes résultats. Sans l'opérateur **DAJ** toutes les interventions peuvent être programmées mais avec un temps de traitement supérieur.

L'ALNS sans Destruction Coût Journée (la colonne *Sans DCJ*) obtient des résultats qui sont inférieurs à ceux d'ALNS en utilisant toutes les composantes, sauf pour l'instance a_09 pour laquelle nous obtenons le même meilleur résultat mais avec un temps de traitement supérieur. Pour toutes les instances, toutes les interventions peuvent être planifiées mais au prix de temps de traitement plus importants, d'où la nécessité du **DCJ**.

Pour résumer, les quatre opérateurs de destruction sont nécessaires. Ils sont bénéfiques pour explorer le voisinage d'une solution, soit en obtenant de meilleurs résultats, soit en raccourcissant le temps de traitement.

	ALNS		Sans INP		Sans NMD		Sans AC		Sans Adaptif	
	Min	t(s)	Min	t(s)	Min	t(s)	Min	t(s)	Min	t(s)
a_01	20.64	116.25	20.70	226.27	20.76	141.14	20.75	149.44	20.85	238.43
a_02	19.41	119.86	19.42	259.44	19.49	117.86	19.45	120.27	19.59	240.48
a_03	22.13	117.25	22.43	363.59	22.18	256.03	22.21	284.99	22.27	216.09
a_04	20.37	129.81	20.41	187.91	20.38	144.80	20.43	161.11	20.37	532.06
a_05	23.77	1109.92	24.35	3777.95	24.28	3680.23	24.55	4328.10	28.37	5999.34
a_06	22.30	72.59	22.39	188.19	22.39	71.11	22.39	119.32	22.42	92.27
a_07	21.44	93.54	21.52	188.27	21.52	133.41	21.52	182.70	21.57	163.06
a_08	22.59	548.21	22.73	1681.28	23.34	1733.85	22.83	1739.60	25.61	2310.89
a_09	20.52	77.80	20.56	171.90	20.60	79.87	20.53	125.69	20.55	230.76
a_10	20.33	78.33	20.34	188.63	20.35	115.90	20.37	104.35	20.35	236.90

Tableau 4.4: Impact des opérateurs de réparation INP, NMD et AC, et impact du mécanisme d'adaptation.

Dans la Table 4.4, nous montrons les résultats pour les trois opérateurs de réparation (voir Section 4.3.2.3) et pour le mécanisme adaptatif (voir section 4.3.2.4) qui gère le choix des opérateurs de destruction et de réparation.

Globalement, pour les trois opérateurs de réparation, les Interventions Non planifiées en Premiers (INP), Nombre de Modes Demandés (NMD) et Agrégation de Critères (AC), nous observons d'abord qu'aucun n'a réussi à atteindre les résultats de l'approche ALNS en utilisant tous les composants, et, ensuite, que les temps de traitement sont généralement plus importants. C'est notamment le cas pour les instances a_05 et a_08.

En résumé, les trois opérateurs de réparation sont nécessaires, et sont bénéfiques pour explorer le voisinage d'une solution, soit en obtenant de meilleurs résultats, soit en réduisant le temps de traitement.

La dernière expérience a été menée pour mettre en évidence l'efficacité du mécanisme adaptatif. Nous avons désactivé ce mécanisme en fixant la même probabilité pour chaque opérateur de destruction ($1/4$) et la même probabilité pour chaque opérateur de réparation ($1/3$).

Comme on peut le voir dans la colonne *Sans Adaptif*, nous obtenons des résultats qui sont inférieurs à ceux de l'ALNS qui utilise ce composant, et les temps de traitement sont plus importants. Ceci est particulièrement évident pour les instances a_05 et a_08 pour lesquelles, sans ce mécanisme adaptatif, nous obtenons les pires temps de traitement de cette série d'expériences.

Le mécanisme adaptatif qui gère le choix des opérateurs de destruction et de réparation au cours de l'ALNS est nécessaire pour obtenir de bons résultats dans un bon temps de traitement pour le problème de planification d'admission de patients non urgents que nous traitons ici.

4.4.5 Comparaison de l'ALNS avec la littérature

Dans la Table 4.5, pour chaque instance sous la rubrique *ACI*, nous indiquons les résultats obtenus par *ACI* (voir [Riise et al., 2016]). La colonne *UB* indique la meilleure valeur connue, ou borne supérieure, et les colonnes *Min*, *Max*, *Avg* indiquent la valeur minimale, la valeur maximale et la valeur moyenne. Sous la rubrique *ALNS*, nous indiquons également les valeurs *Min*, *Max* et *Avg* que nous avons obtenues. La colonne *t(s)* indique les temps de traitement moyens en secondes. Nous indiquons en caractères gras les résultats qui sont meilleurs que ceux de l'UB.

	ACI							ALNS						
	UB	Min	Max	Avg	MizAv%	MizUB%	t(s)	Min	Max	Avg	MizAv%	MizUB%	MizMi%	t(s)
a_01 *	21.58	21.85	22.23	22.06	0.95	-1.25	600	20.64	20.94	20.77	0.63	4.36	5.54	116.25
a_02 *	22.83	23.25	23.56	23.4	0.64	-1.84	600	19.41	19.60	19.53	0.61	14.98	16.52	119.86
a_03 *	24.52	24.58	24.88	24.74	0.65	-0.24	600	22.13	22.49	22.28	0.67	9.75	9.97	117.25
a_04 *	21.97	22.28	22.66	22.48	0.89	-1.41	600	20.37	20.48	20.45	0.39	7.28	8.57	129.81
a_05	5113.88	6341.85	9397.78	7987.31	20.60	-24.01	600	23.77	4024.46	1425.0	98.33	99.54	99.63	1109.92
a_06 *	23.56	23.56	24.03	23.82	1.09	0.00	600	22.30	22.44	22.41	0.49	5.35	5.35	72.59
a_07 *	22.73	23.26	23.57	23.39	0.56	-2.33	600	21.44	21.62	21.54	0.46	5.68	7.82	93.54
a_08	2086.32	3148.49	4244.57	3407.31	7.60	-50.91	600	22.59	4023.4	423.21	94.66	98.92	99.28	548.21
a_09 *	24.83	24.99	25.98	25.37	1.50	-0.64	600	20.52	20.64	20.60	0.39	17.36	17.89	77.80
a_10 *	24.47	24.75	25.1	24.9	0.60	-1.14	600	20.33	20.42	20.36	0.15	16.92	17.86	78.33
Pour '*'					0.86	-1.11					0.47	10.21	11.19	100.67

Tableau 4.5: ALNS par rapport à ACI

À des fins de comparaison, nous utilisons les valeurs $(\text{Min} - \text{Avg})/\text{Avg}$ désignées dans les en-têtes de colonne par *MizAv* et $(\text{UB} - \text{Min})/\text{UB}$ désignées dans les en-têtes de colonne par *MizUB*. La colonne *MizMi* indique les valeurs $(\text{Min}(\text{ACI}) - \text{Min}(\text{ALNS}))/\text{Min}(\text{ACI})$.

La première nous permet de montrer à quel point les résultats moyens de l'ACI ou de l'ALNS (pour dix exécutions) sont proches du minimum pour chaque instance, tandis que la seconde montre à quel point les résultats sont éloignés de la meilleure valeur connue de l'UB. La première n'est vraiment significative que pour les instances pour lesquelles toutes les interventions peuvent être planifiées. Nous rappelons que la fonction objectif (voir équation (4.1a)-(4.1c)) est une somme pondérée de termes normalisés. Le poids W_{un} utilisé pour le terme qui évalue les interventions non planifiées est le plus grand par rapport aux autres.

Cela montre clairement que les interventions non planifiées sont un problème très majeur. Il n'est pas équitable de comparer les résultats des instances pour lesquelles toutes les interventions sont planifiées à celles pour lesquelles certaines sont laissées non planifiées parce que le terme le plus élevé est nul dans le premier cas.

Nous avons ajouté une '*' aux noms des instances pour lesquelles l'ACI et l'ALNS ont réussi à planifier toutes les interventions. Pour ces instances, la dernière ligne indique alors les valeurs moyennes de *MizAv*, *MizUB* et *MizMi*, ainsi que la valeur moyenne du temps de traitement de l'ALNS.

Pour les huit instances pour lesquelles l'ACI et l'ALNS ont réussi à planifier toutes les interventions, l'ALNS est plus performant. Le *MizAv* moyen est presque réduit de

moitié, le *MizUB* est amélioré d'environ 10,2%, et le *MizMi* est amélioré d'environ 11,2%. Les temps de traitement sont plus courts que le délai fixé dans [Riise et al., 2016].

Pour les deux instances pour lesquelles l'approche ACI n'a pas réussi à planifier toutes les interventions, l'ALNS a réussi. Pour l'instance *a_08*, cela prend environ cinq minutes et, pour l'instance *a_05*, cela prend environ 20 minutes. Des résultats strictement meilleurs sont obtenus pour toutes les instances, toutes les interventions des patients sont planifiées, ce qui n'était pas le cas avec l'approche ACI.

4.5 CONCLUSION ET PERSPECTIVES

Dans ce chapitre, nous avons abordé le problème d'admission des patients. Il s'agit de planifier dans un horizon de plusieurs mois des interventions chirurgicales de centaines de patients doivent être planifiées. L'objectif premier des planificateurs hospitaliers est de planifier toutes les interventions en tenant compte de la disponibilité des salles d'opérations et des chirurgiens, mais aussi des contraintes liées à l'intervention chirurgicale de chaque patient.

Nous avons introduit une heuristique 2PSC-EM à deux phases et une approche ALNS (Adaptive Large Neighborhood Search). L'ALNS utilise des mécanismes adaptatifs pour la destruction, la construction et l'acceptation de solutions. Un bon compromis entre un mécanisme adaptatif de diversification et un mécanisme adaptatif d'acceptation est nécessaire afin de ne pas perdre de temps de traitement en explorant inutilement des solutions de faible qualité.

Des expériences numériques ont été menées sur des instances de référence issues de la littérature. Les résultats montrent que l'heuristique 2PSC-EM a obtenu de meilleurs résultats qu'une approche heuristique générale non dédiée au problème d'admission. Cependant, les deux heuristiques n'ont pas réussi à planifier toutes les interventions pour certaines instances. Les expériences pour le réglage des paramètres de l'approche ALNS ont été menées pour obtenir un bon compromis entre la qualité de la solution et le temps de traitement. L'évaluation de l'efficacité de chaque composant de l'approche ALNS montre que tous les composants sont nécessaires pour obtenir de bonnes solutions dans de bons délais de traitement. L'approche ALNS a obtenu des résultats strictement meilleurs pour chaque instance de la littérature, toutes les interventions sont programmées et que les contraintes des patients sont mieux respectées.

Une perspective de travail future serait d'envisager des objectifs et des contraintes supplémentaires sur les charges de travail minimales et/ou maximales des chirurgiens ou sur l'utilisation des salles d'opération afin d'obtenir des plannings équilibrés en ce qui concerne l'utilisation des ressources. L'équité entre les chirurgiens est une question sensible pour une organisation hospitalière. L'équilibrage de la charge des salles permettrait une meilleure utilisation de ces ressources. D'autres approches basées sur d'autres métaheuristiques ou en intégrant des méthodes d'apprentissage comme celles présentées dans [Karimi-Mamaghan et al., 2022], [Bengio et al., 2021] et [Queiroz dos Santos et al., 2014] serait à explorer en vue d'améliorer les résultats et d'intégrer des critères additionnels.

CONCLUSION GÉNÉRALE PERSPECTIVES

Les services des urgences constituent un maillon essentiel de la chaîne de soins dans les systèmes de santé modernes. L'objectif principal est d'offrir une prise en charge médicale rapide et efficace pour les patients dont l'état de santé nécessite un traitement médical urgent. Les patients qui subissent des temps d'attente trop longs dans les services des urgences présentent un risque accru de mortalité, une durée de séjour plus longue à l'hôpital et une moins bonne satisfaction à la suite du traitement. La complexité de la gestion des urgences est causée par le flux imprévisible des arrivées de patients. Un certain nombre de mesures de performance, par exemple les temps d'attente, la qualité des soins, la durée du traitement, la durée du séjour, peuvent se dégrader lorsqu'il y a des tensions dans le service des urgences. Les tensions induisent des difficultés pour les patients et le personnel médical. Une meilleure utilisation des ressources permettrait d'améliorer le fonctionnement des services des urgences.

Dans le cadre de l'ANR OIILH, nous avons étudié et proposé des solutions pour le problème d'ordonnancement des tâches de parcours de soins des patients dans le service des urgences du CHU Jeanne de Flandres de Lille. Nous avons de plus étudié et proposé des solutions pour la planification d'interventions chirurgicales. En effet, le service des urgences génère des flux de patients pour lesquels des interventions chirurgicales sont à prévoir (aval).

Pour le problème des urgences, nous avons étudié en premier lieu le cas statique. Dans ce cas, nous supposons que les informations sur les patients et leurs parcours sont connus au début de planification. Nous avons réalisé un générateur d'instances basé sur les données réelles du service d'urgence du CHU Jeanne de Flandres de Lille. À partir des observations sur le terrain, nous avons implémenté une approche heuristique constructive (PRH) permettant de gérer les ressources dans le service des urgences en suivant deux règles métier utilisées par les praticiens. Nous avons proposé une modélisation mathématique (Programmation Linéaire en Nombres Entiers PLNE) du problème. Le modèle a été implémenté et utilisé pour obtenir des solutions de références pour les instances générées. Le passage à une résolution approchée du problème était nécessaire car la résolution exacte du PLNE ne trouve pas de solution réalisable en un temps raisonnable pour certaines instances. Nous avons proposé deux approches heuristiques.

Nous avons élaboré une approche heuristique itérative basée sur la destruction et la construction d'une solution *Iterative Destruction Construction* (IDCH). Les résultats montrent que des solutions de meilleure qualité par rapport à l'heuristique constructive PRH peuvent être obtenues en des temps raisonnables. Des solutions optimales peuvent être

trouvées pour les petites instances. Ensuite, nous avons élaboré une méta-heuristique *Adaptive Large Neighborhood Search* (ALNS) plus efficace. Nous avons proposé plusieurs voisinages de destruction et de construction et expérimenté deux processus adaptatifs d'acceptation de solutions non améliorantes (probabiliste et déterministe). Les résultats expérimentaux montrent l'efficacité de cette approche avec des temps de calcul faibles qui sont adéquats par rapport à la réactivité requise dans le cas dynamique du problème.

Le cas dynamique du problème a également été étudié. Ce n'est qu'une fois que les patients sont arrivés qu'il est possible de connaître leurs besoins spécifiques et les différentes possibilités de traitement qui leur conviennent. Nous avons proposé une approche de planification prédictive et réactive. La méthode proposée consiste à élaborer un planning sans tenir compte des perturbations dynamiques potentielles, puis à ajuster le planning en fonction de la survenance des événements. Nous construisons, modifions et améliorons le planning dans le cas dynamique en utilisant les méthodes de résolution du cas statique. La nature imprévisible des arrivées des patients, des demandes de traitement et des retards de traitement est prise en compte dans l'algorithme dynamique. Des événements tels que l'arrivée d'un patient et la fin d'une tâche de traitement entraînent une planification réactive. L'approche proposée est capable de gérer la dynamique du service d'urgence.






Le service des urgences génère potentiellement des flux de patients pour d'autres services de l'hôpital. Une visite aux urgences peut révéler qu'un patient a besoin d'une intervention chirurgicale à programmer plus tard. Cela crée des patients en chirurgie élective. Pour le service de chirurgie, des centaines d'interventions chirurgicales non urgentes et non immédiates doivent être programmées des mois ou des semaines à l'avance. Le niveau d'urgence du patient est pris en compte à l'aide de dates (au plus tôt, au plus tard). Nous avons développé deux approches de résolution pour ce problème en aval du service des urgences. Nous avons proposé une heuristique 2PSC-CHI en deux phases basée sur une heuristique constructive utilisée pour le RCPSP (Resource Constrained Project Scheduling) et un opérateur de voisinage local. Cette heuristique, comme celles proposées dans la littérature, ne pouvait pas planifier toutes les interventions chirurgicales. Nous avons proposé une approche ALNS qui donne des résultats strictement meilleurs que ceux de la littérature. Elle place toutes les interventions de l'ensemble du jeu test de référence.

Cette étude a montré que ces méthodes permettraient d'améliorer la planification et l'ordonnancement au sein des systèmes hospitaliers. Elle montre également l'interaction possible entre les services des urgences et les services en aval. Le problème de planification des admissions peut être utilisé comme outil d'aide à la décision lorsqu'une intervention à réaliser est détectée par le service des urgences. Il serait possible de déterminer si le planning chirurgical actuel permettra de prendre en charge le futur patient électif ou de l'orienter vers un autre hôpital.

Sur la base de ce travail, nous pourrions étendre la problématique. Un problème plus large pourrait être envisagé en considérant une capacité non standard telle que les couloirs qui peuvent exceptionnellement être utilisés pendant une courte période pour accueillir quelques patients. Une capacité supplémentaire peut être aussi mobilisée pour faire face à une situation de tension sur une plus longue période, comme dans le cas de la pandémie de Covid. Équilibrer l'utilisation des ressources, les charges

de travail minimales et maximales des différents praticiens et l'utilisation des salles d'opération pourraient être pris en compte à l'avenir (critères d'équité et d'équilibrage). D'autres approches basées sur des méta-heuristiques différentes ou intégrant des méthodes d'apprentissage, telles que celles présentées dans [Karimi-Mamaghan et al., 2022], [Bengio et al., 2021] et [Queiroz dos Santos et al., 2014] pourraient être étudiées à des fins de comparaison en vue d'améliorer les résultats et d'intégrer des critères supplémentaires. L'expérimentation in situ de l'algorithme dynamique dans un système d'aide à la décision en temps réel permettra d'avoir des retours des praticiens afin d'y apporter des améliorations.

PARCOURS DES PATIENTS

-  Attente dans un lieu dédié externe au SUA
-  Attente dans un lieu dédié interne au SUA
-  Traitement dans un lieu dédié interne au SUA
-  Traitement dans un lieu dédié externe au SUA
-  Déplacement du patient

Code couleur

A.1 PARCOURS DIVERS/CONSULTATION GÉNÉRALE

A.1.1 *Sans examens*

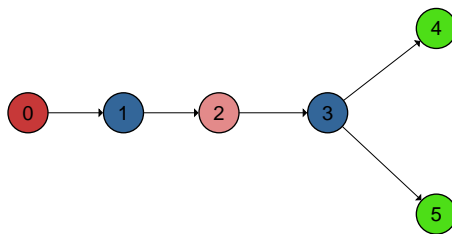


Fig A.1: Divers sans examens : Graphe de tâches

N°	Nom	Lieu	Ressource	Durée
0	Attente avant entrée CC	Externe	∅	?
1	Entrée CC	Transport	Infirmière Aide-soignante	5
2	Attente avant consultation	Salle d'attente	∅	?
3	Déplacement vers consultation	Transport	Infirmière Aide-soignante	5
4	Consultation médicale	Box	Médecin urgentiste Interne médecine	15
5	Prise de signes vitaux	Box	Infirmière	5

Tableau A.1: Divers sans examens : description des tâches

A.1.2 Avec examen biologique uniquement

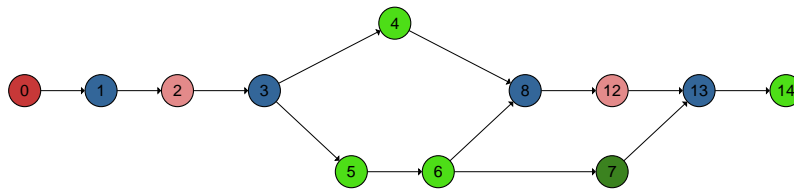


Fig A.2: Divers avec uniquement examen biologique : Graphe de tâches

N°	Nom	Lieu	Ressource	Durée
0	Attente avant entrée CC	Externe	∅	?
1	Entrée CC	Transport	Infirmière Aide-soignante	5
2	Attente avant consultation	Salle d'attente	∅	?
3	Déplacement vers consultation	Transport	Infirmière Aide-soignante	5
4	Consultation médicale	Box	Médecin urgentiste Interne médecine	15
5	Prise de signes vitaux	Box	Infirmière	5
6	Prise de sang	Box	Infirmière	5
7	Analyse biologique	Externe	Externe	Donnée
8	Déplacement vers lieu d'attente	Transport	Infirmière Aide-soignante	5
12	Attente avant évaluation	Salle d'attente	∅	?
13	Déplacement vers évaluation	Transport	Infirmière Aide-soignante	5
14	Évaluation médicale	Box	Médecin urgentiste Interne médecine	15

Tableau A.2: Divers avec uniquement examen biologique : Description des tâches

A.1.3 Avec autres examens, mais sans examen biologique

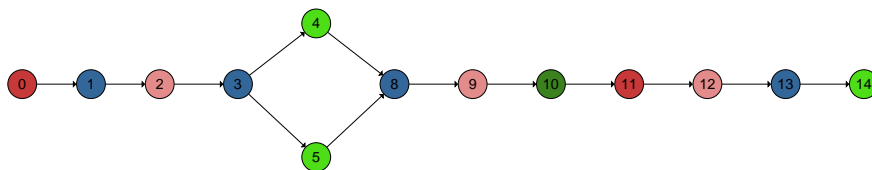


Fig A.3: Divers avec examens sans examen biologique : Graphe de tâches

N°	Nom	Lieu	Ressource	Durée
0	Attente avant entrée CC	Externe	∅	?
1	Entrée CC	Transport	Infirmière Aide-soignante	5
2	Attente avant consultation	Salle d'attente	∅	?
3	Déplacement vers consultation	Transport	Infirmière Aide-soignante	5
4	Consultation médicale	Box	Médecin urgentiste Interne médecine	15
5	Prise de signes vitaux	Box	Infirmière	5
8	Déplacement vers lieu d'attente	Transport	Infirmière Aide-soignante	5
9	Attente avant examen(s)	Salle d'attente	∅	?
10	Examens	Externe	Externe	Donnée
11	Attente avant retour CC	Salle d'attente	∅	?
12	Attente avant évaluation	Salle d'attente	∅	?
13	Déplacement vers évaluation	Transport	Infirmière Aide-soignante	5
14	Évaluation médicale	Box	Médecin urgentiste Interne médecine	15

Tableau A.3: Divers avec examens sans examen biologique : Description des tâches

A.1.4 Avec examens (biologique + autres)

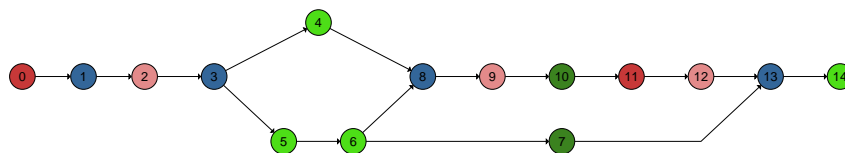


Fig A.4: Divers avec examens dont examen biologique : Graphe de tâches

N°	Nom	Lieu	Ressource	Durée
0	Attente avant entrée CC	Externe	∅	?
1	Entrée CC	Transport	Infirmière Aide-soignante	5
2	Attente avant consultation	Salle d'attente	∅	?
3	Déplacement vers consultation	Transport	Infirmière Aide-soignante	5
4	Consultation médicale	Box	Médecin urgentiste Interne médecine	15
5	Prise de signes vitaux	Box	Infirmière	5
6	Prise de sang	Box	Infirmière	5
7	Analyse biologique	Externe	Externe	Donnée
8	Déplacement vers lieu d'attente	Transport	Infirmière Aide-soignante	5
9	Attente avant examen(s)	Salle d'attente	∅	?
10	Examens	Externe	Externe	Donnée
11	Attente avant retour CC	Salle d'attente	∅	?
12	Attente avant évaluation	Salle d'attente	∅	?
13	Déplacement vers évaluation	Transport	Infirmière Aide-soignante	5
14	Évaluation médicale	Box	Médecin urgentiste Interne médecine	15

Tableau A.4: Divers avec examens dont examen biologique : Description des tâches

A.2 PARCOURS ORL

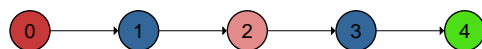
A.2.1 *Sans examens*

Fig A.5: ORL sans examens : Graphe de tâches

N°	Nom	Lieu	Ressource	Durée
0	Attente avant entrée CC	Externe	∅	?
1	Entrée CC	Transport	Infirmière Aide-soignante	5
2	Attente avant consultation	Salle d'attente	∅	?
3	Déplacement vers consultation	Transport	Infirmière Aide-soignante	5
4	Consultation médicale	Box ORL	Interne ORL	15

Tableau A.5: ORL sans examens : Description des tâches

A.2.2 Avec examen biologique uniquement

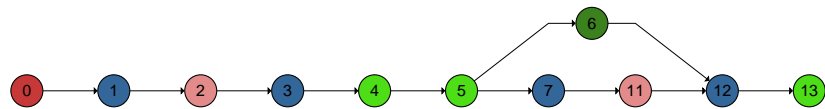


Fig A.6: ORL avec uniquement examen biologique : Graphe de tâches

N°	Nom	Lieu	Ressource	Durée
0	Attente avant entrée CC	Externe	∅	?
1	Entrée CC	Transport	Infirmière Aide-soignante	5
2	Attente avant consultation	Salle d'attente	∅	?
3	Déplacement vers consultation	Transport	Infirmière Aide-soignante	5
4	Consultation médicale	Box ORL	Interne ORL	15
5	Prise de sang	Box ORL	Infirmière	5
6	Analyse biologique	Externe	Externe	Donnée
7	Déplacement vers lieu d'attente	Transport	Infirmière Aide-soignante	5
8	Attente avant examen(s)	Salle d'attente	∅	?
9	Examens	Externe	Externe	Donnée
10	Attente avant retour CC	Salle d'attente	∅	?
11	Attente avant évaluation	Salle d'attente	∅	?
12	Déplacement vers évaluation	Transport	Infirmière Aide-soignante	5
13	Évaluation médicale	Box ORL	Interne ORL	15

Tableau A.6: ORL avec uniquement examen biologique : Description des tâches

A.2.3 Avec autres examens, mais sans examen biologique



Fig A.7: ORL avec examens sans examen biologique : Graphe de tâches

N°	Nom	Lieu	Ressource	Durée
0	Attente avant entrée CC	Externe	∅	?
1	Entrée CC	Transport	Infirmière Aide-soignante	5
2	Attente avant consultation	Salle d'attente	∅	?
3	Déplacement vers consultation	Transport	Infirmière Aide-soignante	5
4	Consultation médicale	Box ORL	Interne ORL	15
7	Déplacement vers lieu d'attente	Transport	Infirmière Aide-soignante	5
11	Attente avant évaluation	Salle d'attente	∅	?
12	Déplacement vers évaluation	Transport	Infirmière Aide-soignante	5
13	Évaluation médicale	Box ORL	Interne ORL	15

Tableau A.7: ORL avec examens sans examen biologique : Description des tâches

A.2.4 Avec examens (biologique + autres)

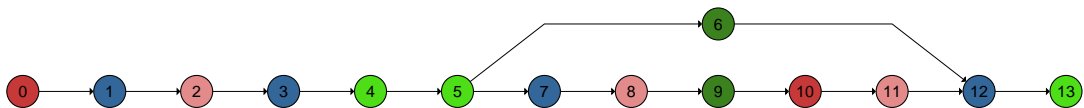


Fig A.8: ORL avec examens dont examen biologique : Graphe de tâches

N°	Nom	Lieu	Ressource	Durée
0	Attente avant entrée CC	Externe	∅	?
1	Entrée CC	Transport	Infirmière Aide-soignante	5
2	Attente avant consultation	Salle d'attente	∅	?
3	Déplacement vers consultation	Transport	Infirmière Aide-soignante	5
4	Consultation médicale	Box ORL	Interne ORL	15
5	Prise de sang	Box ORL	Infirmière	5
6	Analyse biologique	Externe	Externe	Donnée
7	Déplacement vers lieu d'attente	Transport	Infirmière Aide-soignante	5
11	Attente avant évaluation	Salle d'attente	∅	?
12	Déplacement vers évaluation	Transport	Infirmière Aide-soignante	5
13	Évaluation médicale	Box ORL	Interne ORL	15

Tableau A.8: ORL avec examens dont examen biologique : Description des tâches

A.3 PARCOURS STOMATOLOGIE

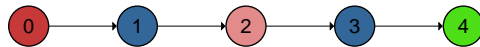
A.3.1 *Sans examens*

Fig A.9: Stomatologie sans examens : Graphe de tâches

N°	Nom	Lieu	Ressource	Durée
0	Attente avant entrée CC	Externe	∅	?
1	Entrée CC	Transport	Infirmière Aide-soignante	5
2	Attente avant consultation	Salle d'attente	∅	?
3	Déplacement vers consultation	Transport	Infirmière Aide-soignante	5
4	Consultation médicale	Box	Médecin urgentiste Interne Stomato	5

Tableau A.9: Stomatologie sans examens : Description des tâches

A.3.2 Avec examen biologique uniquement

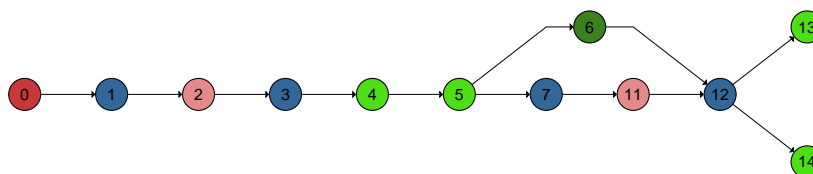


Fig A.10: Stomatologie avec uniquement examen biologique : Graphe de tâches

N°	Nom	Lieu	Ressource	Durée
0	Attente avant entrée CC	Externe	∅	?
1	Entrée CC	Transport	Infirmière Aide-soignante	5
2	Attente avant consultation	Salle d'attente	∅	?
3	Déplacement vers consultation	Transport	Infirmière Aide-soignante	5
4	Consultation médicale	Box	Médecin urgentiste Interne Stomato	5
5	Prise de sang	Box	Infirmière	5
6	Analyse biologique	Externe	Externe	Donnée
7	Déplacement vers lieu d'attente	Transport	Infirmière Aide-soignante	5
11	Attente avant évaluation	Salle d'attente	∅	?
12	Déplacement vers évaluation	Transport	Infirmière Aide-soignante	5
13	Intervention Stomato	Salle Stomato	Interne Stomato	15
14	Soutien Stomato	Salle Stomato	Infirmière	15

Tableau A.10: Stomatologie avec uniquement examen biologique : Description des tâches

A.3.3 Avec autres examens, mais sans examen biologique

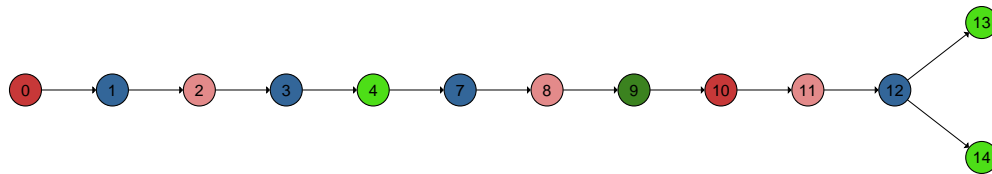


Fig A.11: Stomatologie avec examens sans examen biologique : Graphe de tâches

N°	Nom	Lieu	Ressource	Durée
0	Attente avant entrée CC	Externe	∅	?
1	Entrée CC	Transport	Infirmière Aide-soignante	5
2	Attente avant consultation	Salle d'attente	∅	?
3	Déplacement vers consultation	Transport	Infirmière Aide-soignante	5
4	Consultation médicale	Box	Médecin urgentiste Interne Stomato	5
7	Déplacement vers lieu d'attente	Transport	Infirmière Aide-soignante	5
8	Attente avant examen(s)	Salle d'attente	∅	?
9	Examens	Externe	Externe	Donnée
10	Attente avant retour CC	Salle d'attente	∅	?
11	Attente avant évaluation	Salle d'attente	∅	?
12	Déplacement vers évaluation	Transport	Infirmière Aide-soignante	5
13	Intervention Stomato	Salle Stomato	Interne Stomato	15
14	Soutien Stomato	Salle Stomato	Infirmière	15

Tableau A.11: Stomatologie avec examens sans examen biologique : Description des tâches

A.3.4 Avec examens (biologique + autres)

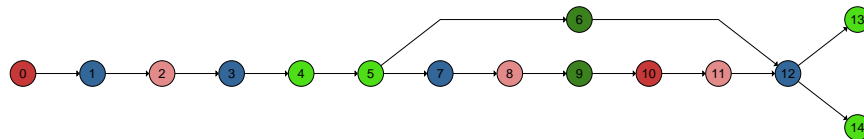


Fig A.12: Stomatologie avec examens dont examen biologique : Graphe de tâches

N°	Nom	Lieu	Ressource	Durée
0	Attente avant entrée CC	Externe	∅	?
1	Entrée CC	Transport	Infirmière Aide-soignante	5
2	Attente avant consultation	Salle d'attente	∅	?
3	Déplacement vers consultation	Transport	Infirmière Aide-soignante	5
4	Consultation médicale	Box	Médecin urgentiste Interne Stomato	5
5	Prise de sang	Box	Infirmière	5
6	Analyse biologique	Externe	Externe	Donnée
7	Déplacement vers lieu d'attente	Transport	Infirmière Aide-soignante	5
8	Attente avant examen(s)	Salle d'attente	∅	?
9	Examens	Externe	Externe	Donnée
10	Attente avant retour CC	Salle d'attente	∅	?
11	Attente avant évaluation	Salle d'attente	∅	?
12	Déplacement vers évaluation	Transport	Infirmière Aide-soignante	5
13	Intervention Stomato	Salle Stomato	Interne Stomato	15
14	Soutien Stomato	Salle Stomato	Infirmière	15

Tableau A.12: Stomatologie avec examens dont examen biologique : Description des tâches

A.4 PARCOURS TRAUMATOLOGIE

On considère pour la suite que si des examens sont prescrits alors l'examen radiologie y figure. Selon son "résultat" le patient peut passer ou non en salle de plâtre.

A.4.1 Sans examens

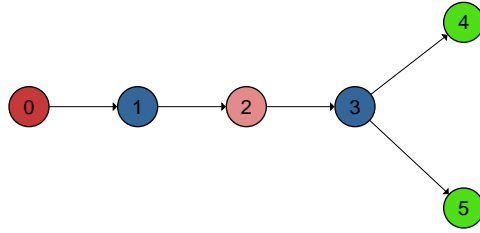


Fig A.13: Traumatologie sans examens : Graphe de tâches

N°	Nom	Lieu	Ressource	Durée
0	Attente avant entrée CC	Externe	∅	?
1	Entrée CC	Transport	Infirmière Aide-soignante	5
2	Attente avant consultation	Salle d'attente	∅	?
3	Déplacement vers consultation	Transport	Infirmière Aide-soignante	5
4	Consultation médicale	Box	Médecin urgentiste Interne Traumato	15
5	Prise de signes vitaux	Box	Infirmière	5

Tableau A.13: Traumatologie sans examens : Description des tâches

A.4.2 Sans plâtre

A.4.2.1 Avec examens (sans examen biologique)

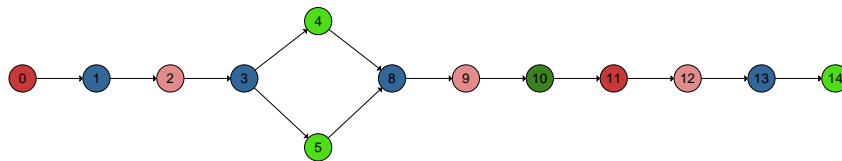


Fig A.14: Traumatologie avec examens sans examen biologique sans plâtre : Graphe de tâches

N°	Nom	Lieu	Ressource	Durée
0	Attente avant entrée CC	Externe	∅	?
1	Entrée CC	Transport	Infirmière Aide-soignante	5
2	Attente avant consultation	Salle d'attente	∅	?
3	Déplacement vers consultation	Transport	Infirmière Aide-soignante	5
4	Consultation médicale	Box	Médecin urgentiste Interne Traumato	15
5	Prise de signes vitaux	Box	Infirmière	5
8	Déplacement vers lieu d'attente	Transport	Infirmière Aide-soignante	5
9	Attente avant examen(s)	Salle d'attente	∅	?
10	Examens	Externe	Externe	Donnée
11	Attente avant retour CC	Salle d'attente	∅	?
12	Attente avant évaluation	Salle d'attente	∅	?
13	Déplacement vers évaluation	Transport	Infirmière Aide-soignante	5
14	Évaluation médicale	Box	Médecin urgentiste Interne médecine	15

Tableau A.14: Traumatologie avec examens sans examen biologique sans plâtre : Description des tâches

A.4.2.2 Avec examens (biologique + autres)

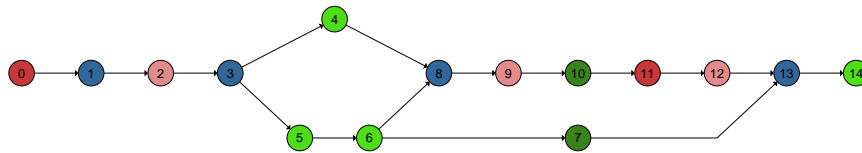


Fig A.15: Traumatologie avec examens dont examen biologique sans plâtre : Graphe de tâches

N°	Nom	Lieu	Ressource	Durée
0	Attente avant entrée CC	Externe	∅	?
1	Entrée CC	Transport	Infirmière Aide-soignante	5
2	Attente avant consultation	Salle d'attente	∅	?
3	Déplacement vers consultation	Transport	Infirmière Aide-soignante	5
4	Consultation médicale	Box	Médecin urgentiste Interne Traumato	15
5	Prise de signes vitaux	Box	Infirmière	5
6	Prise de sang	Box	Infirmière	5
7	Analyse biologique	Externe	Externe	Donnée
8	Déplacement vers lieu d'attente	Transport	Infirmière Aide-soignante	5
9	Attente avant examen(s)	Salle d'attente	∅	?
10	Examens	Externe	Externe	Donnée
11	Attente avant retour CC	Salle d'attente	∅	?
12	Attente avant évaluation	Salle d'attente	∅	?
13	Déplacement vers évaluation	Transport	Infirmière Aide-soignante	5
14	Évaluation médicale	Box	Médecin urgentiste Interne médecine	15

Tableau A.15: Traumatologie avec examens dont examen biologique sans plâtre : Description des tâches

A.4.3 Avec plâtre

A.4.3.1 Avec examens (sans examen biologique)

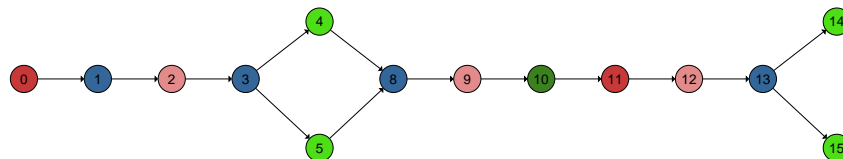


Fig A.16: Traumatologie avec examens sans examen biologique : Graphe de tâches

N°	Nom	Lieu	Ressource	Durée
0	Attente avant entrée CC	Externe	∅	?
1	Entrée CC	Transport	Infirmière Aide-soignante	5
2	Attente avant consultation	Salle d'attente	∅	?
3	Déplacement vers consultation	Transport	Infirmière Aide-soignante	5
4	Consultation médicale	Box	Médecin urgentiste Interne Traumato	15
5	Prise de signes vitaux	Box	Infirmière	5
8	Déplacement vers lieu d'attente	Transport	Infirmière Aide-soignante	5
9	Attente avant examen(s)	Salle d'attente	∅	?
10	Examens	Externe	Externe	Donnée
11	Attente avant retour CC	Salle d'attente	∅	?
12	Attente avant plâtre	Salle d'attente	∅	?
13	Déplacement vers plâtre	Transport	Infirmière Aide-soignante	5
14	Pose de plâtre	Salle de plâtre	Interne Traumato	15
15	Soutien plâtre	Salle de plâtre	Infirmière	15

Tableau A.16: Traumatologie avec examens sans examen biologique : Description des tâches

A.4.3.2 Avec examens (biologique + autres)

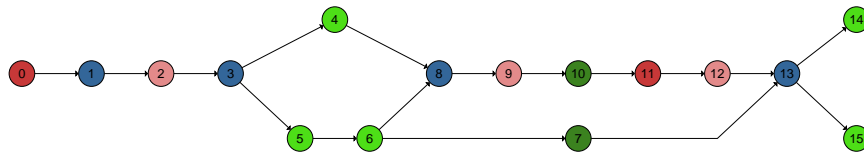


Fig A.17: Traumatologie avec examens dont examen biologique : Graphe de tâches

N°	Nom	Lieu	Ressource	Durée
0	Attente avant entrée CC	Externe	∅	?
1	Entrée CC	Transport	Infirmière Aide-soignante	5
2	Attente avant consultation	Salle d'attente	∅	?
3	Déplacement vers consultation	Transport	Infirmière Aide-soignante	5
4	Consultation médicale	Box	Médecin urgentiste Interne Traumatolo	15
5	Prise de signes vitaux	Box	Infirmière	5
6	Prise de sang	Box	Infirmière	5
7	Analyse biologique	Externe	Externe	Donnée
8	Déplacement vers lieu d'attente	Transport	Infirmière Aide-soignante	5
9	Attente avant examen(s)	Salle d'attente	∅	?
10	Examens	Externe	Externe	Donnée
11	Attente avant retour CC	Salle d'attente	∅	?
12	Attente avant plâtre	Salle d'attente	∅	?
13	Déplacement vers plâtre	Transport	Infirmière Aide-soignante	5
14	Pose de plâtre	Salle de plâtre	Interne Traumatolo	15
15	Soutien plâtre	Salle de plâtre	Infirmière	15

Tableau A.17: Traumatologie avec examens dont examen biologique : Description des tâches

A.5 PARCOURS TRAUMATOLOGIE AVEC SUTURE

On considère pour la suite que si des examens sont prescrits alors l'examen radiologie y figure. Selon son "résultat" le patient peut passer ou non en salle de plâtre.

A.5.1 Sans examens

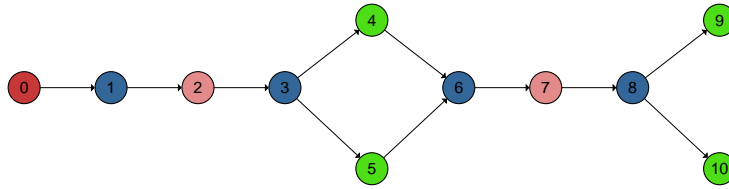


Fig A.18: Traumatologie avec suture sans examens : Graphe de tâches

N°	Nom	Lieu	Ressource	Durée
0	Attente avant entrée CC	Externe	∅	?
1	Entrée CC	Transport	Infirmière Aide-soignante	5
2	Attente avant suture	Salle d'attente	∅	?
3	Déplacement vers suture	Transport	Infirmière Aide-soignante	5
4	Suture	Salle suture	Interne Traumato	15
5	Soutien suture	Salle suture	Infirmière	15
6	Déplacement vers lieu d'attente	Transport	Infirmière Aide-soignante	5
7	Attente avant consultation	Salle d'attente	∅	?
8	Attente vers consultation	Salle d'attente	∅	?
9	Consultation médicale	Box	Médecin urgentiste Interne Traumato	15
10	Prise de signes vitaux	Box	Infirmière	5

Tableau A.18: Traumatologie avec suture sans examens : Description des tâches

A.5.2 Sans plâtre

A.5.2.1 Avec examens (sans examen biologique)

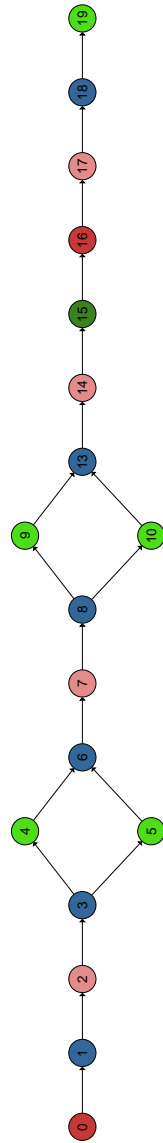


Fig A.19: Traumatologie avec suture avec examens sans examen biologique sans plâtre : Graphe de tâches

A.5.2.2 Avec examens (biologique + autres)

N°	Nom	Lieu	Ressource	Durée
0	Attente avant entrée CC	Externe	∅	?
1	Entrée CC	Transport	Infirmière Aide-soignante	5
2	Attente avant suture	Salle d'attente	∅	?
3	Déplacement vers suture	Transport	Infirmière Aide-soignante	5
4	Suture	Salle suture	Interne Traumato	15
5	Soutien suture	Salle suture	Infirmière	15
6	Déplacement vers lieu d'attente	Transport	Infirmière Aide-soignante	5
7	Attente avant consultation	Salle d'attente	∅	?
8	Attente vers consultation	Salle d'attente	∅	?
9	Consultation médicale	Box	Médecin urgentiste Interne Traumato	15
10	Prise de signes vitaux	Box	Infirmière	5
13	Déplacement vers lieu d'attente	Transport	Infirmière Aide-soignante	5
14	Attente avant examen(s)	Salle d'attente	∅	?
15	Examens	Externe	Externe	Donnée
16	Attente avant retour CC	Salle d'attente	∅	?
17	Attente avant évaluation	Salle d'attente	∅	?
18	Déplacement vers évaluation	Transport	Infirmière Aide-soignante	5
19	Évaluation médicale	Box	Médecin urgentiste Interne médecine	15

Tableau A.19: Traumatologie avec suture avec examens sans examen biologique sans plâtre : Description des tâches

A.5.3 Avec plâtre

A.5.3.1 Avec examens (sans examen biologique)

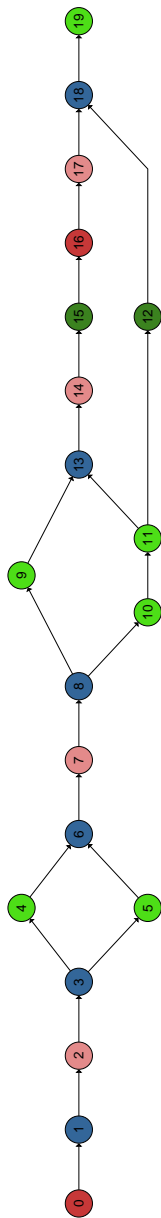


Fig A.20: Traumatologie avec suture avec examens dont examen biologique sans plâtre : Graphe de tâches

A.5.3.2 Avec examens (biologique + autres)

N°	Nom	Lieu	Ressource	Durée
0	Attente avant entrée CC	Externe	∅	?
1	Entrée CC	Transport	Infirmière Aide-soignante	5
2	Attente avant suture	Salle d'attente	∅	?
3	Déplacement vers suture	Transport	Infirmière Aide-soignante	5
4	Suture	Salle suture	Interne Traumatolo	15
5	Soutien suture	Salle suture	Infirmière	15
6	Déplacement vers lieu d'attente	Transport	Infirmière Aide-soignante	5
7	Attente avant consultation	Salle d'attente	∅	?
8	Attente vers consultation	Salle d'attente	∅	?
9	Consultation médicale	Box	Médecin urgentiste Interne Traumatolo	15
10	Prise de signes vitaux	Box	Infirmière	5
11	Prise de sang	Box	Infirmière	5
12	Analyse biologique	Externe	Externe	Donnée
13	Déplacement vers lieu d'attente	Transport	Infirmière Aide-soignante	5
14	Attente avant examen(s)	Salle d'attente	∅	?
15	Examens	Externe	Externe	Donnée
16	Attente avant retour CC	Salle d'attente	∅	?
17	Attente avant évaluation	Salle d'attente	∅	?
18	Déplacement vers évaluation	Transport	Infirmière Aide-soignante	5
19	Évaluation médicale	Box	Médecin urgentiste Interne médecine	15

Tableau A.20: Traumatologie avec suture avec examens dont examen biologique sans plâtre : Description des tâches

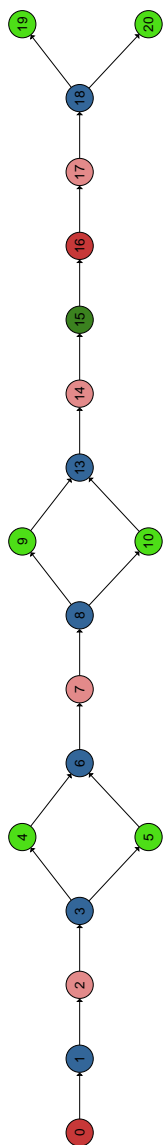


Fig A.21: Traumatologie avec suture avec examens sans examen biologique : Graphe de tâches

N°	Nom	Lieu	Ressource	Durée
0	Attente avant entrée CC	Externe	∅	?
1	Entrée CC	Transport	Infirmière Aide-soignante	5
2	Attente avant suture	Salle d'attente	∅	?
3	Déplacement vers suture	Transport	Infirmière Aide-soignante	5
4	Suture	Salle suture	Interne Traumatolo	15
5	Soutien suture	Salle suture	Infirmière	15
6	Déplacement vers lieu d'attente	Transport	Infirmière Aide-soignante	5
7	Attente avant consultation	Salle d'attente	∅	?
8	Attente vers consultation	Salle d'attente	∅	?
9	Consultation médicale	Box	Médecin urgentiste Interne Traumatolo	15
10	Prise de signes vitaux	Box	Infirmière	5
13	Déplacement vers lieu d'attente	Transport	Infirmière Aide-soignante	5
14	Attente avant examen(s)	Salle d'attente	∅	?
15	Examens	Externe	Externe	Donnée
16	Attente avant retour CC	Salle d'attente	∅	?
17	Attente avant plâtre	Salle d'attente	∅	?
18	Déplacement vers plâtre	Transport	Infirmière Aide-soignante	5
19	Pose de plâtre	Salle de plâtre	Interne Traumatolo	15
20	Soutien plâtre	Salle de plâtre	Infirmière	15

Tableau A.21: Traumatologie avec suture avec examens sans examen biologique : Description des tâches

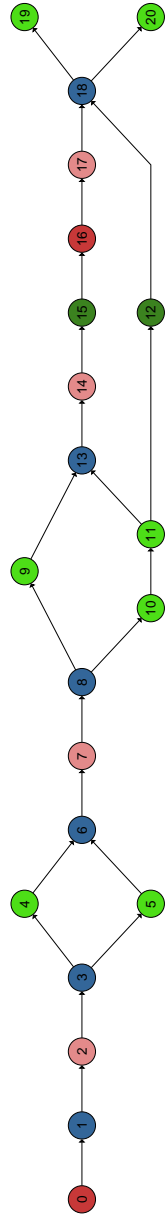


Fig A.22: Traumatologie avec suture avec examens dont examen biologique : Graphe de tâches

N°	Nom	Lieu	Ressource	Durée
0	Attente avant entrée CC	Externe	∅	?
1	Entrée CC	Transport	Infirmière Aide-soignante	5
2	Attente avant suture	Salle d'attente	∅	?
3	Déplacement vers suture	Transport	Infirmière Aide-soignante	5
4	Suture	Salle suture	Interne Traumato	15
5	Soutien suture	Salle suture	Infirmière	15
6	Déplacement vers lieu d'attente	Transport	Infirmière Aide-soignante	5
7	Attente avant consultation	Salle d'attente	∅	?
8	Attente vers consultation	Salle d'attente	∅	?
9	Consultation médicale	Box	Médecin urgentiste Interne Traumato	15
10	Prise de signes vitaux	Box	Infirmière	5
11	Prise de sang	Box	Infirmière	5
12	Analyse biologique	Externe	Externe	Donnée
13	Déplacement vers lieu d'attente	Transport	Infirmière Aide-soignante	5
14	Attente avant examen(s)	Salle d'attente	∅	?
15	Examens	Externe	Externe	Donnée
16	Attente avant retour CC	Salle d'attente	∅	?
17	Attente avant plâtre	Salle d'attente	∅	?
18	Déplacement vers plâtre	Transport	Infirmière Aide-soignante	5
19	Pose de plâtre	Salle de plâtre	Interne Traumato	15
20	Soutien plâtre	Salle de plâtre	Infirmière	15

Tableau A.22: Traumatologie avec suture avec examens dont examen biologique : Description des tâches

BIBLIOGRAPHIE

- Afilal, Mohamed, Farouk Yalaoui, Frédéric Dugardin, Lionel Amodeo, David Laplanche, and Philippe Blua (July 2016). "Forecasting the Emergency Department Patients Flow". In: *Journal of Medical Systems* 40 (7), pp. 1–18. ISSN: 1573689X. DOI: [10.1007/S10916-016-0527-0/TABLES/9](https://doi.org/10.1007/S10916-016-0527-0/TABLES/9). URL: <https://link.springer.com/article/10.1007/s10916-016-0527-0>.
- Ajmi, Faten, Hayfa Zgaya, Sarah Ben Othman, and Slim Hammadi (2019). "Agent-based dynamic optimization for managing the workflow of the patient's pathway". In: *Simulation Modelling Practice and Theory* 96, p. 101935. ISSN: 1569-190X. DOI: <https://doi.org/10.1016/j.simpat.2019.101935>. URL: <http://www.sciencedirect.com/science/article/pii/S1569190X19300607>.
- Akbarzadeh, Babak, Ghasem Moslehi, Mohammad Reisi-Nafchi, and Broos Maenhout (Jan. 2020). "A diving heuristic for planning and scheduling surgical cases in the operating room department with nurse re-rostering". In: *Journal of Scheduling* 2020 23:2 23 (2), pp. 265–288. ISSN: 1099-1425. DOI: [10.1007/S10951-020-00639-6](https://doi.org/10.1007/S10951-020-00639-6).
- Almeida, Bernardo F., Isabel Correia, and Francisco Saldanha-Da-Gama (Sept. 2016). "Priority-based heuristics for the multi-skill resource constrained project scheduling problem". In: *Expert Systems with Applications* 57, pp. 91–103. ISSN: 0957-4174.
- Amarouche, Youcef, Rym Nesrine Guibadj, Elhadja Chaalal, and Aziz Moukrim (Nov. 2020). "Effective neighborhood search with optimal splitting and adaptive memory for the team orienteering problem with time windows". In: *Computers and Operations Research* 123, p. 105039. ISSN: 03050548. DOI: [10.1016/j.cor.2020.105039](https://doi.org/10.1016/j.cor.2020.105039).
- Arezki, Chahinez, Jean-Paul Boufflet, Lahcene Mezouari, and Aziz Moukrim (2020). "Planification d'interventions chirurgicales: heuristiques de construction et d'amélioration". In: *21e congrès annuel de la Société Française de Recherche Opérationnelle et d'Aide à la Décision (ROADEF 2020)*.
- Asplin, Brent R., David J. Magid, Karin V. Rhodes, Leif I. Solberg, Nicole Lurie, and Carlos A. Camargo (2003). "A conceptual model of emergency department crowding". In: *Annals of Emergency Medicine* 42.2, pp. 173–180. ISSN: 0196-0644. DOI: <https://doi.org/10.1067/mem.2003.302>. URL: <https://www.sciencedirect.com/science/article/pii/S019606440300444X>.
- Au, L., G. B. Byrnes, C. A. Bain, M. Fackrell, C. Brand, D. A. Campbell, and P. G. Taylor (2009). "Predicting overflow in an emergency department". In: *IMA Journal of Management Mathematics* 20.1, pp. 39–49. DOI: [10.1093/imaman/dpn007](https://doi.org/10.1093/imaman/dpn007).
- Azadeh, A., M. Hosseinabadi Farahani, S. Torabzadeh, and M. Baghersad (Nov. 2014). "Scheduling prioritized patients in emergency department laboratories". In: *Computer Methods and Programs in Biomedicine* 117.2, pp. 61–70. ISSN: 18727565. DOI: [10.1016/j.cmpb.2014.08.006](https://doi.org/10.1016/j.cmpb.2014.08.006).

- Ballestín, Francisco, Ángeles Pérez, and Sacramento Quintanilla (Feb. 2019). "Scheduling and rescheduling elective patients in operating rooms to minimise the percentage of tardy patients". In: *Journal of Scheduling* 22 (1), pp. 107–118. DOI: [10.1007/S10951-018-0570-4](https://doi.org/10.1007/S10951-018-0570-4).
- Bard, Jonathan F. and Hadi W. Purnomo (July 2005). "Hospital-wide reactive scheduling of nurses with preference considerations". In: *IIE Transactions (Institute of Industrial Engineers)* 37.7, pp. 589–608. ISSN: 0740817X. DOI: [10.1080/07408170590948468](https://doi.org/10.1080/07408170590948468).
- Ben-Said, Asma, Racha El-Hajj, and Aziz Moukrim (2019). "A variable space search heuristic for the Capacitated Team Orienteering Problem". In: *Journal of Heuristics* 25.2, pp. 273–303.
- Bengio, Yoshua, Andrea Lodi, and Antoine Prouvost (2021). "Machine learning for combinatorial optimization: A methodological tour d'horizon". In: *European Journal of Operational Research* 290.2, pp. 405–421. DOI: [10.1016/j.ejor.2020.07.06](https://doi.org/10.1016/j.ejor.2020.07.06).
- Bukkur, Khalid M. M. A., Mohamad Shukri, and Osama Mohammed Elmardi (2018). "A Review for Dynamic Scheduling in Manufacturing". In: *Global Journal of Research In Engineering*.
- Burke, Edmund K., Patrick De Causmaecker, Greet Vanden Berghe, and Hendrik Van Landeghem (2004). "The state of the art of nurse rostering". In: *Journal of Scheduling* 7.6, pp. 441–499. ISSN: 10946136. DOI: [10.1023/B:JOSH.0000046076.75950.0b](https://doi.org/10.1023/B:JOSH.0000046076.75950.0b).
- Burke, Edmund K, Jingpeng Li, and Rong Qu (2009). "A hybrid model of integer programming and variable neighbourhood search for highly-constrained nurse rostering problems". In: *European Journal of Operational Research* 203, pp. 484–493. DOI: [10.1016/j.ejor.2009.07.036](https://doi.org/10.1016/j.ejor.2009.07.036).
- Cardoen, Brecht, Erik Demeulemeester, and Jeroen Beliën (Mar. 2010). "Operating room planning and scheduling: A literature review". In: *European Journal of Operational Research* 201 (3), pp. 921–932. ISSN: 03772217.
- Cardoen, Brecht, Erik Demeulemeester, and Jeroen Beliën (2009). "Sequencing surgical cases in a day-care environment: An exact branch-and-price approach". In: *Computers & Operations Research* 36.9, pp. 2660–2669. ISSN: 0305-0548.
- Castro, Pedro M. and Inês Marques (2015). "Operating room scheduling with Generalized Disjunctive Programming". In: *Computers & Operations Research* 64, pp. 262–273. ISSN: 0305-0548.
- Cheang, Brenda, Andrew Lim, Brian Rodrigues, and B Cheang (2003). "Nurse Rostering Problems: A Bibliographic Survey". In: *European Journal of Operational Research* 151.3, pp. 447–460. DOI: [10.1016/S0377-2217\(03\)00021-3](https://doi.org/10.1016/S0377-2217(03)00021-3). URL: http://ink.library.smu.edu.sg/lkcsb/_research.
- Chen, HaoJie, Guofu Ding, Jian Zhang, and Shengfeng Qin (2019). "Research on priority rules for the stochastic resource constrained multi-project scheduling problem with new project arrival". In: *Computers & Industrial Engineering* 137, p. 106060. ISSN: 0360-8352. DOI: <https://doi.org/10.1016/j.cie.2019.106060>. URL: <https://www.sciencedirect.com/science/article/pii/S0360835219305194>.

- Chen, T. L. and C. C. Wang (Feb. 2016). "Multi-objective simulation optimization for medical capacity allocation in emergency department". In: *Journal of Simulation* 10 (1), pp. 50–68. ISSN: 17477786. DOI: [10.1057/jos.2014.39](https://doi.org/10.1057/jos.2014.39).
- Chouba, Ibtissem, Lionel Amodeo, Taha Arbaoui, Farouk Yalaoui, David Laplanche, and Stéphane Sanchez (Sept. 2022). "Optimization of French Hospital Department Activities Using an Efficient Discrete Event Simulation Model". In: *Applied Sciences* 2022, Vol. 12, Page 9498 12 (19), p. 9498. ISSN: 2076-3417. DOI: [10.3390/APP12199498](https://doi.org/10.3390/APP12199498). URL: <https://www.mdpi.com/2076-3417/12/19/9498/htm><https://www.mdpi.com/2076-3417/12/19/9498>.
- Daknou, Amani, Hayfa Zgaya, Slim Hammadi, and Hervé Hubert (2010). "A dynamic patient scheduling at the emergency department in hospitals". In: *2010 IEEE workshop on health care management (WHCM)*. IEEE, pp. 1–6.
- Denton, Brian, James Viapiano, and Andrea Vogl (Feb. 2007). "Optimization of surgery sequencing and scheduling decisions under uncertainty". In: *Health Care Management Science* 10 (1), pp. 13–24. ISSN: 13869620.
- Dueck, Gunter (Jan. 1993). "New optimization heuristics; The great deluge algorithm and the record-to-record travel". In: *Journal of Computational Physics* 104.1, pp. 86–92. ISSN: 00219991.
- Duguay, Christine and Fatah Chetouane (2007). "Modeling and Improving Emergency Department Systems using Discrete Event Simulation". In: *SIMULATION* 83.4, pp. 311–320. DOI: [10.1177/0037549707083111](https://doi.org/10.1177/0037549707083111). eprint: <https://doi.org/10.1177/0037549707083111>. URL: <https://doi.org/10.1177/0037549707083111>.
- Durasević, Marko and Domagoj Jakobović (2018). "A survey of dispatching rules for the dynamic unrelated machines environment". In: *Expert Systems with Applications* 113, pp. 555–569. ISSN: 0957-4174. DOI: <https://doi.org/10.1016/j.eswa.2018.06.053>. URL: <https://www.sciencedirect.com/science/article/pii/S0957417418304159>.
- Erdogan, S. Ayca and Brian T. Denton (2011). "Surgery Planning and Scheduling". In: *Wiley Encyclopedia of Operations Research and Management Science*. American Cancer Society. Chap. 1, pp. 1–10. ISBN: 9780470400531.
- Ernst, A. T., H. Jiang, M. Krishnamoorthy, and D. Sier (Feb. 2004). "Staff scheduling and rostering: A review of applications, methods and models". In: *European Journal of Operational Research*. Vol. 153. 1. North-Holland, pp. 3–27. DOI: [10.1016/S0377-2217\(03\)00095-X](https://doi.org/10.1016/S0377-2217(03)00095-X).
- Ferreira, Cicero, Fernandes Costa Filho, Dayse Aparecida, Rivera Rocha, Guimarães Fernandes Costa, Wagner Coelho, and Albuquerque Pereira (2012). "Using Constraint Satisfaction Problem approach to solve human resource allocation problems in cooperative health services". In: DOI: [10.1016/j.eswa.2011.07.027](https://doi.org/10.1016/j.eswa.2011.07.027).
- Fischer, Gabriel Souto, Rodrigo da Rosa Righi, Gabriel de Oliveira Ramos, Cristiano André da Costa, and Joel J.P.C. Rodrigues (2020). "EiHealth: Using Internet of Things and data prediction for elastic management of human resources in smart hospitals". In: *Engineering Applications of Artificial Intelligence* 87, p. 103285. ISSN: 0952-1976. DOI: <https://doi.org/10.1016/j.engappai.2019.103285>.

- Forero, Roberto, Kenneth M Hillman, Sally McCarthy, Daniel M Fatovich, Anthony P Joseph, and Drew B Richardson (2010). "Access block and ED overcrowding". In: *Emergency Medicine Australasia* 22.2, pp. 119–135. DOI: <https://doi.org/10.1111/j.1742-6723.2010.01270.x>. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1742-6723.2010.01270.x>. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1742-6723.2010.01270.x>.
- Goren, Selcuk and Ihsan Sabuncuoglu (Jan. 2007). "Robustness and stability measures for scheduling: single-machine environment". In: <http://dx.doi.org/10.1080/07408170701283198> 40 (1), pp. 66–83. ISSN: 0740817X. DOI: 10.1080/07408170701283198. URL: <https://www.tandfonline.com/doi/abs/10.1080/07408170701283198>.
- Green, Linda V., João Soares, James F. Giglio, and Robert A. Green (Jan. 2006). "Using Queueing Theory to Increase the Effectiveness of Emergency Department Provider Staffing". In: *Academic Emergency Medicine* 13.1, pp. 61–68. ISSN: 10696563. DOI: 10.1197/j.aem.2005.07.034. URL: <http://doi.wiley.com/10.1197/j.aem.2005.07.034>.
- Gupta, Diwakar (Jan. 2009). "Surgical Suites' Operations Management". In: *Production and Operations Management* 16 (6), pp. 689–700. ISSN: 10591478.
- Hashemi Doulabi, Seyed Hossein, Louis-Martin Rousseau, and Gilles Pesant (2016). "A constraint-programming-based branch-and-price-and-cut approach for operating room planning and scheduling". In: *INFORMS Journal on Computing* 28.3, pp. 432–448.
- Heger, Jens, Jürgen Branke, Torsten Hildebrandt, and Bernd Scholz-Reiter (2016). "Dynamic adjustment of dispatching rule parameters in flow shops with sequence-dependent set-up times". In: *International Journal of Production Research* 54.22, pp. 6812–6824. DOI: 10.1080/00207543.2016.1178406. eprint: <https://doi.org/10.1080/00207543.2016.1178406>. URL: <https://doi.org/10.1080/00207543.2016.1178406>.
- Hemmati, Ahmad and Lars Magnus Hvattum (Sept. 2017). "Evaluating the importance of randomization in adaptive large neighborhood search". In: *International Transactions in Operational Research* 24 (5), pp. 929–942. ISSN: 14753995.
- Higginson, Ian (2012). "Emergency department crowding". In: *Emergency Medicine Journal* 29.6, pp. 437–443. ISSN: 1472-0205. DOI: 10.1136/emered-2011-200532. eprint: <https://emj.bmj.com/content/29/6/437.full.pdf>. URL: <https://emj.bmj.com/content/29/6/437>.
- Hoot, Nathan R. and Dominik Aronsky (2008). "Systematic Review of Emergency Department Crowding: Causes, Effects, and Solutions". In: *Annals of Emergency Medicine* 52.2, 126–136.e1. ISSN: 0196-0644. DOI: <https://doi.org/10.1016/j.annemergmed.2008.03.014>. URL: <https://www.sciencedirect.com/science/article/pii/S0196064408006069>.
- Hoot, Nathan R., Larry J. LeBlanc, Ian Jones, Scott R. Levin, Chuan Zhou, Cynthia S. Gadd, and Dominik Aronsky (May 2009). "Forecasting Emergency Department Crowding: A Prospective, Real-time Evaluation". In: *Journal of the American Medical Informatics Association* 16.3, pp. 338–345. ISSN: 1067-5027. DOI: 10.1197/jamia.M2772. eprint: <https://academic.oup.com/jamia/article-pdf/16/3/338/2548277/16-3-338.pdf>. URL: <https://doi.org/10.1197/jamia.M2772>.

- Hwang, Ula and John Concato (2004). "Care in the Emergency Department: How Crowded Is Overcrowded?" In: *Academic Emergency Medicine* 11.10, pp. 1097–1101. DOI: <https://doi.org/10.1197/j.aem.2004.07.004>. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1197/j.aem.2004.07.004>. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1197/j.aem.2004.07.004>.
- Hwang, Ula, Melissa L. McCarthy, Dominik Aronsky, Brent Asplin, Peter W. Crane, Catherine K. Craven, Stephen K. Epstein, Christopher Fee, Daniel A. Handel, Jesse M. Pines, Niels K. Rathlev, Robert W. Schafermeyer, Frank L. Zwemer Jr., and Steven L. Bernstein (2011). "Measures of Crowding in the Emergency Department: A Systematic Review". In: *Academic Emergency Medicine* 18.5, pp. 527–538. DOI: <https://doi.org/10.1111/j.1553-2712.2011.01054.x>. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1553-2712.2011.01054.x>. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1553-2712.2011.01054.x>.
- Izady, Navid and Dave Worthington (June 2012). "Setting staffing requirements for time dependent queueing networks: The case of accident and emergency departments". In: *European Journal of Operational Research* 219.3, pp. 531–540. ISSN: 03772217. DOI: [10.1016/j.ejor.2011.10.040](https://doi.org/10.1016/j.ejor.2011.10.040).
- Jayaprakash, Namita, Ronan O'Sullivan, Tareg Bey, Suleman S Ahmed, and Shahram Lotfipour (Nov. 2009). "Crowding and delivery of healthcare in emergency departments: the European perspective". In: *The western journal of emergency medicine* 10.4, 233–239. ISSN: 1936-900X. URL: <https://europepmc.org/articles/PMC2791723>.
- Karimi-Mamaghan, Maryam, Mehrdad Mohammadi, Patrick Meyer, Amir Mohammad Karimi-Mamaghan, and El-Ghazali Talbi (2022). "Machine learning at the service of meta-heuristics for solving combinatorial optimization problems: A state-of-the-art". In: *European Journal of Operational Research* 296.2, pp. 393–422. ISSN: 0377-2217. DOI: <https://doi.org/10.1016/j.ejor.2021.04.032>.
- Kırış, Şafak, Nihat Yüzügüllü, Nurdan Ergün, and A. Alper Çevik (2010). "A knowledge-based scheduling system for Emergency Departments". In: *Knowledge-Based Systems* 23.8, pp. 890–900. ISSN: 0950-7051. DOI: <https://doi.org/10.1016/j.knosys.2010.06.005>.
- Kolisch, Rainer and Sönke Hartmann (1999). "Heuristic algorithms for the resource-constrained project scheduling problem: Classification and computational analysis". In: *Project scheduling*. Springer, pp. 147–178.
- Laarhoven, Peter J. M. van and Emile H. L. Aarts (1987). "Simulated Annealing: Theory and Applications". In: *Simulated Annealing: Theory and Applications*. DOI: [10.1007/978-94-015-7744-1](https://doi.org/10.1007/978-94-015-7744-1).
- LaCalle, Eduardo and Elaine Rabin (2010). "Frequent Users of Emergency Departments: The Myths, the Data, and the Policy Implications". In: *Annals of Emergency Medicine* 56.1, pp. 42–48. ISSN: 0196-0644. DOI: <https://doi.org/10.1016/j.annemergmed.2010.01.032>. URL: <http://www.sciencedirect.com/science/article/pii/S0196064410001058>.
- Lee, Seunghoon and Young Hoon Lee (Mar. 2020). "Improving Emergency Department Efficiency by Patient Scheduling Using Deep Reinforcement Learning". In: *Healthcare* 8.2, pp. 77–78. ISSN: 2227-9032. DOI: [10.3390/healthcare8020077](https://doi.org/10.3390/healthcare8020077).

- Liew, Don, Danny Liew, and Marcus P Kennedy (2003). "Emergency department length of stay independently predicts excess inpatient length of stay". In: *Medical Journal of Australia* 179.10, pp. 524–526. DOI: <https://doi.org/10.5694/j.1326-5377.2003.tb05676.x>. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.5694/j.1326-5377.2003.tb05676.x>. URL: <https://onlinelibrary.wiley.com/doi/abs/10.5694/j.1326-5377.2003.tb05676.x>.
- Luscombe, Ruth and Erhan Kozan (2016). "Dynamic resource allocation to improve emergency department efficiency in real time". In: *European Journal of Operational Research* 255.2, pp. 593–603. DOI: [10.1016/j.ejor.2016.05.03](https://doi.org/10.1016/j.ejor.2016.05.03).
- Magerlein, J. M. and J B Martin (1978). "Surgical demand scheduling: a review." In: *Health services research* 13 4, pp. 418–33.
- Mahmoudzadeh, Houra, Akram Mirahmadi Shalamzari, and Hossein Abouee-Mehrzi (2020). "Robust multi-class multi-period patient scheduling with wait time targets". In: *Operations Research for Health Care* 25, p. 100254. ISSN: 2211-6923. DOI: <https://doi.org/10.1016/j.orhc.2020.100254>. URL: <https://www.sciencedirect.com/science/article/pii/S2211692320300345>.
- Marchesi, Janaina F., Silvio Hamacher, and Julia L. Fleck (Apr. 2020). "A stochastic programming approach to the physician staffing and scheduling problem". In: *Computers and Industrial Engineering* 142. ISSN: 03608352. DOI: [10.1016/J.CIE.2020.106281](https://doi.org/10.1016/J.CIE.2020.106281).
- May, Jerrold H., William E. Spangler, David P. Strum, and Luis G. Vargas (2011). "The Surgical Scheduling Problem: Current Research and Future Opportunities". In: *Production and Operations Management* 20.3, pp. 392–405. DOI: <https://doi.org/10.1111/j.1937-5956.2011.01221.x>.
- Mezouari, Lahcene, Jean-Paul Boufflet, and Aziz Moukrim (2022). "Surgery planning for elective patients: A dedicated heuristic and an effective ALNS". In: *Engineering Applications of Artificial Intelligence* 115, p. 105220.
- (2023). "Scheduling patient pathways in an Emergency Department, ILP model and IDCH approach". submitted.
- Minguillon, Fabio Echsler and Gisela Lanza (2019). "Coupling of centralized and decentralized scheduling for robust production in agile production systems". In: *Procedia CIRP* 79, pp. 385–390.
- Moratori, Patrick, Sanja Petrovic, and Jose Antonio Vazquez-Rodriguez (2012). "Match-up approaches to a dynamic rescheduling problem". In: *International Journal of Production Research* 50.1, pp. 261–276.
- Nouiri, Maroua, Abdelghani Bekrar, Abderrazak Jemai, Damien Trentesaux, Ahmed Chiheb Ammari, and Smail Niar (2017). "Two stage particle swarm optimization to solve the flexible job shop predictive scheduling problem considering possible machine breakdowns". In: *Computers & Industrial Engineering* 112, pp. 595–606. ISSN: 0360-8352. DOI: <https://doi.org/10.1016/j.cie.2017.03.006>. URL: <https://www.sciencedirect.com/science/article/pii/S036083521730089X>.
- Oliveira, Mariana, Valérie Bélanger, Inês Marques, and Angel Ruiz (2020). "Assessing the impact of patient prioritization on operating room schedules". In: *Operations Research for Health Care* 24, p. 100232. ISSN: 2211-6923.

- Ouelhadj, Djamila and Sanja Petrovic (Aug. 2009). "A survey of dynamic scheduling in manufacturing systems". en. In: *Journal of Scheduling* 12.4, pp. 417–431. ISSN: 1099-1425. DOI: [10.1007/s10951-008-0090-8](https://doi.org/10.1007/s10951-008-0090-8). URL: <https://doi.org/10.1007/s10951-008-0090-8> (visited on 01/09/2023).
- Park, Jaesang, Byung In Kim, Myungeun Eom, and Byung Kwan Choi (July 2021). "Operating room scheduling considering surgeons' preferences and cooperative operations". In: *Computers & Industrial Engineering* 157, p. 107306. ISSN: 0360-8352.
- Priore, Paolo, Alberto Gomez, Raúl Pino, and Rafael Rosillo (2014). "Dynamic scheduling of manufacturing systems using machine learning: An updated review". In: *Ai Edam* 28.1, pp. 83–97.
- Queiroz dos Santos, João Paulo, Jorge Dantas de Melo, Adrião Dória Duarte Neto, and Daniel Aloise (2014). "Reactive Search strategies using Reinforcement Learning, local search algorithms and Variable Neighborhood Search". In: *Expert Systems with Applications* 41.10, pp. 4939–4949. ISSN: 0957-4174. DOI: <https://doi.org/10.1016/j.eswa.2014.01.040>.
- Rahimi, Iman and Amir H. Gandomi (May 2020). "A Comprehensive Review and Analysis of Operating Room and Surgery Scheduling". In: *Archives of Computational Methods in Engineering*, pp. 1–22. ISSN: 18861784.
- Richardson, Drew and David Mountain (Apr. 2009). "Myths versus facts in emergency department overcrowding and hospital access block". In: *The Medical journal of Australia* 190, pp. 369–74. DOI: [10.5694/j.1326-5377.2009.tb02451.x](https://doi.org/10.5694/j.1326-5377.2009.tb02451.x).
- Richardson, Drew B (2002). "The access-block effect: relationship between delay to reaching an inpatient bed and inpatient length of stay". In: *Medical Journal of Australia* 177.9, pp. 492–495. DOI: <https://doi.org/10.5694/j.1326-5377.2002.tb04917.x>. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.5694/j.1326-5377.2002.tb04917.x>. URL: <https://onlinelibrary.wiley.com/doi/abs/10.5694/j.1326-5377.2002.tb04917.x>.
- Riise, Atle and Carlo Mannino (2012). *The surgery scheduling problem—a general model*. <https://www.sintef.no/en/publications/publication/1268319/>. Report. Postboks 124 Blindern NO-0314 Oslo NORWAY.
- Riise, Atle, Carlo Mannino, and Edmund K. Burke (2016). "Modelling and solving generalised operational surgery scheduling problems". In: *Computers & Operations Research* 66, pp. 1–11. ISSN: 0305-0548.
- Ropke, Stefan and David Pisinger (2006). "An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows". In: *Transportation Science* 40 (4), pp. 455–472. ISSN: 15265447.
- Roshanaei, Vahid and Bahman Naderi (Aug. 2021). "Solving integrated operating room planning and scheduling: Logic-based Benders decomposition versus Branch-Price-and-Cut". In: *European Journal of Operational Research* 293 (1), pp. 65–78. ISSN: 0377-2217.
- Salmon, Andrew, Sebastian Rachuba, Simon Briscoe, and Martin Pitt (2018). "A structured literature review of simulation modelling applied to Emergency Departments: Current patterns and emerging trends". In: *Operations Research for Health Care* 19,

- pp. 1–13. ISSN: 2211-6923. DOI: <https://doi.org/10.1016/j.orhc.2018.01.001>. URL: <https://www.sciencedirect.com/science/article/pii/S2211692317301042>.
- Samudra, Michael, Carla Van Riet, Erik Demeulemeester, Brecht Cardoen, Nancy Vansteenkiste, and Frank E. Rademakers (Oct. 2016). "Scheduling operating rooms: achievements, challenges and pitfalls". In: *Journal of Scheduling* 19 (5), pp. 493–525. ISSN: 10946136.
- Seshaiah, CV and HB Thiagaraj (2011). "A queueing network congestion model in hospitals". In: *European journal of scientific research* 63.3, pp. 419–427.
- Sier, D, P Tobin, and C McGurk (1997). "Scheduling surgical procedures". In: *Journal of the Operational Research Society* 48.9, pp. 884–891. DOI: [10.1057/palgrave.jors.2600441](https://doi.org/10.1057/palgrave.jors.2600441).
- Sinreich, David, Ola Jabali, and Nico P. Dellaert (Mar. 2012). "Reducing emergency department waiting times by adjusting work shifts considering patient visits to multiple care providers". In: *IIE Transactions* 44.3, pp. 163–180. ISSN: 0740-817X. DOI: [10.1080/0740817X.2011.609875](https://doi.org/10.1080/0740817X.2011.609875).
- SINTEF (2013). *Testbed instances for optimization in health care*. <http://www.sintef.no/projectweb/health-care-optimization/testbed>.
- Soudi, Asie, Mehdi Heydari, and Mohammad Mahdavi Mazdeh (2019). "A new approach for integrated surgical procedure scheduling with arrival uncertainty". In: *International Journal of Operational Research* 34.3, pp. 430–449. URL: <https://EconPapers.repec.org/RePEc:ids:ijores:v:34:y:2019:i:3:p:430-449>.
- Testik, Murat Caner, Banu Yuksel Ozkaya, Salih Aksu, and Osman Ilhami Ozcebe (Apr. 2012). "Discovering blood donor arrival patterns using data mining: A method to investigate service quality at blood centers". In: *Journal of Medical Systems* 36.2, pp. 579–594. ISSN: 01485598. DOI: [10.1007/s10916-010-9519-7](https://doi.org/10.1007/s10916-010-9519-7).
- Tormos, Pilar and Antonio Lova (2003). "An efficient multi-pass heuristic for project scheduling with constrained resources". In: *International Journal of Production Research* 41.5, pp. 1071–1086. DOI: [10.1080/0020754021000033904](https://doi.org/10.1080/0020754021000033904).
- Wang, Tao, Nadine Meskens, and David Duvivier (2015). "Scheduling operating theatres: Mixed integer programming vs. constraint programming". In: *European Journal of Operational Research* 247.2, pp. 401–413. ISSN: 0377-2217.
- Wang, Y, W L Hare, L Vertesi, and A R Rutherford (2011). "Using simulation to model and optimize acute care access in relation to hospital bed count and bed distribution". In: *Journal of Simulation* 5.2, pp. 101–110. DOI: [10.1057/jos.2010.7](https://doi.org/10.1057/jos.2010.7). eprint: <https://doi.org/10.1057/jos.2010.7>. URL: <https://doi.org/10.1057/jos.2010.7>.
- Xiang, Wei (2017). "A multi-objective ACO for operating room scheduling optimization". In: *Natural Computing* 16.4, pp. 607–617.
- Xiang, Wei, Jiao Yin, and Gino Lim (2015). "An ant colony optimization approach for solving an operating room surgery scheduling problem". In: *Computers & Industrial Engineering* 85, pp. 335–345. ISSN: 0360-8352.
- Xie, Xiaolan and Mark A. Lawley (2015). "Operations research in healthcare". In: *International Journal of Production Research* 53.24, pp. 7173–7176.

- Xu, M., T.C. Wong, and K.S. Chin (2014). "A medical procedure-based patient grouping method for an emergency department". In: *Applied Soft Computing* 14. Special issue on hybrid intelligent methods for health technologies, pp. 31–37. ISSN: 1568-4946. DOI: <https://doi.org/10.1016/j.asoc.2013.09.022>. URL: <https://www.sciencedirect.com/science/article/pii/S1568494613003165>.
- Zhang, Ao, Xiaomin Zhu, Qian Lu, and Runtong Zhang (2019a). "Impact of Prioritization on the Outpatient Queuing System in the Emergency Department with Limited Medical Resources". In: *Symmetry* 11.6. ISSN: 2073-8994. DOI: [10.3390/sym11060796](https://doi.org/10.3390/sym11060796). URL: <https://www.mdpi.com/2073-8994/11/6/796>.
- Zhang, Jian, Mahjoub Dridi, and Abdellah El Moudni (2019b). "A two-level optimization model for elective surgery scheduling with downstream capacity constraints". In: *European Journal of Operational Research* 276.2, pp. 602–613. ISSN: 0377-2217. DOI: <https://doi.org/10.1016/j.ejor.2019.01.036>.
- Zhou, Yun, Mahmut Parlar, Vedat Verter, and Shannon Fraser (Sept. 2021). "Surgical Scheduling with Constrained Patient Waiting Times". In: *Production and Operations Management* 30.9, pp. 3253–3271. DOI: [10.1111/poms.13427](https://doi.org/10.1111/poms.13427).

