



HAL
open science

Efficient solution approaches and uncertainty estimation for the project resource leveling problem

Ilia Tarasov

► **To cite this version:**

Ilia Tarasov. Efficient solution approaches and uncertainty estimation for the project resource leveling problem. Other [cs.OH]. Institut National Polytechnique de Toulouse - INPT, 2021. English. NNT : 2021INPT0018 . tel-04169703

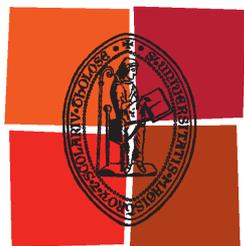
HAL Id: tel-04169703

<https://theses.hal.science/tel-04169703>

Submitted on 24 Jul 2023

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

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



Université
de Toulouse

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par :

Institut National Polytechnique de Toulouse (Toulouse INP)

Discipline ou spécialité :

Informatique et Télécommunication

Présentée et soutenue par :

M. ILIA TARASOV

le lundi 1 février 2021

Titre :

Approches de résolution efficaces et estimations d'incertitude pour le problème de lissage de ressources

Ecole doctorale :

Mathématiques, Informatique, Télécommunications de Toulouse (MITT)

Unité de recherche :

Département d'Ingénierie de Systèmes Complexes (DISC-ISAÉ)

Directeurs de Thèse :

M. ALAIN HAÏT

MME OLGA BATAÏA

Rapporteurs :

M. FRANÇOIS CLAUTIAUX, UNIVERSITE BORDEAUX 1

M. JEAN-CHARLES BILLAUT, UNIVERSITE DE TOURS

Membres du jury :

M. ALEXANDRE DOLGUI, ECOLE NLE SUP DES MINES SAINT ETIENNE, Président

M. ALAIN HAÏT, ISAE-SUPAERO, Membre

M. ALEXANDER LAZAREV, V. A. TRAPEZNIKOV ICS RUSSIAN AS, Membre

MME OLGA BATAÏA, ISAE-SUPAERO, Membre

Efficient solution approaches and uncertainty estimation for the project resource leveling problem

Ilia Tarasov

Thesis defended February 1, 2021

Acknowledgments

At the very beginning of this extensive work, I could not even imagine how many interesting people I would meet. The incredible experience of working together and communicating with them is a great addition to the new knowledge and results achieved.

First and foremost, I wholeheartedly appreciate my supervisors for their guidance and motivations. Professors Alain Haït and Olga Battaïa helped me throughout the work on the thesis with both administrative questions and the research plan. Under their guidance and with their advice, I studied the subject in depth, while being able to choose the direction to explore. My thesis would not have been possible without their support and inspiration.

I am grateful to the members of the defense committee. I thank deeply Professors François Clautiaux and Jean-Charles Billaut for accepting to serve as the reviewers of my thesis and their attention to my work, and Professor Alexandre Dolgui for chairing the committee. I would like to especially thank Professor Alexander Lazarev for his valuable consults and help.

I would like to extend my gratitude to all my collaborators. I appreciate the team of the ISAE Department of Complex Systems Engineering. It was great to do my research in such a warm atmosphere, and I had many productive and fascinating conversations on research and life. I thank Franco Peschiera for very helpful discussions during our work.

My thanks go to my dear family and friends, who have always supported me, no matter how far away they are from me. Vladimir, my old friend, gave me wise advice many times. The lively and erudite Alexey showed me an example of how to overcome difficulties and get through a challenging third year of work. I would like to give my special thanks to Dmitrii. He inspired me to start this research work and helped me a lot at the beginning with organizational issues.

Contents

Résumé en français	7
1 Introduction	14
2 Project planning and scheduling: State-of-the-art	20
2.1 Resource-Constrained Project Scheduling Problem	21
2.1.1 Problem description	21
2.1.2 Mathematical formulations	23
2.2 Resource Leveling and Allocating	28
2.2.1 RLP objective functions and constraints	28
2.2.2 Evolution of RLP models	29
2.3 Solution methods	30
2.3.1 Branch and Bound/Branch and Cut	30
2.3.2 Benders decomposition	32
2.3.3 Constraint programming	35
2.3.4 Heuristics	36
2.4 Uncertainty modeling in project planning	36
2.4.1 Sensitivity analysis	37
2.4.2 Metric approach in scheduling theory	39
3 A generalized RLP: MILP formulation	43
3.1 Flexible resource allocation in RLP	44
3.1.1 Illustrative examples	44
3.1.2 Problem statement	45
3.2 Mathematical Model	47
3.2.1 Problem data	47
3.2.2 Generalized Model description	49
3.3 Generalized model properties	54
3.3.1 Model instance	54
3.4 Computational experiments	55

3.4.1	Tests on existing benchmarks for RLP	55
3.4.2	Generation of new instances	58
3.4.3	Comparison of time-indexed RLP formulations	60
3.4.4	Results of flexible resource allocation	61
3.4.5	Discrete Resource Case	64
3.5	Conclusions	66
4	Benders decomposition for RLP	67
4.1	Basic approach	67
4.2	Benders decomposition for a generalized RLP	70
4.2.1	Subproblem	70
4.2.2	Master problem	71
4.3	Algorithm improvements	73
4.3.1	LB for the Master problem: occupied work volume estimation . . .	73
4.3.2	LB for the Master problem: resource constraints relaxation	74
4.3.3	Disaggregation of the cuts	74
4.3.4	Branch&Benders cuts: using single search tree	75
4.4	Computational tests	77
4.4.1	Models and settings	77
4.4.2	Generation of specific large-scale datasets	78
4.4.3	Impact of number of resource types on the method performances . .	80
4.4.4	Evaluation of the impact of the available resource limit	81
4.5	Conclusions	83
5	Uncertainty and Metric approach	85
5.1	Notations, definitions and goals	85
5.2	Fluctuations in parameters	87
5.2.1	Extra resource cost changes	88
5.2.2	Fluctuations in available resource levels	90
5.2.3	Estimations for the job-related parameters	92
5.3	A generalized metric estimation	97
5.4	Feasibility and properties of the space of problem instances	100
5.5	Experiments	103
5.5.1	Datasets	103
5.5.2	Results	104
5.6	Conclusions	108
6	General conclusions	109

Résumé en français

Introduction

Cette thèse se concentre sur le développement des méthodes d'optimisation afin d'améliorer les processus de planification de ressources et d'ordonnancement de tâches dans des projets complexes. Bien que les plans ne soient pas toujours faciles à réaliser sans déviation en pratique, le processus de planification est nécessaire pour toute activité complexe. La modélisation mathématique de la prise de décision permet de rechercher de meilleures solutions. Le développement de telles modèles et méthodes est le sujet principal de la recherche opérationnelle (RO) et de l'optimisation. La digitalisation des processus et les systèmes informatiques actuels facilitent l'expansion de ces méthodes en pratique, car la complexité de la planification et de l'ordonnancement dans un environnement dynamique et incertain rend indispensable l'utilisation des outils d'aide à la décision appropriés dans de nombreux secteurs notamment fabrication, construction et transport. Dans ce qui suit nous présentons les éléments principaux de l'état de l'art.

Problème de planification des projets à ressources limitées

Le problème est défini comme suit. Les modèles existants peuvent différer dans les fonctions objectif et les contraintes, mais impliquent généralement les éléments suivants :

- l'ensemble de tâches J (activités), chacune ayant une durée p_j ;
- l'ensemble de ressources R (opérateurs, machines, etc.), chacune ayant un montant limité c_r , $r \in R$;
- une tâche donnée $j \in J$ nécessite un montant donné a_{jr} de la ressource $r \in R$ pour sa réalisation ;
- les relations de priorité précisant l'ordre d'exécution acceptable de tâches et éventuellement les décalages souhaitables entre elles sous forme d'ensembles $Prec(j)$ et $Succ(j)$, $j \in J$;
- les contraintes liées à la disponibilité des ressources ;

- l'objectif est de construire un calendrier (c'est-à-dire de définir les dates de début de toutes les tâches $S_j, \forall j \in J$) qui minimise le temps d'exécution du projet en tenant compte des contraintes définies. Si le moment d'achèvement d'une tâche est défini comme $C_j = S_j + p_j$, l'objectif correspondant à la fin du projet peut être exprimé de la manière suivante :

$$C_{max} = \max_{j \in J} C_j \rightarrow \min.$$

Les ressources peuvent être renouvelables ou pas. Les ressources renouvelables seront à nouveau disponibles après l'achèvement de l'activité dans laquelle ils sont impliqués, par exemple, des équipements ou opérateurs. Les ressources non renouvelables, par exemple les matériaux consommables, comme leur nom l'indique, seront consommés pendant l'exécution d'une tâche.

Pritsker a proposé la première formulation mathématique de ce problème [1]. Blazewicz a démontré que ce problème est NP-difficile au sens fort [2]. Les approches de résolution les plus courantes sont basées sur la modélisation MILP [3, 4]. Des algorithmes Branch and Bound ont été développés [5] ainsi que des algorithmes de Branch and Cut (B&C). La décomposition de Benders a aussi été utilisée pour des cas pratiques de problèmes de planification [6, 7, 8, 9]. Sont également fréquemment utilisées les méthodes heuristiques avec des solutions sous-optimales [10, 11, 12], ou la programmation par contraintes (CP) [13].

Problème de nivellement des ressources

Le problème du nivellement des ressources (RLP) est complémentaire au RCPSP, ce problème considère la fin du projet comme une contrainte et optimise l'utilisation des ressources nécessaires pour la respecter. Rieck et Zimmermann [14] ont analysé trois types de fonctions objectif les plus utilisés pour ce problème ($c_r > 0$ est le coût par unité de ressource $r \in R$ et U_{rt} est l'utilisation de la ressource $r \in R$ sur la période $t \in T$) :

- minimiser la somme des variations (lissage de l'utilisation U_{rt} de la ressource $r \in R$ sur des périodes $t \in T$)

$$F_v = \sum_{r \in R} c_r \sum_{t \in T} U_{rt}^2. \quad (1)$$

- minimiser les changements entre les périodes

$$F_a = \sum_{r \in R} c_r \sum_{t \in T} \max\{0, U_{rt} - U_{r,t-1}\}. \quad (2)$$

- minimiser la surcharge totale (si l'utilisation des ressources U_{rt} dépasse la limite des ressources Y_r prévues initialement, $r \in R, t \in T$)

$$F_o = \sum_{r \in R} c_r \sum_{t \in T} \max\{0, U_{rt} - Y_r\}. \quad (3)$$

Dans cette thèse, nous nous concentrons sur le modèle RLP ayant pour objectif de minimiser l'utilisation de ressources supplémentaires, c.f. eq. (3).

A la différence de la formulation classique, nous considérons que la durée de la tâche dépend de la quantité de la ressource qui peut être allouée pour l'exécution de la tâche. Plus cette quantité est importante, plus vite la tâche sera terminée. En pratique, ça permet de modéliser les projets avec des ressources telles que l'énergie, la chaleur, la capacité de transfert de données, le temps CPU, etc. Par exemple, un problème de planification de consommation énergétique pour une usine de fabrication de tuyaux a été étudié par Artigues et al. [15]. Nattaf et al. [16] ont également fourni de nouveaux résultats polyédriques pour les modèles MILP avec allocation de ressources énergétiques. Un autre exemple vient de l'industrie spatiale, la gestion du transfert de données du réseau de satellites aux stations optiques au sol nécessite une allocation de paires satellite-station et une planification du transfert dans des fenêtres de temps limitées, en utilisant un tampon de données et en ajustant la vitesse du transfert [17].

Précédemment, Baydoun et al. ont étudié un problème proche et ont développé le modèle du problème de minimisation des coûts de surcharge avec des contraintes de priorité et de chevauchement des tâches [18] qui est également similaire au problème présenté par Bianco et al. dans [19]. Baydoun et al. ont enrichi l'idée de la fonction de chevauchement avec plusieurs options spécifiques (par exemple, le chevauchement est autorisé si une partie de l'emploi précédent a déjà été mise en œuvre). Un autre problème proche, appelé RCPSP agrégé par période avec des contraintes de ressources considérées globalement sur des périodes de temps, a été décrit par Morin et al. [20], avec la fonction objectif de minimiser makespan, donc du RCPSP, qui est la principale différence avec notre étude.

Modélisation de l'incertitude

Dans une grande partie des applications de la vie réelle, le processus de planification des projets se heurte à des données inexactes ou à des changements inattendus. L'approche la plus courante consiste à représenter les données incertaines comme une valeur aléatoire et à appliquer certains modèles de la théorie des probabilités. Nous nous référons à l'article de Bruni et al. [21] avec une analyse des méthodes stochastiques pour un RCPSP. Une alternative consiste à construire une solution robuste (acceptable pour tout scénario) ou à construire une solution de base, adaptable à tout scénario possible. Un cas de RCPSP robuste avec un ensemble de scénarios impliquant des durées diverses a été présenté par

Artigues et al. [22]. Hazir et al. [23] ont formulé trois modèles d'optimisation robuste pour un problème de compromis temps discret/coût avec une structure similaire. Ces approches sont également qualifiées de réactives et proactives. Hazir et Ulusoy ont publié une récente revue de la littérature [24] sur la modélisation de l'incertitude dans la gestion de projets.

Une approche inverse est l'analyse de sensibilité qui étudie la réaction du modèle LP ou MILP aux changements de paramètres [25, 26]. Dawande et Hooker ont présenté une méthode générale d'analyse de sensibilité pour le MILP [27]. Cette méthode a ensuite été développée par Jia et Ierapetritou [28, 29]. Emelichev et al. [30] ont également étudié un vecteur LP combinatoire et le cas des perturbations des coefficients des fonctions linéaires et ont proposé un rayon de stabilité comme fonction. Plus tard, cette approche a également été utilisée par Gurevsky et al. [31] pour un problème d'équilibrage de chaîne de montage, fournissant des mesures de stabilité et de faisabilité.

Lazarev [32] a présenté une approche de résolution approximative pour les problèmes d'ordonnancement basée sur l'approche métrique. Il s'agit d'un schéma qui construit des solutions sous-optimales avec une précision estimée garantie dans un temps polynomial pour un problème qui est *NP*-difficile en général mais qui a quelques sous-classes d'instances polynomialement solvables. L'idée principale est d'utiliser une solution d'une instance "proche" solvable comme approximation de la solution pour une instance arbitraire. L'instance solvable "la plus proche" est trouvée à l'aide d'une fonction métrique. Elle montre une précision de cette approximation, c'est-à-dire une limite supérieure de la différence de la valeur d'une fonction objectif entre cette solution sous-optimale et une vraie solution optimale.

Il y a donc deux étapes. Tout d'abord, l'instance solvable la plus proche est construite avec une fonction métrique. Et deuxièmement, un algorithme polynomial correspondant est appliqué pour résoudre cette instance. La méthode originale fonctionne si plusieurs conditions sont remplies :

- il est possible de fournir des estimations efficaces de la précision lorsque nous utilisons la solution optimale d'une instance comme solution sous-optimale pour une autre instance ;
- il existe des sous-classes d'instances polynomialement solvables, qui fournissent des solutions de bonne précision à toute instance arbitraire de même taille.

Cette dernière approche a été testée sur certains problèmes d'ordonnancement, plusieurs exemples ont été présentés dans [32], notamment pour le problème de la minimisation de la somme des retards [33] et le problème d'ordonnancement sur une seule machine [34]. Dans cette thèse, nous étudions l'utilisabilité de cette méthode pour notre problème qui est présenté ci-dessous.

RLP généralisée : une nouvelle formulation MILP

Dans cette thèse, nous proposons une nouvelle généralisation de la formulation du RLP minimisant les coûts supplémentaires d'utilisation des ressources proposée par Baydoun et al. [18] et Bianco et al. [19]. Toutefois, notre modèle permet de gérer les contraintes de précedence sans problème de chevauchement. Une autre différence et nouveauté de notre modèle réside dans la manière d'affecter les ressources. Notre formulation permet une allocation plus souple grâce à l'introduction de variables de décision supplémentaires. Les modèles antérieurs utilisent une seule variable de décision pour chaque tâche et chaque période, présentée comme une fraction de la charge de travail [18, 19]. La quantité de toutes les ressources allouées à cette tâche durant cette période est définie par cette variable de décision.

Toutefois, lorsque plusieurs ressources sont nécessaires pour exécuter une tâche qui dure plus d'une période, dans le cas général, elles peuvent être utilisées à un rythme différent pendant une même période. Dans ce cas, l'affectation des différents types de ressources avec un coefficient proportionnel peut restreindre l'espace de solutions. Au contraire, la liberté d'affectation complémentaire peut permettre de mieux gérer le dépassement des quantités des ressources disponibles. Nous avons proposé un nouveau modèle mathématique pour cette formulation en utilisant l'indexation sur le temps. Des expériences réalisées avec le solveur CPLEX 12.8 ont montré que l'allocation indépendante des ressources permet de réduire les coûts de 7 % en moyenne pour les instances utilisées qui sont des instances de référence. Comme ce nouveau modèle emploie plus de variables, la résolution est légèrement plus longue. Des tests supplémentaires sur les instances générées aléatoirement ont montré le potentiel du modèle pour certaines instances où les coûts ont été réduits de 50 % en moyenne.

Cette première contribution a également été présentée dans les publications suivantes :

- Tarasov, I. ; Haït, A. ; Battaïa, O. *Period-Aggregated Resource-Constrained Scheduling Problem with variable job duration*. 19ème Congrès annuel de la Société Française de Recherche Opérationnelle et d'Aide à la Décision (ROADEF 2018), Feb 2018, Lorient, France. ;
- Tarasov, I. ; Haït, A. ; Battaïa, O. *A Generalized MILP Formulation for the Period-Aggregated Resource Leveling Problem with Variable Job Duration*. *Algorithms* 2020, 13, 6.

Algorithme de décomposition de Benders pour RLP

L'algorithme de décomposition de Benders sépare le problème initial en un problème principal MILP et un sous-problème LP. Les sous-problèmes LP utilisent les solutions

partielles du problème maître pour construire des coupes supplémentaires. Pour notre problème RLP généralisé, nous avons élaboré un ensemble d'améliorations de l'algorithme tant au niveau structurel qu'au niveau de l'implémentation technique. Un point important est l'utilisation d'un seul arbre de recherche au lieu de plusieurs exécutions du problème maître (appelé Branch&Benders cuts). Les inégalités valides éliminent les solutions partielles du problème maître non réalisable et réduisent donc le nombre d'itérations. Les bornes inférieures pour le problème maître accélèrent la convergence de la résolution.

Nous avons effectué des expériences numériques avec le solveur CPLEX 12.10. Notre algorithme a été comparé à la décomposition de Benders du solveur intégré et à la norme B&C. Les résultats des expériences numériques montrent la supériorité de notre algorithme. Il surpasse de manière significative la décomposition de Benders intégrée et la norme B&C, en particulier avec un nombre croissant de types de ressources. Ainsi, il est plus adapté pour la résolution des instances de taille plus importante. Par exemple, il a résolu de manière optimale des instances avec 50 tâches et 10 types de ressources.

Cette deuxième contribution a également été présentée dans les publications suivantes :

- *Tarasov, I. ; Haït, A. ; Battaïa, O. Benders decomposition algorithm for a generalized Resource Leveling Problem. 21ème Congrès annuel de la Société Française de Recherche Opérationnelle et d'Aide à la Décision (ROADEF 2020), Feb 2020, Lorient, France. ;*
- *Tarasov, I. ; Haït, A. ; Battaïa, O. Benders decomposition for a Period-Aggregated Resource Leveling Problem with variable job duration. Computers & Operations research, submitted, passing second minor revision.*

Incertitude des données et approche métrique

Nous proposons une approche métrique pour notre problème RLP généralisé. Cette approche cherche à répondre à la question suivante : que devrait faire un décideur si au moment de l'application de la solution optimale les données d'entrée observées ne sont pas exactement les mêmes que celles utilisées lors de l'optimisation ? Nous avons suggéré deux options : conserver la solution complète ou uniquement l'ordonnancement des tâches. La deuxième option présente plusieurs avantages. Elle nous permet d'adapter l'allocation des ressources à des nouvelles conditions grâce à des calculs en temps polynomial (avec le sous-problème LP). Nous développons des métriques permettant d'estimer l'écart en terme de la valeur de la fonction objectif obtenue et la valeur optimale.

Les expériences numériques ont été réalisées pour le cas de la disponibilité des ressources en supposant une variation possible entre 10 et 100 %. La faisabilité et l'optimalité de la solution initiale a été conservée pour 30 à 45 % des instances avec une seule

variation, et jusqu'à 15 % des instances avec plusieurs variations.

Cette approche a également été présentée dans la publication suivante :

- *Tarasov, I. ; Haït, A. ; Battaïa, O. Metric Estimations for a Resource Leveling Problem With Variable Job Duration. 17th International Workshop on Project Management and Scheduling (PMS) 2020/2021 (postponed), submission accepted.*

Perspectives de recherche

Dans cette thèse, nous avons principalement utilisé la modélisation MILP, il sera intéressant de comparer les résultats obtenus avec les résultats fournis par des approches basées sur la programmation par contraintes (CP).

En ce qui concerne le sujet de la variation des données, il sera intéressant d'étudier l'impact des modifications dans les relations de précédence. Si dans ce cas, l'optimalité ne pourra pas être garantie pour les données modifiées, cela appellera au développement d'une méthode efficace de replanification. Dans le contexte où des replanifications sont possibles, les approches employant des méthodes d'apprentissage automatique semblent avoir du potentiel à explorer.

Chapter 1

Introduction

Plans are worthless, but planning is everything.

Dwight D. Eisenhower

Although the plans are not always achievable, the planning process is a mandatory part of everyone's life. Our performance highly depends on effective planning. The planning of large projects for organizations is much more difficult. The impact of planning quality also scales in this case. Planning is the process that involves defining a goal and the instruments, methods, and set of actions required to achieve it. The planning process is valuable because it forces a clear set of goals and allows one to analyze them.

Two terms are close, but they are not equal: planning and scheduling. Planning is more general and answers the questions “what?”, “how?” and potentially “why?”. We dedicate scheduling to the question “when?”. Scheduling is the timetabling of steps under the plan. These terms always appear together, because they define each other. There should be no inconsistencies, as the effective plan requires an effective and achievable schedule.

This research focuses on the mathematical methods that improve planning and scheduling processes. Mathematical modeling of the decision-making allows us to get complex results for which we can justify and guarantee their advantage over other options. Mathematical apparatus and modern computing systems make it possible to implement these methods more and more widely. It is the principal subject of Operations Research (OR) and Optimization.

In this chapter, we briefly discuss the evolution of planning techniques, models, and methods. Then, we will illustrate the effect of implementing these methods and some results, with the motivation to the current research.

At the beginning of the 20th century, people performed manually the planning and scheduling. It required fewer efforts. The development of science and technology significantly accelerated the pace of life in all spheres: the volume, complexity, and scale of

production increased. The pace of everyday life and the speed of people’s movement also sped up.

The planning and scheduling included an increasing number of elements that had to be organized over time, operated with more and more data and parameters of interconnected operations. It became increasingly difficult to make high-quality planning in many industries such as manufacturing, construction, and transportation. We associate the emergence of research in planning and management with these factors.

Naturally, there were attempts to systematize knowledge and use a scientific approach to the organization of work. American scientists Henry Gantt and Frederick Taylor conducted one of the first studies on methods of production organization. Frederick Taylor is the founder of the scientific organization of labor and management. He described the fundamental aspects of this approach in his monograph [35]. Taylor was one of the first to point out the need to use a scientific approach to solving organizational problems, describing management as a science with its laws. He tried to present some of these rules on formalized mathematics language in the form of laws and equations.

Henry Gant published a series of scientific papers from 1903 to 1919 where he proposed a novel way of presenting schedules, named after him: “Gantt Chart” [36]. It is a schematic representation of a schedule that presents jobs as rectangles placed along the time axis, with the length of the rectangle corresponds to the time required to perform the corresponding job. Nowadays it is well-known and widely used in many areas as a standard for visualizing sequences of operations.

Kelley and Walker [37] developed a modeling technique to test the duration of the project, named as Critical Path Method (CPM). The project has a set of project steps (operations) with a defined duration. It is also a set of events (milestones). A project network graph represents the ordering requirements (precedence relations). This method calculates the longest path within the project network from the start to the end.

In the Fig. 1.1 there is an example of a project Gantt chart with an illustration of the CPM. Precedence relations form the project network. We also mark them in the Gantt chart as green vertical lines. Some critical operations determine the project duration (colored in orange). If they change their duration or start time, it will change the project duration. In contrast, other operations (colored in blue) might float in some range (grey-colored zone) with no impact on the overall duration.

Malcolm et al. [38] introduced the Project Evaluation and Review Technique (PERT) for the U.S. Navy. It has the same concept; it represents the project as a network formed by activities and milestones. CPM is used to estimate the duration of the project with additional statistical analysis. PERT exploits three processing time estimations for each operation: optimistic, expected, pessimistic. Based on that, PERT helps to validate the plans and to estimate the risks caused by the uncertainty, i.e. the probability of meeting

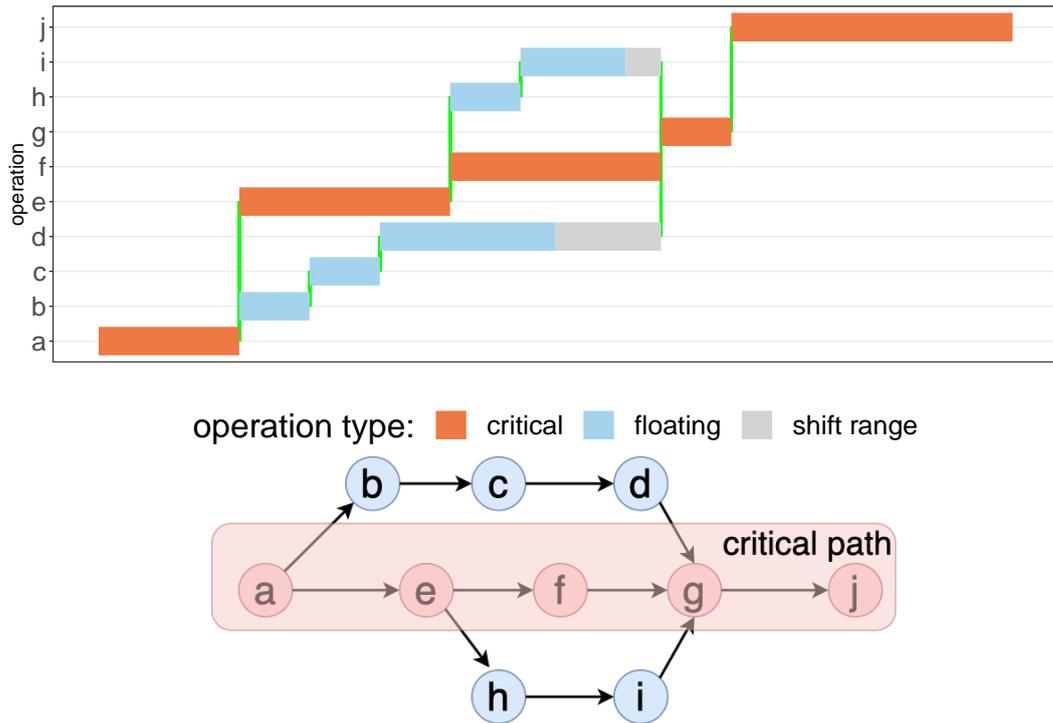


Figure 1.1: An example of Gantt chart and CPM diagram.

an existent schedule. It was a bridge from describing analytics to predictive modeling.

Mathematical methods supplemented analytical tools to perform planning in different cases. It started with the so-called job shop scheduling problem, described as follows. We define a set of jobs (orders, products), each job comprises operations, and a set of devices (machines, processors) process these operations. They can process a job with only one machine. There might be a route with the order of operations. Besides, we know the duration of all operations. A special case with a defined operation processing order is named flow shop scheduling. This problem applies to the lean assembly lines. It is necessary to define the schedule, i.e. assign the order of operations. Bellman introduced the term “scheduling theory” in 1956 [39] and Conway et al. [40] published the first book on scheduling.

This kind of combinatorial problem was classified in terms of complexity with the notions of P and NP classes, NP-completeness and NP-hardness. Cook and Levin proved that the Boolean satisfiability problem is NP-complete [41, 42]. Karp [43] introduced 21 combinatorial problems that are also NP-complete.

Gradually, the researchers enriched scheduling problem formulations in line with practical challenges. They expanded the models with the notions of operational resources and requirements. The concept of a project appeared, Pritsker et al. [1] linked it with the job-shop scheduling terms. The project has several defined elements:

- the set of resource types with limited availability;

- the set of jobs with resource requirements, precedence relations, and due dates;
- the objectives might be varied:
 - to minimize the project completion time (makespan);
 - to minimize the total completion time for all jobs;
 - to minimize total lateness (exceeded due dates);

The makespan minimizing problem was studied as a basic formulation. It was defined as a Resource-Constrained Project Scheduling Problem (RCPCP).

Next, financial and resource management aspects were studied in different branches [44]. For example, the net present value (NPV) maximization problem counted negative or positive cash flow (a payment or a profit) to the operations. Resource availability cost problem (RACP) minimized the cost of used resources with a strict project deadline constraint. Resource Leveling Problem (RLP) managed the variation of resource usage over time periods.

Project scheduling problems were proved to be NP-hard. So, building solutions to these problems have become a serious challenge and remains so to this day. Most common approaches are based on MILP modeling [5, 3, 4], heuristic methods with suboptimal solutions [10, 11, 12], or constraint programming (CP) [13].

Machine Learning (ML) technology is increasingly used in OR. However, it was less frequently adapted for combinatorial planning problems. And there are only a few results close to the project scheduling topic. We can refer to the review of ML in production planning and control by Cadavid et al. [45] and applications for maintenance scheduling by Peschiera et al. [46]. There were several ML applications for other areas: facility locations by Lodi et al. [47], offshore wind parks by Fischetti and Fracaro [48].

The topic of data uncertainty was also developed for project planning [24]. There are several directions. Stochastic optimization expands the initial problem with statistical analysis [21]. Robust optimization gives priority to the stability and acceptability of the solution in case of accidents. Usually, the space of possible changes is considered as a finite set of scenarios [22]. There are several results for RCPSP and RLP [49, 50]. The importance of modeling data uncertainty is attested by industry case reviews that show that most of the projects overrun their budgets and violate pre-defined deadlines simultaneously. It is especially notable for large construction projects. The study¹ of McKinsey & Company illustrated that for mining, oil, and infrastructure projects with a budget of more than one billion dollars, see Fig. 1.2.

¹The construction productivity imperative <https://www.mckinsey.com/business-functions/operations/our-insights/the-construction-productivity-imperative#>

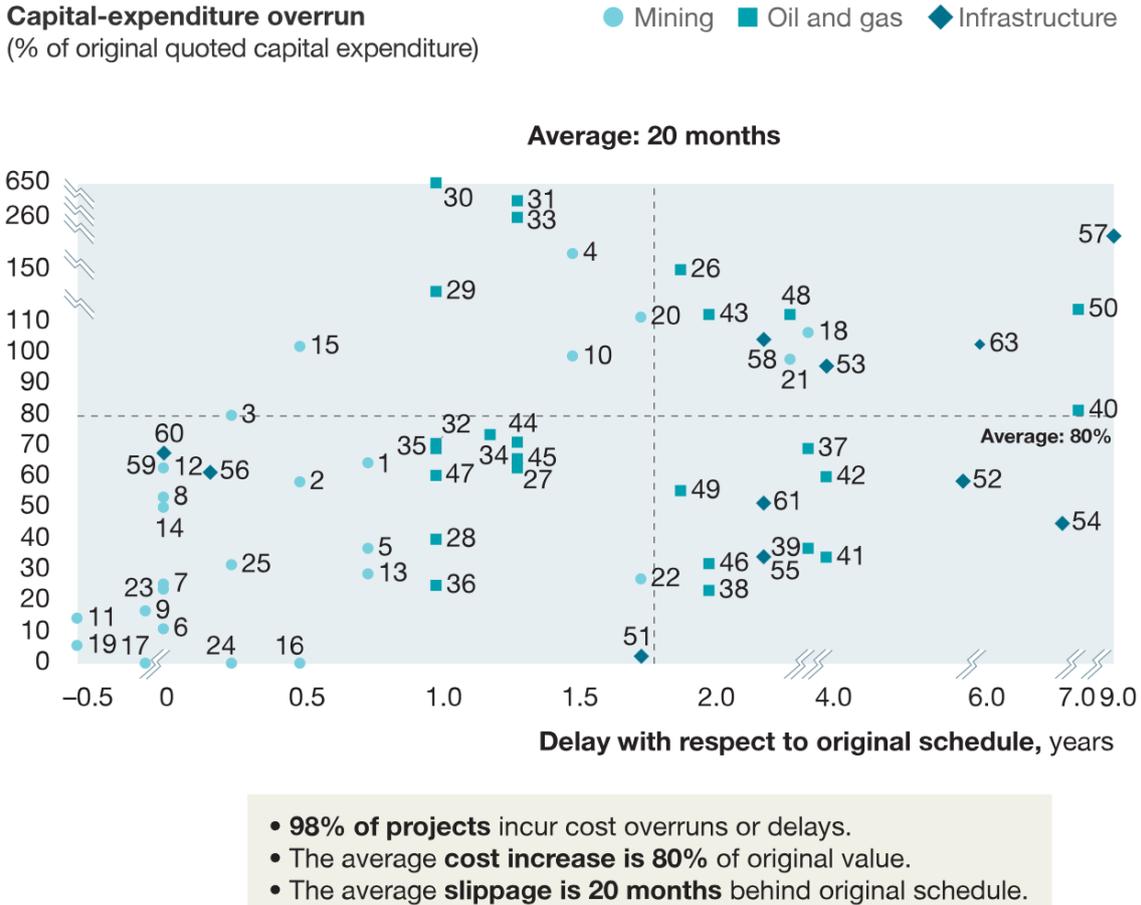


Figure 1.2: Cost overruns and delays of 1B\$+ construction projects. Source: McKinsey&Company.

In this thesis, we study the problem of the overrun of budgets. The consideration of the flexible assignment of resources may help to avoid management risks and reinforce the most important milestones and critical resources. For this purpose, we present a new formulation for a generalized Resource Leveling Problem (RLP) and develop solution methods for it. The thesis is organized in the following manner.

Chapter 2 presents a state-of-the-art for RCPSP and RLP. We discuss the solution methods and uncertainty modeling.

Our first contribution, a new mathematical formulation for a generalized RLP is presented in Chapter 3. The idea behind this formulation is to allow a more flexible assignment of resources and thus to produce solutions with reduced extra costs.

Chapter 4 describes our second contribution, which is the Benders decomposition algorithm that effectively solves the problem we introduced. It involves several improvements designed to accelerate the resolution.

Chapter 5 is dedicated to our third contribution which is a metric approach to manage the data uncertainty for the considered generalized RLP. The realized theoretical study

provides the estimations that allow us to analyze the consequences of data variations and make partial rescheduling.

Finally, Chapter 6 presents general conclusions, proposals, and perspectives for future research.

Chapter 2

Project planning and scheduling: State-of-the-art

Project Scheduling is represented in combinatorial optimization by two principal problems: Resource Constrained Project Scheduling Problem (RCPSP) and Resource Leveling Problem (RLP). The former seeks to minimize the duration of the project and the latter minimizes the resource usage required to respect a fixed project deadline. Both of them are well-known combinatorial optimization problems actively studied in the literature. The existing models differ in the objective functions and the constraints, but usually involve several entities:

- a set of jobs (activities) which must be processed;
- a set of resources (workers, machines, etc.) required to implement the jobs;
- precedence relations that specify the acceptable execution order and time lags;
- resource-related constraints caused by limited available resources;
- an objective function, the mathematical formulation of the goal used to evaluate the quality of the solutions.

A solution must specify the scheduling and resource-job assignments (in particular cases).

In the following, we discuss RCPSP first, since many modeling general project scheduling notations, concepts, and solution approaches were firstly introduced for RCPSP. Then, we focus on RLP which generalization will be introduced and studied in this thesis. In the end, we discuss the management of data uncertainty and how it is covered in combinatorial optimization in general and particularly in project scheduling and resource leveling.

2.1 Resource-Constrained Project Scheduling Problem

2.1.1 Problem description

The RCPSP is a well-known problem that arises in many real-life planning problems in various industries. Its original formulation is as follows: a set of jobs has to be executed without preemption, each job requires renewable resources of limited capacity and has some precedence constraints.

The basic goal for this problem type is to construct a schedule that minimizes the project duration under resource constraints. A schedule might be represented in different forms. Usually, the starting moment of job j is noted as $S_j, \forall j \in J$. The duration of job j is usually denoted as $p_j, j \in J$. In the basic formulation, there is a fixed job duration and it is not allowed to interrupt the jobs. Using these notations, the objective might be formulated as the completion time of the last job (also defined as a makespan). The completion time of job $j \in J$ is directly derived from starting time S_j :

$$C_j = S_j + p_j$$

A feasible solution must satisfy all the constraints namely the precedence and resource constraints. Precedence constraints, i.e. the ordering relationship between the jobs, are usually represented as a set of predecessors or successors, denoted as $Prec(j)$ and $Succ(j)$, $j \in J$, respectively. An alternative is to define it as a set of pairs $(j, i) \in P$. So there is a corresponding directed acyclic graph of precedence $G = (V, A)$, where the vertices of $V = j_1, \dots, j_n$ are the jobs of set J . Arcs $A = \{(i, j) | i, j \in V; i \rightarrow j\}$ form the structure of the precedence constraints. Besides, so-called dummy jobs have been introduced to simplify the calculation of the makespan objective function. These are the "zero" and " $n + 1$ " jobs, which have a processing time of zero and require no resources. Basically, the "zero" job is a project start pointer and is executed first, and the " $n + 1$ " is representing a project finish and so it is executed last. This is achieved with additional precedence relations between these dummy jobs and all other jobs $j \in J$. The makespan (the project completion time) is denoted as C_{max} , and in this case, it is determined by the completion time of " $n + 1$ " job:

$$C_{max} = \max_{j \in J} C_j = C_{n+1}.$$

The planning horizon T represents the time interval available for the project completion. Resource constraints might be divided into two groups: renewable resources (which will be available again after the completion of a job in which they are involved, i.e. equipment and workforce) and nonrenewable resources (materials, fuel, etc.). Resources may be of different types (i.e., skills, type of material, type of equipment) and a given number of resources of a given type might be required to perform a given job. The set of resources

is usually denoted as R . A job $j \in J$ requires a specific amount a_{jr} of resource $r \in R$ for its completion. For the renewable resources these constraints are formulated as follows: for a schedule $\pi = \{S_1, \dots, S_n\}$, in any point of time $t \in T$ and for any resource $r \in R$, the total occupied resource amount is less than the available limit c_r .

The complexity of RCPSP

Pritsker mathematically formalized the classical project management problem with the resource limitation [1], and its NP-completeness was later proved by Blazewicz [2]. In the three-position task designation scheme of Scheduling Theory proposed by Graham [51], the basic RCPSP is denoted as $PS|prec|C_{max}$. In this scheme, of the three characteristics $\alpha|\beta|\gamma$, the first one denotes the model of actors and resources, the second characteristic defines the model and conditions of the job, and the third characteristic points to the objective function. In this case PS – "project scheduling", $prec$ – short form for the precedence relationship, and the C_{max} is a makespan.

RCPSP might be polynomially reduced to the partition problem, which is evidence of the NP-hardness of the RCPSP. The definition of the NP-complete partitioning problem [52]. It is necessary to decide if for a given set of n integers $A = \{a_1, a_2, \dots, a_n\}$, there are two sets of indexes I and J , $I \cup J = \{1, 2, \dots, n\}$, $I \cap J = \emptyset$, such that

$$\sum_{i \in I} a_i = \sum_{j \in J} a_j. \quad (2.1)$$

There is a PSPLIB library [53, 54] for input test examples for which the best results are known.

Extensions of the basic model

The duration of a job may depend on the intensity of the job execution, determined by the number of resources used. More resources are used at the same time, the faster the job is completed. Different modes of execution can be defined. In the latter case, the RCPSP is called multimodal and is referred to as MRCPSp.

The jobs may also have specific time windows for their execution, defined between release times r_j and deadlines D_j (must be distinguished from the non-strict due dates d_j that are usually involved in the objective function). For project scheduling there is a more common notation of the earliest and the latest starting and completion times: EST_j , ECT_j , LST_j , LCT_j , respectively. These parameters were used in two ways. On the one hand, they were used as an input parameter involved in constraints with S_j and C_j . On the other hand, many preprocessing techniques use these values and tighten them, for example, to find the so-called compulsory intervals $[LST_j, ECT_j]$. With a given set of

precedence relations and the bounded planning horizon, these values might be calculated from the structure of the precedence graph and job durations.

The number of resources available at a given point of time may not be fixed by a constant c_r , in this case, a certain function $C_r(t)$, $r \in R$, $t \in T$ is introduced to define the value. This corresponds to a situation where non-renewable resources are being replenished, and workers and equipment can be available according to a specific schedule.

Precedence constraints may be considered in a generalized form with an additional arbitrary δ_{ij} parameter for any precedence relation $i \rightarrow j$, $(i, j \in J)$. This case is denoted as generalized precedence relations (GPR), that may involve the so-called start-to-start, start-to-finish, finish-to-start, and finish-to-finish precedence constraints with a minimal required δ_{ij} time lag between these events. In other words, this approach extends the basic idea that it is not allowed to start job j before job i was finished with a given $i \rightarrow j$ precedence constraint. The basic precedence formulation thus is a particular case of the GPR, described as a start-to-start precedence $i \rightarrow j$ with $\delta_{ij} = p_i$, or a finish-to-start constraint with zero δ_{ij} . These precedence relations were introduced in the pioneer publication of Korbush et al. [55] and later studied by Neumann and Schwindt [56]. Bartush et al. [57] used a notion of temporal constraints and proposed the notation of temporal constraints with minimal and maximal time lags d_{min}^{ij} and d_{max}^{ij} .

2.1.2 Mathematical formulations

Mixed Integer Linear Programming (MILP) is one of the most common ways to formulate RCPSP concurrently with Constraint programming (CP). In the following, we discuss the most used formulations for RCPSP. We refer to the RCPSP MILP formulations review of Artigues et al. [22] for a deeper description of these models and their properties.

Time-indexed formulations

From the pioneer studies of Pritsker and Watters [58] and Pritsker et al. [1], this approach was used to formulate the scheduling problems with a planning horizon divided into discrete periods, including the RCPSP. The models with time-indexed variables involved 0-1 decision variables as a pointer to the status of the job. Basically, with binary variables, there are two options: to point out that the job was started or not in the particular time period (and possibly the same that it was finished) or to point out that the job is in progress in a given time period. If the start-end pointers are used, there are also several options to describe them in the model. We will use the notions from Artigues et al. [22] to illustrate these concepts and the difference. So, there are three basic approaches:

- "pulse" variables x_{jt} , $j \in J$, $t \in T$, such that $x_{jt} = 1$ if and only if activity j starts at time period t ;

- "step" variables ξ_{jt} , $j \in J$, $t \in T$, such that $\xi_{jt} = 1$ if activity j was started at time period t or before;
- "on-off" variables μ_{jt} , $j \in J$, $t \in T$, such that $\mu_{jt} = 1$ if activity j is processed at time period t .

For pulse variables x_{jt} we represent the starting time period S_j of activity $j \in J$ as a

$$S_j = \sum_{t \in T} tx_{jt};$$

so, with a dummy "n+1" job with a zero processing time (meaning that $C_j = S_j$), the makespan objective function is represented as

$$\text{Min} \sum_{t \in T} tx_{n+1,t}.$$

and the constraints for the precedence relations and the resource requirements:

$$\begin{aligned} \sum_{t \in T} tx_{jt} - \sum_{t \in T} tx_{it} &\geq p_i; \quad \forall (i, j) \in A \\ \sum_{j \in J} \sum_{\tau=t-p_j+1}^t a_{jk} x_{j\tau} &\leq c_r; \quad \forall t \in T; \quad \forall r \in R \\ \sum_{t \in T} x_{jt} &= 1; \quad \forall j \in J \\ x_{jt} &\in \{0, 1\}; \quad \forall j \in J; \quad \forall t \in T \end{aligned}$$

This formulation might also involve the constraints that use the earliest and latest time intervals, for example:

$$x_{it} = 0; \quad \forall j \in J; \quad \forall t \in T \setminus \{EST_j, \dots, LST_j\};$$

and the precedence constraints might be rewritten in a form with a stronger LP relaxation, see the paper of Christofides et al. [59]:

$$\sum_{\tau=0}^{t-p_i} x_{i\tau} - \sum_{\tau=0}^t x_{j\tau} \geq 0; \quad \forall (i, j) \in E; \quad \forall t \in T.$$

Pritsker and Watter [58] compared step variables to pulse decision variables. Basically, "step" variables ξ_{jt} point to the starting period with a step change of values. If activity j was started at time period t or before, then $\xi_{jt} = 1$, otherwise $\xi_{jt} = 0$. The step decision variables ξ_{jt} might be produced from pulse variables $\sum_{\tau=0}^t x_{j\tau}$, and $x_{jt} = \xi_{jt} - \xi_{j,t-1}$. These decision variables require an additional set of constraints to define proper non-decreasing variables: $\xi_{j,t-1} \leq \xi_{jt}$, $\forall j \in J$, $\forall t \in T$. They are connected with the original start time

S_j of a job $j \in J$ in the following form:

$$S_j = \sum_{t \in T} t(\xi_{jt} - \xi_{j,t-1});$$

so the objective is built in a same way as for pulse variables ($S_{n+1} = C_{n+1}$):

$$\text{Min} \sum_{t \in T} t(\xi_{n+1,t} - \xi_{n+1,t-1}).$$

The constraints are redefined according to the expressions above. For example, the precedence relations have a form

$$\xi_{j,t-p_j} \geq \xi_{jt}; \quad \forall (i, j) \in E; \quad \forall t \in T.$$

For the "on-off" formulation we use the binary decision variable μ_{jt} , $j \in J$, $t \in T$, to illustrate that job j is being processed in time period $t \in T$ or not. So, $\mu_{jt} = 1$ if activity j is processed at time period t and $\mu_{jt} = 0$ otherwise. Based on that, $\forall j \in J$, $t \in T$,

$$\mu_{jt} = \xi_{jt} - \xi_{j,t-p_j} = \sum_{\tau=t-p_j+1}^t x_{jt}.$$

For dummy jobs with $p_j = 0$ $\xi_{jt} = \mu_{jt}$ and thus the objective function has a form

$$\text{Min} \sum_{t \in T} t(\mu_{n+1,t} - \mu_{n+1,t-1}).$$

And for other jobs it is necessary to consider another case, when $p_j \geq 1$, $\forall j \in J$. To get an expression for x_{jt} and ξ_{jt} , it is necessary to consider all the periods in a range of p_j , i.e. $\tau = t - \alpha p_j$, with $\alpha \in [0, \dots, \lfloor t/p_j \rfloor]$:

$$\xi_{jt} = \sum_{\alpha=0}^{\lfloor t/p_j \rfloor} \mu_{j,t-\alpha p_j}.$$

and with $x_{jt} = \xi_{jt} - \xi_{j,t-1}$ so

$$x_{jt} = \sum_{\alpha=0}^{\lfloor t/p_j \rfloor} \mu_{j,t-\alpha p_j} - \sum_{\alpha=0}^{\lfloor (t-1)/p_j \rfloor} \mu_{j,t-1-\alpha p_j}.$$

To define the starting period S_j , the same structure is used for x_{jt} :

$$S_j = \sum_{t \in T} t \left(\sum_{\alpha=0}^{\lfloor t/p_j \rfloor} \mu_{j,t-\alpha p_j} - \sum_{\alpha=0}^{\lfloor (t-1)/p_j \rfloor} \mu_{j,t-1-\alpha p_j} \right).$$

These basic formulations were modified and upgraded, remaining applicable to generalized problem formulations. For example, to deal with an RCPSP with various jobs' duration. Klein [60] presented another variable γ_{jt} . It is defined so that $\gamma_{jt} = 1$ if a job $j \in J$ is implemented at time period $t \in T$ or after, then $\xi_{jt} + \gamma_{jt} - 1 = \mu_{jt}$ for all jobs $j \in J$ and $\xi_{jt} + \gamma_{jt} - 1 = x_{jt}$ for dummy activities. The combination of these variables with ξ_{jt} makes it possible to construct a model with various job duration.

Time-indexed formulations were improved. Cutting plane inequalities improved these models, as it was stated in the papers of Demassez et al. [61] and Hardin et al. [62].

These models were also widely used and reviewed for particular cases of RCPSP. For example, see the study of Artigues [63], the RCPSP formulation with step variables presented by Bianco and Caramia [64], Bianco et al. [65]. Artigues [66] later clarified the results of polyhedral analysis and the strength of LP-relaxations for these models. Although it was noted that LP-strength is not guaranteeing better results for the corresponding MILP, it was stated that this strict consideration allows distinguishing a real improvement from an equivalent reformulation.

Naber and Kolish studied the time-indexed RCPSP models with varied resource availability [67]. Kyriakidis et al. [68] reviewed the MILP formulations of RCPSP with renewable and non-renewable resources, also with a model for the multi-mode case with a variety in the number of consumed resources. Kreter et al. [69] studied the mathematical formulations for RCPSP with GPR and calendars, with preemptions caused by calendar breaks. Javanmard et al. [70] considered a preemptive multi-skilled resource investment project scheduling problem with the same approach. The model involved a cost objective function and it was significantly more complicated with worker skill levels, it was noticed from the chemical production industry. Garcia-Nieves et al. [71] worked with a model from the construction industry with repetitive activities, considering multiple goals: the resource allocation, the makespan, or the costs.

There are also other ways to use time-indexes variables. A feasible subset approach was introduced by Mingozzi et al. [72]. The concept is to find the subsets of jobs that are allowed to be executed in parallel (i.e., without any precedence relations) with respect to resource constraints. With a set of feasible subsets, it is possible to construct a schedule in the following way. It is necessary to decide in each time period whether a given subset is implemented or not, so a binary decision with a time index is required to implement this technique. There is a drawback that the size of the general set might be exponentially large with respect to the job set size. This model is used with column generation to generate lower bounds [52].

Chain-decomposition formulation is another technique presented by Kimms [73]. With a baseline idea to decompose the graph into several chains, so that each precedence pair is involved in one chain once. Binary variables are used to assign a schedule (a permutation)

to a chain and to point out the completion time of a job in a given period within a given chain. The main drawback is also that the number of these variables is not polynomial, so these models are used with column generation to calculate the bounds.

Sequencing formulations

These formulations directly use a continuous variable of S_j to define the starting moment combined with binary variables. The binary variable may be used to deal with precedence relations only, or it may be also involved in the resource constraints. It was used for continuous-time planning horizons, for example, see the study of Queyranne and Schulz [74].

We can refer to the two forms.

- With a minimal forbidden set, a set of jobs that violate the resource limitations whereas any subset does not imply that. As in the previous examples, this approach states that the number of sets is exponential, which makes it difficult to use in practice. It was proposed by Alvarez-Valdés and Tamarit.
- The flow-based formulation uses the notion of resource flows to represent the demanded and returned amount of resources when a job is started or finished. The continuous decision variable is used to represent the flow coming from one job to another, so it is defined for all possible pairs. It was presented by Artigues et al. [75] and improved by Demassay [61].

We note that in later studies these two described methods were combined. Resource leveling related models involve a combination of continuous starts and binary variables that are used to assign resources correctly. For example, the model of Baydoun et al. [18], which is used to deal with a problem that is close to the problem considered in this work.

Event-based formulations

This concept exploits the idea that all possible events (the start and the end of the job) belong to a bounded set of time moments, the so-called set of events E . It might be achieved with a notion of active schedules when we state that all jobs start at the earliest moment that does not violate the constraints and the ordering of job processing. With given duration and release times, we can tighten the set of possible processing start and end times for any job $j \in J$. This concept is also implemented with different types of binary decision variables.

- Start-end events involves the binary variables $a_{j_e}^+$ and $a_{j_e}^-$ that are used to define an event $e \in E$ for the start and the end of job j processing. It was introduced by Koné et al. [76] and later precised by Artigues et al. in [77].

- On-off events were also presented by Koné et al. [76]. As for the time-indexed formulation, the binary decision variables a_{je} were introduced to point if job $j \in J$ was processed at event $e \in E$.

To sum up, there is a range of commonly used and well-studied practices to formulate the scheduling problems. The development of a new model for this kind of problem should begin with the analysis of close statements and the selection of the most promising models. There are also general modeling recommendations, for example, see the papers of Klotz and Newman [78] and Vielma [79].

2.2 Resource Leveling and Allocating

Another frequently studied problem formulation in project scheduling is the Resource Leveling Problem (RLP) aimed at controlling resource usage over time and avoiding overloads, undesired oscillations in resource usage, or minimizing total resource usage. These formulations were also proved to be NP-hard in the strong sense [80].

2.2.1 RLP objective functions and constraints

The main difference between RLP and RCPSP comes from the fact that they use different objective functions. Thus RLP uses supplementary decision variables related to resource allocation. In the paper of Rieck and Zimmermann [14], three classic types of objective function were analyzed: variations in resource utilization within project duration, total (squared) overload cost, and the total adjustment cost (costs arising from increasing and decreasing resource utilization). So, these objective functions are related either to resource usage oscillations or to the total resource usage. Within a time-indexed formulation they have the following form:

- total variations (smoothing resource $r \in R$ utilization U_{rt} over periods $t \in T$)

$$F_v = \sum_{r \in R} c_r \sum_{t \in T} U_{rt}^2. \quad (2.2)$$

- total adjustment (cumulative price of changes in utilization U_{rt})

$$F_a = \sum_{r \in R} c_r \sum_{t \in T} \max\{0, U_{rt} - U_{r,t-1}\}. \quad (2.3)$$

- total overload (if resource usage U_{rt} exceeded resource limit Y_r , $r \in R$, $t \in T$)

$$F_o = \sum_{r \in R} c_r \sum_{t \in T} \max\{0, U_{rt} - Y_r\}. \quad (2.4)$$

Here $c_r > 0$ is the cost per unit of resource $r \in R$.

In each case, there is a specified project deadline that is fixed. Resource assignment is performed with additional variables describing the resource usage profile of the schedule. It has a form of a step function U_{rt} defined on periods $t \in T$ for each resource $r \in R$.

2.2.2 Evolution of RLP models

Hans in [81] studied RLP with varying intensities of jobs resulting in their variable processing times. In this work, a branch and price approach was developed to solve this problem. Further, this model was considered by Kis [82], who constructed a Branch and Cut algorithm. He described the feasible intensity assignments polytope and outperformed the Branch and Price approach which is dual to B&C. In the model presented by Kis, the jobs were specified in discrete-time intervals, and the precedence relations were period-aggregated making impossible to assign two adjacent jobs to the same time period. These models were used to minimize resource overload.

For practical problems, researchers develop heuristic methods to construct acceptable solutions in a reasonable time. Christodoulou et al. [83] provided an overview of heuristic methods applied to RLP. There are several recently published studies inspired by industrial problems. For example, the construction resource leveling model was solved by meta-heuristic genetic solution approach by Selvam and Tadepalli [84]. Piryonesi et al. [85] developed a simulated annealing algorithm for the same type of applications. Li et al. [86] studied the RLP with uncertain job duration and possible overlapping and proposed a genetic algorithm. Cherkaoui et al. [87] studied tactical scheduling and resource management with minimization of variations in project costs. They considered uncertain resource availability and denoted the general problem as Rough Cut Capacity Planning (RCCP). It is close to the RLP model, with the objective to minimize the resource cost. Baydoun et al. [18] also used this notation. More generally, the management of data uncertainty is an important challenge in the decision-making process and in particular in project scheduling. In the next section, we discuss the methods developed to address it.

In our research, we focus on the RLP model with a goal to minimize extra resource usage for the case of a fixed deadline, see eq. (2.4). We consider that the decision-maker can adjust the amount of resources allocated to a job between a lower and upper bound and this amount becomes a decision variable for each job, resource, and time period (this intensity can be changed from one time period to another). This decision impacts the job duration: more resources are involved in job execution, the faster it is completed.

An industrial case of such management in energy scheduling for a pipe-manufacturing plant was presented by Artigues et al. [15]. This formulation included almost the same goal as RLP: to minimize the overload costs represented as electricity bills that included penalties for power overload. They applied an optimization approach with two stages:

the assignment with sequencing was realized with a CP model and MILP was used for energy consumption planning. Later, Nattaf et al. [88] developed this concept with energy resource allocation and studied several MILP formulations. Nattaf et al. [16] also provided new polyhedral results: a set of valid inequalities and polytope description. We can cite another example from the space industry presented by Capelle et al. [17]. The management of data transfer from satellite networks to ground optical stations requires an allocation of pairs satellite-station and transfer planning. It is possible within limited time windows, data buffer, and transfer speed. A control system for such applications should involve simultaneous resource allocation and planning with the objective to minimize the total costs.

From the theoretical side, the previous studies close to our research are the following. Baydoun et al. [18] developed a model for overload cost minimization problem with precedence constraints and job overlap. It is also similar to the problem presented by Bianco et al. in [19]. Baydoun et al. enriched the idea of the overlapping feature with several specific options. For example, overlapping was allowed if some part of the predecessor's job has already been implemented. Morin et al. [20] formulated the problem with resource constraints considered globally on time periods as Period-Aggregated RCPSP with basic RCPSP makespan objective function.

The main difference of our formulation is the possibility to manage properly precedence constraints and to allocate the resources with more freedom during the different time periods.

2.3 Solution methods

2.3.1 Branch and Bound/Branch and Cut

This approach was firstly introduced by Land and Doig [89]. The researchers studied and improved Branch and Bound algorithms, for example, see a recent survey of Morrison et al. [5] for details. This technique exploits the idea to decompose the problem and consider the smaller subproblems (nodes, or branches). Furthermore, it is possible to significantly reduce the number of iterations, and thus sometimes avoid the consideration of full exponential search space by pruning non-promising subproblems, i.e. removing them from the search space without loss of potential solutions. It is done with a help of bounds, the estimations of the best possible solution of a node. For the minimization (maximization) problem, a lower (upper) bound is calculated for a node and compared to the best-known solution. If a node is not promising to produce better solutions than already known, it is pruned.

This algorithm is recursive, the decomposition is repeated for the subproblems. As

a result, the set of subproblems has a form of a tree and it is defined as a search tree. At each iteration, the procedure to follow is defined by answering the following three questions:

- How to choose a subproblem in the search tree?
- How to construct a bound?
- How to decompose the subproblems?

The choice of the subproblem is not straightforward and there is no globally best procedure. With a form of a search tree, these procedures may be related to graph search methods, like depth-first search (DFS) and Breadth-first search (BrFS), plus the best bound searching. Clausen and Perregaard [90] firstly compared two DFS strategies. Morrison et al. [5] made some conclusions about both methods.

There are some typical ways to construct bounds and perform branching for a MILP. Bounds are usually made from an LP relaxation of the subproblem (ignoring integrality constraints). It is also used to make the branching, as the branching idea is to cut off the non-integral solutions in further relaxed subproblems. If the subproblem produces a solution with variable x and it is not integral and has a value β , the branching is made as follows. Two equal subproblems are created with additional cuts $x \leq \lfloor \beta \rfloor$ and $x \geq \lceil \beta \rceil$.

Furthermore, Branch and Cut (B&C) is a B&B approach expanded with a cutting plane method. When an LP relaxation produces a non-integral solution, there exists a linear inequality that cuts off this non-integral solution without impacting integral solutions. In other words, this linear inequality is a cut that separates this solution from the real feasible set (the convex hull). It may be added to the LP relaxation that will avoid the previous non-integral solution and produce another one that is closer to a convex hull of integral solutions. The cutting planes method was firstly presented by Gomory [91]. Several cutting plane types were developed. Chvatal-Gomory cuts were presented by Chvátal [92], advanced and strengthened by Caprara and Fischetti [93], Letchford and Lodi [94]. Gomory mixed-integer Cuts were improved by Wesselman et al. [95]. Balas et al. [96] introduced Lift-and-project inequalities and later outperformed the original result in Balas et al. [97]. A survey on valid inequalities may be found in the paper of Cornuéjols [98], Dey and Molinaro [99].

For the RCPSP, there are particular studies related to B&B and its components. The researchers try to strengthen the lower bounds and provide effective cutting planes. An overview of lower bounds calculation methods for the basic RCPSP formulation was prepared by Néron et al. [3] and then by Knust [4]. Demasse et al. [61] improved the LP relaxation with cutting planes derived from a constraint propagation preprocessing method. Arkhipov et al. [100] constructed a pseudo-polynomial algorithm providing a

new lower bound based on the estimation of related consumption of two resources. Tesch [101] introduced a general class of valid cutting planes for RCPSP compact MIP models and evaluated by computational experiments. Araujo et al. [102] also used a preprocessing algorithm to strengthen known cutting planes.

The researchers also studied particular types of RCPSP, extensions, and job-shop scheduling subcases that have the same nature of constraints and bounds. Bianco and Caramia [103] dedicated their review on the RCPSP version with generalized precedence relations and without resource constraints. Carlier et al. [104] described the relations between the basic RCPSP and the event scheduling problem. It is an extended formulation with jobs replaced by the events, having an impact on resource availability. Ainbinder et al. [105] presented new lower bounds for Resource Sharing and Scheduling Problem (RSSP). In this formulation, a job consists of a set of partially ordered activities with different resource demands.

In the next contributions, [106] and [107], a project scheduling problem with the project duration minimization objective function was studied with respect to variable job duration and possible overlapping. In [19], the classic total overload objective function with generalized precedence relations with time lags (GPR) was presented. Naber and Kolish considered the discrete-time model of RCPSP with flexible resource distribution in [67], then Naber studied the version of the model with continuous starts and ends of jobs in [108].

2.3.2 Benders decomposition

Benders decomposition, also named variable partitioning and outer linearization, is also a classic partitioning method applicable to mixed-integer programming problems [109]. In this method, the original MILP is separated into two simpler problems: a smaller "master" MILP and a "subproblem" LP. Master problem and dual subproblem are solved in the loop, the master problem solution forms the dual subproblem, and the dual subproblem solution is used to provide cuts for the master problem. The process continues until lower and upper bounds are equal, i.e. the iteration resulting in the equal values of the objective functions in both problems.

This approach demonstrated good results for various problems, such as the hub location problem [110] and project scheduling problems.

A resource flow-based formulation of the RCPSP was studied in [6]. In contrast to a more common RCPSP modeling approach (such as time-indexed or event-indexed), this formulation was proposed especially for being used with a Benders decomposition approach. In the proposed decomposition scheme, all decision variables and constraints except the resource flow part were used in the master problem, with a resource flow part forming the subproblem.

Boschetti et al. [7] studied a multi-mode resource-constrained project scheduling problem (MRCPSPP). This problem is based on the RCPSP with an additional feature: a set of possible modes to implement each job, with varying duration and resource requirements. The decomposition scheme for MRCPSPP was designed in the following manner: the master problem included the definition of the distribution modes for each activity and the subproblem represented the part of RCPSP with a fixed given duration value for each activity.

In the paper of Li [111], a Benders decomposition approach was used to solve an RCPSP-extension model with a resource-assignment objective function instead of an objective involving scheduling variables. These problems arise in a personnel assignment and staffing. The decomposition scheme included the assignment decisions in the master problem and scheduling decisions in the subproblem.

Benders decomposition was employed for the robust version of RCPSP by Bruni et al. [112]. The decomposition was designed as a separation of the decisions for scheduling and uncertainty management. The authors modeled the solution of RCPSP in terms of additional arcs in the precedence graph. The master problem dealt with the sequencing of possible additional arcs. The subproblem dealt with the uncertainty of parameters.

Emde et al [113] investigated a batching single machine scheduling problem with precedence, minimizing maximum lateness. The model was proposed for the single crane in an automated storage and retrieval system (AS/RS). The decision-making process included the assignment to batches and also the ordering of batches. The algorithm was constructed with one search tree for a cut generation. This method was denoted as a branch&Benders cut. Benders decomposition was also implemented for assignment and scheduling in medical operation disruption management. This research was presented in the paper of [114] with a heuristic algorithm to assign and plan operating room blocks. The algorithm was tested on empirical data.

Time-window vehicle routing problem was modeled by Fachini and Armentano [115] with a total cost objective function. The decomposition scheme followed the model structure: all assignments were produced in the Master, with a traveling salesman type Subproblem.

Guney et al. [116] modeled propagation of information in social networks, denoted as an influence maximization problem or a stochastic maximal covering location problem. On the primary level, it involved probabilities of interaction (transition), but the considered model required a scenario-based approach. The authors studied the form of Benders optimality cuts for this model and preprocessing.

Power production is also an active application area for optimization with Benders decomposition. Giuntoli et al. [8] used this type of decomposition for the grid distribution problem. The model was constructed to control the battery storage systems in

order to deal with the uncertainty of renewable resource sources. Short-term hydropower maintenance scheduling problem was resolved with Benders decomposition in the study of Rodriguez [9], with a parallelized accelerated algorithm, configured according to the empirical results.

A survey of the studies on the application of Benders decomposition was presented in [117]. It should be noted that the separation between the master problem and subproblem is based on the analysis of the structure of the initial problem. Although a basic version of Benders decomposition provides good results for particular types of combinatorial problems [110, 7], for others, the general scheme may not be efficient and requires additional improvements in order to accelerate the resolution and convergence. In [117], the authors describe these improvement techniques in the following way. The parts of the decomposition scheme are studied with a description of opportunities to advance. According to the structure of this approach, we point out the following components:

1. Instructions for Subproblem separation;
2. Master/Subproblem solution procedure;
3. Master model modifications;
4. Method for cuts generation.

The first point defines the structure of partitioning. The choice of variables for Master and Subproblem significantly affects the performance and the convergence. For example, Benders decomposition was effectively implemented to solve facility-location problems, with more than 90% of computation time spend to solve the Master (see [118, 110]). On the other hand, Master simplification leads to the growth of the number of iterations.

Secondly, there is a list of possible procedures to solve the Master and Subproblem. For example, it is possible to implement column generation for Master and use it in the Subproblem. The Master problem can be effectively implemented with a Constraint Programming formulation, see the paper of Maher [119] with a description of a Benders decomposition framework involved in SCIP solver.

Another important feature is the implementation of the algorithm without multiple runs of the Master model, i.e. using suboptimal incumbent solutions to make cuts within one branching tree. This method was denoted as Branch&Benders Cuts [120] and successfully implemented for facility location problem. As the resolution of the master problem even within one tree is a computational challenge, a significant effort was made to accelerate the branching process with an improved model and bounds. The master model might be modified or enriched with valid inequalities, or implemented with heuristic approaches, for example, see the paper of Emami et al. [121] with a robust Benders approach provided for a scheduling problem with order acceptance.

The cuts can be improved both in terms of quality and quantity. In some cases, it is even possible to avoid the generation of feasibility cuts with an improved Master model. Magnati and Wong proposed an accelerating scheme based on the construction of pareto-optimal cuts [122] and tested them on a class of facility location problems. In addition, the authors discussed some equivalent reformulations. The approach of Fischetti et al. [120] also involved an improved cut generation based on interior points. An iteration can also produce multiple cuts that contribute to faster convergence. In contrast, some approaches use heuristics clean-up strategies to delete useless cuts.

Tang, Jiang, and Saharidis proposed further improvement for the decomposition scheme for the facility location problem in [123]. Firstly, they considered the master problem without objective function and formulated valid inequalities in order to accelerate the initial progress of lower bounds provided by the master problem. Secondly, the disaggregation approach was applied for the subproblems with the construction of multiple cuts at each iteration of the decomposition. This technique helped to accelerate problem resolution. An approach based on multi-cut generation in case of subproblem infeasibility was developed in [124]. Its effectiveness was demonstrated on the scheduling problem of the multipurpose multiproduct batch plant. The general idea was to generate additional optimality cuts in case of unboundedness of the dual subproblem.

To sum up, Benders decomposition is an efficient solution method that to the best of our knowledge has not been applied yet to RLP.

2.3.3 Constraint programming

Constraint programming (CP) is another paradigm to design and solve scheduling problems, actively developed in recent years. A CP model includes decision variables, domains for variables, and constraints among the decision variables (defined as a Constraint Satisfaction Problem, CSP). It uses such search schemes as constraint propagation and some problem-specific add-ons. Constraint propagation is an iterative process that makes the correspondence between the domains of different variables and linking constraints. The goal is to eliminate the values of variables that do not allow to satisfy all the constraints. The procedure is stopped if it is impossible to make the domains tighter. Constraint propagation is combined with search procedures, for example, backtracking, Branch&Bound, local search methods. The modules that are responsible for each part of the algorithm may vary, and CP does not require a linear form of constraints inside the model.

This approach has already demonstrated good results for scheduling problems from theoretical and practical sides. From the theoretical side, a set of effective constraint propagators was created. See the thesis of Vilím [125] and the paper of Laborie [13] for the overview of existing approaches, based on the resource usage and precedences. Several important terms were introduced for the Cumulative Scheduling Problem (CuSP) with

a single resource in the pioneer studies: compulsory part by Lahrichi [126] and resource profiles by Fox [127]. In a more recent study of Letort et al. [128] a sweep algorithm tightened the earliest (latest) start (end), and Gay et al. [129] updated complexity results for the time-labeled cumulative constraints. Carlier et al. [130] studied the energetic reasoning for the CuSP and a set of jobs with a given duration and implementation time window. The energetic reasoning method tightened the time windows for jobs with respect to the schedule feasibility.

For several datasets for RCPSP, CP models outperform the Mathematical Programming techniques. Recent results of Hauder et al. [131] also showed the potential of CP formulations for project scheduling problems with makespan and resource balancing objectives.

2.3.4 Heuristics

Heuristic methods are frequently used for NP-hard problems. A comprehensive review of heuristic approaches may be found in the papers of Agarwal et al. [10] and Pellerin et al. [11]. The researchers develop heuristics to specific and large RCPSP-cases: a tabu search for extended multi-mode RCPSP by Servranckx and Vanhoucke [132], evolving heuristics for the RCPSP with dynamic resource disruptions, approximate dynamic programming for RCPSP with overlapping by Chu et al. [133], priority rule method for RCPSP with fuzzy processing times by Bhaskar et al. [134]. Guo et al. [135] provided a decision approach to identify the best RCPSP priority rules.

An overview of construction project scheduling quantitative methods was presented by Briskorn and Dienstknecht [12], with a significant part devoted to RCPSP and MRCPSP heuristics.

2.4 Uncertainty modeling in project planning

The uncertainty of the data is one of the important issues for project planning. In the literature, proactive, reactive, and proactive-reactive approaches are distinguished to address this issue. Proactive approaches try to anticipate the data variations, reactive approaches seek to react quickly to the effective changes. Proactive-reactive approaches build a general plan with some anticipation and provide also a reactive method to adapt this plan to the observed data changes.

Hazir and Ulusoy presented a recent literature review [24] on common uncertainty sources and ways to model them in project scheduling. The most frequently used proactive approaches are to represent the uncertain data as a random value or to construct a robust (acceptable for any scenario) solution, or to build a baseline solution, adaptable to any

possible scenario. Another possibility is to study the sensitivity of the model and describe the impact of variations in input parameters. The less frequently used approach is the metric approach discussed in detail here below.

Existing stochastic methods for RCPSP were analyzed by Bruni et al. [21]. Li et al. [49] presented two scheduling policies for RLP with uncertain job times, minimizing the total variation of the occupied resource. The first one solved a deterministic equivalent of the stochastic problem, and the second applied a tabu search directly to the stochastic formulation. The same authors later studied a robust version of RLP and proposed a genetic algorithm [50]. Recent results present a metaheuristic procedure for a formulation of RLP with a variation affecting both duration and minimum time lags between the jobs. Ke and Zhao [136] also studied uncertain job times and proposed a heuristic approach. A case of RCPSP with a set of scenarios with various duration was presented by Artigues et al. [22], with two algorithms, an exact and a heuristic scenario-relaxation procedure. Hazir et al. [23] formulated three robust optimization models for a discrete-time/cost trade-off problem with a similar structure: a set of jobs with a precedence graph and a set of modes characterized by different processing time and cost. The goal is to schedule the jobs and assign the modes to minimize the total cost with respect to a given project deadline.

2.4.1 Sensitivity analysis

Sensitivity analysis is performed in order to study the impact of variations of parameters on the quality of the solution. In the case of MILP, sensitivity analysis can be performed for a model in the following form:

$$\begin{aligned}
 \min \quad & (c + \Delta c)^T x \\
 \text{s.t.} \quad & (A + \Delta A)x \geq (b + \Delta b) \\
 & x \geq 0 \\
 & x_i \in \mathbb{Z}, \forall i \in I
 \end{aligned} \tag{2.5}$$

$$\text{where: } c, \Delta c \in \mathbb{R}^n, A, \Delta A \in \mathbb{R}^{m \times n}, \text{ and } b, \Delta b \in \mathbb{R}^m$$

Dawande and Hooker [27] introduced a general method for sensitivity analysis for MILP with any kind of perturbation allowed $(\Delta c, \Delta A, \Delta b)$. They proposed two approaches: the first one to change the solution while keeping the same objective function value (or shifted in some given range), it was defined as dual analysis; and the second one to find an upper bound for the objective function value for a given perturbation of parameters, it was defined as a primal analysis.

This method was later developed by Jia and Ierapetritou [28, 29]. They studied the batch plant scheduling problem. The dual analysis method of Dawande and Hooker [27] was used to calculate allowed changes. They also used a metric to describe the robustness

of considered schedules. Furthermore, this approach was upgraded and used for the same class of batch plant scheduling problem [137]. Experiments were carried for two use cases: single product production line and two-product line.

We can also refer to the paper of Emelichev et al. [30], who studied a vector combinatorial linear problem and the case of perturbations of coefficients of linear functions and proposed a stability radii as a function. Later this approach was also used by Gurevsky et al. [31] for an assembly line balancing problem, providing stability and feasibility measures in case of possible processing times variations.

Sensitivity is investigated empirically in many areas when operation research methods are applied. For example, it was used by Ren and Gao [138] for the planning of distributed energy systems. The sensitivity of the model was determined for major parameters, such as the demand, prices, and carbon tax rate. Kouvelis et al. [139] constructed a predictive tool, based on a facility network design MILP. This tool allowed us to obtain some data estimations without the execution of a large MILP that describes the process precisely. The hydrothermal energy generation network of Ghana was modeled and studied by sensitivity analysis in the paper of Etwire and Twum [140]. The energy management topic was also reviewed by Moser et al. [141]. They presented a novel modular modeling approach for urban energy management systems with complex system configurations. The model was prepared to resolve the problem of growing sources, storage centers, and consumers. The total cost was improved with an approach that takes into account forecast errors of user's demand or weather errors.

Sensitivity analysis was applied to a large-scale MILP describing the process of pricing and ordering for new customers [142]. It was conducted for a firm with a continuous-flow production process with complex scheduling. The main result consisted of the pricing and planning policies and the identification of the key parameters with a high impact on production volume.

We can note that this topic is closely related to planning in case of unexpected disruptions, as it is the most frequent "unexpected" parameter change in the model. On the one hand, model sensitivity allows us to evaluate the range of acceptable parameter changes. On the other hand, it is used to construct stable models to simplify the calculation of new solutions. Railway scheduling timetabling is an important application area of operations research. MILP models are used to construct the schedules and to revise the schedules in case of urgent disruptions. Khoshniyat and Krasemann [143] made a research about rescheduling policies for a railway network in Southern Sweden. The primary goal was to make a better model to allocate the time slots for urgent maintenance and reschedule the trains properly, but it also involved the analysis of model sensitivity to the values of the big-M constant. Another model dealing with interruptions in the electricity distribution network was presented by [144]. Although the term "sensitivity" was used in a different

meaning as a particular parameter of the model to describe client reactions, this model is also relevant to the topic of disruption resolving. Chaojue and Ming [145] studied the sensitivity in the case of earth-moving and temporary haul road layout design. The real project case was modeled as a MILP, with two goals: to categorize input parameters in several classes of impact and to analytically define the stability region for each parameter.

In some cases, sensitivity analysis is combined with another approach to tackle uncertainty, the scenario-based method. Instead of the infinite number of input data combinations, the uncertainty is discretized into a finite set of scenarios. Silvente et al. [146] illustrated this approach for non-linear programming optimization models. The main idea lied in a reduction of the scenario set with a sensitivity analysis to keep the same quality of results, i.e. to have the same cover of possible changes with fewer scenarios. Two cases were used to demonstrate this method: pricing and extraction optimization for OPEC and household optimization.

2.4.2 Metric approach in scheduling theory

A metric approach for scheduling problems was presented by Lazarev [32]. It is an approximation scheme that builds sub-optimal schedules with guaranteed estimated accuracy in a polynomial time for a problem that is *NP*-hard in general but has some polynomially solvable sub-classes of instances. The idea is based on the construction of a metric function on a space of instances, that shows an upper bound of an objective function value difference. With this metric function, an approximation scheme consists of two stages: firstly, by solving an LP with the objective to minimize the metric distance to find the closest solvable instance; and secondly to apply a corresponding polynomial algorithm to this instance.

More specifically, in this approach, any problem instance is represented with a point in $f(n)$ -dimensional space Ω , where n is the number of jobs in the instance. The metric function is defined in the following way. If the same schedule π is used as the solution for two different instances A and B , it is possible to make the estimation of the difference between objective function values $V_A(\pi)$ and $V_B(\pi)$ for these instances. This estimation is formed as metric $\rho(A, B)$ defined on $\Omega \times \Omega$:

$$|V_A(\pi) - V_B(\pi)| \leq \rho(A, B). \quad (2.6)$$

The second step is used to estimate the deviation from the optimal value for the instance that is not solved. Suppose there are two schedules π^A and π^B which are the optimal solutions for instances A and B , respectively. $V_A(\pi^B)$ is the value of the objective function for instance A if schedule π^B is applied (optimal for instance B) and $V_A(\pi^A)$ is the optimal value of the objective function for instance A since schedule π^A is an optimal schedule for

instance A .

$$V_A(\pi^B) - V_A(\pi^A) \leq \Delta(\rho(A, B)). \quad (2.7)$$

The following Lemma proves that this deviation depends on $\rho(A, B)$.

Lemma 1. *If two schedules π^A and π^B are the optimal solutions for instances A and B , respectively, and there is a defined metric upper bound $\rho(A, B)$ (see eq. 2.6), then*

$$V_A(\pi^B) - V_A(\pi^A) \leq 2\rho(A, B). \quad (2.8)$$

Proof. We will transform the expression in the following way:

$$\begin{aligned} V_A(\pi^B) - V_A(\pi^A) &= \\ &= (V_A(\pi^B) - V_B(\pi^B)) + (V_B(\pi^A) - V_A(\pi^A)) + (V_B(\pi^B) - V_B(\pi^A)). \end{aligned}$$

Here the first and the second group in brackets is less or equal to $\rho(A, B)$, and

$$V_B(\pi^B) - V_B(\pi^A) \leq 0$$

since $V_B(\pi^B)$ is the optimal schedule for instance B .

□

Therefore, we can define $\Delta(A, B) = 2\rho(A, B)$ as the absolute upper bound for accuracy for the case when π^B is used as the solution for instance A instead of the real optimal solution. The original idea is to use schedule π^B that is optimal for some polynomially solvable instance B as the solution for the original NP-hard problem instance A , such that the difference between the objective function values is minimal, i.e., for the minimal value of $\rho(A, B)$. In other words, the resolution of the initial NP-hard problem for instance A is transformed into the resolution of a polynomially solvable instance B and its research by solving an LP minimizing $\Delta(\rho(A, B))$ to search the instance B with the minimal value for metric function $\rho(A, B)$. All known solvable classes are defined by the set of linear inequalities with parameters of instance, therefore the new problem is linear programming. This algorithm is applicable to particular scheduling problems for which proper metric functions can be defined.

To sum up, the method proposed by Lazarev [32] for scheduling problems works if the following conditions are satisfied:

- it is possible to build a proper metric function $\rho(A, B)$, i.e. to construct the estimations of the difference between the objective function values when an optimal solution for one instance is applied to a different problem instance;

- it is possible to provide effective estimations when we use the optimal solution for instance B as the solution for instance A :

$$V_A(\pi^B) - V_A(\pi^A) \leq \Delta(A, B);$$

- there are some "simple" subclasses of the original problem which are solvable in reasonable time and could provide the solutions with good accuracy to an arbitrary instance of the same size.

This metric approach has been shown to be effective in dealing with some NP -hard scheduling problems (several examples presented in [32]), including the total tardiness minimization problem [33] and the single-machine scheduling problem [34].

Further, in [147], the metric approach was combined with a study of the space of problem instances. It was shown that for a classic scheduling problem there are some sets of instances with the same optimal schedule and shifted/scaled values of parameters. So, it is possible to consider a compact subset of instances with a normalization procedure, guaranteeing that any instance can be projected on this subset while keeping its properties such as the structure of the optimal solution and the objective, multiplied by some scale coefficient.

In this thesis, we study the applicability of the metric approach for our generalized version of RLP. We explore the properties of our model and the input data instances. Although in the case of RLP as for RCPSP, it is difficult to specify the solvable subclasses, the first two necessary conditions are still correct and can be used for the theoretical development. We note that for RCPSP, small differences in some parameters may lead to significant step changes in the objective function value. The key issue is the feasibility of the solution: there are strict resource limits and any fluctuation of either job parameters or available resource level may significantly modify the set of feasible solutions. In a generalized version of RLP as well as in basic RLP formulation with the possibility of the use of extra resources, the ratio of available and required resources does not affect the feasibility.

Therefore, for our research we define the following goals:

1. to show that it is possible to apply the original metric approach, i.e. to provide estimations $\Delta(A, B)$ for arbitrary instances A and B to our generalized formulation of RLP;
2. to improve the basic concept $\Delta(A, B) \leq 2\rho(A, B)$ and provide better estimations;
3. to demonstrate how the fluctuations in problem parameters impact the feasibility of known solutions (similar to the stability radii used by [31]).

These contributions are presented in Chapter 5, but before we introduce the mathematical formulation of the problem that we study in this thesis and its MILP model in the next chapter.

Chapter 3

A generalized RLP: MILP formulation

We consider a new generalization of the RLP minimizing extra resource usage cost formulation proposed by Baydoun et al. [18] and Bianco et al. [19]. Our formulation solves the problem with job overlapping in case of precedence constraints and manages them in a normal way. Another difference and novelty of our model lie in the way of resource assignment. On the base of a similar set of constraints and objective function, we propose to allocate resources more flexibly with an additional separation of decision variables. Resource allocation per period has lower and upper limits for each job. For each resource type, there is also a total required amount for each job. The models studied in these papers use one decision variable for each job and each time period. They denoted it as a fraction of realized workload. This variable represents the progress made by the job in each time period. The amount of resource allocated to the period for the job depends on this decision variable, multiplied by the total required amount of this resource type for this job. So, all resources are allocated proportionally in the same time period, i.e. all resource types depend on one decision variable.

However, when multiple resources are needed to execute a job, in the general case, a job can spend them at a different speed. In this case, the allocation of different resource types with one proportional coefficient can excessively restrict solution space and require extra resources to satisfy this constraint. We propose a new generalized formulation where it is not required to spend different resources at the same speed in each time period. This model is a generalization of the resource leveling problem. The contribution of this chapter was also presented in the following works:

- Tarasov, I.; Haït, A.; Battaïa, O. *Period-Aggregated Resource-Constrained Scheduling Problem with variable job duration*. 19ème Congrès annuel de la Société Française de Recherche Opérationnelle et d'Aide à la Décision (ROADEF 2018), Feb 2018, Lorient, France.;
- Tarasov, I.; Haït, A.; Battaïa, O. *A Generalized MILP Formulation for the Period-*

3.1 Flexible resource allocation in RLP

3.1.1 Illustrative examples

We will use two examples to explain the concept of flexible resource allocation. Fig. 3.1 presents an example of solution obtained with the use of job fraction decision variable. There is a schedule above in a form of a Gantt chart, and a resource usage plot with the resource allocation profile and available resource level. Job fraction is defined in resource profile with % sign and it is used to define the job progress with both resources.

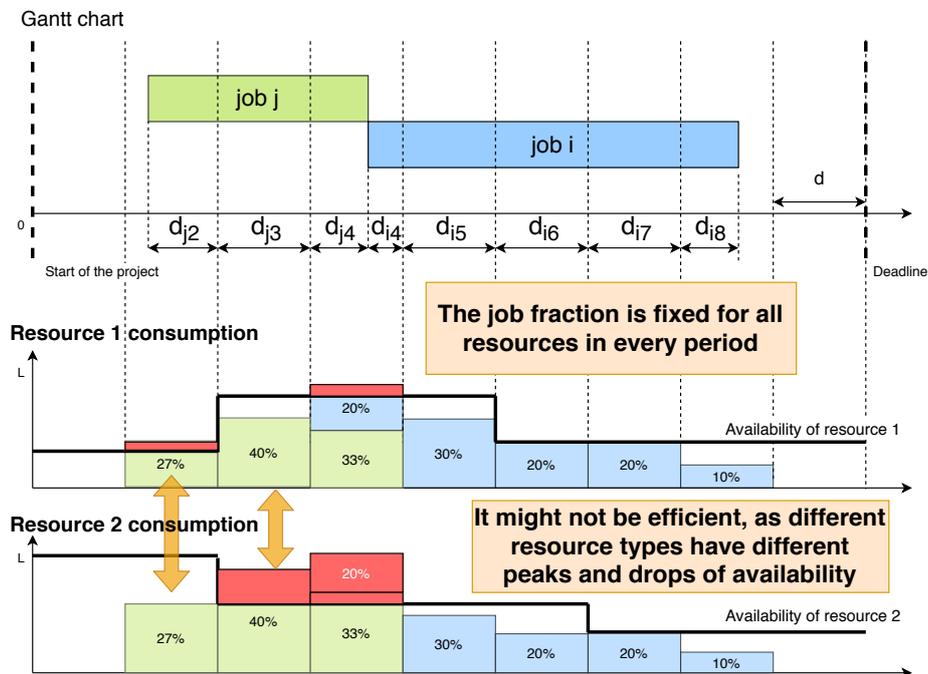


Figure 3.1: Resource allocation with job fraction decision variables

The periods with higher available resource amounts vary for both resources. So, if in some period we utilize all available amounts for one resource, we get another resource type overloaded over the limit. It is feasible, but it is not an optimal solution if we unfix the job fraction and define the progress independently. Fig. 3.2 shows how the resource usage can be improved. In this case, the progress made by a given resource type for a given job in a given time period is a separate decision variable. Thus, it becomes possible to use more available resources and avoid overloads due to this flexible allocation.

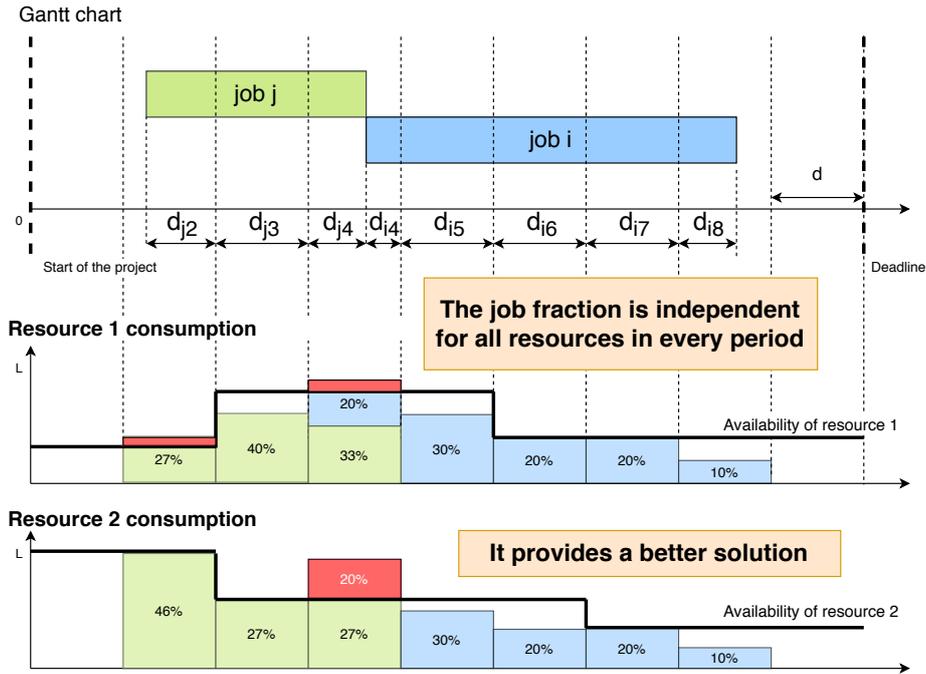


Figure 3.2: Independent resource allocation: job progress is different.

Let's show another example. Suppose there is a problem with one job j and two resources. Job j requires for its completion one unit of resource r_1 and two units of resource r_2 . We start with the models considered in previous studies. If the job is implemented in 2 time periods, a solution defines a unique job fraction per period. It is the same for all resources. In any solution, for this job, there will be a constant ratio of $1/2$ of involved r_1 and r_2 units in each period. We can set the fraction in the first period as 0.3 and in the last period as 0.7 . With this solution, job j uses 0.3 units of r_1 and 0.6 units of r_2 in the first period (total demand multiplied by a fraction). In the second period, job j gets 0.7 units of r_1 and 1.4 units of r_2 .

Our model constructs the solutions avoiding any fixed relations between the resources involved to complete the job in one time period. For example, we can use 0.5 units of resource r_1 and 0.5 units of resource r_2 in the first period, 0.5 units of resource r_1 and 1.5 units of resource r_2 in the second period. This solution has independent ratios of used resources: $1/1$ and $1/3$ in each period. We use more resource r_1 in the first period and more resource r_2 in the second period. We obtain a more flexible solution that better fits the resource availability constraints.

3.1.2 Problem statement

Our formulation avoids job overlapping for jobs linked by precedence constraints, so we do not consider a subcase with allowed overlapping described by Baydoun et al. [18]. Aggregated resource constraints are used as well as the following assumptions: variable

job duration, flexible resource distribution, and varying job intensities from period to period.

Time indexing

In our model, the project deadline is fixed. Our problem is defined for a planning horizon divided into $T = \{1, \dots, m\}$ periods of given length d . The input parameter d was defined in the model instead of unit length interval for the further perspective of studying the models with different values of d (for example, make a bigger period including several initial periods to make relaxation of the model). We consider the case where all jobs have their minimal duration superior to the period length of d , as it has an impact on the structure of precedence constraints. Briefly, a potential chain of more than two jobs (caused by duration less than d) inside one period requires special constraints and variables to handle the precedence relations correctly. To get correct precedence constraints, we have to expand a model and choose a special design of decision variables: add continuous start and end variables for each job. We mention the minimal duration in our model because the job duration might be varied in some range, determined by the resource requirements and assignment limits.

Variable job duration: work volume

In the model, we deal with the notion of work volume to define the jobs and availability of resources. In other words, we do not specify the job as an operation with a fixed duration and resource requirement per period. We set the total required volume of work and the range for job processing intensity. The job intensity inside one period also depends on the period length, it might be multiplied if we change the period length in the model. In this case, the job duration and intensities in a period are decision variables, also defining required resources in each moment.

It could be explained in terms of energy and power, as the origin of such models comes from the practical cases in the energy management industry. For example, we have to manage the distribution of electricity or heat to execute certain jobs. Jobs are defined by the required amount of energy, and in all time periods, there is a limited amount of energy that can be spent without penalties. Besides, there are lower and upper power limits applied to each job. A job can be implemented in less time but with higher required power, or we can use lower power and process it longer. It is also important to mention that the allocated power can be different from one period to another.

We expand the formulation for the case when the resource type is not necessarily the energy. The concept of the workload allows us to deal with discrete resources involved in some continuously measured work (for example, as the case of a person-hour value).

However, we keep the same approach, considering the job work volume on the resource as the aggregation of the required amount of this resource and the time when it is occupied by the job. In the rest of the thesis, we will use the work volume instead of energy.

3.2 Mathematical Model

3.2.1 Problem data

Resources and Jobs

In our formulation, the project duration (and planning horizon T) is limited by a hard deadline and the optimization objective is to respect this deadline with the minimal cost of extra resources used. All resources are renewable. The availability of resource $r \in R$ is limited by amount L_{rt} for each period $t \in T$. An extra amount can be acquired for each resource $r \in R$ but it will have an additional cost e_r . In each period, the completed work volume of each job may be different under the constraint that in total for all periods the work volume is equal to the required value.

This approach corresponds to the practical case where for one job in different time periods different numbers of workers may be assigned. Accordingly, the duration of the job is also a decision variable. It depends on the amount of completed work volume by each resource for this job in each period. The set of jobs J is defined with three parameters for each job $j \in J$: required work volume W_{jr} on each resource $r \in R$ and minimum and maximum limits of assigned resource $r \in R$ per period, $p_{min,jr}$ and $p_{max,jr}$, which defines the lower and upper power limits. So, the duration of job j is limited by the relation of required work volume W_{jr} and the lower and upper limits $p_{min,jr}$ and $p_{max,jr}$ (for each resource type). Briefly, the minimum duration is achieved if the job is executed with an upper limit of assigned resource during all processing time, as well as the maximal duration is achieved if the job is processed with minimal allowed utilization of resources in each period. We introduce a more detailed description of these relations when we present the model.

Precedence relations

There are also precedence instructions. We use an "activity-on-node" notation and define the set of pairs predecessor-successor as P . A job precedence graph is given and it has to be acyclic. Classic approach with dummy zero and $|J| + 1$ activities is used to calculate the minimum required duration of the project and possible job execution time windows according to the precedence graph, we use it only in preprocessing.

There is also a second point that makes this model novel. In related publications of Bianco et al. [19] and Baydoun et al. [18] precedence relations were considered in a

different way, on a different level of accuracy. A basic 0-1 scheduling variable approach for discrete-time models (firstly introduced by Pritsker [1]) is applicable for these models: it is possible to point out the first and the last period index of a processed job to form a schedule and make a correct allocation of resources. Without any supplementary continuous variables, precedence relations are defined only between starting and ending periods. Bianco et al. [19] used this model structure. Generalized precedence relations with time lags (GPR) used by Bianco et al. [19] can be also used in the model.

This approach lacks accuracy, especially if the periods are relatively long. Additional continuous scheduling variables that directly define the starts and ends of job processing lead to more flexible scheduling and thus to the solutions with the better objective function value. For example, they allow a "safe" intersection of preceding and succeeding jobs in one "end-start" period, fixing the condition that in this period the processing of the successor will not start before the end of the predecessor. Baydoun et al. [18] implement this approach with a larger set of decision variables.

In our case, we involved fewer variables than Baydoun et al. [18] while maintaining these capabilities. Precedence relations are defined between binary indexes of starting and ending periods of jobs, and also inside periods with the use of continuous variables to set the duration of the job in a given period. Thus, we consider the "precise" precedence relations. We define them between periods and inside periods. In other words, if $(j_1, j_2) \in P$, jobs j_1 and j_2 might be implemented in one period (the last period when job j_1 is executed). Hans qualified the case with precise precedence relations as "a formidable computation challenge" [81]. Indeed, this assumption makes the model larger as it requires additional variables.

The list of all parameters in our mathematical model is presented in Table 3.1.

Table 3.1: Model nomenclature: Parameters notation.

Parameters	
T	planning horizon, $T = \{1, \dots, m\}$
d	period length
R	resources set
L_{rt}	availability of resource $r \in R$ in period $t \in T$
e_r	extra resource cost
J	jobs set
W_{jr}	job $j \in J$ work volume with resource $r \in R$
$p_{min,jr}$	job $j \in J$ minimal requirement per period in resource $r \in R$
$p_{max,jr}$	job $j \in J$ maximal requirement per period in resource $r \in R$
P	set of arcs in the given precedence graph

3.2.2 Generalized Model description

For any kind of RLP, it is possible to separate the set of variables into two subsets, related to job scheduling and resource allocation. The subset of scheduling variables for discrete-time period models is well studied in project scheduling (starting from pioneer introduction of a 1-0 variable approach by Pritsker et al. [1]). Usually, it is a set of binaries that configure starting and ending periods. If we consider conventional precedence relations and allow the successor to start at the same period where the predecessor ends, we also need to use continuous duration variables responsible for the duration inside the period. We define binary variables responsible for the job starts and ends as S_{jt} and E_{jt} , respectively. They may be used in several ways.

- As a step pointer, i.e., if job j starts at period t , then $\forall t_1 < t S_{jt_1} = 0$ and $\forall t_2 \geq t S_{jt_2} = 1$. In this case the same logic is implemented for E_{jt} , if a job ends at period t , then for $\forall t_1 \leq t E_{jt} = 0$ and $\forall t_2 > t E_{jt} = 1$. To the best of our knowledge, in similar resource leveling models step pointer to job start and end periods is more common (for example, it was used by Baydoun et al. [18], Bianco et al. [19]).
- As an on-off (also referred as pulse) pointer, i.e., if job j starts at period t , then $\forall t_1 \neq t S_{jt_1} = 0$ and $S_{jt} = 1$ (the same logic for E_{jt}). This approach was considered by Tamas Kis and Marton Drotos, [148]. There is an alternative way, with an on-off function U_{jt} , where $U_{jt} = 1$ if job j is implemented at period t , and $U_{jt} = 0$ otherwise.

These approaches were also presented and compared in the literature related to project scheduling, for example, see [108, 22].

The second subset defines resource-allocation decision variables. We propose to generalize it. We introduce separate fraction decision variables c_{jrt} for each job j , resource type r and time period t instead of aggregated job fraction decision variables. This approach makes an independent allocation of each resource type for every job. It leads to more flexible utilization of resources. However, it increases the size of the model. We present all decision variables in Table 3.2.

In the next subsections, we describe three different approaches to represent scheduling constraints with binary variables. A general description of each type is presented in Figure 3.3.

Table 3.2: Model nomenclature: Decision variables notations.

Scheduling decision variables	
$U_{jt} \in \{0, 1\}$	on/off, $U_{jt} = 1$ if job $j \in J$ is executed in period $t \in T$, 0 otherwise
$S_{jt} \in \{0, 1\}$	pulse start, $S_{jt} = 1$ if job $j \in J$ starts in period $t \in T$, 0 otherwise
$E_{jt} \in \{0, 1\}$	pulse end, $E_{jt} = 1$ if job $j \in J$ ends in period $t \in T$, 0 otherwise
$S_{jt}^* \in \{0, 1\}$	step start, $S_{jt}^* = 1$ if job $j \in J$ starts in $\forall t_1 \in T, t_1 \leq t$, 0 otherwise
$E_{jt}^* \in \{0, 1\}$	step end, $E_{jt}^* = 1$ if job $j \in J$ ends in $\forall t_1 \in T, t_1 < t$, 0 otherwise
$d_{jt} \in [0, d]$	duration of job $j \in J$ in period $t \in T$
Resource allocation decision variables	
$c_{jrt} \in [0, dp_{max,jr}]$	work volume of job $j \in J$ with resource $r \in R$ in period $t \in T$
$o_{rt} \in [0, \infty)$	extra cost of resource $r \in R$ in period $t \in T$

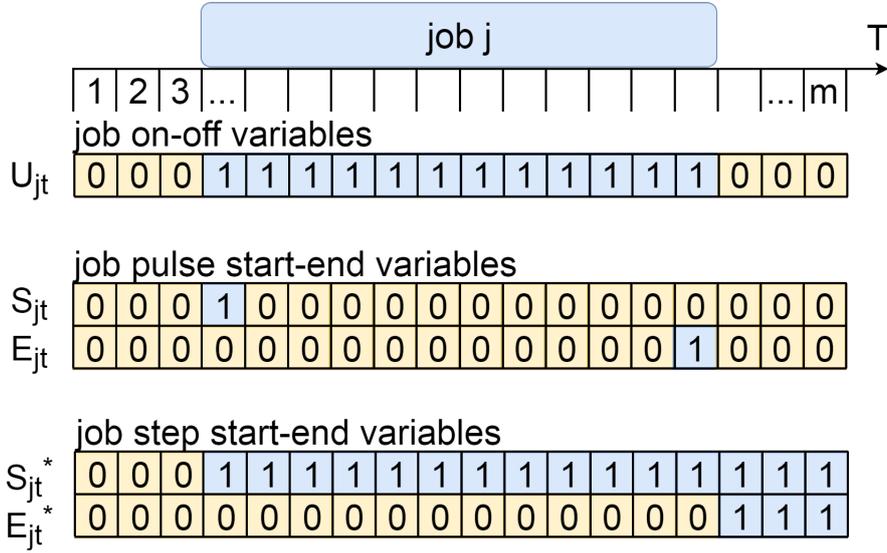


Figure 3.3: Three different ways to utilize binary scheduling variables.

Scheduling Constraints: Job On–Off Formulation

This set of constraints is defined as job on–off implementation formulation, as we use one binary variable implying that the job is implemented inside some period or not. This definition of binaries in project scheduling problems is denoted as pulse or on–off in literature. So we use a variable $U_{jt} \in \{0, 1\}$, $U_{jt} = 1$ if job j is implemented in period t , $U_{jt} = 0$ otherwise. Constraints (3.1) imply that no preemptions are allowed.

$$(t_2 - t_1 - 1) \leq \sum_{l=t_1+1}^{t_2-1} U_{jl} + m(U_{j,t_1} + U_{j,t_2} - 2), \forall t_1 \in T, t_2 \in T, t_1 < t_2, \forall j \in J. (3.1)$$

If the job is implemented in some period $t_1 \in T$ and in some period $t_2 \in T$, $t_1 < t_2$, then it must be also implemented in any period $t_3 \in T$, $t_1 < t_3 < t_2$. In other words, if for

some job j there exists $t_1 \in T$, $t_2 \in T$, $t_1 < t_2$, such that $U_{jt_1} = 1$ and $U_{jt_2} = 1$, then

$$\sum_{t_3=t_1+1}^{t_2-1} U_{jt_3} = t_2 - t_1 - 1.$$

Further, two constraints make the correspondence between binary scheduling variable U_{jt} and job duration d_{jt} . Firstly, if job $j \in J$ is performed in three or more periods, then for all periods between start period and end period job duration is the same as period length, i.e., preemptions inside periods are not allowed. This constraints involve continuous duration and binaries variables $d_{jt} = d$ if $U_{j,t-1} = 1$, $U_{j,t} = 1$, $U_{j,t+1} = 1$:

$$(2 - U_{j,t+1} - U_{j,t-1})d + d_{jt} \geq dU_{jt}, \forall t \in T, \forall j \in J. \quad (3.2)$$

Secondly, if job $j \in J$ is not implemented in period $t \in T$, it must have zero duration inside this period. So $d_{jt} = 0$ is required if $U_{j,t} = 0$:

$$d_{jt} \leq dU_{jt}, \forall t \in T, \forall j \in J. \quad (3.3)$$

Precedence constraints are represented on two levels, on periods and inside each period. Firstly, we state that if there is a precedence $\{j_p, j_s\} \in P$ relation between two jobs, then it is impossible to implement the successor before the last period when predecessor is implemented (it is possible to start the successor at the last period of predecessor implementation). It is represented with the following condition: $\forall t_1 \in T$, if $U_{j_p, t_1} = 1$ then $\forall t_2 \in T$, $t_2 < t_1$, $U_{j_s, t_2} = 0$, and the corresponding constraints are:

$$U_{j_s, t_2} + U_{j_p, t_1} \leq 1, \forall \{j_p, j_s\} \in P, \forall t_1 \in T, \forall t_2 \in T, t_2 < t_1; \quad (3.4)$$

Secondly, we state that if these jobs are implemented in one period, total duration of both jobs is less than period duration. Otherwise, it means that in this period j_p and j_s cross each other.

$$d_{j_1 t} + d_{j_2 t} \leq d, \forall t \in T, \forall (j_1, j_2) \in P. \quad (3.5)$$

We note that we use Constraints (3.5) to take into account a case when for a pair of predecessor-successor, both jobs are implemented in the same period, which is the last period for the predecessor and the first period for the successor.

Scheduling Constraints: Job Pulse Start–End Formulation

We define this set of constraints as pulse start–end formulation since binaries S_{jt} and E_{jt} take the value 1 only in the periods of the job start and end, respectively. We have variable job duration, so it is not sufficient to use only the start pulse decision variable S_{jt} , we also need the end variable E_{jt} . Constraints (3.6) and (3.7) for S_{jt} and E_{jt} insure that each job starts and ends exactly once:

$$\sum_{t \in T} S_{jt} = 1, \forall j \in J; \quad (3.6)$$

$$\sum_{t \in T} E_{jt} = 1, \forall j \in J. \quad (3.7)$$

We force the job duration variable d_{jt} to get zero value in periods t when job j is not implemented. The index of starting period for job j equals $\sum_{t \in T} tS_{jt}$, and ending period index is $\sum_{t \in T} tE_{jt}$. So we state that duration $d_{jt} = 0$ outside interval $[\sum_{t \in T} tS_{jt}, \sum_{t \in T} tE_{jt}]$ with the following constraints. If job j is started in some period after period t or it is finished in some period before period t , then $d_{jt} = 0$:

$$d_{jt} \leq d \left(1 - \sum_{k=t+1}^m S_{jk} - \sum_{l=1}^{t-1} E_{jl} \right), \forall j \in J, \forall t \in T; \quad (3.8)$$

Next constraints have the same sense as Constraints (3.2). In any period between start and end of job j , it must be implemented without preemption inside the period, so $d_{jt} = d$ inside interval $(\sum_{t \in T} tS_{jt}, \sum_{t \in T} tE_{jt})$. If job j was started before period t and it was finished after period t , $d_{jt} = d$:

$$d_{jt} \geq d \left(\sum_{k=1}^{t-1} S_{jk} + \sum_{l=t+1}^m E_{jl} - 1 \right), \forall j \in J, \forall t \in T. \quad (3.9)$$

The precedence constraints require two constraints for binaries and continuous job duration for each pair $\{j_p, j_s\} \in P$, representing precedence constraints for each time period. In this formulation, we can just compare start period index of successor j_s and the end period index of predecessor j_p :

$$\sum_{t_1 \in T} t_1 E_{j_p t_1} \leq \sum_{t_2 \in T} t_2 S_{j_s t_2}, \forall (j_p, j_s) \in P; \quad (3.10)$$

$$d_{j_p t} + d_{j_s t} \leq d, \forall t \in T, \forall (j_p, j_s) \in P. \quad (3.11)$$

Scheduling Constraints: Job Step Start–End Formulation

In this case we use step binaries S_{jt}^* and E_{jt}^* with the following rule:

- if job j starts at period t , then $\forall t_1 < t S_{j t_1}^* = 0$ and $\forall t_2 \geq t S_{j t_2}^* = 1$;
- if job j ends at period t , then $\forall t_1 \leq t E_{j t_1}^* = 1$ and $\forall t_2 > t E_{j t_2}^* = 0$.

This approach is defined as step formulation because for each job the plot with decision variables looks like a non-decreasing step function. A job may be ended only if it was started in the same period or before, and the values of start and end step variables S_{jt}^* and E_{jt}^* must be non-decreasing for each job $j \in J$:

$$S_{jt}^* \geq E_{jt}^*, \forall j \in J, \forall t \in T; \quad (3.12)$$

$$S_{jt}^* \leq S_{j,t+1}^*, \forall j \in J, \forall t \in T; \quad (3.13)$$

$$E_{jt}^* \leq E_{j,t+1}^*, \forall j \in J, \forall t \in T. \quad (3.14)$$

We set up the correspondence between binaries and decision variables $d_{jt} \in [0, d]$. As in previous cases, $d_{jt} = 0$ if the job is not started before period t or it is finished in some period before t :

$$d_{jt} \leq d (S_{jt}^* - E_{jt}^*), \forall j \in J, \forall t \in T. \quad (3.15)$$

If a job is implemented in three or more periods, inside all periods between the first one and the last one job duration is the same as the length of the period:

$$d_{jt} \geq d (S_{jt}^* + S_{j,t-1}^* - 1 - E_{jt}^* - E_{j,t+1}^*), \forall j \in J, \forall t \in T. \quad (3.16)$$

It is also necessary to configure precedence constraints. Successor and predecessor both can be implemented in one period only if it is the last period of the predecessor and the first period of the successor:

$$S_{j_2 t}^* \leq E_{j_1, t+1}^*, \forall t \in T, \forall (j_1, j_2) \in P; \quad (3.17)$$

$$d_{j_1 t} + d_{j_2 t} \leq d, \forall t \in T, \forall (j_1, j_2) \in P. \quad (3.18)$$

Resource Allocation Constraints and Objective Function

In this set of constraints, we use only d_{jt} from scheduling decision variables. We denote as $c_{jrt} \in [0, p_{max,jr}d]$ the volume of work related to job $j \in J$ in period $t \in T$ done by resource $r \in R$. It has upper and lower limits defined by the minimal and maximal amount of assigned resources and job duration:

$$p_{min,jr}d_{jt} \leq c_{jrt} \leq p_{max,jr}d_{jt}, \forall j \in J, \forall r \in R, \forall t \in T. \quad (3.19)$$

All resource types must implement given total amount required to complete each job:

$$\sum_{t \in T} c_{jrt} = W_{jr}, \forall j \in J, \forall r \in R. \quad (3.20)$$

We define the cost of the overload of resource $r \in R$ in period $t \in T$ as o_{rt} . In the objective function, we use the amount of extra usage of each resource o_{rt} , defined by the following constraints:

$$o_{rt} \geq e_r \left(\sum_{j \in J} c_{jrt} - L_{rt} \right), \forall t \in T, \forall r \in R. \quad (3.21)$$

The use of this variable helps to calculate precisely the amount of extra resources needed. It is impossible to use directly the difference between the sum of c_{jrt} and the available

resource L_{rt} in a given period as we need to cut off the cases when this difference is negative. The objective function is the minimization of the extra resource allocation cost:

$$\text{Minimize } \sum_{r \in R} \sum_{t \in T} o_{rt}. \quad (3.22)$$

3.3 Generalized model properties

3.3.1 Model instance

A problem instance is described by N parameters (see Table 3.1), where

$$N = 1 + |R|(|T| + 1 + 3|J|) + |P|, \quad (3.23)$$

and in general case $|P|$ is bounded by $\frac{|J|(|J|-1)}{2}$.

Definition 1. *By instance I , we define the following set of parameters,*

$$I = \{d^I, L_{rt}^I, e_r^I, W_{jr}^I, p_{min,jr}^I, p_{max,jr}^I, (i, j)_p^I; t \in T, r \in R, j \in J, p \in P\} \quad (3.24)$$

We will skip the upper instance index for parameters if it is not required for understanding.

According to this formulation, the solvability criterion can be defined for a given input data instance I as follows. We define it in this subsection and later use it to improve the basic Benders decomposition scheme. An instance I is solvable if it is possible to build a feasible solution for it respecting all constraints.

To guarantee that a feasible solution exists for a given instance I , we have to consider the following elements. We can state that resource allocation is guaranteed to be possible if all processing times are in a correct range, i.e. $\forall j \in J, \sum_{t \in T} d_{jt} \in [d_{min,j}, d_{max,j}]$. These bounds depend on minimal and maximal resource usage. We define for each job j the minimal and maximal duration as $d_{min,j}$ and $d_{max,j}$:

$$d_{min,j} = \max_{r \in R} \frac{W_{jr}}{p_{max,jr}}, \quad d_{max,j} = \min_{r \in R} \frac{W_{jr}}{p_{min,jr}}, \quad (3.25)$$

Here we use the maximum function to calculate the value of minimal job duration. Each ratio $\frac{W_{jr}}{p_{max,jr}}$ defines the minimal duration required to satisfy constraints for resource r . If multiple resources are involved in the same job, we take the maximal value. The same logic is applied to calculate the maximal duration.

The solution must also satisfy the precedence relations. Without strict constraints involving resource limit variables L_{rt} (used only to evaluate the objective function value), it is possible to formulate these requirements in the following way. Precedence constraints form several chains (with possible intersections and branches) of jobs. There is at least

one guaranteed feasible solution if it is possible to implement the longest chain within a given planning horizon (with a deadline as a bound). With various job duration, it is enough to consider a chain with minimal allowed feasible durations $d_{min,j}$ to state that there is at least one feasible solution. To find the minimum required length of the longest job chain, we introduce several other supplementary parameters in addition to $d_{min,j}$ and $d_{max,j}$.

We calculate the set of $g_{j_1j_2}$ – minimal required intervals between the starts of two jobs $j_1 \in J$ and $j_2 \in J$, $j_1 \neq j_2$, these values are determined by the form of the directed acyclic graph of precedence relations. We apply the method of the longest path calculation in the directed acyclic graphs with topological sorting. Furthermore, with the classic addition of dummy "0" and " $|J| + 1$ " jobs with zero work volumes, we find the chain of jobs with the longest total minimal required duration in the precedence graph (where "0" job is the predecessor of all jobs and " $|J| + 1$ " job is the successor of all jobs). In our notations it equals $g_{0,|J|+1}$.

Definition 2. *Instance I is solvable (i.e. a feasible solution exists), if in this instance:*

- *obviously, the limits of all assigned resources are defined correctly for all jobs,*

$$p_{min,jr} \leq p_{max,jr}, \quad \forall j \in J, \quad \forall r \in R; \quad (3.26)$$

- *for every job $j \in J$ the minimal duration is less than or equal to the maximal duration,*

$$d_{min,j} \leq d_{max,j}, \quad \forall j \in J; \quad (3.27)$$

- *the chain of jobs with the longest total minimal required duration in the precedence graph is less than the total duration of the planning horizon,*

$$g_{0,|J|+1} \leq |T|d. \quad (3.28)$$

Condition (3.27) is necessary to check the feasibility of values of d_{jt} if multiple resources are used for the same job. If each job requires exactly one resource, condition (3.26) is enough. In Chapter 4, constraints (3.27) extend the model in proposed decomposition approach. Chapter 5 also exploits these notations to study other model properties.

3.4 Computational experiments

3.4.1 Tests on existing benchmarks for RLP

To evaluate our approach, we firstly evaluated the possibility to use the datasets presented in [149] and [150]. Kolish et al. [149] described the results on the dataset for the basic

RLP formulation. These instances are available online¹. Instances were generated with a tool ProGen/Max that constructs instances for project scheduling problems with minimal and maximal time lags between jobs. The acyclic precedence graph is constructed in two different ways (see Schwindt [151]). This dataset includes instances with 10–30 activities and 1–5 resources, each job requiring only one resource type. There is also a second set of instances². The computational results on these instances for basic RLP can be found in the paper of Rieck et al. [150]. These instances have 10-50 jobs and 1-5 resources.

However, these datasets were built for the formulation of RLP that differs significantly from our case. For our problem, the following parameters must be additionally defined:

- total job workload, since in basic RLP, each job has a constant allocated resource volume per period;
- lower and upper bounds for resource allocation, since in basic RLP, job duration is fixed and these limits are not defined;
- function of resource availability per time period, since in basic RLP, it is fixed for each resource type.

We also need to test larger instances with 10 resources and 50 jobs. Hans [81] and Baydoun et al. [18] are used the same dataset, further called Rough-Cut Capacity Planning (RCCP) dataset. This dataset is more relevant in terms of scale and data to our problem. Although there is a lack of some parameters such as lower limits or resource cost, it is possible to use this dataset to illustrate the structural difference of the solutions provided by the model of Baydoun et al. [18] and ours. The dataset includes 450 instances in total, divided into 45 groups. Each group was generated with a given number of jobs, resource types and average slack. The list of used values is presented in Table 3.3. The average slack ϕ is defined as follows where D_j is the deadline and R_j is the release date of job j :

$$\phi = \frac{\sum_{j \in J} D_j - d_{min,j} - R_j + 1}{|J|},$$

Using these parameters, the instances are generated with procedure ProGen. The first step is to make several branches from the initial activity (defined as 0 or a source) and several branches from the activities with no successors to the final one (usually defined as $n+1$ or a sink). The number of branches in each case defines the structure of the precedence graph. At the next step, additional arcs are generated to fit defined parameters for the

¹<https://www.wiwi.tu-clausthal.de/en/chairs/produktion/research/research-areas/project-generator/rlpmax/>

²<https://www.wiwi.tu-clausthal.de/abteilungen/unternehmensforschung/forschung/benchmark-instances/exact-results-for-single-mode-resource-levelling-problems/>

problem instance. Another important point in the procedure is to avoid the so-called redundant arcs. An arc (j_0, j_s) is called redundant if there are arcs $(j_0, j_1), \dots, (j_{s-1}, j_s) \in P$ and $s \geq 2$. In the procedure, this is verified with a set of special rules.

Table 3.3: The scales of parameters in the datasets used by Baydoun et al. [18]

$ J $	10	20	50		
$ R $	3	10	20		
ϕ	2	5	10	15	20

In order to compare our model with the model presented by Baydoun et al. [81] with the same solver, we create a relaxed version of our model that we refer to as aggregated fraction model. It is constructed with the use of a fraction decision variable $l_{jt} \in [0, 1]$ instead of generalized separated variables c_{jrt} , and $c_{jrt} = W_{jr}l_{jt}$, $j \in J$, $r \in R$, $t \in T$. We run both original and relaxed models using the IBM ILOG CPLEX 12.8 mathematical programming solver with Java code on a workstation with 4 thread 2.70 GHz processor and 8 Gb RAM. For each instance I we compare the ratio of objective function values in both solutions, defined by $X(I)$:

$$X(I) = \frac{V_c(I)}{V_l(I)}, \quad (3.29)$$

where $V_c(I)$ and $V_l(I)$ are the optimal cost obtained respectively with decision variables c_{jrt} and l_{jt} .

For our original model, we additionally use the following parameters:

- we use the value of the minimal duration of activity j from RCCP to calculate the value of $p_{max,jr}$;
- with a lack of lower limits in RCCP we assume $p_{min,jr} = 0.4p_{max,jr}$;
- with a lack of prices in RCCP all extra resource cost values $e_r = 1.0$.

The results are illustrated in Table 3.4. In the first block (a) we present the results obtained within time limit of 300 seconds for both models. We can conclude, that for 5/6 of solved instances the generalized model provided better solutions. In the second block (b) we compare the solutions obtained within the same computational time, required by the aggregated model to reach an optimal solution. In all cases, the values of X are larger than the results in the block (a) but still less than 1.0, so we can conclude that: although our model is larger in terms of a number of decision variables, it provides better solutions within the same time limit (mostly suboptimal, this means that by relaxing time limit, better solutions can be found).

Table 3.4: $X(I)$ values for instances of RCCP datasets in two cases: (a) without time limit and (b) with time limit equal to the solution time to optimality for the aggregated fraction model.

	Solution time	Data set	Min	Q1	Median	Mean	Q3	Max
(a) not fixed		<i>j10_r03</i>	0.76	0.84	0.88	0.88	0.91	0.97
		<i>j10_r10</i>	0.93	0.96	0.97	0.96	0.98	0.99
		<i>j10_r20</i>	0.99	0.99	1.00	0.99	1.00	1.00
		<i>j20_r03</i>	0.71	0.82	0.85	0.84	0.88	0.94
		<i>j20_r10</i>	0.87	0.92	0.95	0.94	0.96	0.98
		<i>j20_r20</i>	0.97	0.98	0.98	0.98	0.99	1.00
(b) equal		<i>j10_r03</i>	0.81	0.88	0.90	0.91	0.95	1.02
		<i>j10_r10</i>	0.93	0.97	0.97	0.97	0.99	1.00
		<i>j10_r20</i>	0.99	0.99	1.00	1.00	1.00	1.01
		<i>j20_r03</i>	0.73	0.82	0.86	0.86	0.90	0.95
		<i>j20_r10</i>	0.87	0.92	0.95	0.94	0.97	0.98
		<i>j20_r20</i>	0.96	0.98	0.99	0.99	0.99	1.00

3.4.2 Generation of new instances

Data completeness and diversity in benchmarks is an important issue, discussed in the literature as a separated object of research. Here we briefly describe the most important points and methods that we used to generate new diverse and representative datasets. We note that in our model there are three major interacting parts of parameters:

1. given resource level L_{rt} ;
2. job-resource linking parameters (W_{jr} , $p_{min,jr}$ and $p_{max,jr}$);
3. precedence relations P .

A given resource level does not impact the feasibility of a schedule, it only impacts the range of possible objective function values in a set of solutions. The second part involves all parameters related to jobs. For these parameters, we have determined the conditions of solvability (see Definition 2). Precedence relations are important both in terms of feasibility and the quality of solutions. They are represented as a direct acyclic graph with activities on nodes. On the one hand, it is necessary to generate a sophisticated structure that represents precedence relations of activities in real projects. On the other hand, it is necessary to be able to vary it and provide diverse structures. To generate the various precedence relations one should not only construct formally different direct acyclic

3. produce random $p_{min,jr}$ and $p_{max,jr}$ such that an instance is solvable, i.e.
 - $p_{min,jr} \leq p_{max,jr}$;
 - $d_{min,j} \geq d$;
 - resulting job duration bounds are synchronized $d_{min,j} \leq d_{max,j}$;
 - longest path in graph $|P|$ composed with minimal durations is less than the length of planning horizon.
4. construct L_{rt} in a range $[0, \frac{\sum_{j \in J} p_{max,jr} d}{SP \cdot |J| + 1}]$;

Here we define an upper bound for L_{rt} in the following way: $\sum_{j \in J} p_{max,jr} d$ is the maximal total required amount for resource $r \in R$ if we do not take into account precedence constraints. With a given number of precedence constraints, the highest consumption is reached with parallel jobs ($SP = 0$) and the lowest consumption limit is for a chain ($SP = 1.0$). With the denominator, we tune the limit according to the graph structure.

3.4.3 Comparison of time-indexed RLP formulations

We study three different versions of the mathematical model. Each version has the same Objective Function (3.22), and resource allocation constraints (3.19)–(3.21) and a different implementation of scheduling constraints:

1. job on–off variables, presented in Section 3.2.2;
2. job pulse start–end variables, presented in Section 3.2.2;
3. job step start–end variables, presented in Section 3.2.2.

To study the impact of these scheduling constraints on the performance of the model, we run numerical experiments on two datasets. Each dataset includes 100 instances. Parameters of these datasets are presented in Table 3.5. The instances were generated using a continuous uniform distribution of parameters. Precedence graphs were created with the given total number of directed edges under the condition of its acyclicity.

Table 3.5: Instance datasets parameters.

Data Set	$ T $	d	$ J $	$ R $	$ P $	L_{rt}	W_{jr}	$p_{min,jr}$	$p_{max,jr}$
<i>inst_j10_r5</i>	15	1	10	5	10	[0.0, 70.0]	[30.0, 50.0]	[1.0, 5.0]	[6.0, 10.0]
<i>inst_j15_r5</i>	20	1	15	5	15	[0.0, 70.0]	[30.0, 50.0]	[1.0, 5.0]	[6.0, 10.0]

We define three Mixed-Integer Linear Programming models. These models are implemented using the IBM ILOG CPLEX 12.8 mathematical programming solver with Java code on a workstation with 4 thread 2.70 GHz processor and 8 Gb RAM. Figure 3.5 presents solution times obtained for these two datasets with three different model versions.

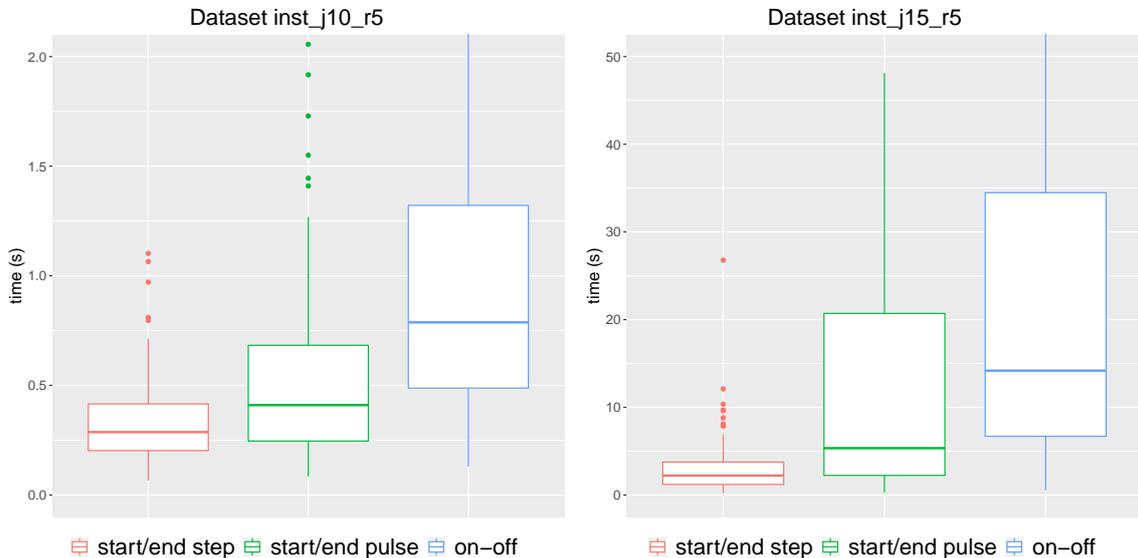


Figure 3.5: Different model formulation time boxplots for instances of datasets *inst_j10_r5* and *inst_j15_r5*.

We can conclude that the best performance in terms of solution time is obtained for the generalized model with step formulation of scheduling constraints. This type of constraint has been also used in models of Baydoun et al. [18] and Bianco et al. [19]. In next sections, we use the generalized model with step start and end variables (presented in Section 3.2.2) to compare it with an aggregated fraction approach.

3.4.4 Results of flexible resource allocation

RCCP benchmark tests have already demonstrated that flexible resource allocation leads to solutions with lower objective function values (see Table 3.4). In this section, the generalized model is compared to aggregated fraction model again. Additional tests give a more precise picture for comparison. Firstly, we run both models and compare the time spent to construct the optimal solution. The results presented in Figure 3.6 confirm that the larger generalized model is slower.

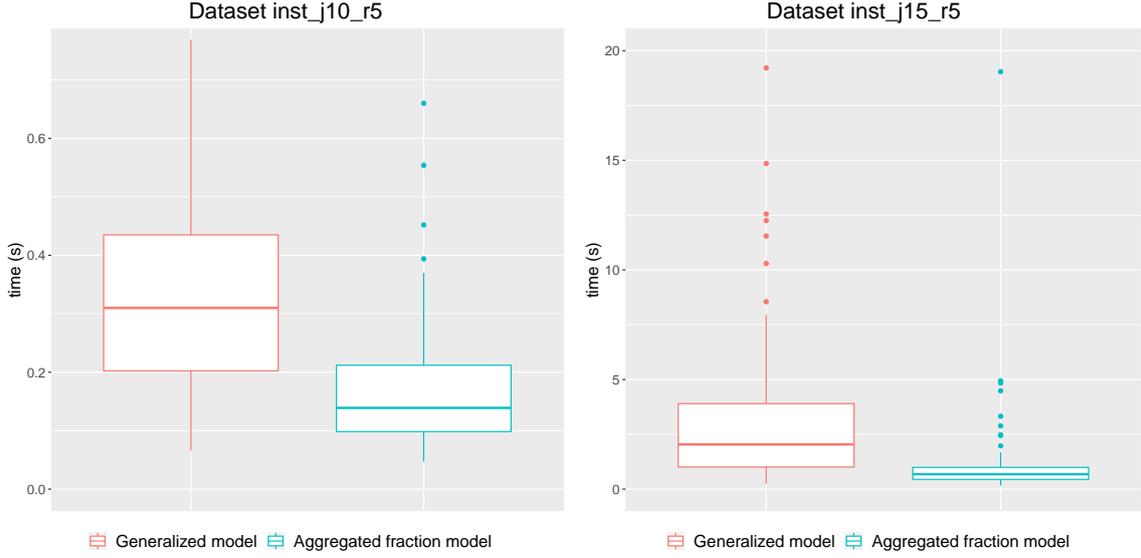


Figure 3.6: Time boxplots for datasets $inst_j10_r5$ and $inst_j15_r5$.

However, the generalized model provides more flexible solutions that allow to get much better solutions in terms of the objective function value. In Table 3.6 there is summary data of $X(I)$ for two new datasets of instances. On average, for both datasets, the solution provided by the generalized model has two times less objective function value. The same result is confirmed if the solution time is limited to the time required for faster aggregated model to reach the optimal solution, the results of this comparison are presented in Table 3.7.

Table 3.6: $X(I)$ values for instances of datasets $inst_j10_r5$ and $inst_j15_r5$.

Data Set	Min	Q1	Median	Mean	Q3	Max
$inst_j10_r5$	0.17	0.41	0.49	0.48	0.55	0.71
$inst_j15_r5$	0.25	0.42	0.51	0.50	0.56	0.72

Table 3.7: $X(I)$ value summary for instances of datasets $inst_j10_r5$ and $inst_j15_r5$, with time limit equal to the optimal solution time for the aggregated fraction model.

Data Set	Min	Q1	Median	Mean	Q3	Max
$inst_j10_r5$	0.19	0.45	0.51	0.53	0.59	0.87
$inst_j15_r5$	0.25	0.46	0.55	0.55	0.61	0.9

The value of $X(I)$ depends on the instance parameters. For example, if the availability of resources is higher in periods, then the objective function value decreases for both

models. In this case, we obtain lower values of $X(I)$. We can demonstrate this on new datasets $inst_j10_r5_2$ and $inst_j15_r5_2$ with distribution $L_{rt} \in [0, 140]$ instead of $L_{rt} \in [0, 70]$, see Table 3.8.

Table 3.8: $X(I)$ values for instances of datasets $inst_j10_r5_2$ and $inst_j15_r5_2$, in two cases: a) without time limit and b) with time limit equal to the optimal solution time for the aggregated fraction model.

Time limit	Data set	Min	Q1	Median	Mean	Q3	Max
a) Not fixed	$inst_j10_r5_2$	0.1	0.3	0.36	0.36	0.42	0.63
	$inst_j15_r5_2$	0.04	0.26	0.32	0.32	0.39	0.52
b) Aggregated fraction model solution time	$inst_j10_r5_2$	0.1	0.33	0.4	0.41	0.47	0.97
	$inst_j15_r5_2$	0.09	0.27	0.37	0.36	0.43	0.59

For further analysis, we generate larger instances and vary the number of resource types: we generate two datasets with 5 and 10 resource types and 30 jobs. In Table 3.9, we present the ranges used to generate these datasets. Each dataset contains 30 instances.

Table 3.9: Parameters for large datasets

Data Set	$ T $	d	$ J $	$ R $	$ P $	L_{rt}	W_{jr}	$p_{min,jr}$	$p_{max,jr}$
$inst_j30_r5$	35	1	30	5	30	[0.0, 70.0]	[30.0, 50.0]	[1.0, 5.0]	[6.0, 10.0]
$inst_j30_r10$	35	1	35	10	30	[0.0, 70.0]	[30.0, 50.0]	[1.0, 5.0]	[6.0, 10.0]

In Table 3.10 we present the results for these datasets obtained within 5 minute time limit. We note that we compare suboptimal solutions, obtained in a given time limit.

Table 3.10: $X(I)$ value summary for instances of datasets $inst_j30_r5$ and $inst_j30_r10$, with a 5 min time limit.

Data Set	Min	Q1	Median	Mean	Q3	Max
$inst_j30_r5$	0.33	0.45	0.48	0.49	0.53	0.62
$inst_j30_r10$	0.42	0.49	0.51	0.51	0.54	0.58

These results show that within the same time limit, our model provides better solutions in all cases. The worst-case ratio is around 0.6. We illustrate the gap values in Table 3.11 to make conclusions about the real optimal objective function values. The

aggregated fraction model has a low relative gap in all cases. No significant progress can be achieved by the aggregated fraction model with higher time limits, so it cannot outperform generalized model in solution quality.

Table 3.11: Relative gap value summary for instances of datasets $inst_j30_r5$ and $inst_j30_r5$, with 5 min time limit.

Data Set	Model	Min	Q1	Median	Mean	Q3	Max
$inst_j30_r5$	Aggregated fraction	0.004	0.011	0.016	0.018	0.023	0.053
	Generalized	0.05	0.11	0.16	0.16	0.19	0.3
$inst_j30_r10$	Aggregated fraction	0.007	0.01	0.012	0.014	0.019	0.032
	Generalized	0.07	0.12	0.14	0.16	0.18	0.38

Computational experiments confirm that we can achieve better solutions within the same solution time limit with the same solver if we apply the generalized formulation approach.

3.4.5 Discrete Resource Case

In the Objective Function (3.22), continuous overloading variables are used to calculate the cost of extra resources. It is compliant with such continuous resources as electricity or heat. However, in practice, such resources as machines or human operators can be only available in discrete units. For this case, two possible models can be used. Firstly, decision variable o_{rt} can be defined as integer with the minimal unit of each resource q_r . New variables o_{rt}^* set the number of extra units used and they replace o_{rt} in Objective Function (3.22) and Constraint (3.21):

$$\text{Minimize } \sum_{r \in R} \sum_{t \in T} e_r q_r o_{rt}^*; \quad (3.30)$$

$$q_r o_{rt}^* \geq \sum_{j \in J} c_{jrt} - L_{rt}, \quad \forall t \in T, \quad \forall r \in R. \quad (3.31)$$

We define this model as *DO* (discrete objective). This case can be used not only for discrete resources, but it also suits the usual practice when additional resources could be demanded in some packages, for example, the batteries. Secondly, it is also possible to define other decision variables related to resource allocation as integers. This corresponds to the case when we have discrete resources and we allocate a discrete amount of workload to all periods. We define this model as *DO&R* (discrete objective and resources).

A computational experiment has been run to compare the behavior of continuous and discrete versions of the model. Seven models with different types and parameters of

overload variables were considered: The original model with continuous overload variables, three versions of the discrete model with different resource unit size $q_r = q, \forall r \in R$, equal to 1, 3, and 5, defined as DO , and the same values of q for the discrete resource allocation case defined as $DO\&R$.

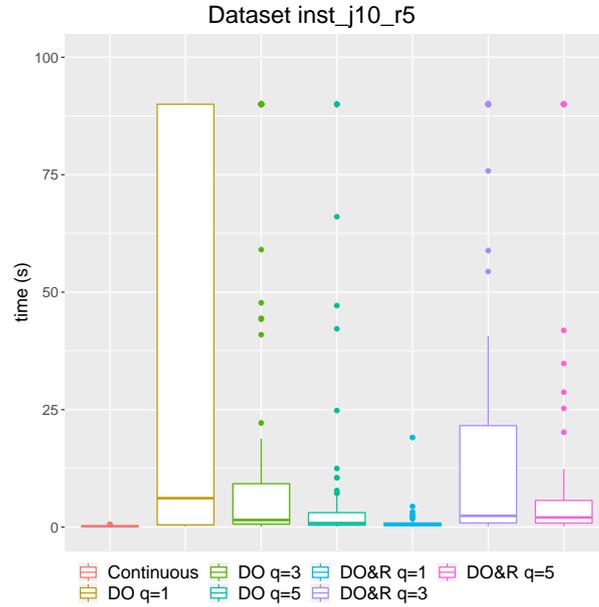


Figure 3.7: Time boxplot for different cases of discrete objective and allocation for dataset $inst_j10_r5$.

Figure 3.7 presents the computational results for dataset $inst_j10_r5$ with a time limit of 90 s for all models. Each column is a boxplot that aggregates the data about the solution time, defining the median, lower and upper values, and quartiles. In addition, Table 3.12 provides the mean values of the objective function, solution time, and gap which was not presented in Figure 3.7.

Table 3.12: Mean values for different models.

Model Overload Type	Objective	Time, s	Gap	δ
Continuous	471.5	0.18	0	-
DO $q = 1$	474.1	38.3	0.001	0.006
DO $q = 3$	487.4	18.6	0.001	0.04
DO $q = 5$	502.6	8	0.0005	0.02
$DR\&O$ $q = 1$	482.5	0.9	0	0.06
$DR\&O$ $q = 3$	469.9	20.3	0.001	0.07
$DR\&O$ $q = 5$	471.5	12	0.001	0.09

Here we also compare the values of the optimal solutions provided by discrete model

V_{DO} and continuous generalized model (V_C), and calculate relative delta. For example, for instance I and model DO it is

$$\delta_{V_{DO}}(I) = \frac{V_{DO}(I) - V_C(I)}{V_C(I)}$$

As it could be expected, the computation time for the discrete model is higher than for the basic continuous model. For some instances, no optimal solution has been reached in 90 s for the discrete model, while all the instances were solved for the basic continuous model faster than in a second. However, the information about the gap provided by the solver shows that the main issue is in the proof of optimality: all instances had a very small gap value when the time limit was reached. It is also interesting to note that the second model type $DO\&R$ with the discrete allocation of resources provides optimal solutions much faster than the first type DO . Consequently, it is possible to use these discrete models with some reasonable small gap tolerance.

3.5 Conclusions

In this chapter, we propose a new mathematical formulation for a Resource Leveling Problem with a variable duration of jobs. We consider the extra resource usage cost as the objective function which has to be minimized. Extra resources are needed in case of a lack of available resources, as jobs must be processed within a fixed planning horizon before a deadline. The main idea behind this new formulation is to provide a more flexible allocation of different resources to jobs, which allows minimizing the total cost. We consider different time-indexed formulations for scheduling decision variables and constraints.

This new formulation approach is compared to other RLP formulations with overload which were found in the literature. We defined them as aggregated fraction models to underline the main difference. The numerical experiments show that, even if the generalized formulation uses more variables and constraints, it provides better solutions. The same conclusion has been made for different numerical experiments for various datasets. We have also proposed and evaluated the variant with discrete resources.

In other words, this chapter described the idea of generalizing the model and its properties. The next chapter offers an accelerated solution algorithm based on Benders decomposition.

Chapter 4

Benders decomposition for RLP

This chapter presents a new solution approach based on Benders decomposition for the generalized RLP model introduced in the previous chapter. We enhance our algorithm with several improvements, such as valid inequalities reducing the number of iterations, and resource-disaggregated subproblems with multi-cut generation and additional lower bounds accelerating the convergence at each iteration. We present our approach and evaluate its performance in a numerical experiment.

The contribution of this chapter was also presented in the following works:

- *Tarasov, I.; Haït, A.; Battaïa, O. Benders decomposition algorithm for a generalized Resource Leveling Problem. 21ème Congrès annuel de la Société Française de Recherche Opérationnelle et d'Aide à la Décision (ROADEF 2020), Feb 2020, Lorient, France.;*
- *Tarasov, I.; Haït, A.; Battaïa, O. Benders decomposition for a Period-Aggregated Resource Leveling Problem with variable job duration. Computers & Operations research, accepted for publication.*

4.1 Basic approach

The original Benders decomposition is applicable to MILP, e.g. in this form:

$$\begin{aligned} \min \quad & f^T y + c^T x \\ \text{s.t.} \quad & Ay = b \\ & By + Dx = d \\ & y \in \mathbb{Z}_+^{n_1} \\ & x \geq 0, x \in \mathbb{R}^{n_2} \end{aligned} \tag{4.1}$$

$$\begin{aligned} \text{where: } & f \in \mathbb{R}^{n_1}, c \in \mathbb{R}^{n_2}, \\ & A \in \mathbb{R}^{m_1 \times n_1}, \text{ and } b \in \mathbb{R}^{m_1} \\ & B \in \mathbb{R}^{m_2 \times n_1}, D \in \mathbb{R}^{m_2 \times n_2}, \text{ and } d \in \mathbb{R}^{m_2}. \end{aligned}$$

Two types of decision variables are involved. There is an integer vector $y \in \mathbb{Z}_+^{n_1}$ and a continuous $x \in \mathbb{R}^{n_2}$, both with positive values. The constraint set can be also separated: there are constraints related to the integer vector only, and also the connecting constraints defining the values of continuous vector x . It is possible to reformulate it as a master problem:

$$\begin{aligned} \min \quad & f^T \bar{y} + \eta \\ \text{s.t.} \quad & Ay = b \\ & y \in \mathbb{Z}_+^{n_1}; \end{aligned} \tag{4.2}$$

whereas the values of η for a given \bar{y} are derived from the subproblem:

$$\begin{aligned} \min \quad & \eta = c^T x \\ \text{s.t.} \quad & Dx = d - B\bar{y} \\ & x \geq 0, x \in \mathbb{R}^{n_2}. \end{aligned} \tag{4.3}$$

If model (4.3) does not have a solution, then there is no feasible solution for model (4.2). Otherwise, if (y^*, x^*) is an optimal solution of the original MILP (4.2), then x^* is an optimal solution of subproblem (4.3).

Subproblem (4.3) is LP. With vector π corresponding to the original constraint $Dx = d - B\bar{y}$ its dual has a form

$$\begin{aligned} \max \quad & \pi^T (d - B\bar{y}) \\ \text{s.t.} \quad & \pi^T d \leq c \\ & \pi \in \mathbb{R}^{m_2}. \end{aligned} \tag{4.4}$$

This dual subproblem has several important properties according to the duality theory.

- The optimal value of the objective function is the same for an LP and its dual form (strong duality theorem). In this case, $c^T x^* = \pi^{*T} (d - B\bar{y})$, if x^* and π^* are the optimal vales.
- The space of feasible solutions does not depend on the values of \bar{y} .
- If LP (4.3) does not have a solution, then its dual LP (4.4) is not bounded. If LP (4.3) has a finite solution, then dual LP (4.4) also has a finite solution.
- With a nonempty space of feasible solutions $\pi^T d \leq c$, the dual LP can be either unbounded or bounded. The unbounded dual subproblem states that a given vector \bar{y} is not feasible.
- Infeasible vectors \bar{y} can be eliminated from the solution space of the master problem with additional cuts, if we get all the set of extreme rays P_f :

$$r_p^T (d - B\bar{y}) \leq 0; \forall p \in P_f. \tag{4.5}$$

These cuts are defined as feasibility cuts.

- With a finite optimal objective function value of the dual subproblem (4.4), the optimal solution π is used to define the contribution to the objective function η within a set of extreme points P_o :

$$\pi_p^T(d - By) \leq \eta; \forall p \in P_o. \quad (4.6)$$

These cuts are defined as optimality cuts.

Figure 4.1 illustrates the general idea of Benders decomposition.

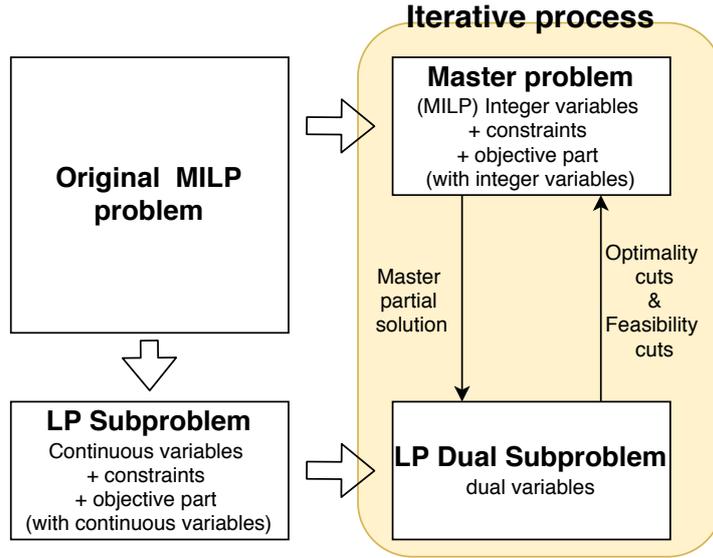


Figure 4.1: General scheme of a Benders decomposition

Finally, we can represent the master problem separately from the subproblem:

$$\begin{aligned} \min \quad & f^T y + \eta \\ \text{s.t.} \quad & Ay = b \\ & r_p^T(d - By) \leq 0; \forall p \in P_f; \\ & \pi_p^T(d - By) \leq \eta; \forall p \in P_o; \\ & y \in \mathbb{Z}_+^{n_1}. \end{aligned} \quad (4.7)$$

In order to avoid the consideration of full sets P_o and P_f in one model, an iterative approach is realized [109]. In each step, the master problem is solved and the optimal solution y is analyzed. A given solution y may lead to an unbounded dual subproblem, then an extreme ray is used to cut off this solution with an additional constraint of the form (4.5). Otherwise, an optimal solution of the dual subproblem is used to extend second set P_o and add new constraint of the form (4.6).

Basically, in each iteration, this process updates the lower bound of the objective function (formed by the master problem) and an upper bound (derived from the dual subproblem). Thus, it is possible to evaluate the gap and stop the process when the gap

is low enough or equals zero. In other words, the process stops with a decision that there is no need to add new cuts.

4.2 Benders decomposition for a generalized RLP

We propose the following realization of the Benders decomposition for the generalized formulation of RLP introduced in the previous chapter: scheduling decision variables S_{jt} , E_{jt} , d_{jt} are assigned to the master problem, and remaining workload distribution decision variables c_{jrt} and o_{rt} are assigned to the subproblem. This approach makes a compromise between the complexity of the master problem and the number of iterations required.

In our Benders decomposition scheme, the master problem and the dual subproblem are solved in the loop, in each iteration a feasibility cut or an optimality cut is generated. With the proposed decomposition scheme, we make the partial solution provided by the master problem always feasible, so we never use the generation of feasibility cuts. We achieve it with two additional cuts, bounding the overall duration of the job with values $d_{min,j}$ and $d_{max,j}$, which were calculated in preprocessing. In the next sections, there is a description of master MILP and subproblem LP. We start with the subproblem to explain the cuts first, and then introduce the master problem.

4.2.1 Subproblem

Consider the subproblem first. In the primal subproblem $PS(\bar{d}_{jt})$, the values of d_{jt} from the master problem are used as input given data:

$$\text{Minimize } \sum_{r \in R} \sum_{t \in T} o_{rt} \quad (4.8)$$

$$p_{min,jr} \bar{d}_{jt} \leq c_{jrt}, \quad \forall j \in J, \forall r \in R, \forall t \in T; \quad (4.9)$$

$$c_{jrt} \leq p_{max,jr} \bar{d}_{jt}, \quad \forall j \in J, \forall r \in R, \forall t \in T; \quad (4.10)$$

$$\sum_{t \in T} c_{jrt} = W_{jr}, \quad \forall j \in J, \forall r \in R; \quad (4.11)$$

$$o_{rt} \geq e_r \left(\sum_{j \in J} c_{jrt} - L_{rt} \right), \quad \forall t \in T, \forall r \in R; \quad (4.12)$$

$$o_{rt} \geq 0, \quad \forall t \in T, \forall r \in R. \quad (4.13)$$

Let $\alpha_{jrt} \geq 0$, $\beta_{jrt} \geq 0$, γ_{jr} , $\mu_{rt} \geq 0$ be the dual variables associated with constraints (4.9), (4.10), (4.11), (4.12), respectively. Dual subproblem $DS(\bar{d}_{jt})$, which is the dual of $PS(\bar{d}_{jt})$, can be stated as follows:

$$\text{Maximize } \sum_{j \in J} \sum_{r \in R} \gamma_{jr} W_{jr} - \sum_{r \in R} \sum_{t \in T} \mu_{rt} e_r L_{rt} - \sum_{j \in J} \sum_{t \in T} d_{jt} \sum_{r \in R} (\beta_{jrt} p_{max,jr} - \alpha_{jrt} p_{min,jr}) \quad (4.14)$$

$$\alpha_{jrt} - \beta_{jrt} + \gamma_{jr} - \mu_{rt}e_r \leq 0, \forall j \in J, \forall r \in R, \forall t \in T; \quad (4.15)$$

$$\mu_{rt} \leq 1, \forall r \in R, \forall t \in T. \quad (4.16)$$

4.2.2 Master problem

Let us define the set of dual subproblem solution points $(\alpha_{jrt}, \beta_{jrt}, \gamma_{jr}, \mu_{rt})$ as P_o . This set is updated after each iteration with new points provided by the solved dual subproblem. An extra variable η is the representation of the objective function in the master problem which links it with the corresponding set of optimality (4.18) cuts. In this case, master problem MP is presented in the following way:

$$\text{Minimize } \eta \quad (4.17)$$

$$\eta \geq \sum_{j \in J} \sum_{r \in R} \gamma_{jr} W_{jr} - \sum_{r \in R} \sum_{t \in T} \mu_{rt} e_r L_{rt} - \sum_{j \in J} \sum_{t \in T} d_{jt} \sum_{r \in R} (\beta_{jrt} p_{max,jr} - \alpha_{jrt} p_{min,jr}) \quad (4.18)$$

$$\forall (\alpha_{jrt}, \beta_{jrt}, \gamma_{jr}, \mu_{rt}) \in P_o;$$

We use all the constraints related to scheduling decision variables:

$$S_{jt} \geq E_{jt}, \forall j \in J, \forall t \in T; \quad (4.19)$$

$$S_{jt} \leq S_{j,t+1}, \forall j \in J, \forall t \in T; \quad (4.20)$$

$$E_{jt} \leq E_{j,t+1}, \forall j \in J, \forall t \in T. \quad (4.21)$$

$$d_{jt} \leq d(S_{jt} - E_{jt}), \forall j \in J, \forall t \in T. \quad (4.22)$$

$$d_{jt} \geq d(S_{jt} + S_{j,t-1} - 1 - E_{jt} - E_{j,t+1}), \forall j \in J, \forall t \in T. \quad (4.23)$$

$$S_{j_2t} \leq E_{j_1,t+1}, \forall t \in T, \forall (j_1, j_2) \in P; \quad (4.24)$$

$$d_{j_1t} + d_{j_2t} \leq d, \forall t \in T, \forall (j_1, j_2) \in P. \quad (4.25)$$

To achieve guaranteed feasibility of the solutions provided by the master problem (and to reach the bounded dual subproblem), we add the following constraints:

$$\sum_{t \in T} d_{jt} \geq d_{min,j}, \forall j \in J; \quad (4.26)$$

$$\sum_{t \in T} d_{jt} \leq d_{max,j}, \forall j \in J. \quad (4.27)$$

If the problem instance is solvable and the master problem includes constraints (4.26-4.27), the subproblem will always be bounded and therefore we can skip the stage of adding feasibility cuts to the master problem.

The solution of the master problem generates the values for S_{jt} , E_{jt} and d_{jt} that describes the schedule, and the subproblem provides the best corresponding resource allocation. We denote the objective function value of the solution for the master problem as $v(MP^n)$ and for the dual subproblem as $v(DS(\bar{d}_{jt}))$. The Benders scheme for our problem is presented in Figure 4.2. In blue blocks, we describe the partial solutions provided by the master and dual subproblem. We skip the condition of an unbounded master problem solution and the generation of feasibility cuts (red-colored part) as we use additional constraints to guarantee the feasibility of the master problem solution (and therefore the boundness of the dual subproblem). The stopping criterion is marked with green color: the optimal solution is obtained if the lower (LB) and upper (UB) bounds are equal. In our case, each iteration provides a new LB, which is the value of the objective obtained for the master problem, and an updated UB obtained with the dual subproblem (marked with a violet color).

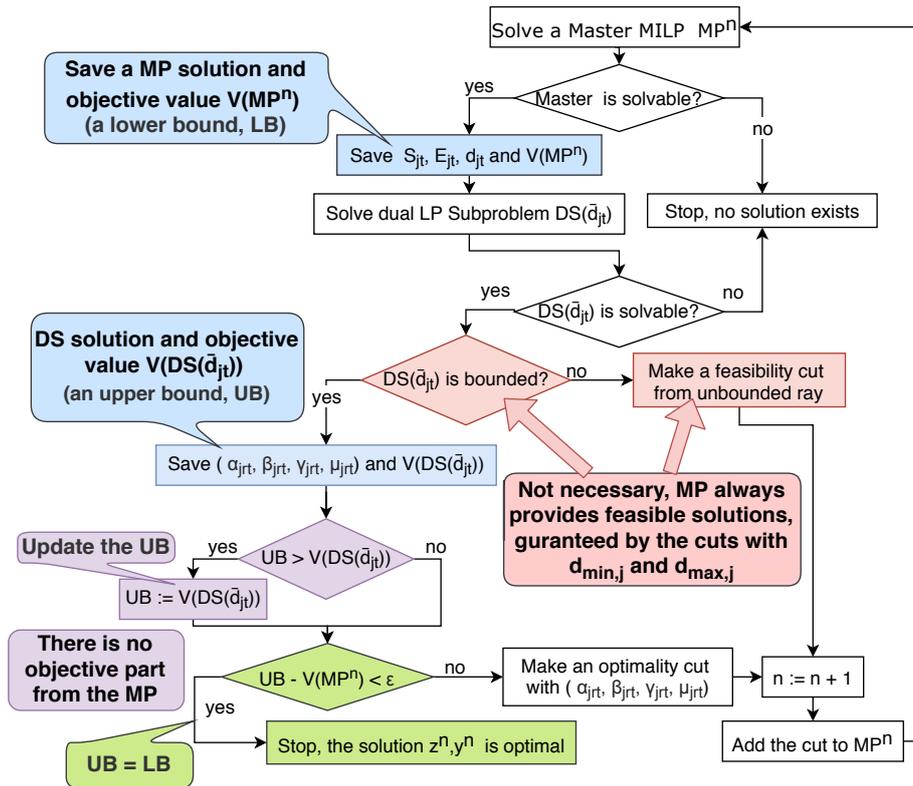


Figure 4.2: Basic Benders algorithm for considered problem

4.3 Algorithm improvements

In order to enhance the performances of our Benders decomposition, we develop the following improvements to the initial scheme. Firstly, we develop advanced procedures for calculating the lower bound for the master problem. One of the main reasons for the low performance of the decomposition scheme is the absence of the objective function component in the master problem. This slows progress in the lower bound produced by the master problem (we study a minimization problem). We propose some master problem upgrades to take into account lower bounds, as it is possible to give some estimations of the initial objective function value based only on the values of d_{jt} in the candidate solution. Such a construction of valid inequalities was described in [123]. Secondly, we present multi-cut generation from the disaggregation of the dual subproblem. Finally, the basic Benders decomposition scheme is reformulated to be used with the CPLEX callback technology.

4.3.1 LB for the Master problem: occupied work volume estimation

The idea of our procedure for calculating the lower bound can be described in the following way: for each resource, we compare the work volume required to perform all jobs with the estimated total work volume available with given resource capacity. In this case, the overload is equal to the difference between these two values.

We define as U_{rt} the total usage of resource r in time period t which is affordable with respect to the workload capacity in this period and the set of jobs which are scheduled in this period. An appropriate estimation can be achieved with two constraints:

$$0 \leq U_{rt} \leq L_{rt}, \forall r \in R, \forall t \in T; \quad (4.28)$$

$$U_{rt} \leq \sum_{j \in J} p_{max,jr} d_{jt}, \forall r \in R, \forall t \in T. \quad (4.29)$$

Constraints (4.28) set up the general bounds for U_{rt} which should not exceed the available resource limit in each period, and (4.29) define the upper value from the maximal possible consumption of the resource by jobs scheduled in the given period. The lower bound valid inequality has the following form:

$$\nu_r \geq e_r \left(\sum_{j \in J} W_{jr} - \sum_{t \in T} U_{rt} \right), \forall r \in R, \quad (4.30)$$

where ν_r is the lower bound contribution of resource r . This formulation provides a good lower bounds using only the scheduling part of the problem without taking into consideration job work volume distribution.

4.3.2 LB for the Master problem: resource constraints relaxation

The second option is to use several additional constraints in the master problem, defining the distribution of work volume (which are presented directly in the primal subproblem): aggregated (3.19), (3.20) and (3.21). We define decision variables C_{rt} , aggregating the contributions of all jobs, i.e. $C_{rt} = \sum_{j \in J} c_{jrt}$. We add the following constraints to the master problem:

$$\sum_{j \in J} p_{min,jr} d_{jt} \leq C_{rt} \leq \sum_{j \in J} p_{max,jr} d_{jt}, \quad \forall r \in R, \forall t \in T; \quad (4.31)$$

$$\sum_{t \in T} C_{rt} = \sum_{j \in J} W_{jr}, \quad \forall r \in R; \quad (4.32)$$

$$o_{rt} \geq e_r(C_{rt} - L_{rt}), \quad \forall t \in T, \forall r \in R; \quad (4.33)$$

This set of constraints also provides the lower bounds for the resource overload from the schedule.

These two approaches use a different logic: in 4.3.1 we take the difference between total required and available work volume with no direct consideration of work volume constraints, whereas in 4.3.2 we significantly simplify the original model with the work volume constraint disaggregation. Both approaches are used in tests separately.

4.3.3 Disaggregation of the cuts

In addition to the master problem lower bound improvements, it is possible to perform disaggregation and consider the resources separately due to the subproblem structure.

At each iteration of the algorithm, we solve $|R|$ subproblems $DS(\bar{d}_{jt})_r$ and produce $|R|$ optimality cuts. The dual subproblem $DS(\bar{d}_{jt})_r$ is the disaggregated part of $DS(\bar{d}_{jt})$:

$$\text{Maximize} \sum_{j \in J} \gamma_{jr} W_{jr} - \sum_{t \in T} \mu_{rt} e_r L_{rt} - \sum_{j \in J} \sum_{t \in T} d_{jt} (\beta_{jrt} p_{max,jr} - \alpha_{jrt} p_{min,jr}) \quad (4.34)$$

$$\alpha_{jrt} - \beta_{jrt} + \gamma_{jr} - \mu_{rt} e_r \leq 0, \quad \forall j \in J, \forall t \in T; \quad (4.35)$$

$$\mu_{rt} \leq 1, \quad \forall t \in T. \quad (4.36)$$

According to the lower bound improvements and changes of the subproblem, we obtain a new master problem MP^* with new sets of points $(\alpha_{jt}, \beta_{jt}, \gamma_j, \mu_t) \in P_{o,r}$ for optimality cuts. The set of decision variables also includes the original scheduling decision variables S_{jt} E_{jt} d_{jt} plus the lower bound decision variables, for occupied estimation LB (see 4.3.1) it is U_{rt} and ν_r for the objective function, for resource-distribution LB (see 4.3.2) it is

C_{rt} and o_{rt} for the objective function. We demonstrate the formulation of MP^* with the occupied estimation LB from 4.3.1 below.

$$\text{Minimize } \sum_{r \in R} \nu_r \quad (4.37)$$

Since our master problem always leads to a bounded dual subproblem, only optimality cuts are used. With new disaggregated dual subproblems, we add $|R|$ new cuts to the master problem (new points to the $P_{o,r}$) instead of one cut presented in (4.18):

$$\begin{aligned} \nu_r \geq \sum_{j \in J} \gamma_{jr} W_{jr} - \sum_{t \in T} \mu_{rt} e_r L_{rt} - \sum_{j \in J} \sum_{t \in T} d_{jt} (\beta_{jrt} p_{max,jr} - \alpha_{jrt} p_{min,jr}), \quad (4.38) \\ \forall (\alpha_{jrt}, \beta_{jrt}, \gamma_{jr}, \mu_{tr}) \in P_{o,r}, \forall r \in R; \end{aligned}$$

We add the lower bound calculation scheme

$$\nu_r \geq e_r \left(\sum_{j \in J} W_{jr} - \sum_{t \in T} U_{rt} \right), \forall r \in R; \quad (4.39)$$

$$0 \leq U_{rt} \leq L_{rt}, \forall r \in R, \forall t \in T; \quad (4.40)$$

$$U_{rt} \leq \sum_{j \in J} p_{max,jr} d_{jt}, \forall r \in R, \forall t \in T. \quad (4.41)$$

If we use the second resource-distribution LB, then these three constraints will be replaced by constraints (4.31)–(4.33) with additional variable C_{rt} and ν_r will be replaced by $\sum_{t \in T} o_{rt}$ in the objective function (4.37) and the constraint (4.38). In MP^* , we also use all the original constraints (4.19)–(4.25) plus additional feasibility constraints (4.26)–(4.27).

4.3.4 Branch&Benders cuts: using single search tree

Branch&Benders cuts are implemented in order to exploit a single search tree of the master problem instead of an iterative resolution to optimality. Technically, the single search tree is implemented with the so-called callback interfaces.

IBM ILOG CPLEX solver v12.10 was used to implement our model. Callback functions were implemented to generate Benders cuts for the Branch&Bound search. A callback is a user function that can be executed during the optimization process to get and modify such data as:

- Candidate – the solver has found a new integer-feasible candidate for an incumbent solution;
- GlobalProgress (LocalProgress) – progress in global (local thread) search;
- Relaxation – the solver has found a new solution which is not feasible;

- ThreadUp – new thread was activated;
- ThreadDown – the thread was deactivated;

ThreadUp and ThreadDown contexts are used to make the calculations with the dual subproblems thread-safe. If a thread was activated with the context ThreadUP, the dual subproblem copy is created and stored separately, and this copy is deleted with the ThreadDown context.

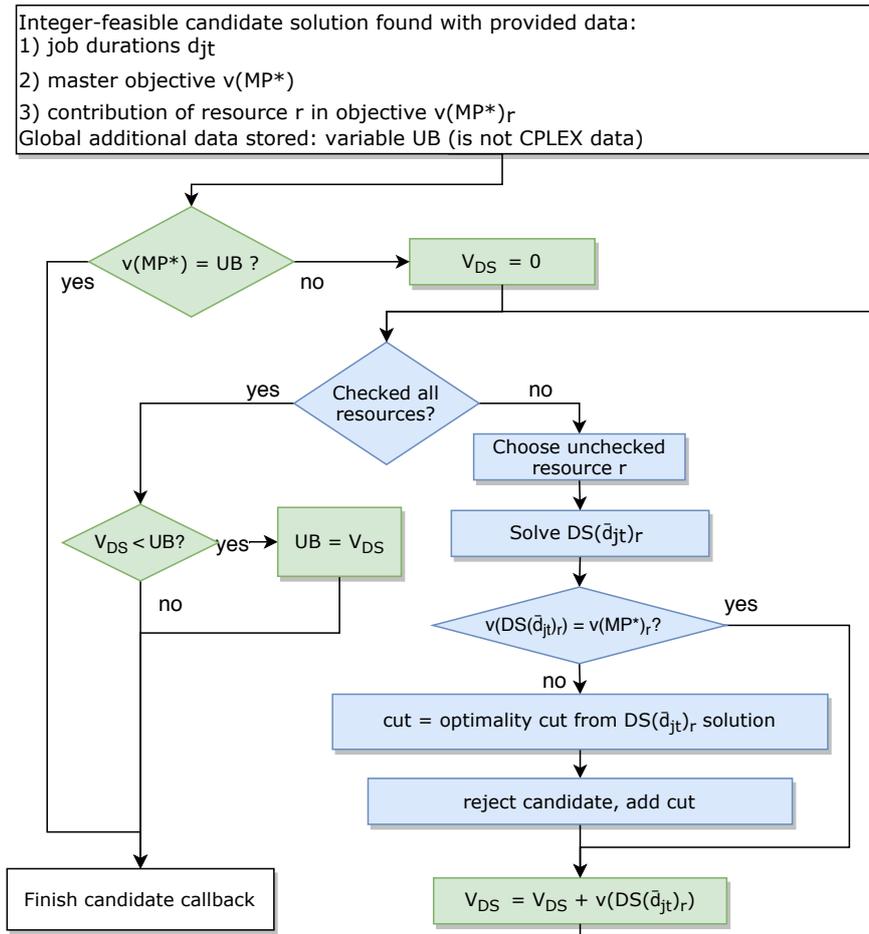


Figure 4.3: Callback procedure algorithm

We run master problem MP^* and set up the Candidate context. In our decomposition, the Candidate context is used as a pointer to start the resolution of the dual subproblem, see Figure 4.3. If a new integer-feasible solution was found in the master problem during the process, then we follow the basic part of our scheme. We consider the resource set and for all $r \in R$, we perform the following steps. The dual subproblem for resource r is constructed based on the candidate solution. An optimality cut is added to the master problem in the case where the optimality is not reached by the master problem, i.e. if the candidate objective function component value from resource r is not equal to the subproblem value.

The basic part of the scheme is marked in blue, and the blocks referring to the additional termination condition are marked in green. This additional condition allows us to skip several iterations in the loop when MP^* integer-feasible candidate with optimal objective function value was found. The condition skips the loop and leaves the context callback procedure when the value of the candidate's objective function $v(MP^*)$ is equal to the UB , i.e. when a candidate solution for the master problem has reached the minimal value of the global objective function which has been found before by the dual subproblems. As the $v(MP^*)$ is nondecreasing lower bound in the decomposition, and UB is the upper bound produced by dual subproblems, it is the termination condition.

We can also mention several additional important (in terms of performance) technical aspects:

- in JAVA a thread-safety requires to have a separated copy of a Subproblem (in our case, a set of $|R|$ subproblems);
- we rely on the example of a new Benders implementation with callbacks provided by IBM ("BendersATSP2.java" in a CPLEX code example pack);
- in this example a new Subproblem object is created when a thread is launched and destroyed when the thread is down, we improve it by keeping the same set of Subproblem objects with an iterative modification.

4.4 Computational tests

4.4.1 Models and settings

We run the experiments with IBM ILOG CPLEX solver version 12.10 on the workstation with an 8-core 2.60 GHz processor and 16 Gb RAM. Four models are compared, Table 4.1 provides the list of models and short notations. We compare the original model from Section 3.2.2 with three algorithms that use Benders decomposition. Firstly, the Benders decomposition scheme is implemented in the IBM ILOG CPLEX solver starting from version 12.7. It is possible to tune it with annotations, we annotate it to follow the proposed subproblem disaggregation by the resources. Secondly, we consider the algorithm which uses the master problem from the basic decomposition scheme with the callback Benders procedure and all the improvements. We proposed two different ways to implement the lower bounds and study them separately. Both LB approaches used simultaneously overconstrain the MP^* and lead to bad performance.

Table 4.1: List of compared solution approaches and their descriptions

Notation	Description
M1	the original model, presented in Section 3.2.2
M1+BD	original model with CPLEX included Benders decomposition
BC1	improved Benders Callbacks + lower bound 1 (see 4.3.1)
BC2	improved Benders Callbacks + lower bound 2 (see 4.3.2)

4.4.2 Generation of specific large-scale datasets

To evaluate our decomposition approach, we needed to test it on large-scale problem instances with a fully consistent set of required parameters. Since such instances were not available in the literature, we generated them in the following way. We used the same approach of data generation as in the RCCP dataset, especially for the generation of the precedence graph. Figure 4.4 illustrates two examples of precedence graph: the first was taken from this RCCP dataset and a second was generated on the same base for our instances.

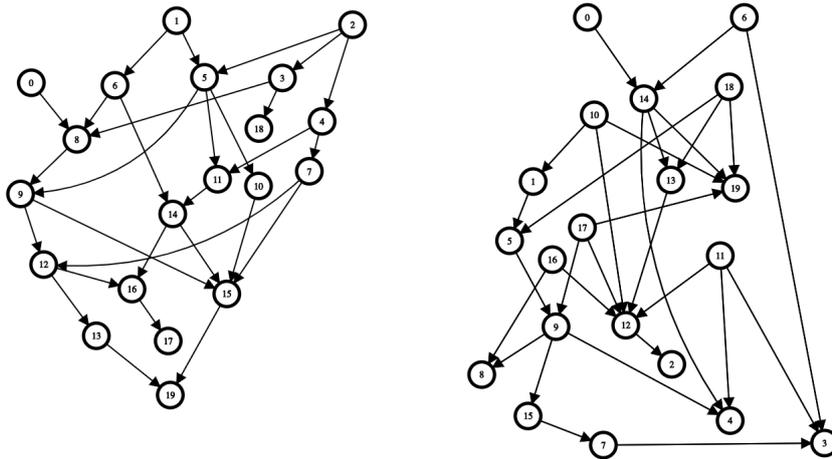


Figure 4.4: Two precedence graphs: the 1st one for an instance from RCCP dataset and the 2d one generated for our dataset

In order to evaluate our decomposition approach, three following aspects were important for data generation:

- a number of feasible solutions depends on the density of precedence graph and job duration upper and lower limits;
- the efficiency of Benders decomposition depends on the number of resource types (since we disaggregate the cuts by resources);

- a ratio of available resource amount and demand of activities has a strong influence on the objective function value.

Taking into account these aspects, we generated our dataset by varying the following parameters:

- the structure of the precedence graph;
- a number of resources;
- the distribution of resources through time periods.

The values of parameters were taken at random within feasible intervals in order to create feasible instances. For example, according to condition (3.27) and equations (3.25) it is necessary to consider the following relation between amount and limits of different resources:

$$\max_{r \in R} \frac{W_{jr}}{p_{max,jr}} \leq \min_{r \in R} \frac{W_{jr}}{p_{min,jr}}, \quad (4.42)$$

so

$$\forall j \in J, \forall r_1, r_2 \in R: \frac{W_{jr_1}}{p_{max,jr_1}} \leq \frac{W_{jr_2}}{p_{min,jr_2}} \quad (4.43)$$

To guarantee these conditions we have to set up the following intervals to generate the values of problem parameters:

$$\begin{aligned} W_{jr} &\in [W_{jr}^L, W_{jr}^U], \\ p_{min,jr} &\in [p_{min,jr}^L, p_{min,jr}^U], \\ p_{max,jr} &\in [p_{max,jr}^L, p_{max,jr}^U], \end{aligned}$$

with an additional condition for the bound values (maximal possible for the left side and the minimal possible for the right side)

$$\forall j \in J, \forall r_1, r_2 \in R: \frac{W_{jr_1}^U}{p_{max,jr_1}^L} \leq \frac{W_{jr_2}^L}{p_{min,jr_2}^U}. \quad (4.44)$$

Condition (4.44) defines the correspondence between the limits of intervals (marked with L and U) to generate a feasible instance.

In total, we generated 6 different data sets with 50 different problem instances each. Three first data sets of the problem instances were generated to evaluate the performance of the basic model and the decomposition scheme depending on the number of resources. Each data set includes 50 problem instances having the same size of sets T , J , R , P . All other parameters were randomly generated taking into account the problem properties (e.g. $p_{min,jr}$ was always less than $p_{max,jr}$, etc.). The parameters of these data sets are presented in Table 4.2 where $|P|$ reports the number of direct precedence relations in graph P .

Table 4.2: Impact of the resources: parameters of datasets

Data set	$ T $	d	$ J $	$ R $	$ P $
inst_j40_r1	40	2	40	1	40
inst_j40_r5	40	2	40	5	40
inst_j40_r10	40	2	40	10	40

Secondly, we generated three datasets of larger instances with 50 jobs and 10 resources. We considered different ranges of L_{rt} , since the correspondence between available resources L_{rt} and the demand represented by W_{jr} and limits ($p_{min,jr}$ and $p_{max,jr}$) has an important impact on the solution process. The parameters of these data sets are presented in Table 4.3. Each dataset consists of 50 instances.

Table 4.3: Parameters for datasets of large instances

Data set	$ T $	d	$ J $	$ R $	$ P $	L_{rt}
inst_j50_r10a	60	2	50	10	50	[0,80]
inst_j50_r10b	60	2	50	10	50	[0,100]
inst_j50_r10c	60	2	50	10	50	[0,120]

4.4.3 Impact of number of resource types on the method performances

According to our structure of the Benders decomposition scheme, the number of cuts generated in each iteration depends on the number of resources in the problem. Therefore, the first factor evaluated in the numerical experiment is the impact of the number of resources on the performance of the decomposition approach. The evaluation is made on the data sets presented in Table 4.2. We use the time box plots to represent the median, quartiles, maximal, and minimal computation time or gap values in the data sets.

Figure 4.5 shows that in the case of one resource the original model demonstrates a better performance than decomposition schemes with callbacks. Automatic Benders decomposition is slower than the other models. However, when the number of resources increases, we can observe different results. With 5 resources, Benders decomposition performs better than the basic model. In case of 10 resources, Callback schemes with Benders

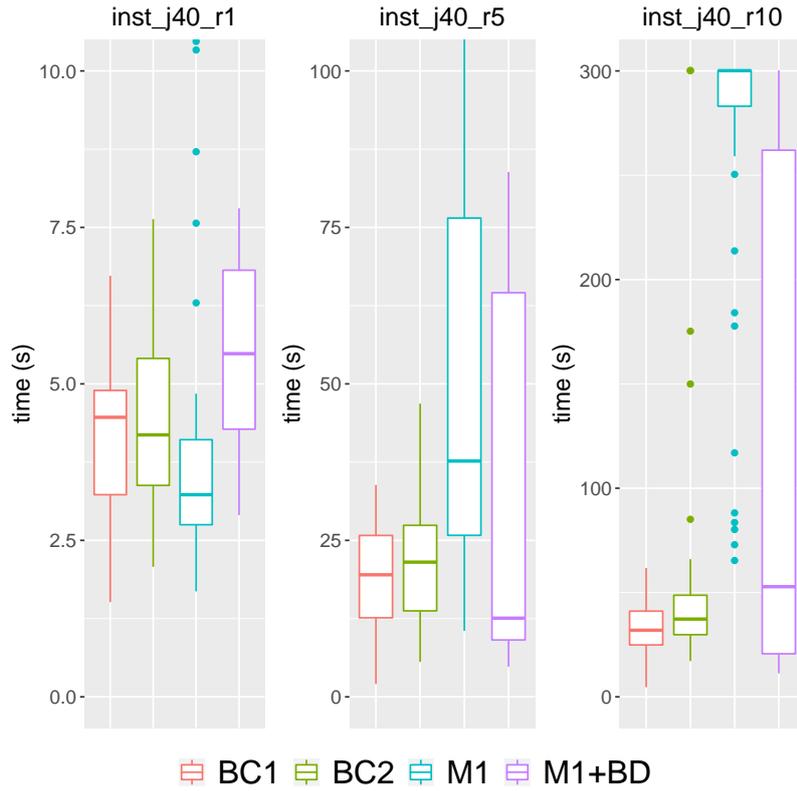


Figure 4.5: Time box plots for datasets with 40 jobs and different number of resources (1, 5 and 10)

decomposition outperform the basic model and built-in CPLEX procedure. The decomposition algorithm provides a significant advantage in the mean and minimal/maximal computation time as well as in the value distribution.

4.4.4 Evaluation of the impact of the available resource limit

To evaluate the impact of the available resource limit on the performances of solution methods, we use datasets from Table 4.3. For each instance, computation time was limited to 10 minutes.

We present the results of calculations for each dataset in the form of two boxplots, with the values of the time and relative gap.

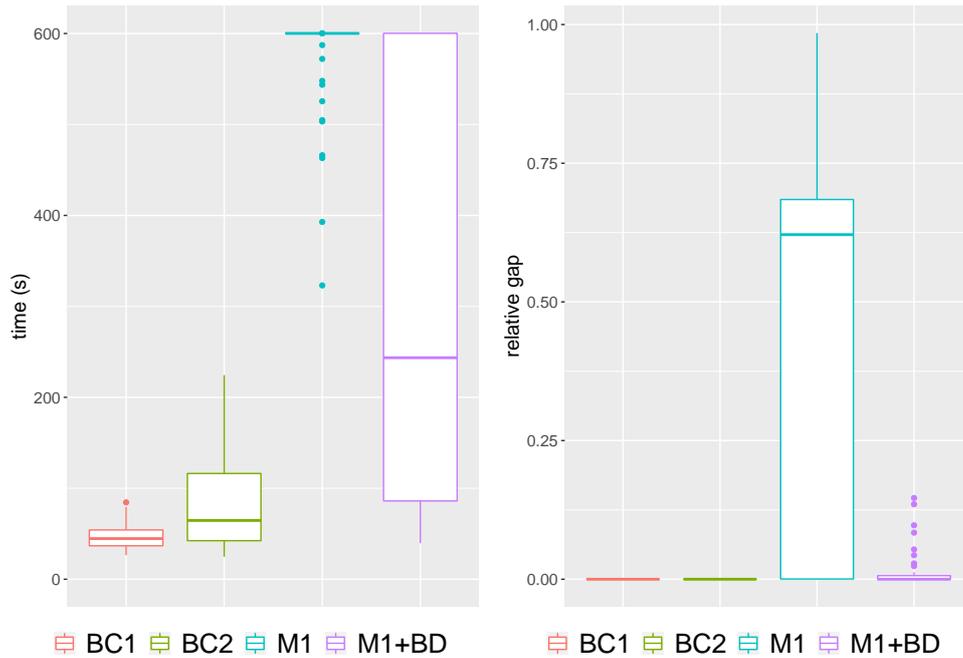


Figure 4.6: Time and relative gap box plots for the data set *inst_j50_r10a*

In Figure 4.6, we see the results for dataset *inst_j50_r10*. Our Benders decomposition scheme with callbacks significantly outperforms the original model and built-in CPLEX Benders (both versions). The built-in CPLEX Benders procedure provided optimal solutions within a significantly larger time, and the majority of the instances have not been solved to optimality by the original MILP model within a 10-minute time limit.

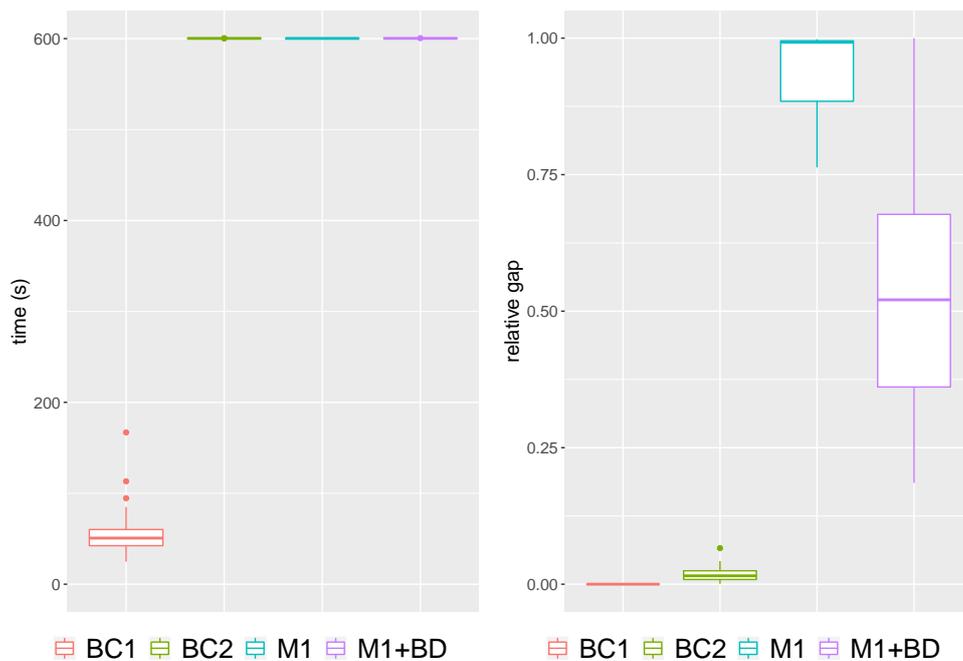


Figure 4.7: Time and relative gap box plots for the data set *inst_j50_r10b*

In Figure 4.7 we present the results for the dataset with the increased available resource limit. The first version of Benders decomposition scheme (BC1) with callbacks solves all instances to optimality in 60 seconds at average. All other tested approaches have not reached optimal solutions within a 10-minute time limit. In the Figure 4.8, we can notice the same results for the last dataset with a higher resource limit.

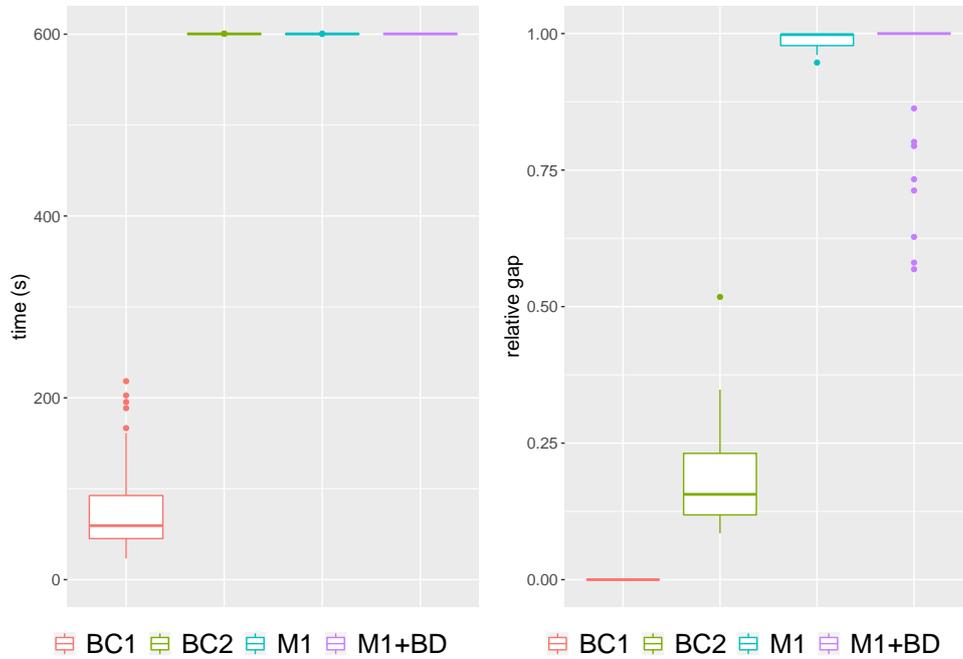


Figure 4.8: Time and relative gap box plots for the data set *inst_j50_r10c*

To sum up, this is a brief review of observations made on the base of numerical experiments:

- the decomposition with resource-disaggregated cuts works better (all versions BC1, BC2, M1+BD) with a growing number of resources (the same was proposed according to the decomposition scheme structure);
- on relatively large instances (50 jobs, 10 resources) Benders decomposition with callbacks (BC1) is significantly better in terms of performance on all datasets with various ranges of available resource values (compared by both time and gap).

4.5 Conclusions

In this chapter, we presented a Benders decomposition approach for a generalized version of a Resource Leveling Problem with an independent and flexible resource usage through the time periods. The Benders decomposition algorithm was enhanced with lower bound procedures and the generation of multiple cuts at each iteration. We also implemented

valid inequalities for the master problem. The decomposition algorithm was implemented with a single search tree (with CPLEX 12.10 callback functions) the cuts were generated for each feasible integer candidate with our algorithm instead of iterative master problem-solving.

The numerical experiments showed that the efficiency of our decomposition method in comparison to the basic model depends on the number of resources in the problem. Our method performs better for problem instances with a high number of resources. This is explained by the fact that the number of cuts generated at each iteration of the decomposition scheme depends on the number of resources. In the results obtained in the computational experiments, the decomposition algorithm provides a significant advantage in comparison to the basic model for instances with a larger number of resources. For the instances with 50 jobs and 10 resources, it provided optimal solutions, and other tested approaches didn't reach optimality within a significantly higher computational time limit. The next chapter discusses the variety of input data from a different perspective considering the issue of its uncertainty.

Chapter 5

Uncertainty and Metric approach

In this chapter, we present an approach defined as a metric estimation, which initially was proposed for scheduling problems by Lazarev [32]. If we consider a space of problem instances, we can represent any fluctuation in input data as a transfer from one point to another within this space. We study how such a transfer can be used for a Resource Leveling Problem to estimate the feasibility and optimality of solutions found for one point applied to another point.

The contribution of this chapter was also presented in the following work:

- *Tarasov, I.; Haït, A.; Battaïa, O. Metric Estimations for a Resource Levelling Problem With Variable Job Duration. 17th International Workshop on Project Management and Scheduling (PMS) 2020/2021 (postponed), submission accepted.*

5.1 Notations, definitions and goals

The following definitions and notations will be used in this chapter. In Tab. 5.1 we recall the decision variables of our MILP model for generalized RLP.

Decision variables	
S_{jt}	binary, if j starts at period t , then $\forall t_1 < t S_{jt_1} = 0, \forall t_2 \geq t S_{jt_2} = 1$;
E_{jt}	binary, if j ends at period t , then $\forall t_1 \leq t E_{jt} = 0, \forall t_2 > t E_{jt} = 1$;
d_{jt}	duration of job $j \in J$ in period $t \in T, d_{jt} \in [0, d]$
c_{jrt}	work volume of job $j \in J$ on resource $r \in R$ in period $t \in T$
o_{rt}	extra cost of resource $r \in R$ in period $t \in T, o_{rt} \geq 0$

Table 5.1: Model decision variables

Definition 3. By solution σ we define the set of values of the decision variables:

$$\sigma = \{d_{jt}, c_{jrt}; t \in T, r \in R, j \in J\} \quad (5.1)$$

In the solution, we do not consider binary variables S_{jt} , E_{jt} , overload cost o_{rt} , because the solution is fully determined by job duration d_{jt} , and contribution c_{jrt} per period, other variables are used to form linear constraints in the model.

We will also use the partial solution with the scheduling variables.

Definition 4. By schedule π we define the set of decision variable values

$$\pi = \{d_{jt}; t \in T, j \in J\} \quad (5.2)$$

The necessary conditions to be satisfied to realise a substitution of a schedule (or a full solution) of one problem instance to another instance are defined as follows:

Definition 5. The schedule π is applicable to instance I , if:

- instance I is solvable;
- schedule π is feasible, i.e. $\sum_T d_{jt} \in [d_{min,j}, d_{max,j}]$, $j \in J$;
- the schedule satisfies all the precedence relations, i.e. if $d_{j_2t} \geq 0$, then $\sum_{k=t}^{|T|} d_{j_1k} = 0$ for all $(j_1, j_2) \in P$, $t \in T$.

Definition 6. The solution σ is applicable to instance I , if:

- the scheduling part of σ is applicable to I ;
- the following inequalities are correct

$$d_{jt}p_{min,jr} \leq c_{jrt} \leq d_{jt}p_{max,jr}, j \in J, r \in R, t \in T; \quad (5.3)$$

$$\sum_{t \in T} c_{jrt} = W_{jr}, j \in J, r \in R. \quad (5.4)$$

We will also make a following definition to use it in next sections.

Definition 7. Define the LP part of the considered RLP model, with

- a set of decision variables c_{jrt} , o_{rt} ;
- all the constraints related to these variables, see equations (3.20), (3.19), (3.21);
- an original objective function (3.22);

as a subproblem $PS(\bar{d}_{jt})$.

We note that this LP defines the resource allocation part based on a fixed scheduling part with given job times \bar{d}_{jt} . The following notations will be further used:

- for a schedule π or a solution σ a superscript index A means that this solution/schedule is optimal for instance A ;
- for all estimations, a superscript index σ or π means an estimation for a solution or a schedule (for example, $\rho_e^\sigma(A, B)$ or $\rho_e^\pi(A, B)$);
- $\sigma^A(\pi)$ is an optimal resource allocation provided by the LP subproblem on the basis of schedule π for instance A .
- $V^A(\sigma)$ is the value of the objective function for solution σ and instance A , and $V^A(\sigma^A(\pi))$ is the value of the objective function for solution σ produced on the basis of schedule π .

Firstly, we study the difference in solution quality for two similar instances of the same size containing some differences in parameters. We demonstrate that for the considered model with aggregated objective function this difference is bounded and we can provide estimations of the bounds with a linear dependence on the difference in values of the concerned parameters.

5.2 Fluctuations in parameters

In this section, we define the fluctuations in the parameters of problem instances that will be considered in our study.

First of all, we do not consider the fluctuations in precedence constraints.

Secondly, we consider that the scale of the problem does not change, it means that neither number of jobs nor the number of types of resources changes.

The parameters that will be considered as possibly fluctuating are listed here below:

- e_r – extra cost of resource $r \in R$;
- L_{rt} – availability of resource $r \in R$ in period $t \in T$;
- $p_{min,jr}$ – job $j \in J$ minimal requirement per period in resource $r \in R$;
- $p_{max,jr}$ – job $j \in J$ maximal requirement per period in resource $r \in R$;
- W_{jr} – job $j \in J$ work volume with resource $r \in R$.

We consider that fluctuations in these parameters are independent.

5.2.1 Extra resource cost changes

Here below, we evaluate the impact of fluctuations in extra resource cost.

Lemma 2. *Consider instances A and B , which differ only by parameters e_r . If we apply the same solution σ to the both instances, the upper bound for objective function values difference can be evaluated as follows:*

$$|V^A(\sigma) - V^B(\sigma)| \leq \rho_e^\sigma(A, B), \quad (5.5)$$

where $\rho_e^\sigma(A, B)$ is represented in the following form

$$\rho_e^\sigma(A, B) = \max\left\{\sum_{r \in R} [e_r^A - e_r^B]^- \sum_{j \in J} W_{jr}, \sum_{r \in R} [e_r^A - e_r^B]^+ \sum_{j \in J} W_{jr}\right\}. \quad (5.6)$$

We use $[..]^-$ and $[..]^+$ to define $[x]^- = \min\{x, 0\}$, and $[x]^+ = \max\{x, 0\}$.

Proof.

$$|V^A(\sigma) - V^B(\sigma)| = \left| \sum_{r \in R} \sum_{t \in T} e_r^A o_{rt}^A - \sum_{r \in R} \sum_{t \in T} e_r^B o_{rt}^B \right|,$$

here $o_{rt} = \max\{0, \sum_{j \in J} c_{jrt} - L_{rt}\}$,

$$|V^A(\sigma) - V^B(\sigma)| = \sum_{r \in R} |e_r^A - e_r^B| \sum_{t \in T} \max\{0, \sum_{j \in J} c_{jrt} - L_{rt}\}.$$

The right side is still solution-dependent, since for each resource type, the extra cost difference is multiplied by actual overload volume in solution σ . We can form a solution-independent estimation with an upper bound for each $r \in R$

$$\sum_{t \in T} \max\{0, \sum_{j \in J} c_{jrt} - L_{rt}\} \leq \sum_{j \in J} W_{jr}.$$

Therefore, we have a solution-independent aggregated upper bound:

$$|V^A(\sigma) - V^B(\sigma)| \leq \sum_{r \in R} |e_r^A - e_r^B| \sum_{j \in J} W_{jr}. \quad (5.7)$$

Moreover, we note that with this objective function form the aggregated positive and negative values of difference $e_r^A - e_r^B$ compensate each other. If there exists two resource types $r_1, r_2 \in R$, and $e_{r_1}^A < e_{r_1}^B$, $e_{r_2}^A > e_{r_2}^B$, then the total objective function values difference will be in a range

$$[(e_{r_2}^B - e_{r_2}^A) \sum_{j \in J} W_{jr_2}, (e_{r_1}^B - e_{r_1}^A) \sum_{j \in J} W_{jr_1}].$$

In general, with an arbitrary set R , we can estimate $\rho_e^\sigma(A, B)$ as

$$\rho_e^\sigma(A, B) = \max\left\{\sum_{r \in R} \min\{e_r^A - e_r^B, 0\} \sum_{j \in J} W_{jr}, \sum_{r \in R} \max\{e_r^A - e_r^B, 0\} \sum_{j \in J} W_{jr}\right\}.$$

□

This estimation is applicable to any solution. Let's consider an optimal solution for instance A .

Lemma 3. *Consider instances A and B which differ only by parameters e_r . If we apply the optimal solution of instance A (σ^A) to instance B , then the upper bound on the difference between the values of the objective function can be evaluated as follows:*

$$V^B(\sigma^A) - V^B(\sigma^B) \leq \Delta_e^\sigma(A, B) = \sum_{r \in R} |e_r^A - e_r^B| \sum_{j \in J} W_{jr}. \quad (5.8)$$

Proof. In this case, we need to compare 4 values: $V^A(\sigma^A)$; $V^B(\sigma^B)$; $V^A(\sigma^B)$; $V^B(\sigma^A)$.

We note that $V^B(\sigma^A) \geq V^B(\sigma^B)$ and $V^A(\sigma^B) \geq V^A(\sigma^A)$ for any pair of instances A and B , so there exist six possible cases to be considered:

1. $V^A(\sigma^A) \leq V^A(\sigma^B) \leq V^B(\sigma^B) \leq V^B(\sigma^A)$;
2. $V^B(\sigma^B) \leq V^B(\sigma^A) \leq V^A(\sigma^A) \leq V^A(\sigma^B)$;
3. $V^A(\sigma^A) \leq V^B(\sigma^B) \leq V^A(\sigma^B) \leq V^B(\sigma^A)$;
4. $V^B(\sigma^B) \leq V^A(\sigma^A) \leq V^B(\sigma^A) \leq V^A(\sigma^B)$;
5. $V^B(\sigma^B) \leq V^A(\sigma^A) \leq V^A(\sigma^B) \leq V^B(\sigma^A)$;
6. $V^A(\sigma^A) \leq V^B(\sigma^B) \leq V^B(\sigma^A) \leq V^A(\sigma^B)$;

For cases 1-4, we can use Lemma 2 to prove that considered difference is less than the right side of inequality (5.7).

We prove the same for case 5 in the following way. We use the same approach as in Lemma 2. Firstly, we show that:

$$V^A(\sigma^A) - V^B(\sigma^B) \leq \sum_{r \in R} [e_r^A - e_r^B]^+ \sum_{j \in J} W_{jr}.$$

Instance B can provide a better solution in comparison to instance A only by reducing the total cost, i.e.

$$V^B(\sigma^A) - V^A(\sigma^A) \leq \sum_{r \in R} [e_r^A - e_r^B]^- \sum_{j \in J} W_{jr}.$$

Solution σ^A applied to instance B may provide a worse objective function value, with the difference up to the total reduction of resource amount. These two components are bounded and form the initial difference $V^B(\sigma^A) - V^B(\sigma^B)$. Therefore, in total the upper bound is the same as in inequality (5.8). The same logic can be applied for case 6. □

Although e_r is bounded and the estimation from lemma 3 forms a correct metric function, there is a nonlinear relation between changes in any other parameter and e_r . Moreover, there is another parameter W_{jr} used in the estimation.

However, since in the objective function, an extra cost is a coefficient of the impact made by some given resource, in any instance for any $r \in R$ we can keep a same schedule if we multiply e_r by an arbitrary coefficient k and all resource-related parameters (W_{jr} , L_{rt} , $p_{min,jr}$, $p_{max,jr}$) by $\frac{1}{k}$.

Suppose there are two instances A and B , and there is a difference $e_{r_1}^A \neq e_{r_1}^B$ for some resource type r_1 . We can modify instance B into \bar{B} by multiplying $e_{r_1}^B$ by $k = \frac{e_{r_1}^A}{e_{r_1}^B}$, and W_{jr} , L_{rt} , $p_{min,jr}$, $p_{max,jr}$ by $\frac{1}{k}$. There will be the same optimal schedule (or schedules) for instances B and \bar{B} . A solution (resource allocation) will be the same for all resources except resource r_1 , that will differ at the same proportion. Therefore, we can always consider a pair of instances with equal extra resource usage cost.

5.2.2 Fluctuations in available resource levels

This parameter type defines entire resource supply, however, in contrast to RCPSP, for RLP, it has no impact on the solution feasibility,

Lemma 4. *Consider instances A and B which differ by parameters L_{rt} . If we apply the same solution σ to the both instances, the upper bound for objective function values difference can be evaluated as follows:*

$$|V^A(\sigma) - V^B(\sigma)| \leq \rho_L^\sigma(A, B), \quad (5.9)$$

where $\rho_L^\sigma(A, B)$ is a particular metric upper estimation,

$$\rho_L^\sigma(A, B) = \max\left\{\sum_{r \in R} e_r \sum_{t \in T} [L_{rt}^B - L_{rt}^A]^+, \sum_{r \in R} e_r \sum_{t \in T} [L_{rt}^B - L_{rt}^A]^-\right\}. \quad (5.10)$$

Proof. $|V^A(\sigma) - V^B(\sigma)| = \left| \sum_{r \in R} \sum_{t \in T} e_r^A o_{rt}^A - \sum_{r \in R} \sum_{t \in T} e_r^B o_{rt}^B \right|, \quad (5.11)$

here $o_{rt} = \max\{0, \sum_{j \in J} c_{jrt} - L_{rt}\}$, and taking into account that costs are equal $e_r^A = e_r^B = e_r$ and $|\max\{a, b\} - \max\{c, d\}| \leq \max\{|a - c|, |b - d|\}$,

$$|V^A(\sigma) - V^B(\sigma)| \leq \sum_{r \in R} \sum_{t \in T} |e_r^A (\sum_{j \in J} c_{jrt} - L_{rt}^A) - e_r^B (\sum_{j \in J} c_{jrt} - L_{rt}^B)|. \quad (5.12)$$

So, with for identical solutions, we obtain the following result:

$$|V^A(\sigma) - V^B(\sigma)| \leq \sum_{r \in R} e_r \sum_{t \in T} |L_{rt}^A - L_{rt}^B|. \quad (5.13)$$

As in Lemma 2, we propose a precise upper estimation $\rho_L(A, B)$ with the following explanation. In instance B , several resources are available differently from instance A , each

difference $\Delta L_{rt} \neq 0$ leads to a limited possible impact on the value of the objective function. Suppose that the first difference is positive, i.e. $\Delta L_1 > 0$. Then the objective function difference lies within $[0, \Delta V_1]$, where ΔV_1 is ΔL_1 multiplied by corresponding extra resource usage cost e_r . If there is another $\Delta L_2 > 0$, then the impact on the objective is $[0, \Delta V_1 + \Delta V_2]$. If $\Delta L_3 < 0$, then the range left bound is shifted: $[\Delta V_3, \Delta V_1 + \Delta V_2]$.

If we take into account all k differences, aggregated range for the objective function value variation is

$$\left[\sum_{i=1}^k \min\{\Delta V_i, 0\}, \sum_{j=1}^k \max\{\Delta V_j, 0\} \right],$$

i.e. here bounds are formed by the sum of all negative and positive changes. More precisely, $\Delta V_x = e_r \Delta L_x$ if $\Delta L_x = L_{rt}^B - L_{rt}^A$ (it was applied at period t for resource r). Then we can represent the range of possible differences in the values of the objective function in the following way

$$\left[\sum_{r \in R} e_r \sum_{t \in T} \min\{(L_{rt}^B - L_{rt}^A), 0\}, \sum_{r \in R} e_r \sum_{t \in T} \max\{(L_{rt}^B - L_{rt}^A), 0\} \right].$$

We can compare absolute values of these bounds to estimate the absolute value of the difference:

$$\rho_L^\sigma(A, B) = \max\left\{ \sum_{r \in R} e_r \sum_{t \in T} [L_{rt}^B - L_{rt}^A]^{-}, \sum_{r \in R} e_r \sum_{t \in T} [L_{rt}^B - L_{rt}^A]^{+} \right\}.$$

□

The estimation (5.10) can be up to 2 times less than the basic estimation (5.13). It corresponds to the case when

$$\left| \sum_{i=1}^k \min\{\Delta V_i, 0\} \right| = \left| \sum_{j=1}^k \max\{\Delta V_j, 0\} \right|.$$

We also point out that this estimation does not depend on *solution* σ . If this solution is applicable to instance I and the resource availability levels are equal, the objective function keeps the same value. As it has been presented in the definition, it means that the required work volume W_{jr} parameters are also equal

$$\sum_{t \in T} c_{jrt} = W_{jr}, \quad j \in J, \quad r \in R;$$

and c_{jrt} satisfies the new period contribution limits with $p_{min,jr}$ and $p_{max,jr}$:

$$d_{jt} p_{min,jr} \leq c_{jrt} \leq d_{jt} p_{max,jr}, \quad j \in J, \quad r \in R, \quad t \in T.$$

Lemma 5. Consider instances A and B which differ by parameters L_{rt} . If we apply optimal solution of instance A σ^A to instance B , then the upper bound for the difference in the value of the objective function can be evaluated as follows:

$$V^B(\sigma^A) - V^B(\sigma^B) \leq \Delta_L^\sigma(A, B) = \sum_{r \in R} e_r \sum_{t \in T} |L_{rt}^A - L_{rt}^B| \quad (5.14)$$

Proof. We rely on the possible orderings, already considered in the proof of Lemma 3. These options are general and do not depend on the type of parameters that varies.

For cases 1-4, we can again use Lemma 4 to prove that the considered difference is less than the right side of inequality (5.13).

A special case 5 is considered in the same way. Instance B can provide a better solution only with additional amount of resources:

$$V^A(\sigma^A) - V^B(\sigma^B) \leq \sum_{r \in R} e_r \sum_{t \in T} [L_{rt}^B - L_{rt}^A]^+$$

For instance B , the same solution σ^A may provide a worse value of the objective function, in this case, the difference can be estimated as follows:

$$V^B(\sigma^A) - V^A(\sigma^A) \leq \sum_{r \in R} e_r \sum_{t \in T} [L_{rt}^B - L_{rt}^A]^-$$

These two components form the initial difference in the inequality (5.13). □

5.2.3 Estimations for the job-related parameters

In this subsection, we consider the case where two instances have some differences in the values of parameters $p_{min,jr}$, $p_{max,jr}$, or W_{jr} . In contrast to the overload cost and available resource level, in this case it is necessary to check if a particular solution (or a schedule) is applicable to a particular instance A or B (see Definitions 5 and 6).

Upper and lower assigned amount

For the assignment limits, we note that the differences in these parameters do not have any impact on the value of the objective function.

Lemma 6. Consider instances A and B which differ by parameters $p_{min,jr}$ or (and) $p_{max,jr}$. If a solution σ is applicable to both instances, there is no change for the objective function value, i.e.

$$V^A(\sigma) = V^B(\sigma), \quad \rho_{p_{min}/max}(A, B) = 0.$$

Proof. These parameters are limiting the amount of workload and resource $r \in R$ utilized by job $j \in J$, but do not modify directly the value of the objective function. As it was mentioned above,

$$|V^A(\sigma) - V^B(\sigma)| = \left| \sum_{r \in R} \sum_{t \in T} o_{rt}^A - \sum_{r \in R} \sum_{t \in T} o_{rt}^B \right|,$$

and $o_{rt} = \max\{0, e_r(\sum_{j \in J} c_{jrt} - L_{rt})\}$. If σ is applicable to A and B , then

$$d_{jt}^\sigma p_{min,jr}^A \leq c_{jrt}^\sigma \leq d_{jt}^\sigma p_{max,jr}^A, \quad j \in J, r \in R, t \in T;$$

$$d_{jt}^\sigma p_{min,jr}^B \leq c_{jrt}^\sigma \leq d_{jt}^\sigma p_{max,jr}^B, \quad j \in J, r \in R, t \in T;$$

the values of c_{jrt}^σ will not be changed, as any other part of $|V^A(\sigma) - V^B(\sigma)|$.

□

So, it is possible to formulate the following estimation for the optimal solutions.

Lemma 7. *Consider instances A and B which differ by parameters $p_{max,jr}$ and/or $p_{min,jr}$. Suppose that an optimal solution σ^A of instance A is applicable to instance B and an optimal solution σ^B of instance B is applicable to instance A . If we apply an optimal solution of instance A i.e. σ^A as a solution to instance B , then we obtain the same value of the objective function.*

$$\Delta_{a,p_{min}/max}^\sigma(A, B) = V^B(\sigma^A) - V^B(\sigma^B) = 0 \quad (5.15)$$

Proof. With the condition that both solutions are applicable to both instances, we can directly use Lemma 1, as we can estimate all the components $(\rho_{p,max}(A, B, \sigma^A), \rho_{p,max}(A, B, \sigma^B))$, as well as the same values for $p_{min,jr}$.

If both solutions are applicable to both instances, it means that it is not necessary to modify solution σ^A if it is applied to instance B to reach the optimal value of the objective function, and the same for σ^B applied to instance A .

We can also show that the difference can be more than zero if either σ^A is not applicable to B , or σ^B is not applicable to A , as a consequence, it is impossible to use Lemmas 1 and 6. For $p_{max,jr}$, if $\Delta^{p_{max}}(A, B) > 0$, it means that solution σ^A applied to instance B must be modified to achieve an optimal solution. If $p_{max,jr}^A < p_{max,jr}^B$, then in some period t we are allocating $c_{jrt}^B > p_{max,jr}^A d_{jt}^B$ of resource r for a job j , so a resulting σ^B is not applicable to instance A . If there is a difference $p_{max,jr}^A > p_{max,jr}^B$, then it means that in some period t we have to reduce an allocation of resource r for a job j , as $c_{jrt}^A > p_{max,jr}^B d_{jt}^B$, so σ^A is not applicable to instance B . It is possible to formulate a similar statement for parameters $p_{min,jr}$.

□

Lemma 8. Consider instances A and B which differ only by parameters $p_{max,jr}$ (or $p_{min,jr}$). If a schedule π is applicable to both instances, there is an upper bound for the difference in the values of the objective function can be estimated as follows:

$$|V^A(\sigma^A(\pi)) - V^B(\sigma^B(\pi))| \leq \rho_{p_{max}}^\pi(A, B) \quad (5.16)$$

where

$$\rho_{p_{max}}^\pi(A, B) = md \max\left\{\sum_{r \in R} e_r \sum_{j \in J} [p_{max,jr}^A - p_{max,jr}^B]^-, \sum_{r \in R} e_r \sum_{j \in J} [p_{max,jr}^A - p_{max,jr}^B]^+\right\} \quad (5.17)$$

Proof. As in proofs of Lemmas 2 or 4, we estimate the impact of one parameter's fluctuation and then aggregate all fluctuations.

If $p_{max,jr}^A < p_{max,jr}^B$ for some $r \in R$ and $j \in J$ in instances A and B , the difference in the values of their objective function will lie within the following range:

$$[e_r(p_{max,jr}^A - p_{max,jr}^B)md, 0],$$

If $p_{max,jr}^A > p_{max,jr}^B$, the range will be

$$[0, e_r(p_{max,jr}^A - p_{max,jr}^B)md],$$

Any arbitrary set of fluctuations will form the following range representing an estimation of $\rho_{p_{max}}^\pi(A, B)$:

$$\left[\sum_{r \in R} e_r md \sum_{j \in J} [p_{max,jr}^A - p_{max,jr}^B]^-, \sum_{r \in R} e_r md \sum_{j \in J} [p_{max,jr}^A - p_{max,jr}^B]^+\right]$$

□

Lemma 9. Consider instances A and B which differ by parameters $p_{max,jr}$. Suppose that an optimal solution σ^A of instance A is applicable to instance B and an optimal solution σ^B of instance B is not applicable to instance A . If we apply the optimal solution of instance A σ^A as a solution to instance B , then the objective function values difference is bounded above by the following expression

$$\Delta_{n,p_{max}}^\sigma(A, B) = V^B(\sigma^A) - V^B(\sigma^B) \leq \sum_{r \in R} e_r md \sum_{j \in J} |p_{max,jr}^A - p_{max,jr}^B| \quad (5.18)$$

Proof. We note that in our case any fluctuations in values $p_{max,jr}$ (and/or $p_{min,jr}$) do not impact the objective function, as it was shown in Lemma 6, i.e. $V^A(\sigma^A) = V^B(\sigma^A)$. We can use the same approach as in the proof of Lemma 5 to compare $V^A(\sigma^A)$ and $V^B(\sigma^B)$. Firstly, the absolute value of the difference has the form:

$$|V^A(\sigma^A) - V^B(\sigma^B)| \leq \sum_{r \in R} e_r \sum_{t \in T} \sum_{j \in J} |c_{jrt}^A - c_{jrt}^B|.$$

Secondly, taking into account the limits for c_{jrt} ,

$$p_{min,jr}^A d_{jt}^A \leq c_{jrt}^A \leq p_{max,jr}^A d_{jt}^A$$

$$p_{min,jr}^B d_{jt}^B \leq c_{jrt}^B \leq p_{max,jr}^B d_{jt}^B,$$

we can provide an upper estimation

$$|V^A(\sigma^A) - V^B(\sigma^B)| \leq \sum_{r \in R} e_r m \sum_{j \in J} |p_{max,jr}^A - p_{max,jr}^B| d,$$

as $d_{jt} \in [0, d]$ and there are m periods inside the planning horizon. □

The same expressions can be formed for the lower limit $p_{min,jr}$.

Work volume fluctuations

Consider the case where two instances A and B differ by values of W_{jr} . It is impossible to apply the same solution σ (with the same values of c_{jrt}) to both instances, so the idea of a metric estimation is not applicable in the same form. We will provide an estimation for the case when only the schedule is applied. With this schedule, an optimal resource allocation is constructed in a polynomial time (an optimal resource allocation with a given schedule is an LP). The schedule must be applicable to both instances (see Definition 5).

Lemma 10. *Consider instances A and B which differ only by W_{jr} . If we apply the same schedule π to the both instances, the upper bound for the difference in values of the objective function can be estimated as follows:*

$$|V^A(\sigma^A(\pi)) - V^B(\sigma^B(\pi))| \leq \rho_W^\pi(A, B), \quad (5.19)$$

where

$$\rho_W^\pi(A, B) = \max\left\{\sum_{r \in R} e_r \sum_{j \in J} [W_{jr}^A - W_{jr}^B]^+, \sum_{r \in R} e_r \sum_{j \in J} [W_{jr}^A - W_{jr}^B]^-\right\}. \quad (5.20)$$

Proof. We can refer to proof of Lemma 4. In this case, it is also possible to evaluate an upper bound for the difference in the values of the objective function and to consider it as an independent sum of estimations for fluctuations $W_{jr}^A \neq W_{jr}^B$.

Each fluctuation $W_{jr}^A > W_{jr}^B$ may lead to the difference in the values of the objective function within the following range $[W_{jr}^B - W_{jr}^A, 0]$. An upper bound for an aggregation of all these changes can be represented with the following range

$$\left[\sum_{r \in R} e_r \sum_{j \in J} \min\{(W_{jr}^B - W_{jr}^A), 0\}, 0\right].$$

The same approach can be applied for the case where $W_{jr}^A < W_{jr}^B$. The difference in the values of the objective function caused by all fluctuations of W_{jr} :

$$[0, \sum_{r \in R} e_r \sum_{j \in J} \max\{(W_{jr}^B - W_{jr}^A), 0\}].$$

If we regroup the two previous cases, we obtain the following range:

$$[\sum_{r \in R} e_r \sum_{j \in J} \min\{(W_{jr}^B - W_{jr}^A), 0\}, \sum_{r \in R} e_r \sum_{j \in J} \max\{(W_{jr}^B - W_{jr}^A), 0\}],$$

and the following estimation

$$\rho_W^\pi(A, B) = \max\{\sum_{r \in R} e_r \sum_{j \in J} \max\{(W_{jr}^A - W_{jr}^B), 0\}, \sum_{r \in R} e_r \sum_{j \in J} \max\{(W_{jr}^B - W_{jr}^A), 0\}\}$$

□

Secondly, we construct an estimation for an optimal schedule for instance A applied to instance B .

Lemma 11. *Consider instances A and B which differ only by parameters W_{jr} . If we apply the optimal schedule π^A of instance A to instance B , then the upper bound for the difference in the values of the objective function can be estimated as follows:*

$$V^B(\sigma^B(\pi^A)) - V^B(\sigma^B(\pi^B)) \leq \Delta_W^\pi(A, B) = \sum_{r \in R} e_r \sum_{j \in J} |W_{jr}^A - W_{jr}^B|. \quad (5.21)$$

Proof. As in the proof of Lemma 5, we consider possible orderings of the values of $V^A(\sigma^A(\pi^A)); V^B(\sigma^B(\pi^B)); V^A(\sigma^A(\pi^B)); V^B(\sigma^B(\pi^A))$.

For the cases 1-4 and 6 we use Lemma 10, concluding that the considered difference is less than the right side of inequality (5.20), that is less than (5.21).

We prove the same for case 5 in the following way. We use the same approach as in Lemma 10. Firstly, we show that the difference

$$V^A(\sigma^A(\pi^A)) - V^B(\sigma^B(\pi^B)) \leq \sum_{r \in R} e_r \sum_{j \in J} \max\{(W_{jr}^A - W_{jr}^B), 0\}$$

Instance B can provide a better solution only with additional amount of resource. Secondly, we provide the estimation for the difference

$$V^B(\sigma^B(\pi^A)) - V^A(\sigma^A(\pi^A)) \leq \sum_{r \in R} e_r \sum_{j \in J} \max\{(W_{jr}^B - W_{jr}^A), 0\}$$

In instance B the same schedule π^A may provide worse objective function value, with the difference up to the total reduction of the resource amount. These two components are

bounded and form the initial difference $V^B(\sigma^B(\pi^A)) - V^B(\sigma^B(\pi^B))$, therefore, the upper bound is the same as in inequality (5.20). □

5.3 A generalized metric estimation

In previous sections, we have described the upper estimations for all parameters except the number of periods, jobs, resources, and precedence graph. For each parameter, we have proved that there exists an upper bound for the difference in the values of the objective function that linearly depends on the difference in the values of this parameter for two problem instances.

As it was noticed, in our case of RLP it is necessary to take into account that the fluctuations of some parameters may lead to the infeasibility of a solution or the infeasibility of a schedule. In Table 5.2, we present the set of considered parameters and the impact of their fluctuations on the objective function and the feasibility of the solution and the schedule.

We can conclude that since here we have too strict requirements (for example, any change of W_{jr} leads to the infeasibility of a solution), it is reasonable to consider an estimation for the optimal schedule $\Delta^\pi(A, B)$. In addition, since a complete solution can be obtained from a schedule in a polynomial time, this option is the most acceptable for an arbitrary difference between a pair of problem instances.

Table 5.2: The summary of all estimations

parameter	possible to use?		metric
	σ	π	
e_r	any σ	any π	$\Delta_e^\pi(A, B)$
L_{rt}	any σ	any π	$\Delta_L^\pi(A, B)$
$p_{min,jr}$	applicable σ	applicable π	$\Delta_{p_{min}}^\pi(A, B)$
$p_{max,jr}$	applicable σ	applicable π	$\Delta_{p_{max}}^\pi(A, B)$
W_{jr}	impossible	applicable π	$\Delta_W^\pi(A, B)$
Overall	impossible	applicable π	$\Delta^\pi(A, B)$

Here below, Theorem 1 aggregates the estimations for two instances A and E , varying in any of the listed parameters.

Theorem 1. *Suppose there are two instances A and E that differ by parameters L_{rt} , W_{jr} , $p_{min,jr}$, and $p_{max,jr}$. If we apply a schedule π^A that is optimal for instance A to instance*

E , the upper bound for difference in the values of the objective function can be estimated as follows:

$$V^E(\sigma^E(\pi^A)) - V^E(\sigma^E) \leq \Delta^\pi(A, E); \quad (5.22)$$

and

$$\Delta^\pi(A, E) = \Delta_L^\pi(A, E) + \Delta_{p_{min}}^\pi(A, E) + \Delta_{p_{max}}^\pi(A, E) + \Delta_W^\pi(A, E). \quad (5.23)$$

Proof. It is possible to separate this function and proof this statement with a chain of additional instances:

- B , all parameters equal to instance A except L_{rt} , and $L_{rt}^B = L_{rt}^E$;
- C , all parameters equal to instance B except $p_{min,jr}$, $p_{min,jr}^C = p_{min,jr}^E$;
- D , all parameters equal to instance E except $p_{max,jr}$, $p_{max,jr}^D = p_{max,jr}^E$;

We note that instances D and E differ only in parameters W_{jr} .

As each expression includes a sum of absolute values, $\Delta^\pi(A, E)$ has an additive property:

$$\Delta^\pi(A, E) \leq \Delta^\pi(A, B) + \Delta^\pi(B, C) + \Delta^\pi(C, D) + \Delta^\pi(D, E).$$

We take into account that $\Delta^\pi(A, B) = \Delta_L^\pi(A, B)$, $\Delta^\pi(B, C) = \Delta_{p_{min}}^\pi(B, C)$, $\Delta^\pi(C, D) = \Delta_{p_{max}}^\pi(C, D)$, $\Delta^\pi(D, E) = \Delta_W^\pi(D, E)$, so

$$V^E(\sigma^E(\pi^A)) - V^E(\sigma^E) \leq \Delta_L^\pi(A, B) + \Delta_{p_{min}}^\pi(B, C) + \Delta_{p_{max}}^\pi(C, D) + \Delta_W^\pi(D, E),$$

and as the parameters of all instances B, C, D are either equal to parameters A or E ,

$$V^E(\sigma^E(\pi^A)) - V^E(\sigma^E) \leq \Delta_L^\pi(A, E) + \Delta_{p_{min}}^\pi(A, E) + \Delta_{p_{max}}^\pi(A, E) + \Delta_W^\pi(A, E).$$

□

We formulate linear inequalities guaranteeing the applicability of one schedule found for one instance to a different problem instance.

Lemma 12. *Suppose there are two instances A and B that differ in one of parameters $p_{min,jr}$, $p_{max,jr}$ or W_{jr} . Any schedule π , applicable to instance A , is also applicable to instance B , if B is solvable and:*

$$d_{min,j}^B \leq d_{min,j}^A; \quad d_{max,j}^A \leq d_{max,j}^B; \quad \forall j \in J, \quad (5.24)$$

or it can be reformulated in a linear form for the parameters of B :

$$W_{jr}^B \leq d_{min,j}^A p_{max,jr}^B; \quad d_{max,j}^A p_{min,jr}^B \leq W_{jr}^B; \quad \forall j \in J; \quad \forall r \in R. \quad (5.25)$$

Proof. From Definition 5, we see that a schedule π must guarantee that

$$\sum_{t \in T} d_{jt} \in [d_{min,j}^B, d_{max,j}^B]; \forall j \in J.$$

If it is applicable to A , then

$$\sum_{t \in T} d_{jt} \in [d_{min,j}^A, d_{max,j}^A]; \forall j \in J.$$

Basically, we can guarantee that π is applicable to B if range $[d_{min,j}^A, d_{max,j}^A]$ is fully included in $[d_{min,j}^B, d_{max,j}^B]$, so

$$d_{min,j}^B \leq d_{min,j}^A; d_{max,j}^A \leq d_{max,j}^B; \forall j \in J.$$

A linear condition for B is obtained from the definition of $d_{min,j}$ and $d_{max,j}$ (3.25). If we consider $d_{min,j}^A$ and $d_{max,j}^A$ given and fixed, we rewrite the conditions (5.24):

$$\max_{r \in R} \frac{W_{jr}^B}{p_{max,jr}^B} \leq d_{min,j}^A; d_{max,j}^A \leq \min_{r \in R} \frac{W_{jr}^B}{p_{min,jr}^B}; \forall j \in J,$$

and reformulate these conditions without a maximum:

$$\frac{W_{jr}^B}{p_{max,jr}^B} \leq d_{min,j}^A; d_{max,j}^A \leq \frac{W_{jr}^B}{p_{min,jr}^B}; \forall j \in J; \forall r \in R.$$

□

This lemma also allows us to study the robustness of the schedule in relation to these parameters.

We have proved that the basic approach with metric estimations is applicable to our version of RLP. The majority of relevant parameters that can fluctuate bring a linearly-dependent change of the solution quality. Still, for our case of RLP, the main issue consists of the description of solvable subsets of problem instances.

We can also improve the estimations presented here above for another situation: where we have instance A and a given (sub)-optimal baseline schedule π^A . Instance B represents a modified version of instance A affected by possible fluctuations.

For this situation, we can provide a non-metric estimation, depending on π^A , defined as $\Delta^\pi(A, B, \pi^A)$. Based on the schemes used in the previous lemmas, we provide a list of improved partial schedule-dependent estimations:

- $\Delta_e^\pi(A, B, \pi^A) = \sum_{r \in R} |e_r^A - e_r^B| \sum_{t \in T} o_{rt}^B(\pi^A) \leq \Delta_e^\pi(A, B)$
- $\Delta_L^\pi(A, B, \pi^A) = \Delta_L^\pi(A, B)$

- $\Delta_{p,min}^\pi(A, B, \pi^A) = \sum_{r \in R} e_r \sum_{t \in T} \sum_{j \in J} |p_{min,jr}^A - p_{min,jr}^B| d_{jt}^A \leq \Delta_{p,min}^\pi(A, B)$
- $\Delta_{p,max}^\pi(A, B, \pi^A) = \sum_{r \in R} e_r \sum_{t \in T} \sum_{j \in J} |p_{max,jr}^A - p_{max,jr}^B| d_{jt}^A \leq \Delta_{p,max}^\pi(A, B)$
- $\Delta_W^\pi(A, B, \pi^A) = \Delta_W^\pi(A, B)$

With a generalized expression (same approach as Theorem 1)

$$\begin{aligned} \Delta^\pi(A, B, \pi^A) &= \Delta_e^\pi(A, B, \pi^A) + \Delta_L^\pi(A, B, \pi^A) + \\ &+ \Delta_{p,min}^\pi(A, B, \pi^A) + \Delta_{p,max}^\pi(A, B, \pi^A) + \Delta_W^\pi(A, B, \pi^A). \end{aligned} \quad (5.26)$$

5.4 Feasibility and properties of the space of problem instances

In [31], there is a measure denoted as a stability radius for the feasibility and optimality of schedules in case of some shifts in parameters. We consider the range of possible changes providing feasibility and optimality in Table 5.3

Table 5.3: Impact of the fluctuations of parameters: allowed changes guaranteeing feasibility and optimality

parameter	changes guaranteeing feasibility	
	applying π	applying σ
e_r	∞	∞
L_{rt}	∞	∞
$p_{min,jr}$	Lemma 12	applicable σ
$p_{max,jr}$	Lemma 12	applicable σ
W_{jr}	Lemma 12	0
Overall	Lemma 12	0

Further, we consider some scaling properties of the space of problem instances. We can show that there are some transformations of instances, leading to the same set of feasible schedules and the same optimal schedule. In this case, a new optimal value is predictable, as well as a whole solution.

Lemma 13. *Suppose that instance B is produced from instance A by the following transformation: all job-resource-related parameters are multiplied by a coefficient $k > 0$. We will define it as $kA = B$ meaning that $kL_{rt}^A = L_{rt}^B$; $\forall r \in R$; $\forall t \in T$; and $kp_{min,jr}^A = p_{min,jr}^B$; $kp_{max,jr}^A = p_{max,jr}^B$; $kW_{jr}^A = W_{jr}^B$; $\forall j \in J$; $\forall r \in R$. In this case, both instances A and B have a same set of feasible schedules and a set of optimal schedules with scaled solution variables*

$$kc_{jrt}^A = c_{jrt}^B; \forall j \in J; \forall r \in R; ko_{rt}^A = o_{rt}^B; \forall r \in R; \forall t \in T. \quad (5.27)$$

Thus, objective function values are also scaled:

$$V^B(\sigma^B) = kV^A(\sigma^A). \quad (5.28)$$

Proof. Firstly, this transformation does not change any parameter involved in the definition of a schedule, applicable to an instance (see Def. 5). It does not change precedence relations nor values of minimal and maximal duration. These values equal to a ratio of required workload W_{jr} and a maximal or a minimal amount of allocated resource ($p_{max,jr}$ or $p_{min,jr}$), both multiplied by k . Thus, such a transformed instance is still solvable.

Secondly, we consider the solutions. If schedule π^A with variables d_{jt}^A is optimal for instance A , providing a solution $\sigma^A(\pi^A)$ with variables c_{jrt}^A , then this schedule is also applicable to instance B . It produces a scaled optimal solution $\sigma^B(\pi^A)$ with variables c_{jrt}^B . This solution is also optimal, as the solution variables c_{jrt} are defined on a base of a schedule (i.e. variables d_{jt}), that are connected by the constraints (3.19). We can represent these constraints with parameters of instance A :

$$\left. \begin{array}{l} kp_{min,jr}^A d_{jt}^A = p_{min,jr}^B d_{jt}^B \leq c_{jrt}^B \\ c_{jrt}^B \leq p_{max,jr}^B d_{jt}^B = kp_{max,jr}^A d_{jt}^A \end{array} \right\} \quad \forall j \in J, \forall r \in R, \forall t \in T;$$

and by W_{jr} with constraints (3.20)

$$\sum_{t \in T} c_{jrt}^B = W_{jr}^B = kW_{jr}^A, \quad \forall j \in J, \forall r \in R.$$

All these linear constraints are scaled for instance B , and it keeps the same ratio between all these parameters. Finally, objective function (3.22) involves variables: $o_{rt} \in [0, \infty)$

$$\text{Minimize } \sum_{r \in R} \sum_{t \in T} e_r o_{rt},$$

defined by constraints (3.21):

$$o_{rt} \geq \sum_{j \in J} c_{jrt} - L_{rt}, \quad \forall t \in T, \forall r \in R;$$

where both c_{jrt} and L_{rt} are multiplied by k in instance B :

$$\sum_{j \in J} c_{jrt}^B - L_{rt}^B = k \left(\sum_{j \in J} c_{jrt}^A - L_{rt}^A \right), \forall t \in T, \forall r \in R.$$

As we minimize o_{rt} , then there is no reason to change neither a structure of a schedule or a solution in the changed instance B . That is why a solution with variables c_{jrt}^B is optimal, as well as schedule π^A providing it with variables d_{jt}^A . Therefore, there exists solution σ^B with variables $c_{jrt}^B = kc_{jrt}^A$ based on the same schedule and it is optimal with the following objective value

$$V^B(\sigma^B) = kV^A(\sigma^A).$$

□

We can use this scaling feature to improve the previous estimation. There is an original instance A that was affected by some changes and transformed into instance B , with a given (sub)optimal schedule π^A (and a solution σ^A). An original estimation $\Delta(A, B)$ is constructed between two instances A and B for the case where we apply a given schedule π^A to instance B . We can use a two stage-approach:

1. construct a scaled instance kA ($k > 0$) that will have a better estimation $\Delta(kA, B) \leq \Delta(A, B)$;
2. apply $\pi^A = \pi^{kA}$ to B as there are equal optimal schedules in instances A and kA , estimation $\Delta(kA, B)$ can be considered as the estimation in this case.

We illustrate this scheme in Figure 5.1. We consider two points A and B of Ω , space of problem instances. A metric estimation $\Delta(A, B)$ can be improved, if we can find an instance kA that is closer to B and can keep the same solution. Here we mark two squares representing a set of instances with a fixed l_1 metric, equal to a metric between A and B , and a metric between kA and B . We define a l_1 norm $\|\cdot\|_1$ on space Ω based on function $\Delta(A, B)$:

$$\|A\| = \sum_{r \in R} e_r^A \left(\sum_{t \in T} |L_{rt}^A| + md \sum_{j \in J} (|p_{min,jr}^A| + |p_{max,jr}^A|) + \sum_{j \in J} |W_{jr}^A| \right).$$

Norm axioms are verified for this formula:

$$\|A\| = 0 \iff A = 0;$$

$$\|kA\| = |k| \cdot \|A\|;$$

$$\|A + B\| \leq \|A\| + \|B\|;$$

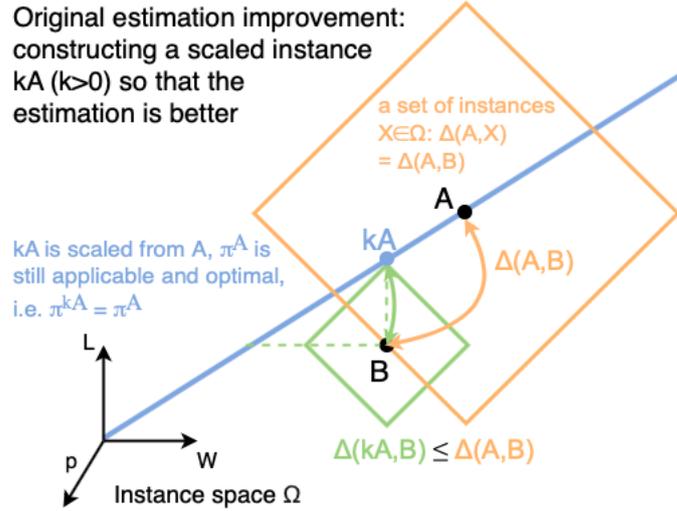


Figure 5.1: Representation of scaling case.

and

$$\Delta(A, B) = \|A - B\|.$$

5.5 Experiments

5.5.1 Datasets

We perform several tests to evaluate our approach. First, we evaluate the frequency of situations where an original schedule π^A is still applicable to instance B and remains optimal. Secondly, we determine the accuracy of our estimations in the case when we partially reconstruct the solution for a new instance with the LP subproblem.

For each instance A we find an optimal solution (or a suboptimal one obtained with a time limit) and construct an instance B on the basis of A with some changes. We will vary the absolute value and the number of changes in parameters W_{jr} and L_{rt} . These parameters appear to be the most interesting as they are frequently affected by unexpected changes in practice. After that, we solve instance B to find a real optimal solution whose value is used as a baseline measure for comparison.

We use a MILP model that was described in Chapter 3. This MILP model is used to solve the instances and the LP subproblem (see Definition 7) is used to obtain an optimal solution $\sigma^B(\pi^A)$ based on given schedule π^A . The tests are realized with a solver CPLEX 12.10.

We study the relative accuracy $G(A, B)$ of our estimation defined as follows:

$$G(A, B) = \frac{V^B(\sigma^B(\pi^A)) - V^B(\sigma^B(\pi^B))}{\Delta^\pi(A, B)}. \quad (5.29)$$

We note that $G(A, B) = 0$ when $V^B(\sigma^B(\pi^A)) = V^B(\sigma^B(\pi^B))$, meaning that schedule π^A remains optimal and provides an optimal solution for a new instance in a polynomial time.

Table 5.4: Parameters of datasets

Group	$ T $	d	$ J $	$ R $	$ P $	SP
inst_j10_r3	20		10	3	5–15	
inst_j15_r3	25	2	15	3	5–20	0.2–0.8
inst_j20_r5	30		20	5	5–40	

These estimations are evaluated for different fluctuations in terms of modified parameters and the magnitude of fluctuations. We vary the scale and the structure of the instances according to the list presented in Table 5.4. There are three groups, having a different number of resources and jobs, with a varying precedence graph size and structure. We also define a set of given magnitude $|\delta|$ of possible input data changes and a number of these changes. For each group and each pair magnitude-quantity, we generate 200 random instances.

5.5.2 Results

We start with dataset inst_j10_r3. The results are presented in Table 5.5. Firstly, we vary workload W_{jr} . With a fixed magnitude of changes $|\delta| = 5.0$ (equal to 10% of W^U), we increase the number of changes from 1 to 30, that is the maximal value, as $N_{max} = |J| \cdot |R| = 30$. The accuracy is predictably decreasing with the growth of changes. A real value of the difference in the objective function is between 0.18 and 0.3 of our estimation.

We note that the schedule is quite stable in terms of optimality for one change of higher magnitude. For about 30-40% of instances, it is possible to reconstruct the solution for a modified instance with an existing schedule and keep it optimal.

With variations in the given available resource amount L_{rt} values, we can make the same conclusion. A lot of variations of low magnitude gradually reduce the percentage of schedules remaining optimal, with a relative accuracy decreasing from 0.3 to 0.1.

We note that we obtained a surprisingly high percentage of saved schedules (theoretically, it should decrease with a growing magnitude) on the highest magnitude of changes for both parameters. It can be explained by the impact of the restriction that

Table 5.5: Results for dataset group inst_j10_r3.

Variations	N changes	$ \delta $	% cases $\pi^A = \pi^B$ (remains optimal)	other cases: G(A,B)			
				min	mean	max	
W_{jr} , with $N_{max} = 30$	1	5.0	40	0.0001	0.3267	0.7826	
	5		15	0.0009	0.2713	0.7679	
	10		3	0.0011	0.2239	0.6551	
	20		0	0.0466	0.1971	0.4700	
	30		0	0.0351	0.1817	0.4592	
		10.0	41	0.0222	0.4505	1.0	
	1	15.0	28	0.0013	0.3324	0.9310	
		20.0	39	0.0046	0.3156	0.9756	
	L_{rt} , with $N_{max} = 60$	1	5.0	60	0.0047	0.3132	0.9636
		5		17	0.0012	0.1727	0.5015
10		3		0.0049	0.1554	0.3975	
20		0		0.0110	0.1312	0.3099	
30		1		0.0017	0.1257	0.2933	
40		0	0.0169	0.1096	0.2783		
		10.0	45	0.0015	0.2871	0.9297	
1		20.0	35	0.0009	0.2059	0.6129	
		30.0	41	0.0001	0.1539	0.6507	
		40.0	49	0.0028	0.1822	0.6147	

reduced the changes to avoid negative values of W_{jr}^B and L_{rt}^B : $W_{jr}^B = \max\{W_{jr}^A \pm \delta, 0\}$, $L_{rt}^B = \max\{L_{rt}^A \pm \delta, 0\}$.

We increase the number of jobs in the second dataset inst_j15_r3. The results are described in Table 5.6 with the same structure of presentation. For this dataset, we also note that the schedule is more stable in the case of a single change of higher magnitude than a set of many relatively smaller changes (even taking into account the limitations of magnitude to avoid negative parameters).

The results for the third dataset inst_j20_r5, are presented in Table 5.7. For this last dataset, we note that with an increased scale of instances the mean relative accuracy was increased, especially for the high-magnitude single shift.

Table 5.6: Results for dataset inst_j15_r3.

Variations	N changes	$ \delta $	% cases $\pi^A = \pi^B$ (remains optimal)	other cases: G(A,B)				
				min	mean	max		
W_{jr} , with $N_{max} = 30$	1	5.0	40	0.0001	0.3267	0.7826		
	2		34	0.0019	0.2345	0.9629		
	3		14	0.0001	0.2803	0.7609		
	4		14	0.0003	0.2506	0.9294		
	5		15	0.0018	0.2713	0.7679		
	6		4	0.0016	0.2319	0.7610		
	7		1	0.0009	0.2023	0.5777		
	8		4	0.0115	0.2087	0.6459		
	9		5	0.0057	0.2083	0.6782		
	10		3	0.0011	0.2239	0.6551		
	20		0	0.0466	0.1971	0.4700		
	30		0	0.0351	0.1817	0.4592		
				10.0	41	0.0222	0.4505	1.0
			1	15.0	28	0.0013	0.3324	0.9310
		20.0	39	0.0046	0.3156	0.9756		
L_{rt} , with $N_{max} = 60$	1	5.0	60	0.0047	0.3132	0.9636		
	2		40	0.0001	0.2236	0.8172		
	3		25	0.0016	0.2282	0.7787		
	4		13	0.0165	0.2404	0.7391		
	5		17	0.0012	0.2023	0.7119		
	6		9	0.0006	0.1882	0.5800		
	7		8	0.0028	0.1774	0.6919		
	8		6	0.0018	0.1611	0.5672		
	9		3	0.0037	0.1543	0.4278		
	10		3	0.0049	0.1554	0.3975		
	20		0	0.0110	0.1312	0.3099		
	30		1	0.0017	0.1257	0.2933		
	40		0	0.0169	0.1096	0.2783		
				10.0	45	0.0015	0.2871	0.9297
	1	20.0	35	0.0009	0.2059	0.6129		
		30.0	41	0.0001	0.1539	0.6507		
		40.0	49	0.0028	0.1822	0.6147		

Table 5.7: Results for dataset inst_j20_r5.

Variations	N changes	$ \delta $	% cases $\pi^A = \pi^B$ (remains optimal)	other cases: G(A,B)				
				min	mean	max		
W_{jr} , with $N_{max} = 100$	1	5.0	39	0.0015	0.3880	1.0		
	2		35	0.0024	0.3509	0.9219		
	3		27	0.0105	0.2896	0.9789		
	4		22	0.0009	0.2355	0.8836		
	5		22	0.0014	0.2271	0.9314		
	6		12	0.0070	0.1976	0.8712		
	7		14	0.0010	0.1586	0.8450		
	8		17	0.0003	0.1740	0.8093		
	9		9	0.0081	0.1577	0.8290		
	10		9	0.0018	0.1847	0.9628		
	20		4	0.0021	0.1299	0.6112		
	30		4	0.0015	0.1121	0.6652		
	40		3	0.0081	0.1067	0.6300		
				10.0	37	0.0052	0.4120	0.9511
	1	20.0	33	0.0150	0.3929	0.9639		
		30.0	33	0.0113	0.4455	1.0		
		40.0	23	0.0073	0.4002	1.0		
L_{rt} , with $N_{max} = 150$	1	5.0	53	0.0033	0.4043	1.0		
	2		32	0.0014	0.3621	0.9964		
	3		23	0.0034	0.2541	0.9452		
	4		21	0.0097	0.2257	0.9905		
	5		15	0.0035	0.2047	0.9604		
	6		20	0.0083	0.1714	0.6456		
	7		12	0.0006	0.1842	0.7771		
	8		10	0.0002	0.1530	0.4659		
	9		7	0.0051	0.1449	0.5921		
	20		2	0.0001	0.1124	0.3250		
	40		0	0.0044	0.1033	0.2825		
	60		0	0.0218	0.0907	0.1834		
				10.0	41	0.0057	0.3906	1.0
			1	20.0	35	0.0091	0.3502	0.9911
	30.0	30		0.0005	0.3309	0.9991		
	40.0	24		0.0047	0.3052	0.9577		

Experimental results demonstrate that a schedule has significant stability. The reallocation of resources within the same schedule works better for a single peak change (about 30-45%), but also possible in some multiple-change cases (1-15%).

The value of relative accuracy $G(A, B)$ for the cases where the optimality was not reached is gradually decreasing with the number and the magnitude of changes. The percentage of saved optimal schedules is higher for the case of single fluctuation in input data: with the mean value about 0.3–0.4 for a single fluctuation and 0.1–0.2 for multiple changes. The accuracy of the method is not decreasing for large-scale problem instances.

5.6 Conclusions

To sum up, we have developed the concept of metric-based estimation for a generalized formulation of the RLP problem with an objective function minimizing an extra resource usage cost. The idea is based on the consideration of the instance space and the upper bound metric accuracy estimations for the case where a solution found for one instance is applied to another instance.

We developed this method for the resource leveling problems and proved that in our case some practically significant variations in input data provide linearly bounded changes in solution quality. It was demonstrated for all resource-job parameters, such as demanded and available resource amount. We considered the problem as two logical parts: scheduling and resource allocation. Resource allocation can be performed with an LP model with a given baseline schedule. We described theoretically the feasibility requirements for a baseline schedule, and use it as a robust partial solution. Experimental results show that a significant part of resource-related input data variations still allows us to apply the baseline schedule found for one instance to the modified problem instance.

Chapter 6

General conclusions

This thesis proposes new modeling techniques to tackle problems with integrated project planning and scheduling. These problems arise in all areas and become more and more important nowadays. The scale and complexity of projects are growing, and it increases the risk of possible losses from poor quality planning. For some systems (e.g., transport), the mode of operation is usually predicted for a long time but requires a quick response in case of sudden incidents and changes in the situation. Besides the quality of planning, the speed of analysis and reaction is also important.

Mathematically justified decision-making can improve both indicators. Chapter 1 illustrates modern practical cases, challenges, and the achievements of planning systems. It shows that current models aim at efficiency and resource management and the respect of deadlines. We also note that in many real-life cases, the uncertainty of the data makes high-quality planning more difficult. Unexpected changes and accidents ruin the plans and set tough limits on the time given to rebuild alternative solutions.

This thesis is dedicated to the development of mathematical models for project planning and scheduling with attention to these points. In Chapter 2, we present state-of-the-art models and methods starting with the original formulation of Resource-Constrained Project Scheduling Problem. Many modeling techniques and solution methods were developed for RCPSp, but are also common for a wide class of project planning problems as Resource Leveling Problems (RLP). In contrast to RCPSp that minimizes the project completion time, models the objective functions for RLP are oriented to the resource utilization. We also review the uncertainty modeling for these classes of planning problems.

Chapter 3 introduces and describes the particular RLP statement studied in this thesis. It is a generalized RLP that reduces resource overload costs within a fixed project deadline. The problem includes additional features: variable job duration and resource allocation per job over periods. We propose a flexible resource allocation that allows the assignment of different resource types independently in one time period for each job. On the one hand, it requires additional decision variables and increases the size of the model. On

the other hand, it allows constructing better solutions with lower costs. We study this model on benchmark instances and compare our model with the basic formulations from the literature.

Tests on existing RLP benchmarks show that independent resource allocation reduces the costs by 7% on average. Even if the solution time is longer, however, even within an equal time limit, our model provides better suboptimal solutions (by 6% on average). Additional tests on new generated data sets show that for a higher level of available resources, our model may provide the reduction of costs up to 50% on average.

In Chapter 4, we describe a new solution approach based on Benders decomposition. It decomposes the problem into the MILP master problem and LP subproblem. Dual LP subproblems use the partial solutions of the master problem to construct additional cuts. We combine a set of various improvements, both at the structural level of the algorithm and at the technical implementation level. An important point is the use of a single search tree instead of multiple master problem runs (so-called Branch&Benders cuts). Valid inequalities eliminate infeasible partial solutions for the master problem and therefore reduce the number of iterations. The lower bounds for the master problem accelerate the convergence of the approach. On the technical level, we implemented the single search tree with so-called callback procedures. We performed computational experiments with CPLEX 12.10 solver.

We compared this algorithm with the built-in solver Benders decomposition and standard B&C for our RLP model. The results of numerical experiments show the efficiency of the algorithm. It significantly outperforms built-in Benders decomposition and standard B&C, especially with a growing number of resource types. It is capable of solving larger instances. For example, it solved some problem instances with 50 jobs and 10 resource types to optimality. Within the same time limit, the built-in Benders and B&C had a relative gap from 1.0 to 0.5.

We discuss the uncertainty of the data for our formulation of RLP in Chapter 5. We explore a metric approach that uses the solution found for one problem instance in order to propose it to another problem instance with the estimated difference in the value of the objective function. For our formulation of RLP, it is difficult to find polynomially solvable subsets, however, it is possible to construct an estimation of the upper bound for the error on the optimal value for the problem instance that has not been solved. This estimation may help to decide if it is necessary to solve this new instance. For example, if an accident reduces the number of available resources, it is important to decide if the initially planned solution still can be used without important losses or the project should be immediately rescheduled in order to avoid significant penalties.

We propose and study two options: either it is possible to keep the initial solution or a schedule only. The schedule is a set of decision variables defining the start and end

times for jobs. The second option has several advantages. It allows us to adapt the resource allocation to new conditions in polynomial time (by solving an LP subproblem). Thus, higher accuracy can be achieved with this suboptimal approximation. We study the conditions where a solution or a schedule found for problem instance A can be applicable to problem instance B .

First, we provide an upper bound estimation for the case where a solution found for one problem instance is applied to another problem instance. Second, we formulate the conditions where it is possible, we assess the robustness of a schedule. We study variations of all input parameters except for the precedence relations. The computational experiments are conducted on the instances with variations in resource demand and supply within the range of 10%–100% in magnitude and with the increasing number of variations from one to the possible maximum. The option of keeping the schedule is stable both in terms of feasibility and optimality. It is possible to keep the schedule and guarantee the optimality of the solution in 30-45% of cases for a single variation, and up to 15% for multiple variations.

The generalized formulation of RLP studied in this thesis can be enriched with more practical constraints as consideration of non-renewable resources, discrete resource capacities, different objective functions, etc.

In terms of future solution methods, it would be interesting to implement the whole problem or its scheduling part with constraint programming, since the current CP solvers show relatively good results for scheduling problems.

Regarding data uncertainty, experimental results inspire us to find a theoretical description of conditions that guarantee that the solution remains optimal with given data variations. Another point is to study the variations of precedence relations. That leads to the next step, the development of rescheduling procedures for the cases where the known schedule can no more be applied to a changed instance. For the class of problems that can be solved in a repetitive way, Machine Learning approaches can be also promising to anticipate the changes in the input data.

Chapter 7

Bibliography

- [1] A. Alan B. Pritsker, Lawrence J. Watters, and Philip M. Wolfe. Multiproject scheduling with limited resources: A zero-one programming approach. *Management Science*, 16(1):93–108, 1969.
- [2] J. Blazewicz, J.K. Lenstra, and A.H.G.Rinnooy Kan. Scheduling subject to resource constraints: classification and complexity. *Discrete Applied Mathematics*, 5(1):11–24, 1983.
- [3] Emmanuel Néron, Christian Artigues, Philippe Baptiste, Jacques Carlier, Jean Damay, Sophie Demassey, and Philippe Laborie. *Lower Bounds for Resource Constrained Project Scheduling Problem*, pages 167–204. Springer US, Boston, MA, 2006.
- [4] Sigrid Knust. *Lower Bounds on the Minimum Project Duration*, pages 43–55. Springer International Publishing, Cham, 2015.
- [5] David R. Morrison, Sheldon H. Jacobson, Jason J. Sauppe, and Edward C. Sewell. Branch-and-bound algorithms: A survey of recent advances in searching, branching, and pruning. *Discrete Optimization*, 19:79–102, 2016.
- [6] S.E. Terblanche and J.H. Van Vuuren. Benders decomposition of the resource constrained project scheduling problem. Technical Report FABWI-N-RA-2017-530, Centre for Business Mathematics and Informatics, North-West University, South Africa, 2017.
- [7] M. Boschetti and V. Maniezzo. Benders decomposition, lagrangean relaxation and metaheuristic design. *Journal of Heuristics*, 15(3):283–312, Jun 2009.
- [8] Marco Giuntoli, Milos Subasic, and Susanne Schmitt. Control of distribution grids with storage using nested benders’ decomposition. *Electric Power Systems Research*, 190:106663, 2021.

- [9] Jesús A. Rodríguez, Miguel F. Anjos, Pascal Côté, and Guy Desaulniers. Accelerating benders decomposition for short-term hydropower maintenance scheduling. *European Journal of Operational Research*, 289(1):240–253, 2021.
- [10] Anurag Agarwal, Selcuk Colak, and Selcuk Erenguc. *Metaheuristic Methods*, pages 57–74. Springer International Publishing, Cham, 2015.
- [11] 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.
- [12] Dirk Briskorn and Michael Dienstknecht. Survey of quantitative methods in construction. *Computers & Operations Research*, 92:194–207, 2018.
- [13] Philippe Laborie. Algorithms for propagating resource constraints in ai planning and scheduling: Existing approaches and new results. *Artificial Intelligence*, 143(2):151–188, 2003.
- [14] Julia Rieck and Jürgen Zimmermann. *Exact Methods for Resource Leveling Problems*, pages 361–387. Springer International Publishing, Cham, 2015.
- [15] Christian Artigues, Pierre Lopez, and Alain Haït. The energy scheduling problem: Industrial case-study and constraint propagation techniques. *International Journal of Production Economics*, 143(1):13–23, 2013.
- [16] Margaux Nattaf, Markó Horváth, Tamás Kis, Christian Artigues, and Pierre Lopez. Polyhedral results and valid inequalities for the continuous energy-constrained scheduling problem. *Discrete Applied Mathematics*, 258, 04 2019.
- [17] Mikaël Capelle, Marie-José Hugué, Nicolas Jozefowicz, and Xavier Olive. Ground stations networks for free-space optical communications: maximizing the data transfer. *Electronic Notes in Discrete Mathematics*, 64:255–264, 2018. 8th International Network Optimization Conference - INOC 2017.
- [18] G. Baydoun, A. Haït, R. Pellerin, B. Cément, and G. Bouvignies. A rough-cut capacity planning model with overlapping. *OR Spectrum*, 38(2):335–364, Mar 2016.
- [19] 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, Mar 2016.
- [20] P.-A. Morin, C. Artigues, and A. Haït. Periodically aggregated resource-constrained project scheduling problem. *European J. of Industrial Engineering*, 11:792, 01 2017.

- [21] Maria Elena Bruni, Patrizia Beraldi, and Francesca Guerriero. *The Stochastic Resource-Constrained Project Scheduling Problem*, pages 811–835. Springer International Publishing, Cham, 2015.
- [22] Christian Artigues, Oumar Koné, Pierre Lopez, and Marcel Mongeau. *Mixed-Integer Linear Programming Formulations*, pages 17–41. Springer International Publishing, Cham, 2015.
- [23] Oncu Hazir, Erdal Erel, and Yavuz Gunalay. Robust optimization models for the discrete time/cost trade-off problem. *International Journal of Production Economics*, 130(1):87–95, 2011.
- [24] Oncu Hazir and Gunduz Ulusoy. A classification and review of approaches and methods for modeling uncertainty in projects. *International Journal of Production Economics*, 223:107522, 2020.
- [25] Stan Schenkerman. Sensitivity of linear programs to related changes in multiple inputs*. *Decision Sciences*, 24(4):879–891, 1993.
- [26] Harvey M. Wagner. Global sensitivity analysis. *Operations Research*, 43(6):948–969, 1995.
- [27] M. W. Dawande and J. N. Hooker. Inference-based sensitivity analysis for mixed integer/linear programming. *Operations Research*, 48(4):623–634, 2000.
- [28] Zhenya Jia and Marianthi G. Ierapetritou. Short-term scheduling under uncertainty using MILP sensitivity analysis. *Industrial & Engineering Chemistry Research*, 43(14):3782–3791, 2004.
- [29] M. Ierapetritou and Zhenya Jia. Scheduling under uncertainty using MILP sensitivity analysis. In *European Symposium on Computer-Aided Process Engineering-14*, volume 18 of *Computer Aided Chemical Engineering*, pages 931–936. Elsevier, 2004.
- [30] Vladimir Emelichev, Evgeny Gurevsky, and Andrey Platonov. On stability and quasi-stability radii for a vector combinatorial problem with a parametric optimality principle. *Buletinul Academiei de Ştiinţe a Republicii Moldova. Matematica*, (2), 2009.
- [31] Evgeny Gurevsky, Olga Battaïa, and Alexandre Dolgui. Stability measure for a generalized assembly line balancing problem. *Discrete Applied Mathematics*, 161(3):377–394, 2013.

- [32] A. A. Lazarev. Estimates of the absolute error and a scheme for an approximate solution to scheduling problems. *Computational Mathematics and Mathematical Physics*, 49(2):373–386, Feb 2009.
- [33] A. A. Lazarev, P. S. Korenev, and A. A. Sologub. A metric for total tardiness minimization. *Automation and Remote Control*, 78(4):732–740, Apr 2017.
- [34] A. A. Lazarev and A. G. Kvaratskheliya. Metrics in scheduling problems. *Doklady Mathematics*, 81(3):497–499, Jun 2010.
- [35] Frederick Winslow Taylor. *The Principles of Scientific Management*. McMaster University Archive for the History of Economic Thought, 1911.
- [36] H.L. Gantt. A graphical daily balance in manufacture. *Transactions of the American Society of Mechanical Engineers*, 24:1322–1336, 1903.
- [37] James E. Kelley and Morgan R. Walker. Critical-path planning and scheduling. In *Papers Presented at the December 1-3, 1959, Eastern Joint IRE-AIEE-ACM Computer Conference*, IRE-AIEE-ACM '59 (Eastern), page 160–173, New York, NY, USA, 1959. Association for Computing Machinery.
- [38] D. G. Malcolm, J. H. Roseboom, C. E. Clark, and W. Fazar. Application of a technique for research and development program evaluation. *Operations Research*, 7(5):646–669, 1959.
- [39] Richard Bellman. Mathematical aspects of scheduling theory. *Journal of the Society for Industrial and Applied Mathematics*, 4(3):168–205, 1956.
- [40] Richard Walter Conway, W.L. Maxwell, J.A. Cronin, W.L. Maxwell, L.W. Miller, J.R. Shoenfield, and L.W. Miller. *Theory of Scheduling*. Addison-Wesley Publishing Company, 1967.
- [41] Stephen A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the Third Annual ACM Symposium on Theory of Computing*, STOC '71, page 151–158, New York, NY, USA, 1971. Association for Computing Machinery.
- [42] L.A. Levin. Universal sequential search problems. *Probl. Peredachi Inf.*, 9(3):115–116, 1973.
- [43] Richard M. Karp. *Reducibility among Combinatorial Problems*, pages 85–103. Springer US, Boston, MA, 1972.
- [44] Peter Brucker, Silvia Heitmann, and Sigrid Knust. Scheduling railway traffic at a construction site. *OR Spectrum*, (24):19–30, 2002.

- [45] Juan Pablo Usuga Cadavid, Samir Lamouri, Bernard Grabot, Robert Pellerin, and Arnaud Fortin. Machine learning applied in production planning and control: a state-of-the-art in the era of industry 4.0. *Journal of Intelligent Manufacturing*, 31(6):1531–1558, 2020.
- [46] Franco Peschiera, Robert Dell, Johannes Royset, Alain Haït, Nicolas Dupin, and Olga Battaïa. A novel solution approach with ml-based pseudo-cuts for the flight and maintenance planning problem. *OR Spectrum*, 2020.
- [47] Andrea Lodi, Luca Mossina, and Emmanuel Rachelson. Learning to handle parameter perturbations in combinatorial optimization: An application to facility location. *EURO Journal on Transportation and Logistics*, page 100023, 2020.
- [48] Martina Fischetti and Marco Fraccaro. Machine learning meets mathematical optimization to predict the optimal production of offshore wind parks. *Computers & Operations Research*, 106:289–297, 2019.
- [49] Hongbo Li, Zhe Xu, and Erik Demeulemeester. Scheduling policies for the stochastic resource leveling problem. *Journal of Construction Engineering and Management*, 141(2):04014072, 2015.
- [50] Hongbo Li and Erik Demeulemeester. A genetic algorithm for the robust resource leveling problem. *J. of Scheduling*, 19(1):43–60, February 2016.
- [51] R Graham, E Lawler, J Lenstra, and A Rinnooy Kan. Optimization and approximation in deterministic sequencing and scheduling: a survey. *Ann. of Discrete Mathematics*, 5:287–326, 1979.
- [52] P. Brucker and S. Knust. *Complex Scheduling*. Springer-Verlag Berlin Heidelberg, 2012.
- [53] R. Kolisch and S. Hartmann. *Heuristic Algorithms for the Resource-Constrained Project Scheduling Problem: Classification and Computational Analysis*, pages 147–178. Springer US, Boston, MA, 1999.
- [54] R Kolisch and A. Sprecher. Psplib – a project scheduling problem library. *European Journal of Operational Research*, 96(1):205–216, 1997.
- [55] J.A.G.M. Kerbosch and H.J. Schell. *Network planning by the extended metra potential method (EMPM)*. TH Eindhoven. ORS, Vakgr. operationele research : rapport. Technische Hogeschool Eindhoven, 1975. The Dutch version of this report is also available; it has been republished in the 1972 April edition of the journal Informatie.

- [56] Klaus Neumann and Christoph Schwindt. Activity-on-node networks with minimal and maximal time lags and their application to make-to-order production. *Operations-Research-Spektrum*, 19(3):205–217, 1997.
- [57] M. Bartusch, R. H. Möhring, and F. J. Radermacher. Scheduling project networks with resource constraints and time windows. *Annals of Operations Research*, 16(1):199–240, 1988.
- [58] A. Alan B. Pritsker and Lawrence J. Watters. *A Zero-One Programming Approach to Scheduling with Limited Resources*. RAND Corporation, Santa Monica, CA, 1968.
- [59] Nicos Christofides, R. Alvarez-Valdes, and J.M. Tamarit. Project scheduling with resource constraints: A branch and bound approach. *European Journal of Operational Research*, 29(3):262–273, 1987.
- [60] Robert Klein. *Scheduling of resource constrained projects*. Springer US, 2000.
- [61] Sophie Demassej, Christian Artigues, and Philippe Michelon. Constraint-propagation-based cutting planes: An application to the resource-constrained project scheduling problem. *INFORMS Journal on Computing*, 17(1):52–65, 2005.
- [62] Jill R. Hardin, George L. Nemhauser, and Martin W.P. Savelsbergh. Strong valid inequalities for the resource-constrained scheduling problem with uniform resource requirements. *Discrete Optimization*, 5(1):19–35, 2008.
- [63] Christian Artigues. A note on time-indexed formulations for the resource-constrained project scheduling problem. 15 pages, June 2013.
- [64] Lucio Bianco and Massimiliano Caramia. An exact algorithm to minimize the makespan in project scheduling with scarce resources and generalized precedence relations. *European Journal of Operational Research*, 219(1):73–85, 2012.
- [65] Lucio Bianco and Massimiliano Caramia. A new formulation for the project scheduling problem under limited resources. *Flexible Services and Manufacturing Journal*, 25(1):6–24, 2013.
- [66] Christian Artigues. On the strength of time-indexed formulations for the resource-constrained project scheduling problem. *Operations Research Letters*, 45(2):154–159, 2017.
- [67] 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.

- [68] Thomas S. Kyriakidis, Georgios M. Kopanos, and Michael C. Georgiadis. MILP formulations for single- and multi-mode resource-constrained project scheduling problems. *Computers & Chemical Engineering*, 36:369–385, 2012.
- [69] Stefan Kreter, Julia Rieck, and Jürgen Zimmermann. Models and solution procedures for the resource-constrained project scheduling problem with general temporal constraints and calendars. *European Journal of Operational Research*, 251(2):387–403, 2016.
- [70] Shima Javanmard, Behrouz Afshar-Nadjafi, and Seyed Taghi Akhavan Niaki. Pre-emptive multi-skilled resource investment project scheduling problem: Mathematical modelling and solution approaches. *Computers & Chemical Engineering*, 96:55–68, 2017.
- [71] Juan Diego García-Nieves, José Luis Ponz-Tienda, Angélica Ospina-Alvarado, and Mateo Bonilla-Palacios. Multipurpose linear programming optimization model for repetitive activities scheduling in construction projects. *Automation in Construction*, 105:102799, 2019.
- [72] 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(5):714–729, 1998.
- [73] Alf Kimms. *Mathematical Programming and Financial Objectives for Scheduling Projects*, volume 38. 01 2001.
- [74] Maurice Queyranne, Maurice Queyranne, Andreas S. Schulz, and Andreas S. Schulz. Polyhedral approaches to machine scheduling. Technical report, 1994.
- [75] Christian Artigues, Philippe Michelon, and Stéphane Reusser. Insertion techniques for static and dynamic resource-constrained project scheduling. *European Journal of Operational Research*, 149(2):249–267, 2003. Sequencing and Scheduling.
- [76] Oumar Koné, Christian Artigues, Pierre Lopez, and Marcel Mongeau. Comparison of mixed integer linear programming models for the resource-constrained project scheduling problem with consumption and production of resources. *Flexible Services and Manufacturing Journal*, 25(1):25–47, 2013.
- [77] Christian Artigues, Peter Brucker, Sigrid Knust, Oumar Koné, Pierre Lopez, and Marcel Mongeau. A note on "event-based MILP models for resource-constrained project scheduling problems". *Computers and Operations Research*, 40(4):1060–1063, 2013.

- [78] Ed Klotz and Alexandra M. Newman. Practical guidelines for solving difficult mixed integer linear programs. *Surveys in Operations Research and Management Science*, 18(1):18–32, 2013.
- [79] Juan Pablo Vielma. Mixed integer linear programming formulation techniques. *SIAM Review*, 57(1):3–57, 2015.
- [80] Klaus Neumann, Christoph Schwindt, and Jürgen Zimmermann. *Resource-Constrained Project Scheduling — Minimization of General Objective Functions*, pages 175–299. Springer Berlin Heidelberg, Berlin, Heidelberg, 2002.
- [81] E.W. Hans. *Resource Loading by Branch-and-Price Techniques*. PhD thesis, Twente University Press (TUP), Netherlands, 10 2001.
- [82] Tamás Kis. A branch-and-cut algorithm for scheduling of projects with variable-intensity activities. *Mathematical Programming*, 103(3):515–539, Jul 2005.
- [83] Symeon E. Christodoulou, Anastasia Michaelidou-Kamenou, and Georgios Ellinas. *Heuristic Methods for Resource Leveling Problems*, pages 389–407. Springer International Publishing, Cham, 2015.
- [84] T. Ch. Madhavi Tadepalli. Genetic algorithm based optimization for resource leveling problem with precedence constrained scheduling. *International journal of construction management.*, 2019.
- [85] S. Madeh Piryonesi, Mehran Nasserli, and Abdollah Ramezani. Resource leveling in construction projects with activity splitting and resource constraints: a simulated annealing optimization. *Canadian Journal of Civil Engineering*, 46(2):81–86, 2019.
- [86] Hongbo Li, Meng Wang, and Xuebing Dong. Resource leveling in projects with stochastic minimum time lags. *Journal of Construction Engineering and Management*, 145(4):04019015, 2019.
- [87] Kaouthar Cherkaoui, Pierre Baptiste, Robert Pellerin, Alain Haït, and Nathalie Perrier. Proactive tactical planning approach for large scale engineering and construction projects. *The Journal of Modern Project Management*, 5(1), 2017.
- [88] Margaux Nattaf, Christian Artigues, Pierre Lopez, and David Rivreau. Energetic reasoning and mixed-integer linear programming for scheduling with a continuous resource and linear efficiency functions. *OR Spectrum*, 38(2):459–492, 2016.
- [89] A. H. Land and A. G. Doig. An automatic method of solving discrete programming problems. *Econometrica*, 28(3):497–520, 1960.

- [90] J. Clausen and M. Perregaard. On the best search strategy in parallel branch-and-bound: Best-first search versus lazy depth-first search. *Annals of Operations Research*, 90(0):1–17, 1999.
- [91] R.E. Gomory. An algorithm for integer solutions to linear programs. rl graves and p. wolfe (eds.) recent advances in mathematical programming, 1963.
- [92] V. Chvátal. Edmonds polytopes and a hierarchy of combinatorial problems. *Discrete Mathematics*, 4(4):305–337, 1973.
- [93] Alberto Caprara and Matteo Fischetti. $\{0, 1/2\}$ -chvátal-gomory cuts. *Mathematical Programming*, 74(3):221–235, 1996.
- [94] Adam N. Letchford and Andrea Lodi. Strengthening chvátal–gomory cuts and gomory fractional cuts. *Operations Research Letters*, 30(2):74–82, 2002.
- [95] Franz Wesselmann, Achim Koberstein, and Uwe H. Suhl. Strengthening gomory mixed-integer cuts. In Bernhard Fleischmann, Karl-Heinz Borgwardt, Robert Klein, and Axel Tuma, editors, *Operations Research Proceedings 2008*, pages 487–492, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.
- [96] Egon Balas, Sebastián Ceria, and Gérard Cornuéjols. A lift-and-project cutting plane algorithm for mixed 0–1 programs. *Mathematical Programming*, 58(1):295–324, 1993.
- [97] Egon Balas and Michael Perregaard. Lift-and-project for mixed 0–1 programming: recent progress. *Discrete Applied Mathematics*, 123(1):129–154, 2002.
- [98] Gérard Cornuéjols. Valid inequalities for mixed integer linear programs. *Mathematical Programming*, 112(1):3–44, 2008.
- [99] Santanu S. Dey and Marco Molinaro. Theoretical challenges towards cutting-plane selection. *Mathematical Programming*, 170(1):237–266, 2018.
- [100] Dmitry Arkhipov, Olga Battaïa, and Alexander Lazarev. An efficient pseudo-polynomial algorithm for finding a lower bound on the makespan for the resource constrained project scheduling problem. *European Journal of Operational Research*, 275(1):35–44, 2019.
- [101] Alexander Tesch. Improved compact models for the resource-constrained project scheduling problem. In Andreas Fink, Armin Fügenschuh, and Martin Josef Geiger, editors, *Operations Research Proceedings 2016*, pages 25–30, Cham, 2018. Springer International Publishing.

- [102] Janniele A.S. Araujo, Haroldo G. Santos, Bernard Gendron, Sanjay Dominik Jena, Samuel S. Brito, and Danilo S. Souza. Strong bounds for resource constrained project scheduling: Preprocessing and cutting planes. *Computers & Operations Research*, 113:104782, 2020.
- [103] Lucio Bianco and Massimiliano Caramia. *Lower Bounds and Exact Solution Approaches*, pages 77–111. Springer International Publishing, Cham, 2015.
- [104] J. Carlier, A. Moukrim, and A. Sahli. Lower bounds for the event scheduling problem with consumption and production of resources. *Discrete Applied Mathematics*, 234:178–194, 2018. Special Issue on the Ninth International Colloquium on Graphs and Optimization (GO IX), 2014.
- [105] Inessa Ainbinder, Gabriel David Pinto, and Gad Rabinowitz. New lower bounds for solving a scheduling problem with resource collaboration. *Computers & Industrial Engineering*, 127:225–239, 2019.
- [106] Lucio Bianco and Massimiliano Caramia. Minimizing the completion time of a project under resource constraints and feeding precedence relations: a lagrangian relaxation based lower bound. *4OR*, 9(4):371–389, Dec 2011.
- [107] C.-U. Fundeling and N. Trautmann. A priority-rule method for project scheduling with work-content constraints. *European Journal of Operational Research*, 203(3):568–574, 2010.
- [108] Anulark Naber. Resource-constrained project scheduling with flexible resource profiles in continuous time. *Computers & Operations Research*, 84:33–45, 2017.
- [109] J.F. Benders. Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 4(1):238–252, Dec 1962.
- [110] R.S. De Camargo, G. De Miranda, and H. Luna. Benders decomposition for hub location problems with economies of scale. *Transportation Science*, 43(1):86–97, 2009.
- [111] H. Li. *Benders Decomposition Approach for Project Scheduling with Multi-Purpose Resources*, pages 587–601. Springer International Publishing, Cham, 2015.
- [112] M.E. Bruni, L.D.P. Pugliese, P. Beraldi, and F. Guerriero. An adjustable robust optimization model for the resource-constrained project scheduling problem with uncertain activity durations. *Omega*, 71:66–84, 2017.

- [113] Simon Emde, Lukas Polten, and Michel Gendreau. Logic-based benders decomposition for scheduling a batching machine. *Computers & Operations Research*, 113:104777, 2020.
- [114] Mehdi A. Kamran, Behrooz Karimi, and Nico Dellaert. A column-generation-heuristic-based benders’ decomposition for solving adaptive allocation scheduling of patients in operating rooms. *Computers & Industrial Engineering*, 148:106698, 2020.
- [115] Ramon Faganello Fachini and Vinícius Amaral Armentano. Logic-based benders decomposition for the heterogeneous fixed fleet vehicle routing problem with time windows. *Computers & Industrial Engineering*, 148:106641, 2020.
- [116] Evren Guney, Markus Leitner, Mario Ruthmair, and Markus Sinnl. Large-scale influence maximization via maximal covering location. *European Journal of Operational Research*, 289(1):144–164, 2021.
- [117] R. Rahmaniani, T.G. Crainic, M. Gendreau, and W. Rei. The benders decomposition algorithm: A literature review. *European Journal of Operational Research*, 259(3):801–817, 2017.
- [118] Mohammad M. Fazel-Zarandi and J. Christopher Beck. Using logic-based benders decomposition to solve the capacity- and distance-constrained plant location problem. *INFORMS Journal on Computing*, 24(3):387–398, 2012.
- [119] Stephen J. Maher. Implementing the branch-and-cut approach for a general purpose benders’ decomposition framework. *European Journal of Operational Research*, 2020.
- [120] Matteo Fischetti, Ivana Ljubić, and Markus Sinnl. Benders decomposition without separability: A computational study for capacitated facility location problems. *European Journal of Operational Research*, 253(3):557–569, 2016.
- [121] Saeed Emami, Ghasem Moslehi, and Mohammad Sabbagh. A benders decomposition approach for order acceptance and scheduling problem: a robust optimization approach. *Computational and Applied Mathematics*, 36(4):1471–1515, 2017.
- [122] T. L. Magnanti and R. T. Wong. Accelerating benders decomposition: Algorithmic enhancement and model selection criteria. *Operations Research*, 29(3):464–484, 1981.
- [123] Lixin Tang, Wei Jiang, and Georgios K. D. Saharidis. An improved benders decomposition algorithm for the logistics facility location problem with capacity expansions. *Annals of Operations Research*, 210(1):165–190, Nov 2013.

- [124] Georgios K.D. Saharidis and Marianthi G. Ierapetritou. Improving benders decomposition using maximum feasible subsystem (mfs) cut generation strategy. *Computers & Chemical Engineering*, 34(8):1237–1245, 2010.
- [125] Petr Vilím. *Global Constraints in Scheduling*. PhD thesis, Charles University in Prague, Faculty of Mathematics and Physics, Department of Theoretical Computer Science and Mathematical Logic, KTIML MFF, Universita Karlova, Malostranské náměstí 2/25, 118 00 Praha 1, Czech Republic, August 2007.
- [126] Abdelkader Lahrichi. Ordonnancements. la notion de « parties obligatoires » et son application aux problèmes cumulatifs. *RAIRO - Operations Research - Recherche Opérationnelle*, 16(3):241–262, 1982.
- [127] B. Fox. Non-chronological scheduling. In *1990 Simulation and Planning in High Autonomy Systems*, pages 72–77, Los Alamitos, CA, USA, mar 1990. IEEE Computer Society.
- [128] Arnaud Letort, Nicolas Beldiceanu, and Mats Carlsson. A scalable sweep algorithm for the cumulative constraint. In Michela Milano, editor, *Principles and Practice of Constraint Programming*, pages 439–454, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.
- [129] Steven Gay, Renaud Hartert, and Pierre Schaus. Simple and scalable time-table filtering for the cumulative constraint. In Gilles Pesant, editor, *Principles and Practice of Constraint Programming*, pages 149–157, Cham, 2015. Springer International Publishing.
- [130] J. Carlier, E. Pinson, A. Sahli, and A. Jouglet. An $o(n^2)$ algorithm for time-bound adjustments for the cumulative scheduling problem. *European Journal of Operational Research*, 286(2):468–476, 2020.
- [131] Viktoria A. Hauder, Andreas Beham, Sebastian Raggl, Sophie N. Parragh, and Michael Affenzeller. Resource-constrained multi-project scheduling with activity and time flexibility. *Computers & Industrial Engineering*, 150:106857, 2020.
- [132] Tom Servranckx and Mario Vanhoucke. A tabu search procedure for the resource-constrained project scheduling problem with alternative subgraphs. *European Journal of Operational Research*, 273(3):841–860, 2019.
- [133] Zihao Chu, Zhe Xu, and Haitao Li. New heuristics for the rcpsp with multiple overlapping modes. *Computers & Industrial Engineering*, 131:146–156, 2019.

- [134] Tarun Bhaskar, Manabendra N. Pal, and Asim K. Pal. A heuristic method for rcpsp with fuzzy activity times. *European Journal of Operational Research*, 208(1):57–66, 2011.
- [135] Weikang Guo, Mario Vanhoucke, José Coelho, and Jingyu Luo. Automatic detection of the best performing priority rule for the resource-constrained project scheduling problem. *Expert Systems with Applications*, page 114116, 2020.
- [136] Hua Ke and Chenkai Zhao. Uncertain resource leveling problem. *Journal of Intelligent & Fuzzy Systems*, 33:2351–2361, 09 2017.
- [137] Jin Zhu, Xiu Mei Zhang, and Wei Kang. MILP sensitivity analysis for short-term scheduling of batch plants under uncertainty. In *Advanced Materials Science and Technology, ICMST 2010*, volume 181 of *Advanced Materials Research*, pages 577–582. Trans Tech Publications Ltd, 3 2011.
- [138] Hongbo Ren and Weijun Gao. A MILP model for integrated plan and evaluation of distributed energy systems. *Applied Energy*, 87(3):1001–1014, 2010.
- [139] Panos Kouvelis, Charles L. Munson, and Shilei Yang. Robust structural equations for designing and monitoring strategic international facility networks. *Production and Operations Management*, 22(3):535–554, 2013.
- [140] Christian John Etwire and S. Twum. Sensitivity analysis of a mixed integer linear programming model for optimal hydrothermal energy generation for ghana. *International Journal of Scientific & Technology Research*, 4:129–141, 2015.
- [141] A. Moser, D. Muschick, M. Golles, P. Nageler, H. Schranzhofer, T. Mach, C. Ribas Tugores, I. Leusbrock, S. Stark, F. Lackner, and A. Hofer. A MILP-based modular energy management system for urban multi-energy systems: Performance and sensitivity analysis. *Applied Energy*, 261:114342, 2020.
- [142] Zhili Tian, Panos Kouvelis, and Charles L. Munson. Understanding and managing product line complexity: Applying sensitivity analysis to a large-scale MILP model to price and schedule new customer orders. *IIE Transactions*, 47(4):307–328, 2015.
- [143] Fahimeh Khoshniyat and Johanna Törnquist Krasemann. Analysis of Strengths and Weaknesses of a MILP Model for Revising Railway Traffic Timetables. In Gianlorenzo D’Angelo and Twan Dollevoet, editors, *17th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS 2017)*, volume 59 of *OpenAccess Series in Informatics (OASISs)*, pages 10:1–10:17, Dagstuhl, Germany, 2017. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

- [144] Z. Song, J. Zhang, and Q. Zhou. Milp method based network reconfiguration considering customer sensitivity to the service interruptions. In *2016 IEEE Power Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, pages 1–5, 2016.
- [145] Yi Chaojue and Lu Ming. Mixed-integer linear programming based sensitivity analysis in optimization of temporary haul road layout design for earthmoving operations. *Journal of Computing in Civil Engineering*, 33(3):04019021, 2019.
- [146] Javier Silvente, Lazaros G. Papageorgiou, and Vivek Dua. Scenario tree reduction for optimisation under uncertainty using sensitivity analysis. *Computers & Chemical Engineering*, 125:449–459, 2019.
- [147] A. A. Lazarev and D. I. Arkhipov. Estimation of the absolute error and polynomial solvability for a classical np-hard scheduling problem. *Doklady Mathematics*, 97(3):262–265, 2018.
- [148] Tamás Kis and Márton Drótos. *Hard Planning and Scheduling Problems in the Digital Factory*, pages 3–19. Springer International Publishing, Cham, 2017.
- [149] Rainer Kolisch, Christoph Schwindt, and Arno Sprecher. *Benchmark Instances for Project Scheduling Problems*, pages 197–212. Springer US, Boston, MA, 1999.
- [150] Julia Rieck, Jürgen Zimmermann, and Thorsten Gather. Mixed-integer linear programming for resource leveling problems. *European Journal of Operational Research*, 221(1):27–37, 2012.
- [151] Christoph Schwindt. Generation of resource-constrained project scheduling problems with minimal and maximal time lags. Institut für Wirtschaftstheorie und Operations Research, Universität, 1998.
- [152] Rainer Kolisch, Arno Sprecher, and Andreas Drexl. Characterization and generation of a general class of resource-constrained project scheduling problems. *Manage. Sci.*, 41(10):1693–1703, October 1995.
- [153] Erik Demeulemeester, Mario Vanhoucke, and Willy Herroelen. Rangen: A random network generator for activity-on-the-node networks. *Journal of Scheduling*, 6(1):17–38, 2003.
- [154] Mario Vanhoucke, José Coelho, Dieter Debels, Broos Maenhout, and Luís V. Tavares. An evaluation of the adequacy of project network generators with systematically sampled networks. *European Journal of Operational Research*, 187(2):511–524, 2008.

- [155] Mario Vanhoucke, José Coelho, and Jordy Batselier. An overview of project data for integrated project management and control. *The Journal of Modern Project Management*, 3(3), 2016.