systems

MDPI

Article

A Novel Solution Approach Based on Dominance Evaluation Measure for Project Scheduling in Multi-Project Environments

Hamid Reza Yousefzadeh [1,*], Erfan Babaee Tirkolaee [2,3,4] and Farzad Kiani [5]

[1] Department of Mathematics, Payame Noor University (PNU), Tehran P.O. Box 19395-4697, Iran
[2] Department of Industrial Engineering, Istinye University, Istanbul 34396, Turkey; erfan.babaee@istinye.edu.tr
[3] Department of Industrial Engineering and Management, Yuan Ze University, Taoyuan 320315, Taiwan
[4] Department of Mechanics and Mathematics, Western Caspian University, Baku 1001, Azerbaijan
[5] Data Science Application and Research Center (VEBIM), Fatih Sultan Mehmet Vakif University, Istanbul 34445, Turkey; fanka@fsm.edu.tr
* Correspondence: usefzadeh.math@pnu.ac.ir

Abstract: The widely recognized measure for resources called resource strength (RS) does not fully capture the resources complexity of a project. Therefore, it cannot be used as a standalone measure to distinguish the complexity of various instances of project scheduling problems. Consequently, additional resource measures such as total amount of overflow (TAO) have been introduced, which should be used in conjunction with the RS. Extensive experimental studies have shown that as the value of TAO increases in a project, scheduling schemes with higher dimensional scheduling schemes such as bi-directional and tri-directional result in schedules with shorter makespans. In this study, an effective approach is proposed for integrating projects in multi-project environments, called the integrated project approach (IPA), taking into account the influence of TAO and building upon the relation between the TAO and the scheduling generation schemes. To assess the performance of IPA, we develop a new random multi-project generator based on the well-known benchmark sets, which utilizes TAO as a control tool to generate instances. The findings indicate that prioritizing the projects and frequency of the projects integration, facilitated by the proposed IPA, have a positive impact on the quality of multi-project schedules.

Keywords: project scheduling; resource complexity measure; multi-dimensional scheduling scheme; multi-project environment; dominance evaluation measure

check for updates

Citation: Yousefzadeh, H.R.; Tirkolaee, E.B.; Kiani, F. A Novel Solution Approach Based on Dominance Evaluation Measure for Project Scheduling in Multi-Project Environments. *Systems* **2024**, *12*, 476. https://doi.org/10.3390/systems12110476

Academic Editor: Wen-Chyuan Chiang

Received: 12 September 2024
Revised: 13 October 2024
Accepted: 21 October 2024
Published: 7 November 2024

1. Introduction and Background

Scheduling is a fundamental aspect of project management and has a direct impact on overall management performance. The resource-constrained project scheduling problem (RCPSP) has primarily been the emphasis of research efforts in this field, with a wide range of applications [1,2]. The RCPSP is recognized as an NP-hard problem [3]. Projects are typically scheduled while considering the constraints imposed by the availability of resources. Concurrent execution of projects is employed to effectively utilize the limited resources. The resource-constrained multi-project scheduling problem (RCMPSP) is an extension of the RCPSP. It involves scheduling multiple projects that have precedence constraints and competing for the same set of scarce resources [4].

The RCMPSP represents a portfolio of individual projects, each of which satisfies its required resources from a shared pool of renewable and non-renewable resources. This resource pool contains various resource types with varying levels of availability. However, the renewable resources are insufficient to schedule all eligible activities from different projects simultaneously. The primary objective of scheduling is to prioritize eligible activities and then schedule them to optimize the specific objective function. The selection of eligible activities from projects for scheduling is of the utmost importance. The RCMPSP encompasses different types of objective functions, including (i) minimizing the overall project delay, lateness, or tardiness [5]; (ii) minimizing the average project delay [6]; (iii) minimizing the total lateness or lateness penalty [7]; (iv) minimizing the overall project cost [8]; (v) minimizing the delay cost [7] and (vi) resource-leveling [9]. The RCMPSP can be examined in static environments in which the attributes of the projects are assumed to be deterministic and static (e.g., [5,6,10]). It can also be studied in dynamic environments in which new projects or activities may be added during the scheduling process, potentially impacting the scheduling procedure (e.g., [9,11,12]).

The classification of solution methods applied to the RCMPSP and related variants includes both exact and approximate algorithms [13]. Due to the complexity of the RCMPSP, exact methods have limited applicability when addressing real-world instances of the problem. For an overview of exact solution approaches, references such as [14,15], can be consulted for constraint programming. Satic et al. [16] provided insights into dynamic programming, while Davari Ardakani and Dehghani [17] and Van Den Eeckhout et al. [18] focused on linear-programming-based methods. Heuristic approaches have played a crucial role in developing feasible and robust algorithms for solving the RCMPSP. Each heuristic algorithm possesses unique characteristics and can be tailored to meet the specific requirements of the problem at hand. However, no single heuristic algorithm is universally superior; the choice of method often depends on various factors, including the characteristics of the problem, the desired balance between solution quality and computation time, and the availability of computational resources. Hybrid algorithms are typically categorized within the set of approximate algorithms. Approximate solution methods include genetic algorithm (GA) (e.g., [16,19,20]), simulated annealing [21], and swarm intelligence [22]. Other methods such as fuzzy clustering chaotic-based differential evolution algorithm [23], tabu search (TS) (refer to e.g., [21,24]), and multi-agent systems and combinatorial auction (e.g., [19,25]) have also contributed significantly to this area.

It is necessary to mention that the priority rule base can be classified in heuristics and they are used in the design of the algorithms. Browning and Yassine [26] addressed the static RCMPSP with project and portfolio lateness as objective functions. They conducted an experimental analysis of 20 priority rules on 12,320 test problems, which were generated based on project-related characteristics. They also incorporated improved resource measures in the multi-project environments. In another study, Browning and Yassine [27] also analyzed 31 priority rules on 18,480 portfolios comprising 55,440 iterative (cyclical) projects. They showed that the best priority rules for cyclical project portfolios differ from those for acyclic projects. Yassine et al. [28] developed two GA-based heuristics for scheduling activities in product development (PD) projects with the objective of minimizing the makespan. Their study showed that the two proposed GAs converge to globally optimal solutions more quickly than when using 31 published priority rules.

There are generally two approaches used to address the RCMPSP: single project approach (SPA) and multi-project approach (MPA). In the SPA, projects are integrated sequentially to form a super-project, and activities are scheduled based on a single critical path within the super-project (e.g., [5,7,26]). However, this integration may be considered as a drawback since conducting an independent analysis of each project becomes challenging, if not impossible. On the other hand, in the MPA, projects are analyzed separately, and activities within each project are scheduled based on the critical path specific to that project (e.g., [7,26]).

Effective benchmarking is essential for advancing research in the RCMPSP, yet the current literature reveals a scarcity of dedicated generators for these problems specifically designed for these problems. Benchmark sets for the RCMPSP are developed by researchers who incorporate essential elements to capture characteristics examined during their numerical analyses. Table 1 summarizes a review of the RCMPSP benchmark library and existing generators.

Table 1. Overview of RCMPSP benchmark libraries and generators.

Source	Description	Year
Homberger [29,30]	Established the mpsplib [31], a library with 140 DRCMPSP instances organized into problem sets of $M = 2, 5, 10, 20$ single-project instances. Each multi-project instance is derived from Kolisch benchmark (PSBLIB). An additional 60 instances were added in 2012 considering project access to local resources.	2007, 2012
Browning and Yassine [32]	Developed the MNPG generator for RCMPSP, producing random activity-on-node network problems with parameters including network complexity, resource loading, and contention. Generated 12,320 instances by modifying resource measures such as ARLF and AUF.	2010
Pérez et al. [10]	Created the RCMPSPLIB benchmark set using instances from mpsplib [31] and MNPG.	2016
Van Eynde and Vanhoucke [20]	Introduced a dataset for RCMPSP consisting of instances with 6, 12, and 24 projects, each with 60 activities. Assessed the extension of single-project scheduling schemes for multiple projects and proposed decoupled versions of these schemes.	2020

This study makes several significant contributions to the field of the RCMPSP by investigating the static RCMPSP and introducing a novel resource measure, total amount of resource overflows (TAO), tailored specifically for multi-project environments. Additionally, we propose an innovative integrated project approach (IPA) that incorporates a tri-directional schedule generation scheme (trdss) cite40, compared against established approaches, namely, the SPA and the MPA, demonstrating superior effectiveness. We also develop a new random multi-project generator, named GTAO, utilizing TAO as a control parameter, thus providing robust tools for simulating RCMPSP environments. Finally, this study proposes using project delay as a measure of efficiency to evaluate various multi-project solution approaches, introducing a new perspective on the quality of Pareto solutions through a dominance measure.

The remainder of the paper is organized as follows: Section 2 briefly outlines the problem we are addressing. Section 3 offers an overview of two well-established heuristic approaches commonly employed in a multi-project scheduling environment. Section 4 justifies the need for utilizing TAO in a project. Section 5 presents the IPA, including the algorithm for project integration and a brief introduction to the tri-directional schedule generation scheme. In Section 6, we develop a novel RCMPSP generator (GTAO) that uses the TAO as a control parameter. Section 7 discusses experimental results. Finally, in Section 8, we present our conclusions.

2. Problem Description

The presented paper focuses on the RCMPSP, which is described as follows: The RCMPSP involves a set of M projects, all having equal priority. Each project i consists of N_i activities, where $i = 1, \ldots, M$. The total number of activities across all projects is denoted as N, where $N = \sum_{i=1}^{M} N_i$. In this problem, each activity j of project i has a deterministic duration d_{ji} where $j \in J_i$ $(\mid J_i \mid = N_i)$. No preemption is allowed. Minimization of projects makespans is the objective function. Precedence and resource constraints govern the relationship between the activities within each project. Precedence relations are defined by the minimum finish-to-start between activities in each project.

Furthermore, the availability of renewable resources plays a significant role. The total amount of the renewable resource of type k per unit of time is denoted by R_k, where k represents a specific resource type, and it belongs to the set R, which is defined as $R = \{1, \ldots, K\}$. In this context, K is the total number of different renewable resource types, where $K{=}\mid R \mid$. The amount of resource k required by activity j of project i is represented by r_{jik}. However, it should be noted that the value of R_k does not suffice to simultaneously perform all eligible activities. Consequently, during the scheduling process, start times are assigned to activities while considering both the precedence and the resource constraints. The aim is to optimize a specific measure of project completion times in the RCMPSP. For instance, let T_i represent the makespan of project p_i $(i = 1, \ldots, M)$ as determined by a specific scheduling method. The quality of feasible solutions produced by this method across M projects is assessed by minimizing the following multi-objective function:

$$\text{minimize} \quad F(p_1, \ldots, p_M) = \left(\frac{T_1}{LB_1}, \frac{T_2}{LB_2}, \ldots, \frac{T_M}{LB_M} \right),$$

where LB_i denotes the length of the critical path for project p_i.

3. Current Solution Procedures for RCMPSP

Typically, two primary heuristic approaches are employed for solving instances of the RCMPSP. In this section, we provide a brief description of these approaches. While these approaches do not guarantee optimal solutions, they offer practical and efficient strategies for tackling challenges associated with scheduling multiple projects that share resources and constraints.

Given that the implementation of IPA relies on a new resource measure known as TAO, we defer its introduction to Section 5 until after describing the features of TAO.

3.1. Single Project Approach (SPA)

In the SPA, two dummy activities are introduced to connect the start and the endpoints of projects that belong to a multi-project problem. This results in the creation of a single super-project that encompasses all the individual projects. For a more comprehensive understanding, refer to, e.g., [5–7]. The length of a critical path for this super-project is considered as the maximum length among all the critical paths of the individual projects in the given instance of the RCMPSP. In other words, in a problem with M projects, if LB_i represents the length of a critical path of project i, where $i = 1, \ldots, M$, then the length of the critical path for the single super-project is given by

$$LB_{\max} = \max\{LB_i | \ i = 1, \ldots, M\}.$$

Notice that $LB_{\max}$ is used to determine the priority of activities for scheduling. The main advantage of this approach is the existence of a collection of research results concerning the related problem i.e., the RCPSP, which can be applied to the analysis of the RCMPSP (e.g., [5,8,33]). However, it should be acknowledged that this approach does not take into account the specific characteristics of each individual project, making independent analysis of each project more challenging.

3.2. Multi-Project Approach (MPA)

In contrast to the SPA, MPA considers the critical path of each individual project separately. This approach utilizes the critical path of each project to prioritize eligible activities for scheduling. It is worth noting that the choice of critical path significantly influences the performance of both the SPA and the MPA [5]. The primary advantage of the MPA is that it is more realistic, making it more commonly employed in practical applications.

4. Resource Characteristics

For single projects, various measures and distributions have been proposed in the literature to describe resource characteristics. Notable among these resource measures are resource factor—RF [34], resource strength—RS [34], resource density [35], and resource constrainedness [36]. However, Yousefzadeh et al. demonstrated through extensive numerical results that no individual resource measure can fully capture the complexity of projects concerning resources [37]. To illustrate this point, consider the RS as follows:

$$RS_k = \frac{R_k - \max_{j \in J} r_{kj}}{Peak_k - \max_{j \in J} r_{kj}} \quad \forall k \in R. \tag{1}$$

Here, the resource requirement $Peak_k$, where $k \in R$, is defined as

$$Peak_k = \max\{\sum_{j \in A(t)} r_{jk} \mid 0 \leq t \leq T-1\}.$$

In the early schedule, where each activity is scheduled at its earliest start time, $Peak_k$ represents the peak requirement for resource k. The symbol T denotes the total makespan of a multi-project problem P, and $A(t)$ represents the set of activities that are active at time t. Considering (1), it is clear that $Peak_k$ impacts the magnitude of RS_k. However, the effect of the number of $Peak_k$s and also the locations at which these peaks in resource requirements occur throughout the project's life cycle are not addressed in (1). The rationale behind considering the aforementioned factors stems from the observation that scheduling a project with multiple resource peaks, particularly when those peaks occur towards the end of a project's life cycle, appears to be more complicated than scheduling a project with a single resource peak that occurs during the early stages of the project's life cycle. A detailed analysis on the library PSBLIB, regarding the number of resource peaks, reveals that projects with the same level of RS_k do not necessarily have the same number of peaks (refer to Table 2).

Table 2. Different levels of RS vs. number of resource peaks.

Instance	RS	No. of Peaks	Instance	RS	No. of Peaks
J909-1	0.2	1	J9038-2	0.5	5
J905-6	0.2	2	J12025-6	0.5	4
J9021-4	0.2	3	J9030-1	0.5	3
J9037-7	0.2	5	J12025-5	0.5	1
J1201-4	0.2	4	J906-8	0.5	2

Building on the aforementioned observation, a new resource measure, TAO, was proposed by [37]. Unlike the RS measure, which focuses solely on the number of resource peaks, the TAO considers both the number of resource peaks and their positions throughout the project's life cycle. The researchers argued that classifying projects based on the TAO offers a more accurate representation than classifications based on the RS measure. Furthermore, they demonstrated that the TAO value for a particular resource k has a somewhat inverse relationship with the RS_k, and this relationship exhibits a perfect inverse linear relation only when some related parameters are held constant (see [37] for further details).

The measure TAO quantifies the surplus resources within a project, reflecting the extent to which available resources exceed the required or expected levels. This measure assesses resource utilization efficiency and identifies potential excess that can be harnessed for enhanced performance or effectiveness. By measuring this, the TAO provides insights into resource allocation and management across various contexts, highlighting the significance of optimizing resource distribution.

In contrast to the RS, which has a myopic view of resource conflicts, the TAO takes a far-sighted view of resource conflicts throughout the project's life cycle.

5. Integrated Project Approach-IPA

The IPA is fundamentally based on the MPA but differs in its project integration methodology. To facilitate this integration, the IPA utilizes the TAO as a resource-based measure, which assesses the complexity and level of dependency of projects concerning the available resources (refer to Section 4). Previous experimental studies by Yousefzadeh et al. demonstrated that as the TAO increases in a project, higher-dimensional scheduling schemes, such as the bi-directional schedule generation scheme (bidss) and especially, the trdss, produce schedules with shorter makespans [37]. Taking advantage of this finding, the IPA initially calculates the TAO for each project and then integrates those with common TAO areas to create a collection of projects with higher overall TAO values compared to individual projects. In the final phase, the integrated projects are scheduled using the trdss. This integration process not only aims to enhance the TAO of the resulting projects but also involves prioritizing eligible activities from different projects at the scheduling decision time, as the prioritization of activities significantly impacts the performance of scheduling schemes.

5.1. Algorithm for IPA

Algorithm 1 provides a detailed explanation of the project integration process.

Algorithm 1 Projects integration in an RCMPSP.

```
Input:
- An RCMPSP generated by GTAO, comprising $M$ projects with durations,
- $LB_i$ where $i = 1, \dots, M$.
Output: An RCMPSP with $M'$ projects, where $M' < M$.
Generate matrix $M_{TAO} = (m_{ij})_{M \times T}$, where $T = \max_{i=1}^{M} LB_i$ as follows:
for: $t = 1, \cdots, T$
if there is a resource overflow at time instant $t$ of project $i$ then
    $m_{it} = 1$.
else
    $m_{it} = 0$.
end if
if $t > LB_i$ then
    $m_{it} = 0$.
end if
end for
Integrate the $M$ projects using $M_{TAO}$ as follows:
while there is a row in $M_{TAO}$ do
    Calculate the column sums of $M_{TAO}$.
    Save column numbers whose sum is the largest in $\mathbf{C}$.
    if $|\mathbf{C}| > 1$ then
        Define $L$ as the number of rows in $M_{TAO}$.
        Choose the column $c \in \mathbf{C}$ with the highest rank of each as follows:
        $\forall c \in \mathbf{C}$ calculate the sum of row sums (rank of $c$) belonging to those rows where
        $M_{TAO}[i, c] = 1,\ i = 1 \dots L$.
        Find the maximum sum of the row sums if the maximum is not unique, randomly
        select one column).
        Let the maximum sum of the row sums belong to $c' \in \mathbf{C}$.
    else
        Assign the single maximum column as $c'$.
    end if
    Integrate projects satisfying $M_{TAO}[i, c'] = 1,\ (i = 1 \dots L)$.
    Delete rows from $M_{TAO}$ that satisfy $M_{TAO}[i, c'] = 1,\ (i = 1 \dots L)$.
end while
```

To demonstrate Algorithm 1, we will provide a small example.

Example 1. *Let us assume that in a multi-project problem, there are four projects, $p_i\ (i = 1, \dots, 4)$, with a single resource type. Additionally, we assume that $LB_1 = 3$, $LB_2 = 4$, $LB_3 = 6$, and $LB_4 = 5$. The distribution of the resource overflows for the four projects is given in the matrix M_{TAO} as follows:*

$$M_{TAO} = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}.$$

We will now illustrate the steps of Algorithm 1 to integrate these four projects regarding the given matrix M_{TAO}.

Step 1: Calculate column sums of M_{TAO} and identify the maximum column(s)

- The column sums are computed as follows:
  Column 1 (C_1): 2, Column 2 (C_2): 2;
  Column 3 (C_3): 3, Column 4 (C_4): 3;
  Column 5 (C_5): 1, Column 6 (C_6): 1.

  Therefore, the column sums can be expressed as

$$\mathbf{C} = [C_1, C_2, C_3, C_4, C_5, C_6] = [2, 2, 3, 3, 1, 1].$$

- The maximum column sums are found in Columns 3 and 4.

Step 2: Determine the ranks of the maximum columns (3 and 4)

- To calculate the rank of each maximum column, sum the non-zero entries of the corresponding rows:
  Rank of Column 3:

$$\sum_{j=1}^{4} m_{1j} + \sum_{j=1}^{4} m_{2j} + \sum_{j=1}^{4} m_{4j} = (1+1) + (1+1+1) + (1+1+1) = 8.$$

  Rank of Column 4:

$$\sum_{j=1}^{4} m_{2j} + \sum_{j=1}^{4} m_{3j} + \sum_{j=1}^{4} m_{4j} = (1+1+1) + (1+1+1+1) + (1+1+1) = 10.$$

- Since the rank of Column 4 (10) is greater than the rank of Column 3 (8), Column 4 is selected for integration.

Step 3: Integrate projects corresponding to the selected column

- The non-zero entries in Column 4 correspond to Projects 2, 3, and 4. These projects are integrated, and their corresponding rows are deleted from the M_{TAO}.

Step 4: Update M_{TAO} and assess remaining projects

- After integrating Projects 2, 3, and 4, the updated matrix M_{TAO} has only one remaining row for Project 1:

$$M_{TAO} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

- Since there are no other projects to integrate with project 1, it is processed individually. row 1 is deleted from M_{TAO}, and the algorithmconcludes.

5.2. Computational Complexity Analysis

To analyze the computational complexity of the proposed algorithm for project integration in the resource-constrained multi-project scheduling problem (RCMPSP), we examine each significant part of the algorithm and identify its time complexity.

- Execution Steps of the Algorithm:

1. Generate Matrix M_{TAO}: The matrix $M_{TAO} = (m_{ij})_{M \times T}$ is generated where M is the number of projects and T is the maximum LB_i (lower bound of durations). The algorithm iterates over each project for each time instant. This results in a complexity of

$$O(M \times T).$$

2. Integrate the Projects: The algorithm uses a while loop that continues as long as there are rows in M_{TAO}. The inner operations include

a. Column Sums Calculation: Calculating the column sums requires traversing each column, which has a time complexity of $O(M)$ for the entire T columns, resulting in
$$O(M \times T).$$

b. Choosing the Best Column: This involves checking column sums again and ranks. In the worst case, when multiple columns yield the maximum sums, we will need to sum the row sums for the selected columns, which takes $O(M)$ time for each of the $|C|$ columns.

Assuming that the while loop could potentially run up to $O(M)$ times (in the case where projects are closely integrated), the inner steps could lead to a worst-case scenario of
$$O(M^2 \times T),$$
where each integration reduces the number of projects (rows) in M_{TAO} during each iteration.

Combining all these steps, under the worst-case scenario, the total time complexity can be formally represented as follows:

- Matrix generation: $(O(M \times T))$

While loop iterations (up to $O(M)$) impose costs within each iteration as $O(M \times T)$: $O(M^2 \times T)$.

Thus, the comprehensive time complexity can be approximated as
$$O(M^2 \times T).$$

The algorithm's complexity is notably influenced by the input size, particularly the number of projects M and the time horizon T. In many practical scenarios, because project integrations frequently reduce the number of projects considered, the actual performance can be better than the worst-case complexity suggests.

Before presenting our computational results, we provide a brief explanation of the trdss and refer the reader to [37] for more details.

5.3. Tri-Directional Scheduling Scheme—Trdss

The trdss is a schedule generation scheme that consists of two typical phases [37]. In the first phase, activities are scheduled in three different directions. Activities that have all their predecessors scheduled are considered in the forward sub-schedule (referred to as the forward direction). Activities that have all their successors scheduled are considered in the backward sub-schedule (referred to as the backward direction). Finally, activities that have both their predecessors and successors scheduled are included in the midway sub-schedule (referred to as the midway direction). It is worth noting that the trdss implements the parallel scheduling scheme (PSS). In the second phase, activities from three sub-schedules are left-shifted to construct the complete schedule. Initially, activities from the midway sub-schedule are left-shifted, followed by activities from the backward sub-schedule. Activities from the midway sub-schedule are left-shifted in the order of their start times, taking into account precedence and resource constraints. Subsequently, activities from the backward sub-schedule are left-shifted in non-decreasing order of their scheduled start times.

Studies have shown that the trdss outperforms other state-of-the-art scheduling schemes concerning their makespans, particularly when the level of TAO increases [37].

6. Development of a New RCMPSP Generator

The availability of libraries for RCMPSPs is crucial for advancing the field of project scheduling. They provide a common ground for researchers to compare their results and contribute to developing more efficient and robust algorithms for solving the RCMPSP. To effectively assess the performance of IPA, it is necessary to develop a new RCMPSP generator that can generate multi-project instances based not only on the conventional measures of network and resource complexities, but also on the TAO.

None of the existing generators or instance libraries take the TAO into account as a control parameter. To elaborate further, we applied the suggested IPA on the largest set of the mpsplib [31], which consists of problems with 20 projects and 120 activities (referred to as "mp_j120_a20_nr5_AgentCopp_set"). The experimental results demonstrate that when Algorithm 1 is employed, the resource overflows in the mpsplib instances [31] are distributed in such a way that about approximately 98% of projects in a multi-project problem can be integrated into a single project (referred to as a super-project). In other words, by utilizing Algorithm 1, a multi-project problem with m distinct projects from the mpsplib [31] can be transformed into either a super-project or a new multi-project problem with only two projects, where 98% of projects are integrated into a single project. Hence, this particular multi-project benchmark does not fulfill our requirements. In general, if the resource overflows in the projects occur at the same locations, the performance of the IPA and the MPA is approximately the same and, therefore, there is no need for project integration. Hence, it is advantageous to invoke MPA in such cases.

Therefore, in this paper, we propose the development of a new generator for the RCMPSP, called GTAO. The GTAO incorporates measures of network and resource complexities, as well as the TAO concept. The generator uses RCPSP instances, including J30, J60, J90, and J120 instances from the PSPLIB benchmark as its foundation. To implement the TAO, we consider three following factors:

i. The length of the resource overflows;
ii. The time instants during the project's life cycle at which the resource overflow occurs;
iii. The magnitude of resource overflow at these times.

In other words, the GTAO generates random multi-projects by incorporating the aforementioned three factors. It is important to note that the generator already takes into account the usual measure complexities, as it is based on instances derived from Kolisch RCPSP benchmark [38].

The generation process of GTAO is as follows:

1. First, an RCPSP instance, denoted as p_i, is randomly selected from the set of $J30 \cup J60 \cup J90 \cup J120$;
2. The duration of p_i is divided into T_i units, each unit having a length of one (T_i represents the critical path of project i);
3. Resource overflow is then inserted at specific time instances during the life cycle of the RCPSP. The overflow has a given magnitude at those times, resulting in a random instance of RCMPSP.

The injection of resource overflow at time instant t is carried out using one of the following procedures:

a. Adjusting the amount of resource requirements for certain activities, either by increasing or decreasing them;
b. Dividing activities that are active at time t into sub-activities;
c. Inserting new activities or deleting some existing sub-activities.

In Example 2, we illustrate the mechanics of the GTAO process on a simple project. This can provide a clear understanding of how the process is applied in practice.

Example 2. *Let us consider a small project, denoted as p, which consists of four activities: A, B, C, and D. The project has a total available single resource, R_1 = 2 units per time period, as illustrated in Figure 1. Each box in the figure provides the name, duration, and resource requirements for the respective activities. The overall duration of the project is T = 5 time units. Additionally, Figure 2 shows the early schedule for project p.*

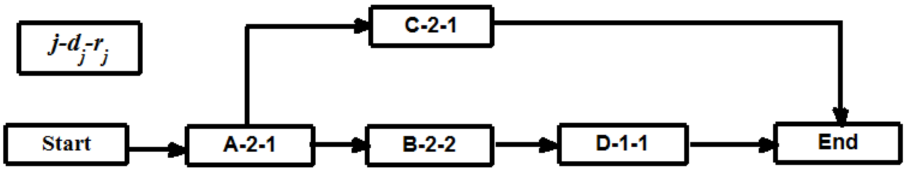


Figure 1. An example of a small project.

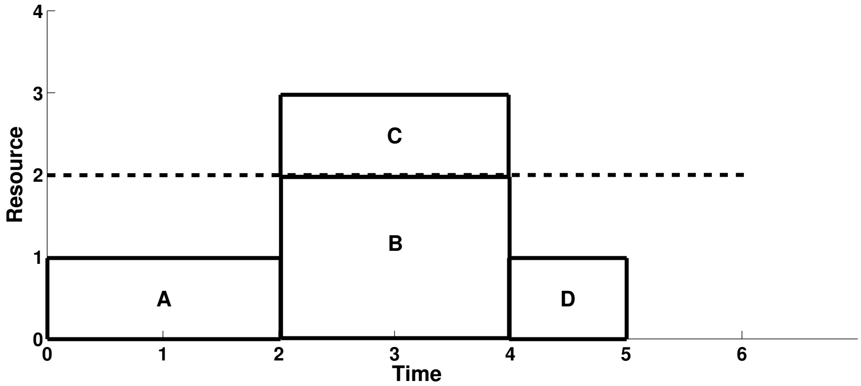


Figure 2. Early schedule for project p.

Now, let us assume that we want project p to experience three instances of resource overflow, each with a magnitude of one, occurring during the time intervals $[0, 1)$, $[2, 3)$, and $[4, 5)$. We can obtain a modified project, denoted as p_{new}, which includes these desired resource overflows, as follows:

1. Introduce a new activity, labeled as $A1$, during the time interval $[0, 1)$;
2. Divide activity C into two sub-activities ($C1$ and $C2$) and delete $C2$, which would have started at $t = 3$;
3. Increase the resource requirement of activity D, creating a new activity called $D1$.

The early schedule for project p_{new} is presented in Figure 3, where it is evident that the resource overflows occur during the specified time intervals throughout the project's life cycle.

It is worth noting that the introduction of $A1$ into the project introduces new precedence relations, as depicted in Figure 4.

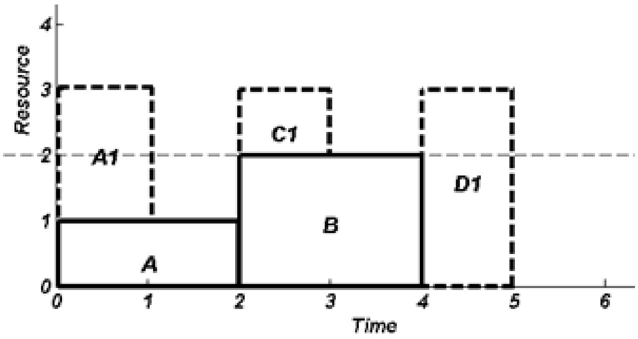


Figure 3. Early schedule for project p_{new}.

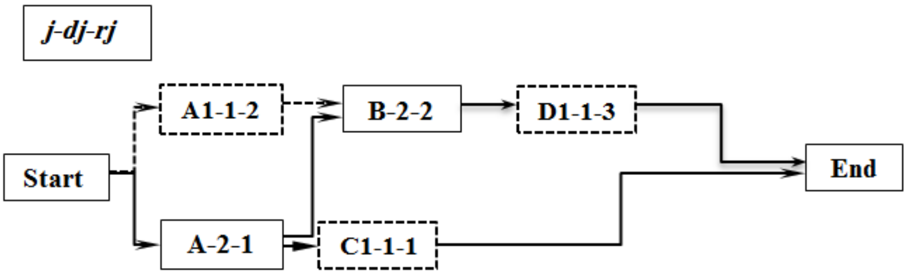


Figure 4. Activity-on-node representation of project p_{new}.

7. Computational Results

To assess the performance of the IPA and to compare it with other MPAs, 14 distinct portfolios of multi-projects are generated randomly. Each portfolio consists of between 10 and 23 projects. Generated projects differ in the overflows magnitude and the time intervals within which the overflows occur (each project is generated by the GTAO). From each portfolio, 30 random instances are generated.

7.1. Dominance Measure for the Objective Function

To compare the performance of different multi-project solution approaches, project delay is considered as a measure of efficiency. This objective function determines the quality of the solution obtained by different approaches. Let us assume that there are M projects, $p_1, \ldots, p_M$ in a multi-project problem P, and let us consider S solution approaches. If T_{si} represents the makespan of p_i obtained by applying solution approach s, then the RCMPSP with minimizing the makespan can be considered as a multi-objective optimization problem:

$$\text{minimize } \{F_s = (T_{s1}, \ldots, T_{sM}) \mid s = 1, \ldots, S\}.$$

Therefore, evaluating the performance of S solution approaches is equivalent to evaluating S Pareto vectors with a dimension of M.

He et al. [39] proposed a new definition for Pareto optimality when the number of objective functions exceeds five. Their definition is based on a fuzzy membership function. They indicated that when the size of the Pareto vector is greater than five, the majority of Pareto vectors will be classified as non-dominated vectors, making the accurate evaluation of Pareto solutions difficult. They assigned a crisp value to each Pareto vector and ranked the Pareto vectors accordingly. In this paper, we draw inspiration from their definition and present a new definition of Pareto-dominance to evaluate the performance of the $S = 3$ scheduling approaches, i.e., the SPA, MPA, and IPA. The following procedure explains how we can assign real values (referred to as the degree of closeness to a reference or ideal point) to objective vectors (on the Pareto front) generated by different scheduling approaches, allowing us to compare objective vectors accordingly.

Consider the objective vectors on S, $V_s = (T_{s1}, \cdots, T_{sM}) \in \mathbb{R}^M_{>0}$ where $s = 1, \ldots, S$, representing makespans of a multi-project problem P with M projects obtained by S scheduling approaches. Additionally, let the vector $LB = (LB_1, \cdots, LB_M)$ be a reference point, where LB_i represents the critical path of project p_i, with $i = 1, \ldots, M$. To define a dominance measure, we calculate the degree of closeness to the reference point LB for each solution approach s, denoted as $\Phi(V_s)$, where $\Phi(.) : \mathbb{R}^M \to \mathbb{R}$ is the Gaussian function defined as

$$\Phi(x) = \frac{1}{M} \sum_{i=1}^{M} \exp\left(-\frac{1}{2} \left(\frac{x - LB_i}{\sigma_P} \right)^2 \right),$$

with the parameter σ_P, representing the deviation from the reference point, defined as

$$\sigma_P = \frac{1}{lateness_{max}} \left(\min_{s=1}^{S} \{T_{si}\} - LB_i \right) \qquad (i = 1, \ldots, M),$$

where

$$lateness_{max} = \max_{i=1}^{M} \left\{ \min_{s=1}^{S} \{T_{si}\} - LB_i \right\}.$$

The value of $\Phi(.)$ serves as an evaluation measure. Therefore, we provide the following definition.

Definition 1 (Dominance measure)**.** *The objective vector V_A dominates the objective vector V_B if $\Phi(V_A) > \Phi(V_B)$*

According to Definition 1, we can conclude that solution approach A outperforms solution approach B, if the objective vector V_A dominates the objective vector V_B.

Remark 1. *If only the total completion time of a multi-project problem P is considered, the objective vectors corresponding to S different solution approaches are calculated as follows:*

$$V_s := (FT_s)_{1\times 1} \in \mathbb{R}^{\geq 0} \qquad (s = 1, \ldots, S),$$

where FT_s represents the finishing time of the last activity of problem P obtained by solution approach s $(s = 1, \ldots, S)$. In this case, for convenience, we denote the value of dominance $\Phi(V_s)$ by $\phi(s)$.

7.2. Experimental Results

In this section, we evaluate the performance of three solution approaches, i.e., the SPA, the MPA, and the IPA. The scheduling generation scheme used is trdss. We summarize the results of the dominance measure $\phi(i)$, where i = SPA and MPA, in Figure 5. It is important to note that the number of projects varies between 10 and 23. Figure 5a provides an example where in instances with 17 projects, the average dominance values of the SPA and the MPA solutions are 1.9943 and 2.1987, respectively. Thus, for this instance, the quality of the solution obtained from the MPA outperforms the SPA. However, the results are reversed for instances with 22 projects, where the dominance values of the SPA and the MPA solutions are 2.2417 and 2.1267, respectively.

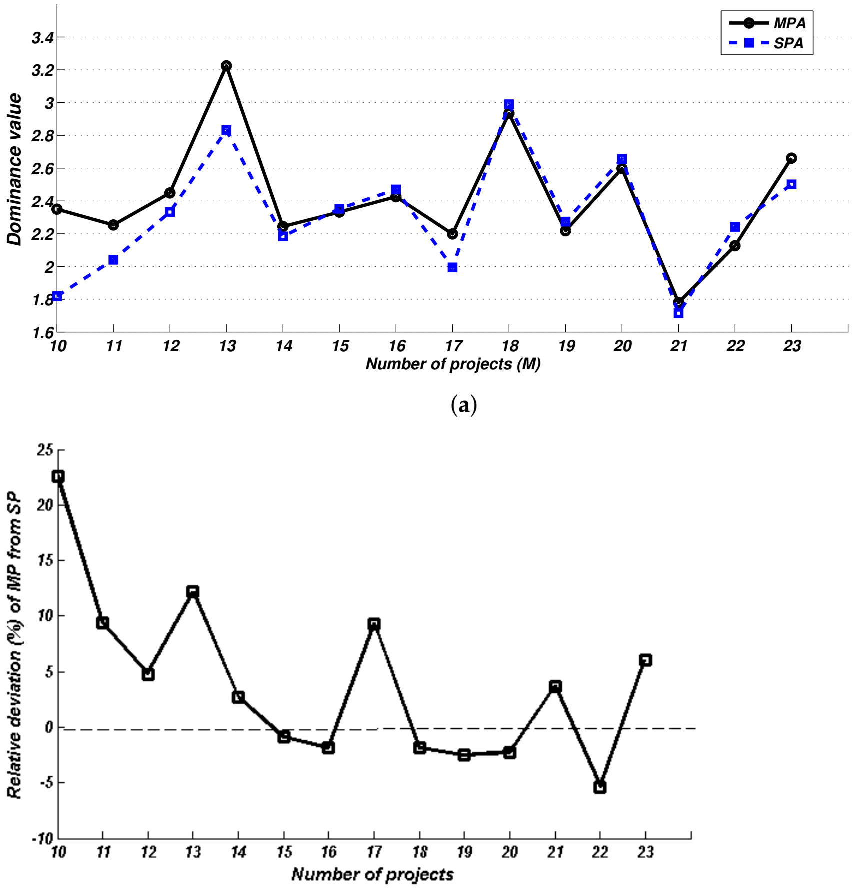
In Figure 5b, we consider relative deviations (%) of the MPA from the SPA using the following measure:

$$\text{Relative deviation} = \left(\frac{\phi(SPA) - \phi(MPA)}{\phi(SPA)} \right) \times 100.$$

We observe that for certain instances, prioritizing projects in an RCMPSP and then scheduling them by the MPA is more effective than using the SPA. This is evident for instances with $10, 11, 12, 13, 14, 17, 21$, and 23 projects. However, for other instances, the trend is reversed, such as instances with $15, 16, 18, 19, 20$, and 22 projects. In general, as the number of projects in an RCMPSP increases, the relative deviation of the MPA from the SPA decreases. Consequently, there is no significant difference between the two solution approaches.

Now, let us examine the impact of the proposed IPA on the scheduling of multi-project problems. The scheduling scheme used is trdss, as before. As mentioned earlier, in the IPA, projects of the RCMPSP are prioritized based on the TAO concept. Specifically, the IPA converts a problem with M_1 projects into a new problem with M_2 projects, where $M_2 < M_1$ (refer to Algorithm 1). The new problem is then solved by invoking the MPA, denoted as

IP(MPA). It is worth noting that the benchmark instances used are the same as those in Figure 5, and results are illustrated in Figure 6.



(**a**)

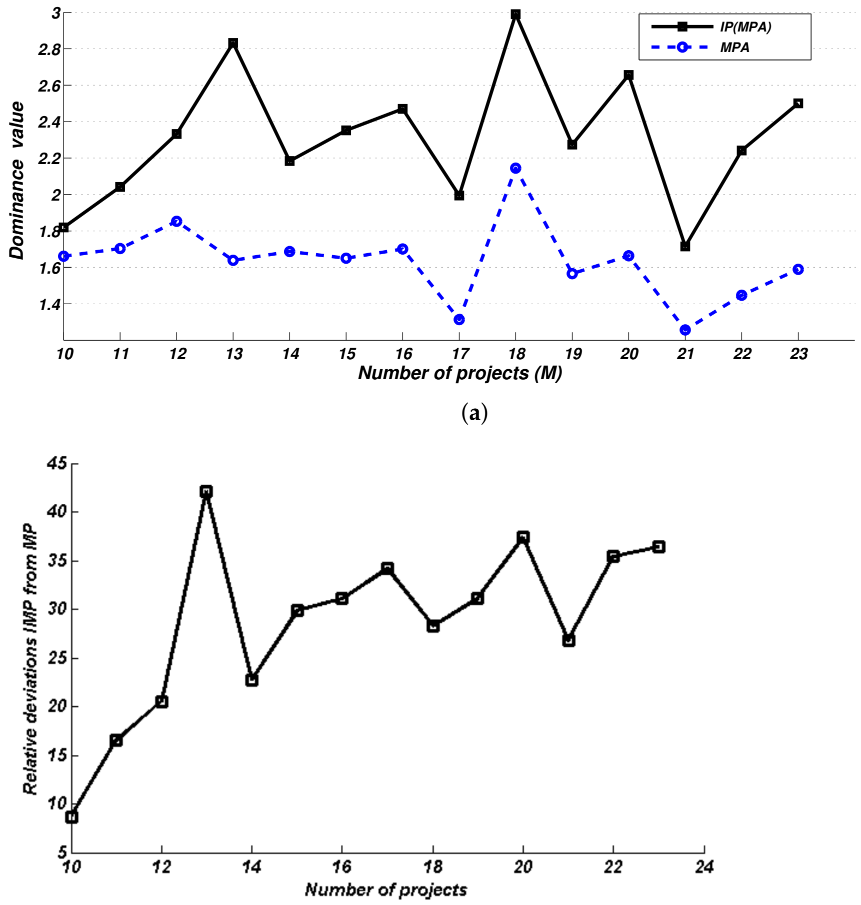
(**b**)

Figure 5. Comparing the performance of SPA and the MPA. (**a**) Dominance measure $\phi(.)$. (**b**) Relative deviation.

Looking at Figure 6a, the results indicate that for instances with 17 projects, for example, when the IPA is applied, the dominance value $\phi(.)$ associated with the MPA and the IP(MPA) solutions are 1.3130 and 1.993, respectively. In other words, the quality of solutions obtained from the IP(MPA) surpasses that of the MPA. In addition, we can conclude that when using a fixed scheduling scheme (in this case, the trdss), prioritizing projects leads to affecting the quality of schedules by solution approaches. In Figure 6b, we consider relative deviations (%) of the IP(MPA) from the MPA, calculated as follows:

$$\text{Relative deviation} = \left(\frac{\phi(MPA) - \phi(IP(MPA))}{\phi(MPA)}\right) \times 100.$$

It is observed that prioritizing projects and then prioritizing activities using the IP(MPA) is more effective than using the MPA for all instances (see Figure 6b). Regarding Figure 6, we can conclude that the efficiency of the IP(MPA) surpasses that of the MPA. This is highlighted as the size of the projects increases, meaning that the quality of solutions obtained from the IP(MPA) is superior to those from the MPA, and this difference in quality increases as the problem size grows.



(**a**)

(**b**)

Figure 6. Comparing the performance of MPA and the IP(MPA). (**a**) Dominance measure $\phi(.)$. (**b**) Relative deviation.

Furthermore, it is interesting to investigate whether the frequency of project integration in the RCMPSP leads to more efficient schedules concerning the objective function. To explore this, we compare the MPA (i.e., without project integration based on the TAO) to the proposed IP(MPA) while varying the number of integrated projects. Figure 7 shows relative deviations of the IP(MPA) from the MPA regarding the frequency of projects integration. When looking closer at Figure 7, numerical results demonstrate that when the frequency of projects integration in an instance is 93.3%, for example, the relative deviation of the IP(MPA) from the MPA is 8.7%. However, this value improves to 42.13% when the frequency of projects integration increases to 96.92%. In other words, this figure indicates that the frequency of projects integration can be affected positively by the performance scheduling scheme.

Observation 1. Based on numerical findings, prioritizing projects and frequency of projects integration, facilitated by the proposed IPA, positively impacts the quality of multi-project schedules.

Observation 2. It is important to note that the efficiency of solutions obtained from the IP(MPA), regardless of the problem size, is higher than that of other solution approaches.

Note that a statistical analysis at the significance level of $\alpha = 0.05$ reveals a significant difference between the IP(MPA) and the MPA.

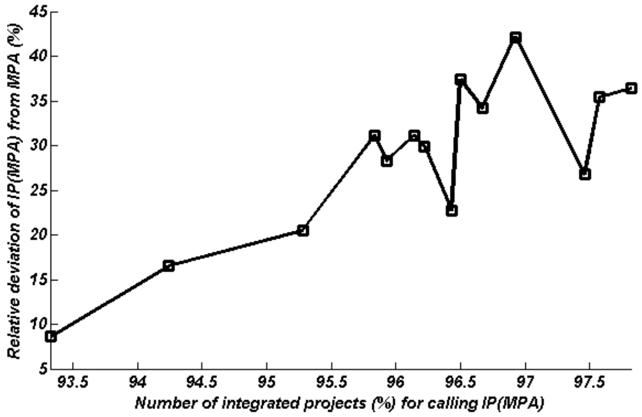


Figure 7. Relative deviation of IP(MPA) from the MPA regarding the number of integrated projects.

Finally, the results presented in Table 3 indicate that differences in computational times among the three mentioned approaches are negligible. In other words, there is no need to spend a remarkable amount of time to produce the solutions with higher quality. Interestingly, in some cases, the $IP(MPA)$ has less computing time compared to the SPA.

It is noteworthy that the average CPU time for prioritizing projects before invoking the IP(MPA) is approximately 0.11 s.

Table 3. Average relative deviation (%) of computational times from the minimum time among the three solution approaches for each instance.

Number Projects	10	11	12	13	14	15	16	17	18	19	20	21	22	23
SP	2.84	0.00	1.48	1.54	3.31	2.24	1.28	0.00	4.70	3.70	0.84	0.47	4.35	0.00
MP	2.56	0.042	1.11	0.00	0.00	0.00	0.00	0.69	0.00	0.00	0.00	0.00	0.00	1.81
IP(MPA)	0.00	0.508	0.00	0.13	2.83	1.67	2.06	2.38	0.79	0.36	2.79	1.76	3.22	4.26

Therefore, the experimental results demonstrate that prioritizing the projects improves the performance of the SPA and the MPA without incurring additional computational costs.

8. Conclusions

In this paper, we proposed a novel approach called the integrated project approach (IPA) for prioritizing projects prior to resource-constrained multi-project scheduling problems (RCMPSPs). The IPA integrates projects based on a resource-based measure known as the total amount of resource overflow (TAO) and employs a tri-directional schedule generation scheme (trdss) as an innovative multi-directional scheduling method.

The IPA builds upon multi-project approaches (MPAs), specifically single and multi-project approaches (referred to as SPA and MPA). However, it distinguishes itself by the way in which projects are integrated, relying on the TAO to facilitate this integration process. The IPA is implemented within the MPA framework referred to as IP(MPA).

To evaluate the performance of the IPA, we designed a project generator called GTAO, which generates multi-project problems while considering the TAO as a control parameter. In using the trdss for scheduling projects, the IPA prioritizes the projects in an RCMPSP based on their TAO values and positions and then integrates them accordingly.

Our results demonstrated that prioritizing projects and the frequency of project integration based on the TAO lead to more efficient schedules, particularly in minimizing project makespans. This highlights that, within a multi-project environment, effective project prioritization can significantly enhance the performance of scheduling generation schemes without incurring additional computational costs.

Furthermore, the application of artificial neural networks [40], particularly focused on topological structures for project ranking and scheduling in the context of resource mobility and uncertainty, presents a promising area for future research.

Author Contributions: Writing—original draft, H.R.Y.; Methodology, H.R.Y.; Software, H.R.Y.; Data curation, H.R.Y.; Project administration, H.R.Y.; Writing—review & editing, E.B.T. and F.K.; Validation, E.B.T. and F.K.; Formal analysis, E.B.T. and H.R.Y.; Resources, E.B.T.; Investigation, F.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The datasets used and/or analyzed during the current study are available from the corresponding author upon reasonable request.

Acknowledgments: Last, we would like to pay our gratitude and our respects to Hamed Reza Tareghian. After seriously helping to write, produce results, and organize this paper, Tareghian passed away in January of 2018. He was a dedicated professor in the Department of Applied Mathematics at Ferdowsi University of Mashhad, Iran.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

RCPSP	Resource-constrained project scheduling problem
RCMPSP	Resource-constrained multi-project scheduling problem
SPA	Single project approach
MPA	Multi-project approach
IPA	Integrated project approach
IP(MPA)	IPA equipped with MPA
trdss	Tri-directional schedule generation scheme
bidss	Bi-directional schedule generation scheme
TAO	Total amount of resource overflow
GTAO	RCMPSP generator based on TAO
RS	Resource strength
RF	Resource factor
DRCMPSP	Decentralized resource-constrained multi-project scheduling problem
ARLF	Average resource loading factor
AUF	Average utilization factor
PSBLIB	Kolisch benchmark

Notations

M	Total number of projects within an RCMPSP
P	A multi-project problem
p_i	Project i
N_i	Number of activities in project $i \quad i = 1, \ldots, M$
J_i	Set of activities associated with project i where $\lvert J_i \rvert = N_i$
$N = \sum_{i=1}^{M} N_i$	Overall total of activities across all projects
R	Set of all resource types
K	Total number of different resources where $\lvert R \rvert = K$
R_k	Total amount of renewable resource of type k where $k \in R$
r_{jik}	Amount of resource k required by activity j in project i
T	Total makespan for a multi-project scheduling problem
T_i	Makespan of project i
T_{si}	Makespan of project p_i using solution approach s
$MTAO = (m_{ij})_{M \times T}$	Resource overflows matrix with $T = \max_{i=1}^{M} LB_i$
LB_i	Length of critical path for project i
$Peak_k$	Maximum requirement for resource k over the scheduling period
$A(t)$	Set of activities that are active at time t
S	Total number of solution approaches, defined as $S = \{SPA, MPA, IPA\}$

F_s	Makespan of the last activity in a multi-project P using scheduling approach s
V_s	Vector of makespans for a multi-project P based on scheduling approach s
$\Phi(.)$	Measure of closeness to the reference point LB (dominance value)
σ_P	Deviation from the reference point for a multi-project P

References

1. Song, Y.; Wang, J.; Lu, J.; Si, X. Research on collaborative scheduling of multiple projects of prefabricated building based on the niche genetic-raccoon family optimization algorithm. *Alex. Eng. J.* **2023**, *64*, 1015–1033. [CrossRef]
2. Agrama, F.A.E.M. Linear projects scheduling using spreadsheets features. *Alex. Eng. J.* **2011** *50*, 179–185. [CrossRef]
3. Blazewicz, J.; Lenstra, J.; Rinnoy, K.A. Scheduling subject to resource constraints: Classification and complexity. *Discret. Appl. Math.* **1983**, *5*, 11–24. [CrossRef]
4. Katsavounis, S. Scheduling multiple concurrent projects using shared resources with allocation costs and technical constraints. In Proceedings of the Third IEEE International Information and Communication Technologies Theory to Applications, Damascus, Syria, 7–11 April 2008; pp. 1–6.
5. Kurtulus, I.; Davis, E.W. Multi project scheduling: Categorization of heuristic Rules performance. *Manag. Sci.* **1982**, *28*, 161–172. [CrossRef]
6. Lova, A.; ; Tormos, P. Analysis of scheduling schemes and heuristic rules performance in resource constrained multi-project scheduling. *Ann. Oper. Res.* **1982**, *102*, 263–286. [CrossRef]
7. Kurtulus, I.; Narula, S.C. Multi Project Scheduling: Analysis of Project Management. *Iie Trans.* **1985**, *17*, 58–65. [CrossRef]
8. Talbot, F.B. Resource-constrained project scheduling with time-resource tradeoffs: The non-preemptive case. *Manag. Sci.* **1982**, *28*, 1197–1210. [CrossRef]
9. Woodworth, B.M.; Willie, C.J. A heuristic algorithm for resource leveling in multi-project, multi-resource scheduling. *Decis. Sci.* **1975**, *6*, 525–540. [CrossRef]
10. Pérez, E.; Posada, M.; Lorenzana, A. Taking advantage of solving the resource constrained multi-project scheduling problems using multi-modal genetic algorithms. *Soft Comput.* **2016**, *20*, 1879–1896. [CrossRef]
11. Concalves, J.F.; Mendes, J.J.M.; Resende, M.G.C. *A Genetic Algorithm for the Resource Constrained Multi-Project Scheduling Problem*; AT& T Labs Technical Report TD-668LM4; AT& T Labs: Murray Hill, NJ, USA, 2004.
12. Wauters, T.; Verbeeck, K.; De Causmaecker, P.; Berghe, G.V. A learning-based optimization approach to multi-project scheduling. *J. Sched.* **2015**, *18*, 61–74. [CrossRef]
13. Sánchez, M.G.; Lalla-Ruiz, E.; Gil, A.F.; Castro, C.; Voß, S. Resource-constrained multi-project scheduling problem: A survey. *Eur. J. Oper. Res.* **2023**, *309*, 958–976. [CrossRef]
14. Hauder, V. A.; Beham, A.; Raggl, S.; Parragh, S.N.; Affenzeller, M. Resource-constrained multi-project scheduling with activity and time flexibility. *Comput. Ind. Eng.* **2020**, *150*, 106857. [CrossRef]
15. Liu, J.; Lu, M. Robust dual-level optimization framework for resource-constrained multiproject scheduling for a prefabrication facility in construction. *J. Comput. Civ. Eng.* **2019**, *33*, 04018067. [CrossRef]
16. Satic, U.; Jacko, P.C.; Kirkbride, C.; Performance evaluation of scheduling policies for the dynamic and stochastic resource-constrained multi-project scheduling problem. *Int. J. Prod. Res.* **2022**, *60*, 1411–1423. [CrossRef]
17. Davari Ardakani H.; Dehghani A. Multi-objective optimization of multi-mode resource-constrained project selection and scheduling problem considering resource leveling and time-varying resource usage. *Int. J. Supply Oper. Manag.* **2022**, *9*, 34–55.
18. Van Den Eeckhout, M.; Vanhoucke, M.; Maenhout, B. A column generation-based diving heuristic to solve the multi-project personnel staffing problem with calendar constraints and resource sharing. *Comput. Oper. Res.* **2021**, *128*, 105163. [CrossRef]
19. Fu, F.; Zhou, H. A combined multi-agent system for distributed multi-project scheduling problems. *Appl. Soft Comput.* **2021**, *107*, 107402. [CrossRef]
20. Van Eynde, R.; Vanhoucke, M. Resource-constrained multi-project scheduling: Benchmark datasets and decoupled scheduling. *J. Sched.* **2020**, *23*, 301–325. [CrossRef]
21. He, Y.; He, Z.; Wang, N. Tabu search and simulated annealing for resource-constrained multi-project scheduling to minimize maximal cash flow gap. *J. Ind. Manag. Optim.* **2021**, *17*, 2451–2474. [CrossRef]
22. Gómez, M.; Fernández, A.; Castro, C. Integrating a SMT solver based local search in ant colony optimization for solving RCMPSP. In Proceedings of the 2019 IEEE Latin American Conference on Computational Intelligence (la-CCI), Guayaquil, Ecuador, 11–15 November 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1–6.
23. Tran, D.H.; Cheng, M.Y.; Pham, A.D. Using Fuzzy Clustering Chaotic-based Differential Evolution to solve multiple resources leveling in the multiple projects scheduling problem. *Alex. Eng. J.* **2016**, *55*, 1541-1552. [CrossRef]
24. Zhu, H.; Lu, Z.; Lu, C.; Ren, Y. Modeling and algorithm for resource-constrained multi-project scheduling problem based on detection and rework. *Assem. Autom.* **2021**, *41*, 174–186. [CrossRef]
25. Li, F.; Xu, Z.; Li, H. A multi-agent based cooperative approach to decentralized multi-project scheduling and resource allocation. *Comput. Ind. Eng.* **2021**, *151*, 106961. [CrossRef]
26. Browning, T.R.; Yassine, A.A. Resource Constrained Multi project scheduling: Priority rule performance revisited. *Int. J. Prod. Econ.* **2010**, *126*, 212–228. [CrossRef]

27. Browning, T.R.; Yassine, A.A. Managing a Portfolio of Product Development Projects under Resource Constraints. *Decis. Sci.* **2016**, *47*, 333–372. [CrossRef]
28. Yassine, A.A.; Mostafa, O.; Browning, T.R. Scheduling Multiple, Resource-Constrained, Iterative, Product Development Projects with Genetic Algorithms. *Comput. Ind. Eng.* **2017**, *107*, 39–56. [CrossRef]
29. Homberger, J. A multi agent system for the decentralized resource constrained multi project scheduling problem. *Int. Trans. Oper. Res.* **2007**, *14*, 565–589. [CrossRef]
30. Homberger, J. A (μ, λ)-coordination mechanism for agent-based multi-project scheduling. *OR Spectrum* **2012**, *34*, 107–132. [CrossRef]
31. Available online: www.mpsplib.com (accessed on 5 February 2023).
32. Browning, T.R.; Yassine, A.A. A random generator of resource-constrained multi-project network problems. *J. Sched.* **2010**, *13*, 143–161. [CrossRef]
33. Chiu, H.N.; Tsai, D.M. A Comparison of Single project and Multi project Approaches in Resource Constrained Multi Project Scheduling Problem. *J. Chin. Inst. Ind. Eng.* **1993**, *10*, 171–179.
34. Kolisch, R.; Sprecher, A.; Drexl, A. Characterization and generation of a general class of resource constrained project scheduling problems. *Manag. Sci.* **1995**, *41*, 1693–1703. [CrossRef]
35. Demeulemeester, E.; Vanhoucke, M.; Herroelen, W. RanGen: A random network generator for activity on the node networks. *J. Sched.* **2003**, *6*, 17–38. [CrossRef]
36. Patterson, J.H. Project Scheduling: The effects of problem structure on heuristic scheduling. *Nav. Res. Logist.* **1976**, *23*, 5–123.
37. Yousefzadeh, H.R.; Tareghian, H.R.; Farahi, M.H. Multi-directional scheduling scheme in resource constrained project scheduling problem. *Nav. Res. Logist.* **2014**, *61*, 44–55. [CrossRef]
38. Kolisch, R. Library for Project Scheduling Problems. 2008. Available online: http://129.187.106.231/psplib/ (accessed on 5 February 2023).
39. He, Z.; Yen, G.G.; Zhang, J. Fuzzy-Based Pareto Optimality for Many-Objective Evolutionary Algorithms. *IEEE Trans. Evol. Comput.* **2014**, *18*, 269–285. [CrossRef]
40. Nikolić, V.; Milovančević, M.; Petković, D.; Jocić, D.; Savić, M. Parameters forecasting of laser welding by the artificial intelligence techniques. *Facta Univ. Ser. Mech. Eng.* **2018**, *16*, 193–201. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.