Universidad Técnica Federico Santa María
Departamento de Informática

Doctoral Dissertation

for obtaining the academic degree of
Doctor in Informatics Engineering

Resource-constrained multi-project scheduling problem: taxonomy, variants, and approaches

by

Mariam Gómez Sánchez

Composition of the Jury

Supervised by:	Ph.D. Carlos Castro	UTFSM, Valparaíso - Chile
	Ph.D. Eduardo Lalla-Ruiz	UT, Enschede - The Netherlands
Coreferent:	Ph.D. María Cristina Riff Rojas	UTFSM, Valparaíso - Chile
National evaluator:	Ph.D. Broderick Crawford Labrín	PUCV, Valparaíso - Chile
International evaluator:	Ph.D. Éric Monfroy	UA, Angers - France
Commission chair:	Ph.D. Mauricio Solar Fuentes	UTFSM, Valparaíso - Chile

2023

Declaration of authorship

I hereby declare that I wrote this thesis on the subject

Resource-constrained multi-project scheduling problem: taxonomy, variants, and approaches

by

independently. I did not use any other aids, sources, figures or resources than those stated in the references. I clearly marked all passages that were taken from other sources and cited them correctly.

Furthermore I declare that – to my best knowledge – this work or parts of it have never before been submitted by me or somebody else at this or any other university.

Mariam Gómez Sánchez

Valparaíso, March 27, 2023 ____________________________

To me

Acknowledgements

I want to thank my advisors, Carlos Castro and Eduardo Lalla-Ruiz, for their dedication and for providing me with opportunities for professional growth. Thanks to Martijn Mes and Stefan Voß for their work and support. I am thankful to the professors at UTFSM and UT, as well as the support of the Dirección de Postgrado y Programas of the UTFSM and the Agencia Nacional de Investigación y Desarrollo de Chile through the grant CONICYT-PFCHA (Doctorado Nacional/2017-21171857). I want to thank everyone who has supported me during this long journey, mainly my family and friends, who have lovingly understood that most of the time, I have been busy. And I especially thank Alejandro for always being by my side.

Abstract

Project Management is becoming increasingly crucial in competitive environments such as manufacturing and the service industries. The resource-constrained multi-project scheduling problem (RCMPSP) consists of assigning start times to jobs corresponding to two or more projects that must be executed simultaneously while respecting the precedence between jobs and limited resources. The existing rise in the study of the RCMPSP resulted in numerous works on the topic while proposing different problem features.

This thesis presented a study of the RCMPSP and its variants. The base problem is initially described, and the solution methods and benchmarks used to solve RCMPSPs are classified and analyzed. The implications of the RCMPS with practice and its connection are also discussed. In addition, this research analyzed different variants of the problem based on aspects related to jobs, projects, relationships, resources, and time management. Furthermore, based on the problem variants considered in the reviewed works, a taxonomy allowing (i) the identification and positioning of each RCMPSP variant and (ii) the analysis of the current state-of-the-art of the problem was proposed.

Subsequently, it delved into the research of one of the most studied variants in the literature, RCMPSP with local resources (RCMPSP-L), describing the problem in detail and exposing its corresponding mathematical formulation. To solve the RCMPSP-L, a hybrid algorithm that integrates ant colony optimization with a hill-climbing first-improvement algorithm was proposed. The experimentation process was carried out using a set of instances taken from the MPSPLib. The proposed algorithm presented low variability values, the construction phase provided good-quality solutions, and the local search constantly improved our constructive algorithm. The results provided were compared with other well-known in the literature, evidencing that for different existing metrics, our approach presents competitive results.

Later, new variants of the problem that consider resource mixed accessibility, total resource allocation, delay costs, and resource costs were presented: RCMPSP-MT-W and RCMPSP-MT-W+RC, respectively. Both mentioned variants are described and mathematically formulated. A hybrid algorithm that integrates priority rules, simulated annealing, and tabu search as part

of an iterated local search was proposed to solve the variants. Simple perturbation and joint perturbation were proposed as different perturbation intensities as part of the ILS. To validate the model and the algorithm, 36 instances based on the MPSPLib library were modified. The results show that the application of joint perturbation yields better or equal results than simple perturbation, making greater use of resources with global accessibility. In addition, we identify that although it may not be the only factor, the global accessibility of resources may imply greater freedom of allocation, allowing better results to be obtained.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Project scheduling problems constitute one of the most critical problems for research and practice within the field of Operations Research. Pinto and Prescott (1990) consider the planning of the project schedule as one of the essential success factors in project planning. The first ideas related to this problem emerged at the beginning of the last century when the Gantt charts started to be used in Gantt (1913). Later in the 1930s, the study of problems related to Job Shop Scheduling was started (Jain and Meeran (1999)). In the 1950s, methods for the control and planning of projects, such as program evaluation and review techniques (Malcolm et al. (1959)) and the critical path method (Kelley and Walker (1959)), emerged. Studies and techniques developed to date were based on the existence of almost infinite resources. However, the development of industrial processes and the increase in complex production and consumption lines led, at the beginning of the 1960s, to the study of planning and control of projects considering the existence of limited resources. That resulted in the proposition of the resource-constrained project-scheduling problem (RCPSP) by Dike (1964), which was demonstrated to be NP-hard by Lenstra and Rinnooy Kan (1978).

The RCPSP consists of scheduling processing start times of several jobs that are part of a project optimizing a given objective function while respecting existing resource availability and precedence restrictions. A job can only start executing when its predecessor jobs have been completed and there is enough resource availability. This problem has been widely studied in the literature in research such as Chen (2011); Coelho and Vanhoucke (2011); Kong and Dou (2021); Laurent et al. (2017); Voß and Witt (2007); Wang et al. (2021); Zaman et al. (2021), and Brucker et al. (1999). Recently, Hartmann and Briskorn (2022) developed a follow-up survey to Hartmann and

Briskorn (2010); in both works, the authors presented an overview of variants of the RCPSP.

As an extension of the RCPSP, Pritsker et al. (1969) presented the resource-constrained multi-project scheduling problem (RCMPSP), where several projects must be executed simultaneously. In this sense, the RCMPSP has been one of the most studied and implemented extensions at the business level in different industry sectors. According to data presented by Lova and Tormos (2001), at the beginning of this century, 84% of the companies worldwide were working on multiple projects simultaneously, where efficient and effective management of resources was highlighted as a key factor for the long-term success of companies. In addition, the investigation of Karaa and Nasr (1986) indicated that the efficient management of resources provides a competitive advantage and an added value to those companies considering it.

In the RCMPSP-related literature, different features and objective functions have been studied, generating a wide range of variants related to jobs, projects, relations, resources, and execution times (see Chapter 3). Sometimes the RCPSP and the RCMPSP share features that generate new variants. That, together with a large amount of existing related research, has generated that some features have been studied under different nomenclature, despite presenting the same concept basis. Feature overlapping complicates the understanding of some proposed variants and highlights the need for a detailed study and description of the existing RCMPSP variants that allow for unifying and describing existing concepts. Throughout that study, researchers and practitioners can map what problems have been studied, how much progress has been made, and identify existing features that have not been studied in the literature.

The RCMPSP-related variants have been solved by a wide range of algorithms, including both exact and approximate. Although initially, the problem was solved mainly through mathematical models and priority rule-based heuristics, when studying the investigation of Issa and Tu (2020), it is possible to notice that most of the research works to solve the problem through approximate algorithms, including a wide range of hybrids algorithms. In this sense, Blum et al. (2008) state that combining metaheuristics, or heuristics, can provide efficient behavior when dealing with large-scale problems.

The above-mentioned, together with the importance and theoretical and practical transversality of the RCMPSP, represent the main motivations of this research. In this thesis, we study the RCMPSP based on the proposal of a taxonomy that supports the progress of the problem, as well as hybrid algorithms that allow solving both existing and new RCMPSP variants.

1.1 Research goals

To satisfy our motivations, the main goal of this research is to study the RCMPSP and its variants in depth to obtain a taxonomy that supports the progress of the problem and propose hybrid algorithms to solve existing and new variants.

The specific research objectives to accomplish our main goal are the following:

- Study in depth the existing literature related to RCMPSP, analyzing and discussing features and objective functions proposed, as well as the connection of the problem with the practice through the study of real cases.
- Develop and propose a taxonomy that classifies different aspects of the RCMPSP, providing a better understanding of the current state-of-the-art concerning that problem.
- Identify and propose new variants of the problem that contributes to the state-of-the-art.
- Propose a set of instances for the new proposed variants.
- Study, develop, and assess hybrid algorithms for solving existing and new proposed variants.

1.2 Research contributions

The contributions of this research work are the following:

- Identification, classification, and analysis of the features, objective functions, solution methods, and benchmarks in the literature related to the RCMPSP.
- Definition of a taxonomy that summarizes the features and objective functions found in the literature.
- Proposition of new features problem and definition of mathematical models of the new variants. That is, RCMPSP with resources mixed accessibility, total resource allocation, and weights associated with projects, and RCMPSP with resources mixed accessibility, total resource allocation, weights associated with projects, and resource cost, named RCMPSP-MT-W and RCMPSP-MT-W+RC, respectively.
- Proposition of a set of test instances for the RCMPSP-MT-W and RCMPSP-MT-W+RC

based on the MPSPLib, modifying the availability and accessibility of resources and adding the resource costs to units with global accessibility.

- Development of hybrid algorithms for the resolution of the RCMPSP with local resources (RCMPSP-L), RCMPSP-MT-W, and RCMPSP-MT-W+RC.

1.3 Overview of the thesis

This section indicates the general structure of this thesis.

- Chapter 2: Literature review. This chapter exposes the process of collecting the literature related to the RCMPSP. First, the RCMPSP is described in its most simplified form, referring to it as the base problem. Then, the mathematical model and an illustrative example are presented for the base problem. Finally, the solution methods and benchmarks used in the collected literature are identified, classified, and analyzed.
- Chapter 3: Taxonomy. This chapter proposes a taxonomy that summarizes the RCMPSP features and objective functions studied in the literature collected. The identified features and objective functions are described and classified according to the taxonomy presented. In addition, an analysis and discussion about features and objective functions are presented, as well as those works that address real-world instances.
- Chapter 4: RCMPSP with local resources. This chapter presents a study of the RCMPSP-L through a detailed description of the problem and its mathematical model. A hybrid algorithm is proposed to solve the problem. The approach integrates an ant colony optimization algorithm and a hill-climbing as an improvement solution method. The experimentation process is carried out on instances taken from the MPSPLib. The results obtained are analyzed and compared with state-of-the-art results from the literature.
- Chapter 5: RCMPSP with resources mixed accessibility. Based on the proposed taxonomy, this chapter presents two new variants of the RCMPSP, e.g., RCMPSP-MT-W and RCMPSP-MT-W+RC, RCMPSP with resources mixed accessibility, total resource allocation, and weights associated with projects, and RCMPSP with resources mixed accessibility, total resource allocation, weights associated with projects, and resource cost, respectively. Both variants are described, and their corresponding mathematical model is presented. In addition, a hybrid algorithm that integrates priority rules, simulated annealing, and tabu search as part of an iterated local search is presented. As part of the

ILS, two perturbation intensities are presented, simple and joint perturbation. In addition, the set of modified instances based on the MPSPLib is described. Finally, the results are analyzed concerning the resource usage, influence of the proposed perturbation intensity, and influence of related costs.

- Chapter 6: Conclusions. This chapter summarizes the main findings of this thesis and recommendations for future research.

Chapter 2

Literature review

The RCMPSP has been extensively studied, which is reflected in the large number of existing publications, presenting a considerable increase in the past and current decade (see Figure 2.1). The works Hartmann and Briskorn (2022) and Hartmann and Briskorn (2010) addressed a wide scope of RCPSP variants, including the RCMPSP. Given the vast number of related research from the different RCPSP variants, the authors only addressed a part of the research on the RCMPSP. Thus, there were several features that, to the best of our knowledge, are not described in the existing review works due to their understandable scope. On the other hand, Issa and Tu (2020) presented another survey on the RCPSP and the RCMPSP, which provides a classification in terms of flexibility concerning the resource use, duration, and preemption of the jobs. However, that survey only considered a very small number of RCMPSP-related works.

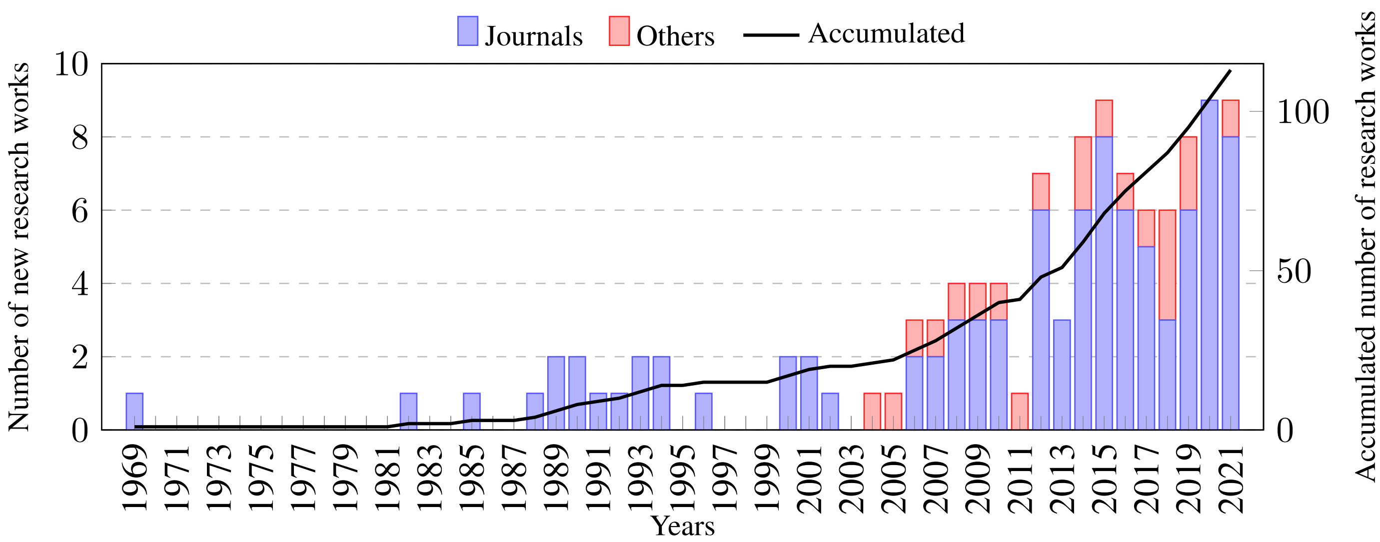

Figure 2.1: Distribution per year of research works that address the RCMPSP collected in the current document.

The scientific literature collected in this research corresponds to manuscripts indexed in *Scopus*, *WoS*, also considering publishers *IEEE Xplore*, *ScienceDirect by Elsevier*, *Wiley* and *SpringerLink*. The search process was based on (i) manuscripts written in the English language, (ii) including journal papers, conferences, and review papers, and (iii) searching for keywords in title, abstracts, and keywords. The used keywords were *resource-constrained*[1], *multi-project*[1], *project management, RCMPSP, RC-MPSP, and scheduling problem*. The full texts of the obtained manuscripts were analyzed to filter the papers that addressed the RCMPSP or related variants. The collected literature is scattered from 1969 to 2021, and the majority corresponds to journal articles in the areas of operations research, artificial intelligence, management science, and scheduling. The journals with the most RCMPSP-related works are the *European Journal of Operational Research*, *Computers & Industrial Engineering*, *Journal of Scheduling*, etc. Figure 2.2 shows the amount of research related to the RCMPSP carried out per year in the main journals. The width of the horizontal bars indicates the number of related papers published by year and journal (between 0 and 3). The total number of papers published in each journal from 1969 to 2021 is indicated on the right side of the figure. Only journals where at least three works were published are shown.

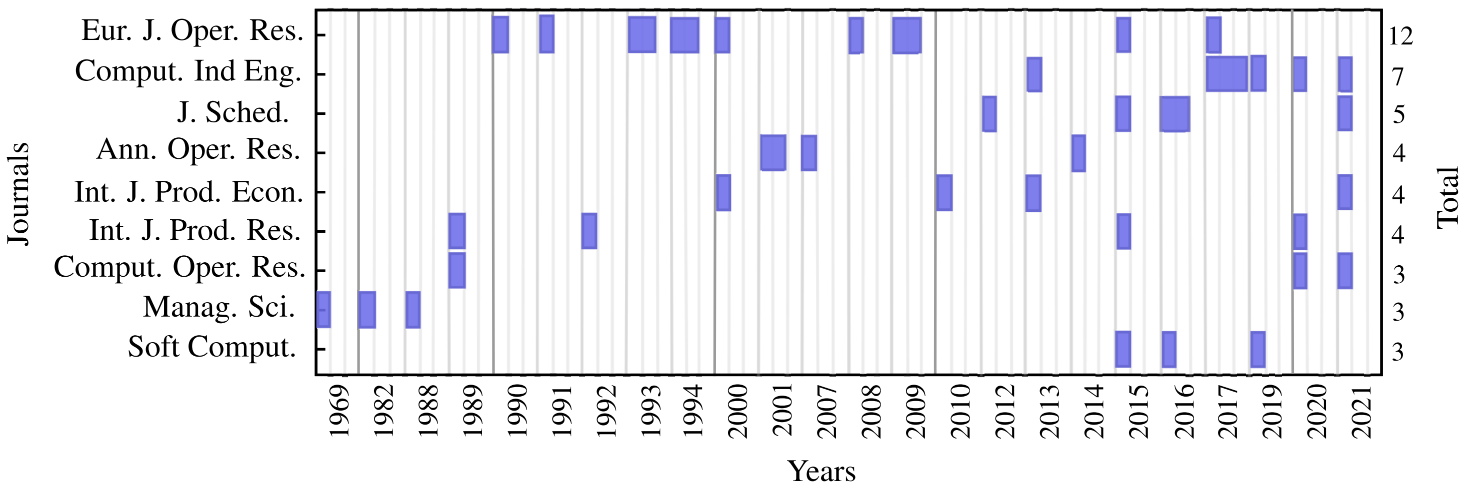

Figure 2.2: Distribution of the number of publications collected in the current document by journal.

The details of this study are organized as follows. In Section 2.1, the description and the mathematical formulation of the RCMPSP in its simplest form and an illustrative example are presented. Section 2.2 and Section 2.3 show the solution methods and benchmarks used to address the RCMPSP variants. Finally, Section 2.4 exposes the conclusions of this chapter.

[1]. The search was performed using the words hyphenated and non-hyphenated.

2.1 RCMPSP

We refer to the *base problem* as the most simplified RCMPSP variant, composed of a set of projects, each composed of a subset of jobs. In order to complete jobs, there is a constant number of shared renewable resources available in each time period. Moreover, each project defines a release date or arrival time, that is, the time from which the jobs can start and the desired date that represents the desired time period in which the project should be completed. Each job corresponding to the project is defined by a processing time and a given quantity required of each resource type. Some jobs belonging to the same project might establish end-start precedence relationships, i.e., a job can only start after a set of predecessor jobs are completed. A feasible schedule for the problem consists of assigning start times to each job in each project, respecting the precedence relationships between jobs, and the availability of resources at each time period. The base problem has been studied by pursuing several objective functions, with the minimization of the time at which all projects are completed or the makespan (M) being the most used one. Table 2.1 shows the existing works in the reviewed literature that address the base problem. The base problem has been studied by pursuing several objective functions (see Section 3.2), with the minimization of the time at which all projects are completed or the makespan (M) being the most used one.

Table 2.1: Research works collected in the current document that address the base problem.

Browning and Yassine (2010b); Chakrabortty et al. (2017); Deckro et al. (1991); Dumond (1992); Dumond and Mabert (1988); Gonçalves et al. (2015, 2008); Kurtulus and Davis (1982); Lawrence and Morton (1993); Linyi and Yan (2007); Lova et al. (2000); Lova and Tormos (2001); Pérez et al. (2016); Sonmez and Uysal (2015); Tsai and Chiu (1996); Tsubakitani and Deckro (1990); Van Eynde and Vanhoucke (2020); Vázquez et al. (2015); Zhang and Sun (2011); Zhuang and Yassine (2004)

2.1.1 Mathematical formulation

In this section, the mathematical model of the base problem aiming at minimizing the makespan is presented. The parameters of the problem are the following:

From AT_i, AD_i, P_{ij} and d_{ij} and assessing that there are no resource constraints, the following

I	Set of projects,
i	Project number, $i \in \{1, 2, \dots, \lvert I \rvert\}$,
N_i	Set of jobs in project i,
j	Job number, $j \in \{1, 2, \dots, \lvert N_i \rvert\}$,
AD_i	Absolute due date (i.e., hard deadline) of project i,
DD_i	Desired due date of project i,
AT_i	Arrival time of project i,
T	Set of time periods, $T = \{1, 2, \dots, max(AD_1, AD_2, \dots, AD_{\lvert I \rvert})\}$,
t	Time period, $t \in \{1, 2, \dots, \lvert T \rvert\}$,
K	Set of different resource types,
k	Resource number, $k \in \{1, 2, \dots, \lvert K \rvert\}$,
P_{ij}	Set of jobs being predecessors of job j of project i,
d_{ij}	Number of periods required to perform job j of project i,
c_{ij}^k	Consumption of resource type k required to perform job j of project i,
a^k	Availability of resource type k at every time period.

parameters, exemplified by Pritsker et al. (1969), can be calculated:

L_{ij}	Earliest possible time period in which job j of project i could be completed,
U_{ij}	Latest possible time period in which job j of project i could be completed.

The following decision variable is defined:

$$X_{ij}^t = \begin{cases} 1, & \text{if job } j \text{ of project } i \text{ is completed in time period } t, \\ 0, & \text{otherwise.} \end{cases}$$

The minimization of the makespan can be expressed as follows:

$$\text{Minimize } \; max(t \cdot X_{ij}^t),\; i \in I, j \in N_i, t \in L_{ij}, \dots, U_{ij} \tag{2.1}$$

The constraints established for the base problem are expressed as follows:

$$\sum_{t=L_{ij}}^{U_{ij}} X_{ij}^t = 1 \;, \quad \forall\, i \in I, j \in N_i \tag{2.2}$$

$$\sum_{t=L_{ij}}^{U_{ij}} t \cdot X_{ij}^t - d_{ij} \geq \sum_{t=L_{im}}^{U_{im}} t \cdot X_{im}^t \;, \quad \forall\, i \in I, j \in N_i, m \in P_{ij} \tag{2.3}$$

$$\sum_{i=1}^{|I|}\sum_{j=1}^{|N_i|}\sum_{q=t}^{t+D_{ij}-1} c_{ij}^{k}\cdot X_{ij}^{q} \leq a^{k} \ , \ \ \forall\, k \in K, t \in T \tag{2.4}$$

$$X_{ij}^{t} \in \{0,1\} \ , \ \ \forall\, i \in I, j \in N_i, t \in T \tag{2.5}$$

Constraints (2.2) ensure that each job has only one completion time. Constraints (2.3) establish the relationship between jobs, where a job cannot start its execution before the jobs of its set of predecessor jobs are completed. Constraints (2.4) ensure that only available resources are consumed at each time period. Constraints (2.5) establish the definition of the decision variables.

2.1.2 Solution example

We present an illustrative example of the base problem. There are two projects: $Project_1$, composed of the jobs Job_{11}, Job_{12}, Job_{13}, and Job_{14}; and $Project_2$, composed of the jobs Job_{21}, Job_{22}, and Job_{23}. The time corresponding to arrival time AT_i is 0 and 2 for $Project_1$ and $Project_2$, respectively. There are three types of renewable and global resources with availability $A_1 = 10$, $A_2 = 9$, and $A_3 = 11$. Table 2.2 shows the duration, predecessor jobs, and resource consumption of each job in each project.

Table 2.2: Data for the illustrative example of the base problem for two projects.

Jobs	d_{ij}	P_{ij}	c_{ij}^1	c_{ij}^2	c_{ij}^3
Job_{11}	3	-	5	-	-
Job_{12}	5	Job_{11}	5	8	-
Job_{13}	4	Job_{12}	3	-	-
Job_{14}	3	Job_{11}	-	3	7
Job_{21}	5	-	-	1	-
Job_{22}	4	Job_{21}	-	3	-
Job_{23}	4	Job_{21}	-	-	3

The graphical representation of several projects sharing global resources using the activity-on-node network can be carried out using the single- or multi-project approach. Both approaches use starting and ending dummy nodes known as *super-source* and *super-sink*. In the single-project approach, there is a starting dummy node and an ending dummy node representing the start and completion of all projects, respectively (see Figure 2.3 (a)). This representation cannot be used when each project pursues its objective function; for example, each project aims to minimize its makespan regardless of the progress of the other projects (e.g., Li et al. (2021)). On the other hand, in the multi-project approach, each project includes a starting dummy node and an ending dummy node (see Figure 2.3 (b)). This representation can be used when both individual goals and a common goal are pursued.

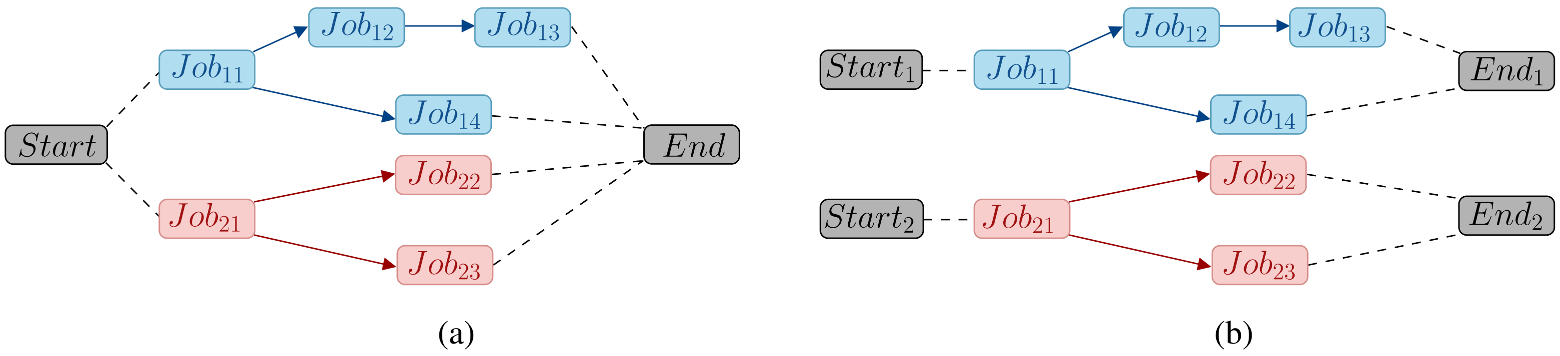

Figure 2.3: (a) Single-project approach, and (b) Multi-project approach.

In the illustrative example, Job_{12} and Job_{14} can be executed from time period 3, if only the precedence relationships are considered (see Table 2.2, Column 3). Since in the problem, there are restricted resources and the required consumption of the second resource by both jobs ($C_{12}+C_{14} = 8 + 3 = 11$) is greater than the availability of the resource in each time period ($A_2 = 9$) both jobs cannot be executed simultaneously. This situation is reflected between several jobs and even different projects; for instance, between Job_{12} and Job_{22}, where there is no established relationship of precedence. Figure 2.4 presents two feasible solutions to the problem.

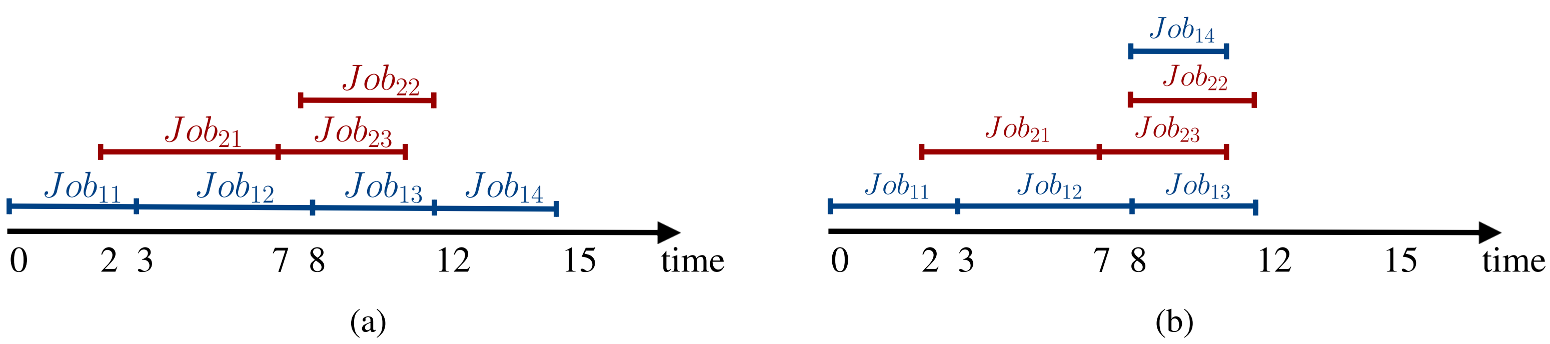

Figure 2.4: (a) Feasible solution, and (b) Optimal solution.

In Figure 2.4 (a), the completion time of Job_{14} (last job executed belonging to $Project_1$) is 15 and in Figure 2.4 (b), the completion time of Job_{13} and Job_{22} (last jobs executed belonging to $Project_1$ and $Project_2$) is 12. The solution in Figure 2.4 (b) presents the lowest possible makespan value; therefore, it represents the optimal solution for the illustrative example.

2.2 Solution methods for RCMPSP

In the beginning, the RCMPSP-related variants were solved mainly through mathematical models and priority rule-based heuristics. However, given problem complexity and the emergence of new solution techniques, the range of proposed solution methodologies was expanded.

Figure 2.5 shows the solution methods classification applied to the RCMPSP and related variants, including exact, approximate, and hybrid algorithms. The yellow and blue colors represent solution methods related to exact and approximate algorithms. The hybrid algorithms based only on approximate algorithms are represented in blue, and the hybrid algorithms supported in both categories (exact and approximate algorithms) are represented by combining both colors. Given the number of different solution methods, similar algorithms used in less than five research works are clustered. Namely, algorithms based on Lagrangian relaxation, goal programming, branch and bound, column generation, and the resolution of the mathematical model are clustered as *Linear Programming-based*. *Swarm Intelligence* contains the algorithms based on ant colony optimization, particle swarm optimization, and artificial bee colony. The research works based on memetic algorithms are clustered as genetic algorithms. Lastly, research works based on variable neighborhood search, critical chain method, robust optimization, gravitational search algorithm, learning, and specific local search heuristics are grouped in the *Other algorithms* category.

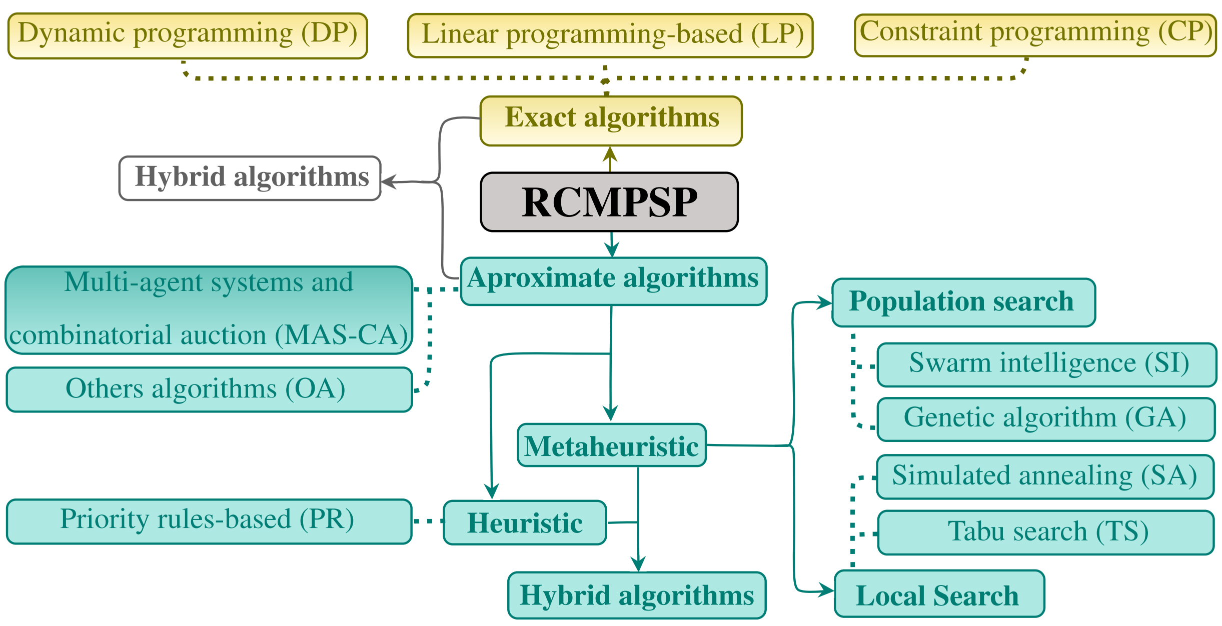

Figure 2.5: Classification of algorithms to solve the RCMPSP and existing variants.

Tables 2.3 and 2.4 show the works that used each approximate or exact algorithm to solve the problem, respectively. Note the priority rule-based group include only defined heuristics. In the collected works, the implicit use of priority rules was found in the design of the algorithms used.

Table 2.3: Reports of approximate solution methods for RCMPSP variants.

Solution methods	Research works
Genetic algorithm (GA)	Asta et al. (2016); Beşikci et al. (2015); Cai and Li (2012); Can and Ulusoy (2014); Chen and Shahandashti (2009); Fu and Zhou (2021); Gonçalves et al. (2015, 2008); Gutjahr et al. (2008); Kolisch and Heimerl (2012); Kumanan et al. (2006); Li and Xu (2018); Li and Liu (2005); Man et al. (2008); Namazian et al. (2019); Ning et al. (2012); Pérez et al. (2016); Rokou et al. (2014); Rostami and Bagherpour (2020); Satic et al. (2022); Shou et al. (2014); Sonmez and Uysal (2015); Suresh et al. (2015); Tasan and Gen (2013); Tayyar et al. (2016); Tian et al. (2018); Van Eynde and Vanhoucke (2020); Xin et al. (2018); Xu and Zhang (2012); Zheng et al. (2014); Zhuang and Yassine (2004)
Hybrid algorithm	Ahmeti and Musliu (2021); Asta et al. (2016); Cai and Li (2012); Can and Ulusoy (2014); Chakrabortty et al. (2017); Chen and Shahandashti (2009); Chen (1994); Confessore et al. (2007); Fu and Zhou (2021); Gómez et al. (2019); Gutjahr et al. (2008); He et al. (2021); Issa et al. (2021); Kolisch and Heimerl (2012); Li and Xu (2018); Liu and Xu (2020); Liu and Lu (2019); Man et al. (2008); Namazian et al. (2019); Ning et al. (2012); Rokou et al. (2014); Shou et al. (2014); Sonmez and Uysal (2015); Tian et al. (2018, 2020); Toffolo et al. (2016); Van Den Eeckhout et al. (2021); Vázquez et al. (2015); Wang et al. (2015); Xu and Feng (2014); Xu and Zhang (2012); Zheng et al. (2014); Zhu et al. (2018, 2021)

(Continued on next page)

Table 2.3 – *(Continued from previous page)*

Solution methods	Research works
Multi-agent systems and combinatorial auction (MAS-CA)	Adhau et al. (2013, 2012); Cheng et al. (2019); Confessore et al. (2007); Fu and Zhou (2021); Homberger (2007, 2012); Jedrzejowicz and Ratajczak-Ropel (2019); Li and Xu (2018); Li et al. (2021); Liu and Xu (2020); Mao et al. (2009); Shou et al. (2014); Song et al. (2018); Wang et al. (2013); Zhang and Chen (2018); Zheng et al. (2014)
Other algorithms (OA)	Ahmeti and Musliu (2021); Amirian and Sahraeian (2017); Asta et al. (2016); Cai and Li (2012); Chakrabortty et al. (2017); Chen and Askin (2009); Chiu and Tsai (2002); Cui et al. (2021); Geiger (2017); Gómez et al. (2019); Hao et al. (2010); Hu et al. (2010, 2015); Ju and Chen (2012); Kannimuthu et al. (2020); Kao et al. (2006); Kim and Schniederjans (1989); Krüger and Scholl (2009); Lee and Lei (2001); Li and Xu (2018); Liu and Lu (2019); Nabipoor et al. (2020); Namazian and Yakhchali (2016); Ning et al. (2012); Rostami and Bagherpour (2020); Satic et al. (2022); Shariatmadari et al. (2017); Sonmez and Uysal (2015); Speranza and Vercellis (1993); Tian et al. (2020); Toffolo et al. (2016); Tsubakitani and Deckro (1990); Van Den Eeckhout et al. (2021); Vázquez et al. (2015); Villafáñez et al. (2019); Wauters et al. (2015); Xu and Zhang (2012); Zheng et al. (2014); Zhu et al. (2018, 2021)

(Continued on next page)

Table 2.3 – *(Continued from previous page)*

Solution methods	Research works
Priority rules-based (PR_b)	Bock and Patterson (1990); Browning and Yassine (2010b); Chakrabortty et al. (2017); Chen et al. (2019); Chen and Ju (2015); Chen (1994); Dumond (1992); Dumond and Mabert (1988); Issa et al. (2021); Kannimuthu et al. (2020); Kolisch (2000); Kolisch and Meyer (2006); Kurtulus (1985); Kurtulus and Davis (1982); Lawrence and Morton (1993); Liu and Xu (2020); Lova and Tormos (2001); Mohanty and Siddiq (1989); Pérez et al. (2016); Pritsker et al. (1969); Rostami and Bagherpour (2020); Satic et al. (2022); Shi et al. (2019); Shou et al. (2014); Singh (2014); Suresh et al. (2015); Tian et al. (2018); Tsai and Chiu (1996); Van Eynde and Vanhoucke (2020); Vázquez et al. (2015); Wang et al. (2015, 2017); Xu and Feng (2014); Zhang and Sun (2011)
Simulated annealing (SA)	Chen and Shahandashti (2009); He et al. (2021); Joo and Chua (2017); Man et al. (2008); Sonmez and Uysal (2015)
Swarm intelligence (SI)	Chen and Ju (2015); Gómez et al. (2019); Gutjahr et al. (2008); Linyi and Yan (2007); Liu et al. (2014); Rokou et al. (2014); Tian et al. (2018); Xu and Feng (2014)
Tabu search (TS)	Ahmeti and Musliu (2021); He et al. (2021); Kolisch and Heimerl (2012); Tian et al. (2018); Zhu et al. (2018, 2021)

Table 2.4: Reports of exact solution methods for RCMPSP variants.

Solution methods	Research works
Constraint programming (CP)	Ahmeti and Musliu (2021); Hauder et al. (2020); Liu and Lu (2019)
Dynamic programming (DP)	Chen et al. (2014); Confessore et al. (2007); Satic et al. (2022); Wang et al. (2015)
Linear programming-based (LP_b)	Araujo et al. (2020); Can and Ulusoy (2014); Chen (1994); Cui et al. (2021); Davari Ardakani and Dehghani (2022); Deckro et al. (1991); Gholizadeh-Tayyar et al. (2016); Hauder et al. (2020); Issa et al. (2021); Krüger and Scholl (2010); Li et al. (2021); Mohanty and Siddiq (1989); Namazian et al. (2019); Rostami and Bagherpour (2020); Speranza and Vercellis (1993); Toffolo et al. (2016); Van Den Eeckhout et al. (2021); Vercellis (1994); Zapata et al. (2008)

Figure 2.6 (a) and (b) show the use percentages of the approximate and exact algorithms, respectively. The approximate and hybrid algorithms are the most used in the literature to solve the RCMPSP; this can be due to the high problem complexity level. Linear programming-based algorithms are the most used exact algorithms. At the same time, heuristics based on priority rules and genetic algorithms are the most representative approximate algorithms, with 24% and 22%, respectively. Algorithms based on multi-agent systems and combinatorial auctions are used in 12% of the works. In contrast, algorithms based on swarm intelligence, tabu search, and simulated annealing are found in at most 6% of the works.
To identify which are the main solution methods for solving the problem in recent years, Figure 2.7 shows, since 2010, the research carried out for the methods that have been studied in at least ten research works.

During the last decade, there was a relevant increase in the hybrid algorithms applied to the RCMPSP and related variants. To the best of our knowledge, the most used algorithms for hybridization are genetic algorithms, priority rule-based, linear programming-based, tabu search, and multi-agent systems and combinatorial auctions. Priority rule-based heuristics and genetic

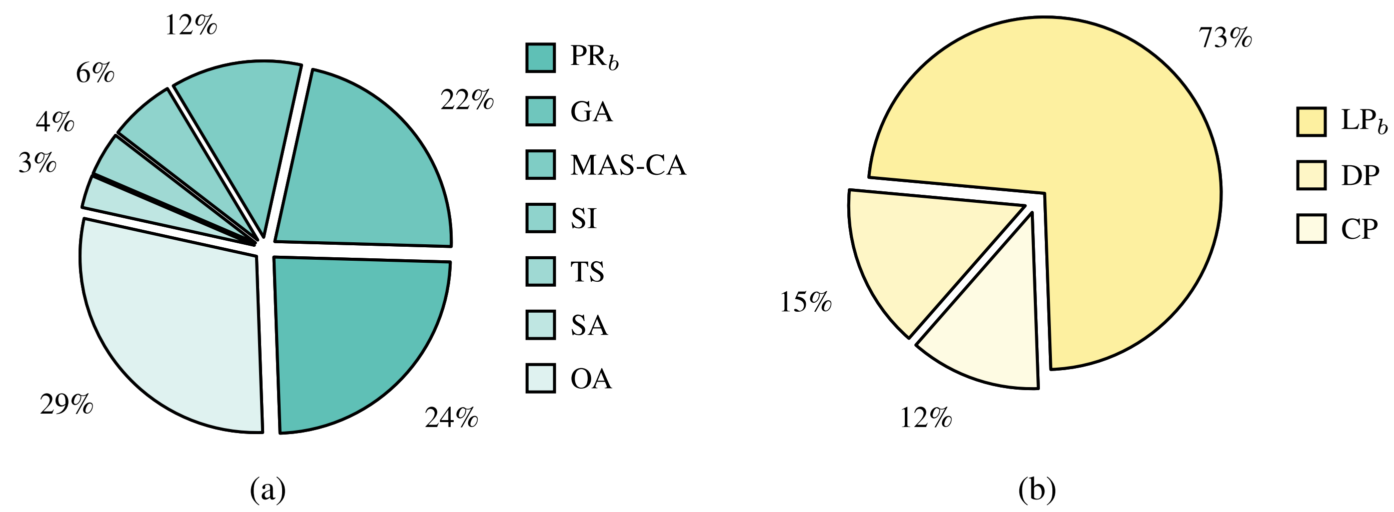

Figure 2.6: Percentage of different solution methods applied to the RCMPSP.

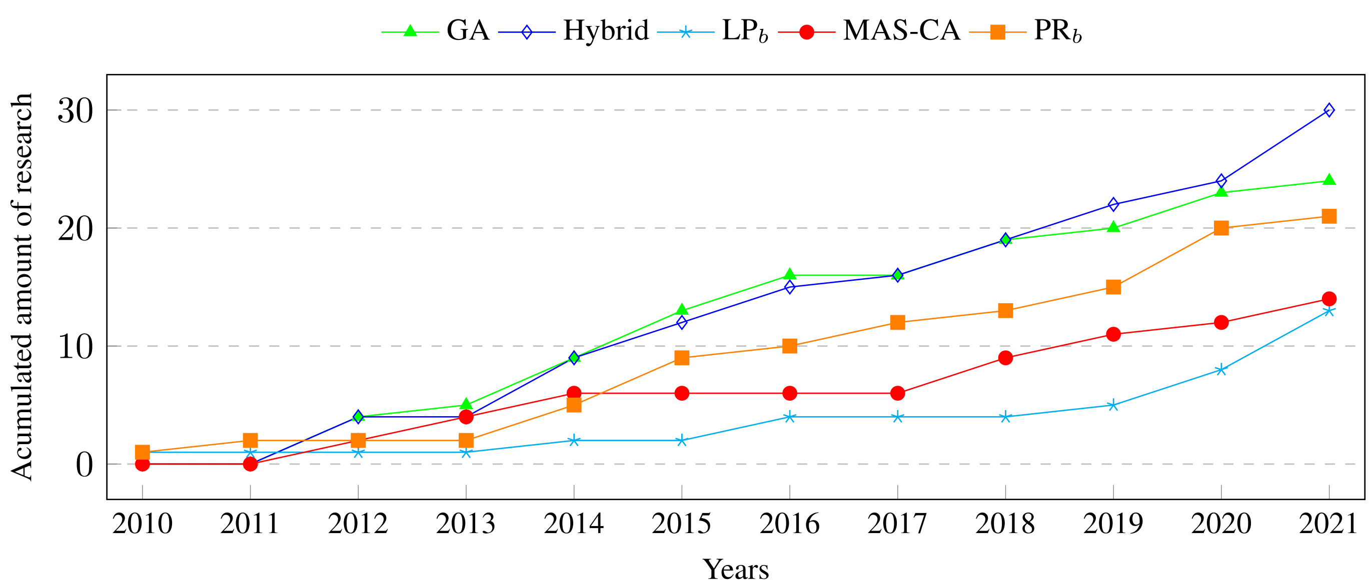

Figure 2.7: Main solution methods applied since 2010.

algorithms are the most used algorithms for solving the problem since the beginning, presenting a stable growth since 2010 and being the most used in the last five years. Moreover, the use of multi-agent systems and combinatorial auctions showed significant growth in the last five years; this can be a consequence of studying different features of the problem. Finally, algorithms based on linear programming showed an important growth in the last two years.

2.3 Benchmarks for RCMPSP

At the beginning of studying the RCMPSP, there were no instance sets to evaluate the performance of the proposed algorithms and most works were validated using problem examples. This practice is still used in recent research, mainly due to the large number of features proposed and the lack of instance sets for each resulting problem variant. In addition, several works evaluate the models and algorithms through own generated instances for the RCMPSP variant addressed.

Kolisch and Sprecher (1997) proposed a project scheduling problem library (PSPLib)[2] for the study of the RCPSP. After it, many works on the RCMPSP used that benchmark in their numerical experiments by combining two or more single-project instances, adding the required aspects to cover the studied features, and generating their specific multi-project instances. Moreover, several works were carried out on real instances, demonstrating the importance of this problem within the industry sector. The first and most recognized set of instances is the multi-project scheduling problem library (MPSPLib)[3], presented by Homberger (2007) specifically for decentralized RCMPSP with local resources. MPSPLib contains 80 generated instances from the combination of 2, 5, 10, or 20 problems (projects) of the PSPLib composed of 30, 90, or 120 jobs, assigning an arrival time to each project and allowing the use of resources globally and locally. Subsequently, this library was expanded by Homberger (2012) adding 60 new instances that present differences in terms of project access to local resources. MPSPLib allows the online evaluation of proposed solutions and provides various objective functions. Browning and Yassine (2010a) present the first multi-network problem generator (MNPG)[4], also used by He et al. (2021). The authors generated 12320 random instances of the problem, but the best values obtained for each possible instance are unknown. Using instances of the MPSPLib and MNPG, a benchmark set known as the resource-constrained multi-project scheduling problem library (RCMPSPLIB)[5] was created by Pérez and Posada and used by Pérez et al. (2016) and Vázquez et al. (2015). Another variant that has a well-defined set of instances is the multi-mode RCMPSP. This set was developed for the MISTA-2013[6] challenge and is composed of project groups taken from the PSPLib by adding multiple execution modes.

Table 2.5 shows the research in the reviewed literature that uses each instance set. Note that only the instance sets present in at least five works are listed, and some investigations perform validation with several types of instances.

[2]http://www.om-db.wi.tum.de/psplib/main.html

[3]http://www.mpsplib.com

[4]http://sbuweb.tcu.edu/tbrowning/RCMPSPinstances.htm

[5]http://www.eii.uva.es/elena/RCMPSPLIB.htm

[6]https://gent.cs.kuleuven.be/mista2013challenge/index.html

Table 2.5: Reports of research works on different types of instances.

Instances	Research works
Generated instances	Amirian and Sahraeian (2017); Araujo et al. (2020); Beşikci et al. (2015); Cai and Li (2012); Can and Ulusoy (2014); Chakrabortty et al. (2017); Chen et al. (2019); Chen and Askin (2009); Chen and Ju (2015); Cheng et al. (2019); Chiu and Tsai (2002); Confessore et al. (2007); Davari Ardakani and Dehghani (2022); Dumond (1992); Dumond and Mabert (1988); Gonçalves et al. (2015, 2008); Hao et al. (2010); Hauder et al. (2020); Hu et al. (2015); Issa et al. (2021); Ju and Chen (2012); Kannimuthu et al. (2020); Kolisch (2000); Kolisch and Heimerl (2012); Kolisch and Meyer (2006); Krüger and Scholl (2009); Krüger and Scholl (2010); Kurtulus (1985); Kurtulus and Davis (1982); Lawrence and Morton (1993); Li et al. (2021); Linyi and Yan (2007); Liu and Lu (2019); Lova et al. (2000); Lova and Tormos (2001); Mao et al. (2009); Nabipoor et al. (2020); Namazian et al. (2019); Pérez et al. (2016); Rokou et al. (2014); Rostami and Bagherpour (2020); Satic et al. (2022); Shariatmadari et al. (2017); Shou et al. (2014); Sonmez and Uysal (2015); Speranza and Vercellis (1993); Suresh et al. (2015); Tasan and Gen (2013); Tian et al. (2020); Tsai and Chiu (1996); Tsubakitani and Deckro (1990); Van Den Eeckhout et al. (2021); Van Eynde and Vanhoucke (2020); Vercellis (1994); Wang et al. (2015, 2017); Zhu et al. (2018, 2021)
Problem examples	Bock and Patterson (1990); Chen et al. (2014); Deckro et al. (1991); Hu et al. (2010); Kao et al. (2006); Kim and Schniederjans (1989); Kumanan et al. (2006); Lee and Lei (2001); Li and Liu (2005); Liu et al. (2014); Man et al. (2008); Mohanty and Siddiq (1989); Namazian and Yakhchali (2016); Ning et al. (2012); Pritsker et al. (1969); Shi et al. (2019); Singh (2014); Zapata et al. (2008); Zhang and Sun (2011); Zhuang and Yassine (2004)

(Continued on next page)

Table 2.5 – *(Continued from previous page)*

Instances	Research works
Real-world instances	Chen and Shahandashti (2009); Chen (1994); Cui et al. (2021); Gholizadeh-Tayyar et al. (2016); Gutjahr et al. (2008); Joo and Chua (2017); Kannimuthu et al. (2020); Kolisch and Meyer (2006); Song et al. (2018); Sonmez and Uysal (2015); Speranza and Vercellis (1993); Tayyar et al. (2016); Tian et al. (2018, 2020); Vercellis (1994); Xin et al. (2018); Xu and Feng (2014); Xu and Zhang (2012); Zhang and Chen (2018)
MPSPLib	Adhau et al. (2013, 2012); Fu and Zhou (2021); Gómez et al. (2019); Homberger (2007, 2012); Jedrzejowicz and Ratajczak-Ropel (2019); Li and Xu (2018); Liu and Xu (2020); Pérez et al. (2016); Villafáñez et al. (2019); Wauters et al. (2015); Zheng et al. (2014)
MISTA-2013	Ahmeti and Musliu (2021); Asta et al. (2016); Geiger (2017); Toffolo et al. (2016); Wauters et al. (2016)

Figure 2.8 shows the percentages of works using the different instance sets described above. In the category *Other instances*, the works that used the MISTA-2013, MNPG, and RCMPSPLIB instance sets were grouped.

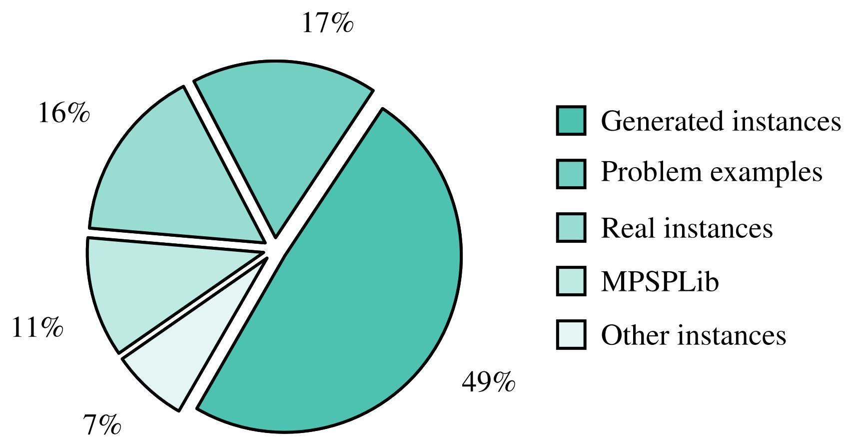

Figure 2.8: Percentage of research on a different set of instances.

The highest percentage is represented by the generated sets in the different works. It is important to note that 16% of the research was conducted using real-world based instances. Regarding the instance sets presented in the literature, the MPSPLib is the most used benchmark. The increased use of this library may be due to the frequent study of different features of the problem.

2.4 Conclusions

In this chapter, we study the literature related to the RCMPSP and its variants, founding that the problem has been widely studied, presenting a considerable attention increase in the past and current decade. Regarding the algorithms used to solve the problem, we found a wide range of solution methods, being the approximate and hybrid algorithms the most used; this can be due to the high problem complexity level. At the same time, heuristics based on priority rules are the most representative approximate algorithms, and linear programming-based algorithms are the most used exact algorithms.

On the other hand, we find that most of the investigations validate the proposed algorithms through instances generated for the problem under study and that several investigations were conducted using real-world based instances. Furthermore, the most used benchmark in the literature is the MPSPLib; this can result from the systematic study of different problem features.

Chapter 3

Taxonomy

This section presents a summary of the RCMPSP variants studied in the literature. Given the nature of the problem and collected papers, variants are analyzed in terms of features and objective functions, presenting classifications for both aspects. These classifications are based on 34 problem features and 20 objective functions identified based on which the proposed taxonomies were formed. An analysis and time perspective of both aspects is also presented, including the connection of the problem with the practice.

This chapter is organized as follows. Section 3.1 exposes the essential features concerning RCMPSP variants. Section 3.2 shows a classification scheme and discussion of the most used objective functions for RCMPSP variants. Section 3.3 presents an analysis and discussion about the previous features and objective function, as well as those works that address real-world instances. Finally, Section 3.4 exposes the conclusions of this chapter.

3.1 Problem features

In order to classify the features found in the studied manuscripts, four main components are defined: jobs, relationships, projects, and resources. Figure 3.1 shows a taxonomy based on all RCMPSP variants proposed. The colors blue, purple, green, and red represent features related to jobs, relationships, projects, and resources, respectively.

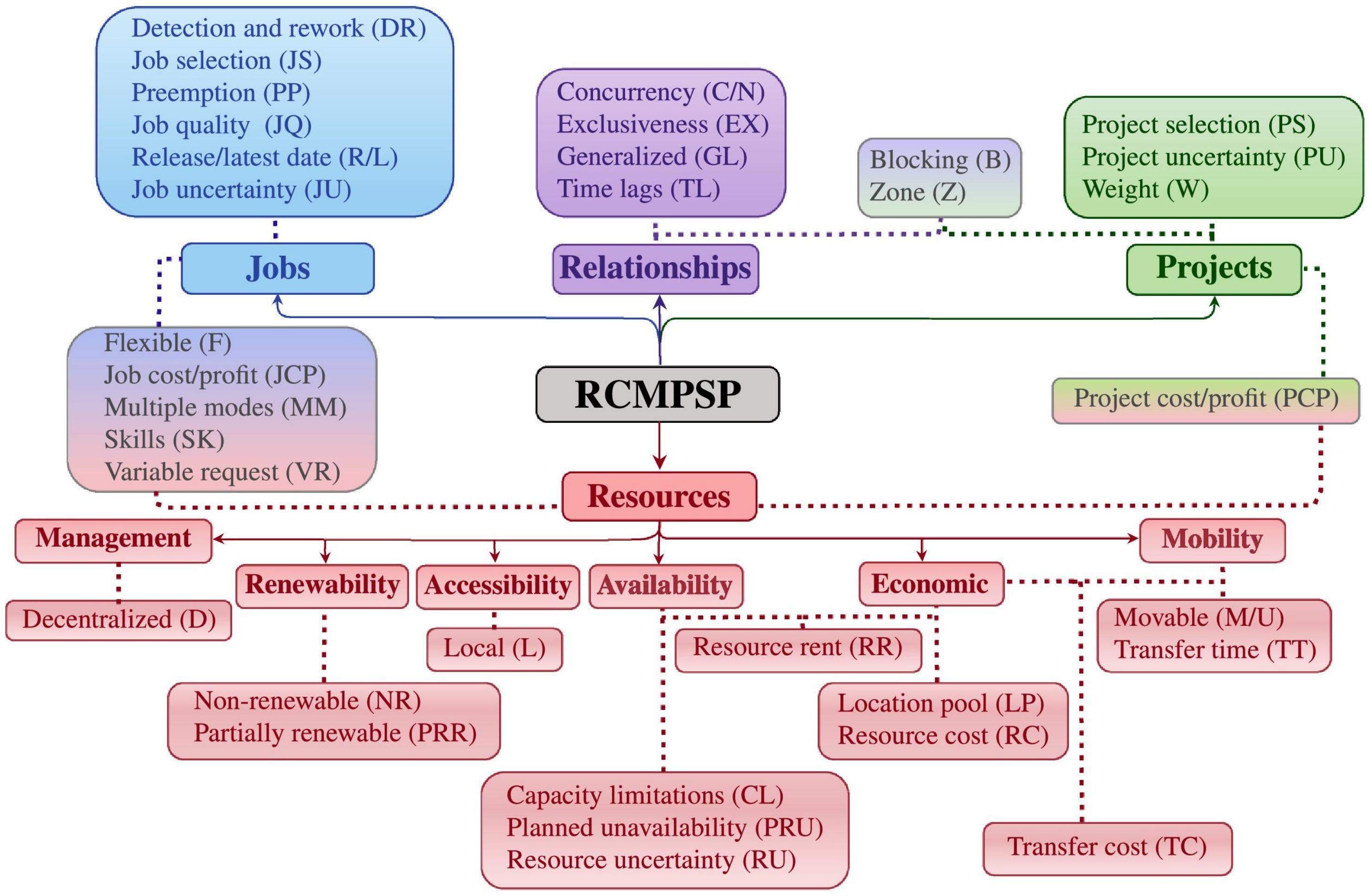

Figure 3.1: Feature taxonomy: classification of the RCMPSP variants.

3.1.1 Job- and resource-based features

One of the essential conditions for starting a job execution is the availability of the required resources. The existence of multiple execution modes (MM) is one of the well-known related features, where a different duration and resource consumption denote each possible job execution mode. Generally, a higher resource consumption implies shorter duration. There may be a set of jobs that need to be executed in the same mode, but commonly the job execution modes are independent. As an illustrative example, consider that cleaning an area takes 30 minutes if only one machine is used and 15 minutes if two machines are used. A particular case of multiple execution modes involves the use of the workforce as a resource in project planning, where each person/worker presents skills or efficiency levels to perform one or more jobs (SK). When all associated parameters are known, e.g., duration and resource consumption of each job when performed by each worker, the situation can be modeled as multiple execution modes. On the other hand, in some cases, the mentioned parameters must be calculated depending on the efficiency or skill level the person/worker has to perform each job and other aspects such as resource availability or required quality. Considering multiple skills or efficiency levels could imply changes in the job duration, resource consumption, or job completion quality.

A similar feature to multiple execution modes is the flexible resource allocation to a job (F) studied by Issa et al. (2021) and Rokou et al. (2014). This involves that executing a certain job requires a total resource amount, i.e., workload, so the resource allocation in each time period is not required following a uniform distribution. In this sense, for each job, a set of modes can be automatically defined, composed of the possible combinations of duration and resource required. Each defined mode for each job must comply with the required resource amount. Closely related to the previously described features, Davari Ardakani and Dehghani (2022) and Kurtulus (1985) analyzed the variable resource requirement (VR). In this case, the job resource requirements vary at each time period but are initially defined; for example, to start executing a job in a machine, two workers are required, but once the execution process is stabilized, only one worker is required to complete the job execution.

In a production process, cash or liquidity can be seen as a resource, so the start of job execution represents a cash outflow, e.g., service payments, while the completion of job execution represents a cash inflow, e.g., the sale of products. This job cost/profit (JCP) was studied in several research works and when considered in a problem, maximizing the net present value is the objective function most frequently considered.

Table 3.1 shows the existing research in the reviewed literature that addresses job- and resource-based features. Note that only features considered in at least five works are listed; the other cases are referenced in this section.

3.1.2 Job-based features

In the base problem, the jobs follow certain primary conditions, for example, deterministic duration and continuous execution. These aspects often move the problem away from real production processes where these features can be modified.

Continuous job execution involves a few outdated production procedures. In this sense, the concept of preemption (PP) arises when job execution interruption is allowed due to job preferences. Job execution splitting can lead to more stable resource use by realizing consumption in periods where resources could remain idle. This feature can be accompanied by costs associated with stopping and resuming the job execution, which can lead, for example, to an analysis of cost versus completion time. The handling of uncertainty always brings a complexity increase in any problem resolution. Despite this, several works studied the existing uncertainty on job

Table 3.1: Research works that address job- and resource-based features.

Features	Research works
Job cost/profit (JCP)	Chiu and Tsai (2002); He et al. (2021); Kolisch and Meyer (2006); Xu and Feng (2014); Zapata et al. (2008)
Multiples modes (MM)	Ahmeti and Musliu (2021); Araujo et al. (2020); Asta et al. (2016); Beşikci et al. (2015); Can and Ulusoy (2014); Chen et al. (2014); Chen and Ju (2015); Cheng et al. (2019); Davari Ardakani and Dehghani (2022); Geiger (2017); Joo and Chua (2017); Ju and Chen (2012); Kannimuthu et al. (2020); Kolisch and Meyer (2006); Kumanan et al. (2006); Pritsker et al. (1969); Speranza and Vercellis (1993); Toffolo et al. (2016); Vercellis (1994); Wauters et al. (2016); Xu and Feng (2014); Zapata et al. (2008); Zhang and Chen (2018)
Skills (SK)	Chen et al. (2014); Cui et al. (2021); Gutjahr et al. (2008); Kolisch and Heimerl (2012); Namazian and Yakhchali (2016)

duration (JU) since unexpected events affecting job duration might occur in real processes.

Job release and/or latest date (R/L) studied by Joo and Chua (2017) and Hao et al. (2010) can have a relevant influence on the schedule. This feature implies that it is impossible to start the job execution before the established release date or finish it after the established latest date. In the literature this concept is also known as time constraints or *start-no-earlier-than* and *start-no-later-than*. When problem conditions do not impose the job release date, the value is calculated based on the predecessors and duration of each job as well as the arrival time and absolute due date of the corresponding project.

As part of the study of multiple execution modes, Kannimuthu et al. (2020) and Xu and Feng (2014) studied job quality (JQ), considering different quality levels for each execution mode. This concept can lead, for example, to an analysis of quality versus completion time or quality versus revenue when additional payments are established depending on the job quality. The quality can be determined depending on the project environment; for example, in construction projects, the quality can be calculated based on the construction quality assessment system (CONQUAS) (BCA, 2017). Detection/inspection and reworks (DR) was also proposed as a job-related variant in the research works presented by Zhu et al. (2021), Zhu et al. (2018), and Bock and Patterson (1990), where, after a project is finished, an error detection process is carried out and, faulty jobs are corrected. The existence of uncertainty when this feature is studied is implicit; therefore, it is not described as an independent feature.

The existence of projects that have a set of jobs whereof only one of them is executed was studied by Hauder et al. (2020) under the concept of alternative activities. In the RCPSP literature, it is related to concepts of logical dependencies, or job selection (JS), the latter being the one used in this research work. The jobs can differ in several characteristics, such as duration, resource consumption, quality, and precedence relationships. When studying the problem under different execution modes for a job, the execution modes of each job can be seen as job selection, where the difference between each pair of jobs is based on the duration and resource requirements. Depending on other characteristics that can be added to the problem, this feature allows, for example, the analysis of the use of resources versus completion time.

Table 3.2 shows the existing research in the reviewed literature that addresses job-based features. Note that only features considered in at least five works are listed, the other cases are referenced in this section.

Table 3.2: Research works that address job-based features.

Features	Research works
Preemption (PP)	Bock and Patterson (1990); Hao et al. (2010); Issa et al. (2021); Joo and Chua (2017); Pritsker et al. (1969)
Job uncertainty (JU)	Chen et al. (2019); Hu et al. (2015); Kim and Schniederjans (1989); Liu and Xu (2020); Mao et al. (2009); Nabipoor et al. (2020); Namazian et al. (2019); Satic et al. (2022); Song et al. (2018); Tian et al. (2020); Wang et al. (2017); Xu and Feng (2014); Xu and Zhang (2012); Zapata et al. (2008)

3.1.3 Resource-based features

Resource allocation is a significant aspect of the RCMPSP that adds a strong degree of complexity. Efficient use and availability of resources are a growing concern in society. As a consequence, there are several resource-based features, so eight sub-classifications are defined (i.e., Management, Renewability, Accessibility, Availability, Mobility, Economic, Availability and Economic, Economic, and Mobility), and 13 features are shown in Figure 3.1.

• *Management*: In the RCMPSP, resource management can be centralized or decentralized. In the first case, only one general decision-maker manages the resources knowing the current status of each project and pursuing a global objective. In the other case, when the resources are managed decentrally (D), the allocation of global resources is carried out by a general decision-maker who pursues a global objective but might not fully know the status of each project. Once this allocation is made, the decision-makers of each project are responsible for assigning execution start times to each job, complying with the resources assigned by the general decision-maker, and pursuing a local objective. Following Homberger (2007), each decision-maker plans exactly one project and is considered "honest" when providing information regarding the planning of its related project.

• *Renewability*: In management processes, there are different types of renewal of existing resources, e.g., renewable, non-renewable, or partially renewable resources. Renewable resources present a fixed existence that is renewed in each time period. Non-renewable resources (NR) arise from the need to optimize resources that present a unique existence for the entire time horizon so that once they are assigned to a job, they are not renewed, e.g., a unique budget. The study of non-renewable resources is closely related to multiple execution modes, given the different consumption of resources. Another related feature, studied by Amirian and Sahraeian (2017), refers to the existence of partially renewable resources (PRR), where the resources are renewed in a set period of time, e.g., a budget renewed several times within the time horizon.

• *Accessibility*: The different access to existing resources is a widely studied feature in the multi-project environment. The accessibility can be distinguished between global and local resources. The global resources can be used for the execution of all jobs of all projects. Otherwise, local resources (L) are defined as resources belonging to a project that can be consumed exclusively by the jobs belonging to the project, e.g., workers of a company branch office. In the papers studied, the resources are defined as only one category; that is, the access to a resource cannot be local and at the same time global. For instance, Fu and Zhou (2021) define local resources as fixed facilities and unskilled workers and global resources as specialized equipment and skilled professionals. Both accessibility options were also studied in relation to resource renewability. For example, Ahmeti and Musliu (2021), Asta et al. (2016), and Wauters et al. (2016) studied local renewable and non-renewable resources, and global non-renewable resources. However, research presented by Geiger (2017) assumes that local resources are only non-renewable and global resources are considered only renewable. On the other hand, although local resources can be studied when working centrally, the beginning of their study associated with the RCMPSP is closely linked to decentralized management, where the decision-makers of each project are responsible for the management of its resources.

• *Availability*: The fixed resource availability at each time period has been considered since the beginning of the RCMPSPs. Studying different conditions where there may be uncertainty in the resource existence or even the non-existence of resources results in problem variants to cope with real situations in production lines. Uncertainty in resource existence (RU) moves the problem to a stochastic area, increasing resolution complexity, especially in large production processes. This aspect was studied by Hu et al. (2015), Wang et al. (2015), Mao et al. (2009), and Zapata et al. (2008). Similarly, Tian et al. (2018) studied resource unavailability planning (RUP). This feature provides flexibility to the resource availability, keeping the problem in a deterministic environment. In this case, unavailability planning refers to periods where there is no specific resource, e.g., break times for machines or workers. On the other hand, when the resources are transported to a project to be used later, the resource availability can be affected by capacity limitations (CL), for example, related to transportation and inventory. In the same way, the job completion can generate products or even waste that needs to be stored and transported later. This feature was studied by Gholizadeh-Tayyar et al. (2016) and Tayyar et al. (2016) and is related to the concept of cumulative resources in the RCPSP.

• *Mobility*: This aspect defines the need to move some resources between jobs to carry out the executions. Mobility can arise between jobs of the same project and different projects. The transfer times (TT) involve all required movements for starting the job execution and can be used to model setup times, e.g., the time it takes to move a resource from one project to another for the execution process or the time it takes to move a job to the area where it can be executed. Transfer times can have a considerable influence when preemption is allowed. In the same way, Tian et al. (2018) considered that the resources could be movable or un-movable (M/U). In the movable case, the jobs are kept in an area, and the resources are transferred to jobs. In the un-movable case, the jobs must move to the resources to receive the requested service, e.g., big machines already fixed in a given space.

• *Economic*: The economic aspect is also reflected in some features found in the literature. The resource cost (RC) is a feature that several works address. Each resource is associated with a cost. That can lead, for example, to an analysis of cost versus completion time, especially when multiple execution modes are considered. In addition, the resource cost is also addressed when overtime is considered as a way of increasing production capacity. The costs in regular hours are usually lower than in overtime, which can lead, for example, to a trade-off analysis of cost for overtime versus cost for the delay. Another cost associated with resources can be inventory costs due to storage. Rostami and Bagherpour (2020) studied the need to designate a resource pool for doing periodic services to projects (LP), establishing that the location of the resource pool is not set. Instead, there are potential locations where the facility for the resource pool can be constructed. Given this, it is required to determine the appropriate location to construct the facility while incurring the lowest possible costs. The construction costs of the facility may vary depending on the selected location and the transportation costs of the resources from the facility to the different projects.

• *Mobility and economic*: As a consequence of the resource mobility, Fu and Zhou (2021), Suresh et al. (2015), Adhau et al. (2013), and Krüger and Scholl (2010), consider that a transfer cost (TC) is also generated due to physically moving resources. Similarly, this cost can be used when coping with transfers between jobs of the same project, between different projects, or even between the resource pool and each project.

• *Availability and economic*: The restricted existence of renewable resources can cause timeouts for a job execution start, generating delays in a project completion and costs for delays. As a consequence Van Den Eeckhout et al. (2021), Gholizadeh-Tayyar et al. (2016), Tayyar et al. (2016), and Kolisch and Heimerl (2012) considered the resource rent (RR).

Table 3.3 shows the existing research in the reviewed literature that addresses resource-based features. Note that only features considered in at least five works are listed, the other cases are referenced in this section.

Table 3.3: Research works that address resource-based features.

Features	Research works
Decentralized (D)	Adhau et al. (2013, 2012); Cheng et al. (2019); Confessore et al. (2007); Fu and Zhou (2021); Homberger (2007, 2012); Jedrzejowicz and Ratajczak-Ropel (2019); Li and Xu (2018); Li et al. (2021); Liu and Xu (2020); Mao et al. (2009); Rostami and Bagherpour (2020); Shi et al. (2019); Wang et al. (2013); Zhang and Chen (2018); Zheng et al. (2014)
Local (L)	Adhau et al. (2013, 2012); Ahmeti and Musliu (2021); Asta et al. (2016); Can and Ulusoy (2014); Confessore et al. (2007); Fu and Zhou (2021); Geiger (2017); Gómez et al. (2019); He et al. (2021); Homberger (2007, 2012); Jedrzejowicz and Ratajczak-Ropel (2019); Li and Xu (2018); Rostami and Bagherpour (2020); Shi et al. (2019); Tian et al. (2020); Villafáñez et al. (2019); Wang et al. (2013); Wauters et al. (2016, 2015); Zhang and Chen (2018); Zheng et al. (2014)
Non-renewable (NR)	Ahmeti and Musliu (2021); Araujo et al. (2020); Asta et al. (2016); Beşikci et al. (2015); Can and Ulusoy (2014); Chen (1994); Cheng et al. (2019); Davari Ardakani and Dehghani (2022); Geiger (2017); Gholizadeh-Tayyar et al. (2016); Krüger and Scholl (2010); Lee and Lei (2001); Li et al. (2021); Liu et al. (2014); Rokou et al. (2014); Shariatmadari et al. (2017); Speranza and Vercellis (1993); Tasan and Gen (2013); Tayyar et al. (2016); Toffolo et al. (2016); Wauters et al. (2016); Xu and Feng (2014); Zhang and Chen (2018)
Transfer time (TT)	Adhau et al. (2013); Cai and Li (2012); Fu and Zhou (2021); Krüger and Scholl (2009); Krüger and Scholl (2010); Liu and Lu (2019); Suresh et al. (2015); Van Den Eeckhout et al. (2021)

(Continued on next page)

Table 3.3 – *(Continued from previous page)*

Features	Research works
Resource cost (RC)	Beşikci et al. (2015); Can and Ulusoy (2014); Chen et al. (2014); Chen (1994); Gholizadeh-Tayyar et al. (2016); Kannimuthu et al. (2020); Kim and Schniederjans (1989); Kolisch (2000); Kolisch and Heimerl (2012); Lee and Lei (2001); Li et al. (2021); Liu et al. (2014); Mao et al. (2009); Ning et al. (2012); Shariatmadari et al. (2017); Speranza and Vercellis (1993); Suresh et al. (2015); Tayyar et al. (2016); Van Den Eeckhout et al. (2021); Vercellis (1994); Wang et al. (2013); Zhang and Chen (2018)

3.1.4 Relationship-based features

In a job schedule environment, there might be many relationships between jobs. The basic relationship between jobs in the RCMPSP is the end-start relationship, which establishes that a job execution cannot start unless all its predecessor jobs have been fully executed.

Tian et al. (2018), and Pritsker et al. (1969) studied the concurrent or non-concurrent jobs execution (C/N). The concurrency can be established through sets of jobs that must be executed simultaneously, and each job must belong exactly to one set. On the other hand, in the non-concurrency relationship, the sets of jobs represent jobs that cannot be executed in parallel. This relationship might be useful in the case that conflicts between jobs need to be incorporated. In addition, there may be jobs that need to be executed exclusively, e.g., a fumigation job in space requires the stoppage of the rest of the jobs required by the project. This kind of relationship was studied by Hao et al. (2010) under the concept of exclusiveness (EX) and can generate delays in the production line.

The joint consideration of basic relationships (i.e., start-start, start-end, end-start, and end-end) was considered in the generalized variant (GL) in the research of Gholizadeh-Tayyar et al. (2016), Tayyar et al. (2016), and Chen and Shahandashti (2009). Start-start establishes that the execution of a job cannot be started until the execution of the related jobs is started. End-end refers to the fact that a job cannot finish its execution until the related jobs are fully executed. Start-end implies that a job cannot finish its execution until the execution of the related jobs is started. As a feature related, time lags (TL) studied by Davari Ardakani and Dehghani (2022), Joo and Chua (2017), Gholizadeh-Tayyar et al. (2016), and Tayyar et al. (2016) establish a minimum or maximum elapse time between start and completion of the execution of the related jobs.

3.1.5 Project-based features

The simultaneous execution of several projects is one of the main RCMPSP features that increase complexity compared to the single project execution. In the base problem, the simultaneous execution of several projects assumed there are no preferences on any project, and existing projects, respective arrival times to the production line, and due dates are known.

In several environments, preferences or priorities on a project are needed to model daily operations (e.g., a shop that accepts special orders or a hospital emergency room). This feature known as weight (W), is closely related to preemption and can result in a wide range of analyzes, for example, related to costs and time. One of the most studied analyzes is related to penalties incurred when a project is delayed, given the direct proportionality between weight and penalty. This weight can be defined in diverse ways. For example, Nabipoor et al. (2020) establish an explicit degree of weight for each project, and Kannimuthu et al. (2020) assign a different delay penalty for each project. In both cases, this priority or penalty is multiplied by the delay time of each project. In addition, Tian et al. (2020) assign different priorities for each project related to the use of resources, so in a conflict, resources are assigned to the project with the highest priority. In all cases, the objective is to prioritize the execution of the jobs corresponding to the most critical projects. On the other hand, a service company can receive a new project associated with economic and social improvements, so it could be relevant to accept the project even if it is not planned. This situation captures the uncertainty regarding project arrival (PU) and permits addressing different situations. For example, Satic et al. (2022) establish that all projects have a probability of arrival, whereas Chen et al. (2019) present initial projects with an established arrival date and a set of new arrival projects. In the latter case, at a random period of time, if the number of projects in execution is less than the maximum number allowed, a project is randomly selected from the new arrival set to start its execution. Furthermore, new projects may present different priority levels when compared to existing or already running projects. For example, Wang et al. (2015) establish that the new projects, called urgent projects, have a higher priority regarding using resources. Similar to the job selection concept, project selection (PS) refers to the existence of a set of projects from which to decide the projects to be executed.

Existing projects can differ in the number of jobs, execution modes, resource consumption, and costs associated with resources or delays, among other features. The selected projects can be related to deadlines, budget constraints, and maximum or the minimum number of projects that can be selected or executed simultaneously. Tasan and Gen (2013) present the project selection through a graph network containing subgraphs. Each subgraph comprises a set of alternative projects, establishing disjunctive (OR relations) or conjunctive (AND relations) relationships.

Table 3.4 shows the existing research in the reviewed literature that addresses project-based features. Note that only features considered in at least five works are listed, the other cases are referenced in this section.

Table 3.4: Research works that address project-based features.

Features	Research works
Project selection (PS)	Amirian and Sahraeian (2017); Chen and Askin (2009); Davari Ardakani and Dehghani (2022); Gutjahr et al. (2008); Kolisch and Meyer (2006); Namazian et al. (2019); Namazian and Yakhchali (2016); Shariatmadari et al. (2017); Shou et al. (2014); Tasan and Gen (2013)
Project uncertainty (PU)	Chen et al. (2019); Hu et al. (2015); Kao et al. (2006); Satic et al. (2022); Wang et al. (2015)
Weight (W)	Adhau et al. (2013, 2012); Beşikci et al. (2015); Chen (1994); Cheng et al. (2019); Chiu and Tsai (2002); Hu et al. (2010); Kannimuthu et al. (2020); Kim and Schniederjans (1989); Kolisch (2000); Lee and Lei (2001); Li and Xu (2018); Li and Liu (2005); Liu and Xu (2020); Liu and Lu (2019); Liu et al. (2014); Man et al. (2008); Mao et al. (2009); Mohanty and Siddiq (1989); Nabipoor et al. (2020); Satic et al. (2022); Singh (2014); Speranza and Vercellis (1993); Tian et al. (2020); Vercellis (1994); Wang et al. (2013, 2015); Xin et al. (2018); Xu and Feng (2014); Xu and Zhang (2012); Zhang and Chen (2018)

3.1.6 Project- and relationship-based features

The precedence relationships have been mainly studied between jobs belonging to the same project. However, Ning et al. (2012) studied relationships between projects, i.e., zone and blocked area. The concurrency relationship mentioned above in Section 3.1.4 can be extended to jobs from different projects under the concept of zone (Z). A set of jobs belonging to different projects must be executed concurrently, maintaining the same duration but executing a common resource consumption. For example, pieces from different projects must be cooled, and a single freezer is used. On the other hand, in a period, a specific reference project can operate as a point of reference for transportation of the resources required by neighboring projects. This set of projects constitutes a blocked area (B), which cannot be accessed during this period. Consequently, executing any job of any project in the blocked area is impossible.

3.1.7 Project- and resource-based features

Similar to job cost/profit, the project cost/profit (PCP) is studied in a multi-project environment with budget considerations. Chen (1994) established an average maintenance cost for each project in each time period. That cost together with others (e.g., the cost of using resources) cannot exceed the available budget in each time period. The completion of a project can also represent an inflow of cash that may (or may not) depend on the time period at which the project ends. For example, Suresh et al. (2015) establish that the obtained benefit is independent of the time period in which each project ends, but it directly influences the cash flow. Whereas Shou et al. (2014) establish a different profit for the completion of the project at each time period. On the other hand, Can and Ulusoy (2014) simultaneously consider that the start and completion of a project represent an outflow and inflow of cash. The study of this feature is closely related to the selection of projects, and in the works where it is considered, maximizing the net present value is the most frequently pursued objective function.

Table 3.5 shows the existing research in the reviewed literature that addresses the project cost/profit feature.

Table 3.5: Research works that address project cost/profit (PCP) feature.

Amirian and Sahraeian (2017); Can and Ulusoy (2014); Chen and Askin (2009); Chen (1994); Chiu and Tsai (2002); Davari Ardakani and Dehghani (2022); Gutjahr et al. (2008); Kim and Schniederjans (1989); Namazian et al. (2019); Namazian and Yakhchali (2016); Shariatmadari et al. (2017); Shou et al. (2014); Speranza and Vercellis (1993); Suresh et al. (2015); Vercellis (1994)

3.2 Objective functions

The objective functions are classified based on three main factors: resources, projects, and time. Figure 3.2 shows the classification scheme proposed. The colors red, green, and orange represent objective functions based on resources, projects, and time, respectively.

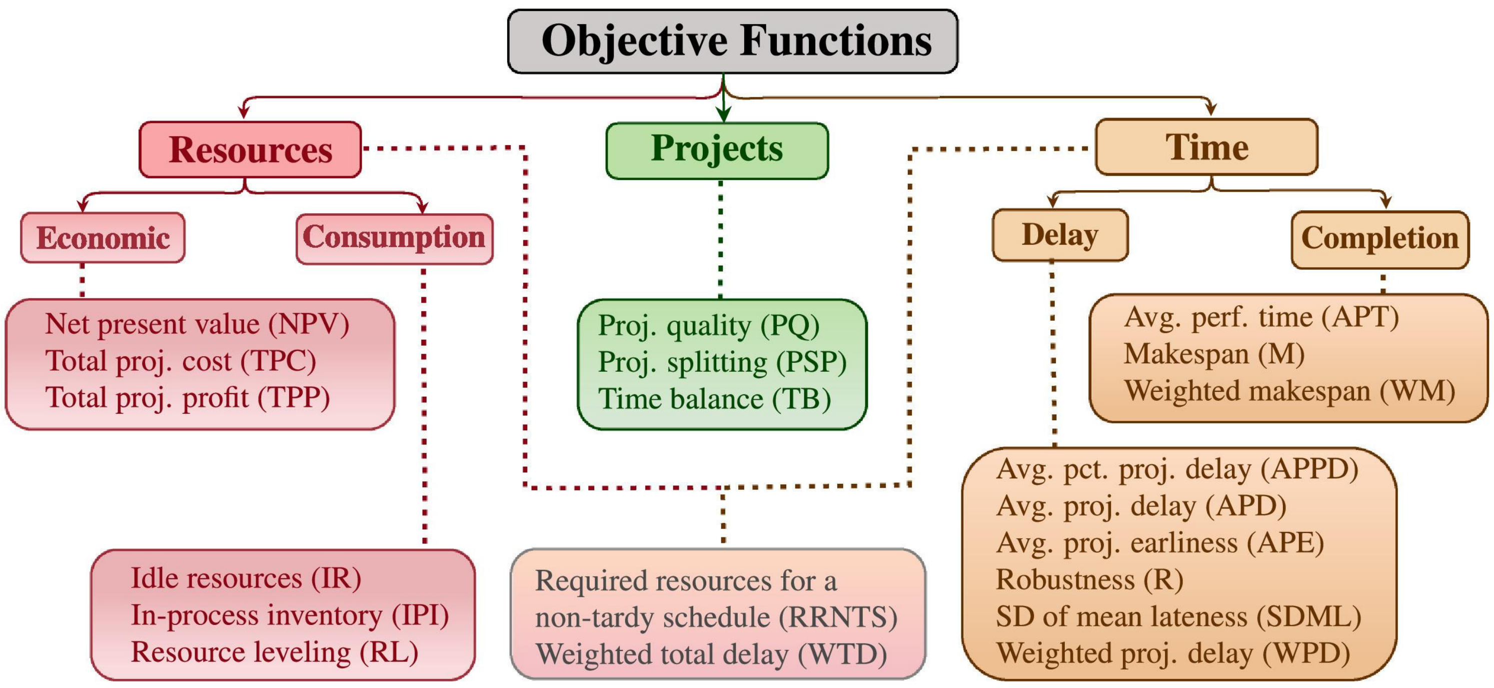

Figure 3.2: Classification of the RCMPSP objective functions.

3.2.1 Resource-based objective functions

The previous section discussed several resource-based features. They led to the emergence of several objective functions aimed at efficient resource use and cost evaluation.

• *Economic*: The valuation of economic aspects is fundamental in project scheduling environments. When assessing discontinued cash flow caused by cost-related features, it may be required to control cash stock to avoid situations where the production process is stopped due to a lack of cash. Because of this, the net present value maximization (NPV) was proposed. As a particular case, Zapata et al. (2008) studied the maximization of the expected net present value by evaluating uncertainty. Regarding evaluating resource costs, the total project cost minimization (TPC), also called the sum of costs, considers the costs related to the use, transfer, inactivity, rent, reserving, and maintenance of the resources. In particular cases, Chen (1994) addressed the cost of the critical project where only the critical projects are valued, Kolisch and Heimerl (2012) studied the labor costs influenced by different skills and efficiency of human personnel, and Rostami and Bagherpour (2020) addressed the sum of construction costs related to resource facility location. Finally, obtaining profits by completing a job and/or a project resulted in the objective of maximizing the total project profit (TPP) that can be evaluated at the end of one or all projects. This objective function is closely related to job and/or project selection features.

• *Consumption*: Resource inactivity occurs when resources are available but not required by any job; this can be due to precedence constraints. This aspect has been studied under the same conditions through two objective functions, i.e., the total resource idle time minimization and the idle resources minimization (IR). Since both measures address the same objective, we refer to this objective function as idle resource minimization. When studying resources in project scheduling, it is essential to assess factors associated with the resource inventory. For instance, resource inventory limitations can cause the execution of jobs corresponding to a project not to be consecutive. This results in additional expenses due to projects waiting for the execution of the related jobs being part of the inventory. Hence, it is required to minimize the number of in-process inventory jobs (IPI), an objective function studied by Lova et al. (2000) and Kim and Schniederjans (1989). On the other hand, stable resource consumption is one of the priorities present in the objective functions that seek to avoid peaks in resource usage. This objective was initially named resource average load factor minimization. However, at the beginning of this century, its study began under the name of resource-leveling maximization (RL), a term used in this work.

Table 3.6 shows the existing research addressing resource-based objective functions. Note that only the objective functions considered in at least five works are listed, the other cases are referenced in this section.

Table 3.6: Research works that address resource-based objective function.

Objective function	Research works
Economics	
Net present value (NPV)	Can and Ulusoy (2014); Chiu and Tsai (2002); Davari Ardakani and Dehghani (2022); He et al. (2021); Speranza and Vercellis (1993); Suresh et al. (2015); Vercellis (1994); Zapata et al. (2008)
Total proj. profit (TPP)	Amirian and Sahraeian (2017); Chen and Askin (2009); Gutjahr et al. (2008); Kolisch and Meyer (2006); Namazian et al. (2019); Satic et al. (2022); Shariatmadari et al. (2017); Shou et al. (2014); Wang et al. (2013)
Total proj. cost (TPC)	Adhau et al. (2013); Amirian and Sahraeian (2017); Chen et al. (2014); Chen (1994); Cheng et al. (2019); Fu and Zhou (2021); Gholizadeh-Tayyar et al. (2016); Kannimuthu et al. (2020); Kolisch and Heimerl (2012); Krüger and Scholl (2010); Liu and Lu (2019); Liu et al. (2014); Mao et al. (2009); Ning et al. (2012); Rostami and Bagherpour (2020); Singh (2014); Tayyar et al. (2016); Van Den Eeckhout et al. (2021); Wang et al. (2013); Xu and Feng (2014); Zhang and Chen (2018)
Consumption	
Idle resources (IR)	Amirian and Sahraeian (2017); Hauder et al. (2020); Krüger and Scholl (2010); Lova et al. (2000); Mohanty and Siddiq (1989); Shi et al. (2019)
Resource leveling (RL)	Davari Ardakani and Dehghani (2022); Issa et al. (2021); Kim and Schniederjans (1989); Liu and Lu (2019); Lova et al. (2000); Mao et al. (2009); Tsubakitani and Deckro (1990)

3.2.2 Time-based objective functions

The following classification groups those objective functions that evaluate the completion time or the delays that may occur in the completion of projects.

• *Delays*: Project delays occur when the completion time is larger than the due date. The project delay is defined as the difference between the makespan and the desired due date if the makespan is greater than the desired due date; otherwise, the delay is zero. There are several objective functions associated with the delays. The average project delay minimization (APD) is the most used objective function regarding this aspect. Considering that the number of projects is a parameter of the problem, in this review, the objective functions based on the total delays are classified as average project delay (e.g., total project delay). The standard deviation allows for evaluating the dispersion of a data set, so Dumond (1992) and Dumond and Mabert (1988) studied as an objective the standard deviation mean lateness minimization (SDML) to balance the delays incurred in each project. Other measures based on delays and efficiency of the generated schedules were studied under average percent project delay, efficiency, and robustness. In this review, the measures related to efficiency, either in average or total values, are classified as minimization of the average percent project delay (APPD) if the job duration is deterministic; and maximization of robustness (R) in the cases of the research presented by Zhu et al. (2021), Chen et al. (2019), and Wang et al. (2017). On the other hand, different weights for each project greatly influence schedule generation. Weights can be associated with penalties for delays; the higher the weight, the higher the delay penalty. Given that, the weighted project delay minimization (WPD) arises as an objective function. Gonçalves et al. (2015), Krüger and Scholl (2010), Gonçalves et al. (2008), and Bock and Patterson (1990) pursue the average project earliness minimization (APE) considering that the early completion sometimes negatively influences the project delivery, so it can also be penalized with the same or a different value used for delays.

• *Completion*: The project completion time or makespan (M) is defined as the time when all jobs associated with a project are fully executed. Performance time, also known as flow time or throughput time, is defined as the difference between the first job execution start time of a project and a project makespan. In a single project environment, the performance time minimization (APT) is equivalent to the makespan minimization. This situation does not happen in a multi-project environment where the makespan is defined as the maximum makespan among all projects and the average performance time as the average performance times of all projects. In this sense, the average performance time minimization attempts to conduct projects as soon as possible. Since the number of projects is a fixed value, minimizing the total performance time is classified as the minimization of the average performance time. Weights are used to minimize the weighted makespan (WM) where the makespan value is multiplied by the corresponding weight, so the higher the weight, the greater the influence on the objective function.

Table 3.7 shows the existing research in the reviewed literature that addresses the time-based objective functions. Note that only the objective functions considered in at least five works are listed, the other cases are referenced in this section.

Table 3.7: Research works that address time-based objective function.

Objective function	Research works
Delay	
Avg. pct. proj. delay (APPD)	Adhau et al. (2013); Browning and Yassine (2010b); Cai and Li (2012); Chakrabortty et al. (2017); Chen et al. (2019); Krüger and Scholl (2009); Lova and Tormos (2001); Mohanty and Siddiq (1989); Pérez et al. (2016); Van Eynde and Vanhoucke (2020); Vázquez et al. (2015); Wang et al. (2017)
Avg. proj. delay (APD)	Adhau et al. (2012); Ahmeti and Musliu (2021); Araujo et al. (2020); Asta et al. (2016); Bock and Patterson (1990); Browning and Yassine (2010b); Chakrabortty et al. (2017); Dumond (1992); Dumond and Mabert (1988); Geiger (2017); Gómez et al. (2019); Gonçalves et al. (2015, 2008); Homberger (2012); Jedrzejowicz and Ratajczak-Ropel (2019); Joo and Chua (2017); Krüger and Scholl (2009); Lova and Tormos (2001); Mohanty and Siddiq (1989); Shi et al. (2019); Singh (2014); Toffolo et al. (2016); Tsai and Chiu (1996); Wauters et al. (2016, 2015); Zheng et al. (2014)

(Continued on next page)

Table 3.7 – *(Continued from previous page)*

Objective function	Research works
Weighted proj. delay (WPD)	Beşikci et al. (2015); Cheng et al. (2019); Ju and Chen (2012); Kim and Schniederjans (1989); Kolisch (2000); Krüger and Scholl (2010); Lawrence and Morton (1993); Lee and Lei (2001); Li and Xu (2018); Li et al. (2021); Liu and Xu (2020); Nabipoor et al. (2020); Pritsker et al. (1969); Tasan and Gen (2013); Tian et al. (2020); Wang et al. (2015); Xu and Zhang (2012); Zhu et al. (2018)
Completion	
Avg. perf. time (APT)	Bock and Patterson (1990); Chen et al. (2014); Deckro et al. (1991); Dumond (1992); Dumond and Mabert (1988); Gonçalves et al. (2015, 2008); Pritsker et al. (1969)

(Continued on next page)

Table 3.7 – *(Continued from previous page)*

Objective function	Research works
Makespan (M)	Ahmeti and Musliu (2021); Asta et al. (2016); Can and Ulusoy (2014); Chen et al. (2014); Chen and Shahandashti (2009); Chen and Ju (2015); Chen (1994); Confessore et al. (2007); Cui et al. (2021); Davari Ardakani and Dehghani (2022); Geiger (2017); Gómez et al. (2019); Hauder et al. (2020); Homberger (2007, 2012); Hu et al. (2015); Issa et al. (2021); Jedrzejowicz and Ratajczak-Ropel (2019); Ju and Chen (2012); Kannimuthu et al. (2020); Kao et al. (2006); Kumanan et al. (2006); Lee and Lei (2001); Li et al. (2021); Linyi and Yan (2007); Namazian et al. (2019); Namazian and Yakhchali (2016); Ning et al. (2012); Pérez et al. (2016); Pritsker et al. (1969); Rokou et al. (2014); Satic et al. (2022); Singh (2014); Song et al. (2018); Sonmez and Uysal (2015); Speranza and Vercellis (1993); Tasan and Gen (2013); Tian et al. (2018, 2020); Toffolo et al. (2016); Vázquez et al. (2015); Villafáñez et al. (2019); Wauters et al. (2016, 2015); Xu and Zhang (2012); Zhang and Chen (2018); Zhu et al. (2021); Zhuang and Yassine (2004)
Weighted makespan (WM)	Hu et al. (2010); Li and Liu (2005); Man et al. (2008); Xin et al. (2018); Xu and Feng (2014); Zhu et al. (2018)

3.2.3 Resource- and time-based objective functions

Project completion time and resource use are two of the most studied factors in the literature. In the previous sections, the related objective functions were defined individually. Nevertheless, in the sequel, we discuss the joint consideration of both objective types.

The weight or cost associated with resources motivated the weighted total delay minimization (WTD) objective. That objective function was used by Mohanty and Siddiq (1989) to minimize the sum of total resources demanded by each project multiplied by the corresponding weighting and delay. Studying multiple execution modes or job selection can result in schedules with variable resource use. In this sense, Lee and Lei (2001) studied the resources use, prioritizing on-time project completion, an objective known as the minimization of required resources for a non-tardy schedule (RRNTS).

3.2.4 Project-based objective functions

In Section 3.1, it can be seen that the project-based features were not extensively studied, whereby there is not a wide range of objective functions that contain project-related aspects.

In the context of continuous project execution, Lova et al. (2000) propose the project splitting minimization (PSP) with the aim of avoiding additional efforts related to project control. The incorporation of multiple execution modes, jobs selection, and/or quality resulted in several objective functions, such as the project quality maximization (PQ), given by the quality of the average jobs, studied by Kannimuthu et al. (2020) and Xu and Feng (2014). Also influenced by the mentioned features, Hauder et al. (2020) studied the time balance maximization (TB) pursuing the lowest difference between the minimum and maximum duration of the selected jobs.

3.2.5 Multiple objectives

The wide range of RCMPSP-related features causes that, in many studies, the problem is solved considering several objectives. Note that this does not necessarily imply multi-objective functions approaches but to propose more than one independently objective for a given variant.

In the collected literature, many works pursued different objective functions analyzing a trade-off, for example, Cheng et al. (2019); Hauder et al. (2020); Issa et al. (2021); Jedrzejowicz and Ratajczak-Ropel (2019) and Chakrabortty et al. (2017). In contrast, several studies presented multi-objective functions, and some divided the problem into phases and pursued different objective functions in each one, for example, Amirian and Sahraeian (2017); Davari Ardakani and Dehghani (2022); Kannimuthu et al. (2020); Namazian et al. (2019) and Asta et al. (2016).

3.3 Analysis and time perspective

This section presents an analysis of the research on the different features and objective functions related to the RCMPSP variants in the reviewed literature. Also, we discuss how the reviewed investigations have been assessing the gap between theory and practice concerning the RCMPSP and related variants, and those works that solved real-world-based cases are analyzed.

3.3.1 Research on the problem features

In the reviewed literature in this survey, 82% of the works consider the base problem linked to different features. Significant differences exist between the number of works that study resource- and relationship-based features, 52% and 6%, respectively. In the study of resource-based features, the existence of local and non-renewable resources are the most representative, with 20% each. Concerning research addressing projects- and jobs-based features, the difference between the usage percentages is 19%, being projects-based features the most significant at 39%. The most studied projects- and jobs-based features are weight and uncertainty, with 27% and 12%, respectively.

The problem evolution is depicted in Figure 3.3. The discussed features are associated with the year in which they were presented. The bars indicate the amount of research carried out from that moment, and the colors follow the pattern used in Figure 3.1.

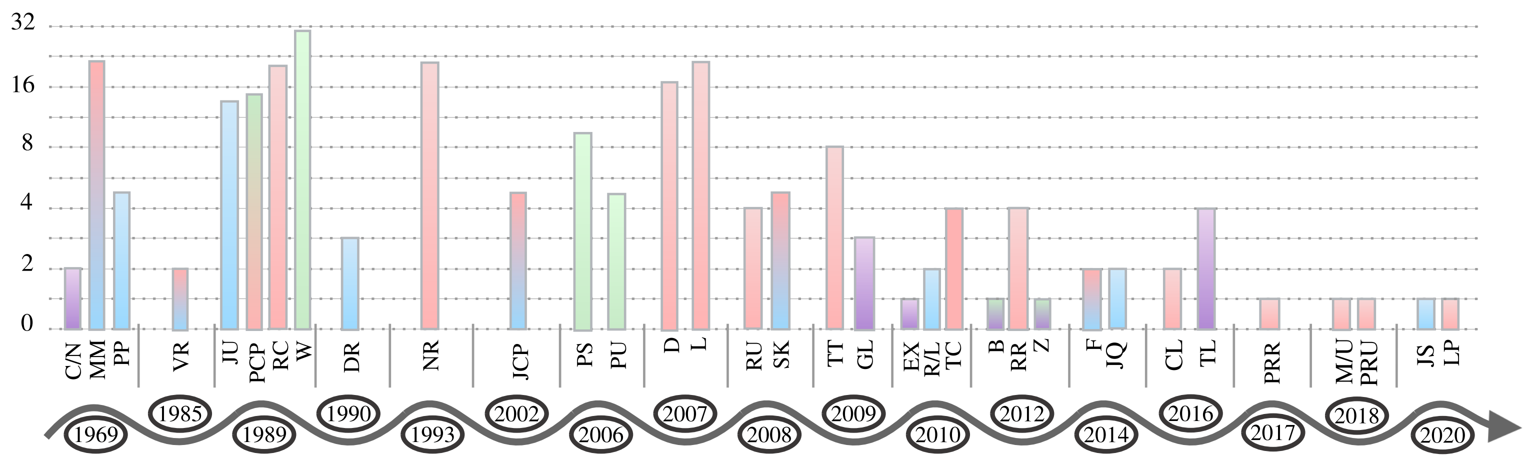

Figure 3.3: Evolution of the problem over the years.

To analyze relevant research topics related to the problem in recent years, Figure 3.4 shows, since 2010, how the research was carried out concerning problem features studied in at least ten research works.

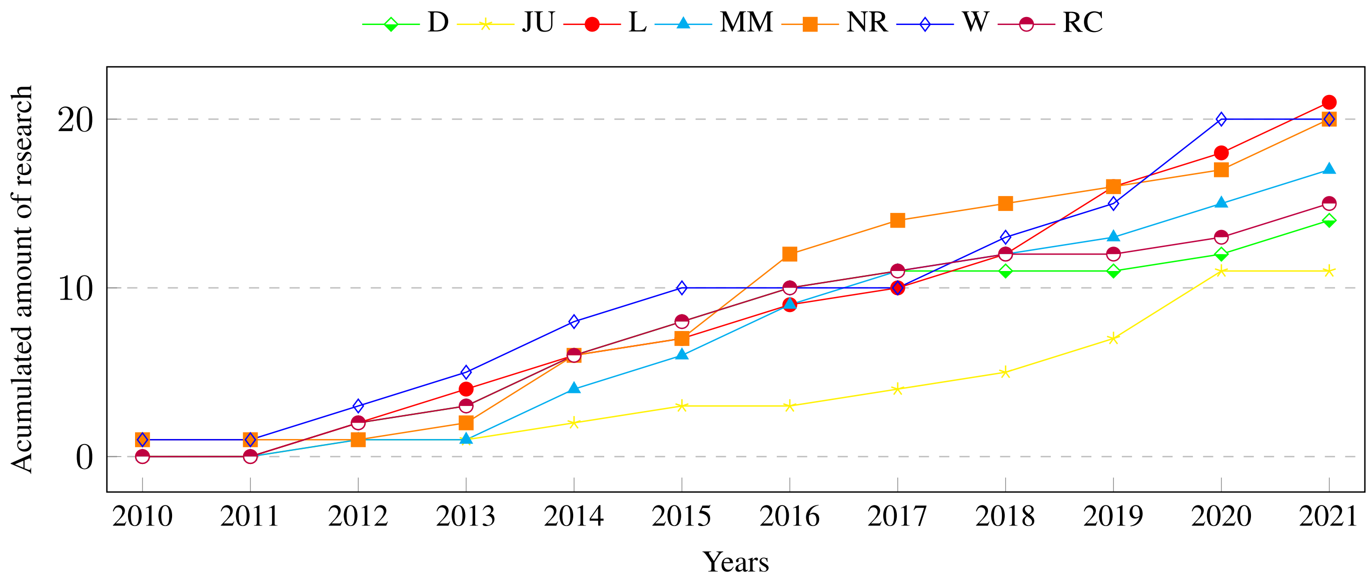

Figure 3.4: Main problem features studied since 2010.

The study of new features associated with the problem exhibited slow growth until the beginning of this century. Since 2010, its growth has increased, resulting in more than 34% of the features associated with the problem. This may be related to the increase in research carried out since 2010. Note that there are features that, after being presented, were not considered again, some were recently presented, and others are associated with features less studied in the literature, as shown in Figure 3.3.

Figures 3.3 and 3.4 show that the most representative features are the use of local and non-renewable resources, the existence of weights associated with projects, multiple execution modes, resource costs, decentralized resource allocation, and job duration uncertainty. In many cases, the combination of local resources and decentralized resource allocation is addressed. This finds its rationale bearing in mind the relationship between these features and the rise of decentralized production models. In the same way, the study of non-renewable resources is closely related to multiple execution modes. The use of non-renewable resources was added to the problem in 1993, presenting 23 related research works, 87% of them in the last decade, representing one of the features with the most significant increase in related research in this period. The existence of weight associated with projects is the most reflected feature in the literature, with notable growth since 2010, presenting 65% of related research since this date. Regarding the uncertainty, this feature generates a higher degree of complexity to the problem but is required to study given its impact on the schedule generation. The study of this feature related to job duration increased, presenting 79% of investigations in the 2010-2021 period, with a significant increase of 57% in the last five years. A large number of research works can also be observed that addressed cost considerations, whether related to resources, jobs, or projects.

These features bring the problem closer to real environments where fundamental elements are costs/profit and budgets. It is important to highlight that, as seen in the figures, researching the base problem is still relevant, despite having been presented more than 50 years ago.

3.3.2 Research on the objective functions

Figure 3.5 (a) shows the works related percentage to the different classifications proposed. For a more in-depth review, Figure 3.5 (b) and (c) show the percentage of works that consider time-based and resource-based objective functions, respectively. The time-based objective functions studied in less than ten research works are grouped in the category *Others*.

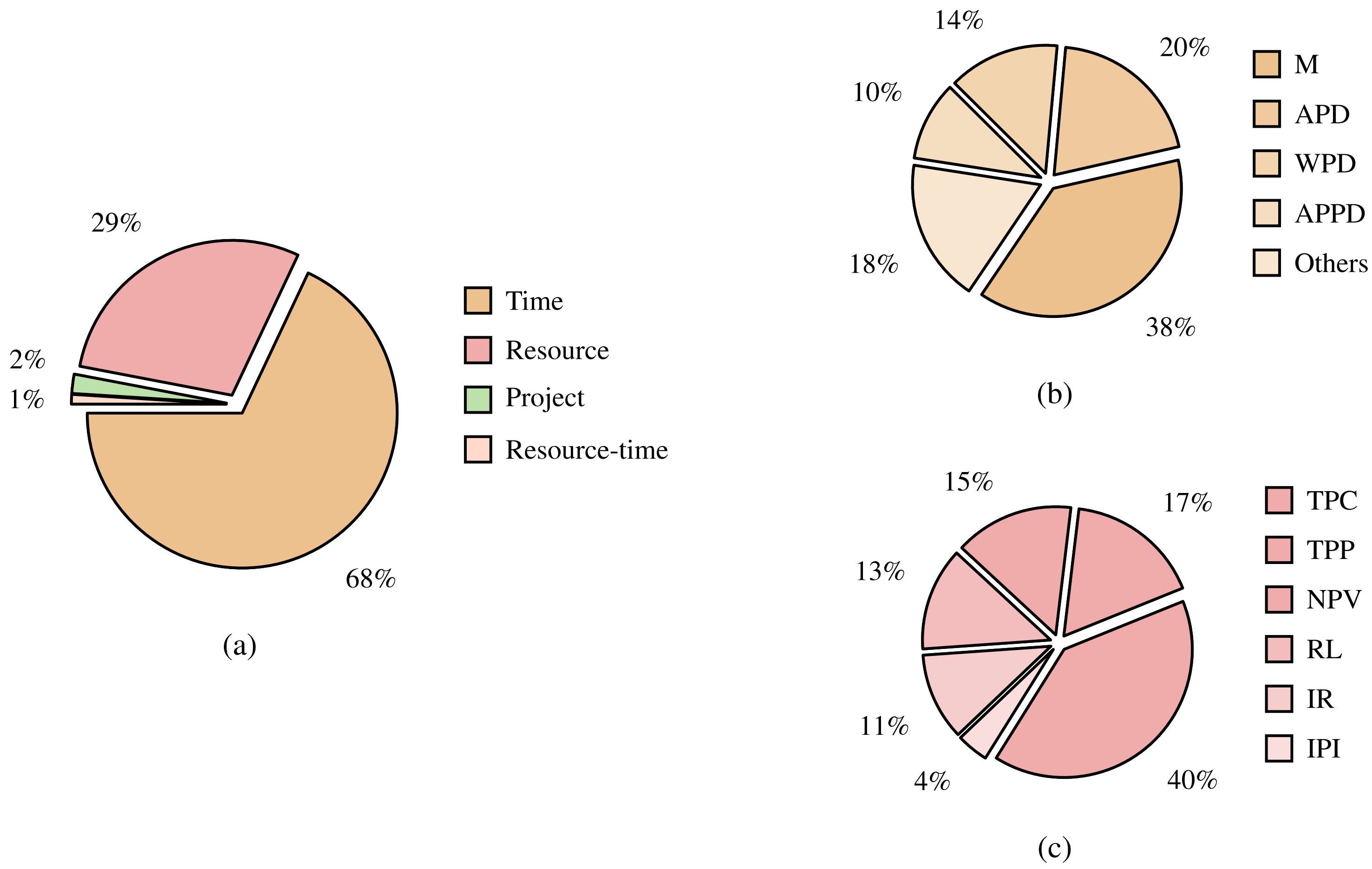

Figure 3.5: Percentage of research by classification of objective function studied.

As can be observed, the time-based objective functions are the most studied, followed by resource-based. On the other hand, project-based objective functions represent a small portion. The most studied objective functions are the makespan minimization and the average project delay minimization, corresponding to completion and delay sub-classifications. The total project cost minimization is the next most pursued objective, followed by the minimization of the weighted project delay and the average percentage of project delay.

The first objective function in the multi-project environment presented in 1969 by Pritsker et al. (1969) was related to time optimization. Despite the elapsed years until today, and although the most studied features are related to resources, the percentages distribution indicates that time optimization is the main factor in the problems related to the RCMPSP.

3.3.3 Connection to practice

With the aim to analyze those works that solved real-world-based cases, Table 3.8 shows the research developed on them. Column 2 shows the application domain. Columns 3-5 show the dimensions of the scenarios handled by each case in terms of the maximum number of projects, the maximum number of jobs, and the maximum number of different renewable resources considered. Following the categorization provided in Sections 3.1 and 3.2, columns 6-13 and 14-34 report the objective function and features, respectively. Moreover, to analyze the different objective functions and features applied in real environments, Figure 3.6 shows the percentage of works for (a) objective functions and (b) problem features.

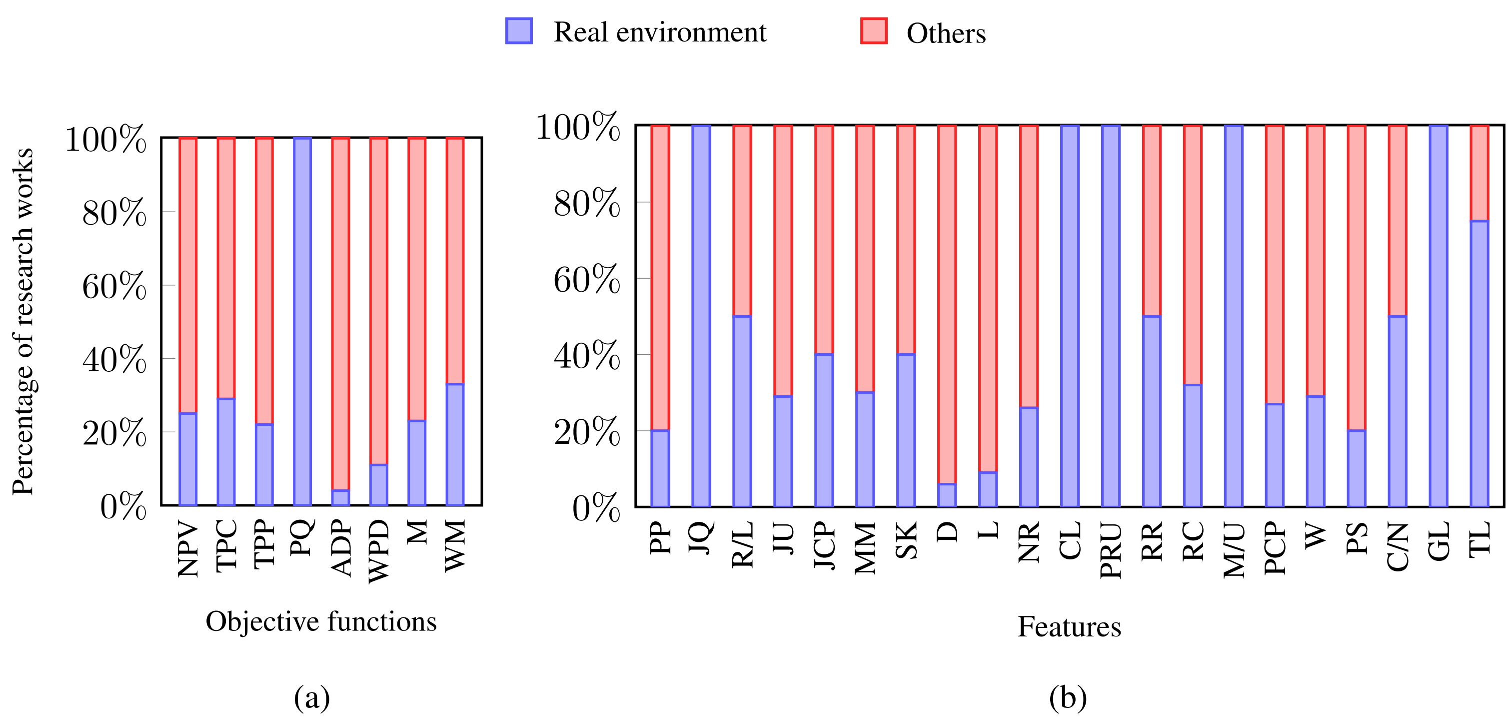

Figure 3.6: Relationship between the literature and the real environments collected in this survey.

The applications in the construction field are the most representative ones. The maximum number of projects per instance and jobs per project in most works do not exceed 18 and 46, respectively. These values, especially the number of jobs per project, are smaller than the similar benchmark values proposed in the literature. Note there are works, for example, Kolisch and Meyer (2006), that generated a set of instances based on the data collected in the company where the research was conducted. On the other hand, in several works, the number of different renewable global resources exceeds the values found in the generated instance sets.

The most used objectives are related to the time category, followed by resource-based objective functions. Regarding the described features, features associated with jobs-resources, resources, and project categories stand out. In this sense, the use of multiple execution modes is the most studied job- and resource-based feature. Moreover, in studying resource-based features, non-renewable resources and the costs associated with resource use are prominent features. Lastly, the feature involving weights is the most used project-based feature.

Based on the reported works, we can observe that only a part of the objective functions found in the literature is pursued, where the makespan minimization and the total project cost minimization are the most used ones. It is worth indicating that the objective function related to project quality maximization was only used in real environments. The remaining objective functions do not exceed the 35% utilization in real problems. On the other hand, the consideration of weights, multiple execution modes, resource cost, and non-renewable resources are features addressed in many works, although they do not exceed 30% of the corresponding investigations carried out compared to the overall available literature. Considering job quality, the features: capacity limitations, planned unavailability, movable and unmovable resources, and generalized precedence relations are studied in a few works but always considered in real environments.

Table 3.8: Research works applied to real environments. Note (*) represents the average value.

	Instance				Objective functions								Features																				
					Rsrc.			Proj.	Time				Jobs				Jobs-rsrc.			Rsrc.							Proj.-rsrc.	Proj.		Relp.			
Research	Domain	# proj.	# jobs	# rsrc.	NPV	TPC	TPP	PQ	APD	WPD	M	WM	PP	QJ	R/L	JU	JCP	MM	SK	D	L	NR	CL	PRU	RR	RC	M/U	PCP	W	PS	C/N	GL	TL
Cui et al. (2021)	Manufacturing	4	288*	-							•								•														
Kannimuthu et al. (2020)	Construction	3	32	22		•		•			•			•				•								•			•				
Tian et al. (2020)	Aviation industry	3	6	1						•	•					•					•								•				
Song et al. (2018)	Production	5	7	6							•					•																	
Tian et al. (2018)	Undefined	12	30	27							•													•			•				•		
Xin et al. (2018)	Heath Care	3	8	3								•																	•				
Zhang and Chen (2018)	Construction	6	20	5		•					•							•		•	•	•				•			•				
Joo and Chua (2017)	Undefined	3	≥ 135*	≥ 20					•				•		•			•															•

(Continued on next page)

Table 3.8 – *(Continued from previous page)*

	Instance				Objective functions								Features																				
					Rsrc.			Proj.	Time				Jobs				Jobs-rsrc.			Rsrc.								Proj.-rsrc.	Proj.		Relp.		
Research	Domain	# proj.	# jobs	# rsrc.	NPV	TPC	TPP	PQ	APD	WPD	M	WM	PP	QJ	R/L	JU	JCP	MM	SK	D	L	NR	CL	PRU	RR	RC	M/U	PCP	W	PS	C/N	GL	TL
Gholizadeh-Tayyar et al. (2016)	Construction	2	14	3		•																•	•		•	•						•	•
Tayyar et al. (2016)	Construction	2	14	3		•																•	•		•	•						•	•
Sonmez and Uysal (2015)	Undefined	3	46	11							•																						
Xu and Feng (2014)	Construction	3	18	2		•		•				•		•		•	•	•				•							•				
Xu and Zhang (2012)	Construction	6	6	3						•	•					•													•				
Chen and Shahandashti (2009)	Undefined	3	46	11							•																					•	
Gutjahr et al. (2008)	E-commerce	18	10	-			•												•									•		•			
Kolisch and Meyer (2006)	Pharmaceutical	100	-	11			•										•	•												•			

(Continued on next page)

Table 3.8 – *(Continued from previous page)*

Research	Instance				Objective functions								Features																				
					Rsrc.			Proj.	Time				Jobs				Jobs-rsrc.			Rsrc.								Proj.-rsrc.	Proj.		Relp.		
	Domain	# proj.	# jobs	# rsrc.	NPV	TPC	TPP	PQ	APD	WPD	M	WM	PP	QJ	R/L	JU	JCP	MM	SK	D	L	NR	CL	PRU	RR	RC	M/U	PCP	W	PS	C/N	GL	TL
Chen (1994)	Maintenance	4	19	9		•					•											•				•		•	•				
Vercellis (1994)	Construction	10	20	6	•													•								•		•	•				
Speranza and Vercellis (1993)	Construction	10	20	6	•						•							•				•				•		•	•				

3.4 Conclusions

This chapter analyzed the investigations collected regarding features and objective functions. As a result, 34 features and 20 objective functions were identified. Based on this, a taxonomy was proposed for both aspects to unify concepts and visualize the current state of RCMPSP-related variants.

The features and objectives were also described in depth and analyzed in terms of the evolution of the problem and the number of existing studies. The analysis showed that the predominant assumption is solving the problem under deterministic conditions. In addition, the latest research lines mainly focus on studying decentralized environments, local and non-renewable resources, multiple execution modes, and weights for each project. On the other hand, time- and resource-based objective functions represent the main categories, with the minimization of the makespan, average project delay, and total project cost being the most predominant objectives.

The works applied to real environments were also analyzed. In this sense, the complexity of the real instances is less than that of the instances proposed in the literature. We also identified that the most studied features are the multiple execution modes, non-renewable resources, and the different weights of the projects. Also, the objectives related to time and resources are the main categories, highlighting makespan minimization and the total project cost minimization.

Chapter 4

RCMPSP with local resources

Resource allocation is one of the most studied aspects in the research literature, which can be seen through the thirteen features proposed in Section 3.1. Local resources represent one of the most significant features since 2010 presented in the literature, and we described in Section 3.1.3. The existence of local resources has been studied primarily in conjunction with other features, for instance, decentralized management, multiple modes of execution, and the existence of non-renewable resources. While the investigations of Villafáñez et al. (2019) and Wauters et al. (2015) are oriented to study the existence of local resources together with the features of the base problem.

On the other hand, the objective functions (see the analysis in Section 3.3.2) related to time are the most tackled in the literature, for example, the minimization of makespan (M) and average project delay (APD). This trend is mainly reflected in the research addressing local resources' existence. Specifically, the works of Villafáñez et al. (2019) and Wauters et al. (2015) pursue the minimization of the M, and Wauters et al. (2015) also pursue the minimization of the APD.

Given those mentioned above, in this study, we focused on solving RCMPSP with local resources (RCMPSP-L), with the aim to minimize the average sum of makespan and delay time of each project. To provide feasible solutions for the RCMPSP-L, we proposed a hybrid approach by integrating an ant colony optimization (ACO) (Dorigo et al., 1996) constructive metaheuristic with a local search method. The local search was developed employing a hill-climbing (HC) first improvement (FI) algorithm to enhance the constructive solutions. As part of HC, we used the Z3 tool proposed by (De Moura and Bjørner, 2008), which is an SMT solver, to support the process of verifying movement satisfaction and precedence of jobs, formalizing the constraints as logic formulas.

Based on the above-mentioned and discussion presented in previous sections, the contribution of this study is three-fold:

- We propose a new mathematical formulation for the RCMPSP-L, a variant of the problem that studies the features of the base problem together with the existence of local resources belonging to each project.
- For solving this variant, we develop a hybrid approach that integrates ant colony optimization (ACO) with a hill-climbing (HC) first-improvement (FI) algorithm.
- Regarding numerical experiments, we provide new bounds for some instances from the MPSPLib library, considering the resource management in the RCMPSP-L variant from the centralized point of view.

The details of this study are presented as follows. Section 4.1 describes the problem and its proposed mathematical model. The proposed hybrid algorithm is described in Section 4.2. Section 4.3 reports the computational experiments, results, and comparisons with approaches proposed in the literature. Finally, in Section 4.4, the study's conclusions are presented.

4.1 Problem description

The RCMPSP-L is based on the definition of the base problem and local resources described in Sections 2.1 and 3.1.3, respectively. Given this, in the RCMPSP-L, a set of projects comprises a subset of jobs. Each project defines a release date or arrival time and the desired date. Each job corresponding to the project is defined by a processing time and quantity required for each resource type. Some jobs belonging to the same project might establish end-start precedence relationships. To complete jobs, there is a constant number of (i) shared renewable resources and (ii) own renewable resources available in each period. The shared or global resources can be used for the execution of all jobs of all projects. Otherwise, the local resources (L) are defined as resources belonging to a project that can be consumed exclusively by the jobs belonging to the project. In addition, the resources are defined as a unique category; that is, the access to a resource cannot be local or global. A feasible schedule for the problem consists of assigning start times to each job in each project, satisfying the precedence relationships between jobs, and the availability of global and local resources at each period.

4.1.1 Mathematical formulation

Following the description mentioned above, the base mathematical model, and the nomenclature described in Section 2.1.1, we defined the following mathematical formulation for the RCMPSP-L.

The description of parameters K and k are modified as follows:

K set of different types of global resource,

k amount of global resource, $k \in \{1, 2, \ldots, |K|\}$,

The parameters c_{ij} and a_k are removed, and the following parameters are added:

E_i set of different types of local resource from project i,

e amount of local resource, $e \in \{1, 2, \ldots, |E_i|\}$,

gc_{ij}^k consumption of type of global resource k required to execute a job j of a project i,

lc_{ij}^e consumption of type of local resource e required to execute a job j of a project i,

ga^k availability of type of global resource k at every period,

la_i^e availability of type of local resource e at every period from a project i.

The following decision variable must also be added:

$$D_i = \begin{cases} 1, & \text{if project } i \text{ is delayed,} \\ 0, & \text{otherwise.} \end{cases}$$

The objective pursued is related to time and can be defined as the minimization of the average of the sum of the makespan and delay time of each project, which can be formulated as shown below:

$$\text{Minimize} \quad \frac{1}{|I|} \sum_{i=1}^{|I|} \Big(\sum_{t=1}^{|T|} t \cdot X_{i|N_i|}^t + (\sum_{t=1}^{|T|} t \cdot X_{i|N_i|}^t - DD_i) \cdot D_i \Big) \tag{4.1}$$

Constraints (2.4) that ensure the use of global resources must be replaced by the following constraints:

$$\sum_{i=1}^{|I|} \sum_{j=1}^{|N_i|} \sum_{q=t}^{t+D_{ij}-1} gc_{ij}^k \cdot X_{ij}^q \leq ga^k \ , \quad \forall\, k \in K, t \in T \tag{4.2}$$

$$\sum_{j=1}^{|N_i|} \sum_{q=t}^{t+D_{ij}-1} lc_{ij}^e \cdot X_{ij}^q \leq la_i^k \ , \quad \forall\, i \in I, k \in K, t \in T \tag{4.3}$$

Constraints (4.2) ensure that the allocation of global resources does not exceed the availability of each global resource at each instant of time. While the constraints (4.3) ensure that the sum of local and global resources assigned are enough to supply the demand of each resource for each project at each instant of time.

To manage project delay times, it is required to add the following constraints:

$$\sum_{t=1}^{|T|} t \cdot X_{i|N_i|}^{t} - DD_i \leq D_i \cdot M \ , \ \ \forall\, i \in I \tag{4.4}$$

When adding the aforementioned decision variable, it is required to add the following constraints to the model:

$$D_i \in \{0, 1\} \ , \ \ \forall\, i \in I \tag{4.5}$$

4.1.2 Illustrative example

Figures 4.1 and 4.2 show an illustrative example of a solution found for the mp_j30_a2_nr4 instance. The instance considers 2 projects, each with a 30 job and only one local resource type. The projects P_1 and P_2 present start time values of 0 and 7, respectively. The rectangles represent the time period that a job is executed. At the same time, the lines that join the rectangles represent the precedence relationships between jobs of the same project. In this case, both projects finish the execution of their jobs in the time period of $M = 56$.

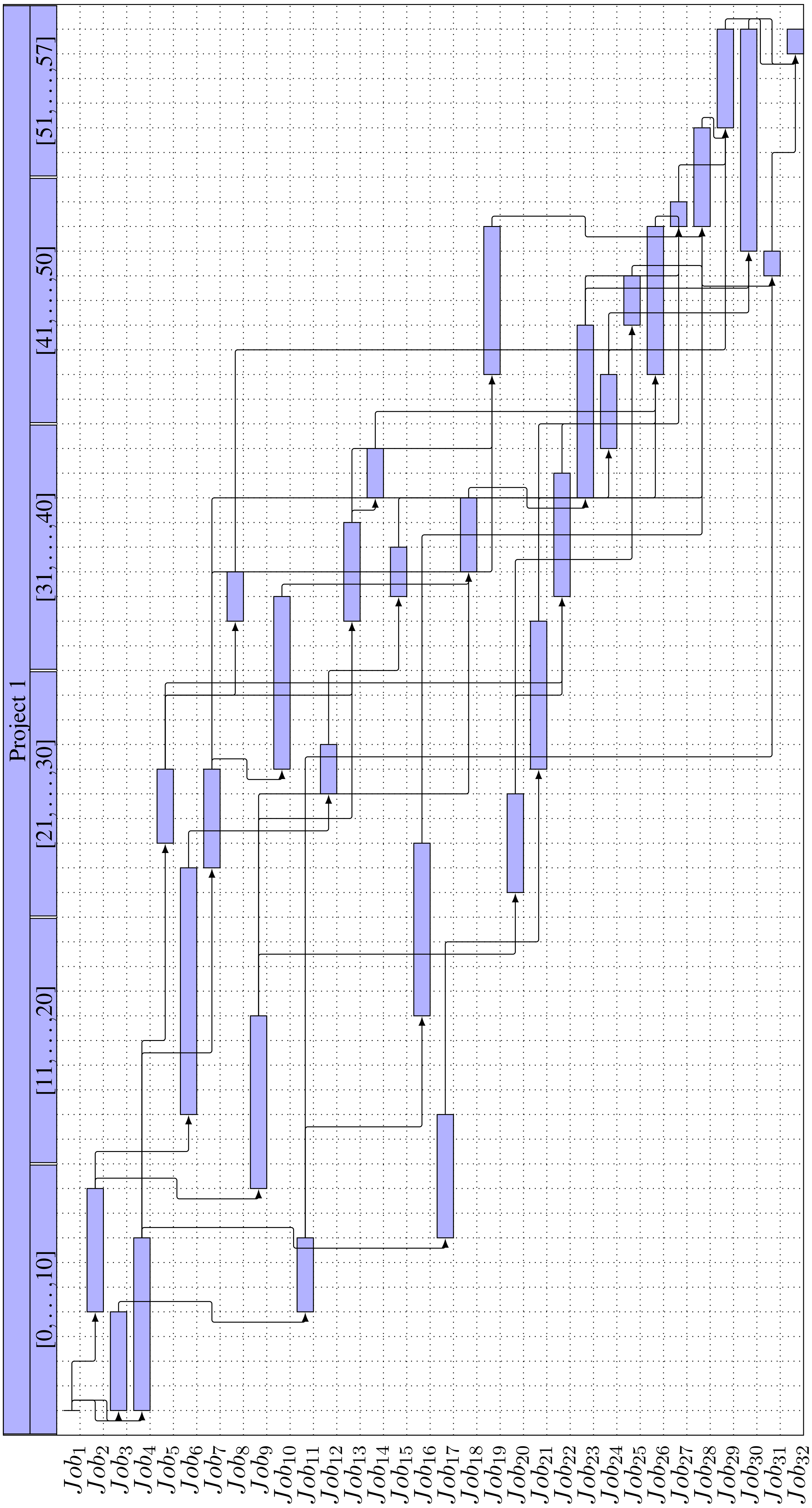

Figure 4.1: Illustrative example of a solution for the RCMPSP-L corresponding to the instance *mp_j30_a2_nr4/j309_9*. This example is composed of Project 1 with 32 jobs. Available at (http://www.mpsplib.com/ranking.php?level=1&id=9).

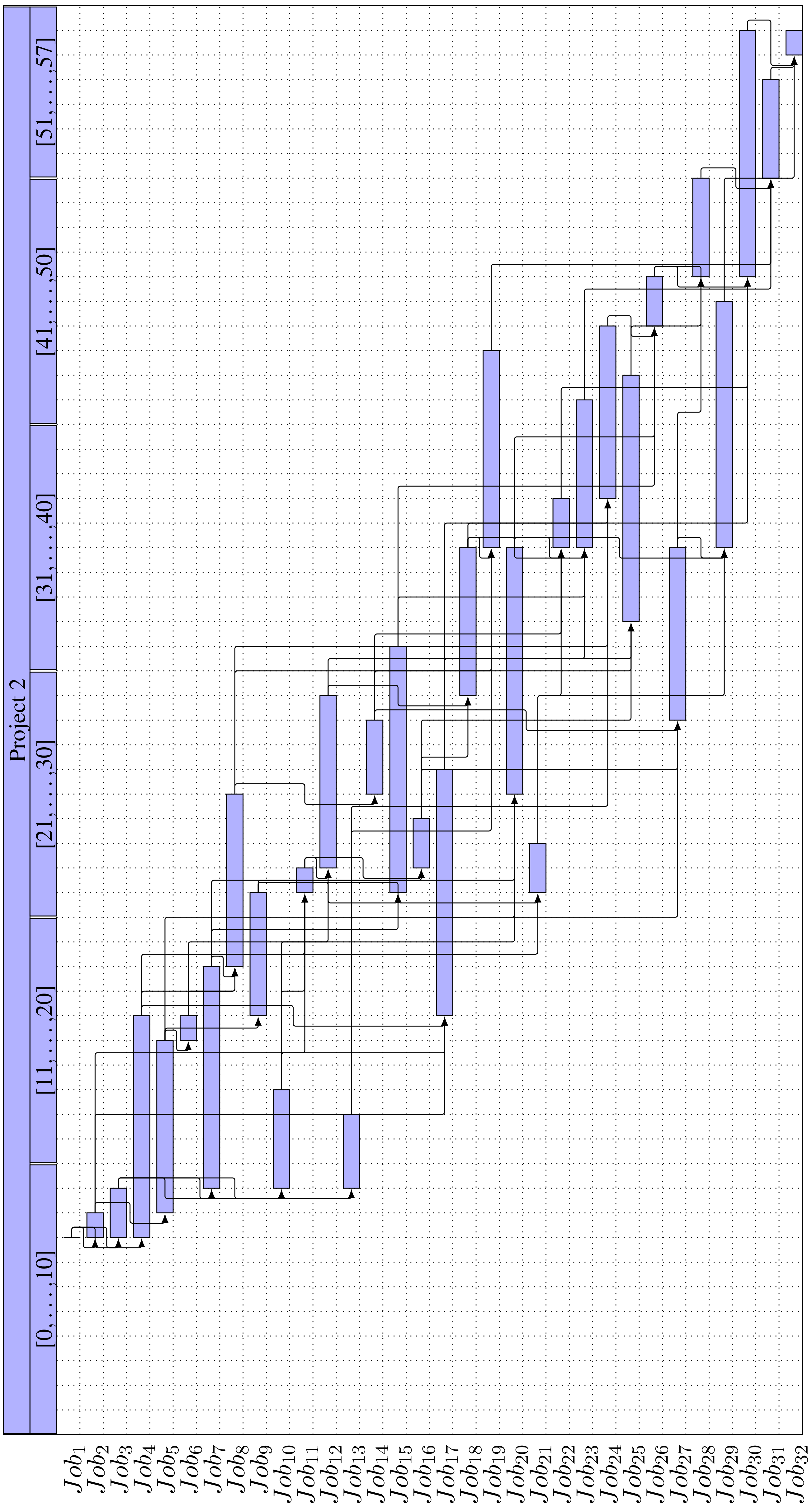

Figure 4.2: Illustrative example of a solution for the RCMPSP-L corresponding to the instance *mp_j30_a2_nr4/j3033_3.sm*. This example is composed of Project 2 with 32 jobs. Available at (http://www.mpsplib.com/ranking.php?level=1&id=9).

4.2 Solution approach

In order to solve RCMPSP-L described in the previous section, we proposed a hybrid approach that integrates an ant colony optimization (ACO) algorithm and a hill-climbing (HC) as an improvement solution method.

The ACO algorithm is a population-based metaheuristic that provides approximate solutions to complex optimization problems. Ant optimization algorithms were introduced by Dorigo et al. (1996) and belong to the population-based heuristics class. The way inspired them was that a colony of ants could identify the closest feeding source to their nest by cooperating with one another through a pheromone trail. There are several applications of ACO and swarm intelligence techniques to a wide range of multi-project scheduling problems, such as RCMPSP with non-renewable resources and flexible resource allocation (Rokou et al., 2014), RCMPSP with job quality, uncertainty and cost/profit, as well as multiple execution modes, non-renewable resources and different weights associated with each project (Xu and Feng, 2014), and the base problem (Linyi and Yan, 2007).

The improvement method is based on an HC algorithm with the first improvement (FI). HC-FI examines the neighboring solutions one by one and selects the first neighboring solution that improves the current solution. After applying the movements, we used the advantages of Z3 API in C++, an SMT solver, to formalize the precedence constraints as logical formulas to verify the feasibility of a solution. Satisfiability modulo theories (SMT) (De Moura and Bjørner, 2011) are suitable to the scheduling problems since they have a strong boolean and arithmetic component (Suy, 2012). In the doctoral thesis of Suy (2012), the SMT solvers can express incompatibilities in precedence relationships in scheduling problems. More examples of using SMT solvers can be seen in the work of Bofill et al. (2020), where the authors evaluate the performance through several SMT formulations (i.e., OptiMathSAT (Sebastiani and Trentin, 2020) and Z3 (De Moura and Bjørner, 2008)) to solve RCPSP.

4.2.1 Representation

To represent a solution, we used a complete graph where each node i, j indicates a job j of project i. Between each pair of nodes, there is an edge $(\langle i, j\rangle, \langle w, x\rangle)$, which represents a transition starting to the job j of project i towards the job w of project x. Each edge $(\langle i, j\rangle, \langle w, x\rangle)$

has associated a pheromone trail $\tau_{(\langle i,j\rangle,\langle w,x\rangle)}$, that represents how frequent has been visited this edge during the search.

The solution of the problem sol is stored in the memory of the ant, represented by a triad vector that indicates the start time sequences. Each triad is composed of a job, the related project, and the start execution time. For example, in a case where the Job_{11} started its execution in period 1, and the job Job_{31} started its execution in period 2, the representation of the solution is composed as shown below:

$$sol = \{\ (1,1,1),\ (1,3,2),\ldots,(|I|,|N_i|,|T|)\}$$

4.2.2 General procedure

Algorithm 1 describes the general structure of our hybrid approach. Initially, in line 1, the number of ants N_{ANTS} and the number of iterations N_{ITER} are set, and the pheromone levels on each edge are initialized with 0.1. For each iteration, the ants must get the pheromone levels information before the remaining ants influence the pheromone matrix. To ensure the above-mentioned, in line 4, we carried out a copy of the pheromone matrix $\mathcal{M}$ that allows the ants to get and modify the information. In lines 6-11, each ant builds a solution to the problem, and the best solution is stored. After that, in line 12, a local search procedure was applied to the constructed solution. Subsequently, in line 13, the pheromone levels on each edge of the complete graph were evaporated to avoid getting stuck at local optimal solutions. Finally, in the lines 14, 15, and 18, the procedure determines the best solution found $\mathcal{S}_F$ for all iterations.

With the aim of avoiding local optimal solutions, we limited the pheromone levels values between 0.0 and 1.5. In addition, the evaporation process was applied $\forall(\langle i,j\rangle,\langle w,x\rangle) \in \mathcal{M}$ using the following equation:

$$\tau_{(\langle i,j\rangle,\langle w,x\rangle)} = (1-\rho)\cdot\tau_{(\langle i,j\rangle,\langle w,x\rangle)}, \quad \forall_{(\langle i,j\rangle,\langle w,x\rangle)} \in \mathcal{M} \tag{4.6}$$

4.2.3 Constructive algorithm

The ant search process in ACO algorithms is guided by two types of information: memoristic information or pheromone trail and heuristic information. The memoristic information is related to each possible path or edge and can be modified by each ant on each iteration. This modification depends on the path of the ant within the complete graph defined by order of the triads

Algorithm 1 General procedure

Parameters initialization
$\mathcal{S}_F = \varnothing$
while $(N_{ITER} > 0)$ **do**
  $\mathcal{M}' = \mathcal{M}$
  $\mathcal{S}_{Ant} = \varnothing$
  for $i = 1 \ldots N_{ANTS}$ **do**
    $sol =$ Constructive algorithm
    if $f(sol) < f(S_{Ant})$ **or** $S_{Ant}.size = 0$ **then**
      $S_{Ant} = sol$
    end if
  end for
  $S_{Ant} =$ Local search(S_{Ant})
  Evaporation of pheromone in $\mathcal{M}$
  if $f(S_{Ant}) < f(\mathcal{S}_F)$ **then**
    $\mathcal{S}_F = S_{Ant}$
  end if
end while
return $\mathcal{S}_F$

within sol. That means if the following vector represents the solution found by an ant:

$$sol = \{(1, 1, 1), \ldots, (i, j, 20), (w, x, 23), \ldots, (|I|, |N_i|, |T|)\}$$

then, the ant traveled the edge $(\langle i, j\rangle, \langle w, x\rangle)$ and $\tau_{(\langle i,j\rangle,\langle w,x\rangle)}$ it was intensified following the quality of the sol and is calculated using the equation:

$$\tau_{(\langle i,j\rangle,\langle w,x\rangle)} = \tau_{(\langle i,j\rangle,\langle w,x\rangle)} + \frac{1}{f(sol)}, \quad \forall((i, j, t), (w, x, t + 1)) \in sol \tag{4.7}$$

where $f(sol)$ represents the value of the fitness function for sol, ensuring that the longer the execution and delay time of the project, and fewer pheromones will be distributed along the path.

On the other hand, each node has an associated heuristic preference denoted as $\eta_{i,j}$ and defined by:

$$\eta_{i,j} = \frac{1}{d_{i,j}} + \frac{1}{DD_i} \tag{4.8}$$

where $d_{i,j}$ is the duration of the job j of project i and DD_i is the desired due date of project i.

Both types of information (memoristic and heuristic) present different levels of influence in the transition rule, which define each step of the ant. Those are defined by α and β, respectively, while the transition rule is established as follows:

$$P_{\langle i,j \rangle, \langle w,x \rangle} = \frac{[\tau_{(\langle i,j \rangle, \langle w,x \rangle)}]^{\alpha} \cdot [\eta_{w,x}]^{\beta}}{\sum_{l,h \in \mathcal{F}} [\tau_{(\langle i,j \rangle, \langle l,h \rangle)}]^{\alpha} \cdot [\eta_{l,h}]^{\beta}} \tag{4.9}$$

Algorithm 2 defines the proposed constructive algorithm. To build a solution, the ant performs N_{STEPS} transitions or steps corresponding to the total jobs of all projects (line 2). In each step, the current node $\langle i, j \rangle$ where the ant is located is defined (line 3). Also, the constraints of the problem are checked, obtaining a set of nodes $\langle l, h \rangle \in \mathcal{F}$ towards which it is feasible to move to start from $\langle i, j \rangle$ (line 4). Then, for each node $\langle w, x \rangle \in \mathcal{F}$ we calculate the probability $P_{(\langle i,j \rangle, \langle w,x \rangle)}$ that the ant's next step is towards node $\langle w, x \rangle$ (lines 5-9). Subsequently, we use the roulette-wheel selection method proposed by Lipowski and Lipowska (2012) to select a node (line 10). In addition, the selected node joined with the corresponding time is stored in sol, and the available resources for the next step are updated (lines 11 and 12). Once the solution is completed, pheromones are deposited on the path traveled according to the quality of the solution found in equation (4.7) (line 14). Finally, the constructive solution is returned (line 15).

4.2.4 Local search algorithm

The improvement method is based on a hill-climbing first improvement (HC-FI) algorithm. In this algorithm, a fundamental aspect is the definition of the movement from which the neighbors of the current solution are generated. In this study, the movement was defined as randomized movement as follows:

A node is randomly selected from the $\mathcal{S}_{Ant}$, $\langle i, j \rangle = random(\mathcal{S}_{Ant})$ and $\mathcal{S}_{Ant} = \mathcal{S}_{Ant} \setminus \langle i, j \rangle$, and $\langle i, j \rangle$ is inserted in another random position $p = random(\mathcal{S}_{Ant})$ and $\mathcal{S}'_{Ant} = \mathcal{S}_{Ant}[p] \cup \langle i, j \rangle, \forall_p \in |\mathcal{S}_{Ant}|, \forall_{\langle i,j \rangle} \in \mathcal{S}_{Ant}$.

After performing a movement, we verified the feasibility of $\mathcal{S}'_{Ant}$ concerning the precedence relations. To support that, we used the SMT solver (Z3, (De Moura and Bjørner, 2008)) as part

Algorithm 2 Constructive algorithm

$sol = \varnothing$
for $i = 1 \dots N_{STEPS}$ **do**
  $\langle i, j \rangle$ = Obtain (sol)
  $\mathcal{F}$ = Check problem's constraints
  for $\langle w, x \rangle$ **in** $\mathcal{F}$ **do**
    Find $\tau_{\langle i,j \rangle, \langle w,x \rangle}$ in $\mathcal{M}'$
    Calculate $\eta_{w,x}$
    Calculate $P_{\langle i,j \rangle, \langle w,x \rangle}$
  end for
  Select a node and apply the movement
  Add the selected node into sol
  Update resource availability
end for
Deposit pheromone in $\mathcal{M}$ for movements involved in sol
return sol

of the HC-FI algorithm. The SMT solvers, as we mentioned above, are based on SAT problem, the first problem known as NP-hard compound by a given set of propositional logic formulas $\Phi = \{\varphi_1, \varphi_2, \dots, \varphi_n\}$ composed by a set of Boolean variables (e.g., $x_1, x_2, \dots, x_n$) which are related by logical connectives (e.g., $\vee$, $\wedge$, or $\neg$). In most SAT contexts and applications, the formulas are presented in conjunctive normal form (CNF), for instance, $\varphi_1 = (\neg x_1 \vee x_2)$, $\varphi_2 = (\neg x_1 \vee x_2 \vee \neg x_3)$ and $\varphi_3 = (\neg x_1 \vee x_3 \vee \neg x_4)$, representing clauses of literals joined by disjunctions, whose final objective is to determine the values must be assigned to the literals so that $\bigwedge_{i=1}^{3} \varphi_i$ is $true$ (Gálvez, 2018). We proposed the following Boolean constraint formulation to verify the feasibility of a movement.

The parameter is defined as follows:

k_{ij} represents the employed position in the memory of the ant for each job j of the project p, $\forall_i \in \{1, \dots, |I|\}, \forall_j \in \{1, \dots, |N_i|\}$.

The decision variable is defined as follows:

J_{ij} represents the employed position related to the ant memory for each job j of the project i, $\forall_i \in \{1, \dots, |I|\}, \forall_j \in \{1, \dots, |N_i|\}$.

The precedence constraints are defined as follows:

$$\left(\bigwedge_{i=1}^{|I|}\bigwedge_{j=1}^{|N_i|}\bigwedge_{q=1}^{|N_i|} J_{ij} < J_{qi}\right) \wedge \left(\bigwedge_{i=1}^{|I|}\bigwedge_{j=1}^{|N_i|} J_{qi} = k_{qi}\right), \quad \forall J_{qi} \in S_{ij} \tag{4.10}$$

where S_{ij} represents the set of successors of the job j of project i.

Algorithm 3 presents the proposed local search algorithm, which receives and improves the best solution S_{Ant} from the constructive phase. During a specific number of iterations N_{ITERB}, a feasible movement is successively searched (lines 1-3). In line 4, a randomized movement is performed. The feasibility of S'_{Ant} was verified with Z3 (lines 5 and 6), checking if the precedence constraints are satisfied in a new order. If the verification is infeasible (UNSAT), then the process is repeated; otherwise (SAT), then the start times of each node $\langle i, j \rangle$ are recalculated in $\mathcal{S}_{Ant}$ considering the availability of resources (line 8). On each iteration, the best solution found so far is stored in $\mathcal{S}_{Ant}$ (lines 9 and 10). Finally, pheromones trails are deposited on the path traveled according to the quality of $\mathcal{S}_{Ant}$ (see equation (4.7)) (line 15).

Algorithm 3 Local search ($\mathcal{S}_{Ant}$)

1: **for** $i = 1 \ldots N_{ITERB}$ **do**
2:     $decision$ = `UNSAT`
3:     **while** $decision$ = `UNSAT` **do**
4:         $\mathcal{S}'_{Ant} =$ Randomized movement($\mathcal{S}_{Ant}$)
5:         $\Phi =$ Boolean formulation($\mathcal{S}'_{Ant}$)
6:         $decision$ = Check problem's constraints(Φ)
7:         **if** $decision$ = `SAT` **then**
8:             $\mathcal{S}'_{Ant} =$ Recalculate periods ($\mathcal{S}'_{Ant}$)
9:             **if** $f(\mathcal{S}'_{Ant}) < f(\mathcal{S}_{Ant})$ **then**
10:                 $\mathcal{S}_{Ant} = \mathcal{S}'_{Ant}$
11:             **end if**
12:         **end if**
13:     **end while**
14: **end for**
15: Deposit pheromone in $\mathcal{M}$ for movements involved in $\mathcal{S}_{Ant}$
16: **return** $\mathcal{S}_{Ant}$

4.3 Experimentation results

The proposed algorithm was implemented in C++ and tested on a processor Intel(R) Xeon(R) CPU E3-1220 3.10 GHz with 32GB RAM memory on CentOS Linux 7.6.18.19 system. Our approach was tested through the online validator available on MPSPLib[1]. Also, we carried out a 100 run for each instance using the values of the parameters indicated in Table 4.1.

4.3.1 Problem instances

In order to validate the proposed algorithm's performance, 24 instances of MPSPLib were selected. The MPSPLib by Homberger (2012) (see description in Section 2.3) was created to study the RCMPSP-L for the management of the resource in a decentralized way. Also, according to the research of Villafáñez et al. (2019), the MPSPLib can be used to study the RCMPSP-L by managing it in a centralized way. This library presents instances that consider a number of projects among 2, 5, 10, and 20, and for each, a number of jobs of 30, 90, or 120. Also, there are 4 types of resources, being defined some of them as local resources. We selected a set of 8 instances for each number of jobs between 30, 90, and 120, which in turn considers 7, 5, 7, and 5 instances of 2, 5, 10, and 20 projects, respectively.

4.3.2 Parameter analysis and tuning

We perform a parameter tuning using ParamILS (Hutter et al., 2009), taking into account some parameter values used in the literature for ACO algorithms. Table 4.1 shows the interval of values provided to the tuner and the optimal parameter configuration provided by the tuner. Moreover, to provide suitable values to the tuner, the following aspects were considered:

- Number of ants and the number of iterations: higher values imply a greater possibility of finding better solutions, but they influence execution times considerably.
- Memoristic information: values close to 0.0 tend to turn the constructive algorithm into a Greedy algorithm.
- Heuristic information: values close to 0.0 tend to provide a quick convergence of the constructive algorithm.

[1] http://www.mpsplib.com/upload.php

- Evaporation levels: quick evaporation of pheromone levels can quickly cause the trails that represent good solutions to become undesirable.

Table 4.1: Parameter setting scenario configurations and results.

Parameter	Description	Value	Interval
N_{ITER}	Number of iterations of the algorithm	100.0	$[50.0, 100.0]$
N_{ITERB}	Number of iterations of the local search	80.0	$[50.0, 80.0]$
N_{ANTS}	Number of ants	100.0	$[50.0, 100.0]$
α	Influence of memoristic information	0.6	$[0.3, 1.0]$
β	Influence of heuristic information	3.0	$[2.0, 4.0]$
ρ	Evaporation level	0.2	$[0.1, 0.5]$

4.3.3 Discussion and literature comparison

Table 4.2 reports the makespan (M) values provided by our algorithm for the selected instances. Columns 1 and 2 show the instance name and the distribution of global and local resources (format $\langle \mathcal{G}{:}E_i \rangle$), respectively. Finally, the last columns 3 and 4, report the best solution for the constructive and the local search algorithm, respectively. As can be seen, the local search procedure always improves the solution found by the constructive algorithm. Nevertheless, we highlight the constructive algorithm as our strength because it provides good-quality solutions due to the transition rule where heuristic information and pheromone trails are integrated.

To analyze the behavior of the proposed algorithm, Figure 4.3 shows boxplot charts of the makespan values for the selected instances after the 100 run. As can be seen, the algorithm presents a stable behavior, with relatively low-value variability, despite the significant differences between the characteristics and sizes of the instances selected to carry out the experimentation process. The most significant variability occurs in mp_j90_a2_nr5, mp_j90_a5_nr4, and mp_j120_a10_nr4, where the extreme upper values of makespan represent outliers found in the search process.

We consider the following essential factors as those that provided stability and good behavior of the algorithm:

- The number of pheromones deposited was proportional to the solution quality provided

Table 4.2: Comparison of components performance of our approach.

Instance	Resources	Constructive	Local search
mp_j30_a2_nr1	$\langle 2:2 \rangle$	75	71
mp_j30_a2_nr4	$\langle 3:1 \rangle$	58	56
mp_j30_a2_nr5	$\langle 1:3 \rangle$	63	59
mp_j30_a5_nr2	$\langle 1:3 \rangle$	83	79
mp_j30_a10_nr1	$\langle 2:2 \rangle$	190	188
mp_j30_a10_nr2	$\langle 1:3 \rangle$	115	112
mp_j30_a10_nr5	$\langle 1:3 \rangle$	189	187
mp_j30_a20_nr4	$\langle 3:1 \rangle$	190	187
mp_j90_a2_nr2	$\langle 1:3 \rangle$	129	125
mp_j90_a2_nr3	$\langle 2:2 \rangle$	116	114
mp_j90_a2_nr5	$\langle 1:3 \rangle$	126	121
mp_j90_a5_nr2	$\langle 1:3 \rangle$	118	114
mp_j90_a5_nr3	$\langle 2:2 \rangle$	151	140
mp_j90_a5_nr4	$\langle 3:1 \rangle$	134	127
mp_j90_a10_nr4	$\langle 3:1 \rangle$	152	150
mp_j90_a20_nr2	$\langle 1:3 \rangle$	169	167
mp_j120_a2_nr5	$\langle 1:3 \rangle$	128	119
mp_j120_a5_nr2	$\langle 1:3 \rangle$	180	178
mp_j120_a10_nr1	$\langle 2:2 \rangle$	137	134
mp_j120_a10_nr3	$\langle 2:2 \rangle$	149	145
mp_j120_a10_nr4	$\langle 3:1 \rangle$	379	376
mp_j120_a20_nr1	$\langle 2:2 \rangle$	91	88
mp_j120_a20_nr3	$\langle 2:2 \rangle$	241	236
mp_j120_a20_nr4	$\langle 3:1 \rangle$	209	205

by considering the makespan value and the delay of each project. Smaller delay and makespan represent a greater amount of pheromones to deposit on each edge corresponding to the path traveled by the ant, generating more desirability for other ants.

- The pheromone is deposited on the best solution found in the local search procedure, although there is no change concerning the solution provided by the constructive algorithm.

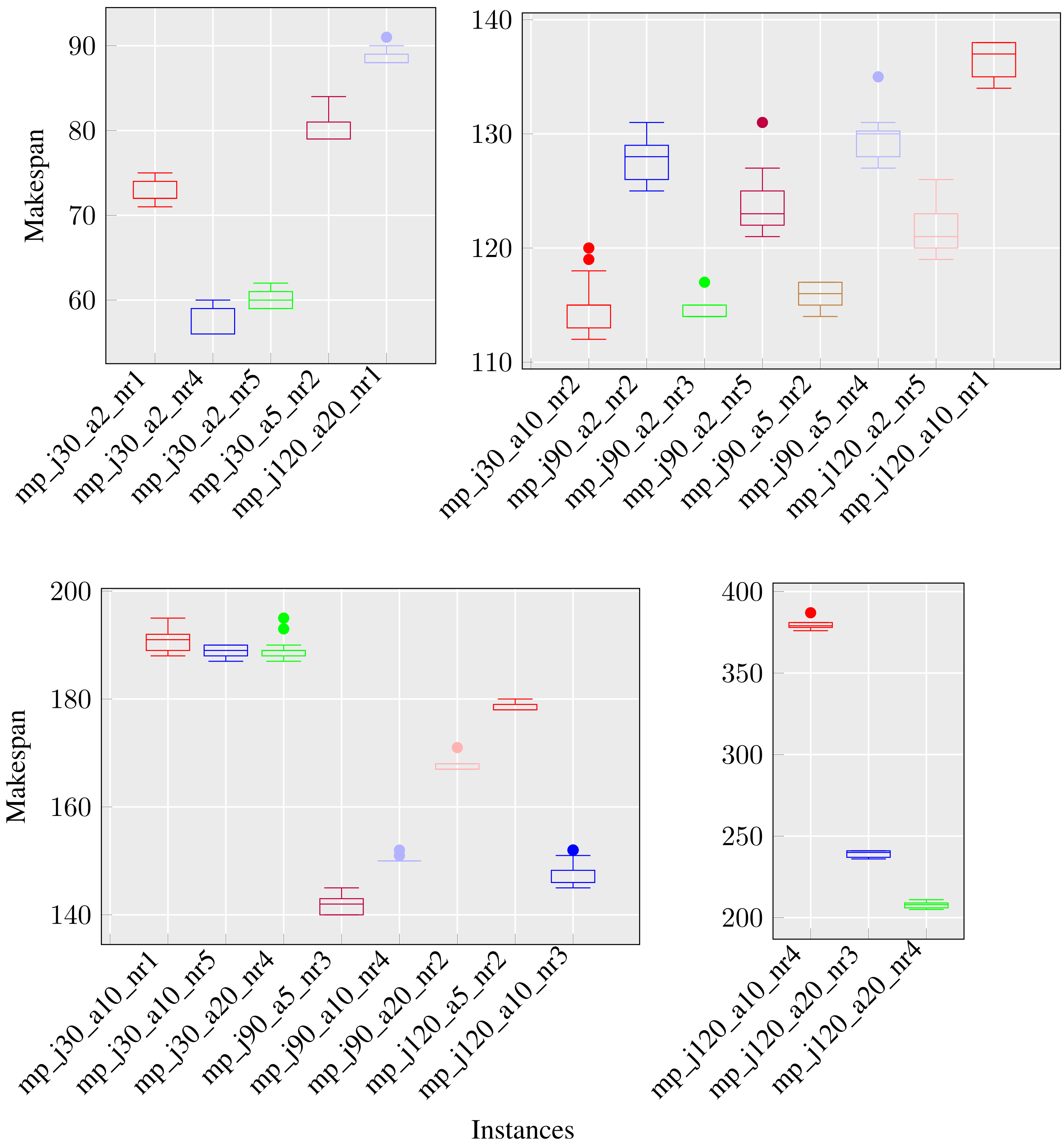

Figure 4.3: Boxplot charts showing the makespan values (M) for RCMPSP-L on MPSPLib instances. Note: 100 runs.

- Pheromone levels do not exceed 1.5, thus encouraging the exploration of solutions by avoiding local optimums.
- The heuristic information allows prioritizing those jobs (i) with a shorter duration, causing a quick release of the resources, and (ii) belonging to projects with less completion time by allocating a wide range within the roulette.

Following the review of Adam et al. (2019), the 'No Free Lunch' theorem establishes that all optimization algorithms perform equally well by averaging over all optimization problems without re-sampling. Despite this, to analyze the proposed algorithm's performance, we compare the results provided by our approach with respect to other well-known approaches.

The results provided for the instances for APD, M, and DPD metrics show that our approach is competitive. To compare our results, we used the results published in the MPSPLib. Even though this problem does not cover decentralized resource management, the results are compared with some approaches that study this feature and others that do not.

In Table 4.3, the columns 2-13 show the values for the previously described metrics for four approaches. As can be seen, our proposed approach provides better results in 21% of the cases, while for 38% of the instances, the same values as those presented in the literature are found, both for the centralized and decentralized approaches. The results shown in the columns 2-4 are those presented by Villafáñez et al. (2019), research that studies the centralized allocation of resources. Our results compared with this research shows that the percentage of instances where we provided better results increases to 42%, while the percentage of cases in which we find equal values is maintained. Concerning the best solutions presented in the table, we provided an average Gap (%) of 1.07%, with at most 7.32% in the instance mp_j120_a20_nr1. Note that we selected the makespan values as a quality measure to perform the comparisons, but our algorithm also offers good quality values for the APD and DPD measures.

Table 4.4 shows the Friedman-k Related Samples test by statistically comparing the results provided for three approaches. According to the test, with a 95% of confidence, the results indicate that for ADP and M, the differences between the samples are significant for the three metrics of the problem. In addition, it is observed that there is no best approach for the three metrics, providing our approach with the lowest average range value for the makespan.

Table 4.3: Performance comparison to other approaches in MPSPLib for RCMPSP-L.

Instance	***Lopez-Paredes/Pajares/Villafañez***			***Tony Wauters***			***Dietz/Homberger***			***Our approach***			
	APD	M	DPD	APD	M	DPD	APD	M	DPD	APD	M	DPD	Gap (%)
mp_j30_a2_nr1	14.00	**71.00**	1.41	13.00	74.00	7.07	14.00	**71.00**	1.41	14.50	**71.00**	0.71	0.00
mp_j30_a2_nr4	14.50	58.00	9.19	13.50	58.00	6.36	14.00	59.00	5.66	13.00	**56.00**	8.49	0.00
mp_j30_a2_nr5	10.50	**59.00**	10.61	10.00	61.00	14.14	9.00	**59.00**	12.73	12.50	**59.00**	7.78	0.00
mp_j30_a5_nr2	17.80	80.00	12.01	17.00	**79.00**	12.33	16.80	84.00	13.31	18.00	**79.00**	12.41	0.00
mp_j30_a10_nr1	134.50	**188.00**	19.52	154.90	192.00	3.07	150.40	190.00	4.45	143.50	**188.00**	13.95	0.00
mp_j30_a10_nr2	13.10	114.00	11.65	18.00	113.00	11.42	17.20	118.00	14.91	17.80	**112.00**	8.88	0.00
mp_j30_a10_nr5	89.00	**187.00**	26.25	58.10	211.00	46.12	79.90	196.00	40.44	93.10	**187.00**	34.69	0.00
mp_j30_a20_nr4	60.25	**183.00**	34.72	34.00	212.00	47.95	84.35	190.00	28.81	67.45	187.00	39.84	2.19
mp_j90_a2_nr2	27.00	**124.00**	22.63	26.50	127.00	27.58	28.50	127.00	24.75	30.50	125.00	19.09	0.81
mp_j90_a2_nr3	0.50	115.00	0.71	0.00	**114.00**	0.00	0.00	**114.00**	0.00	4.50	**114.00**	6.36	0.00
mp_j90_a2_nr5	0.00	**121.00**	0.00	0.00	**121.00**	0.00	0.50	122.00	0.71	0.00	**121.00**	0.00	0.00
mp_j90_a5_nr2	9.40	**114.00**	14.52	8.60	118.00	16.58	7.80	**114.00**	14.81	9.80	**114.00**	15.01	0.00
mp_j90_a5_nr3	3.60	140.00	6.99	3.40	**138.00**	7.60	3.40	**138.00**	7.60	9.00	140.00	16.85	1.45
mp_j90_a5_nr4	16.80	132.00	14.57	8.40	133.00	8.17	12.20	**123.00**	14.01	17.20	127.00	13.85	3.25
mp_j90_a10_nr4	2.10	**150.00**	2.33	1.10	**150.00**	2.08	5.10	**150.00**	6.37	6.20	**150.00**	4.87	0.00

(Continued on next page)

Table 4.3 – *(Continued from previous page)*

Instance	***Lopez-Paredes/Pajares/Villafañez***			***Tony Wauters***			***Dietz/Homberger***			***Our approach***			
	APD	M	DPD	APD	M	DPD	APD	M	DPD	APD	M	DPD	Gap (%)
mp_j90_a20_nr2	10.00	168.00	8.84	7.00	174.00	8.06	8.20	174.00	9.25	9.60	**167.00**	8.23	0.00
mp_j120_a2_nr5	9.00	118.00	1.41	7.50	120.00	6.36	3.00	**114.00**	4.24	16.00	119.00	7.07	4.39
mp_j120_a5_nr2	40.00	181.00	37.15	34.40	**177.00**	34.22	30.20	178.00	34.72	38.60	178.00	35.54	0.56
mp_j120_a10_nr1	65.30	134.00	5.23	43.50	152.00	26.81	64.80	**131.00**	1.03	67.10	134.00	3.28	2.29
mp_j120_a10_nr3	12.80	146.00	7.39	8.50	155.00	13.53	11.50	**142.00**	8.54	16.60	145.00	10.81	2.11
mp_j120_a10_nr4	244.00	**371.00**	14.51	193.00	416.00	81.94	252.00	**371.00**	15.87	255.50	376.00	16.44	1.35
mp_j120_a20_nr1	13.90	91.00	3.45	13.85	93.00	2.62	6.40	**82.00**	1.14	11.75	88.00	1.86	7.32
mp_j120_a20_nr3	52.45	238.00	32.07	40.70	242.00	36.37	54.60	251.00	30.83	59.45	**236.00**	32.74	0.00
mp_j120_a20_nr4	42.85	207.00	17.73	25.95	209.00	26.70	53.50	210.00	22.91	39.85	**205.00**	25.20	0.00
Average	37.64	145.42	13.12	30.87	151.63	18.63	38.64	146.17	13.27	40.48	144.92	14.33	1.07

Table 4.4: Friedman test results.

Approach	APD	M	DPD
Lopez-Paredes/Pajares/Villafañez	2.73	2.35	2.27
Tony Wauters	1.67	3.17	2.79
Dietz/Homberger	2.23	2.54	2.40
Our approach	3.38	1.94	2.54

4.4 Conclusions

In this study, we proposed a new mathematical formulation for the RCMPSP-L. This variant studies the features of the base problem together with the existence of local resources of each project. To solve this variant, we have proposed a hybrid optimization approach that integrates ant colony optimization (ACO) and a local search based on a hill-climbing first improvement (HC-FI) algorithm with the support of Z3.

The computational results show that the proposed approach finds better or equal solutions in more than 50% of the selected instances concerning those known in MPSPLib. The use of ACO shows that the algorithms belonging to swarm intelligence represent a potential solution to solve the RCMPSP-L. A local search algorithm based on HC-FI and Z3 was developed, providing improved solutions. The parameter tuning process using ParamILS allowed our approach to find better makespan values.

Chapter 5

RCMPSP with resources mixed accessibility

As we mentioned in previous sections, in multi-project environments, there is a wide range of features related to resource classification as part of the taxonomy presented in Chapter 3 and Figure 3.1. Adhau et al. (2013) state that in multi-project environments, the number of global resources allocated to a project can influence project delay and duration. In this sense, greater availability of global units can represent greater unit allocation, less delay, and a shorter duration of each project.

According to the classification presented in Section 3.1, in the studied literature, we found that the accessibility of the resources is one of the most studied classifications. Resource accessibility can be global or local and indicates the level of access that the jobs belonging to a project have concerning a resource type. Regarding global accessibility, the resource type has a unique availability shared by all projects. On the other hand, when a resource type is limited to local accessibility (Homberger, 2007), each project has its available units that can only be used to execute its jobs. When addressing accessibility in the multi-project scheduling problem, related works consider that each resource type has either global or local accessibility but not both (see Chapter 4). This might result in inefficient utilization of resources with local accessibility and delay of projects.

Based on the previous discussion, this work studies resources with mixed accessibility, where a resource type can have units with global and units with local accessibility. In contrast to

RCMPSP, in this new problem variant, job requirements can be simultaneously satisfied by global, local units, or both.

The existence of resources with mixed accessibility generates different modes of resource allocation. The resources can be allocated at each time period (unitary) or for the entire job duration (total). When the resources are allocated unitary, in each time period, the assignment fulfills the job requirement in execution but not necessarily with the same resource unit. For example, if a job requires 5 units of resource type 1, in the first period that the job is executed, it can be executed by 3 local units and 2 global units, while in the second period, it can be executed by 5 local units. On the other hand, when the resources are totally allocated, the resource units, global or local, allocated for the execution of a job are constant during the job duration. For example, suppose a job requires 5 units of resource type 1. In that case, if in the first period, the job is executed by 3 local units and 2 global units, the allocation made has to be preserved for the entire job duration. Following the previously proposed nomenclature, we named RCMPSP-MU and RCMPSP-MT, RCMPSP variants that study resources with mixed accessibility, and the allocation is carried out unitary or total, respectively.

To place both features related to the problem within the taxonomy proposed in Figure 3.1, we show in Figure 5.1 where the newly described features would be connected. Only the features related to the *Accessibility* subcategory are shown. A mathematical formulation for the RCMPSP with resources mixed accessibility and unitary resources allocation (RCMPSP-MU) minimizing the average project delay (APD) can be seen in Appendix A.

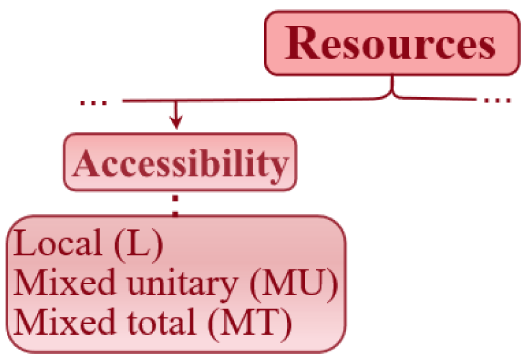

Figure 5.1: Location of new features into the taxonomy proposed in Section 3.1.

On the other hand, a features widely studied in the literature are the weights of projects and costs associated with using resources related to the categories of projects and resources, present on 58% and 42% of the related research carried out in real contexts (see Section 3.3.3). In this sense, Tian et al. (2020) study a practical multi-project scheduling example drawn from Qing'an Corporation, considering global and local accessibility and the cost associated with the delay of each project. Xu and Zhang (2012) consider costs associated with project delays due to the research is related to a large-scale water conservancy and hydropower construction project in the southwest region of China, whose main project is an embankment dam (large water impounding structure). In both works, the authors pursue minimizing the delay cost. Since the delay cost defines the weight or priority of the project, minimizing the delay cost is equivalent to minimizing the weighted project delay (WPD). The paper by Gholizadeh-Tayyar et al. (2016) consider several costs based on the planning of the project and the use of resources on a project called CRIBA to reduce the energy consumption level of buildings in France. The associated costs are related to the rental of resources, the use of own resources, the purchase and storage of resources, and the delay of projects, among others. Zhang and Chen (2018) present an actual study related to China Datang Corporation in charge of managing wind farm construction projects. In their problem, among other features, they consider global and local accessibility and costs associated with using renewable resources. In both works, the authors pursue minimizing the total project cost (TPC).

Considering that the delay and resource costs have been addressed in several studies, especially in real environments, joint with the importance of considering the costs in the different problem variants (e.g., see Tian et al. (2020) and Gholizadeh-Tayyar et al. (2016)), our research focuses on the analysis of both costs in the context of resources with mixed accessibility and total resources allocation.

Based on the previous discussion, the contribution of this study is three-fold:

- We propose two new variants of the RCMPSP with their corresponding mathematical models. Namely, RCMPSP-MT-W and RCMPSP-MT-W+RC. Both variants focus on resources with mixed accessibility, total resource allocation, and weights associated with projects, and the second variant includes the cost per use of units with global accessibility.
- For solving the proposed variants, we develop a hybrid approach that integrates iterated local search algorithm (ILS), simulated annealing (SA), tabu search (TS), and priority rules (PRs). The initial solution is constructed following a PR and improved through the SA. Subsequently, the best solution found in each iteration of ILS is perturbed and improved again with SA. The tabu list is used within the SA to avoid repetitively visiting a solution generated by a movement.
- Regarding experimental results, we provide a modified set of instances based on MP-

SPLib to study the mixed accessibility and resource costs. In this new set, the availability and accessibility of resources are modified, and the resource costs to units with global accessibility are added.

The details of this study are presented as follows. Section 5.1 describes the problem variants and the proposed mathematical models. The proposed hybrid algorithm for solving the presented problems is described in Section 5.2. Section 5.3 reports the computational experiments, results, and resource accessibility analysis. Finally, the study conclusions are presented in Section 5.4.

5.1 Problem descriptions

The RCMPSP-MT-W considers a set of projects, I, comprising a subset of jobs, $N_i, i \in I$. Each project defines an arrival time $AT_i, i \in I$ and the desired due date $DD_i, i \in I$. Each job j corresponding to the project i is defined by a processing time d_{ij} and quantity required for each resource type c_{ij}^k. Some jobs belonging to the same project might establish end-start precedence relationships, generating the set of predecessors P_{ij}. To complete jobs, there exists a set K of resource types. Each resource type $k \in K$ at each time period has availability of units with global accessibility ga^k shared by the projects and availability of units with local accessibility la_i^k for each project $i \in I$. Furthermore, each project has a cost associated with the delay time cd_i. In the case of the RCMPSP-MT-W+RC, the costs associated with using global units cr^k are considered in addition to the parameters mentioned above. A feasible schedule for both variants consists of assigning start times to each job belonging to each project, satisfying the precedence relationships between jobs, and global and local resource units availability at each time period.

5.1.1 Mathematical formulation

The mathematical representation for both variants proposed can be defined based on the mathematical model described in Section 2.1.1.

The parameter a_k is removed, and the following parameters are added:

- ga^k Availability of units with global accessibility of resource type k at every period,
- la_i^k Availability of units with local accessibility of resources type k belonging to the project i,
- cd_i Cost of every time period delayed for project i,
- M_d Upper bound for delay time,

For the RCMPSP-MT-W+RC variant is required to add the following parameter:

cr^k Cost of using units with global accessibility of type resource k.

The following decision variables are defined:

G_{ij}^k Amount of units with global accessibility of resource type k allocated to job j of project i,

L_{ij}^k Amount of units with local accessibility of resource type k allocated to job j of project i.

$$D_i = \begin{cases} 1, & \text{if project } i \text{ is delayed,} \\ 0, & \text{otherwise.} \end{cases} \tag{5.1}$$

For the RCMPSP-MT-W the objective function (5.2) minimizes the weighted project delay (WPD) through the sum of the delay costs of each project. For each project, the difference between the completion time and the desired due date (DD_i) is calculated. If the project is delayed ($D_i = 1$), the difference is multiplied by the project delay cost (cd_i); otherwise, the difference is multiplied by zero.

$$\text{Minimize} \quad \sum_{i=1}^{|I|} \Big(\sum_{t=1}^{|T|} t \cdot X_{i|N_i|}^t - DD_i \Big) \cdot D_i \cdot cd_i \tag{5.2}$$

For the RCMPSP-MT-W+RC the objective function (5.3) minimizes the total project cost (TPC) through the sum of the delay costs (Equation (5.2)) and resource costs of each project. To calculate the cost of using units with global accessibility of each resource type, the consumption of units with global accessibility made by each job ($G_{ij}^k \cdot d_{ij}$) is multiplied by the cost of using units with global accessibility of this resource type (cr^k).

$$\text{Minimize} \quad \sum_{i=1}^{|I|} \Big(\big(\sum_{t=1}^{|T|} t \cdot X_{i|N_i|}^t - DD_i \big) \cdot D_i \cdot cd_i + \sum_{j=1}^{|N_i|} \sum_{k=1}^{|K|} G_{ij}^k \cdot d_{ij} \cdot cr^k \Big) \tag{5.3}$$

The following constraints must replace constraints (2.4) that ensure the satisfactory use of resources with global and local units.

$$\sum_{i=1}^{|I|} \sum_{j=1}^{|N_i|} \sum_{q=t}^{t+D_{ij}-1} G_{ij}^k \cdot X_{ij}^q \leq ga^k \ , \quad \forall\, k \in K, t \in T \tag{5.4}$$

$$\sum_{j=1}^{|N_i|} \sum_{q=t}^{t+D_{ij}-1} L_{ij}^k \cdot X_{ij}^q \leq la_i^k \ , \quad \forall\, i \in I, k \in K, t \in T \tag{5.5}$$

$$G_{ij}^{k} + L_{ij}^{k} = c_{ij}^{k} \ , \quad \forall\, i \in I, j \in N_i, k \in K \tag{5.6}$$

Constraints (5.4) guarantee that the global units allocation does not exceed the current availability of each resource type with global accessibility at each time period. Constraints (5.5) enforce that for each project, the local units allocation does not exceed the current availability of each resource type with local accessibility at each time period. Constraints (5.6) indicate that the allocation of global units added to the local units to each job of each project is enough to supply the corresponding demand.

To manage project delay times, it is required to add the following constraints:

$$\sum_{t=1}^{|T|} t \cdot X_{i|N_i|}^{t} - DD_i \leq D_i \cdot M_d \ , \quad \forall\, i \in I \tag{5.7}$$

When adding the aforementioned decision variables, it is required to add the following constraints to the model:

$$G_{ij}^{k} \geq 0 \ , \quad \forall\, i \in I, j \in N_i, k \in K \tag{5.8}$$

$$L_{ij}^{k} \geq 0 \ , \quad \forall\, i \in I, j \in N_i, k \in K \tag{5.9}$$

$$D_i \in \{0, 1\} \ , \quad \forall\, i \in I \tag{5.10}$$

5.2 Solution approach

A wide range of solution methods has been used to solve the RCMPSP variants that addressed features related to resource and project classifications. Following Section 2.2, the approximate algorithms are the most used to solve the RCMPSP variants, among them PR, SA, TS, and ILS. In addition, hybrid algorithms have been the most popular solution proposal in the last decade. The tabu search (TS, Glover and Laguna (1998)) is an algorithm included in several hybrid approaches presenting better performances. For example, the work of Tian et al. (2018) hybridizes GA with TS. The authors propose three algorithms to solve an RCMPSP variant that addresses planned resource unavailability and movable and unmovable resources in a real environment. The first algorithm consists of an improved serial heuristic algorithm based on priority rules (ISHPR), the second is based on the genetic-particle swarm (ISG-PS), and the third combines TS and genetic-particle swarm (ISG-PSTS). In the experimentation process, the authors show that the ISG-PSTS algorithm performs best to solve the studied variant. The hybridization of simulated annealing (SA, Kirkpatrick et al. (1983)) has also shown better performance when solving different variants of the RCMPSP. In this sense, Chen and Shahandashti (2009) propose the hybridization of SA and GA to solve an RCMPSP variant with generalized relationships. To compare the algorithm performance, the authors implement each algorithm separately, concluding that the hybridization of SA and GA presents better results in real-project environments. We also find works hybridizing SA and TS, presenting good results. For instance, He et al. (2021) develop three approaches to solve a problem variant that deals with the existence of global and local accessibility. The first algorithm is based on TS, the second on SA, and the third on hybridizing both algorithms TS-SA. The computational results show that TS-SA is the most promising for solving the problem under study. Given that the use of TS presents a good performance in the used algorithms to solve different variants of the RCMPSP and that good results were obtained from its hybridization with SA, in this research, we focus on studying the hybridization of SA and TS.

Among the essential aspects of SA is the generation of neighborhoods. In the related literature, many investigations perform this procedure by random movements. For example, Joo and Chua (2017) represent a solution through two strings, the sequence and the selected modes of all jobs and proposes two random moves, one that modifies the order of the sequence and another that changes the selected execution mode. On the other hand, priority rules (PRs) are the most used heuristics in the literature related to the RCMPSP, in most cases embedded within other heuristics (Browning and Yassine, 2010b; Chen and Shahandashti, 2009). Given the aforementioned, added to the fact that our research places greater emphasis on features related to resources, we use a random selection of resource types and priority rules to generate neighborhoods.

Another approach used as a framework for hybridizing algorithms is ILS (Lourenço et al., 2003). For example, Ahmeti and Musliu (2021) combine two hyper-heuristics under the ILS framework to solve an RCMPSP variant that considers multiple execution modes, global and local accessibility, and non-renewable resources. The authors integrate an exact approach, a constraint programming model (CP), with a local search-based algorithm that blends minimum conflicts algorithm and TS heuristics embedded in an iterated local search framework. The hybrid implementation consists of the sequential execution of the CP model and the meta-heuristics. The output solution of one method serves as the initial solution for the other. The perturbation strategy is based on a mode change of one job and reversing a short sequence of randomly selected jobs. By implementing the algorithm, the authors find new bounds for the studied variant in the experimentation process. In this research, we use the ILS procedure to integrate SA, TS, and PRs. The initial solution is generated using PR, while the perturbation strategy is based on sequence modifications by swapping randomly selected jobs.

The representation of a solution is described in Section 5.2.1. The ILS is the general hybridization structure (see Section 5.2.2), starting from an initial solution generated through PR (see Section 5.2.3), and exploring the search space by applying changes or perturbations (see Section 5.2.5). The local search procedure integrated into ILS is SA combined with TS and PRs, and its described in Section 5.2.4.

5.2.1 Representation

We use a triad vector to represent a solution (S) that indicates the start time sequences. Each triad comprises a job, a related project, and the start time. For example, in a case where the Job_{11} started its execution in period 1, and the job Job_{31} started its execution in period 2, the representation of the solution is composed as shown below:

$$S = \{\, (1,1,1),\, (1,3,2), \ldots, (|I|, |N_i|, |T|)\}$$

In addition, this algorithm relies on the G_a auxiliary structure to generate the neighbors in SA. This structure is represented by a hash table whose key is a pair $\langle i, k \rangle$, $i \in \{1, 2, \ldots, |I|\}$, $k \in \{1, 2, \ldots, |K|\}$ and the corresponding value is defined by a vector of integers T, $t \in \{1, 2, \ldots, |T|\}$, indicating the number of units with global accessibility of resource type k allocated to project j at instant t.

Moreover, we use a priority queue structure to represent the *tabu_list* as part of the SA approach. This data structure defines a triads of pairs of jobs (i, j), $i, j \in \{1, 2, \ldots, |I|\}$ and a resource type k, $k \in \{1, 2, \ldots, |K|\}$.

5.2.2 ILS-based general procedure

The general procedure receives as input the list of jobs from all projects (L_{jobs}), the maximum absolute due date of the projects ($maxT$), the minimum and maximum temperature (T_{max}) and (T_{min}), the number of iterations executed before making a temperature change (N_{iter}), and the temperature decrease coefficient (α). Furthermore, for the SA-based local search implementation, G_a stores the allocation of units with global accessibility to a project at each time period.

Algorithm 4 describes the general structure of our hybrid approach. First, in line 1, an initial solution to the problem is generated. In lines 2-3, the SA procedure is applied to the initial solution found, storing the result in S (representing the starting solution for the perturbation procedure) and in $gSol$ (representing the best global solution found). On the starting solution, in line 5, a perturbation procedure is carried out that promotes the exploration of the search space. In line 6, the SA procedure is applied based on the perturbed solution. In the lines 7-8, the best global solution found and the starting solution for the new iteration are updated. Based on perturbations and local search, this procedure is repeated for the number of restarts (N_{rest}) defined (lines 4-9). Finally, in line 10, the best global solution found ($gSol$) is returned.

Algorithm 4 ILS-based algorithm (L_{jobs}, N_{rest}, $maxT$, T_{max}, T_{min}, N_{iter}, α)

1: S_0 = Initial solution (L_{jobs})
2: S = SA (S_0, N_{iter}, $MaxT$, T_{max}, T_{min}, N_{iter}, α, G_a)
3: $gSol$ = S
4: **for** ($r = 0 \ldots N_{rest}$) **do**
5: $\quad S'_0$ = Perturbation (S)
6: $\quad S'$ = Simulated annealing (S'_0, N_{iter}, $MaxT$, T_{max}, T_{min}, N_{iter}, α, G_a)
7: $\quad gSol$ = Best of ($gSol$, S')
8: $\quad S$ = $gSol$
9: **end for**
10: **return:** $gSol$

5.2.3 Initial solution

The initial solution is generated through the first come-first served (FCFS) priority rule. Following the nomenclature presented in Section 2.1.1 and the research presented by Browning and Yassine (2010b), we define FCFS in our problem as $min(L_{ij})$.

Algorithm 5 describes the structure of the initial solution generation process. In line 1, the essential structures for the execution of the algorithm are initialized. Subsequently, for each period (lines 2-6), it is checked which jobs end their execution in line 3. These jobs are added to the initial solution (S_0), removed from the $execution$ list, and the units with global and local accessibility of each resource type used by finished jobs are released. In line 4, the candidate list is updated. The candidate jobs are added to the $candidates$ list and removed from the L_{jobs} list. A candidate job satisfies the condition that all its predecessors are fully executed; that is, they are in S_0. The start of the candidate jobs is made in line 5. The candidates are added to the $execution$ list, removed from the $candidates$ list, and the availability of existing resources is decreased by the amount corresponding to the job requirements. The start of a job candidate depends on the simultaneous availability of all resource units required by the job. In this case, the availability of resource units is verified by the sum of the currently available units with global and local accessibility of each resource type belonging to the project corresponding to the job candidate. In addition, it is established that the units with local accessibility are allocated first, and if required, the remaining demand is supplied with global units. Finally, in line 7, the generated solution (S_0) is returned.

Algorithm 5 Initial solution (L_{jobs}, $maxT$)

1: S_0, $candidates$, $execution$ = ∅
2: **for** ($time = 0 \ldots maxT$) **do**
3:     S_0 = End execution IS ($time$, $execution$, S_0)
4:     $candidates$ = Update candidates IS ($time$, L_{jobs}, $candidates$)
5:     $execution$ = Start execution IS ($time$, $candidates$, $execution$)
6: **end for**
7: **return:** S_0

5.2.4 Simulated annealing

The SA uses the initial solution described above and a movement combining the allocation of units with global accessibility and seven PR for neighborhood generation. The units with global accessibility allocated to each project at each period are stored in the structure G_a. The movement is defined as follows.

- Let:
  - P_l projects less delayed in the current solution,
  - P_m projects more delayed in the current solution,
  - k resource type randomly selected.
- Swap the allocation of units with global accessibility of resource type k between P_l and P_m.
- Build the neighborhood based on the modified allocation of units with global accessibility and the PRs.

Algorithm 6 describes the SA-based local search procedure. In line 1, the solution on which the local search is performed is stored as the best solution. In lines 2 and 11, the structures required for the algorithm are initialized. In line 5, the tabu list is updated. In lines 6 and 7, the most and least delayed projects and a random resource type are obtained. The swap of resource allocation is made in line 8. In line 12, a priority rule is randomly selected to generate the corresponding neighbor. The neighbor generation is made from lines 13-17. Similar to the procedure described in Section 5.2.3, for each period, in line 14, it is checked which jobs finish their execution. These jobs are added to S_{nb}, removed from the $execution$ list, and the units with global and local accessibility of each resource type used by finished jobs are released. In line 15, an update of the candidate list is performed. The candidate jobs are added to the $candidates$ list and removed from the L_{jobs} list. The start of candidate jobs is made in line 16. The candidate jobs are added to the $execution$ list, removed from the $candidates$ list, and the availability of existing resources is decreased in the amount corresponding to the job requirements. In this case, updating units with global accessibility is based on the modified resource allocation, which implies that the global units available for each resource type can differ at each period. Given this, for a job execution, it is essential to verify that the required units of each resource type are available during the job duration. If the solution quality is higher than the best obtained so far, in lines 18-21, the starting solution is updated, and both projects and the swapped resource type are added to the tabu list. If the quality is lower, the probability of acceptance of worse solutions is calculated, and a random number between 0 and 1 is generated in lines 23 and 24. If the generated number is less than the calculated probability, then the starting solution is updated, and both projects and the swapped resource type are added to the tabu list (lines 25-29). Once a transfer is made to a generated neighbor, the best solution is updated in lines 32 and 33. The temperature remains constant for a certain number of iterations before decreasing it in line 36. The procedure is carried out while the temperature is higher than the defined minimum temperature (lines 3-37). Finally, in line 38, the best solution found sol is returned.

Algorithm 6 SA (S_{start}, L_{jobs}, N_{iter}, $MaxT$, T_{max}, T_{min}, N_{iter}, α, G_a)

sol = S_{start}
$tabu\_list$ = $\varnothing$
while $(t > T_{min})$ **do**
  for $(i = 0 \ldots N_{iter})$ **do**
    Check $tabu\_list$
    $less$, $most$ = Least and most delayed project
    R = $random(1, \ldots, |K|)$
    G_a = Exchange (G_a, $less$, $most$, R)
    $flag$ = $true$
    while $(flag)$ **do**
      S_{nb}, $candidates$, $execution$ = $\varnothing$
      PR=$random(1, \ldots, 7)$
      for $(time = 0 \ldots MaxT)$ **do**
        S_{nb} = End execution SA ($time$, $execution$, S_{nb}, G_a)
        $candidates$ = Update candidates SA ($time$, L_{jobs}, $candidates$, PR)
        $execution$ = Start execution SA ($time$, $candidates$, $execution$, G_a)
      end for
      if $f(S_{nb}) < f(S_{start})$ **then**
        S_{start} = S_{nb}
        $tabu\_list.add$ ($less$, $most$, R)
        $flag$ = false
      else
        $rand$ = $random(0, 1)$
        $value$ = $e^{\Delta/q}$
        if $rand < value$ **then**
          S_{start} = S_{nb}
          $tabu\_list.add$ ($less$, $most$, R)
          $flag$ = false
        end if
      end if
    end while
    if $f(S_{start}) < f(sol)$ **then**
      sol = S_{start}
    end if
  end for
  t = $t \cdot \alpha$
end while
return: sol

Table 5.1 describes the seven PRs implemented in this work. Columns 1-2 present the acronym and description of the PR, respectively. Column 3 indicates the basis of the PR following the research presented by Browning and Yassine (2010b), where PRs can be based on jobs, projects, resources, relationships, or the union between them. Finally, column 4 reports the formula for calculating the corresponding PR. The nomenclature used is introduced in Section 2.1.1, and S_{ij} is the set of successors of job j of project i.

Table 5.1: Priority rules description.

Acronym	Description	Basis	Formula
SOF	Shortest operation first	Jobs	$min\,(d_{ij})$
MOF	Longest operation first	Jobs	$max\,(d_{ij})$
LR	Minimum resources usage	Resources	$min\,(\sum_{k=1}^{\|K\|} c_{ij}^{k})$
MR	Maximum resources usage	Resources	$max\,(\sum_{k=1}^{\|K\|} c_{ij}^{k})$
LRD	Less resources usage for complete execution	Jobs-Resources	$min\,\left((\sum_{k=1}^{\|K\|} c_{ij}^{k}) \cdot d_{ij}\right)$
MRD	Maximum resources usage for complete execution	Jobs-Resources	$max\,\left((\sum_{k=1}^{\|K\|} c_{ij}^{k}) \cdot d_{ij}\right)$
MS	Maximum total successors	Relationships	$max\,(\sum_{s=1}^{\|S\|} S_{ij})$

5.2.5 Perturbation procedure

The solution perturbation process in the proposed algorithm is based on the random relocation of a job within the existing solution. Then an update of the start times of each job and the resource consumption at each period is performed. We propose two perturbation intensities. The first is named simple perturbation, where only one job is swapped within the solution. The second is a joint perturbation, where one job of each project is swapped within the solution.

Since the simple perturbation is a special case of the joint one, Algorithm 7 describes the joint perturbation procedure. First, in line 2, a job is randomly selected. In lines 4 and 5, the maximum and minimum positions corresponding to the current solution of the predecessors and successors of the selected job are obtained. Then, a position is randomly selected between both values in line 6. In line 7, the movement of the selected job to the position obtained is carried out. The procedure is repeated as many times as projects exist. In each iteration, a job of a different project is selected in lines 1-8. Then, in line 9, structures required to recalculate the solution are initialized. The recalculation of the solution is made on lines 10-13. The procedure is similar to the one described in Section 5.2.3, but there is no candidate list since the execution order provided in the modified solution must be respected. For each time period, in line 11 it is checked which jobs have finished. These jobs are added to S_0, removed from the $execution$ list, and the units with global and local accessibility of each resource type used by finished jobs are released. In line 12, the jobs are added to the $execution$ list, removed from the S list, and the availability of existing resources is decreased by the amount corresponding to the job requirements. Finally, the output consists of the best solution found ($\mathcal{S}_0$) in all iterations.

Algorithm 7 Perturbation (S, $maxT$)

1: **for** $(i = 1 \ldots |I|)$ **do**
2: r_1 = $random(1, \ldots, |S|)$, ***where*** $(S[r_1].project \equiv i)$
3: j = $S[r_1].job$
4: R_{min} = highest position allocated in S by the predecessors of (i, j)
5: R_{max} = smallest position allocated in S by the successors of (i, j)
6: r_2 = $random(R_{min}, \ldots, R_{max})$
7: S = Movement (S, r_1, r_2)
8: **end for**
9: S_0, $execution$ = $\varnothing$
10: **for** $(time = 0 \ldots MaxT)$ **do**
11: S_0 = End execution P ($time$, $execution$, S_0)
12: $execution$ = Start execution P ($time$, S, $execution$)
13: **end for**
14: **return:** S_0

5.3 Experimentation results

The proposed algorithm was implemented in C++ and tested on Intel(R) Xeon(R) processor E3-1220, 3.10 GHz with 32GB RAM memory on a CentOS Linux 7.6.18.19 system. Our approach was validated through the modified instances proposed in MPSPLib[1]. Also, we carried out a 100 run for each instance using the values of the parameters indicated in Table 5.3.

[1] http://www.mpsplib.com/

5.3.1 Problem instances

To validate our approach to solve both variants proposed, we selected and modified 36 instances of the MPSPLib (see description in Section 2.3). In MPSPLib, for each resource type, the available units with global or local accessibility for each time period are defined. Given this, to configure the set of instances with resources mixed accessibility[2], we modified, for each resource type, the available units with global and local accessibility. Moreover, to preserve the stability of resource availability, for each resource type, the units with global accessibility were uniformly distributed among units with local accessibility of each project and units with global accessibility. In addition, the resource costs were added following the research presented by Zhang and Chen (2018), where, based on the collected data and experts' experience, the costs of renewable resources are defined as random integers between 1 and 8.

The instances selection was carried out uniformly concerning the number of projects (**_a**), jobs (**_j**), and different among projects (**_nr**). Table 5.2 shows the used instances. Column 1 indicates the instance's name, while column 2 shows the reference MPSPLib instance. Column 3 indicates the number of resource types that have mixed availabilities. The total number of successors of the instance is reported in column 4. The average resource required by a job for each resource type is presented in the columns 5-8. Finally, the average resource cost is shown in column 9.

Table 5.2: Principal components related to the used instance set.

Instance	Reference	# Mixed	# Succ.	R_1	R_2	R_3	R_4	Avg. cost
mpma_j30_i2_1	mp_j30_a2_nr1	2	96	5.00	6.47	5.53	5.86	1.50
mpma_j30_i2_2	mp_j30_a2_nr3	2	96	0.75	1.51	1.48	1.86	5.00
mpma_j30_i2_3	mp_j30_a2_nr5	1	116	4.86	3.86	3.72	3.50	8.00
mpma_j30_i5_1	mp_j30_a5_nr1	2	340	1.22	1.25	1.59	1.22	5.50
mpma_j30_i5_2	mp_j30_a5_nr3	2	290	3.27	3.37	3.82	3.59	7.50
mpma_j30_i5_3	mp_j30_a5_nr5	1	290	3.15	2.44	2.78	2.85	8.00
mpma_j30_i10_1	mp_j30_a10_nr1	2	480	5.47	4.63	4.25	5.97	5.00

(Continued on next page)

[2]All RCMPSP-MT-W+RC used instances are online available at: https://github.com/mariamgmez/rcmpsp_mt_w_rc

Table 5.2 – *(Continued from previous page)*

Instance	Reference	# Mixed	# Succ.	R_1	R_2	R_3	R_4	Avg. cost
mpma_j30_i10_2	mp_j30_a10_nr3	2	480	4.53	4.73	5.02	5.18	5.00
mpma_j30_i10_3	mp_j30_a10_nr5	1	680	2.44	2.60	2.43	2.37	3.00
mpma_j30_i20_1	mp_j30_a20_nr1	2	1360	2.66	3.50	5.59	4.53	5.50
mpma_j30_i20_2	mp_j30_a20_nr3	2	960	4.90	4.97	4.79	4.92	2.00
mpma_j30_i20_3	mp_j30_a20_nr5	1	1360	4.16	4.26	4.33	4.09	3.00
mpma_j90_i2_1	mp_j90_a2_nr1	2	332	2.52	3.02	2.49	2.87	5.00
mpma_j90_i2_2	mp_j90_a2_nr3	2	332	2.73	2.93	2.19	3.48	3.00
mpma_j90_i2_3	mp_j90_a2_nr5	1	332	1.49	1.46	1.42	1.29	4.00
mpma_j90_i5_1	mp_j90_a5_nr1	2	830	5.68	5.43	5.47	4.78	4.50
mpma_j90_i5_2	mp_j90_a5_nr3	2	914	3.04	3.14	2.91	2.81	6.00
mpma_j90_i5_3	mp_j90_a5_nr5	1	830	3.04	3.06	2.94	2.77	8.00
mpma_j90_i10_1	mp_j90_a10_nr1	2	1940	2.51	3.02	2.72	2.84	3.50
mpma_j90_i10_2	mp_j90_a10_nr3	2	1940	5.39	5.25	5.32	5.29	4.00
mpma_j90_i10_3	mp_j90_a10_nr5	1	1912	4.97	4.84	5.02	5.03	4.00
mpma_j90_i20_1	mp_j90_a20_nr1	2	3320	2.65	2.96	2.96	2.11	7.50
mpma_j90_i20_2	mp_j90_a20_nr3	2	3152	2.82	2.75	2.61	2.80	3.00
mpma_j90_i20_3	mp_j90_a20_nr5	1	3460	4.21	4.21	4.22	4.19	6.00
mpma_j120_i2_1	mp_j120_a2_nr1	2	514	1.53	1.06	1.40	1.15	2.00
mpma_j120_i2_2	mp_j120_a2_nr3	2	514	5.39	5.48	5.39	5.04	2.50
mpma_j120_i2_3	mp_j120_a2_nr5	1	514	2.07	2.45	2.02	2.03	6.00
mpma_j120_i5_1	mp_j120_a5_nr1	2	915	0.89	1.16	1.84	0.97	4.50
mpma_j120_i5_2	mp_j120_a5_nr3	2	1100	5.41	5.38	5.47	5.38	4.00
mpma_j120_i5_3	mp_j120_a5_nr5	1	1248	4.72	4.71	4.65	4.49	2.00

(Continued on next page)

Table 5.2 – *(Continued from previous page)*

Instance	Reference	# Mixed	# Succ.	R_1	R_2	R_3	R_4	Avg. cost
mpma_j120_i10_1	mp_j120_a10_nr1	2	1830	1.06	1.86	1.05	1.00	6.00
mpma_j120_i10_2	mp_j120_a10_nr3	2	2570	1.42	1.35	1.29	1.35	5.50
mpma_j120_i10_3	mp_j120_a10_nr5	1	2570	2.85	2.77	2.75	2.69	1.00
mpma_j120_i20_1	mp_j120_a20_nr1	2	3660	0.89	1.16	1.83	0.97	7.00
mpma_j120_i20_2	mp_j120_a20_nr3	2	4030	3.15	3.17	3.18	3.15	2.50
mpma_j120_i20_3	mp_j120_a20_nr5	1	4770	3.29	3.30	3.41	3.36	1.00

5.3.2 Parameter analysis and setting

We perform a parameter tuning using ParamILS (Hutter et al., 2009), considering some parameter values used in the literature for ILS, SA, and TS algorithms for solving RCMPSP and RCPSP variants. Moreover, to provide suitable lower and upper limits of the ranges to the tuner, the following aspects are considered:

- Number of restarts and a number of iterations: higher values imply a greater possibility of finding better solutions, but they influence runtime considerably.
- Maximum and minimum temperature: high or small values generate a greater or lesser probability of transition towards solutions of lower quality, respectively.
- Temperature decrease coefficient: higher values imply a slight decrease in temperature, while lower values promote rapid cooling.
- Tabu list size: must be consistent with the number of existing projects.

Table 5.3 shows the interval of values provided to ParamILS and the optimal parameter configuration found by the tuner.

Table 5.3: Parameter setting results.

Parameter	Description	Value	Interval
N_{rest}	Number of restarts of ILS	50.0	$[10, 50]/10$
N_{iter}	Number of iterations before temperature change	50.0	$[10, 50]/10$
T_{max}	Maximum temperature	10000.0	$[1000, 10000]/1000$
T_{min}	Minimum temperature	1.0	$[1, 500]/100$
α	Temperature decrease coefficient	0.9	$[0.1, 0.9]/0.1$
TL_{size}	Tabu list size	$\|I\|/4$	$[\|I\|/2, \|I\|/3, \|I\|/4]$

5.3.3 Results

In this section, we analyze different aspects based on the results of implementing the mathematical models and the hybrid algorithm in the studied problem variants. Section 5.3.3.1 analyzes the results of implementing the proposed mathematical models. Section 5.3.3.2 analyzes the results provided for both variants concerning the influence of the perturbation intensity applied as part of the ILS. Given that the resource costs considerably influence the use of resources, to study the influence of resource accessibility, Section 5.3.3.3 analyzes the results provided for the RCMPSP-MT-W variant. Finally, Section 5.3.3.4 compares the results of both proposed variants.

5.3.3.1 Mathematical formulations analysis

The proposed mathematical formulations (see Section 5.1.1) provide feasible results only to the instance mpma_j30_i2_2 reporting a Gap(%) value of zero. Achieving feasible results is not possible for the remaining test cases within a time limit of 2 hours. The values provided for total makespan (M), average project delay (APD), and weighted project delay (WPD) for both variants studied are 65, 2.5, and 70, respectively. While for the RCMPSP-MT-W+RC the cost for the use of resources obtained (RCP) is 120, incurring a total cost (TC) of 190. Several factors can influence the behavior shown above. The instances mpma_j30_i2_1 and mpma_j30_i2_2 present the least number of successors, which implies fewer precedence constraints and, therefore, less resolution complexity because the number of constraints decreases (see Table 5.2).

Also, both instances present the same number of units with both global and local accessibility. However, when analyzing the average units required per job of each resource type, the average values related to the mpma_j30_i2_2 instance represent a quarter of the requirements of the mpma_j30_i2_1 instance. In addition, the overall average resource availability is higher by 4.5 units for the instance mpma_j30_i2_2 with a value of 41.25. These values generate less resolution complexity for the instance mpma_j30_i2_2 because it reports lower requirements and greater resource availability. Similarly, the remaining instances show high values for the number of successors and resource requirements.

5.3.3.2 Assessment of perturbation intensity

To analyze the influence of the perturbation intensity applied as part of the ILS, Table 5.4 and Table 5.5 show the results after applying our hybrid approach with simple and joint perturbation on the selected instances for RCMPSP-MT-W and RCMPSP-MT-W+RC, respectively. The results shown correspond to the averages for the instances of the same composition. The results found for each instance are shown in Appendix B, Table B.1, and Table B.2.

In Table 5.4, Column 1 shows the instance composition (Comp). The column values indicate #jobs_#projects. Columns 2-5 and 6-9 report the execution times, M, APD, and WPD provided for both perturbation intensities. In the case of Table 5.4, columns 10-12 indicate for M, APD, and WPD, the improvement percentages obtained applying joint perturbation with respect to the application of simple perturbation. In Table 5.5, columns 6-7 and 12-13 show the costs associated with RPC and TC values. In our study, the weight of the projects is defined as the penalty cost for delays, so the total cost corresponds to the sum of the cost for delays together with the cost for the use of resources. Finally, columns 14-16 in Table 5.5 indicate for M, APD, WPD, and TC the improvement percentages obtained applying joint perturbation with respect to the application of simple perturbation.

The values of improvements are calculated through the following formula:

$$Improvement(\%) = (Results_j - Results_s)/Results_s \cdot 100\% \tag{5.11}$$

where $Results_j$ and $Results_s$ correspond to the results provided for the different measures applying simple and joint perturbation, respectively.

For the RCMPSP-MT-W variant, the results show that, on the general average, a greater perturbation intensity (application of joint perturbation) positively influences the solutions found.

Table 5.4: Comparison of components performance of our approach. RCMPSP-MT-W.

Comp	Simple perturbation				Joint perturbation				Improvement		
	T_s (seg)	M_s	APD_s	WPD_s	T_j (seg)	M_j	APD_j	WPD_j	M_i	APD_i	WPD_i
30_2	4.46	72.67	8.83	155.00	4.62	72.67	8.83	155.00	0.00	0.00	0.00
30_5	23.02	91.67	17.27	247.53	26.15	91.67	17.27	247.53	0.00	0.00	0.00
30_10	46.90	107.33	32.10	366.10	56.40	108.33	31.87	365.77	0.97	**-1.15**	**-0.13**
30_20	122.41	149.00	48.67	773.67	150.06	150.00	48.33	762.00	0.56	**-0.68**	**-1.20**
90_2	33.55	134.33	24.00	785.33	36.12	129.33	20.33	670.00	**-3.01**	**-14.14**	**-13.93**
90_5	100.51	159.00	36.47	2120.20	116.05	157.00	35.27	2062.60	**-1.28**	**-6.19**	**-4.98**
90_10	674.68	320.33	142.00	4739.00	730.89	316.67	138.67	4626.13	**-1.42**	**-4.85**	**-5.01**
90_20	706.78	215.67	46.00	2138.33	751.83	210.00	43.33	869.33	**-3.44**	**-24.02**	**-46.77**
120_2	84.05	174.67	54.33	4520.33	89.63	174.67	54.00	4481.67	0.00	**-0.28**	**-0.34**
120_5	153.25	205.67	83.33	5143.67	176.00	203.67	81.33	5073.67	**-2.20**	**-20.00**	**-20.00**
120_10	277.75	175.00	46.00	1616.33	306.16	173.67	45.00	1591.33	**-1.40**	**-3.85**	**-2.62**
120_20	486.62	198.33	56.33	2927.33	586.04	198.33	54.67	2870.00	0.00	**-16.67**	**-16.38**
Avg.	226.16	166.97	49.61	2127.74	252.50	165.50	48.24	1981.25	**-0.94**	**-7.65**	**-9.28**

Table 5.5: Comparison of components performance of our approach. RCMPSP-MT-W+RC.

Comp	Simple perturbation						Joint perturbation						Improvement		
	T_s (seg)	M_s	APD_s	WPD	RPC	TC	T_j (seg)	M_j	APD	WPD	RPC	TC	M_i	APD_i	TC
30_2	5.43	78.33	13.17	187.33	549.00	736.33	7.07	82.67	17.00	213.17	497.00	710.17	4.56	15.33	**-2.49**
30_5	28.64	95.33	20.80	298.33	991.33	1289.67	31.25	96.00	21.60	303.20	911.00	1214.20	1.50	23.16	**-10.89**
30_10	52.67	115.00	37.47	417.87	758.67	1176.53	57.18	109.33	35.17	392.17	762.67	1154.83	**-3.47**	**-5.03**	**-1.41**
30_20	132.38	154.33	51.72	809.60	450.67	1260.27	130.79	152.67	50.75	786.82	458.33	1245.15	**-1.45**	**-2.86**	**-1.10**
90_2	37.16	136.00	25.00	819.33	515.00	1334.33	43.90	135.67	24.83	814.00	532.33	1346.33	**-0.26**	**-2.22**	1.10
90_5	111.05	163.67	42.27	2438.07	4372.00	6810.07	133.88	162.67	41.20	2373.60	4252.67	6626.27	**-0.60**	**-1.10**	**-1.56**
90_10	763.75	321.33	143.40	4767.57	2887.00	7654.57	776.36	316.67	139.03	4626.37	2805.67	7432.03	**-1.93**	**-6.18**	**-3.45**
90_20	900.70	220.67	47.87	2211.22	2564.67	4775.88	919.55	213.33	47.52	2149.10	2545.00	4694.10	**-2.73**	**-2.96**	**-1.81**
120_2	100.41	175.33	55.33	4543.50	3223.67	7767.17	109.76	174.67	54.17	4462.83	3275.00	7737.83	**-0.47**	**-6.14**	**-2.81**
120_5	165.76	207.33	85.20	5190.00	3232.00	8422.00	171.05	202.00	86.27	5320.13	3019.00	8339.13	**-1.43**	9.69	**-5.88**
120_10	316.80	176.33	47.27	1654.43	1535.67	3190.10	357.05	174.67	46.57	1628.53	1518.33	3146.87	**-1.53**	**-2.41**	**-1.56**
120_20	790.07	198.67	56.97	2939.05	1230.33	4169.38	850.42	197.67	56.98	2951.38	1188.00	4139.38	0.59	**-2.41**	**-1.87**
Avg.	283.73	170.19	52.20	2189.69	1859.17	4048.86	299.02	168.17	51.76	2168.44	1813.75	3982.19	**-0.60**	1.41	**-2.81**

When analyzing the averages by a group of similar composition, we found that for the values of APD and WPD, joint perturbation provides the same or better results than simple perturbation, obtaining improvements of up to 46% in the group 90_20. For the values of M, in 83% of the cases the joint perturbation is equal or better than the simple perturbation. Only two groups show better values using simple perturbation, i.e., 30_10 and 30_20. Nevertheless, these improvements are less than 1%.

The results of the RCMPSP-MT-W+RC with resource costs variant show a similar behavior where the joint perturbation application finds better results than simple perturbation in 92% of the cases for the TC value, reaching improvements of up to 10% in the group 30_5. This situation is reflected in 66% of the cases for M and APD. Concerning execution times, joint perturbation shows higher values for both variants in most cases since the job relocation process is carried out as many times as there are projects. In addition, considering costs associated with resource usage requires longer execution times for both intensities of disturbances, with respect to considering only delay costs.

Based on the above results, we use the Friedman-k Related Samples test to determine if there are significant differences between the results provided by applying simple and joint perturbation for the measures described for both variants. Our null hypothesis is, in each case, that there are no significant differences between the distributions. In contrast, the alternative hypothesis states that there are significant differences between them. Table 5.6 summarizes the results provided.

The value corresponding to $\mathcal{X}^2$ for 1 degree of freedom and a significance level of $\alpha = 0.05$ is 3.84. Given this, for the case of RCMPSP-MT-W, the null hypothesis is not rejected for M. On the other hand, in the cases of APD and WPD, the null hypothesis is rejected, so it is considered that there are significant differences between the samples. For the RCMPSP-MT-W+RC, there are no significant differences for M and APD, but there are for TC. Rows 5-6 of Table 5.6 show the average ranges for each case for the Friedman-k Related Samples test. The results show that for each case, despite the existence or not of significant differences, the best-performing values are provided by joint perturbation.

For RCMPSP-MT-W, in most cases where the application of joint perturbation represents an improvement respecting WPD, the corresponding values of M and APD are also lower. However, several instances have no improvements in M values. That is because lower WPD is found through the decrease in the completion time of one or several projects, which does not directly

Table 5.6: Statistical comparison between simple and joint perturbation using Friedman-k Related Samples test.

	RCMPSP-MT			RCMPSP-MT-RC		
	M	APD	WPD	M	APD	TC
Total N	36.00	36.00	36.00	36.00	36.00	36.00
Contrast statistic	2.57	11.27	12.25	1.96	3.00	9.14
Degrees of freedom	1.00	1.00	1.00	1.00	1.00	1.00
Range simple perturbation	1.58	1.68	1.69	1.60	1.63	1.72
Range joint perturbation	1.42	1.32	1.31	1.40	1.38	1.28

influence M but rather the corresponding delay values. As a unique case, the mpma_j90_i20_3 instance decreases M but shows a higher APD. These values indicate that the solution found by joint perturbation presents lower completion time values of one or several projects that directly influence the M value present in the solution found by simple perturbation but presents longer completion times in other projects.

For RCMPSP-MT-W+RC, in 17% of the cases, the decrease in TC is not related to the decrease in M and/or APD but to the decrease in RPC. This occurs because fewer units with global accessibility are being consumed, but greater delays are incurred by executing only some jobs even though global units are available. On the other hand, the instances mpma_j30_i10_1, mpma_j90_i2_3, mpma_j120_i2_3, and mpma_j120_i5_2 present decreases in the values of M and/or APD, but not better values of TC are provided. This behavior may result from greater use of units with global accessibility, which decreases the execution time of the projects and sometimes the delay but increases the costs for resource use and, therefore, the total costs.

The solutions provided for the instance mpma_j90_i2_3 are used to analyze the differences between the solutions found by simple and joint perturbation for RCMPSP-MT-W. In this instance, joint perturbation achieves lower M, APD, and WPD values. This behavior can be related to differences in the job start order and greater use of units with global accessibility. This can be noticed when comparing Figures 5.2 and 5.3 where the global units allocations to each project are shown, applying simple and joint perturbation, respectively.

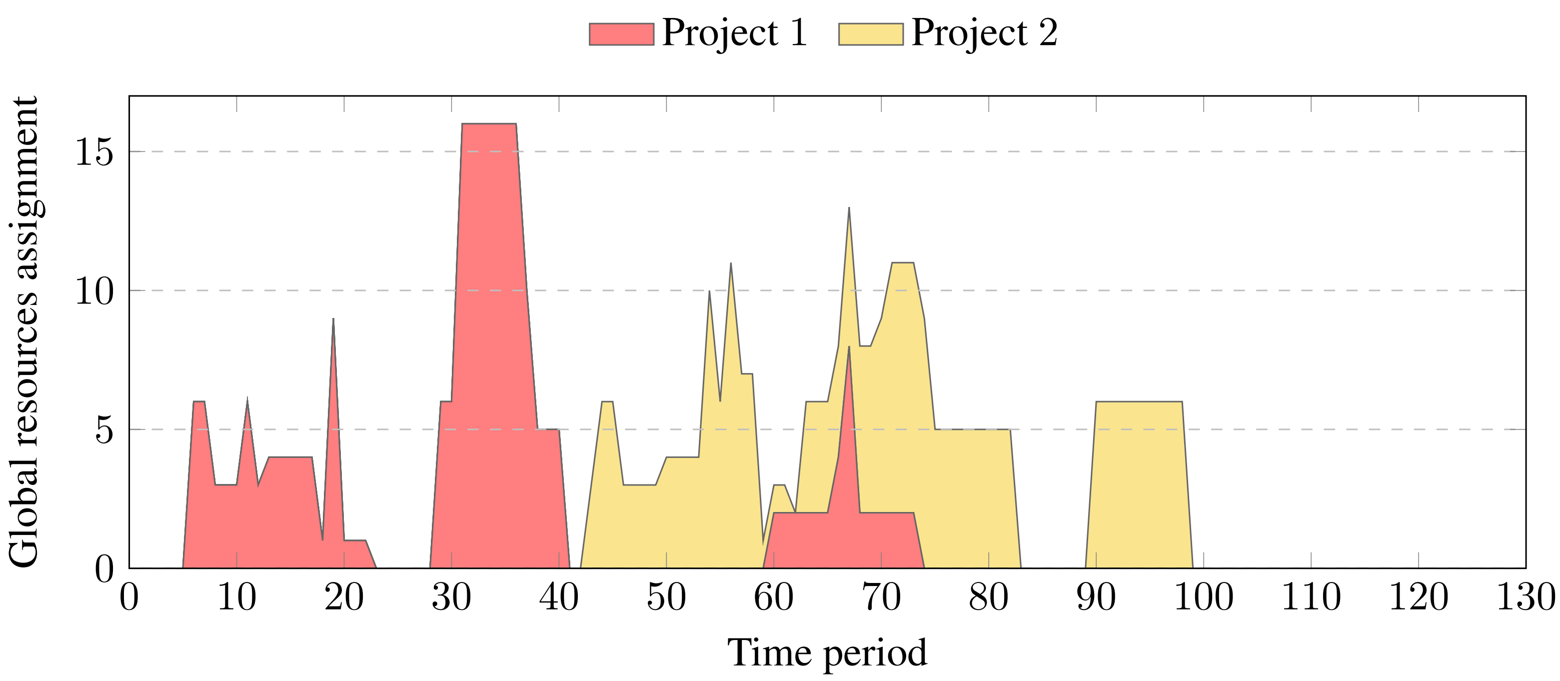

Figure 5.2: Illustration of resource consumption applying the simple perturbation procedure in RCMPSP-MT-W for instance mpma_j90_i2_3.

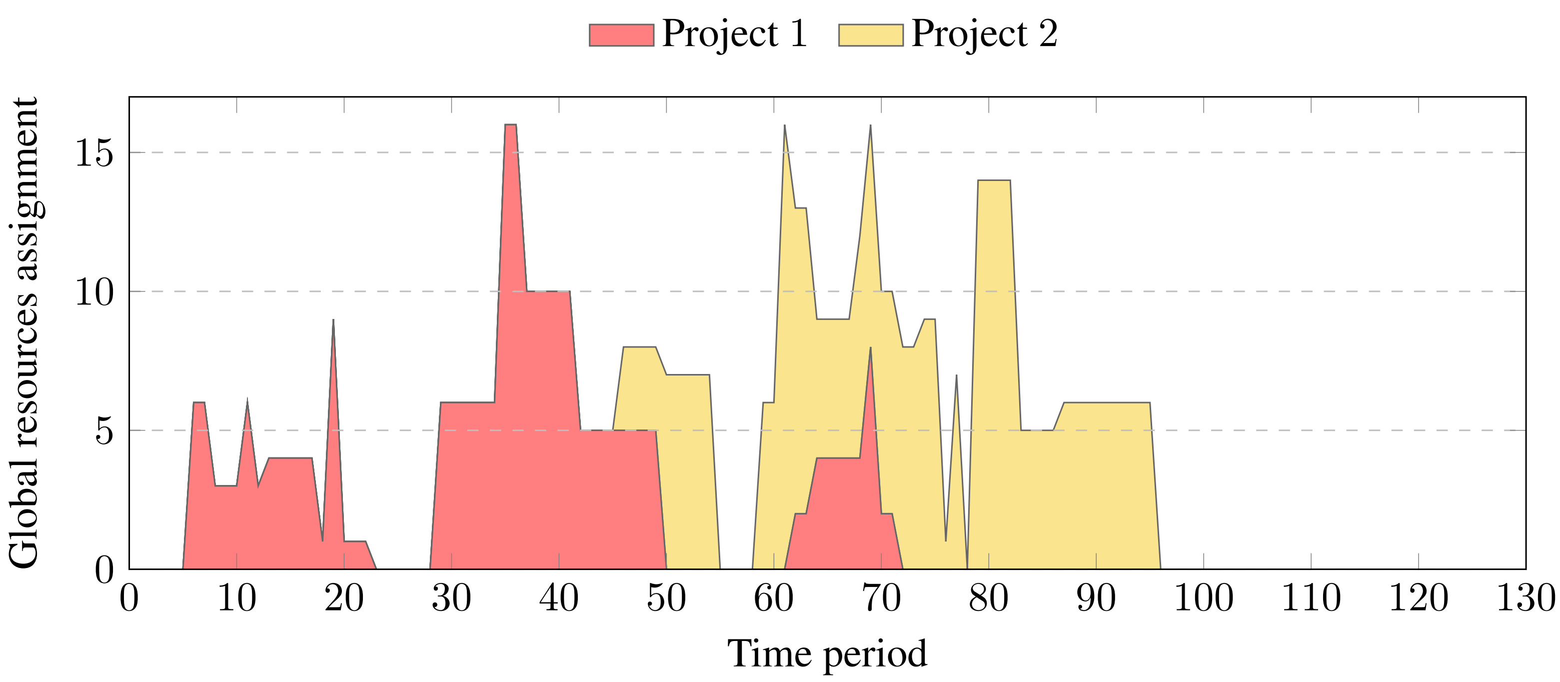

Figure 5.3: Illustration of resource consumption applying the joint perturbation procedure in RCMPSP-MT-W for instance mpma_j90_i2_3.

5.3.3.3 Assessment of resource accessibility

The accessibility of the resources can represent differences when comparing different solutions within a given scenario. According to the features described in Section 3.1 corresponding to the resource accessibility subcategory and the new features analyzed in this study, we explore three cases. The case C_1 corresponds to the existence of units with mixed accessibility. Case C_2 represents when a resource type only has global units, while in the case C_3, the resource only has local units available.

We selected a subset of three instances (1 small, 1 medium, and 1 large) for this analysis.

The instances are mpma_j30_i2_3, mpma_j90_i2_3, and mpma_j120_i2_3 composed of two projects and 30, 90, and 120 jobs per project, respectively. The average resource consumption of the selected instances is between 3.50 and 4.86, 1.29 and 1.49, and 2.02 and 2.45 units, respectively. These instances only present one type of resource with units of mixed accessibility (see Table 5.2). For each instance, we modify the availability of global and local units to create the three cases presented in Table 5.7. Column 1 shows the case identifier. Column 2 presents the global unit availability, while columns 3 and 4 indicate the local unit availability belonging to projects 1 and 2, respectively. Finally, columns 5-7 report the M, APD, and WPD values obtained. We apply joint perturbation as part of the ILS algorithm to provide these values.

Table 5.7: Different cases of resource accessibility.

Instance	Case	Global availability	P_1 availability	P_2 availability	M	APD	WPD
mpma_j30_i2_3	C_1	5	10	10	74	12.0	311
	C_2	25	0	0	70	9.0	235
	C_3	0	13	12	76	15.0	382
mpma_j90_i2_3	C_1	16	8	8	123	5.0	160
	C_2	32	0	0	123	5.0	160
	C_3	0	16	16	123	5.0	160
mpma_j120_i2_3	C_1	20	10	11	142	28.0	874
	C_2	41	0	0	128	20.0	644
	C_3	0	20	21	142	32.5	1058

As can be seen, case C_2 reports the best performance because the access to the analyzed resource type is global, allowing the available global units in each period to be used for job execution corresponding to P_1 and P_2. In the case of C_3, the values of the measures increase considerably because the accessibility of the resources is local, so in several periods, the units belonging to P_1 cannot be used for the execution of jobs corresponding to P_2, and vice versa. The resources mixed accessibility is represented by C_1, where lower values are found than those obtained in C_3 but higher than those found in C_2, which indicates that in some periods, the global units are not enough to supply the existing demand, while there are local units that are not being used.

For the mpma_j90_i2_3 instance, this behavior is not reflected in the values of M, APD, and WPD since these values refer to the general terms of the instance. Note that the values of AT_1 and AT_2 for this instance are 0 and 20, respectively, and the resource utilization rates of both projects, especially P_2, are low. In the results provided for the three cases, P_2 is the last project to complete the execution of its corresponding jobs in the time period 123, presenting a delay of 10 units. This situation might indicate that the sequence of the starting times of jobs belonging to P_2 concerning the use of the resource k_2 is defined by the precedence relations between the jobs and not by the non-existence of resource units. To check this, we ran the algorithm assuming that the jobs belonging to P_2 do not require resources of type k_2, and we got the same results.

On the other hand, P_1 does not present a delay in any of the proposed cases, finishing the execution of its jobs 4, 12, and 3 time periods before DD_1, for the cases C_1, C_2, and C_3, respectively. This can indicate that despite actively using global units, P_1 has enough slack concerning DD_1, which allows it to delay the start execution of some jobs without incurring delay penalties. Nevertheless, when analyzing the finish times of P_1 in the three cases, we find the same behavior for the mpma_j30_i2_3 and mpma_j120_i2_3.

Given that the solutions provided result from the approximate algorithm, we cannot ensure that this behavior is due only to differences in the accessibility of resources. Still, global access to resources may imply additional assignment freedom, allowing better results.

5.3.3.4 Cost analysis

This section analyzes how the costs associated with the use of global resources influence the projects' delays in the solutions found. To carry out the analysis, we compare the results provided for both studied variants. The results shown in Table 5.8 correspond to the averages for the instances of the same composition. The results found for each instance are shown in Appendix B, Table B.3. Table 5.8 column 1 shows the instance composition (Comp). The column values indicate #jobs_#projects. Columns 2-3 and 4-5 show the differences between the values of M and ADP obtained by applying simple and joint perturbation for RCMPSP-MT-W and RCMPSP-MT-W+RC, respectively. The differences are calculated with the following formula:

$$\textit{Difference} = (Results_{\text{RCMPSP-MT-W+RC}} - Results_{\text{RCMPSP-MT-W}}) \tag{5.12}$$

Table 5.8: Differences in M and APD caused by resource costs.

Comp	M_s	APD_s	M_j	APD_j
30_2	5.67	4.33	10.00	8.17
30_5	3.67	3.47	4.33	4.27
30_10	7.67	5.03	1.00	3.30
30_20	5.33	2.78	2.67	2.42
90_2	1.67	1.00	6.33	4.50
90_5	4.67	5.53	5.67	5.93
90_10	1.00	1.30	0.00	0.37
90_20	5.00	1.87	3.33	4.18
120_2	0.67	0.67	0.00	0.17
120_5	1.67	1.33	-1.67	4.93
120_10	1.33	1.03	1.00	1.33
120_20	0.33	0.33	-0.67	2.31
Avg.	3.22	2.39	2.67	3.49

As can be seen, considering resource usage costs generates, on average, higher values of M and APD. This is since when considering resource usage costs, an attempt is made to reduce the use of global units, generating higher values of M and APD and decreasing the use of global units.

However, there are several instances where the values of M are lower when resource usage costs are considered. This situation can be generated because the objective involves resource usage costs and project delay costs. Therefore, while attempting to minimize the use of global units, we attempt to minimize the times related to M, prioritizing the resource units allocation to those cases that generate a delay in the desired completion date of the projects. To achieve this reduction in completion times, consuming more global units at each time period is required, so there is a close relationship between the two mentioned costs. If a very high cost due to delay is considered, an attempt will be made to consume a greater number of global units, even if it is necessary to incur costs for the use of these resources, and at the same time, if the resource usage costs are high, an attempt will be made to reduce the use of global units, even if it results in higher project delay costs.

5.4 Conclusions

In this study, we proposed two RCMPSP variants named RCMPSP-MT-W and RCMPSP-MT-W+RC. Both variants consider resources with mixed accessibility, total allocation, and delay costs for each project. In addition, the second variant also considers resource cost, one of the most studied features in the literature and practice. Both variants are described, and the corresponding mathematical formulations are provided.

We proposed a hybrid algorithm to solve the presented variants that integrated priority rules, simulated annealing, and tabu search as part of an iterated local search. The approach analyzes two perturbation intensities: simple and joint perturbation. To assess the performance of the developed approaches, 36 instances were proposed. The instances were based on the MPSPLib library modifying the availability of global and local units and adding the cost for resource usage.

Given the complexity of the test instances, the proposed mathematical models were only able to find the optimal solution in the instance with the lowest number of successors, the lowest average resource requirement, and the highest resource availability. On the other hand, when analyzing the results obtained from the hybrid algorithm with both perturbation intensities, we found that the application of joint perturbation yields better results with respect to simple perturbation for the proposed instances. Furthermore, we study how the accessibility of resources and the cost can influence the solutions provided. Through the analysis, we conclude that, although it may not be the only factor, the global accessibility of resources may imply greater allocation freedom, allowing better results.

Chapter 6

Conclusions

This thesis presents a study of the resource-constrained multi-project scheduling problem (RCMPSP) and its variants. The basic problem is initially described through its mathematical formulation and an illustrative example. Through a detailed review of the state-of-the-art, we found that the approximate algorithms were the most used for the resolution of the RCMPSP-related problems, highlighting the use of genetic algorithms, priority rule-based algorithms, multi-agent systems, and combinatorial auctions. We also identify that the algorithms mentioned are the most used since 2010, being also noticeable for the use of hybrid algorithms for the resolution of the proposed variants. When analyzing the existing benchmarks, we found that 49% of the investigations were carried out on sets of instances proposed explicitly for the variant studied. This is due to the vast number of variants of the problem that have yet to recognize instance sets. We also identified that 16% of the works studied real cases and that MPSPLib is the most recognized library in the literature since the structure of the proposed instances allows its use to study several variants of the problem.

According to the nature of the problem, the investigations collected in the review were analyzed regarding features and objective functions. Taxonomies for the features and objective functions found were proposed to unify concepts and visualize the current state of RCMPSP-related variants. The 34 features in the literature were classified based on jobs, resources, relationships, and projects. In contrast, the 20 objective functions pursued were classified concerning resources, projects, and time. The features and classified objectives were also described in depth and analyzed in terms of the evolution of the problem and the number of existing studies. The analysis showed that the most used assumption is to solve the problem under deterministic conditions.

Despite this, the last decade presented remarkable growth in the study of uncertainty associated with job duration. The latest research lines mainly focus on studying decentralized environments, local and non-renewable resources, multiple execution modes, and weights for each project. This led to the increase of decentralized solution methods such as multi-agent systems and combinatorial auctions and the development and use of the MPSPLib library as a standard benchmark to validate and compare results. On the other hand, time- and resource-based objective functions were the most studied, with the minimization of the makespan, average project delay, and total project cost being the most predominant objectives. The works applied to real environments were also analyzed. In this sense, in most cases, the complexity of the real instances is less than the complexity of the instances proposed in the literature, evidenced by the number of projects, jobs per project, availability, and resource requirements. We also identified that only 40% of the objectives studied in the literature are pursued in real environments. The most used objectives are those related to the time and resource categories, highlighting makespan minimization and the total project cost minimization. On the other hand, 62% of the features described in the literature have been addressed in real environments. The most studied features are those related to jobs, resources, and project categories, highlighting the study of multiple execution modes, non-renewable resources, and the different weights of the projects.

Among the most studied features of the problem is the use of local resources. Given this, we carried out a deeper study of the RCMPSP-L, a variant of the problem that addresses the features of the base problem together with the existence of resources with local accessibility belonging to each project. Initially, a description of the problem is presented through its mathematical formulation and an illustrative example. Then, a hybrid algorithm is proposed to solve the variant that integrates ant colony optimization with a hill-climbing first-improvement algorithm. The experimentation process was carried out using a set of instances taken from the MPSPLib, a library widely used in the literature, to validate solutions for several problem variants. When analyzing the results obtained, we find that the proposed local search always improves our constructive algorithm. Despite this, the greatest strength of the algorithm is found in the construction phase since it provides good-quality solutions. Furthermore, we found that the proposed algorithm presents low variability values in 100 runs, despite the significant differences in the resolution complexity of the selected instances. Finally, the results obtained were compared with others well-known in the literature, evidencing that for different existing metrics, our approach presents competitive results. Even though our problem does not cover decentralized resource management, the results are compared with some approaches that study this feature and others that do not. Regarding both centralized and decentralized approaches, the proposed algorithm provides better results in 21% of the cases, while for 38% of the instances, the same values as those presented in the literature are found. Whereas the percentage of instances where we obtain better results increases to 42% when we compare only with centralized approaches.

Resource accessibility indicates the level of access that the jobs belonging to a project have concerning a resource type. The reviewed literature establishes that resource accessibility can be global or local, but not both. Given this, we presented features of the problem associated with resource mixed accessibility: a resource type can present units with local accessibility, accessible only by the project to which it corresponds, and in turn, units with global accessibility, shared by all projects. Resources with mixed accessibility bring two modes of resource allocation: unitary or total. In addition, the delay and resource costs have been addressed in several studies, especially in real environments. Given this joint with the importance of considering the costs in the different problem variants (e.g., see Tian et al. (2020) and Gholizadeh-Tayyar et al. (2016)), we presented two variants of the problem named RCMPSP-MT-W and RCMPSP-MT-W+RC. Both variants focus on resources with mixed accessibility, total resource allocation, and weights associated with projects, and the second variant includes cost per use of resource units with global accessibility. The mentioned variants are described, and their corresponding mathematical models are presented. A hybrid algorithm to solve both variants is proposed. It integrates priority rules, simulated annealing, and tabu search as part of an iterated local search. The approach analyzes two perturbation intensities: simple and joint perturbation. To assess the performance of the developed approaches, 36 instances were proposed. The instances were based on the MPSPLib library modifying the availability of global and local units and adding the cost for resource usage. Moreover, we analyzed different aspects based on the results of implementing the mathematical models and the hybrid algorithm in the studied problem variants. Given the complexity of the test instances, the proposed mathematical models were only able to find the optimal solution in the instance with the lowest number of successors, the lowest average resource requirement, and the highest resource availability. On the other hand, when analyzing the results obtained from the hybrid algorithm with both perturbation intensities, we found that the application of joint perturbation yields better results with respect to simple perturbation for the proposed instances. This can be explained given that joint perturbation performs as many perturbations as projects exist, achieving higher search space exploration. In addition, when analyzing the allocation of units with global accessibility, we found that the solutions obtained by applying joint perturbation present a higher use of global units. Furthermore, we study how the accessibility of resources and the cost can influence the solutions provided. We build three cases on a test instance where a specific resource type has only global units available, local units available, or both. Through the analysis, we conclude that, although it may not be the only factor, the global accessibility of resources may imply greater allocation freedom, allowing better results.

Some indications are given concerning aspects that can be studied in future works to promote the use and application of RCMPSP and reduce the gap between theory and practice. For instance, even though many processes are automated, company staff represents most resources in project management, generating high complexity levels in schedule generation. Given this, increasing the study of different skills and performance of a worker compared to the different jobs that staff can execute can improve production processes. Furthermore, the consideration of uncertainty in production processes is an aspect slightly addressed in practice, only associated with job duration, so it is required to widen the scope of future studies in this respect. Concerning objectives, the most used ones in practice are those aligned with economic aspects and/or project completion times without emphasizing the resource use that generates nature-friendly production processes. In this sense, it may be relevant to increase the study of non-renewable and partially renewable resources while pursuing objectives such as reducing resource consumption or even reducing and leveling any resource within the time horizon. Moreover, studying and promoting the use of objectives, such as minimizing idle resources or inventory in processes in real environments, can allow efficient resource use. Also, increasing different precedence relationships where common resource use is made can considerably reduce resource use. In addition, a future line of research may be oriented to study costs associated with the different levels of accessibility, including costs related to the transfer between global and local units projects.

Regarding solution methods to satisfy current production processes, one of the main factors that can be provided is the resilience capacity to schedule changes, especially under uncertain conditions. Furthermore, despite recent work, the increased use of solution methods based on multi-agent systems, combinatorial auctions, and hybrid algorithms recommends a valuable line of research for further study. In this sense, an interesting research direction is the in-depth study of how the specific multi-project features are handled in the algorithms used in the literature to solve the different variants of the RCMPSP. Finally, the evaluation and comparison of the performance of proposed algorithms are essential for RCMPSP progress, so the creation of public libraries valuing the related features studied in the literature is essential for promoting and fostering the study of new problem variants.

Publications and Projects Arising From the Thesis

Conducted research

Journal Articles

- Gómez, M., Lalla-Ruiz, E., Fernández, A., Castro, C., and Voß S. (2022). Resource-Constrained Multi-Project Scheduling Problem: A survey. ***European Journal of Operational Research (EJOR)***.

  Clarivate Web of Science [doi:10.1016/j.ejor.2022.09.033]

- Gómez, M., Lalla-Ruiz, E., Mes, M., Fernández, A., and Castro, C. The resource-constrained multi-project scheduling problem with resources mixed accessibility. ***Expert Systems with Applications (ESWA)***.

  Clarivate Web of Science (Under Review)

Conference Proceedings

- Gómez, M., Fernández, A., and Castro, C. (2019). Integrating a SMT solver based local search in ant colony optimization for solving RCMPSP. In ***2019 IEEE Latin American Conference on Computational Intelligence (LA-CCI)***, Guayaquil, Ecuador, pages 1-6. IEEE.

  Elsevier Scopus [doi:10.1109/LA-CCI47412.2019.9036765]

- Gómez, M., Fernández, A., and Castro, C. (2019). Solving the RCMPSP using ACO and SMT-Solver based Local Search. In ***XIII Chilean Conference on Operations Research (OPTIMA)***, Santa Cruz, Chile.

Optimization field outputs

Journal Articles

- Gómez, M., Masip, Y., Fernández, A., Castro, C., Nuñez, S.M. and Pedrera, J. (2020). A mathematical model for the optimization of renewable energy systems. ***Mathematics***.

  Clarivate Web of Science [doi:10.3390/math9010039]

- Fernández, A., Gómez, M., Castro, C., and Pérez-Alonso, A. (2022). A mixed-integer linear programming model and a metaheuristic approach for the selection and allocation of land parcels problem. ***International Transactions in Operational Research (ITOR)***.

  Clarivate Web of Science [doi:10.1111/itor.13115]

- Fernández, A., Lalla-Ruiz, E., Gómez, M., and Castro, C. (2022). A review of heuristics and hybrid methods for green vehicle routing problems considering emissions. ***Journal of Advanced Transportation (JAT)***, 2022:5714991.

  Clarivate Web of Science [doi:10.1155/2022/5714991]

- Fernández, A., Lalla-Ruiz, E., Gómez, M., and Castro, C. (2022). The cumulative vehicle routing problem with time windows: models and algorithm. ***Annals of Operations Research (ANOR)***.

  Clarivate Web of Science [doi:10.1007/s10479-022-05102-7]

Conference Proceedings

- Masip, Y., Fernández, A., Gómez, M., Castro, C., & Nuñez, S. M. (2019). Optimization of a smart integrated renewable energy system for isolated rural villages using integer linear programming. In ***2019 7th International Engineering, Sciences and Technology Conference (IESTEC)***, Ciudad de Panamá, Panamá, pages 161-166. IEEE.

  Elsevier Scopus [doi:10.1109/IESTEC46403.2019.00-83]

- Fernández, A., Gómez, M., Castro, C., and Masip, Y. (2019). A mixed integer linear programming approach for the 2D strip packing problem with different size options for plots of land in smart floating farms. In ***III International Conference on Agro BigData and Decision Support Systems in Agriculture (BigDSSAgro)***, Valparaíso, Chile, pages 69-72. [isbn:978-956-356-095-4]

- Fernández, A., Gómez, M., Lalla-Ruiz, E., and Castro, C. (2020). Cumulative VRP with time windows: a trade-off analysis. In ***International Conference on Computational Logistics (ICCL)***, Enschede, The Netherlands, pages 277-291. ELSEVIER Scopus [doi:10.1007/978-3-030-59747-4_18]

Research Projects

- Optimization Models and Algorithms for the Resource-Constrained Multi-project Scheduling Problem. 2020-2021. ***Scientific Initiation Incentive Program (PIIC)*** DGIP-UTFSM 015/2020. Universidad Técnica Federico Santa María. Chile.

- Optimization and modeling based on matheuristic methods for solving vehicle green routing problems. 2020-2022. ***Regular Research Line Project DGIIE-UTFSM (PI_LIR_2020_67)***. Universidad Técnica Federico Santa María. Chile.

Appendices

Appendix A

The mathematical formulation for the RCMPSP-MU minimizing the average project delay (APD) can be defined based on the mathematical model described in Section 2.1.1.

The parameter a_k is removed, and the following parameters are added:

ga^k Availability of units with global accessibility of resource type k at every period,

la_i^k Availability of units with local accessibility of resources type k belonging to the project i,

The following decision variable must also be added:

G_i^{kt} Amount of type k units with global accessibility assigned to project i at period t,

$$D_i = \begin{cases} 1, & \text{if project } i \text{ is delayed,} \\ 0, & \text{otherwise.} \end{cases} \tag{6.1}$$

For the RCMPSP-MU-W the objective function (6.2) minimizes the APD through the sum of the delay of each project. For each project, the difference between the completion time and the desired due date (DD_i) is calculated. If the project is delayed ($D_i = 1$), the difference is added; otherwise, the difference is multiplied by zero.

$$\text{Minimize} \quad \sum_{i=1}^{|I|} \Big(\sum_{t=1}^{|T|} t \cdot X_{i|N_i|}^{t} - DD_i \Big) \cdot D_i \tag{6.2}$$

The following constraints must replace constraints (2.4) that ensure the use of global resources:

$$\sum_{i=1}^{|I|} G_i^{kt} \leq ga^k \ , \ \ \forall\, k \in K, t \in T \tag{6.3}$$

$$\sum_{j=1}^{|N_i|} \sum_{q=t}^{t+D_{ij}-1} c_{ij}^k \times X_{ij}^q \leq la_i^k + G_i^{kt} \ , \ \ \forall \, i \in I, k \in K, t \in T \tag{6.4}$$

Constraints (6.3) guarantee that the global units allocation does not exceed the current availability of each resource type with global accessibility at each time period. Constraints (6.4) ensure that the local units added to the allocation of global units are enough to supply the demand for each resource for each project at each period.

To manage project delays, it is required to add the following constraints:

$$\sum_{t=1}^{|T|} t \times X_{i|N_i|}^t - DD_i \leq D_i \times M \ , \ \ \forall \, i \in I \tag{6.5}$$

When adding the aforementioned decision variables, it is required to add the following constraints to the model:

$$G_i^{kt} \geq 0 \ , \ \ \forall \, i \in I, j \in N_i, t \in T \tag{6.6}$$

$$D_i \in \{0, 1\} \ , \ \ \forall \, i \in I \tag{6.7}$$

Appendix B

Table B.1: Comparison of components performance of our approach. RCMPSP-MT-W.

Instance	Simple perturbation				Joint perturbation				Improvement		
	Time_s (seg)	M_s	APD_s	WPD_s	Time_j (seg)	M_j	APD_j	WPD_j	M_i	APD_i	WPD_i
mpma_j30_i2_1	4.77	79.00	12.00	84.00	5.01	79.00	12.00	84.00	0.00	0.00	0.00
mpma_j30_i2_2	3.10	65.00	2.50	70.00	2.90	65.00	2.50	70.00	0.00	0.00	0.00
mpma_j30_i2_3	5.50	74.00	12.00	311.00	5.95	74.00	12.00	311.00	0.00	0.00	0.00
mpma_j30_i5_1	9.90	80.00	8.20	90.20	11.58	80.00	8.20	90.20	0.00	0.00	0.00
mpma_j30_i5_2	44.89	112.00	31.00	420.00	49.87	112.00	31.00	420.00	0.00	0.00	0.00
mpma_j30_i5_3	14.28	83.00	12.60	232.40	16.99	83.00	12.60	232.40	0.00	0.00	0.00
mpma_j30_i10_1	46.67	87.00	30.00	245.00	58.34	87.00	30.00	245.00	0.00	0.00	0.00
mpma_j30_i10_2	39.37	132.00	46.00	600.00	48.04	132.00	46.00	600.00	0.00	0.00	0.00
mpma_j30_i10_3	54.64	103.00	20.30	253.30	62.84	106.00	19.60	252.30	2.91	**-3.45**	**-0.39**
mpma_j30_i20_1	90.55	93.00	22.00	289.00	113.19	93.00	22.00	289.00	0.00	0.00	0.00
mpma_j30_i20_2	107.75	174.00	75.00	1062.00	135.76	174.00	75.00	1062.00	0.00	0.00	0.00

(Continued on next page)

Table B.1 – *(Continued from previous page)*

Instance	Simple perturbation				Joint perturbation				Improvement		
	$Time_s$ (seg)	M_s	APD_s	WPD_s	$Time_j$ (seg)	M_j	APD_j	WPD_j	M_i	APD_i	WPD_i
mpma_j30_i20_3	168.92	180.00	49.00	970.00	201.22	183.00	48.00	935.00	1.67	**-2.04**	**-3.61**
mpma_j90_i2_1	7.42	89.00	0.00	0.00	8.54	89.00	0.00	0.00	0.00	0.00	0.00
mpma_j90_i2_2	69.93	187.00	65.00	2132.00	73.81	176.00	56.00	1850.00	**-5.88**	**-13.85**	**-13.23**
mpma_j90_i2_3	23.31	127.00	7.00	224.00	26.02	123.00	5.00	160.00	**-3.15**	**-28.57**	**-28.57**
mpma_j90_i5_1	99.98	146.00	56.00	3248.00	106.98	146.00	56.00	3248.00	0.00	0.00	0.00
mpma_j90_i5_2	84.86	156.00	19.40	1156.60	98.83	150.00	15.80	983.80	**-3.85**	**-18.56**	**-14.94**
mpma_j90_i5_3	116.68	175.00	34.00	1956.00	142.35	175.00	34.00	1956.00	0.00	0.00	0.00
mpma_j90_i10_1	289.27	154.00	48.00	1400.00	302.80	149.00	42.00	1226.00	**-3.25**	**-12.50**	**-12.43**
mpma_j90_i10_2	920.88	376.00	177.00	6269.00	985.31	387.00	176.00	6159.10	2.93	**-0.56**	**-1.75**
mpma_j90_i10_3	813.91	431.00	201.00	6548.00	904.55	414.00	198.00	6493.30	**-3.94**	**-1.49**	**-0.84**
mpma_j90_i20_1	487.07	123.00	12.00	657.00	524.23	114.00	4.00	210.00	**-7.32**	**-66.67**	**-68.04**
mpma_j90_i20_2	471.11	255.00	27.00	815.00	508.31	253.00	25.00	773.00	**-0.78**	**-7.41**	**-5.15**
mpma_j90_i20_3	1162.15	269.00	99.00	4943.00	1222.96	263.00	101.00	1625.00	**-2.23**	2.02	**-67.13**

(Continued on next page)

Table B.1 – *(Continued from previous page)*

Instance	Simple perturbation				Joint perturbation				Improvement		
	$Time_s$ (seg)	M_s	APD_s	WPD_s	$Time_j$ (seg)	M_j	APD_j	WPD_j	M_i	APD_i	WPD_i
mpma_j120_i2_1	60.50	138.00	16.00	1369.00	65.58	138.00	16.00	1369.00	0.00	0.00	0.00
mpma_j120_i2_2	136.17	244.00	119.00	11318.00	142.32	244.00	118.00	11202.00	0.00	**-0.84**	**-1.02**
mpma_j120_i2_3	55.46	142.00	28.00	874.00	61.00	142.00	28.00	874.00	0.00	0.00	0.00
mpma_j120_i5_1	92.53	91.00	10.00	350.00	101.74	85.00	4.00	140.00	**-6.59**	**-60.00**	**-60.00**
mpma_j120_i5_2	172.60	250.00	109.00	4965.00	208.84	250.00	109.00	4965.00	0.00	0.00	0.00
mpma_j120_i5_3	194.62	276.00	131.00	10116.00	217.43	276.00	131.00	10116.00	0.00	0.00	0.00
mpma_j120_i10_1	285.54	106.00	26.00	962.00	312.50	101.00	23.00	884.00	**-4.72**	**-11.54**	**-8.11**
mpma_j120_i10_2	181.68	193.00	27.00	1247.00	200.73	194.00	27.00	1250.00	0.52	0.00	0.24
mpma_j120_i10_3	366.04	226.00	85.00	2640.00	405.25	226.00	85.00	2640.00	0.00	0.00	0.00
mpma_j120_i20_1	529.70	91.00	10.00	350.00	608.01	91.00	5.00	178.00	0.00	**-50.00**	**-49.14**
mpma_j120_i20_2	517.89	273.00	98.00	5062.00	673.26	273.00	98.00	5062.00	0.00	0.00	0.00
mpma_j120_i20_3	412.28	231.00	61.00	3370.00	476.84	231.00	61.00	3370.00	0.00	0.00	0.00

Table B.2: Comparison of components performance of our approach. RCMPSP-MT-W+RC.

Instance	Simple perturbation						Joint perturbation						Improvement		
	$Time_s$ (seg)	M_s	APD_s	WPD	RPC	TC	$Time_j$ (seg)	M_j	APD	WPD	RPC	TC	M_i	APD_i	TC
mpma_j30_i2_1	6.03	95.00	25.00	178.00	871.00	1049.00	7.31	108.00	36.50	255.50	715.00	970.50	13.68	46.00	**-7.48**
mpma_j30_i2_2	3.56	65.00	2.50	70.00	120.00	190.00	4.91	65.00	2.50	70.00	120.00	190.00	0.00	0.00	0.00
mpma_j30_i2_3	6.70	75.00	12.00	314.00	656.00	970.00	8.99	75.00	12.00	314.00	656.00	970.00	0.00	0.00	0.00
mpma_j30_i5_1	10.25	81.00	8.60	94.60	530.00	624.60	12.35	87.00	14.80	162.80	277.00	439.80	7.41	72.09	**-29.59**
mpma_j30_i5_2	50.29	122.00	41.20	568.00	1702.00	2270.00	51.34	117.00	36.20	494.00	1756.00	2250.00	**-4.10**	**-12.14**	**-0.88**
mpma_j30_i5_3	25.37	83.00	12.60	232.40	742.00	974.40	30.05	84.00	13.80	252.80	700.00	952.80	1.20	9.52	**-2.22**
mpma_j30_i10_1	45.62	93.00	37.10	296.80	651.00	947.80	51.06	94.00	36.70	293.60	666.00	959.60	1.08	**-1.08**	1.24
mpma_j30_i10_2	43.86	149.00	54.30	703.30	1309.00	2012.30	51.23	129.00	48.50	626.60	1322.00	1948.60	**-13.42**	**-10.68**	**-3.17**
mpma_j30_i10_3	68.52	103.00	21.00	253.50	316.00	569.50	69.25	105.00	20.30	256.30	300.00	556.30	1.94	**-3.33**	**-2.32**
mpma_j30_i20_1	103.52	105.00	29.90	388.70	646.00	1034.70	99.73	101.00	27.85	362.00	671.00	1033.00	**-3.81**	**-6.86**	**-0.16**
mpma_j30_i20_2	108.02	174.00	75.80	1062.65	288.00	1350.65	102.62	174.00	75.80	1062.65	288.00	1350.65	0.00	0.00	0.00

(Continued on next page)

Table B.2 – *(Continued from previous page)*

Instance	Simple perturbation						Joint perturbation						Improvement		
	$Time_s$ (seg)	M_s	APD_s	WPD	RPC	TC	$Time_j$ (seg)	M_j	APD	WPD	RPC	TC	M_i	APD_i	TC
mpma_j30_i20_3	185.60	184.00	49.45	977.45	418.00	1395.45	190.02	183.00	48.60	935.80	416.00	1351.80	**-0.54**	**-1.72**	**-3.13**
mpma_j90_i2_1	10.26	89.00	0.00	0.00	0.00	0.00	12.99	89.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
mpma_j90_i2_2	65.29	191.00	67.50	2218.00	699.00	2917.00	70.09	191.00	67.50	2218.00	699.00	2917.00	0.00	0.00	0.00
mpma_j90_i2_3	35.94	128.00	7.50	240.00	846.00	1086.00	48.62	127.00	7.00	224.00	898.00	1122.00	**-0.78**	**-6.67**	3.31
mpma_j90_i5_1	105.42	159.00	69.00	4002.00	6344.00	10346.00	115.32	156.00	66.00	3828.00	5951.00	9779.00	**-1.89**	**-4.35**	**-5.48**
mpma_j90_i5_2	101.69	155.00	21.20	1246.00	2488.00	3734.00	135.01	156.00	22.00	1287.40	2497.00	3784.40	0.65	3.77	1.35
mpma_j90_i5_3	126.04	177.00	36.60	2066.20	4284.00	6350.20	151.30	176.00	35.60	2005.40	4310.00	6315.40	**-0.56**	**-2.73**	**-0.55**
mpma_j90_i10_1	301.21	156.00	50.30	1458.70	1552.00	3010.70	309.26	149.00	42.30	1226.70	1622.00	2848.70	**-4.49**	**-15.90**	**-5.38**
mpma_j90_i10_2	1001.01	377.00	177.80	6278.90	5010.00	11288.90	1015.92	387.00	176.00	6159.10	4808.00	10967.10	2.65	**-1.01**	**-2.85**
mpma_j90_i10_3	989.02	431.00	202.10	6565.10	2099.00	8664.10	1003.89	414.00	198.80	6493.30	1987.00	8480.30	-3.94	**-1.63**	**-2.12**
mpma_j90_i20_1	751.73	124.00	14.50	754.00	3787.00	4541.00	834.20	124.00	14.85	772.20	3762.00	4534.20	0.00	2.41	**-0.15**
mpma_j90_i20_2	698.98	269.00	30.00	928.00	2570.00	3498.00	679.08	253.00	25.75	773.10	2607.00	3380.10	**-5.95**	**-14.17**	**-3.37**

(Continued on next page)

Table B.2 – *(Continued from previous page)*

Instance	Simple perturbation						Joint perturbation						Improvement		
	$Time_s$ (seg)	M_s	APD_s	WPD	RPC	TC	$Time_j$ (seg)	M_j	APD	WPD	RPC	TC	M_i	APD_i	TC
mpma_j90_i20_3	1251.38	269.00	99.10	4951.65	1337.00	6288.65	1245.38	263.00	101.95	4902.00	1266.00	6168.00	**-2.23**	2.88	**-1.92**
mpma_j120_i2_1	73.94	138.00	16.50	1369.50	349.00	1718.50	85.77	137.00	14.00	1162.00	368.00	1530.00	**-0.72**	**-15.15**	**-10.97**
mpma_j120_i2_2	169.05	244.00	119.00	11318.00	6289.00	17607.00	178.29	244.00	119.00	11318.00	6289.00	17607.00	0.00	0.00	0.00
mpma_j120_i2_3	58.23	144.00	30.50	943.00	3033.00	3976.00	65.22	143.00	29.50	908.50	3168.00	4076.50	**-0.69**	**-3.28**	2.53
mpma_j120_i5_1	103.52	93.00	11.60	406.00	1202.00	1608.00	105.09	96.00	15.00	525.00	794.00	1319.00	3.23	29.31	**-17.97**
mpma_j120_i5_2	186.91	253.00	112.60	5094.20	6500.00	11594.20	200.06	234.00	112.00	5318.80	6285.00	11603.80	**-7.51**	**-0.53**	0.08
mpma_j120_i5_3	206.85	276.00	131.40	10069.80	1994.00	12063.80	208.00	276.00	131.80	10116.60	1978.00	12094.60	0.00	0.30	0.26
mpma_j120_i10_1	368.85	109.00	29.00	1073.00	1697.00	2770.00	401.23	104.00	26.9	995.30	1646.00	2641.30	**-4.59**	**-7.24**	**-4.65**
mpma_j120_i10_2	192.52	194.00	27.10	1250.00	2388.00	3638.00	243.90	194.00	27.10	1250.00	2388.00	3638.00	0.00	0.00	0.00
mpma_j120_i10_3	389.02	226.00	85.70	2640.30	522.00	3162.30	426.01	226.00	85.70	2640.30	521.00	3161.30	0.00	0.00	**-0.03**
mpma_j120_i20_1	726.95	92.00	11.00	385.00	1015.00	1400.00	788.22	96.00	10.10	353.50	971.00	1324.50	4.35	**-8.18**	**-5.39**
mpma_j120_i20_2	851.30	273.00	98.50	5062.10	1735.00	6797.10	901.99	266.00	99.44	5130.60	1652.00	6782.60	**-2.56**	0.95	**-0.21**

(Continued on next page)

Table B.2 – *(Continued from previous page)*

Instance	Simple perturbation						Joint perturbation						Improvement		
	Time_s (seg)	M_s	APD_s	WPD	RPC	TC	Time_j (seg)	M_j	APD	WPD	RPC	TC	M_i	APD_i	TC
mpma_j120_i20_3	791.95	231.00	61.40	3370.05	941.00	4311.05	861.05	231.00	61.40	3370.05	941.00	4311.05	0.00	0.00	0.00

Table B.3: Differences in M and APD caused by resource costs.

Instance	M_s	APD_s	M_j	APD_j
mpma_j30_i2_1	16.00	13.00	29.00	24.50
mpma_j30_i2_2	0.00	0.00	0.00	0.00
mpma_j30_i2_3	1.00	0.00	1.00	0.00
mpma_j30_i5_1	1.00	0.40	7.00	6.60
mpma_j30_i5_2	10.00	10.00	5.00	5.00
mpma_j30_i5_3	0.00	0.00	1.00	1.20
mpma_j30_i10_1	6.00	6.40	7.00	6.70
mpma_j30_i10_2	17.00	8.00	-3.00	2.50
mpma_j30_i10_3	0.00	0.70	-1.00	0.70
mpma_j30_i20_1	12.00	7.90	8.00	5.85
mpma_j30_i20_2	0.00	0.00	0.00	0.80
mpma_j30_i20_3	4.00	0.45	0.00	0.60
mpma_j90_i2_1	0.00	0.00	0.00	0.00
mpma_j90_i2_2	4.00	2.50	15.00	11.50
mpma_j90_i2_3	1.00	0.50	4.00	2.00
mpma_j90_i5_1	13.00	13.00	10.00	10.00
mpma_j90_i5_2	-1.00	1.80	6.00	6.20
mpma_j90_i5_3	2.00	1.80	1.00	1.60
mpma_j90_i10_1	2.00	2.00	0.00	0.30
mpma_j90_i10_2	1.00	0.80	0.00	0.00
mpma_j90_i10_3	0.00	1.10	0.00	0.80
mpma_j90_i20_1	1.00	2.50	10.00	10.85

(Continued on next page)

Table B.3 – *(Continued from previous page)*

Instance	M_s	APD_s	M_j	APD_j
mpma_j90_i20_2	14.00	3.00	0.00	0.75
mpma_j90_i20_3	0.00	0.10	0.00	0.95
mpma_j120_i2_1	0.00	0.00	-1.00	-2.00
mpma_j120_i2_2	0.00	0.00	0.00	1.00
mpma_j120_i2_3	2.00	2.00	1.00	1.50
mpma_j120_i5_1	2.00	1.60	11.00	11.00
mpma_j120_i5_2	3.00	2.80	-16.00	3.00
mpma_j120_i5_3	0.00	-0.40	0.00	0.80
mpma_j120_i10_1	3.00	3.00	3.00	3.90
mpma_j120_i10_2	1.00	0.10	0.00	0.10
mpma_j120_i10_3	0.00	0.00	0.00	0.00
mpma_j120_i20_1	1.00	1.00	5.00	5.10
mpma_j120_i20_2	0.00	0.00	-7.00	1.44
mpma_j120_i20_3	0.00	0.00	0.00	0.40

Bibliography

Adam, S. P., Alexandropoulos, S.-A. N., Pardalos, P. M., and Vrahatis, M. N. (2019). *No Free Lunch Theorem: A Review*, pages 57–82.

Adhau, S., Mittal, M., and Mittal, A. (2013). A multi-agent system for decentralized multi-project scheduling with resource transfers. *International Journal of Production Economics*, 146(2):646–661.

Adhau, S., Mittal, M. L., and Mittal, A. (2012). A multi-agent system for distributed multi-project scheduling: An auction-based negotiation approach. *Engineering Applications of Artificial Intelligence*, 25(8):1738–1751.

Ahmeti, A. and Musliu, N. (2021). Hybridizing constraint programming and meta-heuristics for multi-mode resource-constrained multiple projects scheduling problem. In *Proceedings of the 13th International Conference on the Practice and Theory of Automated Timetabling-PATAT*, volume 1, pages 188–206.

Amirian, H. and Sahraeian, R. (2017). Solving a grey project selection scheduling using a simulated shuffled frog leaping algorithm. *Computers & Industrial Engineering*, 107:141–149.

Araujo, J. A., Santos, H. G., Gendron, B., Jena, S. D., Brito, S. S., and Souza, D. S. (2020). Strong bounds for resource constrained project scheduling: Preprocessing and cutting planes. *Computers & Operations Research*, 113:104782.

Asta, S., Karapetyan, D., Kheiri, A., Özcan, E., and Parkes, A. J. (2016). Combining Monte-Carlo and hyper-heuristic methods for the multi-mode resource-constrained multi-project scheduling problem. *Information Sciences*, 373:476–498.

BCA (2017). Conquas: The BCA construction quality assessment system. Technical report., Building and Construction Authority, Singapore.

Beşikci, U., Bilge, Ü., and Ulusoy, G. (2015). Multi-mode resource constrained multi-project scheduling and resource portfolio problem. *European Journal of Operational Research*, 240(1):22–31.

Blum, C., Roli, A., and Sampels, M. (2008). *Hybrid metaheuristics: an emerging approach to optimization*, volume 114. Springer.

Bock, D. and Patterson, J. (1990). A comparison of due date setting, resource assignment, and job preemption heuristics for the multiproject scheduling problem. *Decision Sciences*, 21(2):387–402.

Bofill, M., Coll, J., Suy, J., and Villaret, M. (2020). SMT encodings for resource-constrained project scheduling problems. *Computers & Industrial Engineering*, 149:106777.

Browning, T. R. and Yassine, A. A. (2010a). A random generator of resource-constrained multi-project network problems. *Journal of Scheduling*, 13(2):143–161.

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

Brucker, P., Drexl, A., Möhring, R., Neumann, K., and Pesch, E. (1999). Resource-constrained project scheduling: Notation, classification, models, and methods. *European Journal of Operational Research*, 112(1):3–41.

Cai, Z. and Li, X. (2012). A hybrid genetic algorithm for resource-constrained multi-project scheduling problem with resource transfer time. In *2012 IEEE International Conference on Automation Science and Engineering (CASE)*, pages 569–574. IEEE.

Can, A. and Ulusoy, G. (2014). Multi-project scheduling with two-stage decomposition. *Annals of Operations Research*, 217(1):95–116.

Chakrabortty, R. K., Sarker, R. A., and Essam, D. L. (2017). Resource constrained multi-project scheduling: A priority rule based evolutionary local search approach. In *Intelligent and Evolutionary Systems*, pages 75–86. Springer.

Chen, H., Ding, G., Zhang, J., and Qin, S. (2019). Research on priority rules for the stochastic resource constrained multi-project scheduling problem with new project arrival. *Computers*

& Industrial Engineering, 137:106060.

Chen, J. and Askin, R. G. (2009). Project selection, scheduling and resource allocation with time dependent returns. *European Journal of Operational Research*, 193(1):23–34.

Chen, J.-j., Zhu, J.-l., and Zhang, D.-n. (2014). Multi-project scheduling problem with human resources based on dynamic programming and staff time coefficient. In *2014 International Conference on Management Science & Engineering 21th Annual Conference Proceedings*, pages 1012–1018. IEEE.

Chen, P.-H. and Shahandashti, S. M. (2009). Hybrid of genetic algorithm and simulated annealing for multiple project scheduling with multiple resource constraints. *Automation in Construction*, 18(4):434–443.

Chen, R.-M. (2011). Particle swarm optimization with justification and designed mechanisms for resource-constrained project scheduling problem. *Expert Systems with Applications*, 38(6):7102–7111.

Chen, T. and Ju, C. (2015). Comparative analysis of swarm intelligence and heuristic priority rules for solving multi-project scheduling problem. *International Journal of Computing Science and Mathematics*, 6(6):581–599.

Chen, V. Y. (1994). A 0–1 goal programming model for scheduling multiple maintenance projects at a copper mine. *European Journal of Operational Research*, 76(1):176–191.

Cheng, C.-B., Lo, C.-Y., and Chu, C.-P. (2019). Solving multi-mode resource-constrained multi-project scheduling problem with combinatorial auction mechanisms. *International Journal of Information and Management Sciences*, 30(2):143–167.

Chiu, H. N. and Tsai, D. M. (2002). An efficient search procedure for the resource-constrained multi-project scheduling problem with discounted cash flows. *Construction Management & Economics*, 20(1):55–66.

Coelho, J. and Vanhoucke, M. (2011). Multi-mode resource-constrained project scheduling using RCPSP and SAT solvers. *European Journal of Operational Research*, 213(1):73–82.

Confessore, G., Giordani, S., and Rismondo, S. (2007). A market-based multi-agent system model for decentralized multi-project scheduling. *Annals of Operations Research*, 150(1):115–135.

Cui, L., Liu, X., Lu, S., and Jia, Z. (2021). A variable neighborhood search approach for the resource-constrained multi-project collaborative scheduling problem. *Applied Soft Computing*, 107:107480.

Davari Ardakani, H. and Dehghani, A. (2022). Multi-objective optimization of multi-mode resource-constrained project selection and scheduling problem considering resource leveling and time-varying resource usage. *International Journal of Supply and Operations Management*, 9(1):34–55.

De Moura, L. and Bjørner, N. (2008). Z3: An efficient SMT solver. In *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, pages 337–340. Springer.

De Moura, L. and Bjørner, N. (2011). Satisfiability modulo theories: introduction and applications. *Communications of the ACM*, 54(9):69–77.

Deckro, R. F., Winkofsky, E., Hebert, J. E., and Gagnon, R. (1991). A decomposition approach to multi-project scheduling. *European Journal of Operational Research*, 51(1):110–118.

Dike, S. H. (1964). Project scheduling with resource constraints. *IEEE Transactions on Engineering Management*, EM-11(4):155–157.

Dorigo, M., Maniezzo, V., and Colorni, A. (1996). Ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 26(1):29–41.

Dumond, J. (1992). In a multi-resource environment, how much is enough? *International Journal of Production Research*, 30(2):395–410.

Dumond, J. and Mabert, V. A. (1988). Evaluating project scheduling and due date assignment procedures: An experimental analysis. *Management Science*, 34(1):101–118.

Fu, F. and Zhou, H. (2021). A combined multi-agent system for distributed multi-project scheduling problems. *Applied Soft Computing*, 107:107402.

Gálvez, N. (2018). *A Framework for Autonomous Generation of Strategies in Satisfiability Modulo Theories*. Ph.D. thesis, Universidad Técnica Federico Santa María, Chile.

Gantt, H. L. (1913). *Work, wages, and profits*. Engineering Magazine Company.

Geiger, M. J. (2017). A multi-threaded local search algorithm and computer implementation for the multi-mode, resource-constrained multi-project scheduling problem. *European Journal of Operational Research*, 256(3):729–741.

Gholizadeh-Tayyar, S., Dupont, L., Lamothe, J., and Falcon, M. (2016). Modeling a generalized resource constrained multi project scheduling problem integrated with a forward-backward supply chain planning. *IFAC-PapersOnLine*, 49(12):1283–1288. 8th IFAC Conference on Manufacturing Modelling, Management and Control MIM 2016.

Glover, F. and Laguna, M. (1998). *Tabu Search*, pages 2093–2229. Springer.

Gómez, M., Fernández, A., and Castro, C. (2019). Integrating a SMT solver based local search in ant colony optimization for solving RCMPSP. In *2019 IEEE Latin American Conference on Computational Intelligence (LA-CCI)*, pages 1–6. IEEE.

Gonçalves, J. F., de Magalhães Mendes, J. J., and Resende, M. G. (2015). The basic multi-project scheduling problem. In *Handbook on Project Management and Scheduling Vol. 2*, pages 667–683. Springer.

Gonçalves, J. F., Mendes, J. J., and Resende, M. G. (2008). A genetic algorithm for the resource constrained multi-project scheduling problem. *European Journal of Operational Research*, 189(3):1171–1190.

Gutjahr, W. J., Katzensteiner, S., Reiter, P., Stummer, C., and Denk, M. (2008). Competence-driven project portfolio selection, scheduling and staff assignment. *Central European Journal of Operations Research*, 16(3):281–306.

Hao, Q., Shen, W., Xue, Y., and Wang, S. (2010). Task network-based project dynamic scheduling and schedule coordination. *Advanced Engineering Informatics*, 24(4):417–427.

Hartmann, S. and Briskorn, D. (2010). A survey of variants and extensions of the resource-constrained project scheduling problem. *European Journal of Operational Research*, 207(1):1–14.

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

Hauder, V. A., Beham, A., Raggl, S., Parragh, S. N., and Affenzeller, M. (2020). Resource-constrained multi-project scheduling with activity and time flexibility. *Computers & Industrial Engineering*, 150:106857.

He, Y., He, Z., and Wang, N. (2021). Tabu search and simulated annealing for resource-constrained multi-project scheduling to minimize maximal cash flow gap. *Journal of Industrial & Management Optimization*, 17(5):2451–2474.

Homberger, J. (2007). A multi-agent system for the decentralized resource-constrained multi-project scheduling problem. *International Transactions in Operational Research*, 14(6):565–589.

Homberger, J. (2012). A (μ, λ)-coordination mechanism for agent-based multi-project scheduling. *OR Spectrum*, 34(1):107–132.

Hu, W., Hu, Z., Huang, W., Chen, Y., and Song, W. (2010). Scheduling and optimization of multi-project resources based on cellular automata. In *2010 International Conference on Logistics Engineering and Intelligent Transportation Systems*, pages 1–7. IEEE.

Hu, W., Wang, H., Peng, C., Wang, H., Liang, H., and Du, B. (2015). An outer-inner fuzzy cellular automata algorithm for dynamic uncertainty multi-project scheduling problem. *Soft Computing*, 19(8):2111–2132.

Hutter, F., Hoos, H. H., Leyton-Brown, K., and Stützle, T. (2009). ParamILS: an automatic algorithm configuration framework. *Journal of Artificial Intelligence Research*, 36:267–306.

Issa, S. and Tu, Y. (2020). A survey in the resource-constrained project and multi-project scheduling problems. *Journal of Project Management*, 5(2):117–138.

Issa, S. B., Patterson, R. A., and Tu, Y. (2021). Solving resource-constrained multi-project environment under different activity assumptions. *International Journal of Production Economics*, 232:107936.

Jain, A. and Meeran, S. (1999). Deterministic job-shop scheduling: Past, present and future. *European Journal of Operational Research*, 113(2):390–434.

Jedrzejowicz, P. and Ratajczak-Ropel, E. (2019). A-team solving distributed resource-constrained multi-project scheduling problem. *Vietnam Journal of Computer Science*, 6(4):423–437.

Joo, B. and Chua, P. (2017). Multimode resource-constrained multi-project scheduling with ad hoc activity splitting. In *2017 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, pages 2261–2265. IEEE.

Ju, C. and Chen, T. (2012). Simplifying multiproject scheduling problem based on design structure matrix and its solution by an improved aiNet algorithm. *Discrete Dynamics in Nature and Society*, 2012:713740.

Kannimuthu, M., Raphael, B., Ekambaram, P., and Kuppuswamy, A. (2020). Comparing optimization modeling approaches for the multi-mode resource-constrained multi-project scheduling problem. *Engineering, Construction and Architectural Management*, 27(4):893–916.

Kao, H.-P., Hsieh, B., and Yeh, Y. (2006). A Petri-net based approach for scheduling and rescheduling resource-constrained multiple projects. *Journal of the Chinese Institute of Industrial Engineers*, 23(6):468–477.

Karaa, F. A. and Nasr, A. Y. (1986). Resource management in construction. *Journal of Construction Engineering and Management*, 112(3):346–357.

Kelley, J. E. and Walker, M. R. (1959). Critical-path planning and scheduling. In *Papers presented at the December 1-3, 1959, Eastern Joint IRE-AIEE-ACM Computer Conference*, pages 160–173. ACM.

Kim, S. O. and Schniederjans, M. J. (1989). Heuristic framework for the resource constrained multi-project scheduling problem. *Computers & Operations Research*, 16(6):541–556.

Kirkpatrick, S., Gelatt Jr, C. D., and Vecchi, M. P. (1983). Optimization by simulated annealing. *science*, 220(4598):671–680.

Kolisch, R. (2000). Integrated scheduling, assembly area-and part-assignment for large-scale, make-to-order assemblies. *International Journal of Production Economics*, 64(1-3):127–141.

Kolisch, R. and Heimerl, C. (2012). An efficient metaheuristic for integrated scheduling and staffing IT projects based on a generalized minimum cost flow network. *Naval Research Logistics (NRL)*, 59(2):111–127.

Kolisch, R. and Meyer, K. (2006). Selection and scheduling of pharmaceutical research projects. In *Perspectives in Modern Project Scheduling*, pages 321–344. Springer.

Kolisch, R. and Sprecher, A. (1997). PSPLIB - A project scheduling problem library: OR Software - ORSEP operations research software exchange program. *European Journal of Operational Research*, 96(1):205–216.

Kong, F. and Dou, D. (2021). Resource-constrained project scheduling problem under multiple time constraints. *Journal of Construction Engineering and Management*, 147(2):04020170.

Krüger, D. and Scholl, A. (2009). A heuristic solution framework for the resource constrained (multi-)project scheduling problem with sequence-dependent transfer times. *European Journal of Operational Research*, 197(2):492–508.

Krüger, D. and Scholl, A. (2010). Managing and modelling general resource transfers in (multi-) project scheduling. *OR Spectrum*, 32(2):369–394.

Kumanan, S., Jose, G. J., and Raja, K. (2006). Multi-project scheduling using an heuristic and a genetic algorithm. *The International Journal of Advanced Manufacturing Technology*, 31(3):360–366.

Kurtulus, I. (1985). Multiproject scheduling: Analysis of scheduling strategies under unequal delay penalties. *Journal of Operations Management*, 5(3):291–307.

Kurtulus, I. and Davis, E. (1982). Multi-project scheduling: Categorization of heuristic rules performance. *Management Science*, 28(2):161–172.

Laurent, A., Deroussi, L., Grangeon, N., and Norre, S. (2017). A new extension of the RCPSP in a multi-site context: Mathematical model and metaheuristics. *Computers & Industrial Engineering*, 112:634–644.

Lawrence, S. R. and Morton, T. E. (1993). Resource-constrained multi-project scheduling with tardy costs: Comparing myopic, bottleneck, and resource pricing heuristics. *European Journal of Operational Research*, 64(2):168–187.

Lee, C. Y. and Lei, L. (2001). Multiple-project scheduling with controllable project duration and hard resource constraint: Some solvable cases. *Annals of Operations Research*, 102(1):287–307.

Lenstra, J. K. and Rinnooy Kan, A. (1978). Complexity of scheduling under precedence constraints. *Operations Research*, 26(1):22–35.

Li, F. and Xu, Z. (2018). A multi-agent system for distributed multi-project scheduling with two-stage decomposition. *PloS one*, 13(10):1–24.

Li, F., Xu, Z., and Li, H. (2021). A multi-agent based cooperative approach to decentralized multi-project scheduling and resource allocation. *Computers & Industrial Engineering*, 151:106961.

Li, J. and Liu, W. (2005). A hybrid genetic algorithm for the resource constrained multi-project scheduling problem. In *2005 IEEE Conference on Emerging Technologies and Factory Automation*, volume 2, pages 414–420. IEEE.

Linyi, D. and Yan, L. (2007). A particle swarm optimization for resource-constrained multi-project scheduling problem. In *2007 International Conference on Computational Intelligence and Security (CIS 2007)*, pages 1010–1014. IEEE.

Lipowski, A. and Lipowska, D. (2012). Roulette-wheel selection via stochastic acceptance. *Physica A: Statistical Mechanics and its Applications*, 391(6):2193–2196.

Liu, D. and Xu, Z. (2020). A multi-PR heuristic for distributed multi-project scheduling with uncertain duration. *IEEE Access*, 8:227780–227792.

Liu, J. and Lu, M. (2019). Robust dual-level optimization framework for resource-constrained multiproject scheduling for a prefabrication facility in construction. *Journal of Computing in Civil Engineering*, 33(2):04018067.

Liu, M., Shan, M., and Wu, J. (2014). Multiple R&D projects scheduling optimization with improved particle swarm algorithm. *The Scientific World Journal*, 2014:652135.

Lourenço, H. R., Martin, O. C., and Stützle, T. (2003). *Iterated Local Search*, pages 320–353. Springer US.

Lova, A., Maroto, C., and Tormos, P. (2000). A multicriteria heuristic method to improve resource allocation in multiproject scheduling. *European Journal of Operational Research*, 127(2):408–424.

Lova, A. and Tormos, P. (2001). Analysis of scheduling schemes and heuristic rules performance in resource-constrained multiproject scheduling. *Annals of Operations Research*, 102(1):263–286.

Malcolm, D. G., Roseboom, J. H., Clark, C. E., and Fazar, W. (1959). Application of a technique for research and development program evaluation. *Operations Research*, 7(5):646–669.

Man, Z., Wei, T., Xiang, L., and Lishan, K. (2008). Research on multi-project scheduling problem based on hybrid genetic algorithm. In *2008 International Conference on computer science and software engineering*, volume 1, pages 390–394. IEEE.

Mao, X., Roos, N., and Salden, A. (2009). Stable multi-project scheduling of airport ground handling services by heterogeneous agents. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems*, volume 1, pages 537–544. IFAAMAS.

Mohanty, R. and Siddiq, M. (1989). Multiple projects-multiple resources-constrained scheduling: Some studies. *International Journal of Production Research*, 27(2):261–280.

Nabipoor, A. E., Aghaie, A., and Najafi, A. (2020). Robust optimization for the resource-constrained multi-project scheduling problem with uncertain activity durations. *Scientia Iranica*, 27(1):361–376.

Namazian, A., Haji Yakhchali, S., and Rabbani, M. (2019). Integrated bi-objective project selection and scheduling using bayesian networks: A risk-based approach. *Scientia Iranica*, 26(6):3695–3711.

Namazian, A. and Yakhchali, S. H. (2016). Modeling and solving project portfolio and contractor selection problem based on project scheduling under uncertainty. *Procedia - Social and Behavioral Sciences*, 226:35–42.

Ning, D., Dongdong, G., Martin, F., and Zuhair, H. (2012). A genetic algorithm-based method for look-ahead scheduling in the finishing phase of construction projects. *Advanced Engineering Informatics*, 26(4):737–748.

Pérez, E., Posada, M., and Lorenzana, A. (2016). Taking advantage of solving the resource constrained multi-project scheduling problems using multi-modal genetic algorithms. *Soft Computing*, 20(5):1879–1896.

Pinto, J. K. and Prescott, J. E. (1990). Planning and tactical factors in the project implementation process. *Journal of Management Studies*, 27(3):305–327.

Pritsker, A., Watters, L., and Wolfe, P. (1969). Multiproject scheduling with limited resources: A zero-one programming approach. *Management Science*, 16(1):93–108.

Rokou, E., Dermitzakis, M., and Kirytopoulos, K. (2014). Multi-project flexible resource profiles project scheduling with ant colony optimization. In *2014 IEEE International Conference on Industrial Engineering and Engineering Management*, pages 642–646. IEEE.

Rostami, M. and Bagherpour, M. (2020). A lagrangian relaxation algorithm for facility location of resource-constrained decentralized multi-project scheduling problems. *Operational Research*, 20(2):857–897.

Satic, U., Jacko, P., and Kirkbride, C. (2022). Performance evaluation of scheduling policies for the dynamic and stochastic resource-constrained multi-project scheduling problem. *International Journal of Production Research*, 60(4):1411–1423.

Sebastiani, R. and Trentin, P. (2020). OptiMathSAT: A tool for optimization modulo theories. *Journal of Automated Reasoning*, 64(3):423–460.

Shariatmadari, M., Nahavandi, N., Zegordi, S. H., and Sobhiyah, M. H. (2017). Integrated resource management for simultaneous project selection and scheduling. *Computers & Industrial Engineering*, 109:39–47.

Shi, Y., Du, Z., and Li, J. (2019). Hierarchy model for distributed resource constrained multi-project scheduling problem. In *Proceedings of the 2019 5th International Conference on Industrial and Business Engineering*, pages 123–127. ACM.

Shou, Y., Xiang, W., Li, Y., and Yao, W. (2014). A multiagent evolutionary algorithm for the resource-constrained project portfolio selection and scheduling problem. *Mathematical Problems in Engineering*, 2014:302684.

Singh, A. (2014). Resource constrained multi-project scheduling with priority rules & analytic hierarchy process. *Procedia Engineering*, 69:725–734.

Song, W., Xi, H., Kang, D., and Zhang, J. (2018). An agent-based simulation system for multi-project scheduling under uncertainty. *Simulation Modelling Practice and Theory*, 86:187–203.

Sonmez, R. and Uysal, F. (2015). Backward-forward hybrid genetic algorithm for resource-constrained multiproject scheduling problem. *Journal of Computing in Civil Engineering*, 29(5):04014072.

Speranza, M. G. and Vercellis, C. (1993). Hierarchical models for multi-project planning and scheduling. *European Journal of Operational Research*, 64(2):312–325.

Suresh, M., Dutta, P., and Jain, K. (2015). Resource constrained multi-project scheduling problem with resource transfer times. *Asia-Pacific Journal of Operational Research*, 32(6):1550048.

Suy, J. (2012). *A satisfiability modulo theories approach to constraint programming*. Ph.D. thesis, Universitat de Girona, Spain.

Tasan, S. O. and Gen, M. (2013). An integrated selection and scheduling for disjunctive network problems. *Computers & Industrial Engineering*, 65(1):65–76.

Tayyar, S. G., Lamothe, J., and Dupont, L. (2016). Genetic algorithm for generalized resource constrained multi project scheduling problem integrated with closed loop supply chain planning. In *2016 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, pages 1683–1687. IEEE.

Tian, J., Dong, X., and Han, S. (2018). Optimizing for a resource-constrained multi-project scheduling problem with planned resource unavailability. In *2018 3rd International Conference on Modelling, Simulation and Applied Mathematics (MSAM 2018)*, pages 243–248. Atlantis Press.

Tian, M., Liu, R. J., and Zhang, G. J. (2020). Solving the resource-constrained multi-project scheduling problem with an improved critical chain method. *Journal of the Operational Research Society*, 71(8):1243–1258.

Toffolo, T. A., Santos, H. G., Carvalho, M. A., and Soares, J. A. (2016). An integer programming approach to the multimode resource-constrained multiproject scheduling problem. *Journal of Scheduling*, 19(3):295–307.

Tsai, D. M. and Chiu, H. N. (1996). Two heuristics for scheduling multiple projects with resource constraints. *Construction Management and Economics*, 14(4):325–340.

Tsubakitani, S. and Deckro, R. F. (1990). A heuristic for multi-project scheduling with limited resources in the housing industry. *European Journal of Operational Research*, 49(1):80–91.

Van Den Eeckhout, M., Vanhoucke, M., and Maenhout, B. (2021). A column generation-based diving heuristic to solve the multi-project personnel staffing problem with calendar constraints and resource sharing. *Computers & Operations Research*, 128:105163.

Van Eynde, R. and Vanhoucke, M. (2020). Resource-constrained multi-project scheduling: Benchmark datasets and decoupled scheduling. *Journal of Scheduling*, 23(3):301–325.

Vázquez, E. P., Calvo, M. P., and Ordóñez, P. M. (2015). Learning process on priority rules to solve the RCMPSP. *Journal of Intelligent Manufacturing*, 26(1):123–138.

Vercellis, C. (1994). Constrained multi-project planning problems: A Lagrangean decomposition approach. *European Journal of Operational Research*, 78(2):267–275.

Villafáñez, F., Poza, D., López-Paredes, A., Pajares, J., and del Olmo, R. (2019). A generic heuristic for multi-project scheduling problems with global and local resource constraints (RCMPSP). *Soft Computing*, 23(10):3465–3479.

Voß , S. and Witt, A. (2007). Hybrid flow shop scheduling as a multi-mode multi-project scheduling problem with batching requirements: A real-world application. *International Journal of Production Economics*, 105(2):445–458.

Wang, J., Hu, X., Demeulemeester, E., and Zhao, Y. (2021). A bi-objective robust resource allocation model for the RCPSP considering resource transfer costs. *International Journal of Production Research*, 59(2):367–387.

Wang, L., Zhan, D., Nie, L., and Xu, X. (2013). Schema and solutions for decentralised multi-project scheduling problem. *International Journal of Computer Applications in Technology*, 46(2):142–154.

Wang, X., Chen, Q., Mao, N., Chen, X., and Li, Z. (2015). Proactive approach for stochastic RCMPSP based on multi-priority rule combinations. *International Journal of Production Research*, 53(4):1098–1110.

Wang, Y., He, Z., Kerkhove, L.-P., and Vanhoucke, M. (2017). On the performance of priority rules for the stochastic resource constrained multi-project scheduling problem. *Computers & Industrial Engineering*, 114:223–234.

Wauters, T., Kinable, J., Smet, P., Vancroonenburg, W., Berghe, G. V., and Verstichel, J. (2016). The multi-mode resource-constrained multi-project scheduling problem. *Journal of Scheduling*, 19(3):271–283.

Wauters, T., Verbeeck, K., De Causmaecker, P., and Berghe, G. V. (2015). A learning-based optimization approach to multi-project scheduling. *Journal of Scheduling*, 18(1):61–74.

Xin, S., Su, Q., Wang, Q., and Wang, Q. (2018). Optimization of resource-constrained multi-project scheduling problem based on the genetic algorithm. In *2018 15th International Conference on Service Systems and Service Management (ICSSSM)*, pages 1–6. IEEE.

Xu, J. and Feng, C. (2014). Multimode resource-constrained multiple project scheduling problem under fuzzy random environment and its application to a large scale hydropower construction project. *The Scientific World Journal*, 2014:463692.

Xu, J. and Zhang, Z. (2012). A fuzzy random resource-constrained scheduling model with multiple projects and its application to a working procedure in a large-scale water conservancy and hydropower construction project. *Journal of Scheduling*, 15(2):253–272.

Zaman, F., Elsayed, S., Sarker, R., Essam, D., and Coello Coello, C. A. (2021). An evolutionary approach for resource constrained project scheduling with uncertain changes. *Computers & Operations Research*, 125:105104.

Zapata, J. C., Hodge, B. M., and Reklaitis, G. V. (2008). The multimode resource constrained multiproject scheduling problem: Alternative formulations. *AIChE Journal*, 54(8):2101–2119.

Zhang, L. and Sun, R. (2011). An improvement of resource-constrained multi-project scheduling model based on priority-rule based heuristics. In *8th International Conference on Service Systems and Service Management (ICSSSM)*, pages 1–5. IEEE.

Zhang, Z. and Chen, M. (2018). A bi-level multi-agent system model for decentralized multi-project scheduling of wind power plants. *Journal of Renewable and Sustainable Energy*, 10(3):035502.

Zheng, Z., Guo, Z., Zhu, Y., and Zhang, X. (2014). A critical chains based distributed multi-project scheduling approach. *Neurocomputing*, 143:282–293.

Zhu, H., Lu, Z., and Hu, X. (2018). A modified heuristic algorithm for resource constrained multi-project scheduling problem based on inspection and rework. In *2018 IEEE 14th International Conference on Automation Science and Engineering (CASE)*, pages 1058–1063. IEEE.

Zhu, H., Lu, Z., Lu, C., and Ren, Y. (2021). Modeling and algorithm for resource-constrained multi-project scheduling problem based on detection and rework. *Assembly Automation*, 41(2):174–186.

Zhuang, M. and Yassine, A. A. (2004). Task scheduling of parallel development projects using genetic algorithms. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, volume 1, pages 215–224. ASME.