Titre: Title: Stratégies de résolution exacte du problème RCCP pour améliorer la planification tactique

Auteur: Author: Anis Nourredine

Date: 2025

Type: Mémoire ou thèse / Dissertation or Thesis

Directeurs de recherche: Advisors: Robert Pellerin

Programme: Program: Doctorat en mathématiques

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Stratégies de résolution exacte du problème RCCP pour améliorer la planification tactique

ANIS NOUREDDINE
Département de mathématiques et de génie industriel

Thèse présentée en vue de l'obtention du diplôme de *Philosophiæ Doctor*
Mathématiques

Novembre 2025

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Cette thèse intitulée :

Stratégies de résolution exacte du problème RCCP pour améliorer la planification tactique

présentée par **Anis NOUREDDINE**
en vue de l'obtention du diplôme de *Philosophiæ Doctor*
a été dûment acceptée par le jury d'examen constitué de :

Issmaïl EL HALLAOUI, président
Robert PELLERIN, membre et directeur de recherche
Michel GAMACHE, membre
Claude-Guy QUIMPER, membre externe

DÉDICACE

À la mémoire de mon encadreur, François Soumis. Sa bienveillance et ses conseils avisés continueront de m'inspirer tout au long de mon parcours.

REMERCIEMENTS

Après cinq longues années de recherche, il est difficile de résumer en quelques lignes toute la gratitude que je ressens envers celles et ceux qui m'ont accompagné et soutenu tout au long de ce parcours. J'essaierai ici d'exprimer, du mieux possible, ma reconnaissance à l'égard de toutes les personnes qui ont contribué, de près ou de loin, à chaque étape de ce doctorat, et dont j'ai tant appris, sur le plan scientifique comme personnel.

Je tiens tout d'abord à remercier mes directeurs de thèse. Je commence par le professeur François Soumis, qui nous a quittés, laissant un grand vide derrière lui : un vide scientifique, tant ses idées, son intuition et sa rigueur étaient exceptionnels, mais aussi un vide humain, tant son humour et sa bienveillance marquaient ceux qui le côtoyaient. C'est lui qui m'a donné la chance d'entreprendre ce doctorat et d'intégrer son équipe de recherche, un environnement à la fois exigeant et inspirant. Je lui suis profondément reconnaissant de m'avoir permis de vivre cette aventure scientifique et humaine.

C'est également grâce à lui que j'ai eu la chance de rencontrer le professeur Robert Pellerin, qui était mon codirecteur et qui est devenu par la suite mon directeur de thèse. Grand chercheur et pédagogue, il m'a toujours impressionné par son expertise, la clarté de son raisonnement et la profondeur de ses analyses. Ses conseils précis et ses orientations justes ont profondément enrichi mes travaux et m'ont permis de donner à mes recherches une dimension plus appliquée. J'ai énormément apprécié sa disponibilité, sa bienveillance et son engagement constant à m'aider à progresser. Je lui témoigne toute ma gratitude pour sa confiance, son accompagnement attentif et la qualité de nos échanges tout au long de ces années.

Je remercie également Thales pour son appui et pour le financement d'une partie de mon parcours doctoral. L'intégration à leurs réunions m'a permis de découvrir concrètement les défis de la recherche en milieu industriel et de comprendre comment les approches académiques peuvent contribuer à résoudre des problématiques réelles et complexes.

Même si mon stage à Hydro-Québec ne faisait pas directement partie de mon sujet de doctorat, je souhaite remercier Rachid Hassani pour m'avoir offert l'opportunité d'intégrer son équipe travaillant sur l'optimisation robuste et la prévision de la demande. Sa volonté de me pousser à donner le meilleur de moi-même ont été une source de motivation.

Je souhaite aussi remercier Polytechnique Montréal, le centre de recherche CIRRELT ainsi que le centre de recherche GERAD pour m'avoir offert un cadre de travail stimulant, un environnement scientifique d'une grande richesse et les ressources nécessaires à l'aboutissement de cette thèse.

Je souhaite également remercier mon encadrant de Master, Mohammed Yagouni, dont le soutien s'est prolongé bien au-delà de cette période et qui m'a profondément inspiré tout au long de mon parcours.

Je remercie par ailleurs les membres du jury, Issmail El Hallaoui, Michel Gamache et Claude-Guy Quimper, pour avoir accepté de participer à l'évaluation de cette thèse. Leur expertise et le temps qu'ils ont consacré à son examen sont grandement appréciés.

Mes parents... Que puis-je dire de leur impact ? Ils sont mes piliers, ma base, le socle sur lequel je m'appuie depuis toujours. L'amour de la science m'a été transmis par eux. Leurs conseils avisés et constructifs ont constamment guidé mes choix et mes réflexions. Ils ont toujours cherché à me pousser vers le haut, quelles que soient les circonstances, et m'ont transmis des valeurs humaines fortes et solides. Grâce à eux, j'ai développé un esprit de volonté, de persévérance et de défi qui m'accompagnera toute ma vie.

Mon grand frère Lamine et ma petite sœur Amel occupent une place toute particulière dans ma vie. Lamine a toujours été à mes côtés, présent dans les moments décisifs, par ses conseils lucides et son soutien constant. Sa persévérance et sa rigueur m'ont toujours servi de modèle. Il a cette capacité à aller au bout de tout ce qu'il entreprend, et c'est en l'observant que j'ai souvent retrouvé la motivation d'avancer. Quant à Amel, elle a toujours été là pour écouter, encourager et illuminer nos vies. Véritable bijou de notre famille, sa bienveillance a souvent apporté du réconfort dans les moments difficiles.

Et enfin, ma femme, Randa Ouchene, celle avec qui j'ai partagé chaque étape de ce doctorat : les réussites, les moments de doute, les joies comme les sacrifices. Sa patience, son soutien constant et sa compréhension ont été des forces essentielles tout au long de ce parcours. Au-delà de son appui quotidien, elle a été une véritable partenaire de réflexion, de courage et d'inspiration. Rien de tout cela n'aurait eu la même signification sans elle. Son intelligence et sa sensibilité ont fait de cette aventure une route partagée à deux.

Au fil de mon parcours, j'ai eu la chance de croiser et de côtoyer des amis précieux qui ont su rendre ce chemin plus léger. Je pense notamment à Mohamed El Fassi, avec qui j'ai partagé d'excellents moments d'échanges et de complicité. Au GERAD, j'ai eu la chance de rencontrer des personnes exceptionnelles comme Ryad, Omar, Oussama, Philippes, Mohamed, Abdelhakim, Fairouz, Wissem et tant d'autres avec qui j'ai passé des moments de détente parfaits et enrichissants. À toutes celles et ceux qui ont participé, de près ou de loin, à ces années de recherche et de vie commune, sachez que vous avez tous compté.

RÉSUMÉ

Cette thèse aborde la problématique de la *planification tactique de projets*, à travers l'étude du *Rough-Cut Capacity Planning (RCCP)*. Ce type de modèle s'applique aux premières phases des projets et vise à optimiser, pour chaque période, les intensités d'exécution des lots de travail nécessaires à la réalisation du projet. L'objectif est d'assurer la cohérence entre les besoins du projet et la disponibilité des ressources critiques sur un horizon de moyen terme, avant la phase d'ordonnancement détaillée.

L'objectif général de cette recherche est de proposer de nouvelles formulations linéaires en nombres entiers mixtes (MIP) permettant de résoudre efficacement le problème du RCCP et ses extensions, tout en améliorant à la fois la qualité des solutions et la performance numérique des modèles existants. Les travaux s'appuient sur une analyse approfondie du comportement des solveurs modernes et sur une reformulation méthodique des contraintes, afin de réduire la complexité du modèle, d'améliorer ses performances de résolution et, par là même, d'aborder des extensions plus complexes, également visées par les objectifs de cette recherche.

La première contribution de cette thèse consiste à développer un nouveau modèle MIP en temps continu (CT) pour le *RCCP*, fondé sur une analyse approfondie du comportement du solveur d'optimisation. L'étude du comportement du solveur *CPLEX* a mis en évidence plusieurs sources de ralentissement, notamment la symétrie des solutions, qui bloque la progression de la borne inférieure, ainsi que la faiblesse de la relaxation linéaire après le prétraitement du solveur. Les résultats expérimentaux montrent une réduction moyenne de *72 %* du nombre d'itérations du dual simplexe et une diminution significative du *gap d'optimalité* (c'est-à-dire du gap d'intégralité à la fin de la résolution), permettant de résoudre des instances de taille supérieure à celles traitées avec le modèle de base. Cette contribution met également en lumière l'intérêt d'une analyse du comportement du solveur comme levier d'amélioration des formulations MIP. Enfin, une approche bi-objectif combinant les variantes *Time-driven* et *Resource-driven* a été proposée afin d'aborder des contextes de planification plus proches des applications industrielles réelles.

La deuxième contribution vise à améliorer davantage la formulation en traitant la variante *Time-driven*, à travers la résolution de deux anomalies principales : l'instabilité numérique et la faiblesse de la relaxation linéaire après le prétraitement. Pour cela, une étude comparative a été menée entre plusieurs formulations du RCCP afin d'évaluer leurs performances structurelles. Huit modèles ont été comparés selon deux axes : le temps de résolution et la qualité des solutions. Les résultats montrent que la formulation en temps continu, basée sur les *step*

variables et l'ajout de variables continues, surpasse les autres versions en temps continu pour l'ensemble des critères étudiés. Elle offre des bornes plus serrées, un arbre de branchement plus restreint et une convergence plus rapide vers l'optimalité. Une étude théorique a également été menée entre les modèles testés, montrant que la qualité de la relaxation linéaire n'est qu'un facteur parmi d'autres influençant la performance globale du modèle. En moyenne, le modèle proposé est environ sept fois plus rapide que le modèle en temps continu présenté dans le premier article, et la qualité des solutions est significativement supérieure, atteignant jusqu'à 67 % de gain au niveau des coûts dans certaines instances. Cette étude met en évidence qu'un modèle en temps continu peut rivaliser, voire surpasser, les modèles en temps discret souvent privilégiés dans la littérature pour leur complexité moindre par rapport aux formulations en temps continu, tout en conservant une flexibilité dans la représentation du temps et dans la manière dont les relations de précédence sont traitées, notamment lorsque les périodes sont agrégées. Cette étude comparative a également permis d'identifier les éléments structurels qui influencent le plus le comportement du solveur, notamment les relations de précédence et la manière de lier les périodes aux dates de début et de fin des lots de travail, offrant ainsi des pistes méthodologiques pour la conception de modèles plus performants.

La troisième contribution de la thèse concerne l'intégration du *nivellement des ressources (resource leveling)* au modèle RCCP. Ce volet vise à lisser les fluctuations de charge sur l'horizon de planification afin d'éviter les périodes de surcharge et de sous-utilisation, qui nuisent à la productivité et augmentent les coûts liés à la main-d'œuvre. Pour ce faire, un modèle étendu a été proposé, permettant des intensités non uniformes entre les ressources. Les expériences numériques montrent que ce modèle permet une réduction moyenne de 35 % de la variance inter-périodes, tout en maintenant la durée globale du projet. Une contrainte imposant un profil de charge unimodal a été intégrée, de sorte que la charge augmente jusqu'à un pic avant de décroître jusqu'à la fin de l'horizon. Cette approche permet d'éviter les cycles de recrutement et de licenciement, stabilise la charge globale et offre une représentation plus réaliste du fonctionnement des organisations. Une stratégie incrémentale de résolution a également été développée afin d'améliorer l'efficacité computationnelle du modèle étendu. Cette stratégie ajuste progressivement l'horizon de planification et intègre une procédure d'amélioration incrémentale des bornes inférieures. Cette méthode réduit les temps de calcul sur les grandes instances allant jusqu'à un gain d'un ordre de grandeur. Enfin, une analyse bi-objectif, basée sur la méthode ε-contrainte, a permis d'explorer le compromis entre la durée totale du projet ($C_{\max}$) et la stabilité du profil de charge. L'étude des fronts de Pareto obtenus révèle une corrélation entre la variabilité des charges et le temps d'achèvement du projet.

Mots-clés : Planification tactique, RCCP, MIP, Nivellement de ressources, Optimisation bi-objectif, Profil de charge unimodal, CPLEX.

ABSTRACT

This thesis addresses the issue of *tactical planning of projects* through the study of the *Rough-Cut Capacity Planning (RCCP)* problem. This type of model applies to the early planning phases of projects and aims to optimize, for each period, the execution intensities of the work packages required for project completion. The objective is to ensure consistency between project requirements and the availability of critical resources over a medium-term horizon, prior to the detailed scheduling phase.

The general objective of this research is to propose new mixed-integer linear programming (MIP) formulations to efficiently solve the RCCP problem and its extensions, while improving both the solution quality and the numerical performance of existing models. The work is based on an in-depth analysis of modern solver behavior and on a systematic reformulation of constraints, in order to reduce model complexity, enhance computational performance, and thereby enable the treatment of more complex extensions, which are also part of the objectives of this research.

The first contribution of this thesis consists in developing a new continuous-time (CT) MIP formulation for the *RCCP*, grounded in a detailed analysis of the solver's computational behavior. The study of the *CPLEX* solver revealed several sources of inefficiency, notably the symmetry of solutions that hinders the progress of the lower bound, and the weakness of the linear relaxation after the presolve phase. Experimental results show an average reduction of *72 %* in the number of dual simplex iterations and a significant decrease in the *optimality gap*, enabling the resolution of larger instances than those solvable with the base model. This contribution also highlights the relevance of solver behavior analysis as a lever for improving MIP formulations. Finally, a bi-objective approach combining the *Time-driven* and *Resource-driven* variants was proposed to address planning contexts closer to real industrial applications.

The second contribution aims to further enhance the formulation by refining the *Time-driven* variant, addressing two main issues: numerical instability and the weakness of the linear relaxation after presolve. To this end, a comparative study was conducted among several RCCP formulations to assess their structural performance. Eight models were compared according to two main criteria: solving time and solution quality. Results show that the continuous-time formulation based on *step variables* and the addition of continuous variables outperforms other continuous-time versions across all criteria studied. It provides tighter bounds, a smaller branching tree, and faster convergence to optimality. A theoretical analysis

was also conducted among the tested models, demonstrating that the quality of the linear relaxation is only one of several factors influencing overall model performance. On average, the proposed model is about seven times faster than the best previous continuous formulation, and the solution quality is significantly higher, achieving up to a 67 % improvement in cost in some instances. This study demonstrates that a continuous-time model can rival and even surpass discrete-time formulations, often preferred in the literature for their lower numerical complexity, while maintaining greater flexibility in time representation and in handling precedence relations, particularly when periods are aggregated. The comparative analysis also identified the structural components that most influence solver behavior, notably the precedence relations and the linkage between periods and the start and finish dates of work packages, thereby offering methodological insights for the design of more efficient models.

The third contribution of this thesis focuses on the integration of *resource leveling* into the RCCP model. This part aims to smooth workload fluctuations over the planning horizon in order to avoid periods of overload and underutilization, which reduce productivity and increase labor-related costs. To achieve this, an extended model was proposed, allowing non-uniform execution intensities among resources. Numerical experiments show that this model achieves an average reduction of 35 % in inter-period variance while maintaining the overall project duration. A constraint enforcing a unimodal workload profile was also integrated, ensuring that the workload increases to a peak and then decreases until the end of the planning horizon. This approach prevents cycles of hiring and layoffs, stabilizes the overall workload, and provides a more realistic representation of organizational behavior. An incremental solution strategy was also developed to enhance the computational efficiency of the extended model. This strategy progressively adjusts the planning horizon and incorporates an incremental improvement procedure for lower bounds. The method reduces computation times for large instances by roughly an order of magnitude. Finally, a bi-objective analysis based on the ε-constraint method was conducted to explore the trade-off between total project duration ($C_{\max}$) and workload stability. The analysis of the resulting Pareto fronts reveals a clear correlation between workload variability and project completion time.

Keywords: Tactical planning, RCCP, MIP, Resource leveling, Bi-objective optimization, Unimodal workload profile, CPLEX.

TABLE DES MATIÈRES

LISTE DES TABLEAUX

LISTE DES FIGURES

LISTE DES SIGLES ET ABRÉVIATIONS

Acronyme	**Signification**
CT	*Continuous Time*
CPM	*Critical Path Method*
DT	*Discrete Time*
FRCPSP	*Resource-Constrained Project Scheduling Problem with Flexible Resource Profile*
GA	*Genetic Algorithm*
HM	*Hybrid Metaheuristic*
IG	*Iterated Greedy*
LNS	*Large Neighborhood Search*
LP	*Linear Programming*
LR	*Linear Relaxation*
MIP	*Mixed-Integer Programming*
MRCPSP	*Multi-mode Resource-Constrained Project Scheduling Problem*
PARCPSP	*Periodically Aggregated Resource-Constrained Project Scheduling Problem*
PERT	*Program Evaluation and Review Technique*
PLNE	*Programme linéaire en nombres entiers*
RCCP	*Rough-Cut Capacity Planning*
RCPSP	*Resource-Constrained Project Scheduling Problem*
RCPSVP	*Resource-Constrained Project Scheduling Problem with Variable Intensities*
RLP	*Resource Leveling Problem*
WARCPSP	*Aggregated Resource-Constrained Project Scheduling Problem with Window of Time*
WP	*Work Package*

AVANT-PROPOS

Note sur la thèse

Cette thèse présente une part importante des recherches réalisées au cours de mes cinq années de doctorat à *Polytechnique Montréal*, au sein du *Groupe d'Études et de Recherche en Analyse des Décisions (GERAD)* et du *Centre interuniversitaire de recherche sur les réseaux d'entreprise, la logistique et le transport (CIRRELT)*, sous la direction du *professeur François Soumis* et du *professeur Robert Pellerin.*

Les travaux s'inscrivent dans le domaine de la *planification tactique des projets* et portent sur l'utilisation de la *programmation en nombres entiers mixtes (MIP)* pour résoudre le *problème de Rough-Cut Capacity Planning (RCCP).* L'objectif général est d'améliorer la planification tactique en développant de nouvelles formulations mathématiques pour le problème RCCP, en analysant leur comportement numérique et en explorant leur applicabilité dans des contextes industriels réels.

Ce manuscrit est composé de trois articles scientifiques :

1. **Anis Noureddine**, **François Soumis**, **Robert Pellerin**. *A New MIP RCCP Model for Tackling Tactical Project Planning.* Publié dans *International Journal of Production Research* en décembre 2024 et présenté partiellement à la *Journée de l'Optimisation* (mai 2022) ;
2. **Anis Noureddine**, **François Soumis**, **Robert Pellerin**. *A New Time-Driven Mixed-Integer Programming Model for Rough-Cut Capacity Planning.* Soumis à *Computers & Operations Research* en Mai 2025, révisé en Novembre 2025 et présenté à la *Journée de l'Optimisation* (mai 2024) ;
3. **Anis Noureddine**, **Robert Pellerin**, **François Soumis**, **Raphaël Bourdreault**. *Optimizing Tactical Project Planning with a Resource-Leveled RCCP Model.* Soumis à *OR Spectrum* en octobre 2025 et présenté partiellement à la *Journée de l'Optimisation* (mai 2025).

Cette recherche a été réalisée en collaboration avec l'entreprise *Thales*, partenaire industriel du projet. Cette collaboration a permis de valider certaines hypothèses dans un contexte appliqué et d'enrichir les analyses par des considérations issues de cas réels de planification tactique.

Cette thèse retrace un parcours dédié à la compréhension du problème étudié, à la modélisation et à l'optimisation de systèmes décisionnels au niveau tactique, fondé sur la programmation en nombres entiers mixtes.

Travaux connexes et perspectives

Un travail complémentaire portant sur la variante *Resource-Driven* du problème RCCP est actuellement en préparation en vue d'une soumission dans un journal international. Il a pour objectif de développer une approche destinée à accroître l'efficacité du modèle proposé dans ce contexte particulier, en s'appuyant sur les acquis de la présente thèse.

Par ailleurs, un stage de recherche a été réalisé au sein d'*Hydro-Québec*, portant sur la *réoptimisation robuste du réseau de distribution* à l'aide de la programmation en nombres entiers mixtes, ainsi que sur la *prévision de la demande* à l'aide de modèles d'apprentissage supervisé. Ce travail, mené en collaboration avec *Rachid Hassani*, s'inscrit dans un contexte de planification stratégique.

CHAPITRE 1 INTRODUCTION

Depuis les temps anciens, l'humanité s'est distinguée par sa capacité à concevoir et réaliser des projets de grande ampleur. La construction des pyramides d'Égypte, mobilisant des milliers d'ouvriers pendant plusieurs décennies [64], ou l'édification des cathédrales gothiques, s'étalant parfois sur plus d'un siècle et impliquant des générations entières d'artisans spécialisés [93], témoignent de cette ambition collective. Derrière la fascination qu'inspirent ces ouvrages demeure une nécessité fondamentale : planifier les ressources, anticiper les étapes et coordonner les efforts humains sur des horizons temporels étendus.

Avec la révolution industrielle, le besoin de planification s'est fortement amplifié. Les chemins de fer, les ponts, les réseaux de transport ou encore les grands projets de construction [29] ont imposé une organisation plus stricte, annonçant l'émergence de méthodes scientifiques de gestion [77]. Aujourd'hui, cette exigence continue avec des projets de plus en plus grands : la réparation et la modernisation de navires civils et militaires menées par des entreprises comme Thales [104], ou encore les projets énergétiques développés par Hydro-Québec, tels que les barrages hydroélectriques et les réseaux intelligents [55]. La demande ne cesse d'augmenter. Ces projets, par leur taille et leur complexité, dépassent les capacités d'une organisation fondée uniquement sur l'expérience. Leur réussite exige désormais de recourir à des outils d'aide à la décision. Au cœur de ces outils, les *méthodes d'optimisation* sont devenues essentielles pour planifier adéquatement la charge de travail, utiliser efficacement les ressources et limiter les incertitudes liées à l'exécution de projets complexes.

La planification des projets s'est progressivement organisée autour de trois niveaux complémentaires [5]. Le premier, dit *stratégique*, consiste à anticiper la demande à long terme, à fixer les grandes orientations et à déterminer les capacités globales nécessaires. Le deuxième niveau, qualifié de *tactique*, vise à traduire ces orientations en un plan intermédiaire. Il s'agit de regrouper les activités en lots de travaux et de procéder à une allocation des ressources sur un horizon de moyen terme, sans disposer encore d'une connaissance fine des charges de travail. Enfin, au troisième niveau, celui de la *planification opérationnelle*, les plans stratégiques et tactiques sont déclinés en un ordonnancement détaillé des activités. Cette étape, qui s'inscrit dans le court terme, repose sur des données plus précises et permet d'établir un échéancier complet garantissant la coordination quotidienne des ressources.

La planification tactique a été traitée de différentes manières, parfois par une approche intégrée de planification et d'ordonnancement, où chaque projet requiert une adaptation spécifique [81, 82]. D'autres reposent sur le problème de Rough-Cut Capacity Planning

(RCCP), une méthode qui alloue les ressources à partir de données agrégées et d'intensités variables [32, 39, 47]. Ce dernier s'est ainsi imposé comme un outil essentiel [39]. Il permet de vérifier la faisabilité des plans en tenant compte des ressources critiques, offrant ainsi un cadre réaliste pour assurer la cohérence entre les besoins du projet et les capacités des ressources disponibles [106].

Le problème RCCP diffère fondamentalement du problème Resource-Constrained Project Scheduling Problem (RCPSP) qui est centré sur la planification détaillée des activités avec des dates précises de début et de fin. Le RCCP agit à un niveau tactique supérieur en utilisant des périodes de temps agrégées et des capacités globales pour évaluer la faisabilité des plans. Ces modèles visent généralement à : (1) minimiser la durée du projet sous des capacités de ressources fixées (variante Resource-driven), (2) minimiser le coût des ressources externes pour une durée donnée (variante Time-driven), ou (3) trouver un compromis entre ces deux objectifs [47].

Un enjeu majeur de la planification tactique réside dans le nivellement des ressources. À ce niveau, il s'agit d'anticiper, sur un horizon à moyen terme, la manière dont les ressources critiques seront mobilisées. Une planification mal équilibrée peut conduire à des périodes de surcharge, générant des retards et des surcoûts liés aux heures supplémentaires, ou, au contraire, à des périodes de sous-utilisation, entraînant un gaspillage de capacités disponibles. Le nivellement vise ainsi à lisser l'utilisation des ressources dans le temps afin de stabiliser la charge et d'assurer une meilleure maîtrise des coûts.

La planification tactique est également confrontée à d'importantes incertitudes. L'horizon tactique s'étend souvent sur plusieurs semaines ou mois, période où la demande peut fluctuer, où des retards d'approvisionnement peuvent survenir et où l'indisponibilité de certaines machines ou compétences peut remettre en cause la faisabilité initiale du plan. Pour cette raison, les approches de planification tactique doivent être conçues non seulement pour générer des plans optimisés, mais aussi pour être robustes face aux perturbations et parfois suffisamment flexibles pour s'adapter à de nouveaux scénarios.

De plus, la planification tactique se caractérise par un arbitrage multi-critères. En effet, à ce niveau, il s'agit de prendre simultanément en compte la durée totale du projet, la réduction des coûts liés aux ressources, la qualité de service et la stabilité de la charge de travail. Cette multiplicité d'objectifs rend nécessaire l'utilisation de modèles multi-objectifs et de méthodes d'optimisation capables de générer des compromis acceptables pour les décideurs.

Enfin, l'un des défis les plus pressants porte sur le dimensionnement à grande échelle. Les projets modernes, qu'il s'agisse de centrales électriques, de chantiers navals ou d'infrastructures urbaines, impliquent des centaines d'activités, de ressources et de contraintes. De tels problèmes posent un défi aux méthodes exactes, dont la complexité combinatoire augmente rapidement avec la taille du projet. Dans ce contexte, l'enjeu consiste à développer des formulations et des stratégies de résolution capables de produire, en temps raisonnable, des solutions de qualité, tout en conservant autant que possible les garanties d'optimalité des approches exactes.

Dans un contexte où les solveurs d'optimisation deviennent chaque année plus performants, le choix de la formulation mathématique la mieux adaptée aux caractéristiques du solveur revêt une importance cruciale. Ce choix demeure toutefois relatif, car il ne dépend pas uniquement de critères théoriques, tels que la qualité de la relaxation linéaire, mais aussi de la manière dont le solveur interprète et transforme la structure du modèle. En effet, les solveurs modernes intègrent un large éventail de techniques internes, telles que des heuristiques, des procédures de réduction, des modules de prétraitement et d'autres méthodes capables de transformer de manière significative la matrice initiale du problème.

L'accès à des informations détaillées sur le déroulement de l'algorithme permet une meilleure compréhension des limites rencontrées lors de la résolution. Cette analyse constitue aujourd'hui un axe efficace pour améliorer les performances d'un modèle. Combinée à une compréhension approfondie des pratiques susceptibles d'impacter négativement le comportement du solveur, elle permet d'identifier les sources de difficulté, de modifier certaines contraintes ou encore d'ajouter des coupes efficaces, afin de réduire le nombre de nœuds explorés et de renforcer la relaxation linéaire au sein de l'arborescence de branchement.

Dans ce contexte, la thèse s'articule autour du développement d'un *nouveau modèle en nombres entiers (MIP) pour le RCCP*, conçu pour traiter efficacement des instances de grande taille et capable de supporter des extensions qui ajoutent des éléments de complexité, le nivellement constituant un cas de figure.

Les chapitres de cette thèse s'organisent comme suit. Le chapitre 2 consiste à faire une revue de la littérature sur le problème RCCP ainsi que ceux s'y rapprochant. Nous terminons avec une analyse critique de ces travaux antérieurs dans un but d'identifier des opportunités de recherche. Ensuite, nous énonçons, dans le chapitre 3, nos objectifs de recherche ainsi que la démarche scientifique que nous adoptons. Les chapitres 4, 5 et 6 constituent nos trois contributions présentées sous la forme d'articles scientifiques. Ces trois contributions sont principalement axées sur l'accélération de la résolution à travers l'amélioration des modèles linéaires en nombres entiers, ainsi que sur le développement de stratégies de résolution itératives. La première contribution vise à identifier les limites du modèle de base et à proposer des ajustements visant à renforcer son efficacité de résolution à l'aide d'un solveur MIP. Ces améliorations ont également permis d'aborder le cas bi-objectif, ouvrant la voie à l'analyse d'autres anomalies et à celle de la structure du modèle. Les modèles proposés ont ensuite été comparés à ceux de la littérature afin d'évaluer leurs performances respectives. Enfin, le modèle le plus performant a été étendu pour intégrer le nivellement des ressources, accompagné d'une méthode itérative conçue pour améliorer encore les performances de résolution. Le chapitre 7 discute l'ensemble de ces contributions, tandis que le chapitre 8 présente les conclusions générales et les perspectives de recherche.

CHAPITRE 2 REVUE DE LA LITTÉRATURE

Ce chapitre vise à situer le travail de recherche dans son contexte scientifique. Nous commençons par définir les concepts de base couramment utilisés dans la littérature, avant de présenter les principaux modèles de planification et d'ordonnancement associés au problème de RCCP. L'objectif de cette revue est de mettre en lumière les contributions majeures relatives au RCCP, ses extensions et ses approches de résolution, ainsi que les travaux portant sur le nivellement des ressources. Une analyse critique permettra d'identifier les limites des approches existantes et de positionner les contributions du présent travail de recherche par rapport à l'état de l'art.

2.1 Définitions des concepts de base

L'objectif de cette section est d'introduire les concepts nécessaires à la compréhension du problème de planification tactique étudié, notamment le *RCCP*, le nivellement des ressources et les méthodes exactes et approchées employées pour leur résolution.

2.1.1 Planification hiérarchique

La *planification hiérarchique* désigne la décomposition du processus de planification en plusieurs niveaux interdépendants, différenciés selon l'horizon temporel, le niveau de détail et le type de décision à prendre. De Boer, R (1998) [39] distingue quatre niveaux, repris dans [48, 69] (Figure 2.1) :

— *Planification stratégique* : concerne les décisions à long terme telles que la prévision de la demande en projets, la détermination des capacités globales et la planification des investissements en ressources critiques ;

— *Planification tactique* : s'appuie sur des données agrégées pour vérifier la faisabilité des plans de charge. Elle vise à équilibrer les besoins en ressources et les capacités disponibles à moyen terme, notamment à travers le *RCCP* ;

— *Planification tactique/opérationnelle* : correspond à la planification détaillée des activités et à la gestion des contraintes de ressources, souvent modélisée par le *RCPSP* ; et

— *Planification opérationnelle* : traite l'ordonnancement fin des tâches et l'affectation des ressources spécifiques (machines, opérateurs) sur un horizon de court terme.

Ces niveaux sont interconnectés : les décisions prises à un niveau supérieur constituent

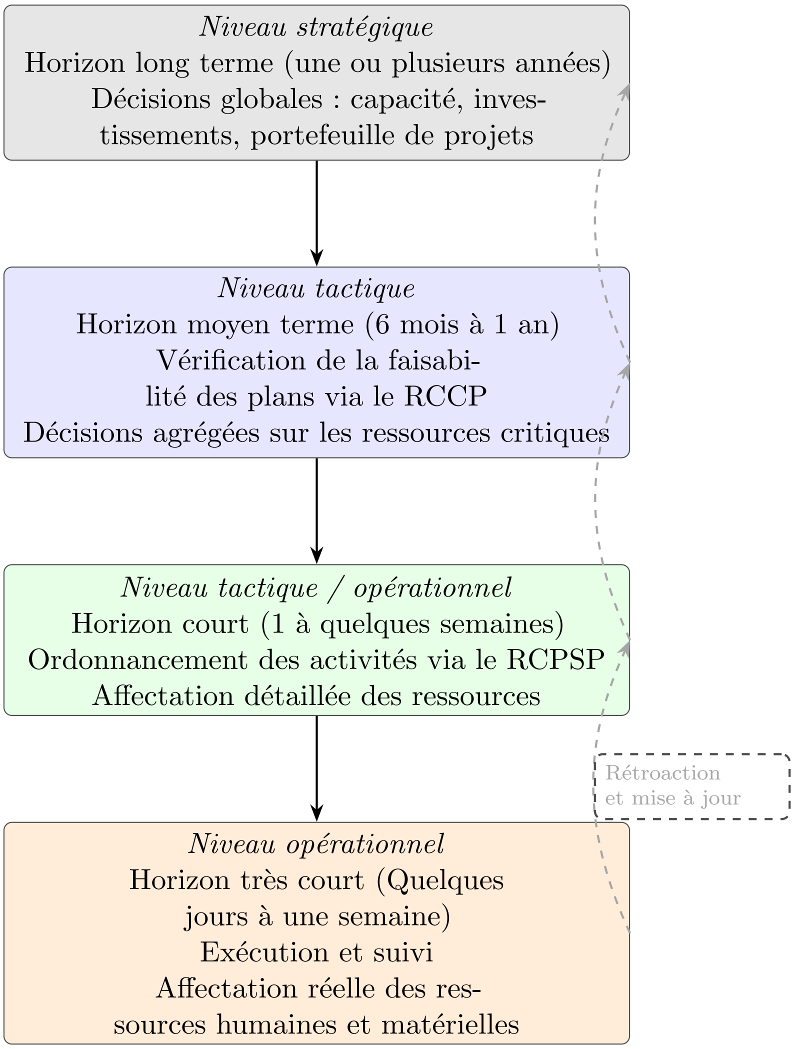

FIGURE 2.1 Structure hiérarchique de planification selon De Boer (1998)

des contraintes ou des objectifs pour le niveau inférieur. Une mise à jour périodique des plans à chaque niveau est nécessaire pour maintenir la cohérence globale du système de planification [30].

2.1.2 Problèmes d'ordonnancement

Un problème d'ordonnancement (*scheduling problem*) consiste à déterminer l'ordre et les dates d'exécution d'un ensemble d'activités ou de tâches, en respectant un ensemble de contraintes. Ces contraintes peuvent porter sur la précédence entre activités, les durées ou la disponibilité des ressources. Parmi les objectifs considérés, on peut citer :

— la minimisation de la durée totale du projet (*makespan*) ;

— la minimisation des retards par rapport aux échéances fixées ; et
— la minimisation du temps moyen d'achèvement des tâches.

L'ordonnancement intervient à *court terme* : il s'intéresse à la séquence exacte d'exécution des activités et à leur affectation précise aux ressources disponibles [25, 89].

Le *Resource-Constrained Project Scheduling Problem (RCPSP)* est la forme classique du problème d'ordonnancement sous contraintes de ressources. Chaque activité possède une durée déterminée et consomme un certain nombre d'unités de ressources. Les contraintes de précédence doivent être respectées et les ressources ne peuvent être utilisées que dans la limite de leur capacité. L'objectif le plus courant est la minimisation du *makespan* [8, 19]. Le RCPSP constitue ainsi un modèle de référence pour la planification opérationnelle (tactique-opérationnelle selon la structure de De Boer (1998)), où chaque activité est explicitement définie par ses dates de début et de fin.

2.1.3 Problèmes de planification

La *planification* s'inscrit à un niveau *plus global et à plus long terme* que l'ordonnancement. Elle vise à élaborer un plan global d'utilisation des ressources et de réalisation des projets, en s'appuyant sur des données agrégées (groupes d'activités, capacités moyennes, périodes de planification). Son objectif principal est d'assurer la *faisabilité globale* des projets en amont, avant de passer à la planification détaillée [5, 39, 47].

Le *Rough-Cut Capacity Planning (RCCP)* est un *problème de planification tactique* situé entre le niveau stratégique et le niveau opérationnel. Il vise à vérifier la faisabilité d'un plan de production ou de projet au regard des capacités globales des ressources critiques [32, 39, 47]. Contrairement au RCPSP, qui traite des activités détaillées avec des durées précises, le RCCP manipule des *lots de travail* et des *capacités globales par période*. Son objectif est d'optimiser les coûts du projet et de fournir une estimation agrégée de la charge à moyen terme.

Les formulations du RCCP traitent généralement trois fonctions objectif :

— *Variante Resource-driven* : minimisation de la durée totale sous des capacités fixes ;
— *Variante Time-driven* : minimisation du coût des ressources externes pour une durée donnée ; et
— *Fonction bi-objectif* : recherche d'un compromis entre coût et durée [47, 83].

2.1.4 Nivellement de ressources

Le *Resource Leveling Problem (RLP)* est un *problème de planification* dont l'objectif est de minimiser les *fluctuations de charge* sur les ressources au cours du temps, tout en respectant les contraintes de précédence et la durée totale du projet. Il vise à produire un *profil d'utilisation régulier*. La qualité du plan de charge est généralement mesurée à l'aide d'indicateurs tels que la *variance* de l'utilisation des ressources ou des indices de régularité [107].

Le nivellement des ressources présente plusieurs défis, notamment la réduction des pics de charge, la limitation des variations d'une période à l'autre et la stabilisation globale du profil d'utilisation. Pour contrer un ou plusieurs de ces défis, plusieurs façons de procéder sont possibles, selon les fonctions objectif adoptées. Parmi les plus courantes, on distingue :

— La *minimisation des écarts absolus* entre les charges successives, pour limiter les variations brusques d'une période à l'autre ;

— La *minimisation du pic de charge maximal*, afin d'éviter les situations de surcharge critique ; et

— La *minimisation de la somme des carrés des écarts* par rapport à la charge moyenne, permettant d'obtenir une répartition plus homogène de l'utilisation des ressources.

2.1.5 Programmation linéaire et en nombres entiers

La *programmation linéaire* (PL) consiste à optimiser une fonction objectif linéaire sous un ensemble de contraintes également linéaires [38, 109]. Ce cadre mathématique permet de représenter une grande variété de problèmes décisionnels où les relations entre variables sont proportionnelles et additives. La solution optimale est généralement obtenue à l'aide de l'algorithme du *simplexe* [34].

La *programmation linéaire en nombres entiers* (PLNE) (ou *Mixed Integer Programming (MIP)*) étend ce cadre en imposant que certaines variables soient entières ou binaires [80, 110]. Cette restriction rend le problème *NP-difficile* en général, c'est-à-dire qu'aucun algorithme polynomial n'est connu pour le résoudre. De plus, la complexité de leur résolution croît de manière exponentielle avec la taille du problème [80]. Cet outil permet de modéliser des décisions discrètes telles que l'activation d'une activité, l'allocation d'une ressource ou la sélection d'un mode de fonctionnement. Les modèles MIP sont ainsi largement utilisés pour formuler des problèmes complexes de planification et d'ordonnancement, combinant des décisions continues (capacités, durées, coûts) et discrètes (affectations, séquences, choix de modes). Les solveurs modernes tels que CPLEX, Gurobi ou SCIP exploitent des techniques avancées de branchement, de coupes et de prétraitement pour résoudre efficacement ces modèles à grande échelle [1, 67].

2.1.5.1 Relaxation linéaire

La *relaxation linéaire (RL)* d'un modèle en nombres entiers consiste à supprimer les contraintes d'intégralité imposées sur certaines variables afin d'obtenir un programme linéaire continu, plus facile à résoudre [34, 80]. Cette relaxation fournit une *borne inférieure* (pour un problème de minimisation) ou une *borne supérieure* (pour un problème de maximisation) du modèle original en nombres entiers. Elle est couramment utilisée dans les méthodes exactes, notamment dans les algorithmes de type *Branch and Bound*, pour évaluer la qualité d'une solution entière et guider l'exploration de l'espace de recherche. Une relaxation dite *serrée* (ou *tight relaxation*) est souhaitable, car elle améliore l'efficacité du solveur en réduisant l'écart entre la solution continue et la solution entière optimale.

2.1.5.2 Problèmes de dégénérescence et d'instabilité numérique

Les *problèmes de dégénérescence* apparaissent lorsque plusieurs contraintes actives définissent une même solution de base dans un programme linéaire [34, 38]. Dans ce cas, le solveur peut explorer plusieurs bases équivalentes sans amélioration de la fonction objectif, entraînant des phénomènes d'oscillation ou un ralentissement de la convergence de l'algorithme du simplexe.

Les *instabilités numériques*, quant à elles, proviennent de coefficients mal conditionnés ou de grande amplitude dans la matrice du modèle, rendant les calculs sensibles aux erreurs d'arrondi [18]. Elles peuvent compromettre la précision des bornes, la détection d'infaisabilités ou encore la cohérence du critère d'optimalité dans un solveur.

Les principales causes d'instabilités numériques résident dans la mauvaise échelle des coefficients, la présence de constantes artificiellement grandes (*big-M*) et les relations de contraintes redondantes ou quasi colinéaires. Ces facteurs entraînent un mauvais conditionnement de la matrice du modèle, amplifiant les erreurs d'arrondi et compromettant la précision du processus de résolution.

Symétrie de solutions

La *symétrie des solutions* survient lorsque plusieurs configurations équivalentes du modèle conduisent à la même valeur de la fonction objectif [67]. Cette redondance augmente inutilement la taille de l'arbre de recherche et ralentit la convergence du solveur. Pour y remédier, des contraintes de casse de symétrie (*symmetry breaking constraints*) peuvent être utilisées afin d'éliminer les solutions équivalentes. L'objectif est d'améliorer l'efficacité du processus de branchement.

2.1.5.3 Relation de dominance entre modèles

Un modèle M_1 est dit *dominant* par rapport à un modèle M_2 si la relaxation linéaire de M_1 fournit une borne plus serrée, ou si son espace de recherche est mieux restreint, pouvant conduire à une convergence plus rapide du solveur [67, 110]. En effet, un modèle dominant améliore la qualité des bornes, réduit le nombre de nœuds explorés.

2.1.6 Méthodes exactes

Les *méthodes exactes* garantissent l'obtention d'une solution optimale en explorant l'espace de recherche de manière systématique [1, 80]. Les principales approches en PLNE comprennent :

— *Branch and Bound* : exploration arborescente de l'espace des solutions fondée sur le calcul de bornes inférieures et supérieures ;

— *Méthodes de coupes* : ajout de contraintes valides (par exemple, coupes de Gomory, plans de Chvátal) pour renforcer la relaxation linéaire ;

— *Branch and Cut* : combinaison des deux stratégies précédentes, utilisée par la plupart des solveurs MIP modernes (CPLEX, Gurobi, SCIP) ; et

— *Méthodes de décomposition* : séparation du problème maître et des sous-problèmes, notamment par des décompositions de Benders et de Dantzig–Wolfe [97].

2.1.7 Méthodes approchées

Les *méthodes approchées* regroupent les approches de résolution qui ne garantissent pas l'obtention d'une solution optimale. Leur objectif principal est de fournir, en un temps de calcul raisonnable, une solution de bonne qualité pour des problèmes où les méthodes exactes deviennent inefficaces [85].

Ces méthodes se divisent en deux grandes familles :

— les *heuristiques*, qui reposent sur des règles ou procédures spécifiques au problème traité et permettent de construire ou d'améliorer rapidement une solution réalisable ; et

— les *métaheuristiques*, qui constituent des cadres de recherche plus généraux, capables de guider ou de combiner plusieurs heuristiques afin d'explorer efficacement l'espace des solutions [84].

2.1.7.1 Heuristiques

Les heuristiques exploitent la structure du problème pour orienter la recherche vers des zones prometteuses de l'espace des solutions [112].

Heuristiques de construction : Une *heuristique de construction* élabore progressivement une solution réalisable à partir d'une situation initiale vide, selon une logique souvent gloutonne (*greedy*), où chaque choix est définitif et non remis en cause.

Heuristiques d'amélioration : Les *heuristiques d'amélioration* partent d'une solution réalisable et la modifient itérativement afin d'en améliorer la qualité. La recherche se poursuit jusqu'à ce qu'aucune amélioration locale ne soit possible ou qu'un critère d'arrêt soit atteint.

2.1.7.2 Métaheuristiques

Une métaheuristique est un processus itératif qui supervise et oriente une ou plusieurs heuristiques pour explorer efficacement l'espace de recherche [84]. Ces méthodes combinent des stratégies d'exploration globale et d'exploitation locale, et peuvent intégrer des mécanismes d'apprentissage pour améliorer la performance au fil des itérations.

2.1.8 Optimisation multi-objectif

Un *problème d'optimisation multi-objectif* consiste à optimiser simultanément plusieurs fonctions objectifs définies sur un même ensemble de solutions réalisables. Autrement dit, il s'agit de rechercher un compromis entre plusieurs critères souvent conflictuels.

Formellement, un **problème multi-objectif** est défini par un ensemble fini S et un vecteur d'objectifs $F : S \to \mathbb{R}^p$, formulé comme suit :

$$\min_{s \in S} F(s) = (f_1(s), f_2(s), \ldots, f_p(s)). \tag{2.1}$$

Résoudre ce type de problème revient à identifier les solutions représentant les meilleurs compromis entre les différents objectifs, selon le concept d'*optimalité de Pareto* [35].

2.1.8.1 Dominance au sens de Pareto

Vecteur dominant Pour un problème de minimisation, un vecteur $F(s)$ **domine** un autre vecteur $F(s')$ si :

$$f_i(s) \leq f_i(s'),\ \forall i \in \{1, \ldots, p\}, \quad \text{et} \quad \exists i' \in \{1, \ldots, p\} \ \mid\ f_{i'}(s) < f_{i'}(s').$$

avec $s, s' \in S$.

Solution efficace Une solution $s \in S$ est dite *efficace* (ou *Pareto-optimale*) si son vecteur d'objectifs $F(s)$ n'est dominé par aucune autre solution de S.

S'il existe un vecteur $F(s)$ qui domine tous les autres vecteurs de l'espace des critères [1], alors les critères sont dits **non conflictuels**.

2.1.8.2 Méthode ε-contrainte

La *méthode ε-contrainte* est une approche exacte utilisée pour résoudre les problèmes d'optimisation multi-objectif. Elle consiste à sélectionner l'une des fonctions objectifs comme fonction principale à optimiser, tandis que les autres sont transformées en contraintes bornées par un paramètre ε [72]. En faisant varier ce paramètre, on génère différentes solutions optimales correspondant à des compromis distincts, formant ainsi le *front de Pareto*.

Cette méthode présente l'avantage de produire des solutions non dominées et de garantir une couverture uniforme du front de Pareto lorsque la génération des bornes ε est bien calibrée.

2.2 Revue abrégée de la littérature

Dans cette revue de littérature, nous présentons une synthèse abrégée des travaux les plus pertinents liés au problème du *RCCP*, afin de situer notre recherche par rapport aux contributions existantes.

2.2.1 Modèles RCPSP et extensions

Avant d'aborder le *RCCP*, il est pertinent de rappeler certains éléments fondamentaux du *RCPSP* classique ainsi que de ses principales variantes. Bien que l'objectif de cette recherche ne soit pas de traiter la résolution d'une extension du RCPSP, certaines variantes du problème RCPSP présentent plusieurs similitudes structurelles avec le RCCP (par exemple, dans les travaux de Kis [60], Hans [47], Haït et Baydoun [51], Naber et Kolisch [79]). L'expérience acquise dans la modélisation de ce dernier constitue ainsi un socle méthodologique essentiel pour le développement de nouveaux modèles RCCP, ce qui constitue l'un des objectifs majeurs de cette thèse.

Le RCPSP de base vise à minimiser la date de fin du projet (makespan) sous contraintes de précédence et de ressources. Ce problème, reconnu comme NP-difficile [19], a fait l'objet de nombreux travaux. La littérature propose diverses formulations en programmation linéaire en nombres entiers, souvent renforcées par des techniques améliorant la relaxation linéaire.

1. L'espace des critères est constitué des valeurs des critères obtenues pour toutes les solutions réalisables.

Le modèle discret initial (DT) de Pritsker et al. [90] repose sur une discrétisation du temps et l'utilisation de variables binaires indiquant le moment de début de chaque tâche. Dans un contexte multi-projets, il étend sa formulation à l'aide de variables vérifiant si une tâche est exécutée à une période t ou avant. Christophides et al. [33] renforce ce modèle en modifiant les contraintes de précédence, ce qui accroît leur nombre mais améliore la qualité de la relaxation.

Une autre approche est proposée par Artigues et al. [9]. Elle est basée sur des variables continues (dates de début) et des variables binaires décrivant l'ordre des tâches et associées à des flux de ressources. Cette formulation se révèle plus adaptée lorsque l'horizon temporel est très large. Koné et al. [63] comparent plusieurs modèles, dont ceux de Pritsker et al. [90], Christophides et al. [33], Artigues et al. [9], et introduisent deux formulations supplémentaires : le modèle Start/End (variables binaires de début et de fin) et le modèle On/Off (variables binaires indiquant si une tâche est en cours à l'instant t). Leur comparaison montre qu'aucun modèle n'est globalement dominant : le modèle de Christophides [33] est plus performant lorsque l'horizon reste modéré, tandis que le modèle On/Off s'impose pour de très grands horizons.

Alvarez-Valdés et Tamarit [4] introduisent des variables binaires indiquant l'ordre des activités et des variables entières pour modéliser leurs dates de fin. Mingozzi et al. [73] proposent une formulation discrète basée sur une représentation en chemins, dans laquelle chaque variable correspond à un chemin réalisable dans le graphe d'ordonnancement. Comme le nombre de chemins possibles peut croître de manière exponentielle, le modèle comporte un nombre exponentiel de variables. Pour le rendre exploitable, les auteurs s'appuient sur une relaxation linéaire résolue par génération de colonnes. Cette relaxation sert ensuite à calculer des bornes inférieures de qualité, intégrées dans un schéma de Branch-and-Price. Klein [61] reformule le modèle discret avec un seul type de variables binaires (activité en cours à t), tandis que Bianco et Caramia [14] proposent une formulation combinant variables binaires (début et fin) et variables continues représentant l'avancement des ressources d'une activité. Ces derniers concluent que leur approche surpasse plusieurs formulations de la littérature et fournit des résultats comparables aux meilleures solutions connues.

Même si les meilleures formulations exactes du RCPSP permettent de résoudre efficacement des instances de taille modérée (jusqu'à 60 tâches), les instances de grande taille nécessitent généralement le recours à des heuristiques et métaheuristiques, ce qui explique le recours aux méthodes approchées dans la littérature [88].

De nombreuses généralisations du RCPSP ont été étudiées. L'une des plus connues est le Multi-Mode RCPSP (MRCPSP) [108], où chaque tâche peut être exécutée selon plusieurs modes caractérisés par des durées et des charges de ressources différentes. Talbot [101] a proposé une extension du modèle discret de Pritsker et al. [90] en intégrant un indice supplémentaire pour représenter le mode choisi. D'autres variantes incluent les modèles de Maniezzo et al. [68] (utilisés notamment pour construire une heuristique basée sur Benders), ou encore ceux de Chakrabortty et al. [28], qui généralisent les modèles de Koné et al. [63] et s'avèrent moins sensibles aux longues durées de tâches.

Des extensions portent également sur la variabilité des ressources. Hartmann [50] introduit une demande et des capacités variables selon la période, hypothèse ensuite généralisée par Zimmermann et Trautmann [114] au MRCPSP. D'autres variantes, comme celles de Kis [60] et Naber et Kolish [79], considèrent que la demande doit être satisfaite pour chaque tâche indépendamment du temps (RCPSVP ou FRCPSP). Ces variantes présentent de fortes similarités avec le RCCP lorsque l'on considère des lots de travail plutôt que des tâches, et seront donc détaillées dans la section suivante.

2.2.2 Modèles RCCP, extensions et méthodes de résolution

2.2.2.1 Planification tactique de projets et modèles RCCP

Les modèles RCCP sont des modèles destinés aux premières phases des grands projets visant à estimer les intensités d'exécution des lots de travail à chaque période pour l'exécution du projet. Ce problème a été prouvé NP-difficile par Kis [60] (étudiant un problème équivalent au RCCP, mais considéré comme étant une variante du RCPSP avec intensité de travail variable « RCPSVP », ce qui pourrait être adapté au niveau de planification tactique). Dans la littérature, le modèle de base a été proposé par De Boer [39], dans lequel il propose pour un contexte multi-projet un modèle dans lequel les contraintes de précédence ne sont pas exprimées linéairement. Elles sont formulées au moyen d'une règle conditionnelle imposant qu'un lot ne puisse être exécuté qu'une fois l'ensemble du travail de ses prédécesseurs accompli. À cet effet, son modèle n'est pas directement résolu, son approche de résolution était de relaxer cette contrainte. Le modèle relaxé est alors linéaire et pourrait être facilement résolu. Pour réparer les violations des relations de précédence, il propose de résoudre itérativement le modèle LP-relaxé et de réparer à chaque fois la solution en utilisant une heuristique. Cette façon de résoudre le problème a été proposée initialement pour la variante Time-driven et puis elle a été adaptée à la variante Resource-driven. Quant à Hans [47], il propose un modèle linéaire mixte ayant pour but de minimiser le coût des capacités externes en pénalisant le retard qui pourrait être causé par la limitation de ressources internes et externes. Les variantes Time-driven et Resource-driven sont donc prises en compte dans le même modèle. Les contraintes de précédence sont prises en compte en interdisant qu'un successeur s'exécute en même temps à la même période avec l'un de ses prédécesseurs. Cependant, seule une exception est permise. Étant donné que ce modèle n'exprime pas les dates de début et fin de manière explicite à l'intérieur des périodes, il se pourrait que plusieurs lots liés par des arcs de précédence soient exécutés dans la même période, ce qui ne permettra pas nécessairement de les exécuter en respectant les relations de précédence et ce même si les contraintes de ressources sont respectées. Afin de surmonter cette difficulté, il permet à un lot "i" d'être exécuté sur la dernière période dans laquelle un de ses prédécesseurs "j" demeure en exécution, mais en limitant le nombre de lots qui seront effectués durant la période concernée afin de limiter les violations des contraintes de précédences.

Par ailleurs, Kis [60] propose un modèle mixte linéaire RCPSVP en minimisant les coûts de ressources externes. Afin de traiter les contraintes de précédence, il propose une nouvelle formulation utilisant un groupe de variables binaires afin de vérifier pour chaque lot s'il peut être traité à un instant t. Il tente ensuite de renforcer les relations de précédence et adapte son modèle au cas de minimisation de la date de fin du projet. Gademann et Schutten [42] ainsi que Wullik et al. [111] utilisent le même modèle, mais proposent différentes méthodes de résolution de programmes linéaires en nombres entiers, notamment celles de décomposition en sous-problèmes plus simples. Cette démarche est due au fait que les relations de précédence dans le modèle engendrent un grand nombre de variables, ce qui induit une difficulté lors de la résolution du modèle relaxé d'une part et des problèmes d'insuffisances de mémoire d'autres parts.

Haït et Baydoun [51] proposent d'exprimer les dates de début et fin au moyen de variables continues pour contourner les difficultés trouvées dans les modèles précédents au niveau des contraintes de précédence, en considérant deux groupes de variables binaires afin de décider pour chaque lot s'il commence ou termine avant ou dans une période p, nécessaire pour l'évaluation périodique des charges des lots. L'avantage principal de ce modèle réside dans le fait que les relations de précédence sont exprimées explicitement. Aucune violation de contraintes de précédence n'est possible au cours d'une même période. De plus, nous n'avons pas besoin de sur-exiger que 2 lots $(i,j) \in E$ ne peuvent être dans une même période puisque nous pouvons exprimer ces contraintes avec des variables continues. Cherkaoui [32] généralise ce modèle dans un contexte de grands projets de construction en variant les durées des périodes, c'est-à-dire que les périodes dans ce cas ne seront pas toujours de longueurs égales.

Naber et Kolish [79] comparent entre 4 modèles linéaires mixtes afin de résoudre le problème FRCPSP (RCPSP with flexible resource profiles) qui est une variante de RCPSP avec la possibilité que la capacité des ressources varie au cours du temps. Dans leur comparaison, ils se basent sur les modèles de Kis [60] dédié au problème RCPSVP et celui de Bianco et Caramia [14] pour le problème RCPSP. Ils considèrent également une variante du RCPSP obtenue en introduisant un ensemble de variables binaires indiquant, pour chaque lot, s'il est en cours d'exécution durant une période p. Ce groupe de variables binaires s'inspire du modèle proposé par Klein [61]. Un autre modèle utilise deux groupes de variables binaires consistant à décider pour chaque lot s'il commence (resp. termine) au début d'une période p. Cette formulation est basée sur les travaux de Ranjbar et Kianfar [91]. Le dernier modèle est basé sur les travaux de Sabzehparvar et Seyed-Hosseini [92] qui ont proposé un modèle géométrique afin de résoudre le problème MRCPSP avec contraintes de précédence généralisées. De ces quatre modèles, les auteurs concluent que le modèle basé sur les travaux de Kis [60] et de Bianco et Caramia [14] se révèle de meilleure qualité au niveau du temps d'exécution ainsi que de la qualité des solutions trouvées par ces deux modèles.

Naber [78] se base sur le modèle discret de Naber et Kolish [79] ainsi que le modèle RCPSP de Koné et al. [63] afin de proposer un modèle continu basé sur les événements. Un événement dans son modèle pourrait être : un début, une fin d'exécution ou un changement au niveau de la consommation de ressources. Il conclut que son modèle fournit une meilleure solution que le modèle discret, ce qui n'est pas le cas en ce qui concerne le temps d'exécution.

2.2.2.2 Méthodes de résolution du problème RCCP

Vu que le problème est NP-difficile [60], la communauté scientifique propose dans la littérature différentes approches de résolution : méthodes exactes, algorithmes constructifs, recherche locale et méta-heuristiques. À noter que plusieurs méthodes étudiées sont basées sur l'hybridation de méthodes exactes et d'heuristiques. Parmi les heuristiques proposées, De Boer [39] propose une heuristique nommée ICPA (Incremental capacity planning algorithm). Elle consiste à programmer le plus de lots de travail possible avec les ressources internes. Les lots restants sont planifiés avec l'utilisation des ressources externes en minimisant les coûts associés à cette utilisation. De plus, le même auteur propose un autre type d'heuristiques, ceux basés sur la relaxation linéaire du modèle fournissant une solution initiale, qui pourrait être réalisable ou non réalisable pour le problème RCCP. Le but de ces heuristiques consiste en l'amélioration d'une solution initiale réalisable ou la réparation des solutions non réalisables. À cet effet, il propose une heuristique basée sur la LP-relaxation afin de réparer les solutions initiales fournies par un programme linéaire correspondant ne prenant pas en compte les relations de précédence. Ainsi, l'heuristique utilisée essaye de réparer la solution au niveau des contraintes de précédence.

Gademann et Schutten [42] comparent deux heuristiques basées sur la LP-relaxation dans un contexte multi-projet. Afin de faire face principalement aux contraintes de précédence, les auteurs proposent de réduire le problème original à un problème LP en introduisant des fenêtres de temps nommées « ATW » dans lesquelles les lots peuvent s'exécuter. Trois catégories d'heuristiques sont proposées : des heuristiques de construction d'une solution initiale, celles réparant les solutions non réalisables et celles améliorant une solution réalisable. Par ailleurs, Wullink et al. [111] propose à son tour plusieurs heuristiques basées sur ses deux procédures : LAP (Largest Activity Part) et ALAP (Adaptative search LAP), fournissant des solutions réalisables. Ces procédures sont combinées avec le modèle LP associé au RCCP.

Dans le même contexte, Gademann et Schutten [42] proposent trois catégories d'heuristiques. Des heuristiques constructives basées sur le programme linéaire essayant de trouver des fenêtres de temps avec lesquelles leur programme linéaire proposé puisse trouver une solution réalisable. Une autre catégorie d'heuristiques a été utilisée, celle qui répare les solutions non réalisables afin d'obtenir une solution réalisable. Une 3ème catégorie d'heuristiques a été proposée en essayant d'améliorer une solution réalisable en modifiant les fenêtres de temps. Une autre façon d'y procéder est proposée par Masmoudi et Haït [69], il utilise la méthode du recuit simulé visant à faire une recherche locale pour améliorer la solution courante et l'appliquant sur le modèle de Hans [47].

Hans [47] propose une méthode de Branch and Price et inclut ainsi la génération de colonnes dans son approche. Le développement d'une telle méthode est dû au fait que lorsque le nombre de lots augmente, une simple méthode exacte de branch and bound a des difficultés pour résoudre le problème RCCP étant donné que le nombre de variables et de contraintes deviennent très grand.

Tritschler et al. [105] proposent une métaheuristique hybride (HM) pour le problème FRCPSP en discrétisant le temps. Le HM considéré consiste à utiliser l'algorithme génétique (GA) hybridé à l'heuristique de recherche à voisinage variable (VNS).

2.2.2.3 Quelques extensions du problème RCCP

Plusieurs tentatives d'améliorations des hypothèses de RCCP ont été proposées dans la littérature pour satisfaire aux besoins des entreprises en pratique. Kis [59] définit des relations de précédence comme suit. Soit (i,j) dans E, alors : le lot j ne peut commencer avant l'exécution d'un certain pourcentage du lot i. Ceci implique l'ajout de variables binaires. Citons aussi les travaux de Baydoun et al. [12] consistant à proposer une extension du modèle Time-Driven RCCP ayant pour but de traiter la possibilité de permettre le chevauchement entre deux lots selon différents modes d'exécution en considérant des charges additionnelles s'ajoutant aux charges initiales des lots. Ces charges sont nommées « retouches ». Chaque mode représente le pourcentage de charges du lot exécuté dans lequel ses lots successeurs peuvent commencer. Alfieri et al. [6] effectuent des études dans le même sens en proposant d'autres types de relations de précédence.

Soulignant que la majorité des auteurs ont proposé des modèles déterministes pour le RCCP, un autre type de tentatives d'améliorations a été étudié, visant à contourner les difficultés dues aux imprévus pouvant se dérouler lors de l'exécution du projet. Wullink et al. [111] propose d'inclure dans son modèle le fait qu'il y a des incertitudes dans l'estimation des charges. Il exécute à cet effet plusieurs modes pour les lots qu'ont un grand impact sur la qualité de la solution lors des modifications liées à l'incertitude et des imprévus. L'objectif est de minimiser le coût du projet. Étant donnée la complexité du modèle résultant, il propose de résoudre le problème en utilisant l'heuristique proposée par Gademann et Schutten [42]. Par ailleurs, Masmoudi et Haït [69] tente de faire face aux imprévus en considérant une distribution continue. Cherkaoui et al. [32], afin de faire face aux imprévus liés à l'agrégation des données, propose de considérer une certaine marge sur les capacités des ressources. Elle étudie à cet effet l'impact de cette procédure sur la qualité des solutions trouvées.

2.2.3 Nivellement de ressources

La littérature sur le lissage des ressources propose une variété d'approches, allant des heuristiques locales aux formulations exactes. Les méthodes heuristiques incluent notamment des procédures de recherche locale visant à décaler les activités non critiques pour réduire les fluctuations de charge (Leu et Yang [65], Hegazy [53]), parfois renforcées par des métaheuristiques comme le recuit simulé ou les algorithmes génétiques (El-Rayes et Jun [40], Senouci et Eldin [94]). Ces approches visent surtout la réduction de la variance de l'utilisation des ressources ou du pic de charge afin d'obtenir des profils plus réguliers.

Du point de vue des indicateurs, El-Rayes et Jun [40] introduisent le *Release and ReHire Index* (RHH) et le *Resource Idle Days Index* (RID) pour mesurer respectivement l'instabilité de la main-d'œuvre et les jours d'inactivité. Dans le même contexte, Wanjari et Tawalare [107] comparent plusieurs métriques et soulignent l'intérêt d'un profil *unimodal*, montée jusqu'à un pic, puis décroissance, comme cible de référence.

En ce qui concerne les méthodes exactes, Mattila et Abraham [70] formulent un modèle MIP pour des projets linéaires (routes, pipelines) en minimisant les écarts absolus de charge entre périodes successives, en intégrant les techniques de type *Critical Path Method* (CPM). Des méthodes exactes dédiées au nivellement sont également étudiées par Gather et Zimmermann [44], qui analysent et résolvent différentes variantes par des schémas exacts. Sur un cas industriel, Damci et Polat [37] comparent plusieurs fonctions objectif et montrent qu'optimiser un critère ne garantit pas l'amélioration des autres, ce qui plaide pour des approches multi-objectifs.

Lorsque les intensités d'exécution sont variables, Bianco, Caramia et Giordani [15] proposent un modèle de programmation mathématique intégrant des précédences généralisées et des intensités variables, résolu par relaxation lagrangienne pour limiter le recours aux ressources externes. Les mêmes auteurs [16] introduisent un coût total d'ajustement dans un MIP où durées et intensités sont décisionnelles. À l'autre extrême, Tarasov, Haït et Battaïa [102, 103] permettent des intensités pleinement indépendantes par ressource (cadre RCPSVP), ce qui accroît la flexibilité mais peut nuire à la coordination entre corps de métiers.

En ce qui concerne les méthodes approchées, Ballestín, Schwindt et Zimmermann [10] proposent une *Iterated Greedy* (IG) pour des contextes *make-to-order* et étudient plusieurs objectifs (somme des carrés, écarts absolus à la moyenne, pic de charge). Masmoudi et Haït [69] combinent algorithmes génétiques et logique floue pour des opérations de maintenance d'hélicoptères. Hegazy [53] recourt à un algorithme génétique (GA) pour traiter simultanément allocation et nivellement, tandis que Son et Skibniewski [96] proposent une approche multi-heuristique appuyée par un recuit simulé.

2.2.4 Analyse critique de la littérature

La littérature oppose depuis longtemps la garantie d'optimalité et la capacité de résoudre des problèmes de grande taille. Les méthodes exactes trouvent des solutions optimales sur des problèmes de taille moyenne, parfois petite pour des projets réels, tandis que les heuristiques et métaheuristiques gèrent mieux les grands cas, mais sans garantie d'optimalité. Au-delà de ce constat classique, les temps de calcul dépendent avant tout de la formulation du modèle et de la méthode de résolution adoptée.

Dans le domaine du RCCP, rares sont les travaux qui analysent en profondeur le comportement du solveur afin d'en tirer des enseignements pour améliorer la performance ou ajuster les contraintes. Bien souvent, ces dernières sont simplement déduites de la structure théorique du modèle, sans considération pour leurs effets pratiques sur la résolution au niveau du solveur. Cependant, ces outils, devenus au fil du temps de plus en plus performants, sont désormais capables de détecter ou d'induire certaines contraintes de manière implicite. Il devient donc essentiel de comprendre ce que le solveur déduit déjà automatiquement, afin de n'ajouter que les contraintes réellement manquantes et utiles à la convergence.

De plus, les formulations en temps continu offrent généralement une meilleure flexibilité et une représentation plus réaliste du processus. Cependant, elles demeurent encore peu exploitées dans le contexte du RCCP, principalement en raison d'un temps d'exécution souvent plus élevé [78]. Cependant, étant donné le peu de formulations proposées, ce constat pourrait être remis en cause, notamment du fait que ces modèles sont généralement composés d'un mixte de variables entières et continues, engendrant éventuellement des anomalies au niveau de la résolution par les solveurs MIP. En réalité, plusieurs leviers d'amélioration comme la gestion des symétries, le renforcement de la relaxation linéaire ou la stabilisation numérique peuvent être identifiés à partir des informations fournies par les solveurs eux-mêmes.

D'autre part, aucune étude ne compare l'impact réel de considérer le temps continu relativement au temps discret, en termes de temps d'exécution et de gain réel au niveau de la qualité de la solution. Il n'y a pas non plus de comparaison entre divers modèles avec diverses combinaisons de variables et de contraintes permettant de faire des constats sur la meilleure structure possible. Pourtant, la majorité des auteurs, lorsqu'ils traitent des extensions de modèle, étendent la formulation de base et rajoutent des contraintes et/ou des variables. Plus la formulation de base est performante, plus la variante le sera. On peut voir cela dans le cas où les contraintes de précédence sont généralisées ou lorsque la version stochastique est considérée.

Pour ce qui est du nivellement de ressources, ce concept a été traité dans un contexte tactique, mais les axes étudiés ne couvrent pas l'équilibrage de charges. Ils visent surtout à réduire les coûts externes. Au niveau de l'application de ce contexte (pas nécessairement dans le cas du RCCP), la plupart des études se limitent à mesurer les fluctuations de charge (variance, pics, écarts entre périodes, indicateurs comme RHH ou RID), sans imposer une *forme temporelle* du profil imposé. Un planning peut ainsi sembler régulier selon une métrique, tout en gardant plusieurs phases de montée ou de descente. De plus, la modélisation des *intensités d'exécution* repose généralement sur l'hypothèse qu'elles sont identiques pour tous les lots de travail. Bien que quelques références abordent le cas où l'intensité est totalement indépendante pour chaque ressource, l'impact réel de cette hypothèse n'a pas été étudié en profondeur, notamment dans un cadre de nivellement de ressources. Enfin, la prise en compte des différents métiers ou spécialités n'est que très rarement considérée dans ces travaux.

CHAPITRE 3 MÉTHODOLOGIE DE RECHERCHE

Ce chapitre présente la démarche méthodologique adoptée dans le cadre de cette thèse. L'objectif est de décrire les étapes ayant conduit au développement, à l'expérimentation et à la validation de nouvelles formulations pour le RCCP, en s'appuyant à la fois sur une réflexion théorique et sur une validation computationnelle.

3.1 Cadre général de la méthodologie

Cette recherche adopte une démarche méthodologique allant de la modélisation mathématique du problème de base à l'expérimentation numérique, puis à l'analyse des performances. Elle vise à développer, comparer et améliorer différentes formulations du RCCP afin d'en identifier les limites et les leviers d'amélioration, avant d'étendre le modèle à des cas pratiques et plus complexes.

Le travail a été conduit en plusieurs étapes complémentaires. Dans un premier temps, un modèle de référence a été sélectionné et analysé en profondeur afin de comprendre les facteurs influençant son comportement en résolution, notamment en lien avec les mécanismes internes du solveur. Cette analyse a permis d'identifier plusieurs pistes d'amélioration, menant à la proposition de nouvelles variantes de modèles visant à renforcer l'efficacité et la stabilité numérique des formulations. Enfin, le meilleur modèle issu de cette étude a été étendu pour intégrer une problématique plus réaliste : le *nivellement des ressources*, souvent négligé dans les approches tactiques classiques.

Bien que les solveurs d'optimisation aient connu d'importants progrès, leur performance ne dépend pas seulement de la qualité du modèle relaxé, mais aussi de la manière dont le problème est formulé. Deux modèles présentant la même qualité du relaxé peuvent conduire à des comportements de résolution très différents. Cette variabilité s'explique par la sensibilité du solveur à certains aspects structurels du modèle, ce qui nécessite une analyse approfondie afin d'identifier les leviers potentiels d'amélioration. Dès lors, comprendre ces mécanismes et identifier les anomalies de résolution devient essentiel pour orienter le solveur vers la solution optimale et exploiter pleinement ses capacités. L'objectif global est donc de proposer des formulations plus performantes, capables de réduire les temps de calcul tout en garantissant des solutions de qualité.

Les objectifs spécifiques de la thèse se déclinent comme suit :

1. Améliorer les performances du modèle de référence à travers l'analyse et la compréhension

du comportement du solveur ;

2. Proposer plusieurs types de modèles afin d'identifier la formulation la plus performante, en capitalisant sur les enseignements issus du premier objectif ; et
3. Étendre le meilleur modèle disponible dans la littérature pour intégrer le problème du nivellement des ressources et accélérer ses performances.

Ces objectifs reposent en grande partie sur l'hypothèse que les solveurs actuels, grâce à leurs algorithmes internes, ont le potentiel de traiter des problèmes de taille industrielle. Toutefois, il est nécessaire d'interpréter les sources de difficulté et de les corriger autant que possible, en les guidant par des ajustements appropriés au niveau du modèle.

La Figure 3.1 illustre de manière synthétique les principales étapes de cette démarche.

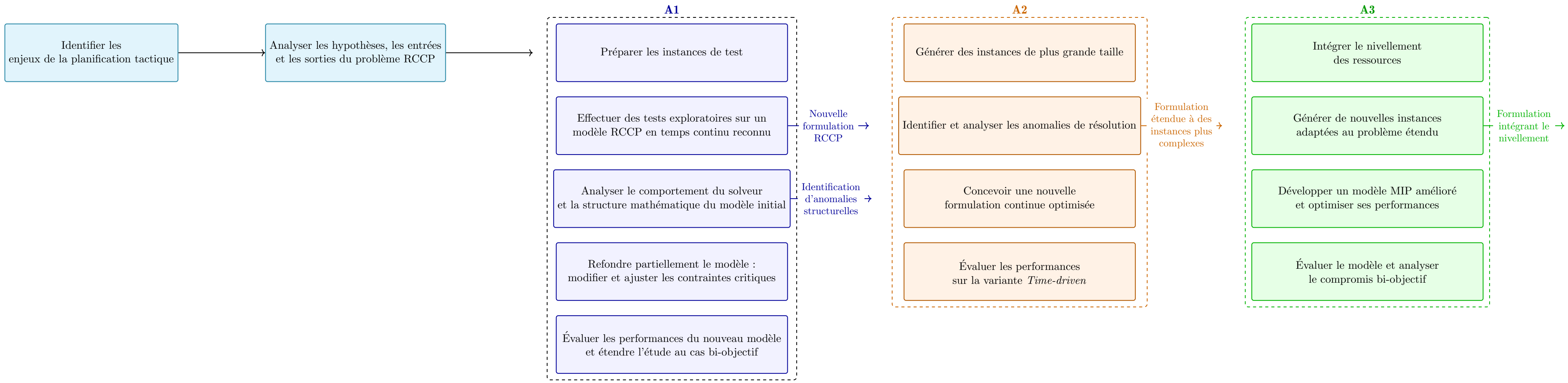

FIGURE 3.1 Schéma général de la méthodologie de recherche

3.2 Description détaillée des étapes de la recherche

Après avoir identifié les enjeux liés à la planification tactique, ainsi que le problème étudié (RCCP) adapté à ce niveau d'analyse, ses hypothèses et ses principales contraintes, nous avons préparé les instances à tester, reproduire ou générer selon les besoins de l'étude. Les étapes suivantes ont ensuite été réalisées :

3.2.1 Tests exploratoires

Cette étape consiste à sélectionner un modèle RCCP en temps continu déjà proposé dans la littérature [30] et à le tester sur des instances. Elle est divisée en trois étapes :

1. Vérifier la cohérence et la faisabilité des instances testées ;
2. Analyser les temps de calcul et la structure du modèle ; et
3. Identifier les premiers points faibles (instabilités, gaps élevés, lenteur de convergence, etc.).

3.2.2 Analyse du comportement du solveur

L'analyse du comportement du solveur CPLEX permet de mieux comprendre les difficultés rencontrées. Les indicateurs étudiés sont principalement :

1. Le nombre de nœuds explorés ;
2. Le nombre d'itérations du simplexe dual ;
3. Les gaps d'intégralité ; et
4. La qualité de la relaxation linéaire.

3.2.3 Refonte partielle du modèle

Cette étape a pour objectif de corriger les limitations identifiées en ajustant certaines contraintes du modèle, à partir de ce que nous appelons des anomalies de résolution. L'analyse de ces anomalies permet de formuler des hypothèses sur leurs causes principales, souvent liées à la structure ou à la formulation du modèle. Ces hypothèses orientent ensuite des modifications ciblées, visant à corriger ou à atténuer les effets observés.

Chaque ajustement est ensuite testé expérimentalement afin d'évaluer son impact sur la performance globale du solveur et de confirmer la pertinence de la correction apportée.

3.2.4 Développement des formulations mathématiques

Après la première phase d'amélioration du modèle et une fois les indicateurs identifiés permettant d'analyser les facteurs influençant la complexité d'une instance lors de la résolution, trois variantes ont été développées à partir des constats tirés de cette analyse : la variante *Resource-driven*, la variante *Time-driven* et une troisième, représentant un compromis entre les deux, obtenue par la construction du front de Pareto. Chaque variante a ensuite été comparée au modèle de référence afin d'en évaluer les apports.

3.2.5 Nouvelle formulation continue optimisée

À partir des constats précédents, de nouvelles formulations mathématiques, en temps discret et en temps continu, ont été développées et reformulées en combinant plusieurs familles de variables et de contraintes, afin de permettre le traitement d'instances de plus grande taille et d'aborder des variantes plus complexes. Une comparaison théorique des relaxations linéaires et des relations de dominance a également été menée.

3.2.6 Comparaison avec la littérature

Cette étape confronte les modèles proposés aux formulations déjà établies dans la littérature (notre 1er modèle [83], [60], [79]). La variante choisie dans ce cas est le *Time-driven*. Les critères de comparaison sont :

1. La qualité des solutions (coût du projet) ;
2. Le temps de résolution ;
3. La robustesse en termes de performances ; et
4. L'applicabilité à grande échelle (scalabilité).

3.2.7 Extension au nivellement des ressources

Enfin, la méthodologie est complétée par une extension visant le *nivellement des ressources*. L'objectif est de réduire les fluctuations d'utilisation tout en maîtrisant la durée du projet. Afin d'étudier le nivellement, nous avons procédé à :

1. L'introduction d'un profil de charge unimodal (croissant/décroissant) ;
2. L'étude de l'impact de la relaxation de la contrainte d'uniformité des intensités entre les ressources ; et
3. L'analyse du compromis entre le lissage de la charge et la durée totale du projet.

3.3 Méthodologie expérimentale

En parallèle du travail théorique, une méthodologie expérimentale est mise en place :

1. *Génération d'instances* : Sur le plan expérimental, l'étude évalue la *scalabilité* en générant de nouvelles instances ou en se fiant aux instances connues de la littérature selon la densité des relations de précédence, le nombre de ressources et le niveau d'agrégation temporelle, la complexité de l'instance est définie ;
2. *Evaluation de performances* : Les informations accessibles par le solveur permettent d'observer le *gap* d'intégralité initial, la croissance de l'arbre, le nombre de coupes et la répartition des temps de calcul (prétraitement, relaxation, exploration), limites de temps, paramètres du solveur ; et
3. *Analyse des résultats* : présentée sous forme de tableaux comparatifs mettant en évidence les anciennes et les nouvelles versions du modèle, ainsi que des indicateurs statistiques permettant d'identifier les facteurs influençant la complexité des instances. Des schémas synthétiques, tels que le tracé des fronts de Pareto, sont utilisées pour l'analyse des résultats expérimentaux.

3.4 Contributions de la thèse

Dans ce contexte, la thèse s'articule autour de trois contributions principales, chacune présentée sous la forme d'un article scientifique. La première contribution (Chapitre 4) de cette thèse consiste à développer un *nouveau modèle en nombres entiers pour le RCCP*, conçu pour traiter efficacement des instances de grande taille. La méthodologie repose sur une démarche progressive : (i) partir d'une formulation de base en temps continu ; (ii) analyser en profondeur ce modèle selon deux axes : d'une part, l'interprétation des informations fournies par le solveur afin de repérer les anomalies de résolution, et d'autre part, l'examen minutieux des contraintes de la formulation ; (iii) formuler des hypothèses et en tirer des observations permettant de mettre en lumière les limites du modèle initial ; (iv) proposer des améliorations correctives sur les contraintes et les variables, afin de renforcer la structure du modèle et d'accélérer l'exploration de l'espace de solutions. L'objectif est de résoudre des *instances de plus grande taille* que celles accessibles avec le modèle de base.

En augmentant encore l'ordre de grandeur des instances, de nouvelles anomalies seront détectées, notamment la qualité des bornes inférieures, ouvrant ainsi de nouvelles perspectives pour traiter des instances encore plus vastes (Chapitre 5). Des reformulation seront introduites, améliorant à la fois la qualité de la relaxation linéaire et la stabilité numérique. Conscients du rôle déterminant de la structure du modèle sur les performances du solveur, nous allons mené une étude comparative entre plusieurs formulations continues, puis identifié et affiné la variante la plus performante, de type *Time-driven*. À travers cette contribution, nous allons aussi étudier la différence entre les modèles en temps continu et ceux en temps discret. Le principal critère d'analyse porte sur l'impact de la *flexibilité*, considérée comme un avantage des modèles continus par rapport aux modèles discrets, ainsi que sur la différence observée en termes de temps d'exécution.

Une fois le modèle consolidé, nous intégrerons un aspect essentiel en pratique : le *nivellement de ressources*, abordé selon deux volets complémentaires. Le premier vise à limiter les fluctuations de la main-d'œuvre afin d'éviter les cycles coûteux de recrutement et de licenciement, tout en réduisant les variations de l'utilisation des ressources et en tenant compte de la durée globale du projet. Le second volet vise à améliorer l'efficacité de la résolution, d'une part en proposant une heuristique de construction permettant d'affiner l'horizon de planification, et d'autre part grâce à une stratégie incrémentale qui améliore progressivement la borne inférieure. Enfin, le front de Pareto est tracé afin de mener une analyse bi-objectif mettant en évidence le compromis entre le nivellement des ressources et la durée du projet.

CHAPITRE 4 ARTICLE 1 : A NEW MIP RCCP MODEL FOR TACKLING TACTICAL PROJECT PLANNING

Anis Noureddine[1,2,3], François Soumis[1,2], Robert Pellerin[1,3]

[1]Département de mathématiques et de génie industriel, Polytechnique Montréal
[2]GERAD, Montréal, Canada
[3]CIRRELT, Montréal, Canada

Submitted : 23 April 2024 ; Accepted : 28 November 2024 ; Published online : 19 December 2024.
International Journal of Production Research (IJPR), 2024 (online) / 2025 (print).
DOI : 10.1080/00207543.2024.2442087

Abstract

This paper proposes a new continuous-time linear mixed model for rough cut capacity planning (RCCP) adapted to tackle various tactical project planning scenarios. RCCP models are typically designed for the early phases of projects to decide the work package's execution intensities for each project planning period. The goal is to solve large-scale, complex instances while ensuring both optimality and efficient resolution times. We propose modifications to the constraints of one of the best models in this field to improve its resolution. Recognizing that overlap between two consecutive periods adversely affects the performance of the MIP solver, we introduce minimal, fictitious boundaries between these periods. Additional strategies, derived from analyzing initial constraints and solver behavior, ensure solution optimality. The objectives considered are : minimizing the costs of using external resources by setting the project's end date (Time-driven) and minimizing the makespan (Resource-driven). The bi-objective case is also addressed. The improved model reduces the average number of dual simplex iterations of CPLEX by 72%. Moreover, only 0.7% of large instances remained unresolved with the new model compared to nearly 30% of instances with the previous best model. The fast resolution of a single-objective problem opens up opportunities for multi-criteria approaches, making the model adaptable to more complex needs in practice.

keywords RCCP ; project planning ; linear mixed program.

4.1 Introduction

The process of planning complex projects is by nature iterative, beginning with a broad definition of the project content. Engineering and supply activities are often conducted iteratively in the early phases of the project to better define its content [26, 31, 39]. These iterations lead to different project plans, often described as planning levels. Among proposed planning methodologies, [39] recognizes three main levels. The first level (strategic level) aims at predicting the long-term workload and the main milestone dates. The second level (tactical planning) consists of developing an aggregate schedule for the project work packages without precise knowledge of the required activities [52]. At this level, allocating resources to work packages is done by time period, usually week or month [32]. The third level consists of detailed scheduling of the project activities. Specific resource requirements by activity and precedence constraints are then known, and the goal is to establish a start and end date for each activity.

The academic community in project planning has primarily focused on the Resource-Constrained Project Scheduling Problem (RCPSP) [88], the third planning level. However, RCPSP may not be ideal for tactical project planning due to differences in the nature of decisions, objectives, resources, constraints, time horizon, and representation [52, 81, 82, 100]. At the operational level, RCPSP is used to schedule activities to minimize project duration. Unlike tactical planning, the time horizon is shorter and divided into discrete periods, assuming constant resource capacities. It requires prior knowledge of activity durations, resource consumption, and precedence relationships, which is feasible due to higher data accuracy.

In contrast, tactical project planning covers a broader horizon, where activities are not yet fully defined [6, 12, 13]. The RCCP approach divides planning into periods (usually weeks or months), with known general resource capacities. Work packages are initially described in terms of workload (e.g., man-hours) and later broken down into detailed activities. Unlike scheduling problems, the periods of execution and the intensity of each work package are decision variables. Note that some authors consider certain variants of Resource-Constrained Problems as adaptable to tactical planning, such as RCPSVP (Resource-Constrained Project Scheduling Problem with Variable Intensity Activities) or FRCPSP (RCPSP with Flexible Resource Profiles) [12, 32, 60, 79].

The RCCP establishes detailed resource availability profiles for the operational level. Early decisions made in RCCP, such as total makespan and milestone dates, serve as inputs or constraints for later planning cycles. RCCP helps identify capacity issues and potential delays early, enabling efficient resource allocation and adjustments when necessary, making RCPSP easier to manage [32, 47]. Two variants of RCCP can be distinguished : Resource-driven and Time-driven [32]. The Resource-driven consists of minimizing the project's total duration under a constraint of not exceeding the resource capacity at each period. Alternatively, one can also seek to minimize the execution costs under constraints on the project's end date. This variant is referred to as Time-driven.

The majority of mathematical models addressing these problems operate in a discrete-time framework. However, this poses challenges, particularly with precedence constraints in the tactical planning context. To overcome these limitations, [32, 51, 78] proposed continuous-time models adapted to tactical planning. On the other hand, [79] demonstrated that the choice of variables and constraints significantly influences the efficiency of the resolution process. [23], when addressing the multi-project FRCPSP, demonstrated that modern solvers are capable of solving many real-world instance sizes within reasonable computation times. However, not all instances are solved optimally. In their testing of the FRCPSP model proposed by [79], [105] did not find feasible solutions for all large instances. [105] also demonstrated that, even for small instances, optimality could not be guaranteed when using metaheuristics. Additionally, [17], when addressing the RCPSVP problem with generalized precedence relations, they noted that achieving the optimal solution is not straightforward, even when resource constraints are neglected.

Motivated by these insights, the limited number of studies of RCCP models, and recent technological advancements that enable solvers like CPLEX to solve large-scale instances, this paper introduces a new RCCP model adapted to tactical project planning, which constitutes one of the main contributions of this work. We improve the performance of the continuous/discrete model proposed by [32], which was adapted from the RCPSVP model developed by [51]. This model is recognized as one of the most efficient in the literature [32]. The improvement is achieved by modifying its constraints to enhance resolution speed. Our enhancements were developed using the following approach. We first conducted a thorough analysis of the problem. By exploiting information provided by a MIP solver, we analyzed the structure of the feasible solutions and studied the model constraints. This allowed us to identify certain inefficiencies, referred to as anomalies, where the model experienced slow progression towards the solution. These anomalies explain why the model was not performing optimally or where improvements could be made. For each problem identified in the diagnostic phase, we developed corrective measures and proposed strategies, which represent another contribution of this paper. To ensure that the improved model performs significantly better across various scenarios, we conducted tests using diverse objective functions and a large number of instances with varying characteristics. We first consider the Time-driven variant over Cherkaoui's instances, which are distinguished by their complexity [32]. Then, we tackle instances of a distinct nature, primarily defined by a limited number of precedence relations. We evaluate the two fundamental variants : Time-driven and Resource-driven. Upon achieving very promising results, we address a bi-objective mathematical model and suggest an ϵ-constraint method for obtaining a Pareto frontier. Building on these promising results, one of the important contributions of this paper is the extension of our model to a bi-objective mathematical model. To address this, we introduced an ϵ-constraint method that generates a Pareto frontier, allowing decision-makers to explore trade-offs between the two objectives.

The remainder of the paper is organized as follows. First, Section 4.2 aims to establish a brief literature review on RCCP models. In Section 4.3, we describe our problem and the model used. In Section 4.4, we propose modifications to the formulation of the initial model, identifying some theoretical properties of these changes. In Section 4.5, we present our methodology to implement, validate, and test our model. In Section 4.6, we analyze the initial model's performance and undertake a comparison with the initial RCCP formulation. This comparison aims to highlight the ways in which our approach enhances the efficiency of the MIP solver. The paper concludes with research perspectives.

4.2 Theoretical Background

RCCP models are adequate for planning projects at the tactical level [52, 87]. In this case, the requested schedule is intended to be aggregated and includes only *work packages*. These summarize the type of work to be done and the resource requirements, described as workload (i.e., men-hours or men-days) [39].

4.2.1 Mathematical models for the classic RCCP problem

This problem was proved NP-hard by [60] (study of an equivalent problem to RCCP, but considered a variant of Resource Constrained Project Scheduling Problem with variable intensity activities RCPSVP). De Boer proposed the basic model in the literature [39], where he proposes a model in a multi-project context in which the precedence constraints are not expressed linearly. Its resolution approach was initially proposed for the Time-driven variant. It was then adapted to the Resource-driven variant. [47] proposes a discrete-time mixed linear model aimed at minimizing the cost of external capacities by penalizing the delay that the limitation of internal and external resources could cause. The Time-driven and Resource-driven variants are addressed in the same model. Precedence constraints are considered by prohibiting a successor from executing at the same period as one of its predecessors. However, only one exception is allowed : if a work package i ends in the period p and a work package j is successive to i, then j can be executed in p, but by limiting the number of work packages executed in the period p. This reduces the risk of unfeasibility because when it is tolerated that two successive work packages be performed in the same period, there is no constraint requiring that they respect this succession in the same period.

[60] proposes a discrete-time linear mixed model RCPSVP for the Time-driven variant. He strengthens the LP formulation of the precedence constraints and then adapts his model for the Resource-driven variant. [42] and [111] use the same model but propose different methods for solving integer linear programs. They decompose the original problem into simpler sub-problems to reduce the number of variables produced by precedence relations and to avoid insufficient memory when solving the relaxed model.

For these reasons, [51] propose to express the start and end dates using continuous variables to reduce the difficulties found in previous models at the level of precedence constraints, adding two groups of binary variables indexed by the set of work packages and aggregate periods. These variables are necessary for the periodic evaluation of workloads. The main advantage of this model lies in the fact that the precedence relations are explicitly presented. No precedence constraint violation is possible during the same period. Moreover, we do not need to over-constrain that two successive work packages cannot be in the same period since we can express these constraints with continuous variables. [32] generalize this model in the context of large construction projects by varying the durations of the periods. In this case, the lengths of the periods are not consistently equal. The authors have shown that aggregating these periods greatly reduces the planning effort, but this method may not realistically account for costs.

4.2.2 RCCP variants applied to different project types in tactical planning

Intended to solve some variants in the context of tactical planning, [59] introduces a variant of the RCPSVP by extending generalized precedence relations, known as feeding precedence relations. [6] model concurrent engineering and production, commonly used by ETO companies, by further generalizing the extension of [59]. [12] propose an RCCP model, adapted from [51], which allows overlapping between work packages using different possible modes. [17] tackle the RCPSVP model with feeding precedence relations. Considering a project network without explicitly accounting for resource constraints, they propose a forward recursion algorithm to determine the earliest start and finish times for each activity, with the objective of minimizing the project duration. Several optimization models have been proposed to manage the lack of information in the ETO context [81]. [26] presents a discrete-time deterministic RCCP model based on a real-world ETO industrial setting. Building on this, [27] extends the model of [6] by introducing a robust RCCP to tackle uncertainty. We can also cite [46] that propose four discrete-time models adapted for tactical-level production planning of prefabricated engineered wood products in an Engineer-to-Order (ETO) context. The models focus on minimizing costs, project makespan, and set-up times. They were applied to schedule production and assess the impact of introducing one or more projects within the same time frame. Several authors consider approaches that support various levels of planning, including tactical planning [81,82].

4.2.3 Flexible Resource-Constrained Project Scheduling (FRCPSP)

Aiming to solve the Flexible-RCPSP, a variant of the RCPSP where resource capacity can vary over time and is suitable for the RCCP problem [12], [79] compare four discrete-time linear mixed models. In their formulations, they are based on different combinations of decision variables and constraints already proposed for the RCPSP [14, 61], the RCPSVP [60], the multi-mode RCPSP with generalized precedence constraints [92] and the Flexible-RCPSP with discrete resources [91]. From these formulations, the authors conclude that the model based on the work of [60] and [14] proves to be of better quality in terms of execution time and the quality of the solutions found by this model. [24] integrate the formulations of [61] and [91] to tackle a multi-criteria, real-world FRCPSP problem. To address a particular problem in the disaster response phase, [11] propose a discrete-time FRCPSP model tailored to this context, incorporating additional features such as potential interruptions and varying skill levels. [105] proposed a hybrid metaheuristic and compared its performance with the DT-3 model developed by [79]. They demonstrated that for large instances, the MIP model fails to find feasible solutions. However, for small instances, the metaheuristic does not achieve the optimal solutions found by the MIP model. [23] introduce a multi-project FRCPSP model for tactical planning in the ETO (Engineer-To-Order) environment, adapting the DT-3 model developed by [79]. Although the results show that modern solvers can solve real-world instance sizes within reasonable computation times, not all instances are solved optimally.

However, all these discrete-time models are characterized by a vast number of variables and constraints, which could lead to the consumption of all the memory for large instances. This problem led [78] to develop the Flexible-RCPSP model, based on the discrete-time model of [79] and the RCPSP model of [63] to propose a continuous event-based model. An event could be : the start, the end, or any variation in resource consumption. He concludes that his model provides a better solution than the discrete-time model, which is not the case regarding the running time.

4.2.4 Summary of works in tactical planning

Table 4.1 summarizes the most important works in tactical planning. In the table, RI refers to real project instances, FI to fictitious instances, and GPR to generalized precedence relations.

Table 4.1 The most important works in tactical planning

Authors	Context	Objective Function	Capacity	Precedence Relations	Start/End Dates	Solution Method	Test Instances
[39]	Multi-projects	Time-driven and Resource-driven	Flexible	Over-demand or risk of violating precedence relationships	Discrete	Heuristic	FI
[47]	Multi-projects	Time-driven and Resource-driven	Flexible	Over-demand or risk of violating precedence relationships	Discrete	Exact + Heuristic	FI
[111]	ETO	Min. expected cost of external resources	Flexible	Over-demand or risk of violating precedence relationships	Discrete	Exact + Heuristic	FI
[42]	Multi-projects	Time-driven	Flexible	Time windows	Discrete	Heuristic	FI
[60]	Projects	Time-driven	Flexible	Over-demand	Discrete	Exact	FI
[59]	Projects	Time-driven	Flexible	Over-demand (GPR)	Discrete	Exact	FI
[6]	ETO	Resource-driven	Flexible	Over-demand (GPR)	Discrete	Exact	RI, FI
[51]	Projects	Time-driven	Flexible	Easily expressed at the cost of adding continuous start and end dates	Continuous	Exact	FI
[79]	Projects	Min. makespan	Rigid	Over-demand	Discrete	Exact	FI
[32]	Projects	Time-driven and Resource-driven	Flexible	Easily expressed at the cost of adding continuous start and end dates	Continuous	Exact	FI
[12]	Multi-projects	Min. regular resource cost and project duration	Flexible	Easily expressed at the cost of adding continuous start and end dates (GPR)	Continuous	Exact	FI
[26], [27]	ETO	Min. cadence costs	Flexible	Over-demand	Discrete	Exact	RI
[78]	Projects	Min. Makespan	Rigid	Easily expressed at the cost of adding continuous start and end dates	Continuous	Exact + Heuristic	FI
[24]	Projects	Multi-objective	Rigid	Over-demand	Discrete	Exact + Heuristic	RI
[105]	Projects	Resource-driven	Rigid	Over-demand	Discrete	Metaheuristic	FI
[11]	Projects	Min. makespan	Rigid	Over-demand	Discrete	Exact + Heuristic	RI
[23]	ETO	Min. total weighted lateness	Flexible	Over-demand	Discrete	Exact	RI
[46]	ETO	Multi-objective	Flexible	Over-demand	Discrete	Exact	RI
[17]	Projects	Min. Makespan	Rigid	Over-demand (GPR)	Discrete	Forward recursion	FI

Given that solvers encounter difficulties in solving the cited models for large instances (starting from 50 work packages), particularly in articles using exact methods, and that heuristics and metaheuristics do not guarantee optimal solutions, we propose a new model by modifying the formulation of [32], which closely resembles the model of [51] when period lengths are the same. The latter has shown competitive results compared to [60]. We opt for a continuous event-based model, as such models generally yield better solutions than their discrete-time counterparts [78].

4.3 Formulation of the mathematical model

Recognizing the need to solve the tactical planning problem, we present the RCCP model introduced by [32]. This model is inspired from the RCPSVP model from [51] but allow for varying period durations rather than uniform lengths. It is a mixed linear model and is mainly characterized by two different representations of time. A continuous representation of start and end dates to express precedence relations and a discrete representation based on aggregate periods to express capacity constraints. So, in this section, we formally describe the initial RCCP model used.

4.3.1 Input and notations

At this planning phase, a horizon, denoted H, is discretized into several time periods. Let us denote by P the set of these periods. Thus, each period p is characterized by a duration noted D_p, generally presented according to the number of weeks. Let I denote the set of work packages. A set of resources, denoted R, is necessary to execute the work. Each work package i requires a certain workload Q_{ri} (men-hours) of resource r. The total required workload of a work package is therefore deduced by summing all the workloads by resources used for the considered work package, denoted $Q_i = \sum_{r \in R} Q_{ri}$. Each work package is characterized by a maximum workload Q_i^{max} (expressed in men-hours) that can be executed for work package i for one unit of time (week in our case), a minimum workload Q_i^{min} (men-hours), a minimum start date in which it can begin, denoted S_i and a delivery time for the equipment pieces associated with the work package, denoted Del_i. A capacity of internal resources L_{rp} is available at period p per resource r. In addition, a capacity of external resources M_{rp} per resource r at period p is available. The use of this type of resource is penalized by an additional cost, denoted c_{rp}. Finally, the set E of network arcs representing end-start precedence relations between work packages is defined as follows : if a work package i precedes a work package j, then : $(i, j) \in E$.

4.3.2 Initial formulation of the model : *Strategy 0*

We introduce the initial model as a set of variables, multiple sets of constraints, and the objective functions considered in this paper. The *Strategy 0* is resolving the initial model using a MIP solver.

Decision Variables

x_{ip} = intensity of work package i at period p ($0 \leq x_{ip} \leq 1$) as a fraction of Q_i.
ts_i = work package start date i.
tf_i = work package end date i.
d_{ip} = duration of work package i at period p.
zs_{ip} = 1 if work package i starts at period p or earlier.
0 otherwise.
zf_{ip} = 1 if work package i ends at period p or earlier.
0 otherwise.
z_{rp} = quantity used of internal resource r over the period p.
w_{rp} = quantity used of external resource r over the period p.
$C^{\mathbb{R}}_{max}$ = project end date.
c = cost of using external resources.

Constraints determining the start and end dates of work packages in periods : CT1

$$ts_i \geq \sum_{k=1}^{p} D_k \times (1 - zs_{ip}), \qquad \forall i \in I, p \in P. \quad (1)$$

$$ts_i \leq \sum_{k=1}^{p} D_k + (H - \sum_{k=1}^{p} D_k) \times (1 - zs_{ip}), \qquad \forall i \in I, p \in P. \quad (2)$$

$$zs_{ip} \geq zs_{ip-1}, \qquad \forall i \in I, p \in P. \quad (3)$$

$$tf_i \geq \sum_{k=1}^{p} D_k \times (1 - zf_{ip}), \qquad \forall i \in I, p \in P. \quad (4)$$

$$tf_i \leq \sum_{k=1}^{p} D_k + (H - \sum_{k=1}^{p} D_k) \times (1 - zf_{ip}), \qquad \forall i \in I, p \in P. \quad (5)$$

$$zf_{ip} \geq zf_{ip-1}, \qquad \forall i \in I, p \in P. \quad (6)$$

Constraints (1) and (2) indicate the lower and upper bounds of the work package start dates relative to their positioning in the periods. The same principle is applied in constraints (4) and (5) regarding work package end dates. Constraints (3), respectively (6) indicate that if a binary variable zs_{ip}, respectively zf_{ip} is equal to 1, then all the periods that follow p are equal to 1.

Constraints determining work package durations in periods : CT2

$$d_{ip} \leq D_p \times (zs_{ip} - zf_{ip-1}), \qquad \forall i \in I, p \in P. \quad (7)$$

$$d_{ip} \geq D_p \times (zs_{ip-1} - zf_{ip}), \qquad \forall i \in I, p \in P. \quad (8)$$

$$d_{ip} \geq tf_i - \sum_{k=1}^{p} D_k + D_p \times zs_{ip-1} - H \times (1 - zf_{ip}), \qquad \forall i \in I, p \in P. \quad (9)$$

$$d_{ip} \geq \sum_{k=1}^{p} D_k \times (1 - zs_{ip-1}) - ts_i - D_p \times zf_{ip}, \qquad \forall i \in I, p \in P. \quad (10)$$

$$\sum_{p \in P} d_{ip} = tf_i - ts_i, \qquad \forall i \in I. \quad (11)$$

Constraints (7) indicate that d_{ip} is equal to 0 when work package i is not executed at period p. It will be upper bounded by D_p otherwise (D_p being the duration of the period p). Constraints (8) state that if a work package starts before period p and ends after period p, then the duration is D_p. Constraints (9) and (10) take into consideration the case where the start date and the end date, respectively, are strictly included in the period p (strictly greater than the end date of the period $p-1$ and strictly less than the end date of the period p) to bound the durations in the period p appropriately. Constraints (11) indicate that the total duration equals the difference between the start and end date.

Time Constraints and Precedence Relations : CT3

$$ts_i \geq S_i + Del_i, \qquad \forall i \in I. \quad (12)$$

$$ts_j \geq tf_i, \qquad \forall (i,j) \in E. \quad (13)$$

$$H \geq tf_i, \qquad \forall i \in I. \quad (14)$$

Constraints (12) indicate that work package i must start after the delivery of equipment pieces of work package i. Constraints (13) require the respect of precedence relations between any pair of work packages $(i,j) \in E$. Constraints (14) prohibit a work package end date from exceeding the allocated time horizon.

Capacity Constraints : CT4

$$x_{ip} \times \sum_{r \in R} Q_{ri} \leq Q_i^{max} \times d_{ip}, \qquad \forall i \in I, p \in P. \quad (15)$$

$$x_{ip} \times \sum_{r \in R} Q_{ri} \geq Q_i^{min} \times d_{ip}, \qquad \forall i \in I, p \in P. \quad (16)$$

$$\sum_{p \in P} x_{ip} = 1, \qquad \forall i \in I. \quad (17)$$

$$z_{rp} + w_{rp} = \sum_{i \in I} x_{ip} \times Q_{ri}, \qquad \forall r \in R, p \in P. \quad (18)$$

$$z_{rp} \leq L_{rp}, \qquad \forall r \in R, p \in P. \quad (19)$$

$$w_{rp} \leq M_{rp}, \qquad \forall r \in R, p \in P. \quad (20)$$

$$c = \sum_{r \in R} \sum_{p \in P} c_{rp} \times w_{rp}. \quad (21)$$

Constraints (15) and (16) require that the maximum workload and the minimum workload of a work package, respectively, must be respected. Constraints (17) indicate that the required workload of a work package must be fully satisfied. Constraints (18) indicate that the workload must be satisfied using internal and external resources. Also, constraints (19) and (20) indicate that the quantity used of internal and external resources must not exceed their capacity limits. Finally, constraint (21) calculates the cost of using external resources.

Objective functions

We deal with the following objective functions :

1. *Resource-driven variant* : Minimize $C^{\mathbb{R}}_{max}$ under the constraints : $tf_i \leq C^{\mathbb{R}}_{max}$, $\forall\ i \in I$ and prohibiting the use of external resources.
2. *Time-driven variant* : Minimize c under the constraints : $tf_i \leq F$, $\forall i \in I$. (22) F being a fixed integer value.

We refer to the initial formulation dealing with the Resource-driven variant as (MR1) and to the Time-driven variant as (MT1). Then, the two functions are considered simultaneously, and an appropriate method is proposed to identify the Pareto frontier. We refer to the resulting formulation as (MB1).

4.4 Improvements in the resolution of the initial model

We propose a redefinition of constraints (4) to improve the model's efficiency. Additionally, as previously discussed, we recommend a resolution strategy for each objective function to empirically validate the optimality of the solutions generated by our model. This will allow us to compare the optimal solutions from the initial model with those from the revised model. The modifications made to the model may stem from direct observations after analyzing the initial constraints or from an evaluation of the results obtained during testing of the initial model, which we deemed necessary to address the identified anomalies and improve the overall performance of (MT1) and (MR1), also identifying some theoretical properties of these changes. Finally, we propose an adapted ϵ-constraint method incorporating these improvements to address (MB1).

4.4.1 Improving the resolution of the formulation (MT1) : Strategy 1

Suppose we have a solution x_1 with a cost C, scheduled over a time horizon of $F = L + 1$. L coincides with the end of one of the periods. Consider the following decision problem, denoted by (L_s) : Is there a solution x_2 that incurs a cost C', planned within a time horizon of $F = L$?

Two possible distinct values of zf_{ip} for the same planning : Let illustrate an example of the solutions x_1 and x_2 in Figure 4.1. The blue and green solutions represent x_1 and x_2 with $L = 20$. P_j represents the jth aggregated period.

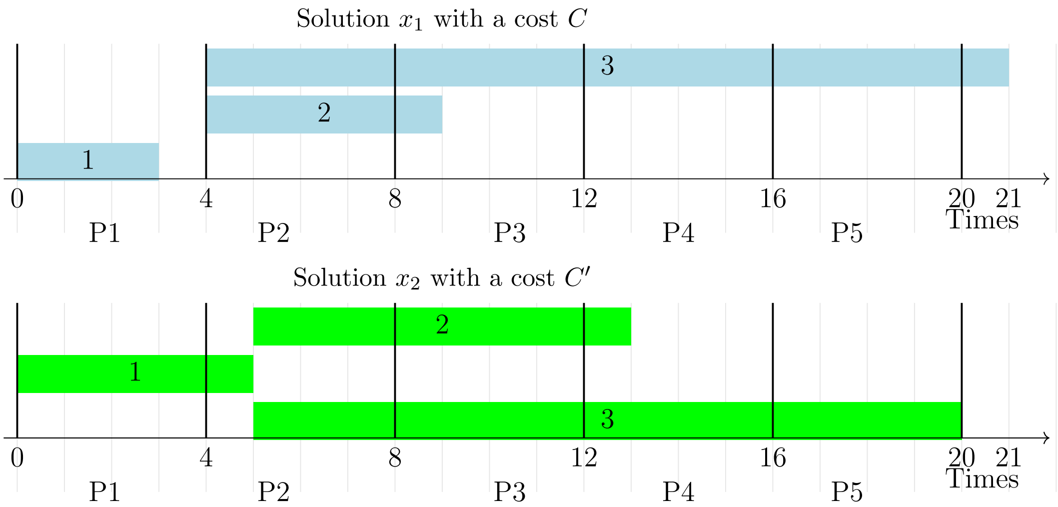

FIGURE 4.1 Illustration of solution derived from solving L_s

— Taking the green planning, two possible solutions are equivalent for the same planning :

— $zf_{35} = 1$ (work package 3 ends at period 5 with $d_{35} = D_5$) ; or

— $zf_{35} = 0$ and $zf_{36} = 1$ (work package 3 ends at period 6 with $d_{36} = 0$).

— If we have a solution like the one in blue, we want to know if there is a solution of value 20. By adding the additional constraints : $tf_i \leq 20.zf_{i5} \ \forall \ i \in I$, this will exclude the solution with $zf_{35} = 0$ and $zf_{36} = 1$.

Let's denote by p the period such that $L = \sum_{k=1}^{p} D_k$, and L coinciding with the end of p. We deduce that we can use the constraints : $tf_i \leq L.zf_{ip} \ \forall i \in I \qquad (22')$

This will exclude the possibility that $zf_{ip} = 0$. Indeed, when the upper bound coincides with the end of the period, these constraints indicate that if there is a solution with a value of L, all the decision variables zf_{ip} must be activated at 1. Indeed, in this case, we are supposed to end the execution of all work packages at period p or before. Otherwise, it means that there is no solution with a value of L. This will avoid unnecessarily searching for equivalent solutions.

Replacing the constraints (22) with the constraints (22′) in the initial model (MT1) results in a new model, denoted as (MT2).

Proposition 1. *Replacing constraints (22) with constraints (22′) in a decision problem (L_s) allows us to conclude on the existence of a solution and reduces the search space.*

Proof. Let us show that if there exists a feasible scheduling x_2 for (L_s), which also satisfies (MT1), such that $F = L$, then : we can always obtain an equivalent feasible scheduling for (MT1) and (MT2) by modifying some values of the binary variables and fixing the continuous variables.

The values of the binary variables that potentially need to be modified are those of zf_{jq} such that : $tf_j = L$ and q is the period such that L is at the end of q. Let us enumerate all the possibilities :

1. $\exists \ q \in P, \exists \ j \in I, \sum_{k=1}^{q-1} D_k < tf_j < \sum_{k=1}^{q} D_k \implies zf_{jq} = 1$. Upon setting $p = q$ and $i = j$, constraints (22′) and (22) yield identical upper bounds.
2. $\exists \ q \in P, \exists \ j \in I, tf_j = \sum_{k=1}^{q} D_k$ and $zf_{jq} = 1$. Constraints (22′) and (22) are equivalent upon setting $p = q$ and $i = j$.
3. $\exists \ q \in P, \exists \ j \in I, tf_j = \sum_{k=1}^{q} D_k$ and $zf_{jq} = 0$. By setting $zf_{jq} = 1$, we handle all possible cases according to the positioning of ts_j (illustrated in Figure 4.2). The constraints at risk of being violated are constraints CT1 and CT2 when one or more of these constraints depends on the binary variable zf_{jq}. We recalculate the bounds of a continuous variable whenever it is dependent on zf_{jq} :

I. $\sum_{k=1}^{q-1} D_k < ts_j < \sum_{k=1}^{q} D_k \implies zs_{jq} = 1$. By setting $zf_{jq} = 1$, we replace in the constraints CT1. We obtain : $\sum_{k=1}^{q-1} D_k \leq tf_j \leq \sum_{k=1}^{q} D_k$. By replacing in the constraints CT2 : $d_{jq} = tf_j - ts_j$ (obtained by the constraint of group (11)) and $d_{jq+1} = 0$ (upper bound obtained by the constraint of group (7)). These bounds are valid for (MT1) and (MT2).

II. $ts_j \leq \sum_{k=1}^{q-1} D_k$ and $zs_{jq-1} = 1$. By setting $zf_{jq} = 1$, we replace in the constraints CT1. We obtain : $\sum_{k=1}^{q-1} D_k \leq tf_j \leq \sum_{k=1}^{q} D_k$. By replacing in the constraints CT2 : $d_{jq} = D_q$ and $d_{jq+1} = 0$ (upper bound obtained by the constraint of group (7)). These bounds are valid for (MT1) and (MT2).

III. $ts_j = \sum_{k=1}^{q-1} D_k$ and $zs_{jq-1} = 0$. By setting $zf_{jq} = 1$, we replace in the constraints CT1. We obtain : $\sum_{k=1}^{q-1} D_k \leq tf_j \leq \sum_{k=1}^{q} D_k$. By replacing in the constraints CT2 : $d_{jq} = tf_j - ts_j$. These bounds are valid for (MT1) and (MT2).

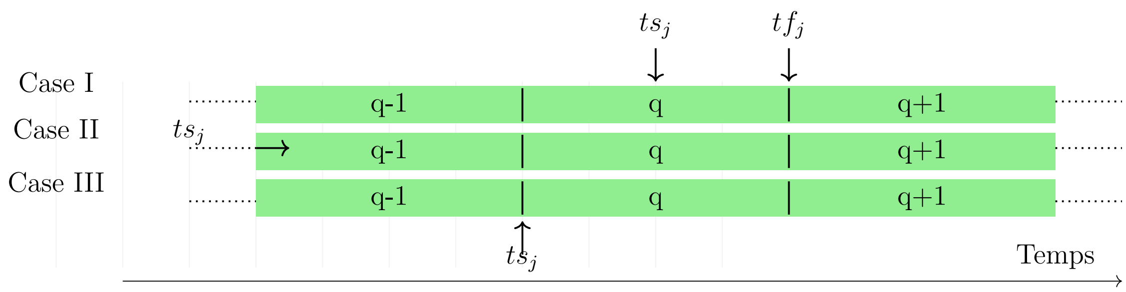

FIGURE 4.2 Illustration of the possible cases requiring an adjustment in the value of zf_{jq}

□

We conclude that (MT2) is a correct formulation for the Time-driven variant.

4.4.2 Improving the formulation (MR1) : Strategy 2

This Section proposes an alternative strategy to solve the Resource-driven variant. This strategy guarantees the optimal solution when the project duration is an integer value.

4.4.2.1 Integer project duration to avoid the long tailing effect

In practice, the project's total execution time is an integer. A fractional optimal value will be approximated to an integer (ceiling, round up always) to determine the number of weeks required to complete the project. We exploit this aspect to avoid long-tailing effects. This is achieved by substituting the variable $C^{\mathbb{R}}_{\text{max}}$, initially defined in real numbers, with $C^{\mathbb{Z}}_{\text{max}}$, defined in integers.

Lemma 1. *Assuming integer values for the total execution time, the optimal value remains unaffected when replacing the variable $C^{\mathbb{R}}_{max}$ with $C^{\mathbb{Z}}_{max}$. Furthermore, if the behavior of the solver remains consistent*[1]*, this substitution facilitates a quicker conclusion of the resolution process.*

Proof. Assuming that the total execution time of the project is discretized, and $\lceil Z^* \rceil$ the integer optimal value. It is easy to observe that changing the nature of C_{max} from $\mathbb{R}$ to $\mathbb{Z}$ results in the same optimal value $\lceil Z^* \rceil$.

Moreover, let b^n_l, $\underline{b}^n$, and $\bar{b}^n$ be the fractional lower bound of the current node l, the best fractional lower and upper bound at the current iteration n, respectively. Under the assumption that the solver's behavior is consistent, we aim to show that branching can be terminated prematurely using $C^{\mathbb{Z}}_{max}$. If $\underline{b}^n < \bar{b}^n < \underline{b}^n + 1$ and $\lceil \bar{b}^n \rceil = \lceil \underline{b}^n \rceil$, the algorithm stops when using $C^{\mathbb{Z}}_{max}$. This is because $\underline{b}^n$ and $\bar{b}^n$ will be approximated to $\lceil \underline{b}^n \rceil$ and $\lceil \bar{b}^n \rceil$ respectively, both in $\mathbb{Z}$, which is not the case with $C^{\mathbb{R}}_{max}$. If $\bar{b}^n \geq \underline{b}^n + 1$, $b^n_l < \bar{b}^n < b^n_l + 1$ and $\lceil b^n_l \rceil = \lceil \bar{b}^n \rceil$, the node l will automatically be pruned when using $C^{\mathbb{Z}}_{max}$, which is not the case with $C^{\mathbb{R}}_{max}$. Otherwise, the node l will remain active in both cases, undertaking identical steps.

$\square$

4.4.2.2 Two steps : integer and continuous project durations

We are considering a two-step approach. The first step involves solving the model up to an easily attainable gap. In the second phase, we will use key information gained during the first phase, such as the best lower bound found, to formulate a new optimization problem. This approach simplifies the two optimization steps, making the overall process easier than solving the original problem.

Re-optimization procedure :

1. This hypothesis implies that the solver applies the same Branch and Cut algorithm in both scenarios and makes identical node selections for branching

We consider the following notations : let us denote by (P) the problem, which consists in finding the continuous optimal value Z^* for the considered objective : minimization of the variable $C^{\mathbb{R}}_{max}$ of continuous nature. $\lceil Z^* \rceil$ represents the upper integer of the optimal value of the problem (P). $C^{\mathbb{Z}}_{max}$ is the integer variable to be minimized using the first step. The best upper bound of $C^{\mathbb{Z}}_{max}$ using the first step is denoted $\bar{b}$. $\underline{b}$ is the best integer lower bound using the first step.

Consider the same problem with an absolute gap ***abs-gap=1*** (the absolute tolerance between the upper and lower bound is at most one unit of time).

We express the *re-optimization procedure* as follows :

1. *Step 1 (ST1)* : Solve the problem with $C^{\mathbb{Z}}_{max}$ (integer) until $\bar{b} - \underline{b} \leq 1$, $\lceil Z^* \rceil \leftarrow \bar{b}$.
2. *Step 2 (ST2)* : If $\bar{b} - \underline{b} = 0$, stop. Otherwise, re-optimize considering $C^{\mathbb{R}}_{max}$ (continuous) and add a constraint : $C^{\mathbb{R}}_{max} \leq \underline{b}$.   (23)
   If only one solution is found, stop ; $\lceil Z^* \rceil \leftarrow \bar{b} - 1$.

Solving the original problem (P) with a solver, one will obtain : $\bar{b}^*$ the best upper bound found and $\underline{b^*}$ the best lower bound found. Without imposing a tolerance on the gap, we ultimately obtain : $Z^* = \bar{b}^* = \underline{b^*}$.

Proposition 2. *The re-optimization procedure provides an optimal solution.*

Proof. At the exit of *Step 1*, we have :

$$0 \leq \bar{b} - \underline{b} \leq 1 \implies 0 \leq C^{\mathbb{Z}}_{max} - \underline{b} \leq 1$$

As demonstrated in Lemma 1, having $C^{\mathbb{Z}}_{max}$ does not impact on the optimal solution. The same applies when we consider the same optimization problem but with a tolerance on the absolute gap. Let us now show that *Step 2* allows us to conclude on the optimal solution by enumerating all possible values of $\lceil Z^* \rceil$

I. $\bar{b} = \underline{b} \implies \lceil Z^* \rceil = C^{\mathbb{Z}}_{max}$. The solution is already optimal

II. $\bar{b} = \underline{b} + 1$ and $\underline{b} = \lceil \bar{b}^* \rceil \implies \lceil Z^* \rceil = C^{\mathbb{Z}}_{max} - 1$. Under constraint (23), a better solution is found given that : $C^{\mathbb{R}}_{max} \leq \underline{b} = C^{\mathbb{Z}}_{max} - 1$.

III. $\bar{b} = \underline{b} + 1$ and $\underline{b} = \lceil \underline{b^*} \rceil - 1 \implies \lceil Z^* \rceil = C^{\mathbb{Z}}_{max}$. Applying constraint (23) proves that no solution better than $\bar{b}$ exists given that in this case : $C^{\mathbb{R}}_{max} > \underline{b} = C^{\mathbb{Z}}_{max} - 1$.

□

Remark : In *Step 2*, adding cut branching on fractional lower bound involves representing $C^{\mathbb{R}}_{max}$ as a continuous variable. This ensures the continuity of the lower bounds obtained through continuous relaxation. This could improve the algorithm performance since the branching will always be made on the node with the lowest lower bound. The possibility of choosing arbitrarily between two bounds close to the same integer will therefore be excluded.

Improving Step 2 by adding constraints : To enhance the search for a better solution than that found in *Step 1* ($ST1$), we can require that :
$C^{\mathbb{R}}_{max} \leq \underline{b}.zf_{ip}, \forall i \in I \qquad (23')$

Proposition 1 can be adapted to show that replacing the constraint (23) with constraint (23′) reduces the search space of the decision problem formulated as ($ST2$). We denote the second step of re-optimization, which uses the constraints (23′), as ($ST2'$), and we refer to the resulting model as (MR2).

4.4.3 Modifications to the initial formulations

We aim to reformulate constraints (4) by analyzing the structure of feasible solutions provided by the initial model. We are particularly interested in examining the CT1 and CT2 constraints to extend Proposition 1.

Observation about constraints (4) and (5) : Since using these constraints, if the end date of a work package i coincides with the end of a period p, we can encounter two types of solutions :

1. **Direct Period Completion (DC)** : This refers to the possibility where : $zf_{ip} = 1$. It indicates that the work package i completes exactly at the end of period p.
2. **Overlapping Period Completion (OC)** : In this case : $zf_{ip} = 0$ and $zf_{ip+1} = 1$. This reflects a situation where the work package completion is acknowledged in the transition between the two periods.

We aim to establish a fictive frontier between the end of period p and the start of the following period $p + 1$. When a work package i is scheduled at the beginning of a period, there's a potential risk of overlap, denoted as (OC). To mitigate this risk, we replace the constraints (4) with the following :

$$tf_i \geq \sum_{k=1}^{p} D_k \times (1 - zf_{ip}) + \delta \times (1 - zf_{ip}), \quad \forall i \in I, p \in P. \quad (4')$$

This reformulation allows for the neglecting of the overlap (OC) possibility for all work packages without impacting the optimal value, provided that : $\delta \leq \min\limits_{p \in P, i \in I} \{tf_i - \sum_{k=1}^{p} D_k \mid \sum_{k=1}^{p} D_k < tf_i\}$.

tf being a vector of the end dates for the work packages of an optimal solution in the initial model. The modified version will be referred to as the improved model.

Proposition 3. $\exists\ \delta > 0$ *such that replacing constraints* (4) *with constraints (4′) provides a correct formulation for RCCP, which reduces the search space resulting from the initial model.*

Proof. Let x be a feasible schedule for the initial model. We will show that we can create an equivalent feasible schedule x' by adjusting the binary variables and maintaining the continuous variables at their current values. Let us enumerate all the possibilities, based on the positioning of the end dates for the work packages :

1. $\exists\ j \in I, \exists\ q \in P, \sum_{k=1}^{q-1} D_k + \delta \leq tf_j < \sum_{k=1}^{q} D_k \implies zf_{jq} = 1$. The constraints (4) ensure that $\sum_{k=1}^{q-1} D_k \leq tf_j$, and the constraints (4′) ensure that $\sum_{k=1}^{q-1} D_k + \delta \leq tf_j$. The choice of $\delta \leq \min_{p \in P, i \in I} \{tf_i - \sum_{k=1}^{p} D_k \mid \sum_{k=1}^{p} D_k < tf_i\}$ ensures that tf_j does not violate constraints (4′).
2. $\exists\ j \in I, \exists\ q \in P,\ tf_j = \sum_{k=1}^{q} D_k,\ zf_{jq} = 1$. We derive bounds identical to those in case (1). This ensures that tf_j does not violate constraints (4′).
3. $\exists\ j \in I, \exists\ q \in P,\ tf_j = \sum_{k=1}^{q} D_k$ and $zf_{jq} = 0$. In this case, tf_j does not satisfy the constraints (4′) given that $\sum_{k=1}^{q} D_k + \delta \leq tf_j$, which necessitates a shift of tf_j by δ. By setting $zf_{jq} = 1$, tf_j can be recalibrated to $\sum_{k=1}^{q} D_k$, respecting the constraints (4′). Indeed, by substituting into constraints (4′) and (5), we obtain : $\sum_{k=1}^{q-1} D_k + \epsilon \leq tf_j \leq \sum_{k=1}^{q} D_k$. To prove that the new solution is feasible for the initial model, we encounter the same situation as in the proof of Proposition 1, where $tf_j = F$, F being the duration of the project. If $tf_j < F$, we must verify the validity of constraints (7). In such case : $zs_{jq+1} = 1$. Upon substitution, we find : $d_{jq+1} = 0$, confirming this as a valid bound. For further verification, the same methodology used in the proof of Proposition 1 is applied to convert the solution x into x'.

From (1), (2), and (3), we deduce the result of the proposition. □

Remark 1 : To guarantee the optimal solution, we can assign any value to δ within the interval $]T, \min_{p \in P, i \in I} \{tf_i - \sum_{k=1}^{p} D_k \mid \sum_{k=1}^{p} D_k < tf_i\}[$. tf being a vector of the end dates for the work packages of an optimal solution in the initial model. Here, T signifies the feasibility tolerance used in the MIP solver, representing the maximum extent to which the basic variables of a model are allowed to deviate from their specified bounds.

Remark 2 : We can also prove with the same principle that constraints (1) can be reformulated as follows :

$$ts_i \geq \sum_{k=1}^{p} D_k \times (1 - zs_{ip}) + \delta \times (1 - zs_{ip}), \quad \forall i \in I, p \in P. \quad (1')$$

Remark 3 : $\lim_{\delta \to 0} Z^*_{x_\delta} = Z^*_x$; $Z^*_{x_\delta}$ and Z^*_x respectively being the optimal values of the improved model (with a fixed δ) and the initial model. x_δ and x respectively being the optimal solutions of the improved and the initial model.

Remark 4 : The improved model is more adaptable for problems where the end date is preferably not positioned at the beginning of a new period. It may be impractical to allocate a very small workload of a work package to a new period.

4.4.4 Optimizing the Bi-objective function : A Practical Approach

This section addresses the trade-off between the Resource-driven and Time-driven variants. It aims to demonstrate the practicality of our approach in resolving real-world scenarios that require a compromise between the two objectives. As such, we propose a bi-objective method that aims to identify a Pareto frontier, providing a range of weakly non-dominated solutions.

4.4.4.1 Description of the solution approach : ϵ-constraint method

To tackle this bi-objective problem, we have employed the ϵ-constraint method, leveraging previous research. This approach for multi-objective problems involves several steps. This method selects one objective for optimization while treating the other as a constraint. Introducing the parameter ϵ, we define an acceptable level of constraint violation. Then, we optimize the selected objective within the defined constraint. Repeating this process with different values of ϵ generates a range of weakly non-dominated solutions.

We choose to fix the value of makespan minimization as a constraint in each iteration. This choice is motivated by the fact that we rely on discrete values for the project duration, which limits the number of optimizations required. This approach is outlined in Algorithm 1.

Algorithm 1: ϵ-constraint Algorithm for identifying the Pareto frontier

Input : Instance, Improved model
Output : P (a set of weakly non-dominant solutions)

$c = 0$;
Resolve the Resource-driven using the improved model ;
$P = \emptyset$;
$\epsilon = C^{\mathbb{Z}}_{\max}$;
$P = P \cup (\epsilon, c)$;
$\epsilon = \epsilon - 1$;
`// Let` S_ϵ `be the search space of the Time-driven problem,` $F = \epsilon$
while $S_\epsilon \neq \emptyset$ **do**
    Add the constraints (22′) to resolve the Time-driven : $tf_i \leq \epsilon \cdot zf_{ip} \quad \forall i \in I$;
    `//` ϵ `strictly included in` p `or coinciding with the end of` p`.`
    Obtain the optimal value c ;
    $P = P \cup (\epsilon, c)$;
    $\epsilon = \epsilon - 1$;
end

We denote the improved model used to tackle the bi-objective case as (MB2).

4.4.4.2 Pareto Front Analysis : Min-Max Method

To select the best trade-off between the two objectives, the Min-Max Method was employed. This method involves identifying the ideal solution for each objective function. The ideal solution ($C_{max_{\text{ideal}}}$, $Cost_{\text{ideal}}$) is defined as the point with the **minimum value** for both criteria : `Time-driven` ($C_{max_{\min}}$) and `Resource-driven` ($Cost_{\min}$). Both objectives were then **normalized** using the following formulas :

$$\text{Normalized } C^s_{max} = \frac{C^s_{max} - C_{max_{\text{ideal}}}}{C_{max_{\max}} - C_{max_{\min}}}$$

$$\text{Normalized } Cost^s = \frac{Cost^s - Cost_{\text{ideal}}}{Cost_{\max} - Cost_{\min}}$$

This normalization ensures that both objectives are scaled between 0 and 1, making them directly comparable. We calculated the **Min-Max distance** for each point s on the Pareto front. The Min-Max method focuses on the **maximum deviation** from the ideal solution :

$$\text{Min-Max Distance} = \max_{s \in P} \left(\text{Normalized } C^s_{max}, \text{Normalized } Cost^s\right)$$

For each point, we identified the largest deviation between the normalized C^s_{max} and $Cost^s$. The goal of the Min-Max method is to minimize this maximum deviation. The point with the **smallest Min-Max distance** was identified as the **best compromise** solution.

4.5 Methodological Framework

This section presents the process followed to implement, validate, and test the RCCP model. Our framework consists of : instance generation, resolution approach, and validation.

4.5.1 Instance Generation

We used two main sets of instances for our experiments :

1. **Cherkaoui's aggregated instances :** we utilize the aggregated instances proposed by [32] to address the Time-driven variant. These theoretical instances, originally derived from De Boer's work [39], are modified to assess RCCP problems. Each class of these instances is distinguished by two parameters : the number of work packages, admitting three possible values ($|I|$ = 10, 20 or 50) and the number of resources ($|R|$= 3, 10 or 20). The initial four periods are detailed, each having a duration of 1. Subsequent periods are allocated a duration, noted as Δ. Cherkaoui sets Δ to 4. If the total horizon is not evenly divided by Δ, the duration of the fifth period is adjusted to the remainder of the horizon divided by Δ.

   We further modify the duration Δ to values 1, 2, and 3 to test our model in these varied scenarios. For every specific value of Δ, 450 instances are examined, resulting in a total of 1800 instances.
2. **Galmard's tactical overhaul project instances [43]** : These instances are characterized by a limited number of precedence relations and a fixed planning horizon of 40 periods, with each period lasting 4 units. We generated 90 instances for each objective function : resource-driven, time-driven, and bi-objective cases, resulting in a total of 270 instances.

4.5.2 Resolution Approach

The resolution approach aims to efficiently find optimal solutions to the RCCP model while minimizing computation time. We implemented the RCCP model using the IBM ILOG CPLEX 22.1.1 MIP solver. The experiments were conducted on a computing grid equipped with dual Intel Xeon Gold 6258R CPUs (2.70 GHz) and a total memory capacity of 512 GB. The solver was executed using a single thread. The time limits were set as follows :

For Cherkaoui's instances, 50 000 seconds were allocated to ensure a fair comparison and a sufficient number of optimal solutions. For Galmard's instances, we limited the execution time to 5 000 seconds. To experimentally prove that the improved model consistently provides the optimal solution, we propose alternative strategies to solve the initial model, relying on several tools that help us analyze the model's performance and progressively identify the potential causes that slow down the resolution algorithm in obtaining and proving the optimal solution. These causes, which we refer to as anomalies, may be linked to issues such as the degeneracy of Simplex, the weakness of the relaxed model, symmetry in the solutions, etc. The diagnostics for these anomalies during model resolution include information that reflects the tested instances' complexity relative to the chosen resolution strategy, as well as details on the algorithm's progress before execution is stopped. This procedure can be repeated each time causes for the algorithm's slowdown are identified. It is further elaborated upon when addressing the resource-driven variant.

After selecting the best model for the mono-objective case, we extend our analysis to a bi-objective optimization problem. The ϵ-constraint method is applied to manage the two objectives. For instances requiring multi-criteria analysis, Pareto fronts are generated to illustrate the trade-offs between the objectives. To select the best trade-off between the two objectives, the Min-Max Method is employed

4.5.3 Validation Process

We recorded several performance indicators, including :

1. **Execution Time** : The total execution time was averaged across each class of instances.
2. **Optimality Gap** : The optimality gap was calculated to measure the difference between the best integer solution and the relaxed solution at the end of the resolution. CPLEX computes this gap using the formula $\frac{|\text{lower bound} - \text{best integer}|}{(10^{-10} + |\text{best integer}|)}$.
3. **Worst-case Optimality Gap** : The largest observed optimality gap was also tracked to assess the most challenging instances.
4. **Branching Efficiency** : We analyzed the number of nodes explored and the number of dual Simplex iterations, highlighting the algorithm's ability to terminate before the time limit. Additionally, we measured the number of branches executed to find the best integer solution to evaluate the solver's efficiency in identifying optimal solutions.
5. **Instances of Different Nature** : To experimentally demonstrate that the model supports instances of various types, we tested a large number of instances generated by [32], reducing the length of periods, and also tested different instances from [43] in the case of the Resource-driven and Time-driven variants.

To better understand the behavior of each tested model across different types of instances, we conduct a statistical analysis based on visualizations, correlation tests and ANOVA tests. This allows us to examine how factors such as the number of WPs, period lengths, available resources, and precedence relations affect solving times and the number of dual simplex iterations.

4.6 Computational experiments and discussions

This section tests the improved model and the resolution strategies proposed in Section 4.4, compares them with the initial model, and discusses the results obtained, following the resolution approach described in Section 4.5.

4.6.1 Results and discussion

The efficacy of the proposed approaches is closely linked to the solver's behavior in addressing the RCCP model. To this end, we present our findings for each test in a singular table for clarity and coherence. The **Instances** column categorizes the class of instances examined, which is divided into two main elements : **Nb-pack**, indicating the number of work packages, and **Nb-res**, signifying the number of available resources. Each combination of **Nb-pack** and **Nb-res** is represented by two lines within the table. **Tps (sec)**, denotes the total average execution time in seconds. **Opt-gap (%)** is defined as the optimality gap, calculated by the relative tolerance between the upper and lower bounds at the execution's end to calculate the distance between the best integer solution and the best-relaxed solution. Consequently, a gap greater than 0% signifies that the solution's optimality is not guaranteed. **W-gap (%)** represents the largest optimality gap observed. **Nb-nodes** represents the average number of nodes explored in the branch-and-cut algorithm. **Iterations** denotes the necessary number of iterations of the dual Simplex until the algorithm terminates. These last two elements furnish an overarching perspective on the frequency with which CPLEX solves the relaxed sub-problems. Finally, **N-Opt** represents the number of branches executed to find the best integer solution, highlighting the algorithm's efficiency in identifying the optimal solution, especially when it ends before the time limit.

All aspects considered in the model analysis will be averaged over each class of instances. Note that all numbers of fractional averages in our tables are rounded to the nearest integer for elements that are supposed to be integers (like **Nb-nodes**, **Iterations**, etc.). If the execution times are large enough, all numbers representing execution times greater than 100 seconds will be approximated to an integer to visualize the results better. If necessary, depending on the associated results for each strategy, we add additional elements to complete the diagnosis and set a criterion for choosing complex instances to solve (when using the basic model) to have a more precise analysis of the results.

4.6.1.1 Comparing the improved and initial models on Cherkaoui's instances

We initiate our analysis by testing Cherkaoui's instances with $\Delta = 4$, adhering to the premise that larger values of Δ tend to simplify the instances [32]. The first and second rows for each class correspond to the initial and improved model. Without loss of generality, we fix a small value of $\delta = 10^{-4}$ (Table 4.2).

TABLE 4.2 Comparison between initial (init.) and improved (impr.) model for $\Delta = 4$

Instances		**Models**	**Tps (sec)**	**Opt-gap (%)**	**W-gap (%)**	**Nb-nodes**	**Iterations**	**N-Opt**
Nb-pack	**Nb-res**							
10	3	Init.	0.04	0.00	0.00	23	393	3
			0.02	0.00	0.00	3	154	1
	10	Init.	0.05	0.00	0.00	50	628	3
			0.02	0.00	0.00	2	156	0
	20	Init.	0.07	0.00	0.00	74	847	9
			0.03	0.00	0.00	3	230	0
20	3	Init.	0.54	0.00	0.00	649	7 939	177
			0.16	0.00	0.00	104	1 702	25
	10	Init.	0.75	0.00	0.00	654	9 125	193
			0.16	0.00	0.00	52	1 347	21
	20	Init.	1.28	0.00	0.00	902	13 173	222
			0.18	0.00	0.00	61	1 480	17
50	3	Init.	142	0.00	0.00	105 157	874 835	18 102
			9.15	0.00	0.00	2 548	69 428	903
	10	Init.	652	0.00	0.00	190 120	7 348 140	41 793
			19.06	0.00	0.00	2 918	138 517	1 791
	20	Init.	2 486	0.00	0.00	336 154	23 567 400	285 938
			23.44	0.00	0.00	3 642	206 424	1 370

The improved model demonstrates clear advantages over Cherkaoui's model, particularly in terms of execution time. In some cases, it has been observed to be up to 100 times faster. Notably, these impressive results were achieved with relatively uncomplicated instances.

Moving forward, we will investigate the performance of both models using disaggregated instances ($\Delta = 1$) (Table 4.3). This experiment will provide valuable insights into their relative strengths and weaknesses when dealing with more detailed data.

TABLE 4.3 Comparison between initial (init.) and improved (impr.) model for $\Delta = 1$

Instances		Models	Tps	Opt-gap	W-gap	Nb-nodes	Iterations	N-Opt
Nb-pack	Nb-res		(sec)	(%)	(%)			
10	3	Init.	0.19	0.00	0.00	159	2 225	27
		Impr.	0.07	0.00	0.00	35	538	3
	10	Init.	0.17	0.00	0.00	97	1 522	17
		Impr.	0.09	0.00	0.00	35	633	3
	20	Init.	0.19	0.00	0.00	102	1 978	29
		Impr.	0.10	0.00	0.00	21	641	6
20	3	Init.	25.90	0.00	0.00	12 289	270 503	4 031
		Impr.	4.58	0.00	0.00	5 424	71 205	116
	10	Init.	33.37	0.00	0.00	10 819	300 151	3 705
		Impr.	1.22	0.00	0.00	582	10 860	147
	20	Init.	35.78	0.00	0.00	7 967	252 344	2 773
		Impr.	1.51	0.00	0.00	471	10 645	67
50	3	Init.	31 231	0.79	6.00	2 181 490	146 804 000	1 230 430
		Impr.	3 762	0.03	1.05	835 131	25 824 900	65 785
	10	Init.	38 845	1.67	6.60	1 084 610	147 582 000	691 851
		Impr.	3 370	0.02	0.30	318 170	13 295 900	68 148
	20	Init.	45 345	1.43	9.00	1 236 910	157 487 000	873 090
		Impr.	4 754	0.04	1.00	135 315	14 024 500	59 480

Building upon the previous results, the improved model consistently performs better than the initial model, particularly in instances with 50 work packages. Our observations reveal a significantly lower number of nodes explored by the improved model while achieving optimality. Additionally, the optimality gap is consistently smaller.

Following the improved model's success, we propose two approaches for further evaluation :

1. We aggregate the time periods from the disaggregated instances and then apply both the initial and improved models to these instances. This method is anticipated to resolve a greater number of instances but might result in less realistic solutions due to the change in granularity.

   Table 4.4 compares the performance of initial and improved models across different scenarios, delineated by the Δ parameter values ranging from 1 to 4.

   The analysis with the improved model reveals average deviations from the optimal solution. Where negative percentages indicate a shortfall from the optimal cost relative to the disaggregated instances, a 0% deviation represents perfect accuracy. It is crucial to note that wherever the initial model established optimality, the improved model always replicated these results.

   As Δ decreases, suggesting increased complexity, the initial model's effectiveness in finding optimal solutions declines, evident from the decreasing number of resolved instances (#OPT-Initial) with lower Δ values.

The improved model demonstrates enhanced performance, consistently achieving or approaching the optimal solution (#OPT-Improved) across all Δ settings, indicating its superior ability to adapt to various levels of complexity.

TABLE 4.4 Comparative performance of initial and improved models across different Δ settings

Instances tested	**M-Dev (%)**	**#OPT-Initial**	**#OPT-Improved**
Δ=4	-22.71	150	150
Δ=3	-15.91	147	150
Δ=2	-7.70	127	150
Δ=1	0.00	48	145

2. We evaluate the effects of modifications made to the initial model and compare them with the outcomes of aggregating periods together in terms of execution time, given that the first approach preserves a realistic cost (Figure 4.3).

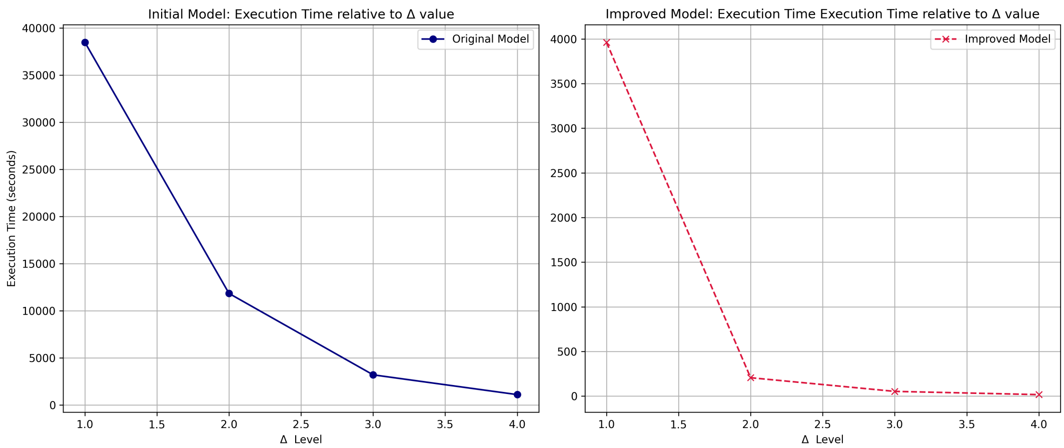

FIGURE 4.3 Impact of Δ value over execution time : Initial model vs Improved model

Remark : The vertical scale of the initial model is ten times that of the improved model.

We infer that the improved model is as sensitive to changes in the Δ value as the initial model, indicating that the performance gap between the two models remains consistent even as Δ increases.

Effect of resource availability on solving time : We grouped the instances based on the number of WPs (10 WPs, 20 WPs, and 50 WPs) with varying numbers of resources and tested the correlation between the execution times of the initial and improved models. The Pearson-correlation tests were performed as follows :

— 10 WPs instances : Correlation = 0.552, p-value = 3.52×10^{-49}

— 20 WPs instances : Correlation = 0.521, p-value = 4.69×10^{-43}

— 50 WPs instances : Correlation = 0.294, p-value = 1.95×10^{-13}

The relatively weak correlation for the 50 WPs instances prompted us to further examine the behavior of the solver in both models. We calculated the average execution time for both the initial and improved models and generated the bar chart in Figure 4.4.

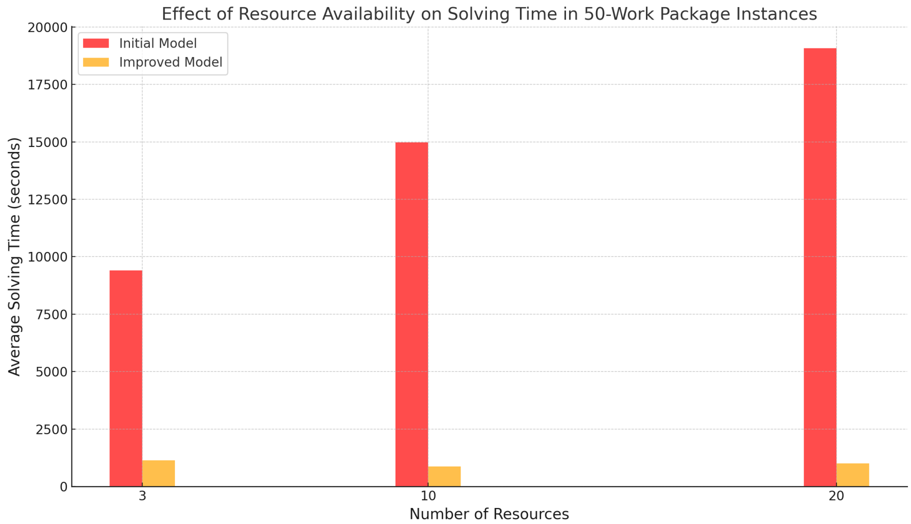

FIGURE 4.4 Impact of the number of resources on execution time

By performing a statistical analysis using ANOVA, we obtained the following results for 50 WPs :

— **Initial Model :** F-statistic = 10.84, p-value = 2.38e-05 (significant)

— **Improved Model :** F-statistic = 0.13, p-value = 0.880 (not significant)

Remark : The p-value is considered significant when it is less than or equal to 0.05.

This indicates that the execution time of the improved model is relatively stable, in contrast to the initial model. The improved model's performance remains unchanged even with varying resource numbers.

We now explore further instances to evaluate better the impact stemming from changes.

4.6.1.2 Analysing and testing instances of different nature

We test Galmard's instances [43] characterized by a limited number of precedence relations. Rather than spending extensive time to obtain an optimal solution, we suggest alternate approaches. These methods stem from an understanding of the solver's performance or direct observations, as elaborated in Section 4.4. The main goal is to obtain the optimal solutions to assess the effectiveness and enhancements brought by the improved model.

Time-driven variant : First empirical result

We evaluate three methods. Initially, we apply the (MT1) approach. Without loss of generality, we set $L = 16$ for all the instances. Given the shorter horizon and fewer precedence relations in these instances compared to previous cases, it is possible for many work packages to terminate in L. Under such conditions, the (MT2) formulation might offer potential advantages. It's important to note that this method theoretically guarantees the attainment of the optimal solution. Lastly, we assess the improved model and compare the optimal solutions derived from both the (MT2) formulation and the improved model. The first, second, and third rows for each category correspond respectively to the initial (MT1), (MT2) (based on *Strategy 1*) and the improved model (Table 4.5).

TABLE 4.5 Comparison of the initial (MT1), Strategy 1 (MT2) and improved (impr.) models for the Time-driven variant

Instances		Models	Tps (sec)	Opt-gap (%)	W-gap (%)	Nb-nodes	Iterations	N-Opt
Nb-pack	Nb-res							
10	3	MT1	288	0.00	0.00	1 496 020	10 718 400	58 452
		MT2	0.02	0.00	0.00	0	177	0
		Impr.	0.03	0.00	0.00	0	189	0
	10	MT1	6.23	0.00	0.00	17 398	208 731	239
		MT2	0.03	0.00	0.00	10	346	9
		Impr.	0.03	0.00	0.00	1	234	1
	20	MT1	0.49	0.00	0.00	1 051	18 698	101
		MT2	0.03	0.00	0.00	7	310	4
		Impr.	0.02	0.00	0.00	0	220	0
20	3	MT1	4 058	61.00	100.00	11 396 000	87 612 100	2 876 820
		MT2	0.03	0.00	0.00	0	272	0
		Impr.	0.03	0.00	0.00	0	318	0
	10	MT1	4 316	22.00	57.27	8 009 830	107 806 000	1 091 700
		MT2	0.08	0.00	0.00	27	948	26
		Impr.	0.07	0.00	0.00	9	673	8
	20	MT1	115	0.00	0.00	192 776	3 260 990	727
		MT2	0.07	0.00	0.00	31	785	29
		Impr.	0.05	0.00	0.00	4	396	3
50	3	MT1	4 056	78.00	100.00	6 916 520	59 070 100	222 389
		MT2	0.08	0.00	0.00	0	679	0
		Impr.	0.09	0.00	0.00	0	810	0
	10	MT1	4 554	89.00	100.00	5 464 910	72 753 300	1 365 740
		MT2	0.13	0.00	0.00	3	820	3
		Impr.	0.16	0.00	0.00	3	903	3
	20	MT1	4 547	62.00	100.00	3 176 406	60 212 500	1 526 230
		MT2	0.35	0.00	0.00	35	2 265	34
		Impr.	0.34	0.00	0.00	19	1 912	19

Remark : A gap of 100% means that the lower bound equals 0.

It should be noted that 42 instances have not been resolved using the initial model. A very large optimality gaps characterize the instances. The cuts generated by the solver at the nodes are insufficient to obtain a good integrality gap at the end of the resolution algorithm.

The (MT2) formulation (Strategy 1) demonstrates significant efficiency. The gain in execution time for strategy 1 compared to the initial model is approximately 99.996%. This can be attributed to the fact that many completion dates are closely aligned with the value L. As a result, the constraints introduced may generate cuts that impact a broader range of variables.

Regarding the improved model, it is significant to note that its optimal values are also optimal for the initial model across all tested instances. This model generally exhibits a modest improvement over the (MT2) formulation. It is noteworthy that in situations where there are few work packages whose end dates are close to L (as in Cherkaoui's instances), the improved model performs better without losing any optimal information, especially when the boundaries between two periods are quite small.

Effect of resource availability on number of dual simplex iterations : We illustrate the instances for each WP size (10 WPs, 20 WPs, and 50 WPs) using the (MT1), (MT2), and improved models. The chart in Figure 4.5 compares how resource availability affects the performance of each model. Note that we use the number of iterations as the performance criterion because the execution time is relatively short and can be influenced by machine variability. The number of iterations provides a more reliable indicator of performance.

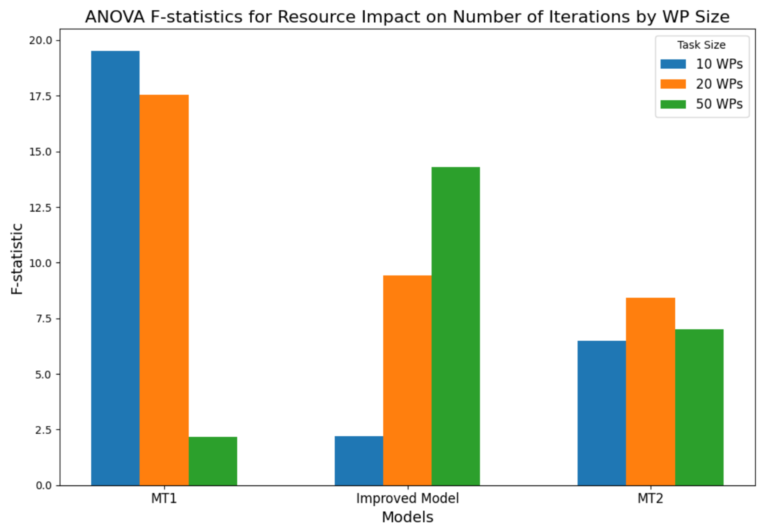

FIGURE 4.5 Impact of the number of resources on dual simplex iterations : Time-driven variant

These results show that most models exhibit a significant effect, except for the improved model in the case of 10 WPs and the initial model in the case of 50 WPs. This can be explained by many instances reaching the time limit before completing execution. For (MT2), we observe that it is not sensitive to the number of resources as the WP size increases, indicating that this model better supports the structure of the resource distribution.

We now explore further instances to evaluate better the impact stemming from changes.

4.6.1.3 Makespan minimization

We also applied another approach dedicated to proving the optimal solution for all these instances. The objective is to demonstrate experimentally that our choice of δ always leads to optimal solutions.

This section is dedicated to solving the makespan minimization problem. We have developed three consecutive approaches, where each subsequent approach builds on the gains from the previous one and incorporates additional observations. Let's begin by testing the initial formulation (MR1).

Resource-driven variant : Second empirical result

As mentioned, we begin by examining the solver's performance. Consequently, we present Table 4.6, which displays the outcomes related to complex instances.

Analysis of complex instances : To have more precise information on the possible cause slowing down the resolution of certain instances and for comparison purposes between different resolution strategies, we also consider a set of complex instances to solve. The criteria used to identify these instances are :

— Instances that have not been resolved within a 5 000 second limit. This will allow to analyze more precisely the causes that can slow down the resolution of these types of instances. This is because calculating the average value of each element for all instances in the class might not reflect the real difficulty of complex instances (if a large number of instances are easy in the same class).

— Instances that require at least 1 million nodes to explore or 10 million iterations of the dual Simplex (very large number of nodes and iterations) without reaching the 5 000 second limit time. This will allow having a more exact measure of the impact of the next changes made at the level of the model resolution, since if we would only consider the instances taking 5 000 seconds, we will not know how many nodes will require the resolution of an instance (with a 0% optimality gap).

TABLE 4.6 Results associated with complex instances by testing *(MR1)* formulation

Instances			Tps (sec)	Opt-gap %	Nb-nodes	Iterations	N-Opt
Nb-pack	**Nb-res**	**Num-inst**					
20	3	1	5 000	0.05	8 666 348	177 966 539	8 189 918
		2	5 000	2.17	12 955 968	196 733 340	12 352 867
		7	822	0.00	1 834 935	25 318 049	121 351
		8	272	0.00	486 447	10 120 642	4 961
50	3	1	5 000	2.69	3 356 736	100 114 527	22 166
		2	5 000	1.03	4 897 226	96 807 085	339 246
		3	5 000	2.48	5 377 873	89 285 194	10 017
		4	5 000	2.89	3 970 137	95 486 782	14 837
		5	5 000	4.38	4 447 767	74 409 465	4 241 867
		6	5 000	3.29	4 397 168	90 219 266	4 225 835
		7	5 000	2.01	10 406 045	140 796 624	10 238 445
		8	5 000	3.76	1 736 521	41 176 010	23 002
		9	5 000	0.62	5 720 515	144 289 514	5 424 828
		10	5 000	0.22	2 576 178	47 235 396	2 504 178
	10	2	5 000	1.49	2 453 757	68 224 129	6 873
		3	954	0.00	1 028 096	17 490 203	22 091
		4	5 000	1.00	1 528 974	87 980 703	940
		5	2 942	0.00	3 236 983	53 652 209	1 248
		6	2 314	0.00	4 982 806	31 282 539	1 674
		7	5 000	1.00	1 528 974	87 980 703	940
		8	5 000	1.78	4 974 159	94 882 025	245 516
		9	164	0.00	140 435	1 964 715	14 485
		10	1 074	0.00	1 252 222	15 447 463	1 320

This table shows that the difficulty in solving instance classes increases when the number of work packages increases. However, the difficulty is not just related to this aspect. Indeed, a limited number of resources could further complicate the problem. 16 instances, mainly characterized by a fairly large number of work packages and limited resources, have yet to be resolved. Instances requiring 20 resources are clearly easier to solve than instances of 3 or 10 resources. This could be because the number of possible permutations in this type of instance is greater when there are fewer resources. We also note that the solver generates a considerably large number of nodes and iterations in 5 000 seconds. However, the optimality gaps are small, meaning the upper bound is quite close to the optimal value.

First observation (F1) : By combining the previous information, namely **Opt-gap** and **N-Opt**, these elements lead us to look for another interesting information : the ***absolute optimality gap*** calculated by CPLEX, consisting of finding the absolute tolerance between the upper and lower bound ($|best\ bound - best\ integer|$) on all 16 unresolved instances. We calculate the maximum absolute optimality gap found in all these instances.

We illustrate the improvement of the largest absolute optimality gap found, which is the gap associated with the 50-3-5 instance (**Nb-pack-Nb-res-Num-inst**) as a function of the number of nodes in Figure 4.6. For this purpose, it is established that the absolute gap remains within one unit of time for all instances. [2]

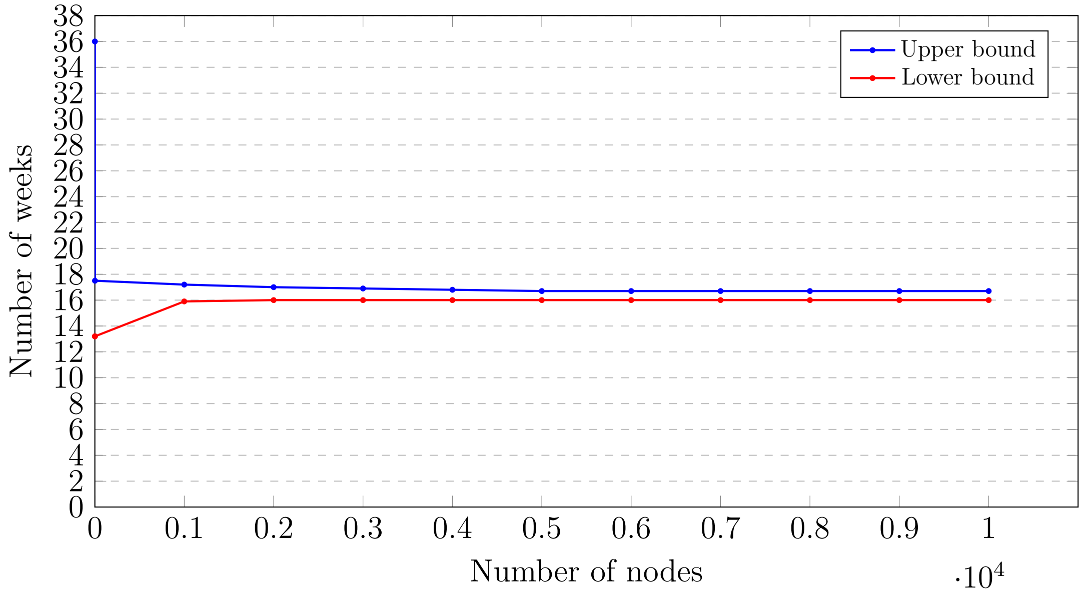

FIGURE 4.6 Curves illustrating the largest absolute optimality gap found

Second observation (F2) : We note that replacing $C_{max}^{\mathbb{R}}$ by $C_{max}^{\mathbb{Z}}$ does not fully address the issue in the case of 50-3-5 instance, as the lower bound is already close to an integer value. Consequently, in numerous cases, this adjustment fails to solve the problem effectively ; the solver often requires extensive time either to reduce the solution to match the value of a lower bound or to enhance the lower bound itself to demonstrate optimality. Moreover, the lower bound aligns with the end of a period since, in these instances, $\Delta = 4$. Including constraints $(23')$ could be advantageous in cutting the branching.

To illustrate the problem encountered prohibiting the resolution of this category of instances, we replace $C_{max}^{\mathbb{R}}$ by $C_{max}^{\mathbb{Z}}$ and calculate the average upper and lower bound of the unresolved instances as a function of the number of nodes until the integrality gap no longer decreases (after less than 10 000 nodes only, the integrality gap no longer changes in all instances).

Figure 4.7 illustrates the improvement of the average lower and upper bounds.

2. Importantly, this absolute optimality gap should be distinguished from the relative optimality gap, referred to as **Opt-gap**

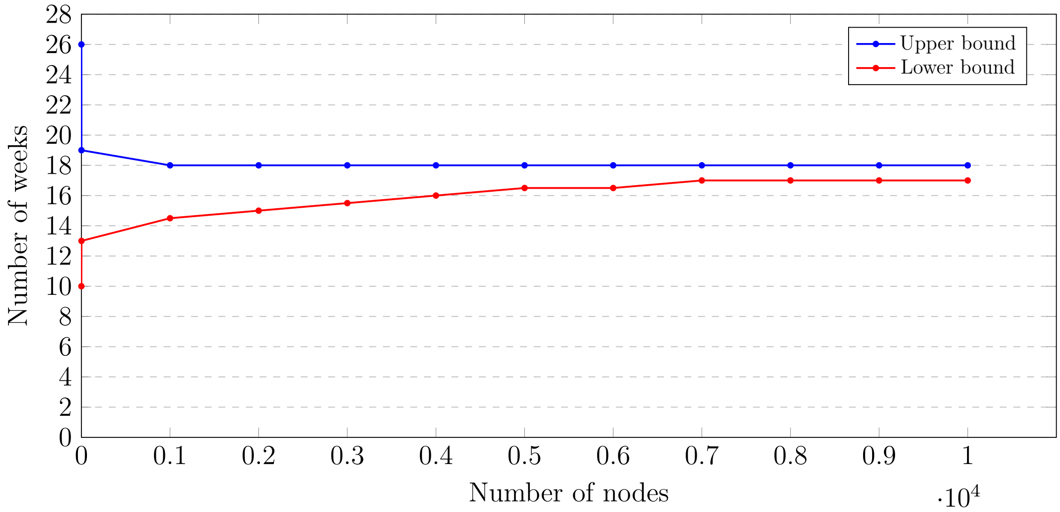

FIGURE 4.7 Curves illustrating the average absolute gap for unresolved Galmard's instances using $C_{max}^{\mathbb{Z}}$

This brings us to another observation.

Third observation (F3) : The necessary number of nodes to reach an absolute optimality gap of at most 1 is very small relative to the total number of nodes. Based on the previous remark, we propose another way to solve the problem.

As discussed in Section 4.4, we implement a re-optimization strategy that segments the problem into two distinct phases (Strategy 2). The initial phase is identified as the more straightforward component of the problem, whereas the second phase represents the hardest step. In addressing the latter, we further our understanding of the lower bound before applying constraints (23′). We use the (MR2) approach to obtain an optimal solution for the initial model. We compare it with the improved model. Consequently, Table 4.7 illustrates the behavior of the solver. For each class of instances, the first, second, and third rows correspond to the results obtained using the (MR1), Strategy 2, and improved formulations.

Table 4.7 Comparison of the initial (MR1), Strategy 2 (Strat2.) and improved (impr.) models for the Resource-driven variant

Instances		Models	Tps	Opt-gap	W-gap	Nb-nodes	Iterations	N-Opt
Nb-pack	Nb-res		(sec)	(%)	(%)			
10	3	MR1	2.14	0.00	0.00	5 295	64 524	562
		Strat2.	0.40	0.00	0.00	56	3 114	20
		Impr.	0.41	0.00	0.00	34	2 508	33
	10	MR1	0.37	0.00	0.00	213	4 966	72
		Strat2.	0.27	0.00	0.00	15	2 053	15
		Impr.	0.29	0.00	0.00	5	1 341	5
	20	MR1	0.16	0.00	0.00	9	1 290	4
		Strat2.	0.14	0.00	0.00	1	952	1
		Impr.	0.16	0.00	0.00	0	617	0
20	3	MR1	1 179	0.23	2.17	2 513 750	42 390 100	2 077 570
		Strat2.	1.14	0.00	0.00	279	9 279	110
		Impr.	0.71	0.00	0.00	88	5 492	68
	10	MR1	13.57	0.00	0.00	20 656	314 473	1 523
		Strat2.	1.34	0.00	0.00	236	11 489	75
		Impr.	0.75	0.00	0.00	79	5 418	75
	20	MR1	0.63	0.00	0.00	328	6 037	42
		Strat2.	0.46	0.00	0.00	5	2 349	4
		Impr.	0.32	0.00	0.00	0	1 316	0
50	3	MR1	4 534	2.32	4.38	4 431 000	87 258 800	2 454 030
		Strat2.	3.29	0.00	0.00	843	24 593	449
		Impr.	3.41	0.00	0.00	1 659	37 563	1 657
	10	MR1	2 762	0.45	1.78	2 226 410	41 891 200	280 006
		Strat2.	4.17	0.00	0.00	626	23 767	201
		Impr.	3.37	0.00	0.00	332	16 615	318
	20	MR1	36.58	0.00	0.00	33 697	615 121	251
		Strat2.	3.81	0.00	0.00	356	23 632	104
		Impr.	2.06	0.00	0.00	127	9 417	122

We observe that the algorithm demonstrates significantly improved performance with both Strategy 2 and the improved model, successfully resolving all instance classes on average in under 10 seconds. The percentage gain in solving time for Strategy 2 compared to the initial model is approximately 99.83%. There has been a notable reduction in the number of nodes. The complexity of several instances was primarily due to the positioning of work packages, which were often only one unit away from the start of periods, thereby complicating the process of finding a lower bound capable of efficiently pruning all remaining nodes to prove the optimality of the solutions.

Furthermore, it is noteworthy that the optimal solutions identified for the improved model are also optimal for the original model in all tested instances. This indicates that the value of δ is sufficiently small to almost guarantee obtaining the optimal solution. Additionally, this small value of δ effectively reduces the number of nodes that need to be examined, further enhancing the efficiency of the solution process.

Effect of resource availability on model performance : We illustrate the effect of the number of resources for each WP size and compare (MR1), (MR2), and the improved model. The chart in Figure 4.8 compares how resource availability affects the performance of each model. The complexity criterion is the number of dual simplex iterations.

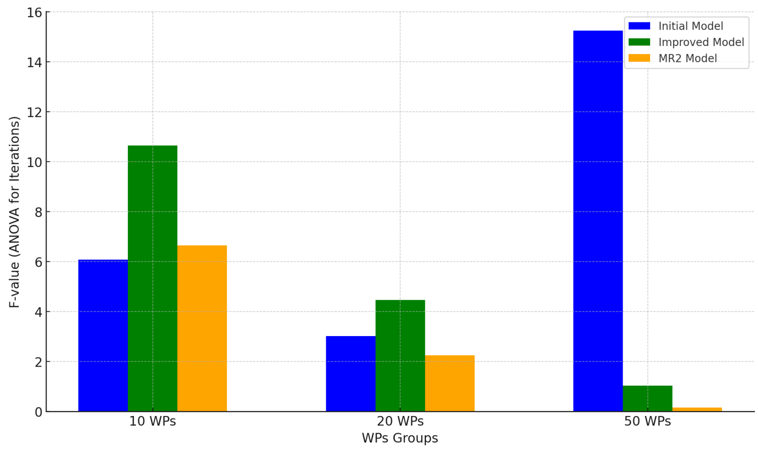

FIGURE 4.8 ANOVA F-values for number of iterations across models : Resource-driven variant

The results demonstrate that the initial model shows a more significant change in iterations for the 50 WPs group. The improved model performs more consistently across different WP sizes, showing moderate F-values and suggesting more stable iteration requirements. The MR2 model, with lower F-values, indicates the least variation in the number of iterations needed across different WP sizes, particularly for 50 WPs, where it maintains greater stability in performance.

We now explore further instances to evaluate better the impact stemming from changes.

4.6.1.4 Optimizing the Bi-objective function

This section addresses the trade-off between the Resource-driven and Time-driven variants. We aim to demonstrate the practicality of our approach in resolving real-world scenarios that require a compromise between the two objectives. As such, we test a bi-objective method using the formulation (MB2) to identify a Pareto frontier, providing a range of weakly non-dominated solutions. It is worth noting that if $\epsilon\%4 \neq 0$, we consider the period length, including ϵ, to be $\epsilon\%4$. For example, if $\epsilon = 17$, the fifth period will equal 1 instead of 4. This adjustment will not change the optimal value, as the remaining part of the fifth period (above ϵ) will not be used. This adjustment allows us to incorporate the constraints $(22')$ efficiently.

Therefore, we present the elements for evaluating the performance of the improved model (MB2), described in Section 4.6.1. During the testing of each instance, we accumulate the total values of these elements for all optimization processes using algorithm 1. Additionally, these elements are averaged across each instance class. The obtained results are as follows (Table 4.8) :

TABLE 4.8 Result associated with the bi-objective case

Instances		**Tps**	**Opt-gap**	**W-gap**	**Nb-nodes**	**Iterations**	**N-Opt**
Nb-pack	**Nb-res**	**(sec)**	**(%)**	**(%)**			
10	3	0.57	0.00	0.00	56	4 058	55
	10	0.46	0.00	0.00	13	2 846	11
	20	0.43	0.00	0.00	3	2 142	0
20	3	1.07	0.00	0.00	61	6 880	49
	10	1.49	0.00	0.00	112	9 891	104
	20	1.55	0.00	0.00	73	8 614	49
50	3	3.71	0.00	0.00	1 660	40 410	1 658
	10	4.45	0.00	0.00	344	21 651	329
	20	5.37	0.00	0.00	230	22 276	219

As anticipated, despite performing multiple sequential optimizations, obtaining the Pareto frontier can be accomplished within a few seconds. This is primarily due to the relatively straightforward nature of solving each Time-driven optimization after modifying the model. Additionally, the discrete nature of the project aids in reducing the number of optimization processes required.

Analysis of Pareto Front

The Pareto front was plotted for each selected instance, with the best compromise solution calculated using the Min-Max method, highlighted in **blue**. This visualization shows which solutions offer the optimal balance between the two objectives. Two types of instances were selected : those with a relatively high number of non-dominated points and those with the largest Euclidean distance between the best solutions for each objective (Figures 4.9 and 4.10).

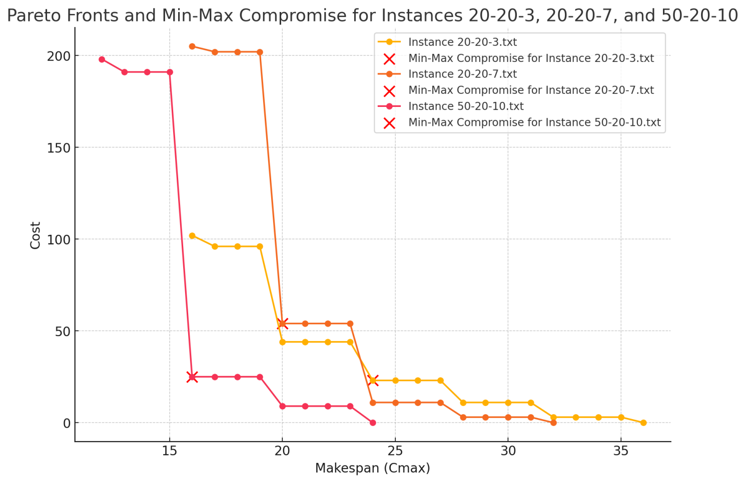

FIGURE 4.9 Pareto front for instances with a fairly good number of non-dominated points

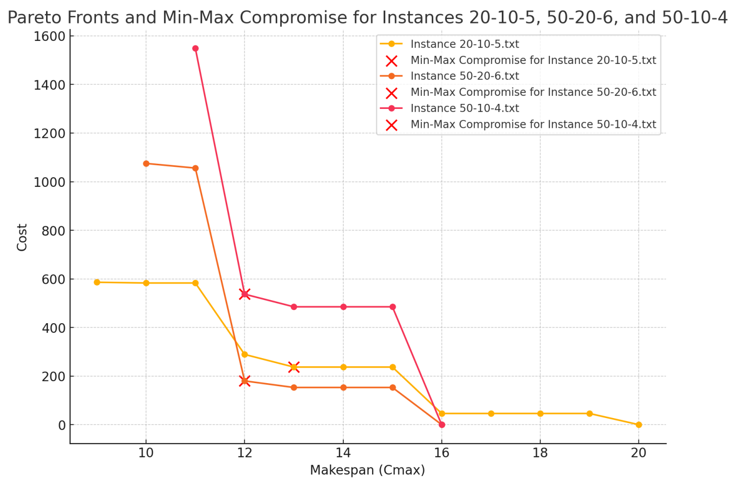

FIGURE 4.10 Pareto front for instances with the largest Euclidean distance between the best solutions for each objective

As the graphics show, the Min-Max method is effective when the decision-maker does not aim to prioritize one objective over another. In cases where such prioritization is needed, the weighted sum method is more appropriate, allowing the decision-maker to assign weights reflecting their preferences for each objective. Furthermore, we observed a significant increase in cost as the duration was reduced. Consequently, further reducing the duration by increasing external capacity is not advisable in our case.

Aiming to evaluate the best trade-offs obtained using the Min-Max method between minimizing completion time and minimizing costs, Table 4.9 presents the average deviation of completion times (Cmax) of the best compromise relative to the minimum (Cmax deviation %), as well as the average deviation of costs relative to the worst-case scenario (W-cost deviation %), for each group of instances. These deviations are expressed as percentages.

TABLE 4.9 Average deviation of completion Times (Cmax) and costs for each instance group

Nb-pack	**Nb-res**	**Cmax deviation %**	**W-cost deviation %**
10	3	46.66	70.53
10	10	48.33	71.12
10	20	48.33	69.16
20	3	40.72	67.84
20	10	33.05	78.40
20	20	31.44	81.62
50	3	26.60	85.12
50	10	18.94	81.83
50	20	30.89	73.00

We observe that, generally, as the number of Work Packages (WPs) increases, the project duration is less affected relative to the ideal Cmax, suggesting that larger problem sizes manage to minimize the time sacrifice while optimizing costs more effectively. Additionally, with 10 resources, the cost of external resources becomes less significant relative to the worst cost for instances with a greater number of WPs.

4.7 Conclusion and future work

We initially emphasize that the same planning outcomes can be represented through various combinations, caused by the interplay between continuous and discrete variables. This interaction often leads to situations where a binary variable may select between two adjacent periods, potentially causing extensive tailing and increased branching within the solution process. To address this, we introduce a nominal artificial boundary. This boundary aims to mitigate the risk of work package completion dates aligning too closely after the start of a period. Additionally, we impose a constraint on the binary variable, requiring it to assume a specific value when the end of a work package falls precisely between two successive periods. To empirically validate that our model does not affect the attainment of the optimal solution, we conducted comparisons with the optimal solutions obtained from the initial model, one of the most performant models in the literature. These comparisons were made possible either by allocating extended computation time (as in the case of Cherkaoui's instances) or by leveraging observations on the solver's behavior to suggest alternative strategies for achieving these optimal solutions. It reveals that our model is highly promising, significantly enhancing the solver's performance. To demonstrate the applicability of our model in real-world settings, we implemented the ε-constraint method to handle the trade-off between the time-driven and resource-driven objectives, a task that was very challenging with the initial model.With the optimization processes being highly efficient and leveraging the discrete nature of project duration, we successfully obtained all non-dominated solutions for the tested instances.

Our model could be further improved by re-formulating the constraints that link the continuous and binary variables, drawing inspiration from the works of Morin et al., specifically in tackling the periodically aggregated RCPSP problem, proposed in 2017 [74], and elaborated upon in subsequent study [75]. The interplay between continuous variables and binary variables currently introduces a large number of constraints and incorporates big-M constraints, which can lead to numerical difficulties for the MIP solver and result in weaker relaxations. In our future research, exploring alternative formulations to mitigate these issues will be a priority. Additionally, it would be intriguing to appropriately apply our approach to mixed models that have been reformulated to include continuous time variables interacting with binary variables. Given the increasing prominence of such models in the literature and their practical applicability, often surpassing that of models with discrete time variables, we aim to enhance continuous time models [76,78]. By incorporating our model, we seek to better meet real-world demands and optimize their practical utility.

Acknowledgments : This research was supported by Mitacs, and we gratefully acknowledge their financial support.

Disclosure statement : No potential conflict of interest was reported by the authors.

Data availability statement : The data that support the findings of this study are available from the corresponding author, A.Noureddine, upon reasonable request.

CHAPITRE 5 ARTICLE 2: A NEW TIME-DRIVEN MIXED-INTEGER PROGRAMMING MODEL FOR ROUGH-CUT CAPACITY PLANNING

Anis Noureddine[1,2,3], François Soumis[1,2], Robert Pellerin[1,3]

[1]Département de mathématiques et de génie industriel, Polytechnique Montréal
[2]GERAD, Montréal, Canada
[3]CIRRELT, Montréal, Canada

Computers & Operations Research, submitted May 26, 2025; revised November 22, 2025.

Abstract

This study investigates the Rough Cut Capacity Planning (RCCP) problem within the context of tactical project planning, aiming to determine the optimal execution intensities of work packages across defined planning periods. It further analyzes the impact of the chosen formulation on solving performance and to highlight the advantages and limitations of each modeling type. We propose several new formulations based on pulse and step variables, and compare them with the best models available in the literature. A comprehensive evaluation of the time-driven variant is undertaken, supported by extensive experimental analysis across a diverse set of instances. Our findings show that adopting a continuous-time modeling, by integrating additional continuous variables, has a limited effect on execution time while offering improved solution quality compared to discrete-time models. In comparison with the best continuous-time formulation from the literature, which failed to solve 83 out of 1200 instances, our best proposed formulation reduced the number of unsolved instances to only 10, achieved an average execution time seven times faster, and yielded cost improvements of up to 67% in some instances.

keywords Continuous-time model; Mixed-integer programming; Tactical project planning; Performance improvement.

5.1 Introduction

Tactical project planning serves as the intermediary stage between strategic and operational planning, focusing on decisions that align long-term objectives with the immediate allocation of limited resources [56, 58]. This stage involves creating aggregated schedules that account for workload and resource capacities, ensuring that project feasibility is evaluated early in the planning process [32, 39, 86, 95].

To address tactical project planning challenges, several authors have proposed integrated models that combine planning and scheduling decisions. The objectives addressed vary across models, with some targeting cost minimization, others focusing on setup times or project duration, or a combination of these. In addition, these models account for uncertainty in processing times [41, 54], multi-site coordination [57], and flexible resource capacities, including subcontracting and overtime [26, 113], and have been applied in different project sectors, including Engineer-to-Order (ETO) contexts [46, 81, 82].

Among tactical project planning models, Rough Cut Capacity Planning (RCCP) is a well-established approach. Its aggregated nature allows planners to evaluate overall resource availability while leaving detailed scheduling to later stages. It is closely related to the Resource-Constrained Project Scheduling Problem (RCPSP). In practice, RCCP serves as a foundation for RCPSP by verifying whether global resource capacities can meet project demand before activity-level scheduling begins [6, 13, 47, 95, 99, 115].

Despite their connection, RCCP and RCPSP models differ in several fundamental aspects. In RCCP, activities are not yet defined, so planning is based on work packages (WPs), which represent groups of future tasks. Precedence constraints are set between WPs, without detailing the internal activity structure. In RCCP, work intensity and durations are variable, while RCPSP assumes fixed durations and known resource intensities. RCCP typically involves fixed total workloads expressed per unit of time-person, contrasting with RCPSP, where the number of resources used per unit time is fixed. While RCCP uses a discretization of aggregated periods, RCPSP operates on detailed time intervals. Finally, RCCP may follow a *resource-driven* variant, aiming to minimize project duration under resource constraints, or a *time-driven* variant, minimizing external costs while meeting a fixed deadline. In contrast, RCPSP typically aims to minimize the project makespan.

Existing RCCP models face limitations in resolution efficiency [78, 79, 83], particularly when dealing with large-scale or resource-intensive problems. The improvements in resolution capacity allow for resolving large-scale instances or the integration of additional constraints, thereby increasing the model's flexibility and adaptability. For instance, constraints such as overlapping WPs can be incorporated, as demonstrated in studies like [12, 15, 17, 59], further enriching the model's applicability in diverse project environments.

Building on this context, the main objective of this study is to identify the most effective RCCP formulation for tactical planning. To achieve this, we propose several new mathematical models. We also include in our analysis the most efficient formulations found in the literature. Each model is tested and compared on a set of problem instances. This comparative analysis provides insights into performance differences and supports the selection of the most suitable formulation by identifying the best trade-off between flexibility, solution quality, and computational efficiency. We take advantage of recent improvements in commercial solvers to solve more complex and realistic problems.

We begin by evaluating discrete-time formulations, which are generally more efficient in terms of computational time. In these models, start and end dates are represented by the number of periods, meaning that WPs can only begin or end at the boundaries of time intervals. However, even when periods are short, solution quality can be affected, and the issue becomes more pronounced as periods are aggregated, due to limited temporal flexibility.

To overcome this limitation, we explore continuous-time models, in which start and end dates can be positioned anywhere within a given period, offering greater flexibility and the potential for improved solution quality. Nevertheless, continuous-time formulations can only yield improved outcomes when the structure of variables and constraints is aligned with the solver's resolution mechanisms. In some cases, formulations that are theoretically sound may still result in slower convergence due to solver-specific behavior [78]. To fully exploit the potential of continuous-time modeling, we propose and test several formulations that integrate components inspired by discrete-time models (e.g., shared variable blocks and constraint structures). This modular approach allows us to maintain the strengths of discrete formulations while improving solution quality and limiting computational burden.

The remainder of the paper is structured as follows : Section 5.2 provides a literature review of the most significant works on RCCP. Section 5.3 presents the formulations tested and details the improvements introduced to accelerate the model's resolution. Section 5.4 outlines the experiments conducted and analyzes the results. Finally, we conclude in Section 5.5 by summarizing the findings and discussing future perspectives.

5.2 Related works

This section provides an overview of relevant research related to RCCP. First, we examine RCPSP variants that share some characteristics with RCCP, such as flexibility in workload intensity, resource profiles, or the use of aggregated planning periods. Next, we discuss mathematical programming methods used in tactical planning. Finally, we review RCCP formulations, identifying their limitations and highlighting recent advancements aimed at improving computational efficiency and scalability.

5.2.1 Pertinent RCPSP variants for RCCP

To better align with practical applications, several extensions of the RCPSP have been developed. [50] introduces a variant of the RCPSP that allows resource demands and capacities to vary depending on the time period. This approach was proved to be effective in the context of a medical research project. [114] further generalize this concept, leading to the development of MRCPSP (Multi-Mode RCPSP), which incorporates multiple execution modes for activities.

Another extension is the periodically aggregated RCPSP (PARCPSP), proposed by [74], which considers the average usage of resources over aggregated periods of equal length. While resource consumption is averaged, temporal aspects, such as the duration of activity, are modeled with precision. This extension acts as an intermediary between the tactical and operational planning levels. The authors introduced a continuous-time MIP formulation along with a hybrid method to improve computational efficiency. Building on this work, [76] propose a new MIP formulation for PARCPSP. They also analyzed the problem's complexity and showed that it is at least as difficult as the classical RCPSP. The window aggregated RCPSP (WARCPSP), proposed by [71], generalizes the PARCPSP by allowing more flexible window configurations. To solve it, a heuristic conflict resolution algorithm was proposed to minimize the makespan by adjusting plans across different levels of time aggregation.

The RCPSP with variable intensity activities (RCPSVP) was studied by [60], who proposed a MIP formulation reinforced with valid inequalities to improve the model. This formulation addresses the resource-driven variant of the problem. Later, [59] extended this model by tackling the RCPSVP with generalized precedence constraints and by minimizing the penalty cost of violating the resource capacity constraints. [17] introduce a forward recursion approach to determine the earliest start and finish times under generalized precedence constraints, neglecting resource constraints. Unlike the classical RCPSP, finding the earliest start and end dates is not straightforward, as it does not rely simply on finding the longest path. [51] propose an original RCPSVP formulation that combines continuous-time for start and end dates with a discrete evaluation of resources. Lastly, [103] introduce a new optimization problem by considering the RCPSVP with execution intensities defined separately for each resource, unlike classical RCPSVP formulations, which assume a shared intensity across all resources for a given activity. Their model minimizes external resource usage costs to tackle resource leveling problem. For solving this generalized problem, they propose a Benders decomposition approach.

The RCPSP with flexible resource profiles (FRCPSP) has been addressed by [79], who propose and compare four discrete-time mixed-integer programming (MIP) formulations. They conclude that the model based on [60] and [14] performs better in terms of execution time and solution quality. This work has inspired several authors to study and solve practical variants of the problem. For instance, [105] propose a matheuristic approach and compare it with an MIP model inspired by [79], while [24] address a multi-project FRCPSP by adapting the model of [79].

5.2.2 RCCP formulations for addressing tactical planning

[39] introduces a discrete-time, non-linear RCCP model with only non-linear precedence constraints. To solve this model, the precedence constraints are relaxed and incorporated heuristically. [42], [111], and [47] propose a discrete-time MIP RCCP formulation for time-driven contexts. However, these models face two main limitations. First, even with long time periods, the variables still depend on the time horizon. Otherwise, the limited set of possible start and end dates could lead to infeasible schedules or unnecessarily extended project durations. As a result, solving instances with a large number of WPs and a long planning horizon remains computationally intensive. To address this, decomposition methods are often used to reduce the size of subproblems, such as the column generation approach proposed by [47]. Second, precedence relations between WPs $(i, j) \in E$ may prevent both from being scheduled within the same time period without causing overlaps. To mitigate this, some authors [47] choose to limit such cases explicitly, while others introduce overly restrictive constraints to avoid feasibility issues [60].

Building on the RCPSVP framework introduced by [51], [12] propose an RCCP formulation that incorporates feasible overlapping modes between WPs. Their findings indicate that, while overlapping can accelerate project delivery times, it also helps reduce project costs by allowing for a more balanced workload distribution. Similarly, by building on the RCPSVP framework of [51], [32] introduce a mixed continuous-time RCCP formulation tailored to large-scale construction projects by incorporating variable period lengths. This continuous-time formulation is particularly well suited when time periods are long, as it allows for more accurate modeling of precedence constraints and provides greater flexibility in positioning WP start and end times. In such contexts, it often outperforms discrete-time models in terms of solution quality. However, when time periods are short or not aggregated, continuous-time models may become harder to solve than discrete ones due to increased modeling and computational complexity [78]. [83] propose a reformulated RCCP model, based on the work of [32]. By analyzing solver behavior, they restructured constraints linking end dates and discrete periods, leading to improved model performance. In the time-driven variant, Cherkaoui's instances [32] with 50 WPs show significantly improved computation times under both short and long periods.

Given that the primary objective of this study is to identify the most suitable RCCP model among those proposed in the literature, we adopt the classical RCCP formulations. Enhancing model flexibility, for instance by allowing overlapping WPs or resource specific intensities, represents a possible extension but falls beyond the scope of this review, as our purpose is to establish a clear baseline for comparison.

Table 5.1 provides a comparative overview of RCCP MIP models from the literature that can be solved directly by an MIP solver. We selected models that adhere to the same assumptions as classical RCCP, namely finish to start precedence relations and variable execution across periods with fixed proportional resource usage. Although some of these models are formally classified as RCPSP variants, they were included here because they share similar assumptions and modeling frameworks with RCCP.

In the table, *Aggr.* denotes aggregated precedence relations, where each precedence constraint is represented by a single inequality formulated in a compact way by summing the binary decision variables over all time periods. By contrast, *Disagg.* denotes disaggregated precedence relations, in which each precedence relation is decomposed into a set of inequalities defined for every time period.

In terms of time representation, *Aggr.* (aggregated) refers to a division of the planning horizon into uniform time units of equal length. *Aggr-var* (aggregated-variable) also partitions the horizon into broader periods, but the lengths of these periods may vary. In contrast, *Disagg.* (disaggregated) splits the horizon into elementary time units and imposes constraints for each individual period.

Finally, the table distinguishes different types of decision variables. *Pulse variables* are binary variables that take the value 1 only at the exact period when an event occurs (e.g., the start of a WP). *Step variables* are binary variables that indicate whether an event has already occurred by the current period. *On/off variables* are binary variables that indicate whether the WP is in process during the current period. *Event-based variables* are binary variables that are not dependent on periods but on the number of events that occur (start of WPs, end of WPs, or changes in resource intensity).

Table 5.1 Comparative overview of RCCP models (✓= selected category)

Characteristic	**Category**	**[60]**	**[51]**	**[32]**	**DT1 [79]**	**DT2 [79]**	**DT3 [79]**	**DT4 [79]**	**[78]**	**[83]**
Variant	RCPSVP	✓	✓							
	FRCPSP				✓	✓	✓	✓	✓	
	RCCP			✓						✓
Model type	Continuous		✓	✓					✓	✓
	Discrete	✓			✓	✓	✓	✓		
Variable nature	Step	✓	✓	✓			✓			✓
	Pulse					✓				
	On/off				✓			✓		
	Event-based								✓	
[1]**Precedence**	Aggr.		✓	✓				✓		✓
	Disagg.	✓			✓	✓	✓		✓	
(formulation) **Periods**	Aggr.		✓							
	Disagg.	✓			✓	✓	✓	✓	✓	
	Aggr-var.			✓						✓
Objective	Resource-driven	✓		✓	✓	✓	✓	✓	✓	✓
	Time-driven		✓							✓
	Bi-objective									✓
Accelerating the resolution	Valid inequalities/optimality-based cuts	✓							✓	
	Symmetry breaking									✓

5.2.3 Critical analysis

Although various authors propose practical extensions that often increase the complexity of classical models, these advanced formulations generally retain the foundational structure of their base counterparts. As such, the performance and scalability of advanced, complex models are inherently tied to the efficiency of the base model they are derived from. In other words, the stronger and more efficient the foundational model, the better the performance of its extended, more complex versions.

Building on this observation, the present study adopts a deterministic framework, where workload and resource data are considered fixed. Given the importance of cost estimation in tactical planning, the study focuses on developing and analyzing several classical time-driven RCCP models. The objective is to improve their structure and identify the most efficient formulation in terms of computational performance, thereby providing decision-makers for selecting the most appropriate planning approach.

5.3 Formulations of the RCCP model

In this section, we first present the underlying assumptions of the RCCP problem and provide an overview of the discrete-time formulations tested, inspired by models proposed in the RCPSP literature [7,14,60,61]. We then describe a continuous-time MIP formulation originally proposed by [51] and improved by [83], which provides a continuous-time representation. Building upon the work of [76] on the PARCPSP problem, and the discrete-time formulations discussed earlier, we propose reformulated models. These new versions are enhanced with additional constraints designed to accelerate the solving process.

5.3.1 Input and notations

In the RCCP problem, a planning horizon is divided into a set of time periods, where each period p has a duration. These periods may have non-uniform lengths, reflecting the flexibility of RCCP in aggregating planning intervals. The model considers a set of WPs, where each WP is characterized by several parameters. It has a maximum and a minimum workload per unit of time, which define the upper and lower bounds on the effort that can be allocated in any given period. In addition, each WP is associated with a total required workload, expressed in man-hours or man-days, representing the cumulative amount of work needed across all resources to complete it. Each WP also has minimum and maximum execution periods. The RCCP model also relies on a set of renewable resources, each with *internal capacity*, which corresponds to the amount of work that can be performed using the organization's own resources, such as internal staff, in a given period, and *external capacity*, which represents the additional work that can be performed by acquiring resources from outside the organization, such as subcontractors or rented equipment, in that period. Furthermore, precedence constraints between WPs are incorporated to ensure that dependent WPs are executed in the correct sequence.

The notations for the input parameters and decision variables used in solving this problem are presented in Table 5.2. The sets Ps_i and Pf_i denote, respectively, the set of periods in which WP i can start (either strictly within a period or at its beginning) and the set of periods in which it can finish (either strictly within a period or at its end).

Let $\pi(t)$ denote the period to which time t belongs. In order to eliminate unnecessary periods, we assume that $t \in \pi(t)$ if it lies strictly within period p or exactly at its end. This can be mathematically expressed as follows :

$$\pi(t) = \min \left\{ p \;\middle|\; \sum_{k=1}^{p} D_k \;\geq\; t \right\}, \quad t \in H$$

TABLE 5.2 Notations used in the RCCP problem

Notations of the Input Parameters	
I	Set of WPs.
R	Set of resource groups.
H	Discretized planning horizon (time periods).
P	Set of periods.
D_p	Duration of period p (weeks).
Q_{ri}	Load required for WP i by resource group r (man-hours).
$Q_i^{\max}, Q_i^{\min}$	Maximum and minimum workload allowed for WP i in each unit of time.
L_{rp}	Internal capacity of resource group r in period p.
M_{rp}	External capacity of resource group r in period p.
c_{rp}	Cost per unit of external capacity used for resource group r in period p.
E	Set of precedence constraints : $(i,j) \in E$ if WP i precedes j.
ES_i, EF_i	Earliest start and finish dates of WP i.
LS_i, LF_i	Latest start and finish dates of WP i.
$\pi(t)$	The period to which time t belongs (when t lies strictly within period p or exactly at its end).
Ps_i, Pf_i	Sets of periods in which WP i can start and finish, respectively.
Decision Variables	
x_{ip}	Work intensity of WP i at period p ($0 \leq x_{ip} \leq 1$).
ts_i, tf_i	Start and end dates of WP i.
d_{ip}	Duration of WP i in period p.
z_{ip}	1 if WP i starts at the beginning or within period p (in discrete-time : only at the beginning), 0 otherwise
y_{ip}	1 if WP i ends within period p or at its end (in discrete-time : only at the end), 0 otherwise
zs_{ip}	1 if WP i starts at or before the beginning of period p.
ys_{ip}	1 if WP i ends at or before the end of period p.
v_{rp}, w_{rp}	Internal/external resource r usage in period p.
C	Total cost of using external resources.

A period p belongs to Ps_i if $p \in \{\pi(ES_i+1),\ldots,\pi(LS_i+1)\}$, and it belongs to Pf_i if $p \in \{\pi(EF_i),\ldots,\pi(LF_i)\}$.

To deduce the earliest and latest start and finish dates, we consider $\frac{\sum_{r\in R} Q_{ri}}{Q_i^{\max}}$ as the minimum duration of WP i. Similarly, $\frac{\sum_{r\in R} Q_{ri}}{Q_i^{\min}}$ is taken as the maximum feasible duration of WP i. Our adaptation for reducing the number of variables and constraints in RCCP, using earliest and latest dates in aggregated periods, is novel : unlike in the disaggregated case, these dates may be positioned either inside the periods or at their boundaries.

In all formulations presented below, the objective function (1) minimizes the cost of using external resources, calculated as the weighted sum of external resource usage w_{rp} over all resources and periods, defined as :

$$\text{Minimize } C = \sum_{r\in R}\sum_{p\in P} c_{rp} \times w_{rp} \tag{1}$$

To ensure consistency across all RCCP formulations, constraints are labeled using a standardized notation of the form (number + type). The number uniquely identifies the constraint, while the suffix indicates the formulation type : for discrete-time models, (p) denotes constraints involving pulse variables and (d) denotes those involving step variables ; for continuous-time models, (c) refers to the base formulation, (s) to constraints involving pulse variables, and (i) to those involving step variables. Constraints that are common to all models are identified by uppercase letters, while improved or alternative versions are marked with a prime symbol at the end.

5.3.2 Discrete-time RCCP formulations

We begin by presenting the discrete-time RCCP model. These formulations can be practical when the length of periods is relatively small, as explained in Section 5.2.

5.3.2.1 Pulse Discrete-time formulation : *P-RCCP*

The proposed pulse-based RCPSP formulations employ pulse decision variables defined as follows : $z_{ip} = 1$ if and only if WP i starts at the beginning of period p, and $z_{ip} = 0$ otherwise ; similarly, $y_{ip} = 1$ if and only if WP i ends at the end of period p, and $y_{ip} = 0$ otherwise [7, 61].

$$\sum_{p \in Ps_i} z_{ip} = 1, \qquad \forall i \in I \tag{1p}$$

$$\sum_{p \in Pf_i} y_{ip} = 1, \qquad \forall i \in I \tag{2p}$$

$$d_{ip} = D_p \times \left(\sum_{k \in Ps_i \cap \{1,\ldots,p\}} z_{ik} - \sum_{k \in Pf_i \cap \{1,\ldots,p\}} y_{ik} \right), \qquad \forall i \in I,\, p \in P \tag{3p}$$

$$d_{ip} = 0, \qquad \forall i \in I,\, p \in Ps_i^c \cap Pf_i^c \tag{4p}$$

$$\sum_{p \in Ps_i^c} z_{ip} = 0, \qquad \forall i \in I \tag{5p}$$

$$\sum_{p \in Pf_i^c} y_{ip} = 0, \qquad \forall i \in I \tag{6p}$$

$$z_{jp} \leq \sum_{k \in Pf_i \cap \{1,\ldots,p\}} y_{ik}, \qquad \forall (i,j) \in E,\, p \in P \tag{7p}$$

$$x_{ip} \cdot \sum_{r \in R} Q_{ri} \leq Q_i^{\max} \cdot d_{ip}, \qquad \forall i \in I,\, p \in P \tag{A}$$

$$x_{ip} \cdot \sum_{r \in R} Q_{ri} \geq Q_i^{\min} \cdot d_{ip}, \qquad \forall i \in I,\, p \in P \tag{B}$$

$$\sum_{p \in P} x_{ip} = 1, \qquad \forall i \in I \tag{C}$$

$$v_{rp} + w_{rp} = \sum_{i \in I} x_{ip} \cdot Q_{ri}, \qquad \forall r \in R,\, p \in P \tag{D}$$

$$v_{rp} \leq L_{rp}, \qquad \forall r \in R,\, p \in P \tag{E}$$

$$w_{rp} \leq M_{rp}, \qquad \forall r \in R,\, p \in P \tag{F}$$

Constraints (1p) and (5p) ensure that each WP starts within the interval defined by its earliest start and latest start times. Similarly, constraints (2p) and (6p) enforce that each WP finishes within its respective earliest finish and latest finish bounds. The duration of a WP i in period p, denoted d_{ip}, is determined by its start and finish times, as modeled by constraints (3p) and (4p). Constraints (7p) maintain the precedence relationships between WPs. It represents a disaggregated version, as the precedence constraints are enforced period by period. These constraints have been proposed in the context of the FRCPSP problem [79].

Constraints (A) to (F) regulate resource usage and capacities. Constraints (A) and (B) ensure that the workload assigned to each WP respects the defined minimum and maximum bounds for each period. Constraints (C) guarantee that the total workload of a WP is fully assigned across all periods. The allocation of resources between internal (v_{rp}) and external (w_{rp}) resources is managed by constraints (D), while constraints (E) and (F) ensure that the usage of these resources does not exceed their respective capacities.

5.3.2.2 Pulse discrete-time formulation using disaggregated precedence relation : *PD-RCCP*

To strengthen the formulation, we introduce an alternative version of the disaggregated precedence relation, given by constraints (7p'). This type of constraint was originally proposed in the context of the RCPSP by [33]. We adapt it here to fit the discrete-time formulation of the RCCP as follows :

$$\sum_{k \in Ps_j \cap \{1,\dots,p\}} z_{jk} - \sum_{k \in Pf_i \cap \{1,\dots,p\}} y_{i,k} \leq 0, \qquad \forall (i,j) \in E, \forall p \in P \tag{7p'}$$

Proposition 4. *The RCCP formulation obtained by replacing the precedence constraints (7p) with (7p') is stronger.*

Proof. Let us denote by S_{7p} and $S_{7p'}$ the relaxed feasible domains of the P-$RCCP$ and PD-$RCCP$ formulations, respectively.

It can be shown that the disaggregated constraints (7p'), in combination with constraints (5p), imply the aggregated constraints (7p), combined with constraints (5p). Two distinct cases can be considered :

1. If $p \in Ps_j$, starting from constraints (7p'), we obtain :

$$\sum_{k \in Ps_j \cap \{1,\dots,p\}} z_{jk} \leq \sum_{k \in Pf_i \cap \{1,\dots,p\}} y_{ik}.$$

   From this, the following inequality can be derived :

$$z_{jp} \leq \sum_{k \in Pf_i \cap \{1,\dots,p\}} y_{ik} - \sum_{k \in Ps_j \cap \{1,\dots,p-1\}} z_{jk} \Rightarrow z_{jp} \leq \sum_{k \in Pf_i \cap \{1,\dots,p\}} y_{ik},$$

   which corresponds to the constraints (7p).
2. If $p \notin Ps_j$, then $z_{jp} = 0$ (from constraints (5p)). Constraints (7p) are trivially satisfied. Hence, $S_{7p'} \subseteq S_{7p}$.

Let us now show that the converse is not always true by providing a counterexample. We will exhibit a solution $X' \in S_{7p}$ such that $X' \notin S_{7p'}$ (Table 6.3). Let us consider an instance I with 2 WPs and 3 periods of length (3, 2, 1), respectively. The minimal durations of the two WPs are 1. Let us consider $Q_1^{\min} = Q_2^{\min} = 0$, meaning that there is no upper bound on the duration. There is a precedence relation such that $(1, 2) \in E$. In this case, we have $Ps_2 = \{1, 2, 3\}$ and $Pf_1 = \{1, 2, 3\}$. Consider a single resource with $Q_{11} = Q_{12} = 1$, $Q_1^{\max} = Q_2^{\max} = 1$. The available internal and external capacities in each period are both equal to 1.

TABLE 5.3 Example of a relaxed solution that is feasible for *P-RCCP* but infeasible for *PD-RCCP*

WP_i	$z_{i,1}$	$z_{i,2}$	$z_{i,3}$	$y_{i,1}$	$y_{i,2}$	$y_{i,3}$	$x_{i,1}$	$x_{i,2}$	$x_{i,3}$
1	1	0	0	$\mathbf{\frac{1}{4}}$	$\mathbf{\frac{2}{4}}$	$\mathbf{\frac{1}{4}}$	1	0	0
2	$\mathbf{\frac{1}{4}}$	$\mathbf{\frac{3}{4}}$	**0**	0	0	1	$\frac{1}{2}$	$\frac{1}{2}$	0

This solution satisfies the constraints of the *P-RCCP* formulation, but violates the disaggregated constraints of the *PD-RCCP* (for $p = 2 : \frac{1}{4} + \frac{3}{4} > \frac{1}{4} + \frac{2}{4}$), thus proving that : $S_{7p'} \subset S_{7p}$.

□

Observation 1 :

Given that the duration of WPs is variable, when $Q_i^{\min} = 0$, which means that the execution intensity has no lower bound, we can also express constraints (3p) using the following inequality :

$$D_p \times \left(\sum_{k \in Ps_i \cap \{1,\ldots,p\}} z_{ik} - \sum_{k \in Pf_i \cap \{1,\ldots,p\}} y_{ik} \right) \geq d_{ip}, \qquad \forall i \in I, \forall p \in P \qquad \text{(3p')}$$

Solving linear programs (LPs) with inequalities is generally easier than LPs with equalities.

Proposition 5. *In the RCCP problem, if $Q_i^{\min} = 0$, the equality constraints (3p) can be relaxed into the inequality (3p') while still maintaining the validity of the RCCP formulation.*

The proof of Proposition 5 is provided in 5.6.1.

5.3.2.3 Step discrete-time formulation : *ST-RCCP*

The proposed formulation builds upon step-based RCPSP models, introducing step decision variables zs_{ip} and ys_{ip}, which indicate whether WP i has already started before or at the beginning of aggregated period p, and whether it has already finished before or at the end of aggregated period p, respectively [60, 61].

$$tf_i = H - \sum_{p \in \{\pi(EF_i),\ldots,|P|\}} D_p \times ys_{ip}, \qquad \forall i \in I, \tag{1d}$$

$$ts_i = H - \sum_{p \in \{\pi(ES_i+1),\ldots,|P|\}} D_p \times zs_{ip}, \qquad \forall i \in I, \tag{2d}$$

$$zs_{ip} \geq zs_{ip-1}, \qquad \forall i \in I,\, p \in P, \tag{3d}$$

$$ys_{ip} \geq ys_{ip-1}, \qquad \forall i \in I,\, p \in P, \tag{4d}$$

$$d_{ip} = D_p \times (zs_{ip} - ys_{ip}), \qquad \forall i \in I,\, p \in P, \tag{5d}$$

$$ts_j \geq tf_i, \qquad \forall (i,j) \in E, \tag{6d}$$

$$zs_{i\pi(ES_i+1)-1} = 0, \qquad \forall i \in I \tag{7d}$$

$$zs_{i\pi(LS_i+1)} = 1, \qquad \forall i \in I, \tag{8d}$$

$$ys_{i\pi(EF_i)-1} = 0, \qquad \forall i \in I, \tag{9d}$$

$$ys_{i\pi(LF_i)} = 1, \qquad \forall i \in I, \tag{10d}$$

Constraints (4p), (A), (B), (C), (D), (E), and (F)

Constraints (1d) and (2d) respectively compute the start times ts_i and finish times tf_i of each WP $i \in I$, based on a time horizon H and the binary variables zs_{ip} and ys_{ip}, which indicate the active periods for the start and end of the WPs. Additionally, constraints (3d), (4d), (7d), (8d), (9d), and (10d) enforce that once a period p is activated for a WP, all subsequent periods remain active. The total duration of a WP is tied to its start and finish times through constraints (5d), which ensures that $\sum_{p \in P} d_{ip} = tf_i - ts_i$. Constraints (6d) preserve priority relationships between WPs, ensuring that the start time of a WP respects the finish times of its predecessors. These constraints are defined as aggregated constraints since each precedence relation is represented by a single constraint.

5.3.2.4 Step Discrete-time formulation with disaggregated precedence relation : *SD-RCCP*

In the context of the RCPSP, [33] demonstrated that disaggregated constraints outperform aggregated ones. This result was later confirmed in the PARCPSP framework by [76]. This work shows that the same conclusion holds in the RCCP context, where the adapted disaggregated constraints lead to a stronger model formulation. The new constraints are as follows :

$$zs_{jp} - ys_{ip} \leq 0, \qquad \forall (i,j) \in E,\, p \in P \tag{6d'}$$

We refer to it as disaggregated because each precedence relation is represented not by a single constraint, but by a set of constraints, one for each period. In our case, since the duration of the WPs is variable, it is necessary to introduce two types of variables : those associated with the start dates and those used to determine the finish dates.

Proposition 6. *Replacing the aggregated precedence constraints (6d) with the disaggregated precedence constraints (6d') in the ST-RCCP formulation results in a stronger RCCP formulation.*

The complete proof is provided in 5.6.2.

5.3.3 Continuous-time RCCP Models

Building on the discrete-time formulations presented in Section 5.3.2, we now introduce several continuous-time formulations. Among the best continuous models proposed in the literature, the one developed by [83] is of particular interest, as it improves upon the model of [51] by reducing the search space through the elimination of redundant configurations. In our work, we further reduce the number of variables by bounding the start and end dates using the earliest and latest possible times.

5.3.3.1 Continuous step variables enumerating the possible configurations of period positions : $\boldsymbol{B\text{-}RCCP_c}$

We present the RCCP model as follows :

$$ts_i \geq \sum_{k=1}^{p} D_k \times (1 - zs_{ip}), \qquad \forall i \in I,\, p \in P, \tag{1c}$$

$$ts_i \leq \sum_{k=1}^{p} D_k + (H - \sum_{k=1}^{p} D_k) \times (1 - zs_{ip}), \qquad \forall i \in I,\, p \in P, \tag{2c}$$

$$zs_{ip} \geq zs_{ip-1}, \qquad \forall i \in I,\, p \in P, \tag{3c}$$

$$tf_i \geq \sum_{k=1}^{p} D_k \times (1 - ys_{ip}), \qquad \forall i \in I,\, p \in P, \tag{4c}$$

$$tf_i \leq \sum_{k=1}^{p} D_k + (H - \sum_{k=1}^{p} D_k) \times (1 - ys_{ip}) + \epsilon \times (1 - ys_{ip}), \qquad \forall i \in I,\, p \in P, \tag{5c}$$

$$ys_{ip} \geq ys_{ip-1}, \qquad \forall i \in I,\, p \in P, \tag{6c}$$

$$d_{ip} \leq D_p \times (zs_{ip} - ys_{ip-1}), \qquad \forall i \in I,\, p \in P, \tag{7c}$$

$$d_{ip} \geq D_p \times (zs_{ip-1} - ys_{ip}), \qquad \forall i \in I,\, p \in P, \tag{8c}$$

$$d_{ip} \geq tf_i - \sum_{k=1}^{p} D_k + D_p \times zs_{ip-1} - H \times (1 - ys_{ip}), \qquad \forall i \in I,\, p \in P, \tag{9c}$$

$$d_{ip} \geq \sum_{k=1}^{p} D_k \times (1 - zs_{ip-1}) - ts_i - D_p \times ys_{ip}, \qquad \forall i \in I,\, p \in P, \tag{10c}$$

$$\sum_{p \in P} d_{ip} = tf_i - ts_i, \qquad \forall i \in I, \tag{11c}$$

$$ts_j \geq tf_i, \qquad \forall (i,j) \in E, \tag{12c}$$

Constraints (4p),(7d), (8d), (9d), (10d), (A), (B), (C), (D), (E), and (F)

Constraints (1c) and (2c) define the lower and upper bounds for the start time ts_i, while constraints (4c) and (5c) do the same for the end time tf_i. The binary variables zs_{ip} and ys_{ip} ensure consistency by activating the periods during which WPs are scheduled, and constraints (3c) and (6c) ensure that once a period p is active, all subsequent periods also remain active.

Constraints (7c) to (11c) determine the duration d_{ip} of WPs in each period. Constraints (7c) ensure that the duration is zero if the WP is inactive in period p, while constraints (8c) set the duration to D_p if the WP spans the full period. Constraints (9c) and (10c) handle intermediate cases where the start or end lies within the period. The total duration of each WP is linked to its start and end times through constraints (11c). Constraints (12c) enforce precedence relations between WPs.

5.3.3.2 New pulse-variable continuous-time RCCP model : ***P-RCCP$_c$***

We propose a new formulation for the RCCP problem by adapting the discrete-time pulse-based modeling technique used in the *PD-RCCP* model, along with the MIP formulation of [76] developed for the PARCPSP problem.

New variables :
Two additional types of continuous variables are introduced in this model :

$$f_{ip} = \begin{cases} D_p, & \text{if period } p \text{ is after the period in which WP } i \text{ ends,} \\ \sum_{k\in\{1,\ldots,p\}} D_k - tf_i, & \text{if WP } i \text{ ends in period } p, \\ 0, & \text{otherwise.} \end{cases}$$

$$s_{ip} = \begin{cases} D_p, & \text{if period } p \text{ is after the period in which WP } i \text{ starts,} \\ \sum_{k\in\{1,\ldots,p\}} D_k - ts_i, & \text{if WP } i \text{ starts in period } p, \\ 0, & \text{otherwise.} \end{cases}$$

Although only one type of variable is sufficient to formulate the model, the inclusion of both variables f_{ip} and s_{ip} improves model performance. Thus, we use both variables in our formulations.

A Gantt chart (Figure 5.1) is provided to illustrate the different configurations that depend on the scheduling structure of a WP. The WP may :

— Start in period p and finish after p (configuration (a)) ;

— Start before p and finish in p (configuration (b)) ;

— Start and finish entirely within p (configuration (c)) ; and

— Start before p and finish after p (configuration (d)).

The length of the red line intersecting period k represents f_{ik}, and its endpoint corresponds to the end time of the WP i. Similarly, the length of the blue line intersecting period k represents s_{ik}, with its starting point indicating the start time of the WP. The length of the green line intersecting period k corresponds to the duration of the WP within that period.

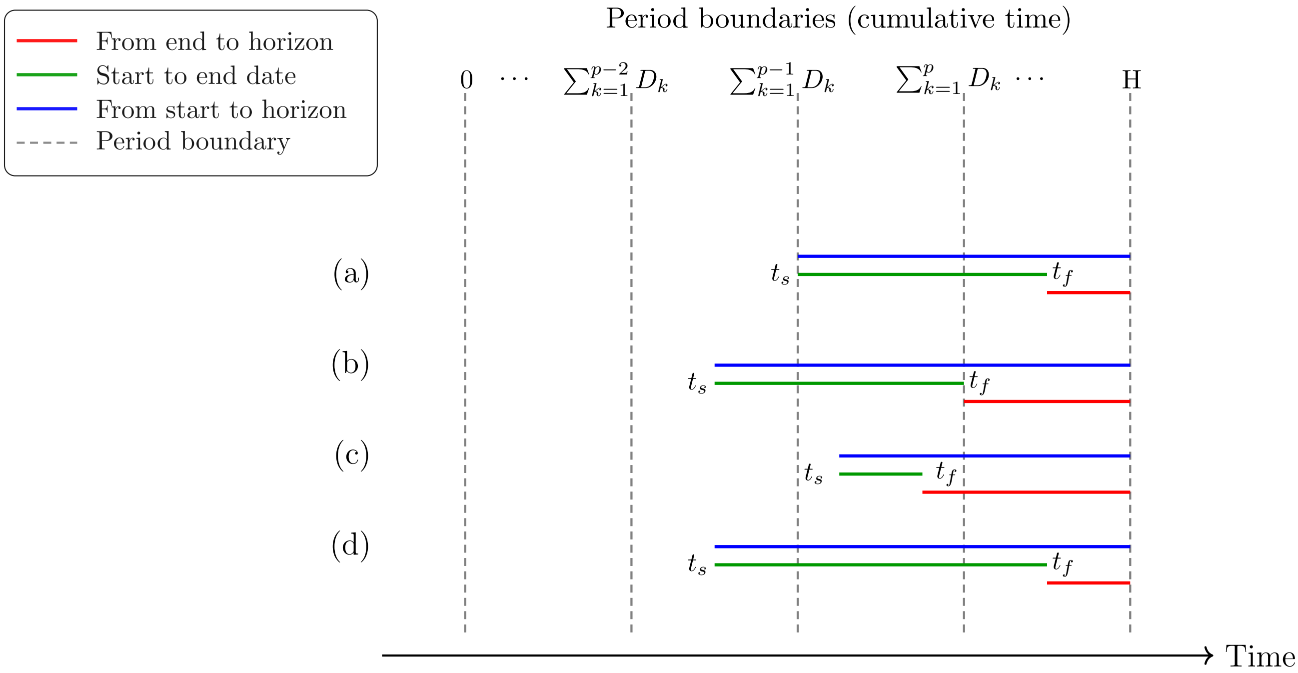

FIGURE 5.1 Illustration of different configurations of start (ts) and end (tf) times

$$d_{ip} = s_{ip} - f_{ip}, \qquad \forall i \in I,\ p \in P, \tag{1s}$$

$$s_{jp} - f_{ip} \leq 0, \qquad \forall (i,j) \in E,\ p \in P, \tag{2s'}$$

$$s_{ip} \leq D_p \times \sum_{k \in Ps_i \cap \{1,\ldots,p\}} z_{ik}, \qquad \forall i \in I,\ p \in P, \tag{3s}$$

$$s_{ip} \geq D_p \times \sum_{k \in Ps_i \cap \{1,\ldots,p-1\}} z_{ik}, \qquad \forall i \in I,\ p \in P, \tag{4s}$$

$$f_{ip} \leq D_p \times \sum_{k \in Pf_i \cap \{1,\ldots,p\}} y_{ik}, \qquad \forall i \in I,\ p \in P, \tag{5s}$$

$$f_{ip} \geq D_p \times \sum_{k \in Pf_i \cap \{1,\ldots,p-1\}} y_{ik}, \qquad \forall i \in I,\ p \in P, \tag{6s}$$

Constraints (1p), (2p), (4p), (5p), (6p), (A), (B), (C), (D), (E), and (F)

Constraints (1s) state that when $s_{ip} = 0$, then $d_{ip} = 0$. When $s_{ip} > 0$, the variable d_{ip} remains active until f_{ip} becomes active. The Gantt chart in Figure 5.1 illustrates an example of this behavior.

Constraints (2s') represent the continuous-time version of the disaggregated precedence constraints (7p'). Constraints (3s) to (6s) ensure that the continuous variables remain consistent with the binary variables. In this formulation, we rely on the results of Proposition 5, as it provides better performance.

An adaptation of the disaggregated constraints (7p) can be formulated as follows :

$$s_{jp} - s_{jp-1} \leq f_{ip}, \qquad \forall (i,j) \in E,\ p \in P, \tag{2s}$$

Using the disaggregated precedence constraints (2s') instead of the disaggregated precedence constraints (2s) in the continuous-time RCCP formulation results in a stronger formulation.

The proof of Theorem 5.3.3.2 is provided in 5.6.3.

The aggregated precedence constraints given in (12c) can be derived using the following identities :

$$ts_i = H - \sum_{p \in P} s_{ip}, \quad tf_i = H - \sum_{p \in P} f_{ip}.$$

Using the disaggregated precedence constraints (2s') in the RCCP formulation results in a stronger model than the one based on aggregated precedence constraints (12c).

The proof of Theorem 5.3.3.2 is provided in 5.6.4.

Observation 2 :

When using the initial formulations, [83] define two types of equivalent solutions :

1. **Direct Period Completion (DC)** : $y_{ip} = 1$, meaning that the WP i ends exactly at the end of period p.
2. **Overlapping Period Completion (OC)** : $y_{ip} = 0$ and $y_{ip+1} = 1$, where the completion is acknowledged between two periods.

They show that eliminating (OC) solutions improves the quality of the lower bound. In the initial formulation (as in configuration (b) of Figure 5.1), the binary variables can choose between (DC) and (OC) modes. These constraints can therefore be interpreted as symmetry breaking constraints.

Model reformulation based on Direct Period Completion (DC) :

To avoid resolution anomalies that can slow convergence, as shown by [83], we add the following constraints :

$$D_p + D_{p+1} - (f_{ip} + f_{ip+1}) \geq (D_p + \epsilon) \times (1 - \sum_{k \in Ps_i \cap \{1,\ldots,p\}} y_{ik}), \quad \forall i \in I, \forall p \in P \qquad (7s')$$

Proposition 7. *Constraints (7s') remove the (OC) solutions when $\epsilon > 0$ and quite small.*

The complete proof is provided in 5.6.5.

Observation 3 : With the substitution

$$s_{ip} = D_p \sum_{k \in Ps_i \cap \{1,\ldots,p\}} z_{ik}, \qquad f_{ip} = D_p \sum_{k \in Pf_i \cap \{1,\ldots,p\}} y_{ik}$$

It follows that the LP-relaxation feasible region of $P\text{-}RCCP_c$ contains that of $PD\text{-}RCCP$.

5.3.3.3 New Step Continuous-Time RCCP Formulation : *ST-RCCP$_c$*

We formulate the step-based continuous-time RCCP model, building upon the *SD-RCCP* formulation, as follows.

$$s_{ip} \leq D_p \times zs_{ip}, \qquad \forall i \in I,\ p \in P, \qquad (1i)$$

$$s_{ip} \geq D_p \times zs_{ip-1}, \qquad \forall i \in I,\ p \in P, \qquad (2i)$$

$$f_{ip} \leq D_p \times ys_{ip}, \qquad \forall i \in I,\ p \in P, \qquad (3i)$$

$$f_{ip} \geq D_p \times ys_{ip-1}, \qquad \forall i \in I,\ p \in P, \qquad (4i)$$

Constraints (4p), (1s), (2s'), (7d), (8d), (9d), (10d), (A), (B), (C), (D), (E), and (F).

Constraints (1i) to (4i) enforce consistency between the continuous variables and the binary step variables.

5.3.3.4 Improved Step Continuous-Time RCCP Formulation : *SI-RCCP$_c$*

We formulate the final step-based continuous-time RCCP model by incorporating the improvement proposed earlier in the formulation *ST-RCCP* (Proposition 7).

$$D_p + D_{p+1} - (f_{ip} + f_{ip+1}) \geq (D_p + \epsilon) \times (1 - ys_{ip}), \qquad \forall i \in I,\ p \in P \qquad (5i')$$

Constraints (5i') correspond to a reformulation of constraints (7s') using step variables instead of pulse variables.

Observation 4 : With the substitution

$$s_{ip} = D_p \times zs_{ip}, \quad f_{ip} = D_p \times ys_{ip}$$

It can be verified that the LP-relaxation feasible region of $SI\text{-}RCCP_c$ contains that of $SD\text{-}RCCP$.

We conclude this section with Figure 5.2, which summarizes the lineage of RCCP formulations along the Pulse (top) and Step (bottom) paths. Arrows point from a base model (left) to a derived model (right), with each label indicating the main change. **APV** denotes a switch to aggregated and variable periods, which requires redefining how the earliest and latest start and finish dates are handled in aggregated instances (Section 5.3.1). **CT** denotes a switch to continuous time through the introduction of new continuous variables in the RCCP problem, adapted to pulse- and step-variable formulations ; parenthesized references indicate added or modified constraints. Green nodes represent the models analyzed in this work, while black nodes correspond to formulations introduced in the literature under the same variable-intensity assumptions. A green arrow indicates LP relaxation inclusion (left $\subseteq$ right), whereas a blue arrow indicates the opposite inclusion (right $\subseteq$ left). (Sections 5.3.2.2, 5.3.2.4, and 5.3.3.2). Finally, the dotted link marks the continuous-time reference model, whose constraint construction logic differs from that of the main lineage.

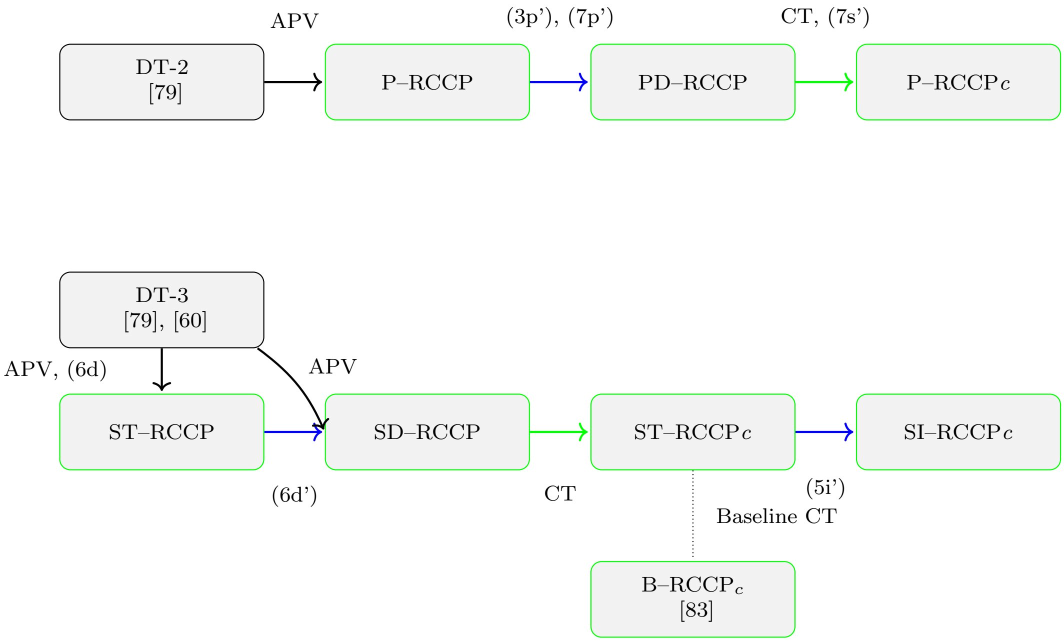

FIGURE 5.2 Compact lineage of RCCP formulations : Pulse (top), Step (bottom)

5.4 Tests and Implementation

In this section, we describe the environment used for our computational experiments. We outline the performance criteria and then detail how the instances were generated and solved. Then, we compare the formulations introduced in Section 5.3, considering the time-driven variant.

5.4.1 Conditions of experiments

The computational experiments were conducted on a computing grid equipped with dual Intel Xeon Gold 6258R CPUs, each running at 2.70 GHz, and a total memory capacity of 512 GB. The IBM ILOG CPLEX 22.1.1 MIP solver was used, with either one or two threads depending on the number of WPs. Performance was evaluated using several metrics, including execution time, number of explored nodes, number of dual simplex iterations, and the integrality gap.

Model Implementation Conditions

We begin our evaluation with the disaggregated instances of [32] using the discrete-time formulations. These instances are based on the time-driven variant and represent an adaptation of the work by De Boer [39] to address the RCCP problem. Each class of instances is defined by two parameters : the number of WPs ($|I|$), taking values of 10, 20, or 50, and the number of resources ($|R|$), taking values of 3, 10, or 20.

In the original Cherkaoui instances, the first four periods are assigned a duration of 1, while the remaining periods are assigned a duration of Δ, where $\Delta = 4$. If the total time horizon is not divisible by Δ, the duration of the fifth period is adjusted to the remainder of the horizon modulo Δ.

According to the findings of [83], for $\Delta > 1$, these instances can be solved to optimality within 5 000 seconds. In the present study, we initially focus on solving the disaggregated versions of the instances (i.e., $\Delta = 1$) across all periods.

We generated additional instances using the same methodology to assess model performance on larger problem sizes, increasing the number of WPs to 80. These instances were also disaggregated to evaluate model scalability and robustness under more complex scenarios. For instances with fewer than 50 WPs, the time limit was 5 000 seconds using a single thread. For instances with 50 or 80 WPs, the execution time limit was also set to 10 000 seconds, but with two threads.

To further highlight the performance of the best-performing model, we generated aggregated instances with 120 WPs. These instances cannot be solved using discrete-time models due to their inherent limitations. Additionally, the best continuous-time model from the literature [83] is unable to solve such large-scale instances to optimality.

5.4.2 Discrete-time RCCP tests

We now compare the discrete-time formulations. Table 6.4 summarizes their performance : *P-RCCP*, *PD-RCCP*, *ST-RCCP* and *SD-RCCP* (described in Section 5.3) across different instance classes of disaggregated instances. **Nb-WPs** indicates the number of WPs ; **Nb-Res** represents the number of resources ; **Models** specifies the tested model ; **Tps (sec)** reports the average execution time (in seconds) ; **Nodes** shows the average number of nodes explored during branch-and-bound ; **Iterations** presents the average number of simplex iterations ; and **Opt-gap (%)** provides the average optimality gap, including instances solved to proven optimality.

Table 6.4 presents the results obtained for easy instances with 10 and 20 WPs.

TABLE 5.4 Summary results for the performance of the discrete-time models for each easy instance class

Nb-Wps	Nb-Res	Models	Tps (sec)	Nodes	Iterations	Opt-gap (%)
10	3	*P-RCCP*	0.02	0	39	0.00
		PD-RCCP	0.02	0	39	0.00
		ST-RCCP	0.02	0	48	0.00
		SD-RCCP	0.01	0	36	0.00
	10	*P-RCCP*	0.02	1	72	0.00
		PD-RCCP	0.02	0	70	0.00
		ST-RCCP	0.02	0	78	0.00
		SD-RCCP	0.01	0	61	0.00
	20	*P-RCCP*	0.03	0	89	0.00
		PD-RCCP	0.03	0	87	0.00
		ST-RCCP	0.02	0	108	0.00
		SD-RCCP	0.01	0	85	0.00
20	3	*P-RCCP*	0.12	32	1 097	0.00
		PD-RCCP	0.14	32	879	0.00
		ST-RCCP	0.10	21	942	0.00
		SD-RCCP	0.05	8	413	0.00
	10	*P-RCCP*	0.11	15	658	0.00
		PD-RCCP	0.12	11	529	0.00
		ST-RCCP	0.11	11	781	0.00
		SD-RCCP	0.06	6	372	0.00
	20	*P-RCCP*	0.13	6	652	0.00
		PD-RCCP	0.14	8	588	0.00
		ST-RCCP	0.13	9	914	0.00
		SD-RCCP	0.07	4	456	0.00

For instances with a moderate number of WPs (10 and 20 WPs), we observe that all tested models reach optimality within a fraction of a second. Although the last model, *SD-RCCP*, tends to outperform the others in terms of the number of dual simplex iterations, the difference remains marginal.

To highlight the differences between the models when testing more complex instances, we include the column **#non-opt**, which indicates the number of instances for which the solver failed to prove optimality. Additionally, the column **#non-feas** reports the number of instances for which no feasible solution was found.

Table 5.5 presents the results obtained.

Table 5.5 Performance of the discrete-time models for each complex instance class

Nb-Wps	Nb-Res	Models	Tps (sec)	Nodes	Iterations	Opt-gap (%)	#non-opt	#non-feas
50	3	*P-RCCP*	1 032.66	784	590 451	3.28	2	1
		PD-RCCP	6.32	1 014	36 163	0.00	0	0
		ST-RCCP	2.63	521	27 556	0.00	0	0
		SD-RCCP	1.76	140	6 828	0.00	0	0
	10	*P-RCCP*	242.71	11 204	639 498	0.00	0	1
		PD-RCCP	64.69	6 792	326 296	0.00	0	0
		ST-RCCP	18.94	1 545	133 212	0.00	0	0
		SD-RCCP	3.62	529	34 139	0.00	0	0
	20	*P-RCCP*	507.72	5 007	700 935	1.28	1	1
		PD-RCCP	174.91	7 295	614 105	0.00	0	0
		ST-RCCP	107.97	5 311	581 723	0.00	0	0
		SD-RCCP	12.87	1 057	82 619	0.00	0	0
80	3	*P-RCCP*	3 621.54	34 546	3 286 270	2.01	7	21
		PD-RCCP	285.20	23 658	1 952 773	0.00	0	0
		ST-RCCP	76.66	6 645	996 295	0.00	0	0
		SD-RCCP	14.48	2 796	173 955	0.00	0	0
	10	*P-RCCP*	5 589.99	670 597	70 406 335	3.07	14	10
		PD-RCCP	2 928.56	204 026	16 094 685	0.21	8	0
		ST-RCCP	1 736.95	57 795	11 342 767	0.12	4	0
		SD-RCCP	417.49	23 535	2 818 495	0.00	0	0
	20	*P-RCCP*	4 347.57	25 809	4 162 085	1.39	8	16
		PD-RCCP	4 129.79	192 535	17 112 872	0.54	16	0
		ST-RCCP	3 030.72	59 486	16 213 877	0.37	10	0
		SD-RCCP	1 260.44	32 304	6 926 348	0.09	3	0

Starting from instances with 50 WPs, the behavior of the models becomes noticeably different. We observe that the formulation of precedence constraints has a significant impact on model performance. In particular, when comparing *P-RCCP* with *PD-RCCP*, the difference is substantial : the disaggregated precedence constraints (7p') lead to consistent improvements across all performance indicators, including execution time, number of explored nodes, and dual simplex iterations. An exception is observed for instances with 80 WPs, where *P-RCCP* exhibits fewer nodes and iterations. This may be explained by the solver spending considerably more time in the presolve phase and in generating cuts to resolve node 0. Furthermore, optimality gaps are consistently reduced across all tested instances.

We also observe that formulations using step variables are significantly better suited to the MIP solver. Although *PD-RCCP* is theoretically stronger than *ST-RCCP*, the empirical results show that *ST-RCCP* performs better in terms of the number of unsolved instances, optimality gaps, and explored nodes. The number of dual simplex iterations is comparable between the two models.

Numerical instabilities. Numerical instability issues were observed primarily in formulations relying on pulse variables, particularly with the P-RCCP model. These difficulties stem from the interaction between the precedence constraints (7p) and constraints involving sums of binary variables, such as (1p) and (2p). In practice, this can lead either to numerical infeasibility (failure to identify feasible solutions) or to excessive solving time, with the solver failing to converge within the time limit of 10 000 seconds.

Replacing the precedence constraints (7p) with the constraints (7p') considerably improved numerical stability and solution efficiency. This improvement may be due to the fact that, in (7p), the variable z_{jp} can take very small values, sometimes as small as the feasibility tolerance set by the solver. Since this can occur at each period p, many variables may remain close to the tolerance threshold. In contrast, in (7p'), the sum $\sum_k z_{jk}$ is governed by a single tolerance, preventing the accumulation of many small values across periods. Additional remedies include parameter tuning (for example, relaxing the feasibility tolerance to 10^{-5} instead of 10^{-6}) and solving the LP relaxation using the barrier method with crossover, an approach known to be more robust under numerical instability. This makes it possible to obtain the same optimal values. In contrast, step-based formulations did not exhibit such issues under the same conditions.

5.4.3 Continuous-Time RCCP Tests

To evaluate the differences between discrete-time and continuous-time models in terms of solver behavior (execution time, number of nodes, and dual simplex iterations) and solution quality, we assess the performance of several continuous-time formulations, including the formulation proposed by [83].

As a first step, we use the same performance indicators as those employed for evaluating the discrete-time formulations.

Table 5.6 presents the results obtained when testing the easy instances (10 and 20 WPs).

TABLE 5.6 Performance of the continuous-time models for each easy instance class

Nb-Wps	Nb-Res	Models	Tps (sec)	Nodes	Iterations	Opt-gap (%)
10	3	$B\text{-}RCCP_c$	0.06	38	639	0.00
		$P\text{-}RCCP_c$	0.02	0	63	0.00
		$ST\text{-}RCCP_c$	0.01	0	44	0.00
		$SI\text{-}RCCP_c$	0.01	0	39	0.00
	10	$B\text{-}RCCP_c$	0.06	31	622	0.00
		$P\text{-}RCCP_c$	0.03	1	89	0.00
		$ST\text{-}RCCP_c$	0.02	1	71	0.00
		$SI\text{-}RCCP_c$	0.01	1	67	0.00
	20	$B\text{-}RCCP_c$	0.07	14	572	0.00
		$P\text{-}RCCP_c$	0.03	1	112	0.00
		$ST\text{-}RCCP_c$	0.02	0	94	0.00
		$SI\text{-}RCCP_c$	0.02	0	91	0.00
20	3	$B\text{-}RCCP_c$	1.75	1 649	24 695	0.00
		$P\text{-}RCCP_c$	0.17	55	1 474	0.00
		$ST\text{-}RCCP_c$	0.08	12	552	0.00
		$SI\text{-}RCCP_c$	0.06	11	508	0.00
	10	$B\text{-}RCCP_c$	0.72	427	8 978	0.00
		$P\text{-}RCCP_c$	0.17	34	1 195	0.00
		$ST\text{-}RCCP_c$	0.10	8	536	0.00
		$SI\text{-}RCCP_c$	0.07	8	476	0.00
	20	$B\text{-}RCCP_c$	0.73	281	7 289	0.00
		$P\text{-}RCCP_c$	0.19	20	1 185	0.00
		$ST\text{-}RCCP_c$	0.12	7	687	0.00
		$SI\text{-}RCCP_c$	0.08	6	596	0.00

Although these instances are easily solved, it appears that models which introduce continuous variables to determine start and end times within each period, while preserving the structure of the discrete-time formulations, are significantly more efficient than the formulation proposed in [83]. Even for relatively small instances, the difference in the number of dual simplex iterations is already noticeable. This gap becomes even more pronounced as the instance size increases to moderate levels. However, we note that for pulse-based formulations, performance remains highly sensitive to how precedence constraints are expressed, similarly to what was observed in the discrete-time version.

We now move on to the more complex instances, those with 50 and 80 WPs (Table 5.7).

Table 5.7 Performance of the continuous-time models for each complex instance class

Nb-Wps	Nb-Res	Models	Tps (sec)	Nodes	Iterations	Opt-gap (%)	#non-opt	#non-feas
50	3	*B-RCCP$_c$*	1 414.89	373 001	10 394 241	0.04	5	2
		P-RCCP$_c$	20.40	2 742	119 432	0.00	0	0
		ST-RCCP$_c$	1.90	322	11 473	0.00	0	0
		SI-RCCP$_c$	1.49	299	10 797	0.00	0	0
	10	*B-RCCP$_c$*	1 189.40	113 169	6 862 938	0.04	2	1
		P-RCCP$_c$	370.03	20 646	1 553 326	0.00	0	0
		ST-RCCP$_c$	15.39	1 039	73 512	0.00	0	0
		SI-RCCP$_c$	9.42	916	65 062	0.00	0	0
	20	*B-RCCP$_c$*	1 293.63	50 635	4 687 493	0.07	3	0
		P-RCCP$_c$	836.37	15 026	2 363 657	0.10	3	0
		ST-RCCP$_c$	66.53	1 754	224 541	0.00	0	0
		SI-RCCP$_c$	41.27	1 543	178 387	0.00	0	0
80	3	*B-RCCP$_c$*	4 194.72	139 347	23 969 024	0.76	5	10
		P-RCCP$_c$	2 201.22	125 300	12 502 869	0.11	5	0
		ST-RCCP$_c$	57.57	5 527	510 234	0.00	0	0
		SI-RCCP$_c$	40.70	3 742	331 321	0.00	0	0
	10	*B-RCCP$_c$*	6 436.25	93 263	28 378 537	4.13	20	4
		P-RCCP$_c$	5 387.34	153 050	21 874 021	0.92	23	0
		ST-RCCP$_c$	1 408.70	26 175	6 144 798	0.07	2	0
		SI-RCCP$_c$	1 116.20	20 658	5 271 978	0.08	3	0
	20	*B-RCCP$_c$*	6 912.76	77 606	26 591 127	3.10	31	0
		P-RCCP$_c$	6 856.84	152 860	20 406 731	1.26	31	0
		ST-RCCP$_c$	3 076.50	32 957	9 950 489	0.33	9	0
		SI-RCCP$_c$	2 192.93	24 317	7 800 981	0.22	7	0

Starting from instances with 50 WPs, the differences among the models become more pronounced. As previously discussed, step-based formulations tend to reduce execution time. Based on the results obtained, it is clear that the type of variables used plays a key role in model performance, particularly when comparing pulse variables with step variables.

However, the structure of the constraints solved by CPLEX has a considerable influence on solver behavior, although to a lesser extent than the impact of introducing additional variables. Ultimately, when comparing the best continuous-time model proposed to date in the RCCP context with the best-performing model from this study, we observe a significant improvement. The *SI-RCCP$_c$* model not only ensures optimality in the majority of cases but also results in an average execution time that is seven times faster.

Numerical instability becomes particularly noticeable for instances with 80 WPs in the *B-RCCP$_c$* model. In some cases, it was necessary to adjust CPLEX parameters to obtain feasible solutions. The constraints (1c), (2c), (3c), (4c), (9c), and (10c) may be the main sources of these issues, since big-M formulations are well known to be a critical source of numerical issues in MIP. When the planning horizon increases, the corresponding M values must increase proportionally, which deteriorates the numerical conditioning of the model. Large coefficients produce weak relaxations and numerical instability, especially when combined with integer variables. This often results in a loss of numerical precision during simplex pivots, and may even lead the solver to declare infeasibility despite the existence of feasible solutions. This phenomenon was observed starting at 80 WPs.

Numerical difficulties could be mitigated by adjusting integrality tolerances and solving the LP relaxation with the barrier method, thereby recovering the same optimal values. In contrast, no such issues were observed with the *SI-RCCP*$_c$ model, whose structure seems to mitigate numerical instability under the same conditions.

To complement this comparison of the linear relaxations of *B-RCCP*$_c$ and *SI-RCCP*$_c$, we further evaluate the performance of the relaxed models. Rather than assessing the relaxed models globally, we focus on their relaxation at node 0, i.e., immediately after the presolve phase and the addition of cuts by CPLEX. This approach is motivated by the observation that the quality of the initial relaxation does not always correlate with the overall performance of the model [63]. Additionally, we compare the number of fractional variables in the relaxed solutions provided by the solver to assess how effectively each model guides convergence toward an integer solution.

Table 5.8 summarizes the root-node (node 0) results across the instance classes. The column **Nb-WP** denotes the number of WPs in the instance. The column **Nb-res** denotes the number of resources. **M-Relax-gap (%)** reports the mean percentage gap between the LP relaxations at node 0 of *B-RCCP*$_c$ and *SI-RCCP*$_c$ (after presolve and cut generation), computed over all instances in the class, thereby reflecting the relative quality of the new model's LP relaxation compared to the baseline, calculated by Min-relax-gap $= \frac{LB_{\mathrm{SI}}-LB_{\mathrm{B}}}{LB_{\mathrm{B}}} \times 100$, where LB_{SI} denotes the bound obtained from the relaxed formulation of *SI-RCCP*$_c$ and LB_{B} denotes the bound from the relaxed formulation of *B-RCCP*$_c$. **Min-Relax-gap (%)** and **Max-Relax-gap (%)** report, respectively, the minimum and maximum of that per-instance percentage gap within the class. **Time-1 (sec)** and **Time-2 (sec)** give the average time (in seconds) to solve the root-node LP relaxation for *B-RCCP*$_c$ and *SI-RCCP*$_c$, respectively. Finally, **IINF-1** and **IINF-2** report the average number of fractional variables among binary variables in the root-node LP relaxation for *B-RCCP*$_c$ and *SI-RCCP*$_c$, respectively, used to assess progress toward an integer solution.

TABLE 5.8 Resolution of node 0 for each instance class

Nb-WP	Nb-res	M-Relax-gap (%)	Min-relax-gap (%)	Max-Relax-gap (%)	Time-1 (sec)	Time-2 (sec)	IINF-1	IINF-2
10	3	6.19	0.00	67.80	0.05	0.02	8.74	1.38
	10	2.38	-1.49	10.25	0.06	0.03	9.46	2.50
	20	1.05	0.00	5.10	0.07	0.03	10.10	4.18
20	3	6.11	-0.06	20.27	0.39	0.06	40.20	15.88
	10	2.92	0.00	8.35	0.39	0.07	46.10	18.18
	20	1.53	0.16	4.84	0.46	0.08	47.00	18.42
50	3	10.84	0.68	62.71	37.40	2.33	129.58	73.34
	10	5.76	0.48	16.78	45.24	3.09	192.76	219.06
	20	3.29	-0.71	6.43	31.52	3.52	366.90	289.34
80	3	0.90	0.01	87.69	2 033.11	31.60	320.36	181.38
	10	0.94	-17.46	100.00	4 193.57	44.35	4 121.88	276.78
	20	0.92	0.73	98.94	2 685.63	49.41	3 193.50	700.74

We observe that *SI-RCCP*$_c$ performs better overall across classes in terms of execution time and the quality of relaxed solutions. For a fixed number of WPs, the relaxed gap is smaller when the number of WPs is low. Additionally, *SI-RCCP*$_c$ results in fewer fractional variables, highlighting the advantage of the relaxation provided by this model. As the number of WPs (Nb-WP) and resources (Nb-res) increase, execution times for both models grow.

For *B-RCCP*$_c$, we observed that solving the relaxation often requires a very long execution time, sometimes reaching the upper limit of 10 000 seconds. This is because the solver cannot solve the relaxation directly and therefore relies heavily on the presolve phase to reinforce the model. In some instances, the presolve process, through the addition of solver-generated cuts, strengthens the formulation and yields tighter bounds, although at the cost of significantly higher computation times. This explains why the LP relaxation of *B-RCCP*$_c$ can provide better bounds in particular cases (e.g., the -17.46% value reported in *Min-relax-gap %*).

Consistent with these observations, the execution time of the *B-RCCP*$_c$ LP formulation grows exponentially with the problem size, whereas the *SI-RCCP*$_c$ LP formulation exhibits nearly linear growth.

5.4.4 Comparison between the discrete and continuous approaches

To highlight the practical impact of improving model performance in the RCCP context, we compare the formulations based on the following aspects :

1. The impact on execution time and the loss in efficiency associated with the continuous-time approach.
2. The impact on solution quality and the potential cost increase resulting from the discrete-time approach (only for instances with short period lengths, since this approach is not realistic for aggregated instances).

For this purpose, we consider instances with 50 and 80 WPs, as they provide a representative benchmark for evaluating the execution times of all discrete-time and continuous-time models (Figure 5.3).

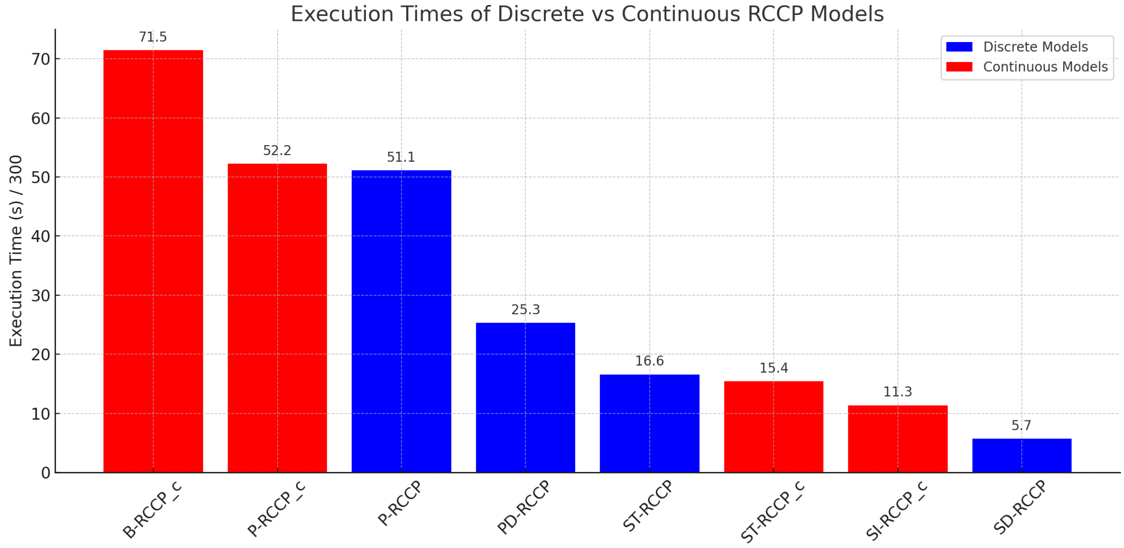

FIGURE 5.3 Execution time comparison between discrete RCCP models (in blue) and continuous RCCP models (in red)

In terms of execution time, step-based formulations clearly outperform pulse-based ones, and strengthened disaggregated precedence relations yield superior execution times compared to the alternatives. Among discrete models, the ranking is therefore : *SD-RCCP* $\rightarrow$ *ST-RCCP* $\rightarrow$ *PD-RCCP* $\rightarrow$ *P-RCCP*. Among continuous models, the order is : *SI-RCCP*$_c$ $\rightarrow$ *ST-RCCP*$_c$ $\rightarrow$ *P-RCCP*$_c$ $\rightarrow$ *B-RCCP*$_c$. Although *B-RCCP*$_c$ is a step-variable formulation, its underlying logic differs from that of the discrete models, which explains its comparatively weaker performance relative to the others. In fact, within the RCCP context, even small changes can lead to significant differences in performance, and in some cases, these differences are influenced by numerical stability issues.

Finally, we compare the impact of the formulations on solution quality. This is illustrated in Figure 5.4.

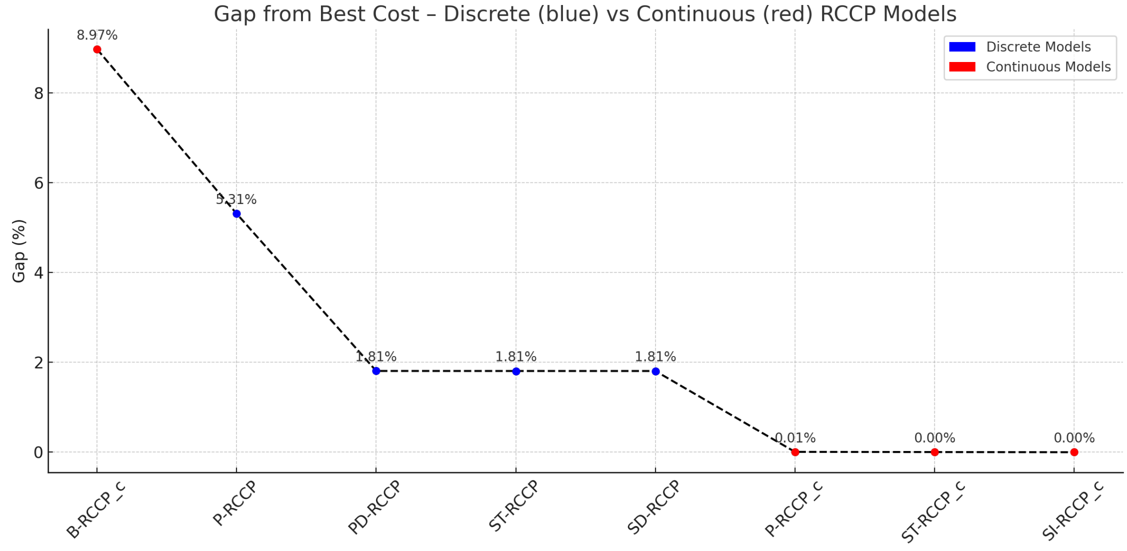

FIGURE 5.4 Gap from the best cost : Continuous vs. Discrete RCCP formulations

When comparing the best continuous-time formulation, $SI\text{-}RCCP_c$, with the best previously studied model, $B\text{-}RCCP_c$, an average improvement of 3.9% was achieved on instances that could not be solved by $B\text{-}RCCP_c$. Notably, for some instances with 80 WPs and 10 resources, improvements of up to 67% were observed. This indicates that the difference lies not only in the lower bound, but also in the quality of the obtained solutions.

Although the discrete formulations solve more instances than the continuous models, which is expected given that continuous formulations are more complex to solve, continuous models tend to yield better costs on large and disaggregated instances. This can be explained by their underlying assumptions : discrete-time formulations approximate the horizon by splitting it into fixed periods, which simplifies modeling but restricts WP start times to the boundaries of these intervals. In contrast, continuous-time formulations allow WPs to start at any time within the horizon, offering a more accurate representation of precedence relations.

Obviously, the cost tends to deteriorate when time periods are aggregated. In some cases, we may even encounter instances with no feasible solution (by testing *SD-RCCP*), since the critical path (CPM) may become extended. This occurs because the earliest and latest start times can no longer fall within an aggregated period and must instead be shifted to the beginning of the corresponding aggregated period.

To emphasize the importance of selecting an appropriate continuous-time formulation, we test the best-performing model, $SI\text{-}RCCP_c$, on larger instances (120 WPs), using an aggregated time structure (Figure 5.5). $SD\text{-}RCCP_c$ is used for testing the disret-time RCCP.

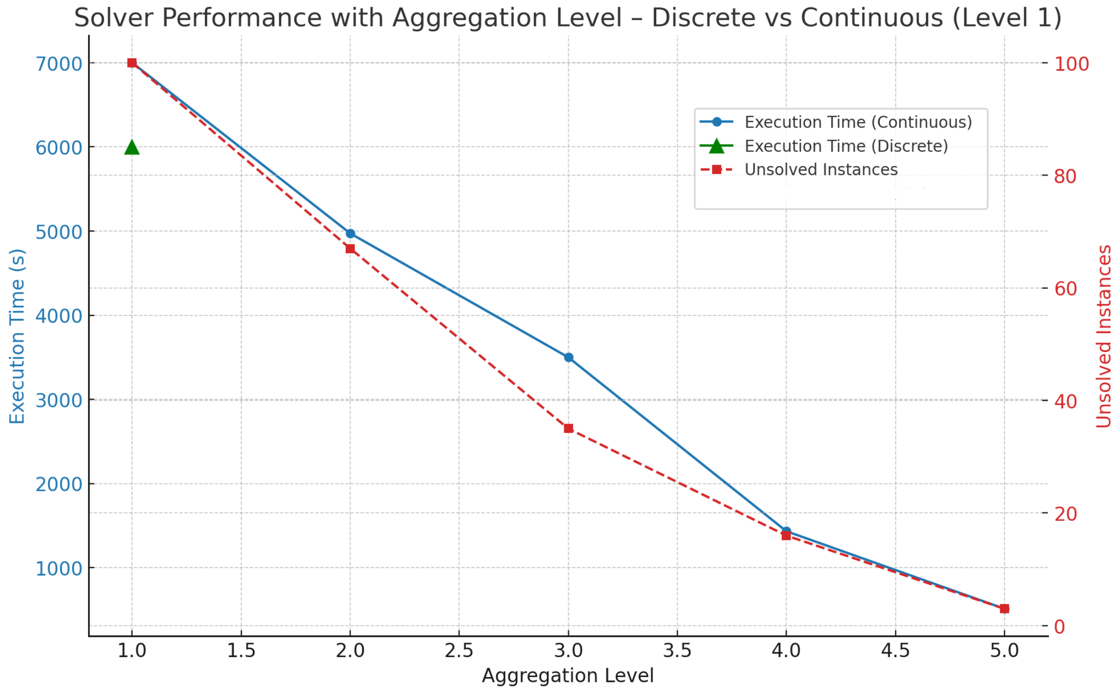

FIGURE 5.5 Solver performance with aggregated level : continuous vs discrete formulations

When the number of WPs becomes too large, neither discrete nor continuous models are able to optimally solve disaggregated instances. Execution time appears to be only marginally affected by whether the best discrete or continuous formulation is used. However, continuous-time models offer greater structural flexibility, particularly in allowing the aggregation of time periods, especially those corresponding to the intermediate and final months of the planning horizon. As demonstrated by [32], this approach is practical and effective in a tactical planning context.

It is important to emphasize that the choice of a continuous-time model, along with its proper refinement, is crucial. Even when periods are aggregated, many models are still unable to handle such large-scale instances, highlighting the need for improved continuous formulations.

These observations highlight the scalability implications of RCCP formulations : while tackling the problem with disaggregated periods quickly becomes intractable as the number of WPs grows (e.g., 120 WPs), continuous-time formulations combined with period aggregation can still handle large-scale instances (Figure 5.5), and the resulting solution is well adapted to tactical planning challenges. This shows that scalability depends not only on problem size but also on the structural choices made in the formulation, such as whether and how periods are aggregated.

5.5 Conclusion and Future Work

In this study, we compared several discrete-time and continuous-time models for the RCCP problem, including well-established formulations from the literature. Our results show that the newly proposed continuous-time models significantly improve the performance of the RCCP, designed to accelerate the resolution process and improve solution quality.

To assess the adaptability of the proposed models, we generated a wide range of test instances. This allowed us to investigate the impact of the structural complexity of continuous-time models on computational performance, offering valuable insights into the behavior of RCCP models under realistic planning conditions.

The $SI\text{-}RCCP_c$ model outperformed previous formulations, particularly in terms of execution time and solution quality. For instance, improvements of up to 67% were observed on some instances with 80 WPs, compared to the $B\text{-}RCCP_c$ model. An average improvement of 3.9% was observed across previously unsolved instances with 50 and 80 WPs. These results highlight the effectiveness of advanced MIP approaches in achieving optimality in tactical project planning. This makes it possible to tackle larger instances with 120 WPs when aggregating periods, something that was not feasible with discrete-time models due to their unsuitable structure, nor with the continuous-time models proposed in the literature, which often suffer from slow solver convergence and, in some cases, fail to find feasible solutions.

Although continuous-time formulations are generally more flexible and better suited to real-world planning contexts, their practical adoption by project managers remains challenging. In future work, we aim to extend the approach toward a resource-driven variant by integrating constraints that account for resource-leveling mechanisms. This would enable better management of fluctuations in resource utilization, which are often costly and destabilizing in practice. Moreover, the current best continuous-time model is already well suited to such a resource-driven extension, since the necessary variables to compute the exact start and end times of WPs are explicitly included in our formulation. Another promising direction is the development of stochastic RCCP formulations, which can be achieved by modeling uncertain parameters, such as resource demands, through a finite set of scenarios.

Additionally, it would be interesting to develop an exact method capable of accelerating makespan minimization while efficiently generating the Pareto front, thereby enabling decision-makers to explore trade-offs between resource utilization and project duration.

Further extensions may also include the use of hybrid methods combining decomposition techniques or learning-based heuristics to improve scalability. Finally, incorporating uncertainty in WP durations and resource availability will bring the model closer to real-world planning environments and improve its robustness.

5.6 Appendix : Proofs of Main Results

5.6.1 Proof of Proposition 5

Proof. Given that the intensity varies from one period to another, the duration of WPs is also variable. Suppose that replacing (3p) with (3p') results in the following condition :

$$\exists p \in P,\ i \in I :\ D_p \times \left(\sum_{k \in Ps_i \cap \{1,\ldots,p\}} z_{ik} - \sum_{k \in Pf_i \cap \{1,\ldots,p\}} y_{ik} \right) > d_{ip}$$

It is always possible to extend the duration d_{ip} to satisfy the original equality (3p). The extended solution remains feasible because constraints (B) are not mandatory when $Q_i^{\min} = 0$, and constraints (A) are inherently more relaxed in this formulation. This adjustment transitions the solution from $S1$ to $S1'$. This situation is illustrated in Figure 5.6.

The length of the red segment intersecting period p represents $D_p \times \sum_{k \in Ps_i \cap \{1,\ldots,p\}} z_{ik}$, with its left endpoint indicating the start time of the WP. Similarly, the length of the blue segment intersecting period p corresponds to $D_p \times \sum_{k \in Pf_i \cap \{1,\ldots,p\}} y_{ik}$, with its left endpoint representing the end time of the WP. The green segment connects the start and end times, thereby representing the total duration of the WP in the considered period. The continuous green line indicates a possible value of d_{ip} obtained from the constraints (3p'), while the dashed green line shows the adjustment when transitioning from (3p') to the constraints (3p) in solution (S_1').

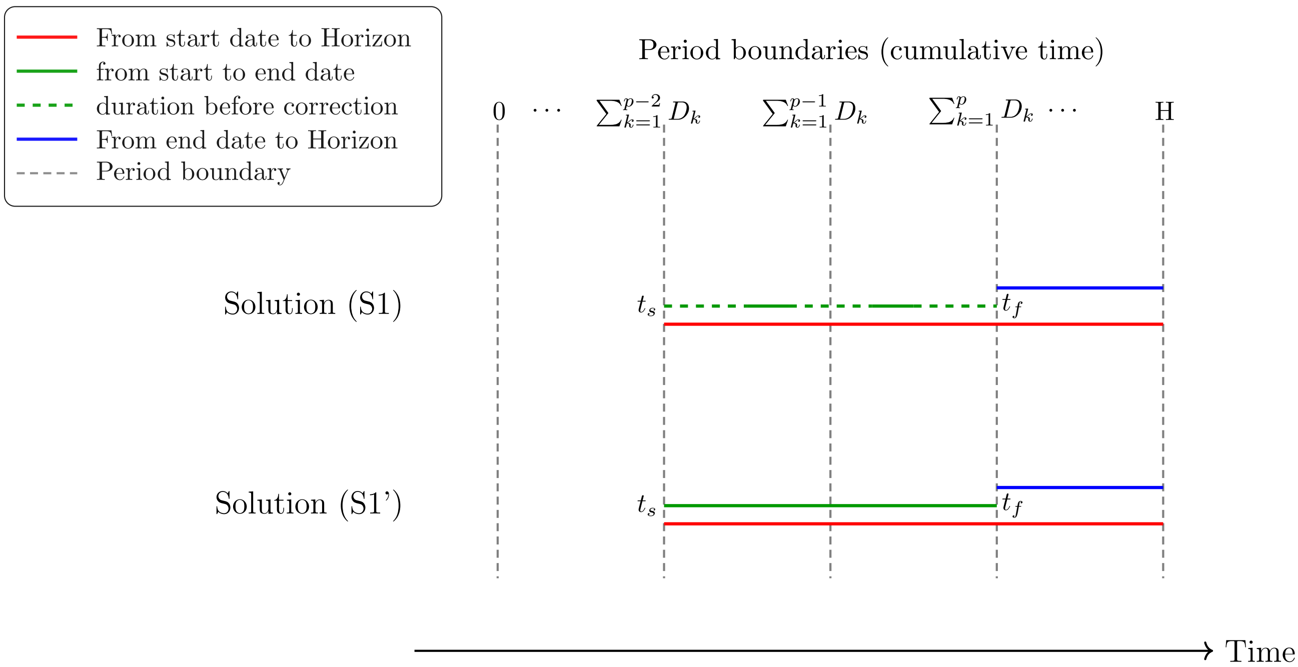

FIGURE 5.6 Impact of relaxing constraints (3p) to (3p')

□

5.6.2 Proof of Proposition 6

Proof. We prove that constraints (6d') imply constraints (6d).

We can rewrite (6d') as $D_p \times (zs_{jp} - zf_{ip}) \leq 0$. Summing over all periods $p \in P$, we obtain :

$$\sum_{k \in P} D_k \times zs_{jk} - \sum_{k \in P} D_k \times ys_{ik} \leq 0$$

This can be decomposed as :

$$\sum_{k \notin \{\pi(ES_j+1),\ldots,|P|\}} D_k \times zs_{jk} + \sum_{k=\pi(ES_j+1)}^{|P|} D_k \times zs_{jk}$$
$$- \left(\sum_{k=\pi(EF_i)}^{|P|} D_k \times ys_{ik} + \sum_{k \notin \{\pi(EF_i),\ldots,|P|\}} D_k \times ys_{ik} \right) \leq 0$$

we obtain :

$$\sum_{k=\pi(ES_j+1)}^{|P|} D_k \times zs_{jk} \leq \sum_{k=\pi(EF_i)}^{|P|} D_k \times ys_{ik}$$

Now, recall from constraints (1d) and (2d) that :

$$tf_i = H - \sum_{k=\pi(EF_i)}^{|P|} D_k \times ys_{ik}, \quad ts_j = H - \sum_{k=\pi(ES_j+1)}^{|P|} D_k \times zs_{jk}$$

Substituting into the inequality, we get :

$$ts_j \geq tf_i,$$

which corresponds to constraints (6d). Thus, constraints (6d') imply constraints (6d).

Now, let us consider a case where the LP relaxation of *SD-RCCP* is tighter than that of *ST-RCCP*. This is illustrated using the same instance presented in the proof of Proposition 4, along with the corresponding relaxed solution provided in Table 5.9.

TABLE 5.9 Example of feasible relaxed solution under *ST-RCCP* and unfeasible when considering *SD-RCCP*

WP_i	$zs_{i,1}$	$zs_{i,2}$	$zs_{i,3}$	$ys_{i,1}$	$ys_{i,2}$	$ys_{i,3}$	$x_{i,1}$	$x_{i,2}$	$x_{i,3}$
1	1	1	1	$\mathbf{\frac{2}{3}}$	$\mathbf{\frac{2}{3}}$	1	1	0	0
2	$\mathbf{\frac{1}{3}}$	1	1	0	$\frac{1}{2}$	1	1	0	0

This solution satisfies constraints (6d) ($3 \times \frac{1}{3} + 2 \times 1 + 1 \leq 3 \times \frac{2}{3} + 2 \times \frac{2}{3} + 1$). However, we observe that for $p = 2 : 1 - \frac{2}{3} > 0$ leads to a violation of the disaggregated precedence constraints (6d'). This demonstrates that the feasible region is more restricted when disaggregated constraints are applied, resulting in a tighter formulation of the problem.

□

5.6.3 Proof of Theorem 5.3.3.2

Proof. It suffices to consider instance I, using the same relaxed solution as in the proof of Proposition 4, and to apply the following substitution :

$$s_{ip} = D_p \times \sum_{k \in Ps_i \cap \{1,\ldots,p\}} z_{ik}, \quad f_{ip} = D_p \times \sum_{k \in Pf_i \cap \{1,\ldots,p\}} y_{ik}.$$

By relaxing the integrality constraints on the variables z_{ip} and y_{ip}, and following the same steps as in the proof of Proposition 4, we obtain the result stated in the theorem.

□

5.6.4 Proof of Theorem 5.3.3.2

Proof. Let us consider instance I, using the same relaxed solution as in the proof of Proposition 6, and apply the following simplification :

$$s_{ip} = D_p \times zs_{ip}, \quad f_{ip} = D_p \times ys_{ip}, \quad zs_{ip} = \sum_{k \in Ps_i \cap \{1,\ldots,p\}} z_{ik}, \quad ys_{ip} = \sum_{k \in Pf_i \cap \{1,\ldots,p\}} y_{ik}.$$

By applying these substitutions and following the same steps as in the proof of Proposition 6, we confirm the strengthening of the formulation when constraints (2s') are used.

□

5.6.5 Proof of Proposition 7

Proof. We prove that constraints (7s') eliminate the (OC) solution without impacting the solution quality when ϵ is sufficiently small. Let us enumerate all the possible cases :

1. If $0 \leq D_p - f_{ip} < \epsilon$, then

$$D_p + D_{p+1} - (f_{ip} + f_{ip+1}) < D_p + \epsilon$$

   and $\sum_{k \in Ps_i \cap \{1,\ldots,p\}} y_{ik} = 0$. If $D_p - f_{ip} = 0$, this case becomes infeasible when constraints (7s') are incorporated, as they enforce $y_{ip} = 1$. As a result, the (OC) possibility is eliminated. If $D_p - f_{ip} > 0$, the end date must be slightly shifted so that $D_p - f_{ip} = \epsilon$.

2. If $D_{p+1} - f_{ip+1} > \epsilon$, then

$$D_p + D_{p+1} - (f_{ip} + f_{ip+1}) > D_p + \epsilon$$

   and $y_{ip+1} = 1$. This case remains feasible even in the presence of constraints (7s').

3. If $D_p - f_{ip} < D_p$, then

$$D_p + D_{p+1} - (f_{ip} + f_{ip+1}) < D_p + \epsilon$$

   and $y_{ip} = 1$. This case also remains feasible without being affected by the constraints (7s').

□

CHAPITRE 6 ARTICLE 3: OPTIMIZING TACTICAL PROJECT PLANNING WITH A RESOURCE-LEVELED RCCP MODEL

Anis Noureddine[1,2,3], Robert Pellerin[1,3], François Soumis[1,2], Raphaël Boudreault[4]

[1] *Département de mathématiques et de génie industriel, Polytechnique Montréal*
[2] *GERAD, Montréal, Canada*
[3] *CIRRELT, Montréal, Canada*
[4] *Thales, Québec, Canada*

OR Spectrum, submitted on 29 October 2025

Abstract

Efficient resource leveling is a critical challenge in ship refit and maintenance planning, where fluctuating labor demand leads to costly subcontracting, overtime, and risks of project delay. To address this issue, this paper presents a continuous-time mixed-integer linear programming approach to Rough Cut Capacity Planning (RCCP) for tactical project planning, which integrates resource leveling by treating work package execution intensities as decision variables. The model allows greater flexibility in execution intensities and enforces a single-peaked resource usage profile, in which demand rises to a peak and then declines, thereby reducing repeated hiring and rehiring while assessing impacts on overall project duration. Computational experiments on specially generated instances show that allowing reduced coupling between resource-specific intensities improves workload leveling by up to 37.16% compared with the classical RCCP, at the cost of increased computation time. To accelerate solution times for makespan in the single-peak resource profile formulation, we introduce an incremental lower-bound tightening strategy that yields about a tenfold average speedup on our test instances. When the planning horizon is too constrained to admit a unimodal profile, the model adapts by minimizing inter-period resource fluctuations to avoid costly variations. Roughly 18% of cases required horizon extension to accommodate unimodality. Finally, a bi-objective ε-constraint analysis quantifies the trade-off between project duration and resource stability for instances with extended horizons. The analysis considers three factors : the total project duration, the variation of workload between periods, and whether or not the workload profile is unimodal.

keywords RCCP, resource leveling, linear mixed programming, single-peak profile, inter-period variations

6.1 Introduction

Tactical planning plays a central role in project management by assigning resources to work packages (WPs) using aggregated data over weekly or monthly time buckets. Positioned before detailed operational scheduling, this phase aims to detect feasibility issues early in the process [32, 39, 86].

One of the main challenges at this planning level is the presence of significant fluctuations in resource utilization, often manifested as sharp peaks and valleys [44, 45]. Such instability typically leads to increased costs, as it often necessitates repeated reoptimization to restore feasible and efficient plans [66]. These fluctuations also have several direct negative impacts on workforce management and operational efficiency : they reduce employee retention, create inefficiencies in later stages of scheduling, require frequent hiring and releasing of personnel, interrupt the learning process of workers, and leave employees underutilized during periods of low demand [40, 49, 70, 98]. Therefore, achieving a leveled resource profile is critical for facilitating efficient project execution.

These issues are especially critical in the naval domain, in particular for ship refit and maintenance planning, since dock periods are fixed years in advance and delays directly affect fleet readiness [2, 21, 22]. Peaks in labor demand often require costly subcontracting or overtime, while valleys create idle time for permanent staff and underutilization of shipyard facilities. Moreover, the wide range of specialized trades involved in refit projects–such as welders, electricians, and pipefitters–makes it particularly challenging to balance workload in a way that avoids repeated hiring and releasing of personnel. Achieving leveled resource utilization is therefore essential, not only for controlling project costs, but also for ensuring that naval and commercial vessels return to service on time.

Various models have been developed to support tactical planning. Some focus on integrated planning and scheduling in engineer-to-order (ETO) environments [81, 82], while others rely on Rough Cut Capacity Planning (RCCP), which assigns resources using aggregated data and variable resource intensities [32, 39, 47]. Unlike the Resource-Constrained Project Scheduling Problem (RCPSP), which addresses on detailed activity-level scheduling with exact start and finish times, RCCP operates at a higher tactical level by using aggregated time buckets and resource capacities to provide early feasibility assessments. These models generally aim to either (1) minimize project duration under fixed resource capacities (time-driven), (2) minimize the cost of external resources under a fixed project duration (resource-driven), or (3) achieve a trade-off between both objectives [47, 83].

Although some of these models address resource leveling by trying to reduce the cost of external resources [15, 103], their objectives do not penalize situations where resource demand reaches very high levels. As a result, they are not effective in reducing fluctuations in resource usage. Other approaches, such as the one proposed by [16], aim to improve workload regularity by minimizing positive variations between consecutive periods, thereby limiting abrupt increases in demand. However, this strategy overlooks workload decreases and does not explicitly control the overall shape of the resource usage profile. For this reason, addressing resource leveling is a complex task, as it involves multiple challenges that must be considered.

In this study, we propose a novel resource-leveling approach integrated into a continuous-time RCCP framework. Our method introduces new constraints to enforce a unimodal workload profile, capturing the benefits of a structured, unimodal resource usage curve. This helps avoid frequent hiring and rehiring of workers and ensures that workload progression aligns with real-world scenarios, where projects typically start gradually, increase resource usage over time, and then decrease progressively toward the end. To improve computational efficiency, we also implement an incremental lower-bound tightening mechanism within the MIP solver. We develop a construction heuristic to minimize project duration when defining the initial planning horizon. When this structured profile significantly increases the total project duration, we propose an alternative formulation that focuses on minimizing inter-period workload fluctuations. Finally, we conduct a bi-objective analysis to explore the trade-off between project duration and resource leveling. To support this study, we generate a new set of benchmark instances tailored to the context of tactical planning.

The remainder of this paper is organized as follows. Section 6.2 reviews the literature on the resource-leveling problem. Section 6.3 presents the mathematical formulations, the acceleration procedure developed for our model, and the construction heuristic. Section 6.4 describes the instance-generation procedure and the experimental setup. Section 6.5 reports the computational results, emphasizing the impact of the proposed leveling constraints. Finally, Section 6.6 concludes the paper and outlines directions for future research.

6.2 Literature Review

The resource leveling problem has been studied in relation to RCCP and other closely related frameworks, with the common goal of reducing fluctuation of workload and improving planning stability. This section reviews the main contributions in this area, focusing on : (i) existing RCCP formulations (or related models) that incorporate resource leveling techniques and their solution approaches, and (ii) comparative studies that summarize the principal methods used for workload leveling.

Many contributions have addressed the resource leveling problem across various planning contexts by minimizing fluctuations in resource usage. Some authors propose exact methods. For example, [70] present a MIP model for resource leveling in linear construction projects such as highways or pipelines. By minimizing absolute differences in resource usage between consecutive periods, their approach smooths staffing levels while respecting continuity constraints. This exact method proves effective in situations where traditional critical path methods (CPM) are not efficient. [45] also propose exact solution methods for resource leveling problems, focusing on minimizing workload fluctuations while meeting project deadlines. They formulate several MIP models and implement branch-and-bound algorithms to handle different leveling objectives, such as reducing variance or workload variation between periods. Their approach guarantees optimal solutions for medium-sized instances. Finally, [37] compare multiple objective functions for resource leveling by applying a MIP-based approach to a real-world construction case study. They conclude that minimizing one leveling objective does not imply a reduction in others relative to the pre-leveling solution, highlighting the need for a multi-objective approach when aiming for multiple forms of leveling.

In contexts where execution intensities are variable, [15] propose a mathematical programming model that incorporates generalized precedence constraints and variable execution intensities to reduce outsourcing needs. A Lagrangian relaxation is applied to solve the model efficiently. [16] address the total adjustment cost problem, proposing a MIP model where both activity durations and execution intensities are decision variables. Their objective is to minimize the total cost of workload adjustments by penalizing increases in resource usage between successive periods.

While previous models may incorporate variable execution intensities, they generally assume that all resources assigned to a WP operate at the same intensity. This widely adopted assumption facilitates workload leveling but can overly restrict flexibility, particularly in tactical planning contexts where WPs are aggregated. To the best of our knowledge, the only works that consider completely independent intensities across resources are those by [102, 103], conducted within the Resource-Constrained Project Scheduling Problem with Variable Intensities (RCPSVP) framework. Although this approach allows each resource to progress at its own pace, it may lead to a lack of coordination between trades, potentially creating situations where the progress of one resource blocks or delays another.

In their first study, [102] propose a MIP model for the fixed-period aggregated resource leveling problem with variable job durations, aiming to minimize the cost of external resources. Their formulation allows nonidentical execution intensities across resources, thereby improving flexibility and solution quality. More recently, [103] applied Benders decomposition to the same model to accelerate the solution process.

Other studies emphasize heuristic or hybrid approaches. [10] formulate a resource leveling model for make-to-order production and study three objective functions : minimizing the sum of squared workloads, the sum of absolute deviations from the mean, and the peak resource usage. An Iterated Greedy metaheuristic is proposed to solve large-scale instances. [69] develop a genetic algorithm (GA) combined with fuzzy logic to address the resource leveling problem in helicopter maintenance scheduling. The model aims to minimize workload variance across periods while handling uncertainties in activity durations and resource availability using fuzzy sets. [53] propose a genetic algorithm to simultaneously address resource allocation and leveling in project scheduling. The model focuses on minimizing the total project duration while reducing resource fluctuations through an adaptive penalty mechanism. By encoding project schedules as chromosomes, the algorithm explores trade-offs between efficiency and workload stability, making it suitable for complex, real-world projects. [96] propose a multi-heuristic approach to address the resource leveling problem in construction projects. The method combines several heuristics to guide the local search process, particularly by shifting non-critical activities to reduce fluctuations in resource demand. A simulated annealing procedure is incorporated to escape local optima and enhance solution quality. The objective is to minimize resource demand variance and align the workload profile with a predefined smoother target. Experimental results demonstrate that this multi-heuristic strategy outperforms traditional single-heuristic methods in producing smoother resource usage profiles.

[40] define two leveling indicators. The first, the Release and ReHire Index (RHH), measures the number of times resources are released and then rehired, capturing workforce instability. The second, the Resource Idle Days Index (RID), quantifies the total number of idle days for resources, reflecting inefficiencies due to underutilization. In the same direction, [107] compare various indicators to evaluate their effectiveness in smoothing resource demand into a unimodal shape. They emphasize that achieving a unimodal resource profile, characterized by a single peak followed by a smooth decline, is often considered the ideal workload profile.

Table 6.2 presents a comparative overview of studies addressing the resource leveling problem, detailing the solution methods, objective functions, and the application contexts in which these approaches were tested.

Despite the diversity of approaches found in the literature, most studies addressing the resource leveling problem focus on minimizing fluctuations using variance-based measures, such as the sum of squared deviations or peak resource demand. While these objective functions provide useful approximations of workload stability, they often fail to enforce a specific temporal structure, such as a single-peaked profile, which is critical in avoiding repeated hiring and layoffs. Moreover, only a few models explicitly consider execution intensities as decision variables or explore continuous-time formulations in the context of resource leveling, which may better reflect real-world conditions. The majority of contributions are also application specific, targeting either construction or production systems. Finally, exact methods such as MIP offer solution guarantees but often suffer from scalability limitations, highlighting the need for acceleration techniques to enhance computational performance.

In addition, while workload leveling has been studied extensively in construction, aeronautics, and manufacturing contexts, its transfer to ship refit and maintenance planning remains limited. Existing enterprise tools used in shipyards provide essential project tracking but offer little support for advanced optimization beyond workload smoothing. As a result, planners often rely on manual adjustments or heuristics that struggle to account for the complexity of specialized trades, constrained dry-dock windows, and long planning horizons typical of ship refit projects.

To address these limitations, this study introduces a novel RCCP model that enforces a single-peak workload profile as a structural constraint for resource leveling. The goal of imposing these constraints is to implicitly prevent excessive ramp-downs, while minimizing hiring and firing cycles, reducing training costs, and improving workforce morale. This structure is particularly valuable in contexts such as construction, seasonal projects, or large-scale manufacturing campaigns. We assess the feasibility of this constraint and analyze its impact on project duration using medium-sized instances.

Building on the model of [102], we extend the formulation to allow greater flexibility in execution intensities and examine the effects of this enhancement in our context. Our approach lies between the rigidity of identical intensities and the complete independence of resources : we allow intensities to differ but maintain a proportional relationship when technical sequencing requires it, for example ensuring that electricians progress at least as quickly as painters to respect operational constraints and preserve task sequencing. Additionally, we explore a bi-objective case that jointly minimizes project duration and inter-period workload variations, illustrating the trade-offs through Pareto front analyses on selected cases.

Table 6.1 Summary of resource leveling studies by method type, objective function, problem type, and application context

Article	Method Type	Objective Function	Problem Type	Application Context
[70]	MIP	Minimize absolute deviation from desired resource usage	Linear schedules (CPM with resource leveling)	Construction (highways)
[96]	Heuristic	Minimize variance (sum of squares)	RCPSP with resource leveling	Construction
[53]	GA	Minimize duration and balance workload	CPM with resource leveling	Project networks
[10]	Heuristic	Minimize sum of squares, absolute deviations, peak	RCPSP with resource leveling	Make-to-order production
[40, 107]	GA	Minimize rehire and idle days	CPM with resource leveling	Construction
[45]	MIP	Minimize deviation from ideal profile	RCPSP with resource leveling	Project scheduling
[69]	GA	Minimize variance of fuzzy workloads	RCPSP with uncertainty including resource leveling	Maintenance (helicopter)
[15, 16]	MIP	Minimize extra and adjustment costs with variable intensities and durations	RCPSP with variable execution intensity including resource leveling	Project scheduling / Production
[102, 103]	MIP + Benders	Minimize external resource cost	Aggregated RCPSP with variable durations including resource leveling	Tactical planning
[37]	GA	Compare leveling objective functions	RCPSP with resource leveling	Construction
[20]	MIP + CP + LNS	Minimize project cost with multimodal constraints	Multimode RCPSP with resource leveling	Aeronautical assembly

6.3 Proposed Resolution Method

This section outlines the resolution approach adopted in this study. It begins with a formal definition of the RCCP problem, followed by the presentation of the MIP formulations developed to model a more flexible variant of the classical RCCP. We then provide a theoretical comparison between the formulation described in the literature and the one proposed here, particularly in the context of resource leveling. Finally, we describe the overall solution strategy combining to solve the resulting model.

6.3.1 Problem Description

We address a tactical planning problem in which a project must be executed over a planning horizon H, divided into discrete time periods P, each of duration D_p (typically measured in weeks). The project consists of a set of WPs, denoted by I, where each WP $i \in I$ requires a workload Q_{ri} from each resource group $r \in R$, expressed in man-hours. The total workload for WP i is thus given by $Q_i = \sum_{r \in R} Q_{ri}$. Each WP must comply with predefined execution intensity constraints : a minimum $Q_i^{\min}$ and a maximum $Q_i^{\max}$ workload per week, both expressed in man-hours. This induces a minimum duration m_i and a maximum duration u_i.

For each resource group r and time period p, a limited capacity K_{rp} is available. Precedence relations among WPs are captured by a set of directed arcs $E \subseteq I \times I$, where $(i, j) \in E$ indicates that WP i must be completed before WP j can start.

To reduce the number of variables and constraints of the formulation, additional parameters are used. ES_i and LS_i denote, respectively, the *earliest* and *latest start times* of WP i, while EF_i and LF_i represent its *earliest* and *latest finish times*. For each WP i, the sets Ps_i and Pf_i respectively indicate the feasible periods for its start and completion. A start may occur either within a period or precisely at its beginning, while a finish may occur within a period or at its end.

Let $\pi(t)$ represent the index of the period containing time t. A time point t is considered to belong to period p if it occurs strictly within p or exactly at its end. Formally : $\pi(t) = \min \{p \mid \sum_{k=1}^{p} D_k \geq t\}, \quad t \in H$.

Accordingly, a period p belongs to Ps_i if $p \in \{\pi(ES_i + 1), \ldots, \pi(LS_i + 1)\}$, and to Pf_i if $p \in \{\pi(EF_i), \ldots, \pi(LF_i)\}$.

Table 6.2 presents the decision variables of the RCCP MIP model.

Table 6.2 Decision variables in the RCCP model

Variable	**Description**
x_{ip}	Fraction of the total workload Q_i of WP i during period p (between 0 and 1)
ts_i	Start time of WP i
tf_i	Finish time of WP i
s_{ip}	Remaining part of period p after the start of WP i
f_{ip}	Remaining part of period p after the end of WP i
d_{ip}	Duration of WP i in period p
zs_{ip}	Binary variable : 1 if WP i starts in or before the beginning of period p, 0 otherwise
zf_{ip}	Binary variable : 1 if WP i finishes in or before the end of period p, 0 otherwise
v_{rp}	Quantity of resource r used in period p
$C_{\max}$	Project completion time

6.3.2 Formulations Overview

In this section, we first describe the classical RCCP model, then adapt it to the flexible variant. We provide a theoretical comparison with formulations proposed in the literature and finally introduce a new model that considers unimodal resource profiles.

6.3.2.1 Classic RCCP Model : C-RCCP

The formulation presented in the *C-RCCP* model addresses the resource-leveling problem as its objective function. The model is based on step variables and assumes a uniform resource intensity across all resources within each time period. This assumption has been widely adopted in most models proposed in the literature.

$$s_{ip} \leq D_p \cdot zs_{ip}, \qquad \forall i \in I,\ p \in P, \tag{1}$$
$$s_{ip} \geq D_p \cdot zs_{ip-1}, \qquad \forall i \in I,\ p \in P, \tag{2}$$
$$f_{ip} \leq D_p \cdot zf_{ip}, \qquad \forall i \in I,\ p \in P, \tag{3}$$
$$f_{ip} \geq D_p \cdot zf_{ip-1}, \qquad \forall i \in I,\ p \in P, \tag{4}$$
$$d_{ip} = s_{ip} - f_{ip}, \qquad \forall i \in I,\ p \in P, \tag{5}$$
$$s_{jp} \leq f_{ip}, \qquad \forall (i,j) \in E,\ p \in P, \tag{6}$$
$$zs_{jp} \leq zf_{ip}, \qquad \forall (i,j) \in E,\ p \in P, \tag{7}$$
$$zs_{i,\pi(ES_i+1)-1} = 0, \qquad \forall i \in I, \tag{8}$$
$$zs_{i,\pi(LS_i+1)} = 1, \qquad \forall i \in I, \tag{9}$$
$$zf_{i,\pi(EF_i)-1} = 0, \qquad \forall i \in I, \tag{10}$$
$$zf_{i,\pi(LF_i)} = 1, \qquad \forall i \in I, \tag{11}$$
$$D_p + D_{p+1} - (f_{ip} + f_{ip+1}) \geq (D_p + \epsilon) \times (1 - zf_{ip}) \qquad \forall i \in I,\ p \in P \tag{12}$$
$$x_{ip} \cdot \sum_{r \in R} Q_{ri} \leq Q_i^{\max} \cdot d_{ip}, \qquad \forall i \in I,\ p \in P, \tag{13}$$
$$x_{ip} \cdot \sum_{r \in R} Q_{ri} \geq Q_i^{\min} \cdot d_{ip}, \qquad \forall i \in I,\ p \in P, \tag{14}$$
$$\sum_{p \in P} x_{ip} = 1, \qquad \forall i \in I, \tag{15}$$
$$v_{rp} = \sum_{i \in I} x_{ip} \cdot Q_{ri}, \qquad \forall r \in R,\ p \in P, \tag{16}$$
$$v_{rp} \leq K_{rp}, \qquad \forall r \in R,\ p \in P, \tag{17}$$

Constraints (1) and (2) ensure that $s_{ip} \in]0, D_p]$ when the WP is executed during period p, and $s_{ip} = 0$ otherwise. The same logic applies to the computation of f_{ip}, as enforced by constraints (3) and (4). Constraints (5) state that if $s_{ip} = 0$, then $d_{ip} = 0$. Conversely, when $s_{ip} > 0$, the variable d_{ip} becomes active and remains so until f_{ip} is activated. Constraints (6) are the continuous-time disaggregated precedence constraints (one per period), as opposed to an aggregated version that sums across periods. This formulation is stronger than the corresponding aggregated version. These constraints can be strengthened by constraints (7). Constraints (8) to (11) are optional tightening constraints designed to eliminate unnecessary variables. They relate to start and end times occurring strictly before or after the earliest and latest feasible periods, respectively. Constraints (12) are introduced to break symmetry when the end date is positioned at the end of a period p. In this case, the variable zf_{ip} could take either 0 or 1. The constraints impose $zf_{ip} = 1$ [83]. Constraints (13) to (17) govern resource allocation and capacity limitations. Specifically, constraints (13) and (14) ensure that the workload assigned to each WP remains within the specified minimum and maximum bounds for each period. Constraints (15) guarantee that the entire workload of each WP is fully distributed throughout the planning horizon. Constraints (16) then compute v_{rp}, which corresponds to the total workload assigned to each resource r in every period p. Meanwhile, constraints (17) ensure that the usage of resources does not exceed their respective capacities.

This formulation assumes a uniform intensity across all resources, which simplifies the modeling of workload distribution but may be an overly restrictive assumption.

6.3.2.2 Flexible RCCP : F-RCCP

This extended formulation is inspired by the work of [103], who introduced complete independence among resource intensities to improve external cost minimization. In this variant, the intensity variable x_{ip} is replaced by ξ_{irp}, which denotes the fraction of the total workload Q_i of WP i assigned to resource r during period p (with values between 0 and 1). Two alternative formulations are presented for this variant.

Formulation F1-RCCP [103] propose to manage the constraints that link the binary variables with the continuous duration variables as follows :

$$zs_{ip+1} \geq zs_{ip}, \qquad \forall i \in I,\ p \in P, \tag{1F}$$

$$zf_{ip+1} \geq zf_{ip}, \qquad \forall i \in I,\ p \in P, \tag{2F}$$

$$d_{ip} \leq D_p \cdot (zs_{ip} - zf_{ip-1}), \qquad \forall i \in I,\ p \in P, \tag{3F}$$

$$d_{ip} \geq D_p \cdot (zs_{ip} + zs_{ip-1} - 1 - zf_{ip-1} - zf_{ip}), \qquad \forall i \in I,\ p \in P, \tag{4F}$$

$$d_{jp} \leq D_p - d_{ip}, \qquad \forall (i,j) \in E,\ p \in P, \tag{5F}$$

$$zs_{ip} \geq zf_{i,p-1}, \qquad \forall i \in I,\ p \in P, \tag{6F}$$

$$\xi_{irp} \cdot Q_{ri} \leq Q_i^{\max} \cdot d_{ip}, \qquad \forall i \in I,\ \forall r \in R,\ \forall p \in P, \tag{13'}$$

$$\xi_{irp} \cdot Q_{ri} \geq Q_i^{\min} \cdot d_{ip}, \qquad \forall i \in I,\ \forall r \in R,\ \forall p \in P, \tag{14'}$$

$$\sum_{p \in P} \xi_{irp} = 1, \qquad \forall i \in I,\ \forall r \in R, \tag{15'}$$

$$v_{rp} = \sum_{i \in I} \xi_{irp} \cdot Q_{ri}, \qquad \forall r \in R,\ \forall p \in P, \tag{16'}$$

Constraints (7), (8), (9), (10), (11), (12) and (17)

Constraints (1F) and (2F) enforce that the associated binary variables are non-decreasing over time, effectively modeling stepwise activation. Constraints (3F) and (4F) are used to bound the duration variable d_{ip}.

If the WP starts before period p (e.g., in $p-1$) and finishes after $p+1$, then Constraint (4F) ensures that the duration during period p is equal to D_p. If the WP starts and finishes entirely before p, or entirely after p, then Constraint (3F) forces the duration in p to be zero. In all other cases, the duration is bounded between 0 and D_p.

Precedence constraints are expressed at two levels : across periods (constraints (7)) and within connecting periods (constraints (5F)). A successor and its predecessor may occur within the same period only if that period simultaneously represents the final period of the predecessor and the initial period of the successor.

Regarding capacity requirements, constraints (13)–(16) are replaced by their refined versions, namely (13')–(16'). The key distinction is that, in the latter, each resource is associated with a specific intensity for each WP and each period, allowing for a more detailed and flexible representation of resource consumption over time. Constraints (17) remain unchanged. It is worth noting that constraints (8)–(12) are not derived from or inspired by the original formulation of [103], but were introduced to enhance this model and to allow for a fair comparison with our proposed formulation.

In this formulation, apart from the duration variable, no additional continuous-time variables are required. However, this model is particularly well-suited for objectives focusing on cost minimization. In contrast, objectives that require the precise start and/or end times of each WP would necessitate the introduction of additional variables, and then additional constraints.

Formulation F2-RCCP The F2-RCCP formulation extends the **C-RCCP** formulation, with the original capacity constraints (13)–(16) replaced by their more flexible counterparts (13')–(16'). All other constraints remain unchanged.

This formulation supports a broader range of objective functions and constraints, including those that require precise timing or sequencing of WPs. Another advantage over the F2-RCCP formulation is that it provides a theoretically tighter linear programming (LP) relaxation.

Proposition 8. *The continuous relaxation of formulation F2-RCCP is stronger than that of F1-RCCP.*

Proof. Let S_{1p} and S_{2p} be the LP relaxed feasible sets of $F1\text{-}RCCP$ and $F2\text{-}RCCP$, respectively. First let us proof that constraints (1)–(6) of $F2\text{-}RCCP$ imply constraints (1F)–(6F) of $F1\text{-}RCCP$:

1. (1F) follows from (1)–(2), and (2F) from (3)–(4).
2. (3F) follows from constraints (1), (4), and (5). Indeed, from these constraints, we have $d_{ip} = s_{ip} - f_{ip}$, $s_{ip} \leq D_p \times zs_{ip}$, and $-f_{ip} \leq -D_p.zf_{ip-1}$, which yields $d_{ip} \leq D_p(zs_{ip} - zf_{i,p-1})$.
3. (4F) follows from (2), (3), and (5). We have $s_{ip} \geq D_p.zs_{i,p-1}$, $-f_{ip} \geq -D_p.zf_{i,p}$, leading to $d_{i,p} \geq D_p.(zs_{i,p-1} - zf_{ip}) \geq D_p.(zs_{i,p} - zf_{ip-1} - 1 + (zs_{ip-1} - zf_{ip}))$. Indeed, $(zs_{ip} - zf_{ip-1} - 1) \leq 0$.
4. (5F) follows from (5) and (6). Indeed, $d_{ip} + d_{jp} = (s_{ip} - f_{ip}) + (s_{jp} - f_{jp}) = (s_{jp} - f_{ip}) + (s_{ip} - f_{jp}) \leq 0 + D_p \leq D_p$.
5. (6F) follows from (1), (3) and (5). We have : $D_p.zs_{ip} \geq s_{ip}$, $s_{ip} \geq f_{ip}$, $f_{ip} \geq D_p.zf_{ip-1} \implies D_p.zs_{ip} \geq D_p.zf_{ip-1} \implies zs_{ip} \geq zf_{ip-1}$

To see that the converse is false, consider an instance with one WP, three periods of lengths $(1, 1, 1)$, $Q_1^{\min} = 0$, one resource with $Q_{1,1} = 0.5$, $Q_1^{\max} = 10$, and unit capacity. Let us consider the following example (Table 6.3) :

TABLE 6.3 Relaxed solution feasible for $F1$-$RCCP$ but infeasible for $F2$-$RCCP$

WP_i	$zs_{i,1}$	$zs_{i,2}$	$zs_{i,3}$	$zf_{i,1}$	$zf_{i,2}$	$zf_{i,3}$	$\xi_{i,1,1}$	$\xi_{i,1,2}$	$\xi_{i,1,3}$
1	0.5	0.5	0.5	$\frac{1}{5}$	$\frac{2}{5}$	$\frac{2}{5}$	0	1	0

This partial solution can be completed by setting $d_{11} = 0$ and $d_{13} = 0.1$, both of which satisfy the bounds in the two formulations.

For $p = 2$: using the $F2$-$RCCP$ formulation, we obtain $s_{12} = 0.5$, and $0.2 \leq f_{12} \leq 0.4$, which implies $0.1 \leq d_{12} \leq 0.3$.

However, using the $F1$-$RCCP$ formulation, we obtain $-0.6 \leq d_{12} \leq 0.3$, which simplifies to $0 \leq d_{12} \leq 0.3$. By fixing $d_{12} = 0.05$, this value satisfies the bounds of d_{12} in $F1$-$RCCP$. Additionally, it satisfies constraints (13), since

$$\xi_{112} = 1 \implies 1 \times 0.5 \leq 10 \times 0.05.$$

For $F2$-$RCCP$, $d_{12} = 0.05$ does *not* satisfy the bounds of d_{12}, which are $[0.1, 0.3]$.

Hence, this solution satisfies $F1$-$RCCP$ but violates $F2$-$RCCP$. Therefore, we have

$$S_{2p} \subset S_{1p}.$$

$\square$

6.3.2.3 Flexible RCCP with Dependent Intensities : FD-RCCP

In practice, resource intensities are not always fully independent. Some resources operate in tandem and exhibit correlated usage patterns. To capture this behavior, we extend the model by introducing resource coupling constraints that account for partial dependency between certain resources.

Let $R_1 \subseteq R$ denote the subset of dependent resources. For all $i \in I$, $p \in P$, and $r_1, r_2 \in R_1$, the following proportionality constraints are added :

$$\xi_{ir_2p} \geq \alpha_{r_1r_2} \cdot \xi_{ir_1p}, \qquad \forall i \in I,\ \forall p \in P,\ \forall r_1, r_2 \in R_1, r_1 < r_2 \tag{D1}$$

$$\xi_{ir_2p} \leq \beta_{r_1r_2} \cdot \xi_{ir_1p}, \qquad \forall i \in I,\ \forall p \in P,\ \forall r_1, r_2 \in R_1, r_1 < r_2 \tag{D2}$$

Constraints (D1) and (D2) impose lower and upper proportionality bounds, respectively, between the intensities of dependent resources. These constraints ensure that the intensity of any resource $r_2 \in R_1$ used for a given WP and period remains within a factor of $\alpha_{r_1 r_2}$ and $\beta_{r_1 r_2}$ relative to another resource $r_1 \in R_1$ $(r_1 < r_2)$

6.3.2.4 Flexible RCCP with Dependent Intensities and Single-Peak Resource Usage Constraint : FDS-RCCP

To enforce single-peak workload profiles, we introduce the following variables :

- $u_{rp} \in \{0, 1\}$: a binary variable indicating whether period p corresponds to or follows the peak usage of resource r.
- $y^c_{rp} = v_{rp}$ before or at the peak period ; this variable helps trace the increasing part of the resource usage profile.
- $y^d_{rp} = v_{rp}$ after the peak period ; this variable helps trace the decreasing part of the resource usage profile.

The FDS-RCCP model is presented below, where the unimodal structure is enforced through constraints (s1)–(s5), (s6a) and (s6b).

$$v_{rp} = y^c_{rp} + y^d_{rp} - K_{rp}, \qquad \forall r \in R,\ \forall p \in P \tag{s1}$$

$$y^c_{rp+1} \geq y^c_{rp}, \qquad \forall r \in R,\ \forall p \in P \tag{s2}$$

$$y^d_{rp} \geq y^d_{rp+1}, \qquad \forall r \in R,\ \forall p \in P \tag{s3}$$

$$y^c_{rp} \geq K_{rp} \cdot u_{rp}, \qquad \forall r \in R,\ \forall p \in P \tag{s4}$$

$$y^d_{rp} \geq K_{rp} \cdot (1 - u_{rp}), \qquad \forall r \in R,\ \forall p \in P \tag{s5}$$

$$y^c_{r|P|} = K_{r|P|}, \qquad \forall r \in R \tag{s6a}$$

$$y^d_{r1} = K_{r1}, \qquad \forall r \in R \tag{s6b}$$

Constraints (1)–(12), (13')–(16'), (17), (D1) and (D2)

Constraint (s1) links the actual resource usage v_{rp} to the combination of increasing and decreasing components, ensuring consistency across the profile. Constraints (s2) and (s3) enforce monotonicity : y^c_{rp} must be non-decreasing over time (representing the buildup), while y^d_{rp} must be non-increasing (representing the ramp-down). Constraints (s4) and (s5) use the binary variable u_{rp} as a switch to divide the planning horizon into two phases : before the peak ($u_{rp} = 0$) and at or after the peak ($u_{rp} = 1$). This mechanism forces the workload profile to have only one peak. Finally, constraints (s6a) and (s6b) initialize the variables. The variable y^c_{rp} increases until it reaches the upper bound of the peak value K_{rp}, while y^d_{rp} starts from this same bound and decreases thereafter.

6.3.3 Objective Functions

We consider three alternative objective functions in this study, each capturing a different optimization goal within the RCCP context :

1. **Makespan Minimization using the model FDS-RCCP**

$$Z = \min C_{\max}$$

   This objective aims to minimize the project duration while enforcing the single-peak resource usage constraint.

2. **Total Variation Minimization using the model FDS-RCCP, FD-RCCP and C-RCCP**

$$Z = \min \sum_{r \in R} \sum_{p \in P} |v_{rp} - v_{rp-1}|$$

   This objective function minimizes the fluctuation in resource usage between consecutive periods. It can be linearized by introducing an auxiliary variable δ_{rp}, representing the absolute variation in total resource usage between two consecutive periods for each resource r. The linearized formulation is as follows :

$$Z = \min \sum_{r \in R} \sum_{p \in P} \delta_{rp} \tag{O2}$$

$$\delta_{rp} \geq v_{rp} - v_{rp-1}, \qquad \forall r \in R,\ p \in P \tag{L1}$$

$$\delta_{rp} \geq -v_{rp} + v_{rp-1}, \qquad \forall r \in R,\ p \in P \tag{L2}$$

3. **Bi-objective Optimization using the model FD-RCCP**

$$Z_1 = \min C_{\max},\ Z_2 = \min \sum_{r \in R} \sum_{p \in P} \delta_{rp}$$

In this setting, both project duration and workload stability are simultaneously optimized. The first objective aims to minimize the makespan $C_{\max}$, while the second minimizes the total variation in resource usage across consecutive periods.

6.3.4 Iterative Re-Optimization Procedure (IRP) : Principle of the Algorithm

The proposed algorithm first solves node 0 of the original problem using a solver to obtain $\underline{b^*}$, the best lower bound found so far for the duration-minimization objective. Then, the end dates of the WPs are bounded by $\underline{b^*}$, and we verify whether a feasible solution exists ; otherwise, we prove that no solution is attainable for this bound. In cases where no solution is found, the lower bound is incremented by 1 unit of time, and the problem is re-optimized. The global optimization stops when the solver finds a feasible solution. In each sub-optimization step, additional constraints can be incorporated by exploiting the information derived from the current lower bound.

Note that this procedure guarantees an optimal solution under the assumption that $C_{\max}$ is an integer. This assumption, also adopted by [83], is motivated by the fact that allowing a continuous makespan may lead to a long tailing effect without any practical benefit, since in practice the value is rounded to the nearest integer.

6.3.4.1 Improving the Optimization Step

We can improve each optimization resolution by leveraging the information from the lower bound. Suppose we have a lower bound $\underline{b^*}$. We can add the constraints below to enhance the resolution process.

$$tf_i \leq \underline{b^*} \cdot zf_{i,\underline{b^*}}, \quad \forall i \in \{1, \dots, I\}. \quad (C17)$$

This constraint enforces $zf_{i,\underline{b^*}} = 1$ whenever a feasible solution exists, thereby eliminating equivalent solutions for these variables without the need to include constraints (12).

6.3.4.2 Improving IRP

The eventual limitation of the Iterative Re-Optimization Procedure (IRP) lies in the time required to prove the infeasibility of an optimization problem before reaching a solution. This issue becomes particularly significant in instances with large planning horizons. In such cases, the procedure may fail to provide feasible solutions within the allowed computational time. In such cases, we propose to adapt the approach in order to ensure a high-quality solution while maintaining sufficiently tight lower bounds, with the objective of efficiently reducing the integrality gap.

The proposed method rapidly increases the lower bound until the optimization problem becomes moderately challenging. If the solver exceeds a reasonable computation time without identifying a solution, the lower bound is incremented again and the process is repeated. Once a feasible solution is found, the algorithm backtracks by slightly decreasing the lower bound and re-optimizes, using the previously obtained solution as a warm start.

The detailed algorithm of the improved iterative re-optimization procedure (IIRP) is presented in 6.7.1.

6.3.5 Construction Heuristic

To reduce the initial project horizon, we propose a Schedule Generation Scheme (SGS) adapted to the RCCP, derived from the RCPSP. This approach generates solutions from a priority list by starting with an empty schedule and incrementally constructing a feasible partial schedule. Here, we employ the Serial Schedule Generation Scheme (SSGS), which schedules each WP at the earliest possible start date according to the priority list while respecting precedence and resource constraints. This method builds the schedule sequentially, one WP at a time, thus ensuring feasibility as each new WP is added.

Serial SGS

Let σ be the permutation that generates the priority list, and let g denote the current step of the algorithm. We define :

- S_g : The set of WPs scheduled up to step g.
- F_g : The set of the finish times FT_j of all WPs in S_g.
- D_g : The set of WPs eligible to be scheduled at step g. That is, WPs whose all predecessors belong to S_{g-1}.

In the context of RCCP, we require both start and finish times, since the duration of each WP may vary. Additionally, let $\hat{R}_r(p)$ denote the remaining resource capacity for resource r at the aggregated period p and RW_{ir} the remaining workload to be executed for WP i for resource r. At each step, a fraction f of the remaining workload is allocated while ensuring that the resource limits $\hat{R}_r(p)$ are not exceeded. Note that this procedure assumes a uniform variable intensity across resources ; therefore, the resulting planning represents an upper bound for the C-RCCP and F2-RCCP formulations.

Algorithm 2: Serial Schedule Generation Scheme for RCCP

Initialization : $F_0 = 0,\ S_0 = \{0\} \quad (FT_0 \leftarrow 0)$

For $g = 1$ to n **do**

Calculate D_g (WPs whose all predecessors are in S_{g-1})

Select one $j \in D_g$

$ES_j \leftarrow \max_{h \in P_j}\{FT_h\}, \quad tf_j \leftarrow ES_j$

while $\max_r\{RW_{jr}\} > 0$ **do**

  if capacities are not available at tf_j (i.e., some $\hat{R}_r(\pi(tf_j)) = 0$) **then** $tf_j \leftarrow tf_j + 1$; **continue**

  Generate a random value v

  if $v < 0.1$ **then**

    Select a fraction f of a demand to be executed at date tf_j

  else

    Select the maximum fraction f feasible at date tf_j respecting capacity.

  end if

  $tf_j \leftarrow tf_j + m_j \times f$ $\{m_j$ is the minimum processing time of WP $j\}$

  Update RW_{jr} for all required r and update $\hat{R}_r(\pi(tf_j))$.

end while

$FT_j \leftarrow tf_j$

$S_g \leftarrow S_{g-1} \cup \{j\}, \quad F_g \leftarrow F_{g-1} \cup \{FT_j\}$

Update D_{g+1} (add any successor whose predecessors are all in S_g)

$$C_{\max} = \max_{j \in S_n} FT_j$$

The initialization assigns the dummy source WP a completion time of 0 and includes it in the partial schedule. At the beginning of each step g, one WP j is selected from the decision set. The finish time of j is determined in continuous time through the following steps :

1. Calculate its earliest precedence-feasible finish time, EF_j.
2. Temporarily update the schedule as fractions of the workload are allocated to WP j until its entire workload is assigned.

However, there are several differences to consider when adapting these methods to the RCCP context :

1. **Execution Intensity** : In RCCP, WPs have variable execution intensities, meaning their resource consumption and progress rates can fluctuate over time. Consequently, planning WPs involves determining both their start times and execution intensities across different periods to calculate their finish times accurately.
2. **Mid-Period Start Times** : Unlike RCPSP, where WPs begin at the start of predefined periods, RCCP allows WPs to start at any point within a period.

6.3.6 Balancing Duration and Stability : A Bi-objective Strategy

This section explores a strategy to evaluate the trade-off between minimizing project duration and minimizing variations in resource usage. We start with the minimal project duration obtained without this structural constraint. We then progressively increase the allowed project horizon, enabling the identification of solutions that comply with the unimodal workload profile by minimizing total variations, up to the duration at which a unimodal shape becomes achievable. Once this shape is obtained, variation minimization is no longer applied.

This stepwise adjustment leads to a set of non-dominated solutions. The resulting Pareto frontier can help practitioners balance time efficiency with implementation stability.

To handle this bi-objective problem, we apply the ϵ-constraint method, a standard scalarization technique in multi-objective optimization. The idea is to optimize one objective ; here, the variation cost ; while treating the project duration as a constraint parameterized by ϵ. In our case, $\epsilon = 1$.

By successively increasing ϵ from the minimal feasible duration (without enforcing a unimodal profile) to the duration where the unimodal behavior becomes possible, this process leads to a set of Pareto solutions that highlight the compromise between project duration and smooth resource allocation.

6.4 Experimental Setup and Instance Generation

This section presents the computational experiments designed to evaluate the proposed RCCP formulations. We describe the instance generation proposed to test our models, outline the experimental setup and performance indicators, and then present the main comparative analyses, including mono and bi-objective scenarios.

6.4.1 Instance Generation

In the RCPSP problem and its variants, traditional indicators are used to reflect the difficulty of instances. These indicators are typically classified into four categories : precedence-oriented, time-oriented, resource-oriented, and hybrid [63]. The most commonly used indicators are :

1. **NC (Network Complexity)** : it represents the average number of precedence arcs per activity, assuming that E contains no redundant arcs. Generally, the difficulty of RCPSP instances decreases as NC increases.
2. **RF (Resource Factor)** : it is defined as the average number of resources required per activity. Typically, the complexity of an instance increases as RF increases.
3. **RS (Resource Strength)** : This metric represents the level of resource availability in an instance. A higher RS indicates greater resource availability, while a lower RS reflects tighter resource constraints. The RS metric considers peak demands under an earliest-start schedule, making it particularly suitable for evaluating instances with a makespan objective.
4. **MRS (Modified Resource Strength)** : Similar to RS, this metric is designed for the Multi-Skill Project Scheduling Problem (MSPSP) [36]. In this context, a multi-skill problem becomes equivalent to the RCPSP when all resources have all the required skills. Studies have shown that as MRS increases, the complexity of solving the instance also increases. This metric offers greater flexibility for addressing objectives beyond makespan.

In the RCCP problem, many authors have proposed instances in the context of Time-driven variant [32,39]. However, there are no specific instances designed for Resource-driven RCCP. [60] adapted the PSPLIB instances from RCPSP [62] by considering the duration of each task as the minimal duration of each task. In this context, a task represents a WP in our case.

Building on the previous elements, we adapt the instance generation methods proposed by [36], for multi-skill RCPSP, and [62], for RCPSP, to develop our own generation approach. This method is designed to create theoretical RCCP instances, which are not necessarily tailored exclusively to the makespan objective, given that MRS is a more general parameter than RS. This approach enables a comparative analysis of the formulations presented in Section 4.3. The main difference between our instances and those typically found in the RCCP literature lies in the requirement for resource capacity to remain relatively uniform across periods, allowing a unimodal resource profile to be traced.

6.4.1.1 Network Generation

To create the graph, we follow the arc generation method of [3]. Contrary to [62], this procedure guarantees a network with the desired parameterization of **NC** without the need to repeat the entire generation process. In addition to the possibility of adding an arc, it is also characterized by the possibility of removing an arc during the procedure of generating the network (when node i still has at least one successor and node j retains at least one predecessor), contrary to [62].

6.4.1.2 Parameters to Generate the Required Workload

1. **NC** : The network is generated using the parameters $NC = \{1.4, 1.7, 2.1\}$. $NC = 1.4$ results in highly cumulative instances due to the relatively low number of precedence relations (when the MRS parameter is quite large).
2. **Minimum Durations of WPs** : As the instances of [39], minimum durations of the WPs are randomly generated in the interval $[1, 5]$.
3. **RF (Resource Factor)** : To ensure variability in resource usage, RF is generated in the set {0.25, 0.5, 0.75}, while the number of resource types is set to 3, 5, and 10, respectively.
4. **Capacity of each resource type** : The capacity is varied from 4 to 20. The greater the size of the WP and the RF factor, the higher the capacity assigned.
5. **MRS (Modified Resource Strength)** : The MRS measures the demand intensity relative to the available resources. It is defined as :

$$MRS = \frac{\sum_{r\in R} K_r}{\sum_{i\in I}\sum_{r\in R} Q_{ri}}$$

[32] propose, in their instance generation, to represent the first periods of the planning horizon in detail, while aggregating later periods into longer time units. This approach reflects the fact that, in long-term projects, near-term activities can be scheduled with greater accuracy,

whereas uncertainty increases further in the horizon, making precise planning less relevant. Following this idea, our instances model the first four periods in detail, each with a length of 1, and aggregate the remaining periods into blocks of four units to capture overall feasibility without unnecessary precision.

The algorithm for generating the instances is adapted from the MSPSP-InstLib library [1] developed by K. D. Young (2017). Five instances are generated for each parameter setting, resulting in a total of 405 instances.

6.4.2 Experimental Setup

All computational experiments were conducted on a high-performance computing grid equipped with dual Intel Xeon Gold 6258R processors (2.70 GHz) and 512 GB of RAM. IBM ILOG CPLEX 22.1.1 was employed as the MIP solver throughout the study.

For single optimization scenarios, where the model was directly solved using CPLEX, a single thread was used with a time limit of 10 000 seconds. In contrast, for experiments involving multiple optimizations, such as iterative procedures or bi-objective formulations, the time limit was reduced to 1 000 seconds.

6.4.3 Performance Indicators

To evaluate model performance, several indicators were considered. The *solving time* refers to the duration, in seconds, required by the solver to find and validate an optimal solution or to reach the time limit. The *number of explored nodes* quantifies the extent of the branch-and-bound process, while the *number of dual simplex iterations* reflects the computational effort invested through dual simplex pivots. The *optimality gap* measures the relative difference between the best known solution and the best lower bound at termination. In bi-objective analyses, two specific indicators were used : the *Gap-dur* (expressed in %) represents the increase in makespan resulting from the enforcement of a single-peak constraint, and the *Gap-var* (in %) captures the corresponding reduction in inter-period workload variation. Additionally, in the bi-objective case, three comparative indicators were computed to assess the difference in variation between two solutions : the maximum deviation percentage, the minimum deviation percentage, and the mean deviation percentage, which represent the highest, lowest, and average variation gap observed, respectively.

The gap between two solutions is computed using the following formula :

1. `https://github.com/youngkd/MSPSP-InstLib`

$$\text{Gap}(\%) = \frac{v_1 - v_2}{v_1} \times 100$$

where v_1 and v_2 represent the values obtained when testing two different models.

6.5 Results and Analysis

The experiments were structured to address the following goals :

1. Compare the classical C-RCCP model to the FD-RCCP formulation, in terms of solving time, solution quality, and scalability.
2. Evaluate the impact of enforcing a single-peak workload profile constraint on both computational performance and project duration extension.
3. Study the trade-off between makespan minimization and workload variation reduction through a bi-objective analysis using the ε-constraint method.

For better readability, the suffix *RCCP* from model labels is omitted in the tables. Each approach is identified by its prefix (e.g., *FD*, *C*, etc.) followed by the corresponding objective function : d for duration minimization and v for workload variation minimization (e.g., C_d for minimizing duration and C_v for minimizing workload variation).

6.5.1 Minimizing Total Variation Between Periods : C-RCCP vs. FD-RCCP

This section investigates the impact of the FD-RCCP formulation, as a more flexible alternative to the classical C-RCCP, on computational performance and solution quality across various instance sizes. The aim is to determine whether the added modeling flexibility leads to notable changes in solver behavior across different instance sizes, and whether any observed improvements in solution quality justify the potential increase in computational cost. Without loss of generality, the planning horizon is fixed to the upper bound obtained from the SSGS heuristic.

The results in Tables 6.4 and 6.5 highlight a fundamental trade-off between computational effort and solution richness when comparing the classic C-RCCP model to the proposed FD-RCCP model, which incorporates resource-dependent intensities.

Table 6.4 reports the average performance of the C-RCCP and FD-RCCP (as introduced in Section 4.3) on multiple instance classes. The column **Nb-Wps** refers to the number of WPs, and **Nb-Res** indicates the number of available resources. **Models** lists the tested formulation. The column **Tps (s)** provides the average solving time in seconds per instance. **Nodes** indicates the average number of branch-and-bound nodes explored, and **Iterations** shows the number of dual simplex iterations performed. Finally, **Opt-gap (%)** represents the average optimality gap, including cases where optimality was proven.

Table 6.4 Comparison of computational performance metrics for FD-RCCP and C-RCCP across varying problem sizes

Nb-WPs	**Nb-Res**	**Models**	**Nodes**	**Iterations**	**Tps (s)**	**Opt-gap (%)**
20	3	FD_v	217.64	21 584.64	2.06	0.00
		C_v	190.51	7 439.82	0.41	0.00
	5	FD_v	101.31	6 811.22	0.71	0.00
		C_v	78.07	1 729.89	0.16	0.00
	10	FD_v	277.93	24 414.00	2.25	0.00
		C_v	121.02	3 177.24	0.27	0.00
40	3	FD_v	9 994.82	1 723 094.93	430.25	0.00
		C_v	1 863.42	285 218.40	32.96	0.00
	5	FD_v	2 995.44	392 142.67	54.64	0.00
		C_v	1 028.02	38 542.31	2.82	0.00
	10	FD_v	8 347.18	2 013 106.04	278.14	0.00
		C_v	2 427.84	110 479.82	8.26	0.00
60	3	FD_v	17 286.36	2 857 038.98	590.10	0.00
		C_v	1 507.67	130 360.20	15.68	0.00
	5	FD_v	20 270.09	5 474 124.40	966.03	0.03
		C_v	5 913.38	421 277.56	36.54	0.00
	10	FD_v	43 193.53	17 065 211.22	3 943.06	0.72
		C_v	49 002.02	3 473 406.96	371.32	0.00

From Table 6.4, we observe that the FD-RCCP model consistently requires significantly more computational time and effort compared to the classical C-RCCP formulation. For instance, in the largest tested case, with 60 WPs and 10 resources, the FD-RCCP model takes over 3 900 seconds and performs more than 17 million simplex iterations, while the C-RCCP model solves the same instance in only 371 seconds with approximately 3.47 million iterations. Across all problem sizes and resource configurations, the number of explored branch-and-bound nodes is systematically higher for FD-RCCP. This increase is a direct consequence of the additional modeling flexibility introduced by the disaggregated formulation. We note that for instance 60-10 (60 WPs and 10 resources), the number of nodes explored is lower than in the C-RCCP model, mainly because some instances stopped due to the time limit.

We also observe that the number of dual simplex iterations increases much more sharply than the number of explored nodes when comparing FD-RCCP to C-RCCP, particularly in instances involving 10 resources. This behavior suggests that the linear relaxations solved at each node in the branch-and-bound tree are more difficult in FD-RCCP due to the increased number of variables and constraints introduced. For instance, in the 60-10 configuration, the ratio between dual simplex iterations and explored nodes is markedly higher than in the C-RCCP formulation, highlighting the increased computational effort required per node.

Table 6.5 presents the variation gaps observed when comparing the solutions obtained using the C-RCCP and FD-RCCP models. The **Gap-max (%)** indicates the highest observed percentage reduction in inter-period workload variation, showcasing configurations where allowing flexible intensities in FD-RCCP significantly improves workload regularity. The **Gap-min (%)** reflects the smallest observed reduction, suggesting that in some cases, the use of flexible intensities has little to no impact compared to the unified-intensity approach of C-RCCP. Lastly, the **Gap-mean (%)** provides the average variation improvement across all solutions for each instance class, offering a global view of the benefits of using flexible, resource-dependent intensity modeling over a uniform workload structure.

Table 6.5 Analysis of workload variation gaps (%) between the FD-RCCP and C-RCCP models by number of WPs (Nb-Wps) and resources (Nb-Res).

Nb-Wps	Nb-Res	%gap-max	%gap-min	%gap-mean
20	3	46.78	0.00	11.19
	5	40.88	0.00	20.29
	10	92.22	13.13	35.16
40	3	17.55	0.00	4.23
	5	58.36	0.00	16.68
	10	80.52	10.96	30.41
60	3	7.09	0.00	1.91
	5	21.11	1.29	7.56
	10	47.20	9.23	22.46

Several trends emerge from Table 6.5, offering insights into the quality of solutions produced by the FD-RCCP model. The number of available resources plays a key role in the observed variation gaps. As the number of resources increases, both the maximum and average gaps tend to rise. For instance, in the configuration with 20 WPs and 10 resources, a maximum gap of 92.22% and an average gap of 35.16% are observed, compared to much smaller gaps for 3 resources. This suggests that when more resources are available, the FD-RCCP model can better leverage its flexibility to smooth out workload variations, unlike the classical model which imposes uniform intensity. Conversely, in configurations with fewer resources, the modeling flexibility has less impact due to more constrained scheduling possibilities.

From this comparison, we can deduce that the additional computational effort required by the FD-RCCP model is justified in scenarios where reducing inter-period workload fluctuations is essential. While the classical C-RCCP formulation remains an effective and efficient baseline ; especially suited for rapid feasibility analyses ; the FD-RCCP offers enhanced modeling flexibility that better captures real-world planning complexities.

6.5.2 Comparative Analysis : Direct vs. Iterative Re-optimization

We now evaluate the impact of enforcing a single peak workload profile on the overall project duration. To this end, we minimize the makespan while incorporating this structural constraint. As highlighted in the previous analysis (Section 6.5.1), the main challenge associated with the FD-RCCP model lies in its computational time. To address this limitation, we introduce the **IIRP** as a strategy to accelerate the solving process. Table 6.6 presents a comparative analysis between the direct resolution of the **FDS-RCCP** model and its resolution through the proposed **IIRP** approach. The time limit for each sub-optimization is set to 500 seconds. The comparison is based on performance metrics, including the number of explored nodes, the number of simplex iterations, the total solving time (in seconds), and the final optimality gap (in %).

Table 6.6 Comparison of computational performance between IIRP and FDS-RCCP across different instance sizes

Nb-Wps	Nb-Res	Models	Nodes	Iterations	Tps (s)	Opt-gap (%)
20	3	IIRP	4.02	2 956.61	19.15	0.00
		FDS$_d$	123.70	121 624.18	22.74	0.00
	5	IIRP	2.27	642.75	2.18	0.00
		FDS$_d$	27.75	10 923.41	2.41	0.00
	10	IIRP	191.76	74 635.34	30.24	0.00
		FDS$_d$	287.32	216 405.63	42.40	0.00
40	3	IIRP	37.58	30 853.42	12.67	0.00
		FDS$_d$	733.58	2 417 142.07	1 056.00	0.13
	5	IIRP	127.25	63 965.11	7.85	0.00
		FDS$_d$	244.73	215 592.45	69.50	0.00
	10	IIRP	358.89	162 628.61	63.75	0.00
		FDS$_d$	2 599.59	2 832 771.68	786.87	0.21
60	3	IIRP	29.69	31 847.18	15.85	0.00
		FDS$_d$	998.51	4 816 224.58	2 741.21	0.92
	5	IIRP	9.38	6 321.60	13.50	0.00
		FDS$_d$	68.09	112 183.20	75.95	0.00
	10	IIRP	352.38	261 271.76	67.20	0.00
		FDS$_d$	2 084.82	4 094 738.11	1 750.18	0.17

For smaller instance sizes (Nb-Wps = 20), the IIRP approach and the direct resolution of the FDS-RCCP model exhibit relatively comparable solving times. The IIRP method maintains a very low number of explored nodes, often fewer than 200, and achieves acceptable solving times regardless of the number of resources, even though the number of simplex iterations is notably lower with IIRP. This could suggest that, in some cases, when the linear relaxation is easy to solve but the presolve phase is time-consuming, solving a sequence of reduced problems may not always lead to notable speed gains.

In contrast, the direct resolution of the FDS-RCCP model becomes increasingly computationally demanding as the number of resources increases. This is evident in the growing number of simplex iterations, which can reach several million, and in solving times that may exceed 10 000 seconds in certain configurations.

As the problem size increases to 40 or 60 WPs, the performance gap between the two methods becomes significantly more pronounced. The FDS-RCCP model can require up to 4.8 million iterations and over 1 700 seconds of solving time for the largest instances. Additionally, some instances were not solved to optimality within the time limit. In contrast, the IIRP procedure finds the optimal solution across all tested configurations, demonstrating a significant improvement in both efficiency and solution quality as instance size increases.

Overall, the IIRP procedure proves to be both efficient and reliable. It significantly reduces the number of explored nodes and simplex iterations compared to the FDS-RCCP model. While the direct resolution of FDS-RCCP may be suitable for small instances, IIRP clearly outperforms it in larger configurations by offering better scalability without compromising solution quality.

Figure 6.1 illustrates the transformation of the workload profile of Resource 1 (instance of 20 WPs, 3 resources : 20-3-2-3) when a single peak constraint is applied. Initially, the resource usage fluctuates with multiple peaks and valleys across periods. After enforcing the unimodal constraint, the profile becomes smoother and exhibits a single prominent peak. This structure promotes a more regular and predictable workload distribution, which can facilitate better planning and reduce variability in resource allocation over time. However, in this case, enforcing the single peak constraint leads to an increase of 2 time units in the minimum project duration. Figure 6.2 highlights the impact of enforcing a single peak constraint on the workload profile

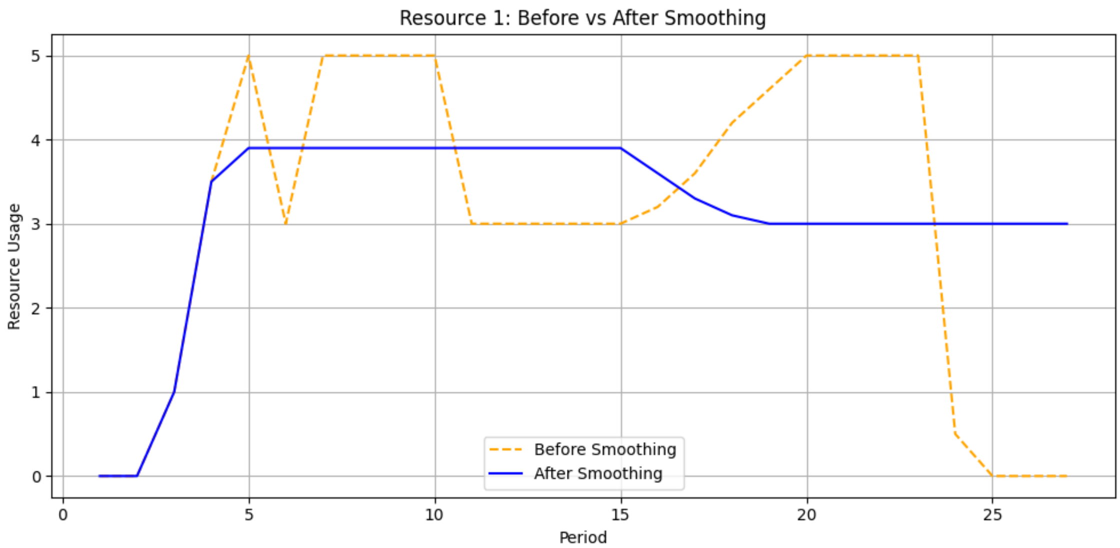

Figure 6.1 Workload profile of Resource 1 before and after enforcing a single-peak structure

of Resource 2 in instance 40-3-3-2 (an instance with 40 WPs and 3 resources). This comparison also illustrates how introducing a structural constraint such as unimodality can significantly reshape the resource usage profile. Without the single peak constraint (Approach A), the workload exhibits multiple sharp peaks, particularly around periods 6 and 11, with abrupt drops in between. In contrast, the unimodal profile (Approach C) results in fewer abrupt fluctuations and a smoother resource distribution, while notably preserving the same project duration. It is also worth noting that this unimodal profile aligns with the quality indicators proposed by [40], as it remains non-increasing after the peak, with no subsequent rise in resource utilization.

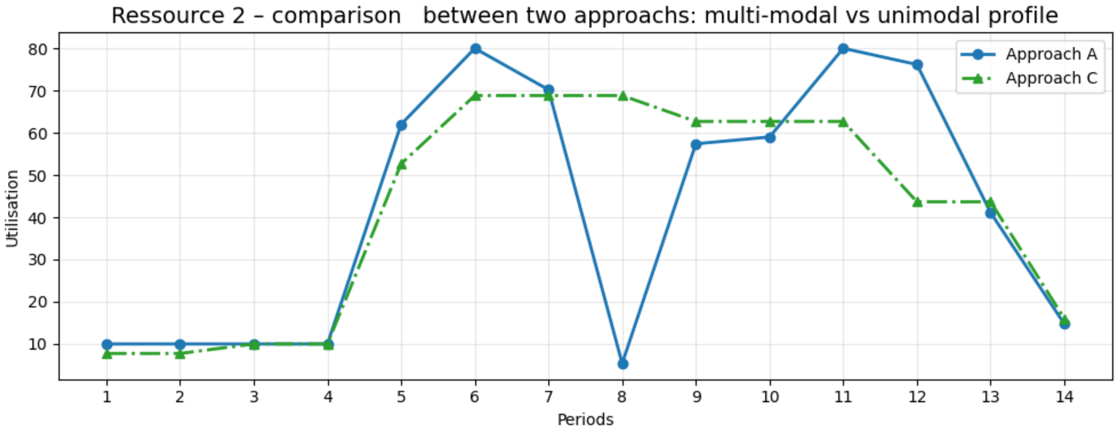

FIGURE 6.2 Comparison of workload profiles for Resource 2 between two approaches : Approach A (minimizing makespan without single peak constraint) and Approach C (minimizing makespan with single peak constraint)

6.5.3 Comparative Analysis : Multimodal vs. Unimodal RCCP

Now, we compare the computational performance of the RCCP model (FD-RCCP), which permits multiple workload peaks throughout the planning horizon, with its unimodal counterpart (FDS-RCCP). The latter enforces a single-peak structure by minimizing workload variation between consecutive periods. In this comparison, the planning horizon is fixed to the minimum project duration obtained under the FDS-RCCP configuration (Table 6.7).

Table 6.7 Performance comparison of FD-RCCP and FDS-RCCP models when minimizing the variation of workloads

Nb-Wps	Nb-Res	Models	Nodes	Iterations	Tps (s)	Opt-gap (%)
20	3	FD_v	241.78	22 870.97	5.12	0.00
		FDS_v	105.75	8 429.44	0.86	0.00
	5	FD_v	1 296.22	805 413.36	175.51	0.00
		FDS_v	49.05	2 432.39	0.29	0.00
	10	FD_v	16 739.58	3 808 061.11	461.03	0.18
		FDS_v	152.78	16 655.39	1.84	0.00
40	3	FD_v	3 104.72	571 385.08	171.93	0.00
		FDS_v	7 913.62	1 675 362.28	369.92	0.00
	5	FD_v	1 400.82	564 812.91	244.76	1.04
		FDS_v	1 355.77	173 573.95	17.02	0.00
	10	FD_v	7 777.13	2 163 945.56	479.94	0.90
		FDS_v	3 139.07	591 693.48	76.16	0.00
60	3	FD_v	3 002.90	600 237.29	99.03	0.00
		FDS_v	3 816.90	617 408.85	95.36	0.00
	5	FD_v	3 688.40	959 277.09	213.91	0.00
		FDS_v	3 934.00	173 573.95	183.78	0.00
	10	FD_v	29 341.80	9 752 616.58	2 509.42	0.36
		FDS_v	20 903.18	591 693.48	1 528.56	0.23

From a computational perspective, the unconstrained model, in which no peak is imposed on the workload profile, tends to be significantly more challenging to solve, particularly for large-sized instances. In contrast, the single-peak model generally requires fewer nodes and less computation time. This improvement results from the more structured nature of its feasible space, which guides the search process more efficiently. However, this structured space may also introduce rigidity, potentially limiting flexibility in instances where a less constrained workload profile would be more appropriate. Most instances solved under this constraint reach proven optimality, whereas the unconstrained workload model occasionally exhibits slightly higher optimality gaps.

In terms of solution quality, Table 6.8 presents the analysis of workload variation gaps (%), categorized by Nb-Wps and Nb-Res.

Table 6.8 Analysis of workload variation gaps (%) by number of WPs (Nb-Wps) and resources (Nb-Res)

Nb-Wps	Nb-Res	%gap-max	%gap-min	%gap-mean
20	3	38.56	0.00	3.72
	5	8.96	0.00	1.11
	10	4.60	0.00	0.36
40	3	36.35	0.00	8.05
	5	23.07	0.00	3.04
	10	4.22	0.00	0.45
60	3	98.07	0.00	12.49
	5	30.96	0.00	6.10
	10	40.11	0.00	5.72

The results highlight the model's sensitivity to increasing problem size and to a reduction in the number of available resources, with higher values of %gap-mean and %gap-max observed in larger instances. The analysis of gap ranges further underlines the differences between the two approaches. The unconstrained model generally exhibits greater inter-period stability, whereas the single-peak constrained model can occasionally produce extreme deviations. This occurs because enforcing a unimodal structure may prevent the model from exploiting multiple peaks that could otherwise reduce workload variations. Consequently, a rigid unimodal constraint can sometimes compromise workload balance compared to the optimal multimodal profile. Nevertheless, the single-peak model achieves a minimum gap of 0% in several instances, confirming that optimal solutions obtained with the FD-RCCP can still be reached under the unimodality constraint, depending on instance characteristics. Moreover, as the number of resources increases, the difference between the two approaches in terms of inter-period variation diminishes. This suggests that imposing a structured workload profile can, in some cases, reduce variation indicators and indirectly optimize resource variations, with the principal advantage of limiting the need for repeated hiring and layoffs.

We also note that the FD-RCCP model is highly sensitive to the horizon parameter. A larger planning horizon does not necessarily make the optimization of workload variations more difficult, as the effect depends on the instance characteristics and may even behave in the opposite way in some cases. When the horizon is relatively short, minimizing variation can still be challenging, leading to noticeable differences in solver behavior between FD-RCCP tests with horizons close to the makespan (Table 6.4) and those near the minimal feasible duration under the unimodal assumption (Table 6.7).

Figure 6.3 illustrates the utilization profile of Resource 1 across periods under four different optimization strategies (instance 40-3-2-3).

Approach A corresponds to the minimization of makespan using the *FD-RCCP* model, which does not impose any single peak constraint. Approach B applies the *FDS-RCCP* model with a single peak constraint while minimizing the inter-period variation. Approach C also uses the *FDS-RCCP* model but focuses on makespan minimization under the single peak constraint. Approach D minimizes inter-period variation using the *FD-RCCP* model, without enforcing the single peak constraint.

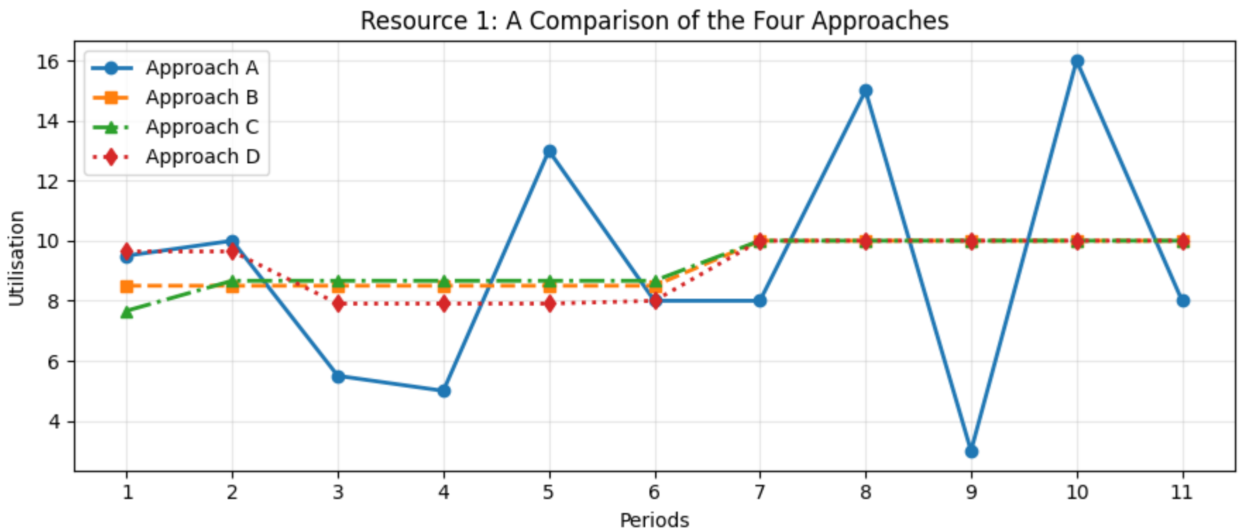

FIGURE 6.3 Comparison of four approaches for Resource 1 of instance 40-3-2-3 in terms of workload utilization across periods

Although the four approaches can generally be distinguished based on their behavior over time, certain instances such as 40-3-2-3 reveal more subtle differences. As shown in Figure 6.3, the early stages of the workload profiles offer valuable insights. Approach D begins with an increase followed by a decrease, reflecting the flexibility permitted by the absence of the single peak constraint while aiming to minimize workload variation. Approach B starts at a high utilization level and maintains a plateau without decreasing, due to the imposed single peak structure combined with the objective of reducing variation. Meanwhile, Approach C increases more gradually and maintains its level, illustrating the compromise reached when minimizing makespan under a single peak constraint. All three approaches outperform Approach A in this regard. In this particular instance, all strategies converge toward similar patterns in the later periods, making the differences less pronounced overall. This suggests that in some configurations, enforcing a single peak profile does not significantly alter the global resource allocation behavior, even though the optimization objectives and constraints differ.

6.5.4 Trade-off Analysis Between Duration and Load Variation in the Single-Peak Profile

This analysis aims to study the trade-off between project duration and workload variation, starting from the minimum duration obtained without the unimodality constraint and extending up to the minimum duration required to achieve a unimodal workload profile (computed separately). The goal is not to impose a single peak constraint, but rather to offer intermediate solutions such as minimizing workload variation when the project duration is insufficient to allow a naturally unimodal profile. This provides decision makers with flexible alternatives depending on the time available. To this end, we use the bi-objective strategy described in Section 6.3.6.

Table 6.9 presents the computational results associated with enforcing a single-peak workload profile across various RCCP instance configurations. For each combination of WPs (Nb-Wps) and resources (Nb-Res), we report the average number of nodes explored, simplex iterations, and solving time required by the solver. Two key performance indicators are also included : **Gap-dur (%)** quantifies the relative increase in project makespan when the unimodal constraint is imposed, while **Gap-var (%)** measures the corresponding reduction in workload variation. It is worth noting that **Gap-dur (%)** and **Gap-var (%)** are computed only for instances where a non-zero duration gap is observed.

TABLE 6.9 Trade-off between makespan increase and workload variation reduction until reaching a single-peak constraint

Nb-Wps	Nb-Res	Nodes	Iterations	Tps (s)	Gap-dur (%)	Gap-var (%)
20	3	2 137.91	211 466.73	31.76	6.85	-35.76
	5	5 024.04	1 835 725.64	286.38	5.97	-29.28
	10	50 136.62	9 370 919.36	761.30	3.59	-23.11
40	3	11 251.09	2 334 522.67	593.15	6.58	-32.06
	5	14 577.53	5 618 664.49	782.05	13.26	-28.26
	10	19 387.42	5 427 022.91	927.17	2.84	-18.60
60	3	11 917.40	1 407 090.76	604.06	31.16	-42.47
	5	14 090.16	1 824 033.60	1280.19	12.92	-27.87
	10	20 629.51	6 982 761.91	4969.31	5.26	-19.41

The results reveal a consistent pattern across all configurations. As the number of WPs or resources increases, the computational effort tends to grow. For the largest configuration (60 WPs), the makespan increases by more than 31%, while the workload variation decreases by over 42%. For 60 WPs and 10 resources, the trade-off becomes more favorable, with only a 5.26% increase in makespan yielding a 19.41% reduction in variation. These findings underscore the need for project managers to weigh their priorities, whether to accept a longer project duration to obtain a unimodal profile or to pursue smoother workload distributions without significantly compromising the planning. We note that the values obtained are not necessarily optimal, given that the time limit for each optimization is 1 000 seconds.

Figure 6.4 illustrates the Pareto fronts of six instances derived from the unconstrained model, which allows workload profiles with multiple peaks. Each curve corresponds to a specific instance configuration and depicts the trade-off between project duration and workload variation. The plotted points represent feasible solutions where the variation is progressively reduced, moving toward the ideal of a single-peak workload profile. As the variation decreases, the project duration generally increases, revealing the cost of imposing smoother workload distributions.

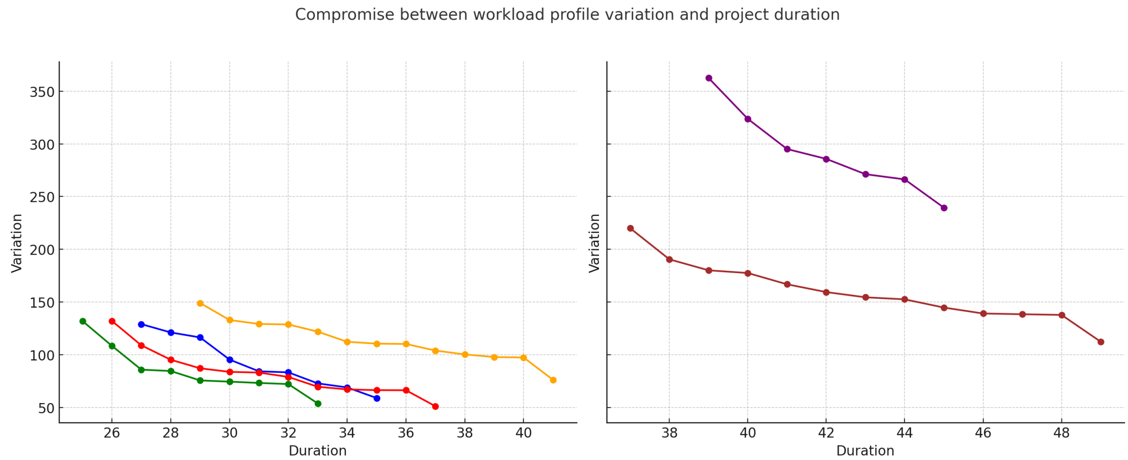

FIGURE 6.4 Pareto fronts illustrating the trade-off between project duration and workload profile variation

The general shape of the curves confirms that achieving smoother workload profiles (i.e., reduced inter-period variation) requires relaxing time constraints, thereby increasing the project makespan.

The graph can be interpreted through three qualitative regions : (1) a region of minimal duration but high workload variation, suitable for rapid execution but potentially unstable ; (2) an intermediate region offering a balanced compromise between workload regularity and makespan ; and (3) a region with very smooth load profiles but extended project durations, relevant only when workload stability is critical.

For configurations represented by steep curves (e.g., the purple series), even a small increase in project duration leads to a significant reduction in variation. Conversely, flatter curves reflect more rigid configurations, where reducing workload variation requires a much larger increase in project duration.

6.6 Conclusion and Perspectives

In this study, we proposed several enhancements to the classical RCCP model to better address realistic planning requirements in tactical project environments, with a focus on solving the workload leveling problem. We began by evaluating the impact of introducing resource-dependent intensities, which adds flexibility to the workload modeling. The results indicate that, even under these additional constraints, the FD-RCCP model produces significantly better-quality solutions compared to the classical C-RCCP, achieving up to 35.16% improvement in workload leveling in certain instance classes, at the cost of increased computation time.

Building on these findings, we further investigated the FD-RCCP model by exploring the effects of enforcing a unimodal workload profile through mathematical constraints. To address the computational challenges associated with this model, we introduced an iterative reoptimization procedure (IIRP), which accelerates the resolution process. Approximately 18% of the tested instances required an extension of the project horizon to accommodate the unimodal constraint, with some particular extensions being substantial. Overall, enforcing a single-peak constraint not only helps avoid frequent hiring and rehiring of workers, but also contributes to substantial reductions in workload variation. These benefits are achieved without significant increases in makespan, particularly in scenarios with greater resource availability.

This observation motivated two complementary analyses. First, we studied the performance of the FDS-RCCP model (which enforces unimodality) relative to the original FD-RCCP, focusing on its impact on the indicator measuring inter-period workload variation reduction. Second, we examined the Pareto front in terms of computational efficiency and solution quality. Additionally, we investigated the trade-off between project duration and workload variation reduction by using a bi-objective ε-constraint approach, gradually increasing the project duration from its minimal feasible value to the point where a unimodal solution becomes possible.

From an industrial perspective, the ability to generate smoother workload profiles has direct implications for shipyard operations. More regular labor demand reduces reliance on overtime and subcontracting, stabilizes employment of skilled trades, and minimizes idle time in critical facilities such as dry docks. These benefits contribute to lower overall project costs and improved on-time delivery of vessels, which is essential for sustaining both commercial competitiveness and naval fleet readiness.

Future research directions include enhancing the model's scalability via decomposition methods, incorporating uncertainty in workload and resource availability, and adapting the approach to specific industrial case studies. It would also be worthwhile to explore alternative workload profiles that incorporate symmetry in addition to unimodality, with the aim of balancing modeling expressiveness and computational efficiency, and studying their impact on project duration.

Disclosure statement No potential conflict of interest was reported by the author(s).

Funding : This research was supported by Jarislowsky/AtkinsRéalis and Mitacs/Thales, and we gratefully acknowledge their financial support.

Data availability The data that support the findings of this study are available from the corresponding author, A.Noureddine, upon reasonable request.

6.7 Appendix

6.7.1 Improved Iterative Re-Optimization Procedure (IIRP)

Algorithm 3: Enhanced Re-Optimization Procedure with Descending Phase

Input: Original problem P, Solver S, Timeout T_1 (e.g., 500 seconds), Total timeout T

Output: Best solution x^* and integrality gap.

```
/* Initialization                                                    */
Set b ← Lower Bound from Node 0;
Set x* ← ∅;
while no feasible solution is found do
    Solve P(b) with solver S using time limit T1;
    if a feasible solution x* is found then
        Save x* as the best solution;
        if all sub-procedures in increasing b up to finding x* do not exceed T1 then
            Stop : x* is the optimal solution;
        end
        else
            /* Descending Phase                                      */
            while b is above the threshold do
                Decrement b;
                Solve P(b) using x* as warm start with limit T1;
                if a new feasible solution x** is found then
                    Save x** as the best solution;
                    Update x* ← x**;
                end
                else
                    if optimization with b exceeds T then
                        No solution exists for b;
                    end
                    else
                        Prove : No solution exists for b;
                        Update b ; Stop;
                    end
                end
            end
        end
    end
    else
        if optimization problem exceeds timeout T1 then
            Increment b by 1;
            Attempt to solve P(b) again;
        end
    end
end
/* Stopping criteria                                                 */
    — x* is feasible and optimal, or
    — The solver exhausts all possibilities within bounds and returns infeasible.
return x* and b;
```

6.7.2 Algorithm for RCCP Instance Generation

Algorithm 4: Instance Generation for RCCP

Input:

$|I| \in \{20, 40, 60\}$: number of WPs.

$R \in \{3, 5, 10\}$: number of resource types.

K_r : renewable capacity of resource type r.

$NC \in \{1.4, 1.7, 2.1\}$: precedence density factor.

$RF \in \{0.25, 0.5, 0.75\}$: resource factor.

MRS : target modified resource strength.

Output: Generated instance

Step 1 : Define Parameters

Draw $|I|$, R, RF, NC from their sets.

Set K_r using a scaling rule that depends on $|I|$ and RF.

Step 2 : Generate Precedence Graph

Randomly generate a directed acyclic graph with density controlled by NC.

Step 3 : Generate WPs and Resource Data

For each WP i, draw a minimal processing time m_i uniformly in $[1, 5]$.

Assign resource requirements to each WP for each resource type based on RF.

Select three types of resources.

Generate dependency coefficients $[\alpha_{r_1 r_2}, \beta_{r_1 r_2}]$ for each resource pair $(r_1, r_2), r_1 < r_2$.

Step 4 : Adjust Demands to Match MRS

Initialize demands for each WP to 0.

Increment demands iteratively until MRS is satisfied.

Verify feasibility after each increment.

If infeasible : Decrement demand and retry.

Output : For each configuration, generate **5** random instances (different seeds), yielding $81 \times 5 = 405$ instances in total.

CHAPITRE 7 DISCUSSION GÉNÉRALE

Les trois articles qui structurent cette thèse s'inscrivent dans une démarche méthodique d'amélioration du *RCCP*, à la fois sur le plan de la structure mathématique et du comportement numérique, dans le but de résoudre des problèmes de grande taille et comportant des contraintes plus pratiques pouvant parfois complexifier le problème. L'objectif de recherche était, parallèlement au développement de modèles de planification plus performants, de comprendre les raisons profondes des différences de performance observées entre les formulations afin d'en déduire une trajectoire d'amélioration progressive. Cette approche, fondée sur une analyse du comportement du solveur et des structures de contraintes, a permis de passer d'observations empiriques à une refonte du modèle *RCCP*, puis à son extension vers des problématiques multi-objectifs et de nivellement des ressources.

Cet objectif a été décliné en trois contributions successives et complémentaires, chacune répondant à un enjeu clé du problème RCCP. La première contribution a posé les fondations d'un modèle en temps continu mieux conditionné, permettant de résoudre des instances de 50 lots de travail. La deuxième contribution a approfondi cette logique d'amélioration en testant plusieurs formulations, démontrant une accélération moyenne d'un facteur sept et une extension de la capacité de résolution de 50 à 80 lots de travail. Enfin, la troisième contribution a intégré la dimension de nivellement des ressources, introduisant une formulation bi-objectif conciliant durée de projet et stabilité de charge, avec une réduction moyenne de 35 % des variations inter-périodes, et un profil unimodal maintenu dans 82 % des cas sans allongement du projet.

Ainsi, ces trois contributions forment un cadre unifié pour la planification tactique, capable d'allier la performance des formulations et leur adaptabilité à des contextes pratiques. Ce travail constitue une contribution vers des modèles MIP de RCCP qui sont réellement exploitables dans les environnements industriels, renforçant le lien entre modélisation mathématique MIP et prise de décision tactique. Il est particulièrement intéressant de constater que l'amélioration des performances ne se traduit pas uniquement par une réduction des écarts d'optimalité, mais également par une amélioration notable de la qualité des solutions. En effet, la comparaison des coûts met en évidence des différences parfois très significatives entre les formulations, confirmant que les gains de rapidité s'accompagnent d'une bien meilleure évaluation du coût du projet.

En pratique, nos travaux montrent que lorsqu'un modèle ne peut être résolu efficacement dès les premières tentatives, il est souvent plus pertinent de comprendre son comportement structurel et numérique avant d'envisager des approches plus complexes et difficiles à adapter.

Cette démarche met en lumière qu'une analyse approfondie de la formulation de base peut révéler des opportunités d'amélioration significatives, souvent négligées lorsqu'on se tourne prématurément vers des techniques de décomposition, des heuristiques ou des métaheuristiques plus difficiles à gérer en pratique. En d'autres termes, un renforcement progressif du modèle peut, à lui seul, permettre de résoudre des instances de plus grande taille, tout en préservant la simplicité, la robustesse et la traçabilité essentielles dans un environnement industriel.

Par ailleurs, étant donné le caractère stochastique et évolutif des projets à ce niveau de planification, toute modification des hypothèses du problème peut entraîner des ajustements plus complexes lorsqu'on utilise des approches de décomposition ou d'optimisation approchée. À l'inverse, une formulation compacte, directement résolue par un solveur, offre une flexibilité permettant d'intégrer rapidement des changements d'objectifs, de contraintes ou de paramètres.

De plus, ces modèles introduisent une nouvelle façon de concevoir la charge de travail et la gestion des ressources. En effet, l'imposition d'une structure particulière du profil de charge permet de mieux gérer le personnel et d'optimiser des indicateurs souvent négligés dans la littérature, tels que le nombre de recrutements et de licenciements. Ces fluctuations de main-d'œuvre réduisent généralement la productivité globale et créent une instabilité organisationnelle au sein des équipes de travail.

Les trois articles contribuent à redéfinir la manière d'aborder les problèmes de planification tactique. Plutôt que de recourir à des algorithmes complexes à implémenter et à maintenir, ils mettent en évidence des opportunités d'amélioration significatives au sein même des modèles MIP. En effet, les résultats obtenus démontrent qu'un modèle MIP amélioré de manière progressive peut offrir des performances bien supérieures à celles généralement attendues à partir d'une formulation de base. Cette approche replace le modèle MIP comme un cadre d'analyse stratégique, capable d'accompagner les ajustements opérationnels sans nécessiter de développements lourds ou spécifiques.

La figure 7.1 illustre les résultats clés ainsi que les contributions proposées de chaque article.

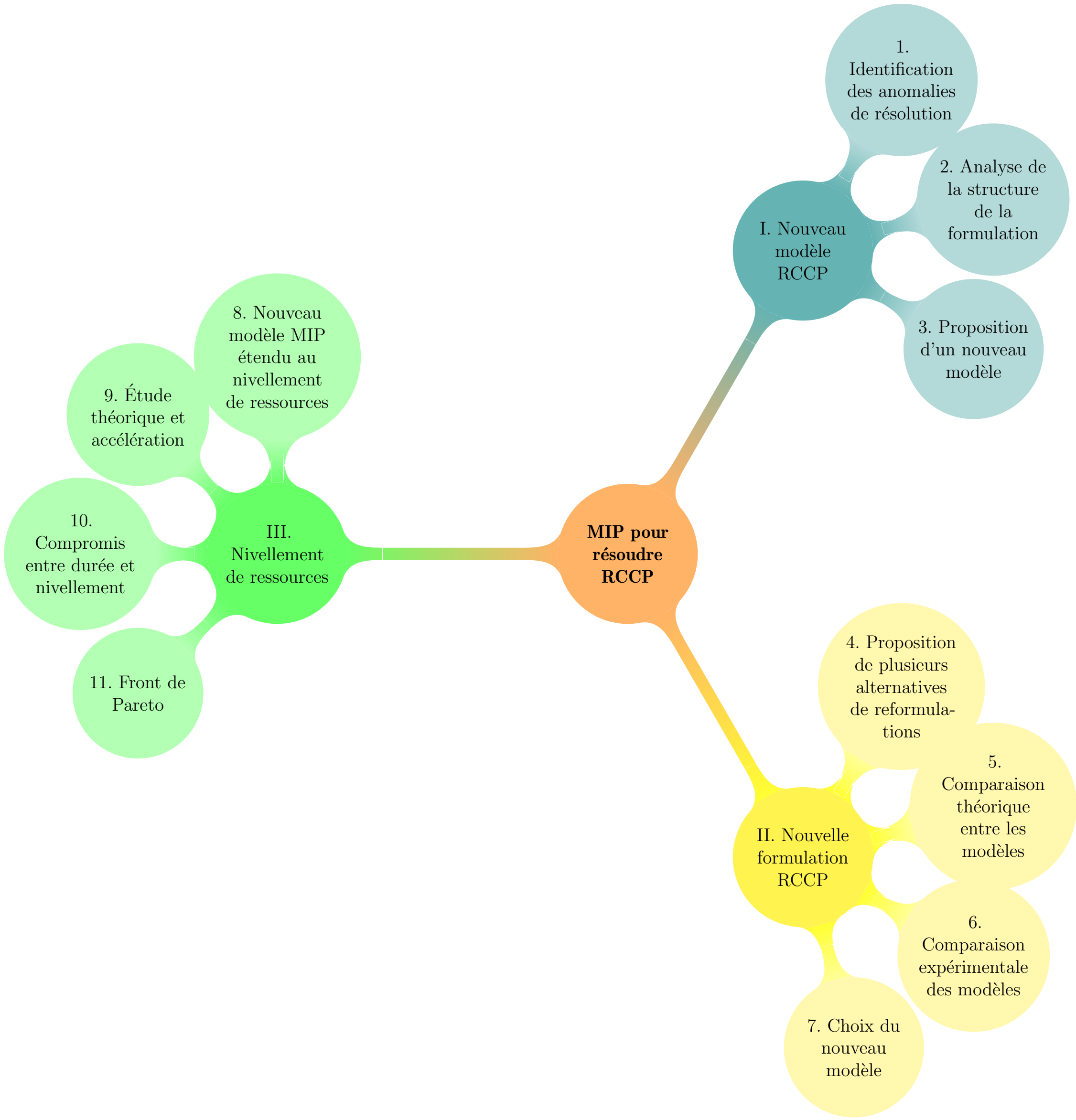

FIGURE 7.1 Contributions et constats de chaque article

CHAPITRE 8 CONCLUSION

Dans cette thèse, nous avons étudié le problème du *RCCP* afin d'améliorer la planification tactique des projets. Au-delà de l'amélioration des performances numériques et de l'impact pratique, les travaux présentés ont cherché à comprendre, expliquer et identifier les facteurs qui influencent le comportement des modèles *RCCP*, en s'appuyant sur l'analyse du comportement du solveur, l'amélioration progressive de la formulation et une validation expérimentale.

Le premier article, publié dans *International Journal of Production Research*, a posé les bases de cette réflexion en mettant en évidence les limites structurelles du modèle de référence. L'analyse approfondie du comportement du solveur a montré que la performance d'un modèle ne dépend pas uniquement de sa taille ou de la qualité du relâché, mais aussi de la manière dont les variables et contraintes sont formulées et interagissent. Ce diagnostic a permis d'identifier des sources de redondance et de symétrie qui ralentissaient la convergence et a conduit à la proposition de formulations alternatives visant à casser la symétrie des solutions et à améliorer la stabilité du modèle. Cette première étape, centrée sur la compréhension et la détection des faiblesses structurelles, a ouvert la voie à une reformulation ciblée du modèle *RCCP*.

Le deuxième article, soumis à *Computers & Operations Research*, a concrétisé cette transition en proposant une série de nouvelles formulations *MIP* inspirées des constats du premier travail. Ce volet a montré qu'il est possible d'obtenir des performances très différentes entre deux modèles partageant un relâché équivalent, soulignant le rôle décisif de la structure dans le comportement du solveur. Les nouvelles formulations, développées dans un cadre en temps continu, ont permis de réduire considérablement les temps de résolution, jusqu'à sept fois plus rapides selon les instances, tout en élargissant significativement la capacité du modèle à traiter des problèmes plus grands : le passage de 50 à 80 *lots de travail*, notamment pour les instances avec 10 et 20 ressources, respectivement. La nouvelle approche se distingue également par des performances comparables à celles des modèles discrets, tout en évitant leurs simplifications qui négligent parfois des solutions prometteuses et plus applicables en pratique. Enfin, une analyse comparative avec les principaux modèles de la littérature a consolidé la position de cette formulation comme une base pour les travaux futurs sur la planification tactique avec des extensions pratiques.

Le troisième article, soumis à *OR Spectrum*, constitue l'aboutissement logique en considérant une extension plus complexe. Partant du modèle en temps continu, il introduit une formulation bi-objectif intégrant le nivellement des ressources, une dimension essentielle pour réduire les coûts d'un projet au niveau tactique. En imposant un profil de charge unimodal (croissant–décroissant), cette approche vise à stabiliser la charge de travail tout au long du projet, répondant ainsi à un besoin pratique : éviter les pics d'activité et les sous-utilisations successives des ressources. L'étude expérimentale a démontré que le modèle permet d'obtenir une charge unimodale sans allonger la durée du projet dans 82 % des cas, et que l'utilisation d'intensités variables pour chaque ressource permet de réduire de 35 % les variations inter-périodes. De plus, une stratégie d'accélération, basée sur le resserrement progressif des bornes inférieures, a été proposée : la résolution s'avère au moins dix fois plus rapide que la résolution directe. Enfin, une approche bi-objectif a été développée afin d'obtenir le front de Pareto.

Concernant l'amélioration des performances du solveur, l'identification des anomalies de résolution constitue, dans plusieurs cas, une information précieuse permettant de mieux comprendre le comportement du modèle. Toutefois, afin d'en atténuer l'impact, nous avons choisi de revoir la formulation et/ou de résoudre le modèle au moyen d'étapes d'optimisation successives, selon une approche incrémentale. Les principales difficultés susceptibles d'être rencontrées dans un contexte stochastique proviennent du fait que certaines anomalies, telles que la lenteur de la résolution du problème relaxé, deviennent à un certain seuil particulièrement difficiles à corriger. Dans de tels cas, le défi consiste à obtenir rapidement une solution de bonne qualité, aussi proche que possible de l'optimal.

Enfin, notre étude s'est concentrée sur la réduction du nombre de nœuds explorés et du nombre d'itérations, sans approfondir les stratégies de branchement susceptibles d'améliorer davantage le processus de recherche de solutions, notamment celles fondées sur des approches guidées par l'apprentissage automatique.

Plusieurs pistes de recherche peuvent être envisagées à la suite de ce travail. Elles s'inscrivent à la fois dans la continuité des contributions de la thèse et dans une logique d'élargissement des approches de modélisation et de résolution du RCCP.

Une première piste concerne la poursuite de l'étude du modèle *RCCP* lui-même. Bien que le premier article ait permis une analyse approfondie de son comportement structurel et numérique, cette dimension mérite d'être explorée avec davantage d'outils, dans un travail dédié. En effet, il serait judicieux d'améliorer les solutions initiales à travers des heuristiques ou métaheuristiques fournissant des solutions de qualité dès le départ, afin de réduire le nombre de branches. Il serait aussi pertinent d'utiliser l'apprentissage automatique afin d'améliorer les stratégies de branchement, notamment dans le cas où le modèle relaxé au nœud 0 est rapide à résoudre.

À partir des résultats du troisième article, il apparaît également que chaque fonction objectif possède ses propres spécificités de résolution. L'analyse du compromis entre la durée de projet et la variation de charge a montré que les caractéristiques de la fonction objectif influencent directement la complexité du modèle et la sensibilité du solveur. Cette observation ouvre la voie à de nouvelles opportunités d'amélioration, notamment à travers la conception de stratégies d'optimisation adaptées à chaque type d'objectif ou à l'optimisation multi-objectif. Dans ce prolongement, un nouvel article actuellement en phase de rédaction est consacré au Resource-Driven RCCP. L'objectif est d'exploiter la procédure incrémentale et de l'améliorer afin d'accélérer les performances dans un contexte ou on fixe les capacités de ressources tout en minimisant la durée du projet.

Une autre piste de recherche prometteuse réside dans la prise en compte de l'aspect stochastique du RCCP. En effet, les décisions tactiques sont souvent confrontées à des incertitudes liées aux durées des lots de travail, aux capacités des ressources ou aux demandes futures. L'introduction d'approches stochastiques ou robustes permettrait de mieux représenter ces aléas et d'évaluer la stabilité des solutions face aux variations de paramètres. Une telle extension favoriserait le passage d'un RCCP déterministe à un cadre décisionnel plus flexible.

Les résultats de la thèse suggèrent une piste de recherche consistant à combiner l'analyse structurelle du modèle avec une étude critique du comportement du solveur. Cette approche, centrée sur la compréhension de l'interaction entre formulation et résolution, pourrait être prolongée dans des méthodes de décomposition (par exemple, Benders, Lagrangienne ou Dantzig-Wolfe) développées avec le même esprit d'analyse. Une telle démarche pourrait aider à éviter certains phénomènes classiques du domaine de l'ordonnancement, tels que la dégénérescence ou la faible efficacité des relaxations, tout en garantissant une meilleure stabilité computationnelle. À plus long terme, il serait également pertinent d'envisager une planification intégrée combinant les niveaux tactique et opérationnel, afin de réduire les incertitudes et d'améliorer la cohérence globale des décisions.

RÉFÉRENCES

[1] Tobias Achterberg. Constraint integer programming. *Ph.D. thesis, TU Berlin*, 2007.

[2] Rashpal Ahluwalia and Denis Pinha. Decision support system for production planning in the ship repair industry. *Industrial and Systems Engineering Review*, 2(1) :52–61, 2014.

[3] Bernardo F. Almeida, Isabel Correia, and Francisco Saldanha-da Gama. An instance generator for the multi-skill resource-constrained project scheduling problem. Technical report, University of Lisbon, 2015.

[4] Ramón Alvarez-Valdés and José M. Tamarit. Project scheduling polyhedron : Dimension, facets and lifting theorems. *European Journal of Operational Research*, 67(2) :204–220, 1993.

[5] Robert N. Anthony. *Planning and Control Systems : A Framework for Analysis*. Harvard University Press, Boston, 1965.

[6] Tullio Tolio Arianna Alfieri and Marcello Urgo. A project scheduling approach to production planning with feeding precedence relations. *International Journal of Production Research*, 49(4) :995–1020, 2011.

[7] Christian Artigues. On the strength of time-indexed formulations for the resource-constrained project scheduling problem. *Operations Research Letters*, 45(2) :154–159, 2017.

[8] Christian Artigues, Sophie Demassey, and Emmanuel Néron, editors. *Resource-Constrained Project Scheduling : Models, Algorithms, Extensions and Applications*. Wiley-ISTE, London, UK, 2008.

[9] Christian Artigues, Philippe Michelon, and Stéphane Reusser. Insertion techniques for static and dynamic resource constrained project scheduling. *European Journal of Operational Research*, 149 :249–267, 2003.

[10] Francisco Ballestín, Christoph Schwindt, and Jürgen Zimmermann. Resource leveling in make-to-order production : Modeling and heuristic solution method. *International Journal of Operations Research*, 4(1) :50–62, 2007.

[11] Niels-Fabian Baur and Julia Rieck. *Project Management with Scarce Resources in Disaster Response*, pages 607–614. 09 2020.

[12] Georges Baydoun, Alain Haït, Robert Pellerin, Bernard Clément, and Guillaume Bouvignies. A rough-cut capacity planning model with overlapping. *OR Spectrum*, 38, 03 2016.

[13] Swapnil Bhalla, Erlend Alfnes, and Hans-Henrik Hvolby. Tools and practices for tactical delivery date setting in engineer-to-order environments : a systematic literature review. *International Journal of Production Research*, 61(7) :2339–2371, 2023.

[14] Lucio Bianco and Massimiliano Caramia. A new formulation for the project scheduling problem under limited resources. *Flexible Services and Manufacturing Journal*, 25, 06 2011.

[15] Lucio Bianco, Massimiliano Caramia, and Stefano Giordani. Resource levelling in project scheduling with generalized precedence relationships and variable execution intensities. *OR Spectrum*, 38(2) :405–425, 2016.

[16] Lucio Bianco, Massimiliano Caramia, and Stefano Giordani. The total adjustment cost problem with variable activity durations and intensities. *European Journal of Industrial Engineering*, 11(6) :708–724, 2017.

[17] Lucio Bianco, Massimiliano Caramia, Stefano Giordani, and Alessio Salvatore. Scheduling activities in project network with feeding precedence relations : an earliest start forward recursion algorithm. *Annals of Operations Research*, 338(1) :193–224, 2024.

[18] Robert E. Bixby. Solving real-world linear programs : A decade and more of progress. *Operations Research*, 50(1) :3–15, 2002.

[19] Jacek Blazewicz, Roman Cellary, Roman Slowinski, and Jozef Weglarz. *Scheduling*

under Resource Constraints : Deterministic Models, volume 7 of *Annals of Operations Research*. Baltzer, Basel, 1986.

[20] Tamara Borreguero Sanchidrián, Tom Portoleau, Christian Artigues, Alvaro García Sánchez, Miguel Ortega Mier, and Pierre Lopez. Large neighborhood search for an aeronautical assembly line time-constrained scheduling problem with multiple modes and a resource leveling objective. *Annals of Operations Research*, 338(1) :13–40, 2024.

[21] Raphaël Boudreault and Vanessa Simard. Long-term planning of preventive maintenance using constraint programming : A naval case study. In *Proceedings of the 14th International Conference on Operations Research and Enterprise Systems, ICORES 2025, Porto, Portugal, February 23–25, 2025*, pages 32–44. SCITEPRESS, 2025.

[22] Raphaël Boudreault, Vanessa Simard, Daniel Lafond, and Claude-Guy Quimper. A constraint programming approach to ship refit project scheduling. In Christine Solnon, editor, *28th International Conference on Principles and Practice of Constraint Programming (CP 2022)*, volume 235 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 10 :1–10 :16, Dagstuhl, Germany, 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

[23] Robert Brachmann and Rainer Kolisch. The impact of flexibility on engineer-to-order production planning. *International Journal of Production Economics*, 239 :108183, 06 2021.

[24] Roland Braune and Karl F Doerner. Real-world flexible resource profile scheduling with multiple criteria : learning scalarization functions for mip and heuristic approaches. *Journal of the Operational Research Society*, 68(8) :952–972, 2017.

[25] Peter Brucker. *Scheduling Algorithms*. Springer, Berlin, 5th edition, 2007.

[26] Andréa Nunes Carvalho, Fabricio Oliveira, and Luiz Felipe Scavarda. Tactical capacity planning in a real-world eto industry case : An action research. *International Journal of Production Economics*, 167 :187–203, 2015.

[27] Andréa Nunes Carvalho, Fabricio Oliveira, and Luiz Felipe Scavarda. Tactical capacity

planning in a real-world eto industry case : A robust optimization approach. *International Journal of Production Economics*, 180 :158–171, 2016.

[28] R. K. Chakrabortty, R. Sarker, and D. Essam. Event-based approaches for solving multi-mode resource constrained project scheduling problem. In *Evolutionary Computation in Combinatorial Optimization*, pages 397–408. Springer, 2014.

[29] Alfred D. Chandler. *The Visible Hand : The Managerial Revolution in American Business*. Harvard University Press, Cambridge, MA, 1977.

[30] Kaouthar Cherkaoui. *Planification tactique des grands projets d'ingénierie et de construction*. Ph.D. thesis, École Polytechnique de Montréal, Canada, 2017. Accessible via Google Scholar.

[31] Kaouthar. Cherkaoui, Pierre Baptiste, Robert Pellerin, Nathalie Perrier, and Alain Haït. Proactive tactical planning approach for large scale engineering and construction projects. *Journal of Modern Project Management*, 5 :96–105, 05 2017.

[32] Kaouthar Cherkaoui, Robert Pellerin, Pierre Baptiste, and Alain Haït. A time driven rccp model with two levels of planning and a reactive planning approach for tactical project planning. *Procedia Computer Science*, 64 :257–264, 2015.

[33] N. Christofides, R. Alvarez-Valdéz, and J. M. Tamarit. Project scheduling with resource constraints : A branch and bound approach. *European Journal of Operational Research*, 29(3) :262–273, 1987.

[34] Vasek Chvátal. *Linear Programming*. W. H. Freeman, 1983.

[35] Yann Collette and Patrick Siarry. *Multiobjective Optimization. Principles and Case Studies*. Springer, 2003.

[36] Isabel Correia, Lídia Lourenço, and Francisco Saldanha da Gama. Project scheduling with flexible resources : Formulation and inequalities. *OR Spectrum*, 34 :639–667, July 2012.

[37] Atilla Damci and Gul Polat. Impacts of different objective functions on resource leveling in construction projects : A case study. *Journal of Civil Engineering and Management*, 20(4) :537–547, 2014.

[38] George B. Dantzig. *Linear Programming and Extensions*. Princeton University Press, 1963.

[39] R. De Boer. *Resource-constrained multi-project management, a hierarchical decision support system.* Ph.D dissertation, University Of Twente, The Netherlands, 1998.

[40] Khaled El-Rayes and Dong Jun. Optimizing resource leveling in construction projects. *Journal of Construction Engineering and Management*, 135(8) :716–726, 2009.

[41] Thomas Felberbauer, Walter J. Gutjahr, and Karl F. Doerner. Stochastic project management : Multiple projects with multi-skilled human resources. *Journal of Scheduling*, 22(3) :271–288, 2019.

[42] Noud Gademann and Marco Schutten. Linear-programming-based heuristics for project capacity planning. *Iie Transactions*, 37, 02 2005.

[43] R. Galmard. Cadre généralisé de planification tactique de projets de remise à niveau de navires. Master dissertation, University Polytechnique Montréal, Canada, 2021.

[44] Thorsten Gather and Jürgen Zimmermann. Exact methods for the resource levelling problem. *Journal of Scheduling*, 14(6) :557–569, 2011.

[45] Thorsten Gather, Jürgen Zimmermann, and Jan-Hendrik Bartels. Exact methods for the resource levelling problem. volume 14, pages 557–569. 12 2011.

[46] Marzieh Ghiyasinasab, Nadia Lehoux, Sylvain Ménard, and Caroline Cloutier. Production planning and project scheduling for engineer-to-order systems-case study for engineered wood production. *International Journal of Production Research*, 59(4) :1068–1087, 2021.

[47] E. W. Hans. *Resource loading by branch-and-price techniques.* Ph.D dissertation,

University Of Twente, The Netherlands, 2001.

[48] Erwin W. Hans, W. Herroelen, R. Leus, and G. Wullink. A hierarchical approach to multi-project planning under uncertainty. *Omega*, 35(5) :563–577, 2001.

[49] R. Harris. *Precedence and Arrow Networking Techniques for Construction.* John Wiley and Sons, New York, NY, 1978.

[50] S. Hartmann. Project scheduling with resource capacities and requests varying with time : A case study. *Flexible Services and Manufacturing Journal*, 25(1) :74–93, 2013.

[51] A Haït and G. Baydoun. A new event-based milp model for the resource-constrained project scheduling problem with variable intensity activity (rcpsvp). *Proceedings of the IEEE International Conference on Industrial Engineering and Engineering Management, Hong Kong*, 12 2012.

[52] Naihui He, David Z. Zhang, and Baris Yuce. Integrated multi-project planning and scheduling - a multiagent approach. *European Journal of Operational Research*, 302(2) :688–699, 2022.

[53] Tarek Hegazy. Optimization of resource allocation and leveling using genetic algorithms. *Journal of Construction Engineering and Management*, 125(3) :167–175, 1999.

[54] Christian Heimerl and Rainer Kolisch. Scheduling and staffing multiple projects with a multi-skilled workforce. *OR Spectrum*, 32(2) :343–368, April 2010.

[55] Hydro-Québec. Portrait des ressources énergétiques d'hydro-québec. `https://www.hydroquebec.com/data/achats-electricite-quebec/pdf/portrait-ressources-energetiques.pdf`, November 2019. Document officiel, Hydro-Québec.

[56] Patrik Jonsson and Stig-Åke Mattsson. Use and applicability of capacity planning methods. *Production and Inventory Management Journal*, 43(3/4) :89–95, 2002.

[57] A. P. Kanyalkar and G. K. Adil. Aggregate and detailed production planning integrating

procurement and distribution plans in a multi-site environment. *International Journal of Production Research*, 45(22) :5329–5353, 2007.

[58] Harold Kerzner. *Project Management : A Systems Approach to Planning, Scheduling, and Controlling.* Wiley, Hoboken, NJ, 12th edition, 2017.

[59] Tamás Kis. Rcps with variable intensity activities and feeding precedence constraints. In *Perspectives in Modern Project Scheduling*, pages 105–129. Springer, 2006.

[60] Tamás Kis. A branch-and-cut algorithm for scheduling of projects with variable-intensity activities. *Math. Program.*, 103 :515–539, 07 2005.

[61] Robert Klein. Scheduling of resource constrained projects. Publications of darmstadt technical university, institute for business studies (bwl), Darmstadt Technical University, Department of Business Administration, Economics and Law, Institute for Business Studies (BWL), 2000.

[62] Rainer Kolisch and Arno Sprecher. Psplib - a project scheduling problem library : Or software - orsep operations research software exchange program. *European Journal of Operational Research*, 96(1) :205–216, 1997.

[63] Oumar Koné, Christian Artigues, Pierre Lopez, and Marcel Mongeau. Event-based milp models for resource-constrained project scheduling problems. *Computers & Operations Research*, 38(1) :3–13, 2011.

[64] Mark Lehner. *The Complete Pyramids : Solving the Ancient Mysteries.* Thames & Hudson Ltd., London, 1997.

[65] Sou-Sen Leu, Ankai Chen, and Chung-Huei Yang. A fuzzy optimal model for construction resource leveling scheduling. *Canadian Journal of Civil Engineering*, 26 :673–684, 1999.

[66] Jing Liu. *Dual-Level Resource-Constrained Multi-Project Scheduling Framework for Prefabrication in Construction.* Ph.d. thesis, University of Alberta, Edmonton, AB, Canada, 2018.

[67] Andrea Lodi. Mixed integer programming computation. *4OR*, 8(2) :145–175, 2010.

[68] Vittorio Maniezzo and Alfredo Mingozzi. A heuristic procedure for the multi-mode project scheduling problem based on benders decomposition. In Jan Weglarz, editor, *Project Scheduling : Recent Models, Algorithms and Applications*, pages 179–196. Kluwer Academic Publishers, Dordrecht, 1999.

[69] Malek Masmoudi and Alain Haït. A GA-based fuzzy resource leveling optimization for helicopter maintenance activity. In *Proceedings of the 7th Conference of the European Society for Fuzzy Logic and Technology (EUSFLAT-11)*, pages 665–672, Aix-les-Bains, France, 2011. Atlantis Press.

[70] Kris G. Mattila and Daniel M. Abraham. Resource leveling of linear schedules using integer linear programming. *Journal of Construction Engineering and Management*, 124(3) :232–244, 1998.

[71] RR Mauritz. Planning in supply networks using aggregated resource feasibility. Master's thesis, University of Twente, 2022.

[72] George Mavrotas. Effective implementation of the ϵ-constraint method in multi-objective mathematical programming problems. *Applied Mathematics and Computation*, 213(2) :455–465, 2009.

[73] Aristide Mingozzi, Vittorio Maniezzo, Salvatore Ricciardelli, and Lucio Bianco. An exact algorithm for the resource constrained project scheduling problem based on a new mathematical formulation. *Management Science*, 44, 05 1996.

[74] Pierre Morin, Christian Artigues, and Alain Haït. Periodically aggregated resource-constrained project scheduling problem. *European Journal of Industrial Engineering*, 11 :792, 01 2017.

[75] Pierre-Antoine Morin, Christian Artigues, and Alain Haït. A column generation scheme for the periodically aggregated resource-constrained project scheduling problem. In *16th International Conference on Project Management and Scheduling*, pages 165–168, Rome, Italy, 2018.

[76] Pierre-Antoine Morin, Christian Artigues, Alain Haït, Tamás Kis, and Frits C.R. Spieksma. A project scheduling problem with periodically aggregated resource-constraints. *Computers & Operations Research*, 141 :105688, 2022.

[77] Peter W. G. Morris. *The Management of Projects.* Thomas Telford Publishing, London, 1994.

[78] Anulark Naber. Resource-constrained project scheduling with flexible resource profiles in continuous time. *Computers & Operations Research*, 84 :33–45, 2017.

[79] Anulark Naber and Rainer Kolisch. Mip models for resource-constrained project scheduling with flexible resource profiles. *European Journal of Operational Research*, 239(2) :335–348, 2014.

[80] George L. Nemhauser and Laurence A. Wolsey. *Integer and Combinatorial Optimization.* Wiley, 1988.

[81] Anas Neumann, Adnene Hajji, Monia Rekik, and Robert Pellerin. A two-level optimization approach for engineer-to-order project scheduling. *IFAC-PapersOnLine*, 55(10) :2587–2592, 2022.

[82] Anas Neumann, Adnene Hajji, Monia Rekik, and Robert Pellerin. Integrated planning and scheduling of engineer-to-order projects using a lamarckian layered genetic algorithm. *International Journal of Production Economics*, 267 :109077, 2024.

[83] Anis Noureddine, François Soumis, and Robert Pellerin and. A new mip rccp model for tackling tactical project planning. *International Journal of Production Research*, 0(0) :1–25, 2024.

[84] Ibrahim H. Osman and Gilbert Laporte. Metaheuristics : A bibliography. *Annals of Operations Research*, 63 :513–623, 1996.

[85] Vangelis Th. Pachos. *Optimisation combinatoire : Concepts fondamentaux.* Hermes Science Publications, Paris, 2005.

[86] R. Pellerin. *Modèle d'ordonnancement dynamique de projets de réfection*. Ph.D dissertation, University Polytechnique Montréal, Canada, 1997.

[87] Robert Pellerin and Nathalie Perrier. A review of methods, techniques and tools for project planning and control. *International Journal of Production Research*, 57(7) :2160–2178, 2019.

[88] Robert Pellerin, Nathalie Perrier, and François Berthaut. A survey of hybrid metaheuristics for the resource-constrained project scheduling problem. *European Journal of Operational Research*, 280(2) :395–416, 2020.

[89] Michael L. Pinedo. *Scheduling : Theory, Algorithms, and Systems*. Springer, New York, 5th edition, 2016.

[90] A. A. B. Pritsker, L. J. Waiters, and P. M. Wolfe. Multiproject scheduling with limited resources : A zero-one programming approach. *Management Science*, 16 :93–108, 1969.

[91] Mohammad Ranjbar and F. Kianfar. Resource-constrained project scheduling problem with flexible work profiles : A genetic algorithm approach. *Scientia Iranica*, 17 :25–35, 06 2010.

[92] Majid Sabzehparvar and Mohammad Seyed hosseini. A mathematical model for the multi-mode resource-constrained project scheduling problem with mode dependent time lags. *The Journal of Supercomputing*, 44 :257–273, 06 2008.

[93] Robert A. Scott. *The Gothic Enterprise : A Guide to Understanding the Medieval Cathedral*. University of California Press, Berkeley, 2003.

[94] Ahmed Senouci and Nader Eldin. Use of genetic algorithms in resource scheduling of construction projects. *Journal of Construction Engineering and Management*, 130(6) :869–877, 2004.

[95] T. L. Smunt. Rough-cut capacity planning in a learning environment. *IEEE Transactions on Engineering Management*, 43(3) :334–341, 1996.

[96] S.W. Son and M.J. Skibniewski. Multiheuristic approach for resource leveling problem in construction. *Journal of Construction Engineering and Management*, 125(1) :23–31, 1999.

[97] François Soumis. Decomposition and column generation. Les Cahiers du GERAD G-97-42, Groupe d'études et de recherche en analyse des décisions, GERAD, Montréal QC H3T 2A7, Canada, June 1997.

[98] J. D. Stevens. *Techniques for Construction Network Scheduling.* McGraw-Hill, New York, NY, 1990.

[99] Muchamad Sugarindra and Raihan Nurdiansyah. Production capacity optimization with rough cut capacity planning (rccp). *IOP Conference Series : Materials Science and Engineering*, 722 :012046, 01 2020.

[100] Maedeh Taghaddos, Ali Mousaei, Hosein Taghaddos, Ulrich Hermann, Yasser Mohamed, and Simaan AbouRizk. Optimized variable resource allocation framework for scheduling of fast-track industrial construction projects. *Automation in Construction*, 158 :105208, 2024.

[101] F. B. Talbot. Resource-constrained project scheduling with time-resource trade-offs : The nonpreemptive case. *Management Science*, 28(10) :1197–1210, 1982.

[102] Ilia Tarasov, Alain Haït, and Olga Battaïa. A generalized milp formulation for the period-aggregated resource leveling problem with variable job duration. *Algorithms*, 13(1) :6, 2019.

[103] Ilia Tarasov, Alain Haït, and Olga Battaïa. Benders decomposition for a period-aggregated resource leveling problem with variable job duration. *Computers & Operations Research*, 132 :105258, 2021.

[104] Thales. La Marine royale canadienne confie l'entretien de sa flotte à Thales pour 35 ans. *Press release*, août 2017.

[105] Martin Tritschler, Anulark Naber, and Rainer Kolisch. A hybrid metaheuristic for

resource-constrained project scheduling with flexible resource profiles. *European Journal of Operational Research*, 262, 03 2017.

[106] Thomas E. Vollmann, William L. Berry, and D. Clay Whybark. *Manufacturing Planning and Control Systems.* McGraw-Hill, New York, 4th edition edition, 1997.

[107] Swapnil Wanjari and Abhay Tawalare. Formulation of a parabolic resource profile of manpower requirement using various empirical formulae of resources levelling. *Procedia Engineering*, 161 :168–173, 12 2016.

[108] Jan Weglarz, Joanna Jozefowska, Magdalena Mika, and Grzegorz Waligora. Project scheduling with finite or infinite number of activity processing modes : A survey. *European Journal of Operational Research*, 208(3) :177–205, 2011.

[109] Wayne L. Winston. *Operations Research : Applications and Algorithms.* Thomson Brooks/Cole, 2004.

[110] Laurence A. Wolsey. *Integer Programming.* Wiley, 1998.

[111] G Wullink, Noud Gademann, Erwin Hans, and Harten van. Scenario-based approach for flexible resource loading under uncertainty. *International Journal of Production Research*, 42, 01 2003.

[112] Mohamed Yagouni. *Contribution à la résolution des problèmes complexes de l'optimisation combinatoire.* Ph.d. thesis, Université des Sciences et de la Technologie Houari Boumediene (USTHB), 2017.

[113] Xufeng Yao, Nourah Almatooq, Ronald G Askin, and Greg Gruber. Capacity planning and production scheduling integration : improving operational efficiency via detailed modelling. *International Journal of Production Research*, 60(24) :7239–7261, 2022.

[114] A. Zimmermann and N. Trautmann. A list-scheduling heuristic for the short-term planning of assessment centers. *Journal of Scheduling*, 21 :131–142, 2018.

[115] Ágota Bányai. The influence of procurement logistics strategy on organizational hierar-

chies : Strategic, tactical, and operational perspectives. *Advanced Logistic Systems - Theory and Practice*, 18(4) :73–88, 2024.