algorithms

MDPI

Article

Multi-Objective Resource-Constrained Scheduling in Large and Repetitive Construction Projects

Vasiliki Lazari, Athanasios Chassiakos * and Stylianos Karatzas

Department of Civil Engineering, University of Patras, 26500 Patras, Greece; civ6908@ac.upatras.gr (V.L.); stylianos.karatzas@upatras.gr (S.K.)

* Correspondence: a.chassiakos@upatras.gr

Abstract: Effective resource management constitutes a cornerstone of construction project success. This is a challenging combinatorial optimization problem with multiple and contradictory objectives whose complexity rises disproportionally with the project size and special characteristics (e.g., repetitive projects). While relevant work exists, there is still a need for thorough modeling of the practical implications of non-optimal decisions. This study proposes a multi-objective model, which can realistically represent the actual loss from not meeting the resource utilization priorities and constraints of a given project, including parameters that assess the cost of exceeding the daily resource availability, the cost of moving resources in and out of the worksite, and the cost of delaying the project completion. Optimization is performed using Genetic Algorithms, with problem setups organized in a spreadsheet format for enhanced readability and the solving is conducted via commercial software. A case study consisting of 16 repetitive projects, totaling 160 activities, tested under different objective and constraint scenarios is used to evaluate the algorithm effectiveness in different project management priorities. The main study conclusions emphasize the importance of conducting multiple analyses for effective decision-making, the increasing necessity for formal optimization as a project's size and complexity increase, and the significant support that formal optimization provides in customizing resource allocation decisions in construction projects.

Keywords: resource-constrained scheduling; resource allocation; resource leveling; multi-objective optimization; repetitive projects; Genetic Algorithms

check for updates

Citation: Lazari, V.; Chassiakos, A.; Karatzas, S. Multi-Objective Resource-Constrained Scheduling in Large and Repetitive Construction Projects. *Algorithms* **2024**, *17*, 351. https://doi.org/10.3390/a17080351

Academic Editor: Massimiliano Caramia

Received: 26 April 2024
Revised: 31 July 2024
Accepted: 7 August 2024
Published: 10 August 2024

1. Introduction

The resource-constrained scheduling problem (RCSP) has been thoroughly investigated over the past few decades and continues to be a topic of considerable interest for researchers due to its vital importance in project management. The purpose of project scheduling analysis is to create optimal time plans and resource selection and allocation strategies while still finishing the project on schedule. This is a classical combinatorial problem with application in several production-based processes, including in the construction industry.

While there have been plenty of research efforts in developing methods and algorithms for the typical RCSP, most of them have been evaluated with rather small and simple project structures. It is known, however, that construction projects are typically large (in terms of the number of activities) and complex (work interdependence), while they are often developed in a repetitive structure of identical sub-projects. In real-world environments, daily work planning is frequently required for a set of repetitive projects (i.e., construction projects that contain recurrent units). These units/sub-projects are made up of a common work structure. Three types of repetitive construction projects can be defined based on the direction of the workflow succession along the sub-projects:

- Horizontal/linear repetitive projects, which have a linear geometrical layout (e.g., highways, pipelines, and tunnels). In this case, sub-projects may be executed in parallel;

- Vertical repetitive projects, which involve the repetition of a sub-project network in a non-linear way (e.g., high-rise buildings). The sub-projects can be executed in a serial manner;
- Multiple repetitive projects, which consist of both horizontal and vertical repetitive processes (e.g., wide-area multistorey buildings).

The resource-constrained scheduling problem typically aims to smooth out resource usage over time and avoid resource overallocation throughout the project life cycle or during project phases. The benefits of such a process are multiple. First, there is a relatively constant resource usage, avoiding the need for continuous resource incoming and outcoming or resources being idle at the construction site. In terms of human resources, the condition of occasional working acts negatively in both their performance and morale. Further, if resource constraints exist (this is typical in construction work), the resource imbalance and variation in time lead to project delays and cost overruns. Finally, in multi-project cases, poor resource usage in one project substantially affects the performance of other projects in which the construction company is involved.

Resource leveling, with or without resource constraints, has been widely analyzed in the past. The majority of existing works use simple statistical criteria, such as the standard deviation of the resource usage values in time, to assess the resource allocation effectiveness. Although this is a useful criterion for some theoretical analysis, it holds two major deficiencies. First, it is not connected with some type of measure that can provide a practical outcome of the loss associated with the observed deviations from the optimal solution (e.g., cost). Most importantly, although it provides a measure of the resource allocation variability extent, it cannot provide a robust indication of the day-by-day resource allocation variability.

The focus of this work is to assess the potential of evolutionary-type optimization methods in obtaining near-optimal resource allocation schemes in large and repetitive projects from a practical point of view (i.e., in real-life applications). With this aim, two decision parameters with direct cost representation of the resource allocation imbalances are considered to provide an objective assessment of the resource allocation success in actual projects. These parameters are the cumulative overallocation of resource demand over the daily availability and the cumulative resource movements in and out of the project due to the varying demands from period to period. The above-real-cost parameters are supplemented by and compared to other no-direct-cost criteria that have been widely used in the resource allocation optimization literature (e.g., the standard deviation of the resource histogram values). Optimization is based on Genetic Algorithm principles and is performed via general-purpose commercial software. These decisions came from the facts that (a) GAs represent a typical and well-performing method of evolutionary computing (as indicated by existing research); and (b) the majority of existing commercial software performs GA optimization. Such a selection seems to be fundamental in real-life construction projects, in which case simple and easy-to-use tools are more likely to be employed by design and scheduling engineers.

This study addresses the resource-constrained scheduling problem of a large-size project consisting of 16 repetitive sub-projects with 10 activities each. A multi-objective optimization formulation is set considering four decision parameters in a weighted-sum formulation. These sub-objectives are analyzed both individually and collectively in several forms and are comparatively evaluated to provide performance indications as well as practical implications regarding the employment of the alternative parameters.

The rest of this paper is organized as follows. Section 2 presents the existing research efforts and results in optimizing resource-constrained scheduling. Section 3 presents the problem description and the proposed model formulation. Section 4 presents the case study development with the corresponding analyses and results. Finally, Section 5 provides a brief discussion of the obtained results and the main conclusions of the study.

2. Related Work

Several approaches and algorithms have been developed in the past to address the typical RCSP problem. These techniques can be divided into three categories: exact methods, heuristic methods, and meta-heuristic and evolutionary algorithms. In the first category, linear/integer programming (LP/IP) has been widely used. In this approach, mathematical relationships are built that linearly express the problem goals and limitations, and pertinent methods (e.g., the Simplex method) are used to develop a solution. Examples of works that solve scheduling problems using linear programming include those in [1,2].

As the project's size and complexity increase, these methods eventually prove to be ineffective for setting up the computerized problem structure, even though they can offer exact solutions. Since the solution space is often large to fully explore, especially as the problem size increases, heuristic techniques are used to first generate lists of feasible solutions and then search among them by bounding and pruning until a (near) optimal solution is obtained (branch and bound algorithms). The main drawbacks of such approaches are that the best solution is not always guaranteed and there is no single universal bounding algorithm that can be used to solve all problems. The work in [3] might be referred to as representative of such type of methods.

Metaheuristic techniques or evolutionary algorithms are used to provide near-optimal solutions when the precise solution to a problem is not known or is computationally demanding to determine. Evolutionary algorithms imitate the selection process in which, within the population set, members that least fit are eliminated while suitable members survive and multiply towards better solutions. In this direction, the studies [4,5] have employed Genetic Algorithms to explore near-optimal solutions in resource-constrained scheduling problems. Genetic Algorithms have also been applied to level daily resource utilization and minimize project makespan [6,7]. Other studies have created hybrid evolutionary algorithms or approaches by combining Genetic Algorithms with other evolutionary algorithms or techniques to produce better results [8]. The Ant Colony Optimization (ACO) algorithm [9], along with various hybrid ACO structures [10], has been employed to address the resource-constrained scheduling problem in both its classical formulation and its extended forms, such as those involving alternative execution modes. The employment of Harmony Search Algorithm (HS) to optimize the multi-mode resource-constrained project scheduling problem is reported in [11]. In a similar vein, the Particle Swarm Optimization (PSO) algorithm has also been utilized for optimizing resource-constrained scheduling problems, with an illustrative application detailed in [12]. The research in [13] uses the Artificial Bee Colony algorithm to solve problems from the PSPLIB library and compares it with several different algorithms from the literature such as GA, PSO, PSO +, ANGEL, and others. The research in [14] evaluates the performance of different types of bee algorithms (Bee Algorithm, Artificial Bee Colony, and Bee Swarm Optimization) for solving RCSP problems. The effectiveness of the Differential Evolution (DE) algorithm has been evaluated in [15]. The works in [16,17] address complex scheduling problems by developing hyper-heuristic algorithms based on Particle Swarm Optimization and Simulated Annealing, respectively. A hyper-heuristic algorithm is defined as a top-level heuristic algorithm that controls the application of other lower-level algorithms applied to the problem solution space. Finally, the concept of entropy-maximization for project resource-leveling has been introduced in [18] based on a minimum moment parameter of the resource time series.

The challenge of maintaining the continuity of various resources working on different tasks or sub-projects is a distinctive trait of repetitive projects. At the same time, the goal of a rather constant usage of the construction company's resources over time remains a top priority. When scheduling and allocating resources in large and repetitive projects, it is harder to visualize and carry out the process than in typical, small, or non-repetitive ones. The aim is still to reduce resource idle time, machinery movement in and out of the workplace, and staff hiring and firing situations. The development of an optimization model for scheduling such projects could result in significant time, resource consumption, and cost savings. The literature includes research efforts focusing on the resource scheduling problem in repetitive and multi-projects and in approaches targeting multiple optimization goals, mainly during the last ten years. The review results are codified in Table 1. The information that is presented includes the optimization sub-objectives, the solving methods, and the main characteristics of the case studies for method evaluation.

Table 1. The literature review outcomes.

No.	Authors	Year	Objective Function	Repetitive/ Multi-Projects	Multi-Objective	No. of Activities	Optimization Method
1	Roca et al. [19]	2008	minimize (1) project makespan, (2) variance of the sum of the resource usage over time	-	Yes	30, 60, 90, 120	NSGA-II
2	Kaiafa and Chassiakos [5]	2015	minimize (1) resource overallocation, (2) project deadline exceedance, (3) day-by-day resource fluctuations	-	Yes	10	Genetic Algorithms
3	Myszkowski et al. [10]	2015	minimize (1) project makespan, and (2) project cost	-	Yes	100, 200	Ant Colony Optimization
4	Beşikçi et al. [20]	2015	minimize total weighted tardiness of projects	M	No	120, 180	Genetic Algorithms
5	Mathew et al. [21]	2016	minimize (1) project duration, (2) project cost	R	Yes	20, 90	Genetic Algorithm
6	Eshraghi [15]	2016	minimize (1) project makespan, (2) resource capacity total cost	-	Yes	30, 60, 120	Differential Evolution Algorithm
7	Yassine et al. [22]	2017	minimize project duration	M	No	30, 60, 90, 120	Genetic Algorithms
8	Ei-Abbasy et al. [23]	2017	minimize (1) duration of the project group, (2) total cost, (3) financing cost, (4) maximum required credit, (5) resource fluctuation, and (6) peak resource demand	M	Yes	9, 25, 30	NSGA-II
9	Eid et al. [24]	2018	minimize (1) project duration, (2) project cost, (3) cumulative project interruptions, (4) project unit delivery delays	R	Yes	20, 75	Genetic Algorithms and Pareto Front sorting
10	Samuel and Mathew [25]	2018	minimize project duration	R	No	20	Genetic Algorithm
11	Salama and Moselhi [26]	2019	minimize (1) project duration, (2) project cost, (3) work interruptions	R	Yes	20	Linear scheduling method (LSM), critical chain project management (CCPM) and Genetic Algorithm (GA)
12	Nieves et al. [27]	2019	minimize project duration	R	No	84	Linear programming
13	Hariga et al. [2]	2019	minimize (1) resource utilization fluctuations, (2) activity splitting costs, (3) direct and indirect activity costs	-	Yes	10	Mixed-integer linear program (MILP)
14	Kannimuthu et al. [28]	2019	minimize (1) project duration, (2) cost, (3) maximize project quality	M	Yes	30, 50, 60, 90	Probabilistic Global Search Lausanne
15	Abido and Elazouni [29]	2021	minimize (1) resource fluctuations, (2) maximum value of the cumulative negative cash, maximize anticipated profit	-	Yes	18	Evolutionary programming (EP)
16	Yuan et al. [30]	2021	minimize (1) makespan, (2) operational cost	M	Yes	30, 60, 90	Hybrid Cooperative Co-evolution Algorithm (HCOEA)

Table 1. *Cont.*

No.	Authors	Year	Objective Function	Repetitive/Multi-Projects	Multi-Objective	No. of Activities	Optimization Method
17	Sharma and Trivedi [31]	2021	minimize (1) project completion time, (2) project completion cost, (3) project resource moment, and (4) maximize the project quality index	-	Yes	13	Opposition-based Non-dominated sorting Genetic Algorithm III (OBNSGA III)
18	Hegazy and Kamarah [11]	2022	minimize total project cost (activity direct cost, project indirect cost, penalties, incentives)	R	No	30, 70	Genetic Algorithm
19	Dai et al. [32]	2023	minimize cumulative absolute deviations between daily and average resource usage	R	No	7, 240	Two-stage GA-based scheduling algorithm
20	He at al. [33]	2023	minimize the maximal cash flow gap	M	No	10, 20, 40, 60	Tabu search
21	Bredael and Vanhoucke [34]	2024	minimize (1) total portfolio makespan, (2) average portfolio makespan, (3) average project delay, (4) average relative gap, (5) squared project delay, (6) maximum project delay, (7) maximum relative gap	M	No	360, 720, 1440	Genetic Algorithm

Summarizing the results of Table 1, existing methods have tackled several aspects of the RCSP, with a variety of decision parameters and solution methods. A main concept that is widely used in this direction is the resource fluctuation assessment. The parameter that is frequently used for such purpose is a statistical moment-type of the resource variability time series. It is argued below that such a parameter does not hold a realistic representation of the loss (cost) associated with high-variability resource utilization. Therefore, parameters that directly refer to the specificities of resource allocation failures need to be considered. A set of such parameters are proposed in the current study with proper justification and effectiveness evaluation. Previous research efforts typically examine small- to medium-sized projects (generally consisting of up to 30–60 tasks). Based on the literature search, 16 studies have analyzed case studies consisting of up to 60 activities, 6 studies up to 90 activities, 10 studies up to 120 activities, while there are 3 studies with up to 200 activities, and 1 with 1440 activities. The present study addresses the resource-constrained scheduling problem with decision parameters in a multi-objective structure that closely represents actual practice. The effectiveness of the alternative decision parameters, individually or collectively, is evaluated on the basis of a large-sized project consisting of 16 repetitive sub-projects and 160 activities in total. This study aims to integrate the scheduling of repetitive projects with a multi-objective optimization approach, addressing a gap in the literature where these two aspects are either underrepresented individually [35] or not studied extensively in combination. The different optimization models, which are developed in this study, are comparatively evaluated to provide performance indications as well as practical implications regarding the employment of the alternative parameters.

3. Proposed Model

In this section, the problem formulation and mathematical structure underpinning the proposed approach are presented, providing the foundation for subsequent analysis and discussion. Based on the task precedence relations and durations, the project structure is implemented in a spreadsheet environment. The start time of each task is associated with the finish time of all predecessors (using a max function) while the finish time is set based on the start time and the task duration. To allow for task re-scheduling and resource allocation improvement, a time lag is considered for each task between the latest predecessor finished and the actual start of the task under consideration. These time lags act as the independent variables of the optimization model. If these lags are constrained to obtain a non-negative value, all schedules developed are valid in terms of task precedence relations and the only additional constraint is for the project delivery by the given deadline. After developing a time schedule, based on task time lags, the number of the cumulative required resources per period is calculated based on the task scheduling and the resource needs of the individual tasks. The process of assigning new time lags is repeated within the iterative optimization process until a termination point is reached. The flowchart of the algorithm implementation is shown in Figure 1.

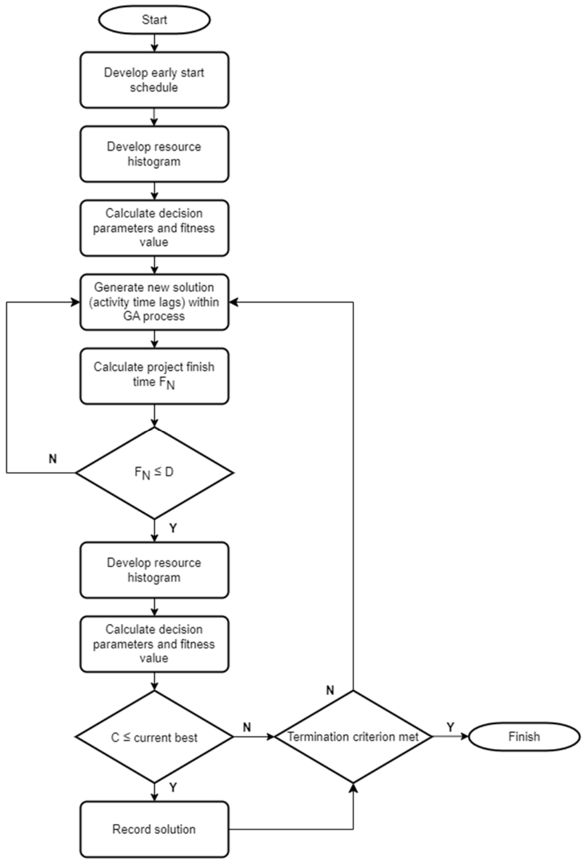

Figure 1. Algorithm implementation flowchart.

The mathematical formulation of the optimization model includes the following notation:

N: number of project tasks;
i: task number;
d_i: duration of task i;
s_i: start time of task i;
f_i: finish time of task i;
s_0: project start time (by assumption $s_0 = 0$);
F_N: project finish time ($F_N = max_{i=1}^{N} f_i$);
D: project duration deadline;
t: time frame ($t = 1, 2, \ldots, D$);
$x_i(t)$: binary variable equal to 1 if $s_i < t \leq f_i$, 0 otherwise;
r_i: number of resources of task i at time period t;
R_t: cumulative resource requirement at period t ($R_t = \sum_{i=1}^{N} r_i \cdot x_i(t)$);
R_m: mean resource usage;
C: total cost of the project.

The following decision variables are defined:

e_i: time shift of task i from predecessors.
The problem constraints are the following:
Task Scheduling Constraints:

$$s_i = max_i f_i + e_i, \forall i, \forall\ predecessor\ j\ of\ task\ i, \tag{1}$$

$$f_i = s_i + d_i,\ \forall i \tag{2}$$

$$e_i\ \geq\ 0,\ \forall i \tag{3}$$

Project Completion Constraints:

$$F_N\ \leq\ D, \tag{4}$$

In the present work, several decision parameters are used, which can be more tangibly associated with the real effects of resource imbalance or constraints in resource availability [36]. These include the following:

- RIO—Resources in and out: this accounts for the sum of resource unit deviations from day to day and represents the cost of moving resources (human and machinery) in and out of the construction site day by day.

$$RIO = \sum_{t=1}^{D+1} |R_t - R_{t-1}|,\ R_0\ =\ R_{D+1}\ =\ 0, \tag{5}$$

- RLE—Resource limit excess: this measures the cumulative number of resources exceeding the daily resource availability and accounts for the financial impact of recruiting more additional resources than initially planned.

$$RLE = \sum_{t=1}^{D} (R_t - R_m),\ \forall\ t \text{ at which } R_t\ >\ R_m, \tag{6}$$

- MaxR—Maximum resource demand: this shows the maximum resource value in the resource histogram indicating the cost from attempting to satisfy the full resource requirements.

$$MaxR\ =\ max_{t=1}^{D} R_t, \tag{7}$$

- STD—Standard deviation: the standard deviation of the resource histogram values in time is considered as well for the sake of comparison.

$$STD\ =\ \sqrt{\frac{\sum_{t=1}^{D} (R_t - R_m)^2}{F_N - 1}}, \tag{8}$$

The above parameters (decision criteria) are incorporated in a weighted-sum objective function of the following form:

$$minC = w_1 \cdot STD + w_2 \cdot RLE + w_3 \cdot MaxR + w_4 \cdot RIO, \tag{9}$$

where w_i accounts for the corresponding unit cost values, defined by the user and representing the problem characteristics. With such a formulation, the distinct criteria can be evaluated both individually and collectively. The weights in the above formulation indicate the importance of each parameter in any specific case. Regarding the physical meaning of the weights, RLE and RIO account for parameters that can be directly valuated in cost terms of the corresponding resource allocation inefficiencies, thus, the related weights hold unit cost values. MaxR and STD are generic parameters that are often used to smooth out temporal deviations of such histograms. For these parameters, the weights are appropriately normalized to provide the desired level of influence in a bi- or multi-objective assessment.

In real-life (construction) projects, scheduling and resource utilization decisions are most probably assisted by simple formulations and handy tools that can be easily used by design engineers. With this aim, the RCSP problem is developed in an Ms-Excel 365 spreadsheet structure to assist development, readability, model performance checking, and result evaluation. The employment of existing commercial software for optimization significantly alleviates the reluctance of the engineer to dig in and develop an optimization algorithm or implement one from the literature. While general-purpose software may not be as effective as a fine-tuned algorithm for a specific problem, it is still a valuable tool to significantly improve resource allocation, especially in large projects. Genetic Algorithms have been found to be effective in resource-constrained scheduling problems, presenting a stable growth since 2010 and being the most used in the last five years [37]. Further, commercial evolutionary algorithm-related optimization software has principally incorporated this type of algorithm. Optimization in this study is performed via a commercial optimization software (Palisade (now named Lumivero) Evolver, v8.6.0, https://lumivero.com/products/decision-tools/evolver/, accessed on 1 August 2024), which runs as an add-in of Ms-Excel. The software allows different parametrization of the GA structure. As part of this analysis, it was found that, although this parametrization has not displayed much impact on the solution quality, a formulation with 50-chromosome population size and crossover and mutation rates of 0.5 and 0.1, respectively, seems to provide somewhat improved results. The problem runs have been performed on a typical PC with the following specifications: Intel® Core (TM) i3-3110M CPU @ 2.4 GHz, RAM 4GB.

4. Case Study

The case study includes a 160-activity project which consists of 16 repetitive identical basic projects of 10 activities each. The whole project is formed by considering four basic projects in a serial execution form, which are then scaled up in a parallel quadruple execution arrangement, as shown in Figure 2. The resulting project can be considered a typical repetitive and large project.

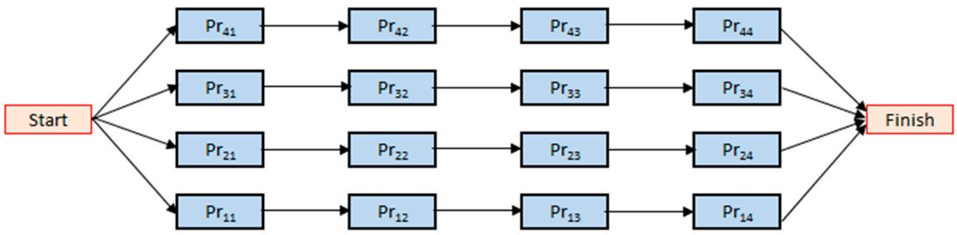

Figure 2. Network diagram structure of the examined project.

The input data of the basic 10-activity project are shown in Table 2, while its network diagram is illustrated in Figure 3 ([38]). The project has been designed to include diverse features of a (construction) project (e.g., activities that can take place in parallel or be executed alone (in the case of task J), time floats of different sizes, etc.). The number of resources for all activities has been assumed at two units per day. Even though in practice, several types of resources are used in (construction) projects, in this case study, a single resource type is considered for all activities. In general, if there are two or more resource types, the effectiveness in resource allocation may be hindered by the conflicting requirements of different resource types of project activities. This means that a schedule that leads to a nice resource histogram of one resource type may not provide a desirable allocation of another resource type. As a result, the global solution may already present some resource allocation inefficiencies and, as such, the capability of the optimizer cannot be objectively assessed. Using a single resource type, the optimal resource pattern may be developed beforehand, especially in small projects. Therefore, the net outcome of the optimization can be assessed. Further, the solution space for a particular project is the same (it includes all feasible alternative schedules), irrespective of the number of resource types. Thus, the problem size remains the same with the exception that some additional calculations for the added resource types are required in each optimization step.

Table 2. Project data for the applied example.

Activity	Predecessors	Duration	Resources
A	Start	5	2
B	Start	10	2
C	Start	4	2
D	A	7	2
E	C	5	2
F	A	4	2
G	B, D, E	3	2
H	C	6	2
I	Start	4	2
J	F, G, H, I	2	2

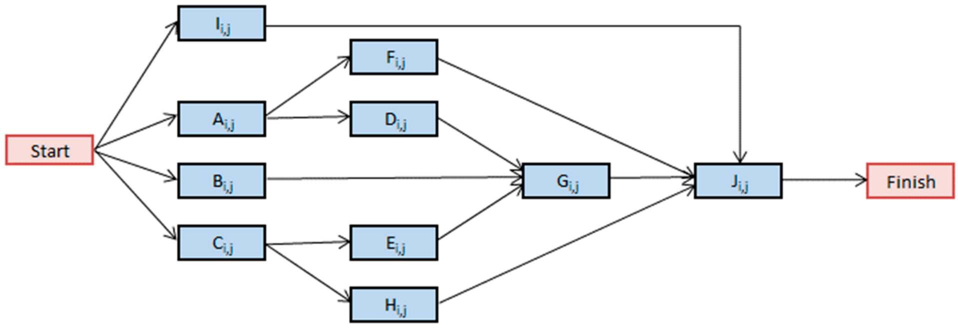

Figure 3. Network diagram of the basic project.

The activity durations and resources have been set in a way that facilitates the development of a perfectly flat resource histogram in certain cases of the following analyses, and this is the reason for assuming uniform resource usage in all activities. Prior knowledge of the exact solution is important in objectively evaluating the actual convergence potential of the optimization process.

With the input data of the case study example, the minimum duration of the basic project is 17 days, while for the full project, it is 68 days. The resource allocation histogram of the project, in its default early start scheduling form, presents significant peaks and valleys in resource demand throughout the project length (Figure 4). The repetitive resource allocation pattern in the diagram results from the serial execution of sub-projects. The high resource demand at specific periods may be difficult to serve, while the high resource fluctuations at specific time points require expenditures for transferring resources in and out of the construction site.

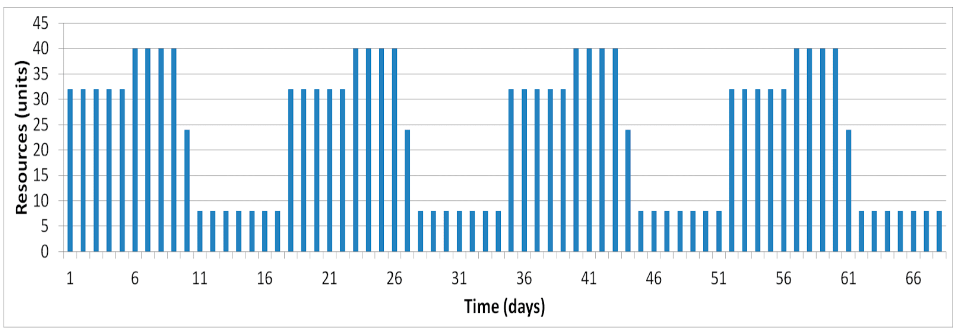

Figure 4. Resource histogram for the early start project schedule.

A typical optimization run develops a resource allocation plan, such as the one shown in Figure 5. The top part presents the resource requirements of the full project, while the following diagrams show the resource allocation of the series of sequential sub-projects.

The different colors indicate the contribution of each basic project within the sequential structure. The red line in the cumulative diagram indicates the daily resource availability, which corresponds to the mean resource requirement and indicates the ideal scenario of a perfect resource allocation scheme. The diagrams reveal that diverse (non-repetitive) resource patterns are built for each sub-project independently from others in both directions of the project scaling-up. This observation indicates that the whole project is treated as a rather "random" structure. It is also shown that the sub-projects in each series do not interfere with each other, following the serial deployment setting and constraints. In terms of resource allocation effectiveness, a highly leveled resource diagram has been developed around the average of 16 resources per day for a 100-day duration.

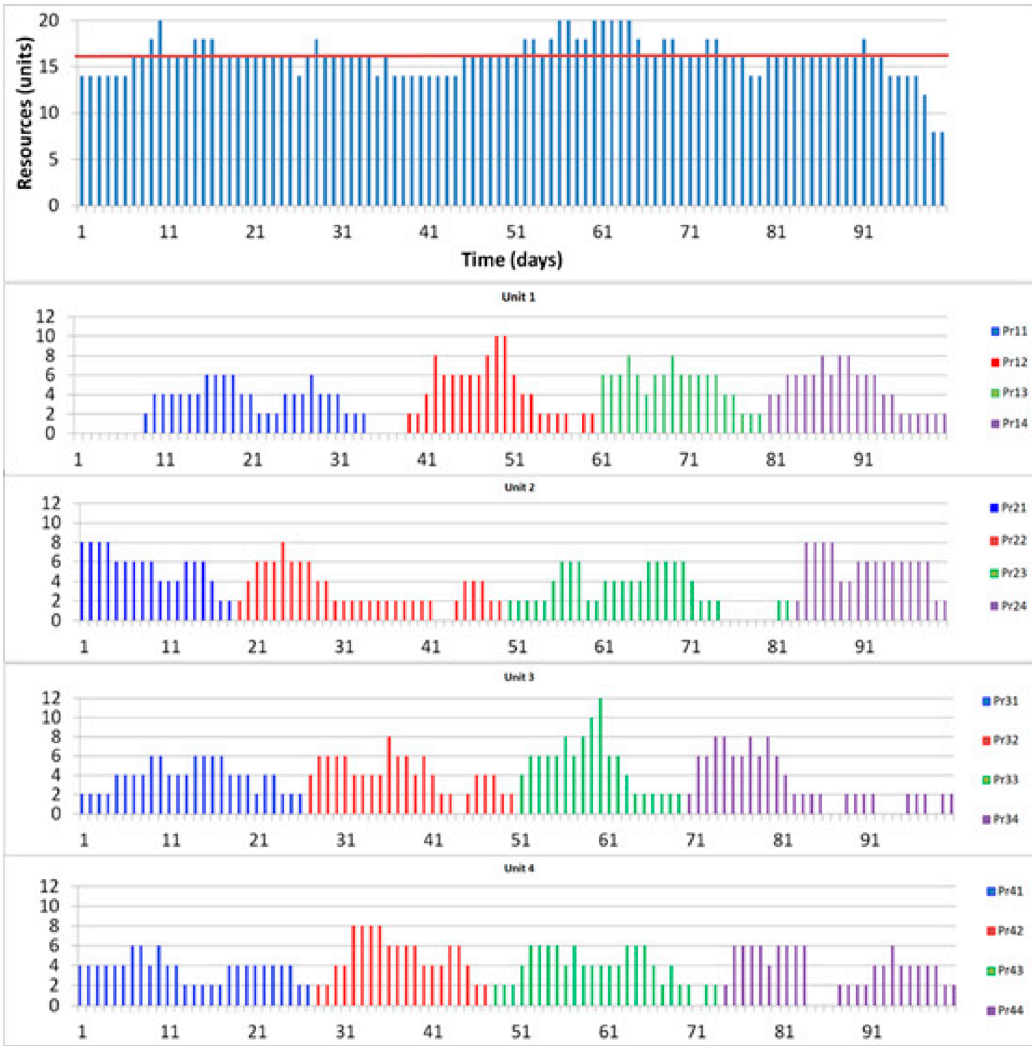

Figure 5. Typical optimized resource allocation diagram. The red line indicates the daily resource availability.

As stated above, the analysis considers alternative optimization criteria either individually or in combination. Initially, each decision criterion is optimized individually. This is conducted to explore the full performance potential of each criterion without any interference with other possible objectives. Next, optimization is performed with pairs of criteria, especially those that are not linked with a linear relation. Finally, two-step optimization is examined, based on distinct criteria, to explore any potential improvements. Alternate constraint levels for resource availability and project completion time are considered to provide a comprehensive assessment of the process output and effectiveness.

Figures 6–12 indicatively present the optimal resource allocation histograms associated with a project duration of 160 days to obtain a pictorial form of the solution quality. The red line in each diagram indicates the daily resource availability. It is observed that the decision criteria STD, RLE, and MaxR end up as quite leveled histograms as they attempt to spread out resources across the whole project makespan and in numbers that are close to the resource availability level (Figure 6, Figure 7, and Figure 8, respectively). Among them, the STD and RLE criteria provide comparable resource histograms (in terms of resource usage fluctuations) and appear to perform better than the MaxR criterion in lessening such fluctuations. In a different direction, the RIO criterion results in resource allocation patterns that avoid high day-by-day resource usage alterations. To minimize the cumulative deviation of such kind, the resulting schedule typically does not fill the whole project length availability (Figure 9). Inevitably, the cumulative resource units exceeding the preset resource availability threshold (i.e., RLE value) increase in comparison to the previous criteria. However, the resource allocation histogram presents fewer and smoother resource transitions from day to day. Figures 10 and 11 illustrate representative resource histograms when bi-objective optimization of two criteria is applied, namely STD or RLE with RIO, respectively. As expected, the obtained solutions lie somewhere in between those provided by the corresponding single-criterion considerations. The importance of this analysis lies in its capacity to consider conflicting criteria with their corresponding actual costs to provide a solution that minimizes the total cost. Finally, the potential effectiveness of a two-step sequential optimization process incorporating different criteria is examined. In this example, the RLE or the STD criterion is first employed, leading to the corresponding best possible solution. Starting from this solution, a second optimization is performed using the RIO criterion. Figure 12 provides an indicative output of the sequential optimization of the STD and RIO criterion. Further discussion of the results is provided next.

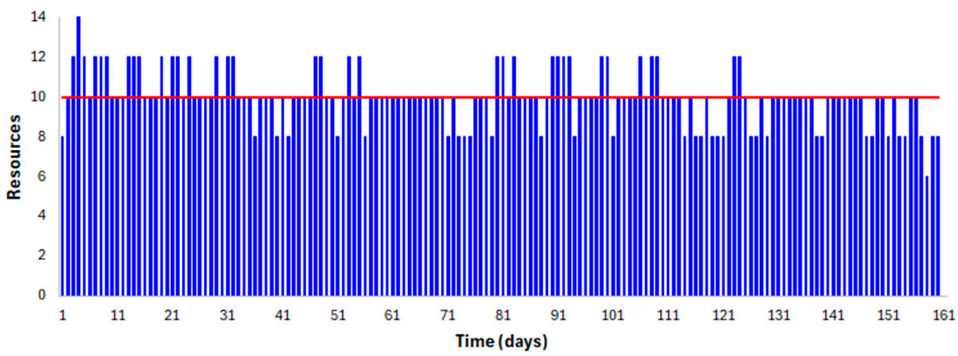

Figure 6. Resource histogram for STD criterion. The red line indicates the daily resource availability.

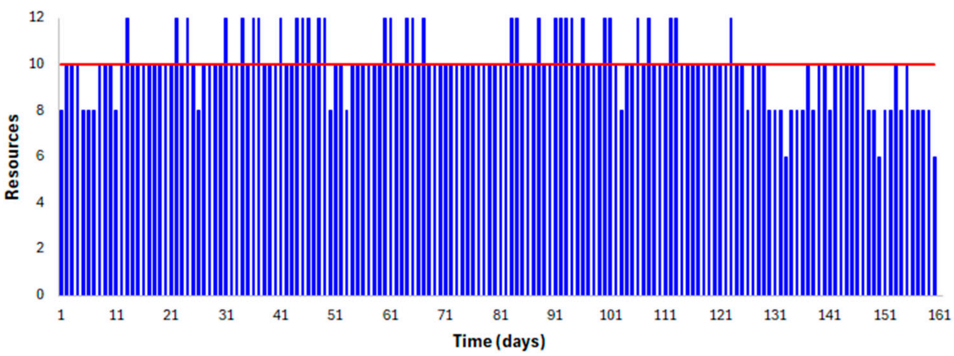

Figure 7. Resource histogram for RLE criterion. The red line indicates the daily resource availability.

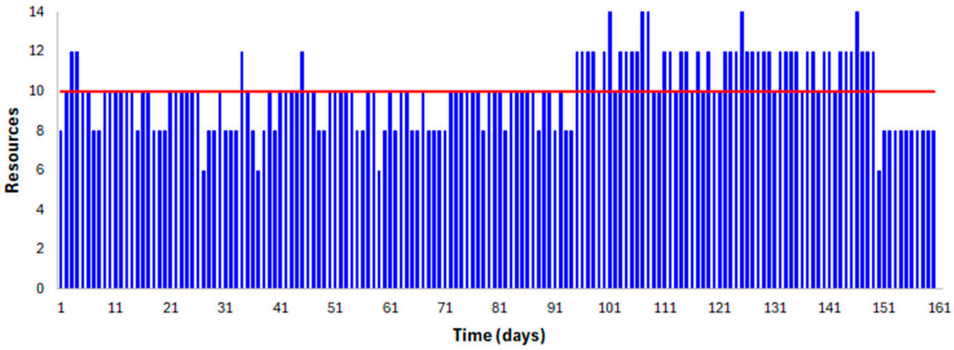

Figure 8. Resource histogram for MaxR criterion. The red line indicates the daily resource availability.

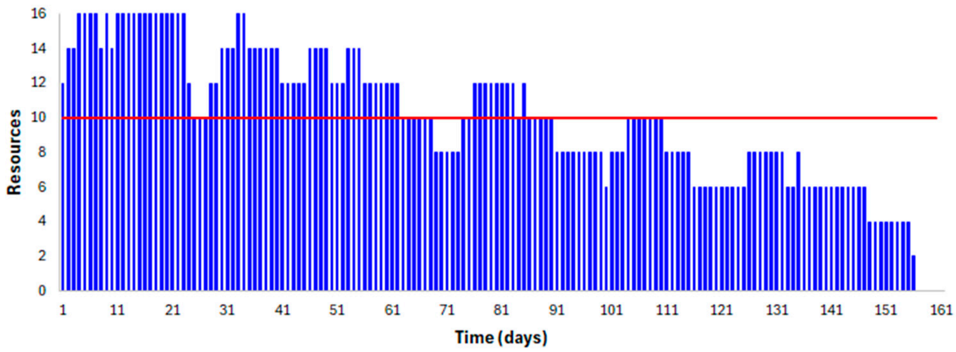

Figure 9. Resource histogram for RIO criterion. The red line indicates the daily resource availability.

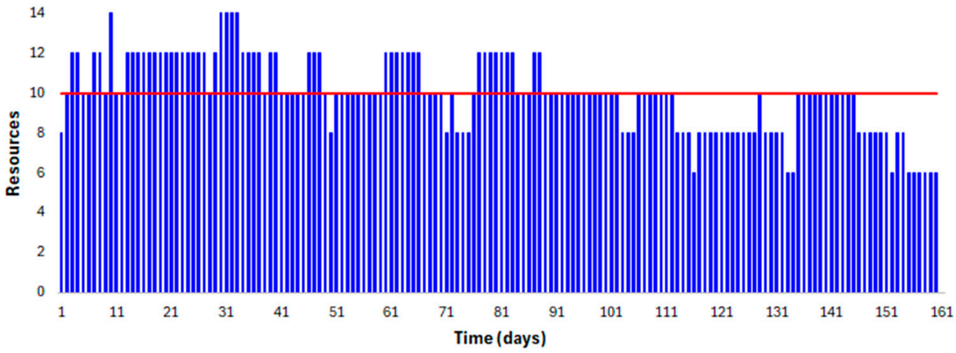

Figure 10. Resource histogram for STD-RIO criterion. The red line indicates the daily resource availability.

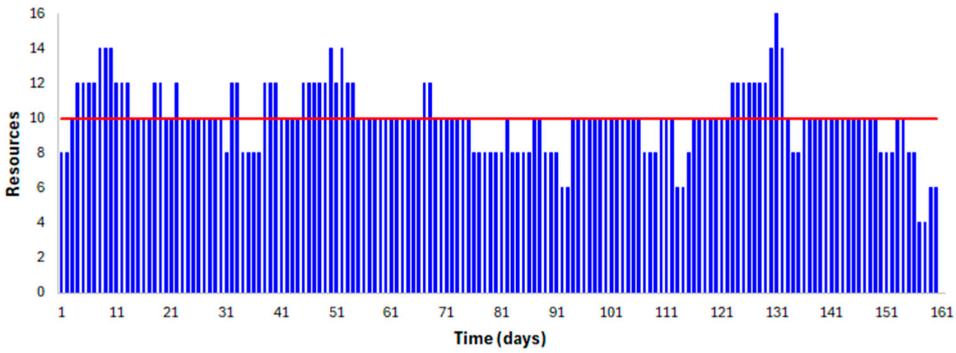

Figure 11. Resource histogram for RLE-RIO criterion. The red line indicates the daily resource availability.

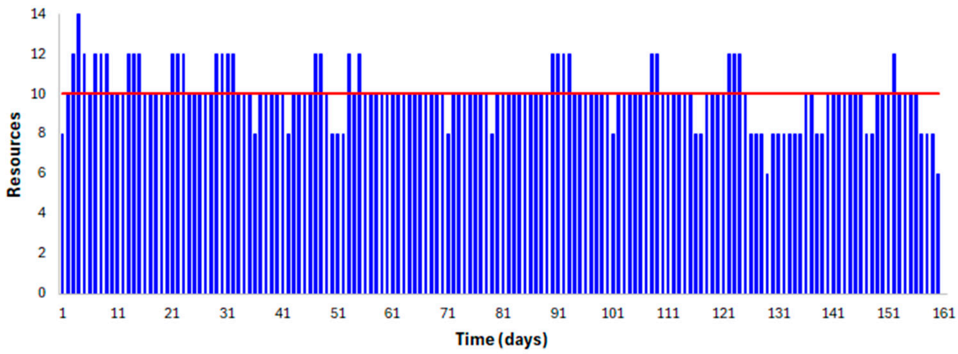

Figure 12. Resource histogram following a 2-step optimization process (STD-RIO). The red line indicates the daily resource availability.

Table 3 presents the input and output values of the analyzed cases. Case 0 refers to the initial (non-optimized) project schedule corresponding to the early start times. This schedule acts as the initial optimization solution in all cases examined. Case 1 represents the solution with the minimum standard deviation (STD) of the daily resource usage as a single objective. Similarly, Case 2 provides the result of minimizing the cost of exceeding the daily resource availability (RLE). In the same path, Case 3 indicates the solution of minimizing the maximum resource demand (MaxR) that is observed during the project. The last single-objective Case 4 examines the minimization of resource movements in and out of the project throughout its length (RIO). The next two cases present the results of bi-objective optimization; specifically, Case 5 considers the criteria STD and RIO, and Case 6 the criteria RLE and RIO. Finally, Cases 7 and 8 incorporate two-step sequential optimizations between criteria STD and RIO and RLE and RIO, respectively, meaning that an optimization run is performed with respect to the first criterion and, upon the optimum solution found, another optimization is run with respect to the second criterion.

The rationale behind such development is as follows. The problem under consideration aims to provide real-life solutions in regard to actual construction (or similar) projects. In this direction, there are two main cost contributions, one from exceeding the resource availability and the other related to the need for transferring resources in and out of the project. In extreme cases, only one of these cost elements may exist. For instance, there may be the case that it is costly to recruit additional resources (e.g., use of special machinery) but the cost of moving such resources is negligible (e.g., sub-projects being next to each other). In such a case, a rather single-objective project is encountered, using either of the three parameters, STD, RLE, or MaxR, which serve this objective (Cases 1, 2, and 3). On the other hand, there may be cases in which additional resources can be easily allocated to the project with no extra cost for overallocation (this mostly concerns human resources of an ordinary specialty). In addition, the cost in and out of the project may be high, for instance, if the sub-projects are dispersedly located. In the general case, both objectives provide some contribution to the resource allocation cost so that optimization of both objectives, in the form of a cost–sum approach, is sought based on their relative unit cost to each other (Cases 5 and 6). The last two cases (7 and 8) aim to investigate whether the optimization of the two conflicting objectives in a serial mode of respective single-objective optimizations can provide better results compared to the concurrent optimization of the two parameters.

The analysis has been extended to several levels of project durations and corresponding resource thresholds. In each case, the RCSP has been considered, employing individual criterion or pairs of criteria and optimization modes (single- or two-step optimization). In each run, all parameter values that correspond to the final solution are recorded. Each optimization case is run three times to account for the expected variability in results due to the stochastic nature of the optimization (GA) algorithm. Among the three outputs, the best one is registered. This is an anticipated course of action in practical applications of such a problem. In a more research-oriented effort, the typical course of action is to perform several trials and record the statistics of all runs (average or median, standard deviation, confidence intervals, etc.). Some indicative results of such kind are presented further below. The numbers in bold in Table 3 indicate the absolute best values of the examined parameters among all optimization tests in each project length/resource availability case.

Table 3. Optimal results for example project.

		Optimization Criteria								
		Case 0	**Case 1**	**Case 2**	**Case 3**	**Case 4**	**Case 5**	**Case 6**	**Case 7**	**Case 8**
		Initial Solution	**STD**	**RLE**	**MaxR**	**RIO**	**Multi-Objective STD-RIO**	**Multi-Objective RLE-RIO**	**Sequential STD-RIO**	**Sequential RLE-RIO**
Resource constraint 6 (D = 267)	F_N	68	267	267	267	231	267	267	267	267
	STD	13.67	**1.18**	1.59	1.26	4.07	1.78	2.14	1.26	1.95
	RLE	1192	**90**	130	100	534	170	194	102	172
	MaxR	40	**8**	10	**8**	16	10	14	**8**	12
	RIO	272	228	258	230	**108**	130	136	164	170
Resource constraint 8 (D = 200)	F_N	68	200	200	200	195	200	200	200	200
	STD	13.67	1.7	1.9	**1.37**	4.76	1.91	2.1	2.04	1.99
	RLE	1056	120	104	**92**	400	136	140	146	134
	MaxR	40	12	14	**10**	18	12	14	14	12
	RIO	272	192	214	174	116	**114**	120	122	134
Resource constraint 10 (D = 160)	F_N	68	160	160	160	156	160	160	160	160
	STD	13.67	1.36	1.35	1.74	3.60	1.88	1.89	**1.31**	2.39
	RLE	976	70	66	102	256	110	98	**62**	124
	MaxR	40	14	**12**	14	16	14	16	14	16
	RIO	272	172	154	176	**88**	98	114	110	104
Resource constraint 12 (D = 134)	F_N	68	134	134	134	134	134	133	134	134
	STD	13.67	1.55	1.39	**1.36**	6.27	2.3	3.48	2.11	2.31
	RLE	896	56	**48**	50	368	106	152	96	96
	MaxR	40	**16**	**16**	**16**	24	18	20	**16**	18
	RIO	272	160	140	148	**70**	96	102	104	112
Resource constraint 16 (D = 100)	F_N	68	100	100	100	100	100	100	100	100
	STD	13.67	1.97	**1.86**	2.05	3.85	2.4	2.2	2.31	2.22
	RLE	736	62	**52**	66	158	86	62	70	66
	MaxR	40	**20**	**20**	**20**	24	**20**	**20**	**20**	22
	RIO	272	168	120	148	**78**	100	84	98	100

Notation. D: project duration deadline, $F_{N:}$ project finish time, STD: standard deviation of daily resource requirements, RLE: resource limit excess (overallocation), MaxR: maximum resource usage, RIO: number of resources in and out of project.

The results in Table 3 show that all criteria end up with considerable resource allocation improvement in comparison to the initial settings. All criteria but RIO eventually converge to the given project duration deadline. An overview of the table results shows the following:

- Criteria STD, RLE, and MaxR aim mainly at resource leveling and overallocation reduction. The RLE and STD provide comparable outputs; in some cases, interestingly, a slightly better value of RLE is obtained when using the STD criterion rather than the RLE itself (and vice versa). Although the results are comparable, the main advantage of the RLE output is that it can be easily valuated in pure cost terms. Instead, the STD value does not have a direct cost representation of the resource allocation inefficiency. The MaxR criterion provides a coarser picture of the optimization output as it accounts only for the peak resource deviation without considering the distribution of individual resources in detail. However, this criterion occasionally ends up with the best RLE and/or STD outputs among other criteria;
- The RLE criterion appears to be more representative of the financial loss than the MaxR since it accounts for any single-resource excess above the availability threshold over the project duration. Instead, the MaxR criterion indicates the maximum excess

but not the cumulative excess along the project. In addition, because the number of resources appears in integer form, several discrete solutions present the same MaxR value (but different RLEs), making it difficult to differentiate among them in terms of detailed effectiveness;

- There is no single criterion among these three that clearly outperforms the other two in all or most cases. The results in Table 3 show that a parameter can obtain a better value when another objective is pursued. This means that it is rather useful to experiment with all three criteria within the analysis process;
- The RIO criterion is not closely aligned with the goals of resource diagram smoothness and resource overallocation minimization. It rather pursues to develop resource allocation patterns with few and low transitions in resource usage from day to day. As a result, it generally permits higher resource overallocation but smoother transitions from level to level in time. The solutions are usually associated with reduced project duration and higher average resource loading. In this case, some extra cost savings can be realized in terms of indirect project costs from duration reduction (not considered in the analysis). Similarly to the RLE criterion, the RIO can be practically valuated in pure cost terms if the unit cost of the resource (primarily machinery type) and movement in and out of the project is known;
- The employment of multi-objective or sequential optimization appears to provide solutions that are often of higher quality than single-objective optimization. In every optimization case, all four parameters associated with the final solution are recorded so that the real-life cost of this solution can be calculated;
- The employment of multiple runs with the same criterion (or criteria) appears also to be beneficial as the stochastic nature of the solution method leads to distinct solutions among alternative runs;
- A useful tip for performing optimization when the objective function takes discrete values (this pertains to RLE, RIO, and mostly MaxR, which obtain integer values within a short range) is the following. The evolutionary process generates and evaluates new solutions as the process goes on. If these new solutions fail to push the fitness value to its next discrete level, the solution population is not promptly and adequately refreshed to effectively search the decision space. To avoid this trapping, one can add a small perturbation in the objective function to let the optimization engine be continuously active. In the present case, a very small STD component (appropriately regulating the weight w4) and not practically affecting the main optimization criterion is added to the objective function to provide such kind of service. In fact, the majority of the results presented in Table 3 (best values in each case) have been obtained with such a formulation. The above augmentation is not useful in cases where the optimization parameter already calculates continuous values (e.g., STD).

To obtain a more global picture of the algorithm's effectiveness in achieving the desired outcome, the best parameter values from Table 3 are presented in relative terms in Figure 13. In particular, the RLE and RIO values are related to the total number of resources used throughout the project while the MaxR and STD values reflect deviations fromthe average resource level in each case. The results indicate that the total number of resources in excess of resource availability (RLE) and the total number of the resource movements in and out of the project (RIO) are in the order of 3% and 7.5% in relation to the cumulative number of resources used throughout the project. The standard deviation of the resource usage is between 10% and 20% in relation to the corresponding mean value. Finally, the maximum resource usage overruns the mean value by 25% to 33%. All deviations tend to increase as the project length increases and the mean resource usage decreases. This may be expected as the alternative schedule formulations progressively become limited towards this end.

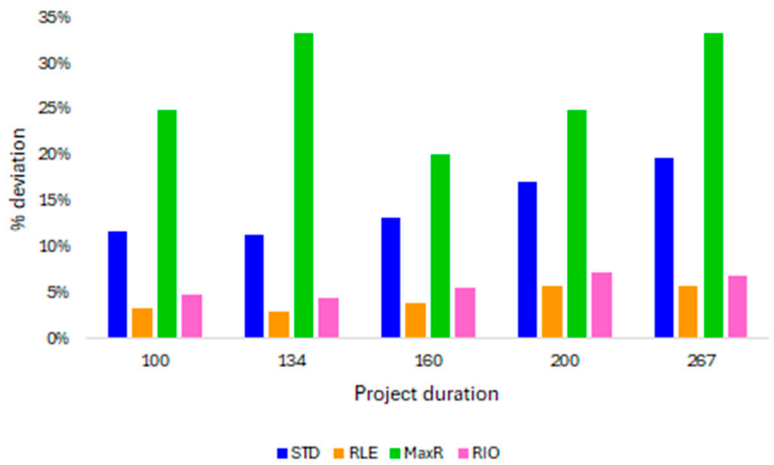

Figure 13. Percent deviation of decision parameter values from ideal solution.

To examine the degree of correlation among the alternate decision parameters in more detail, indicative solutions of all runs are considered and graphically illustrated in Figures 14 and 15. The results of the analysis indicate a high correlation between the STD and RLE criteria. The diagram of Figure 14 illustrates a strong linearity between the two criteria, even in the case that the optimization is based on the RIO criterion. In the latter case, the only difference is that both STD and RLE obtain quite higher but still proportional values in comparison to their own optimization outcome. On the other hand, the correlation between RIO and RLE (similar to STD) does not result in an observable pattern (Figure 15). Instead, it appears that these pairs of criteria create a point cloud with low correlation.

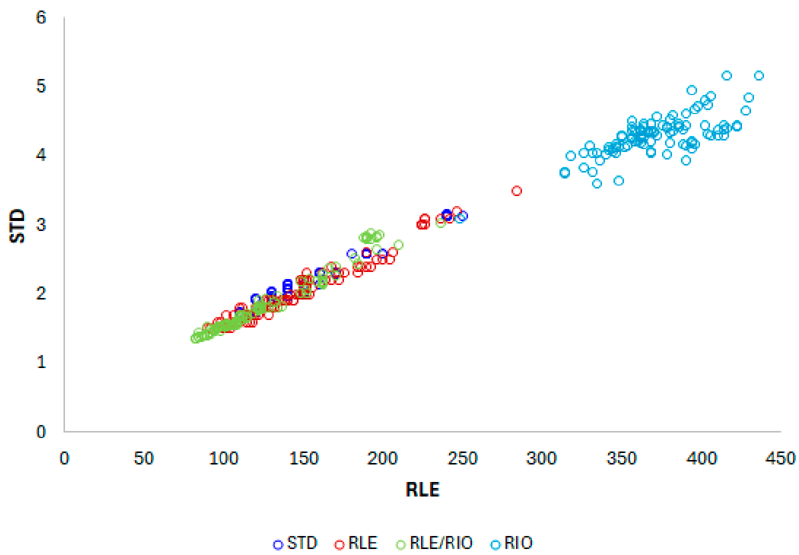

Figure 14. RLE and STD correlation diagram.

As discussed above, the RLE and RIO criteria provide diverging results that appear to lead to somewhat conflicting goals. As a result, trade-off solutions may be obtained depending on the weights that are used in the fitness function (9). Figure 16 presents several solutions that have been obtained with experimentations with different weights and combination runs for the case of a 160-day project. These outputs form a rather scattered trade-off point cloud without, however, a well-developed Pareto front. This result indicates a fair level of randomness of the obtained solutions and, therefore, the need to try different criteria and run combinations.

Figure 15. RLE and RIO correlation diagram.

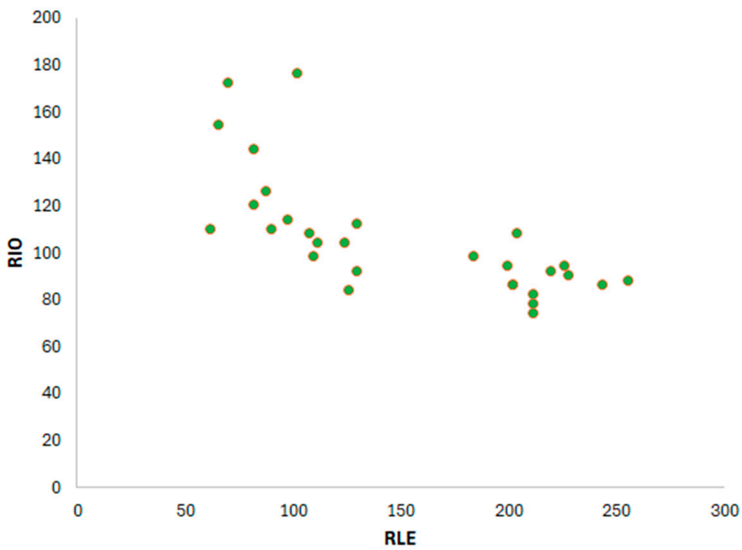

Figure 16. Point cloud of alternative RLE and RIO solutions.

This observation raises another inquiry need. Most of the results in Table 3 have been obtained following three runs of each case and the insertion of the best result among the runs. This can be considered adequate from a practical point of view. In a more theoretical analysis though, a larger number of optimization repetitions would be required with the corresponding statistics (mean, median, and standard deviation) being recorded. This analysis has been indicatively performed for the 160-day project duration case and the criteria RLE and STD. Based on eleven runs, the mean value and the standard deviation of RLE were 79 and 13.5, respectively (the best RLE value in Table 3 is 66 in single-criterion optimization and 62 in two-step optimization). This example shows that the possibility of reaching an extremely good solution with few runs is rather moderate. Nevertheless, even the mean values are not far away from those codified as best, especially if linked to the starting point of the optimization. In practice, therefore, the three-run scenario can provide an adequately good solution in most cases.

Another interesting inquiry is related to the comparison of the obtained solution to the exact solution for the problem under consideration, if the latter can be known a priori. In general, the exact solution is not known and cannot be established a priori, at least in large and randomly developed case studies. In the current case study, and following the project data setup, the solution for the 200-day project duration can be developed manually and ends up in a fully leveled resource histogram. In the other cases, the exact solution for the whole project is not known but it can be ascertained that the resource histogram is not totally plain due to the fact that sub-projects in serial arrangement do not overlap with each other while activity J is executed alone in every sub-project. Comparisons of the outputs in Table 3 that correspond to the 200-day project, versus all other durations, indicate that the algorithm's effectiveness is consistently similar across all duration cases, regardless of the quality of the exact solution. This suggests that the quality of the exact solution does not significantly impact the quality of the approximate solution developed by the evolutionary algorithm. In other words, the evolutionary algorithm produces robust solutions that are effective, even without knowledge of the exact optimal solution.

Besides the solution quality, another typical measure of algorithm effectiveness is the required computational effort (or time) to reach a solution. Figure 17 presents the typical convergence path of the employed algorithm and software across alternative optimization criteria. The convergence results are normalized with respect to the objective values at the start of the process to facilitate comparisons among different criteria; this is indicated with the unity value of the y-axis at the beginning of time. In each criterion, results of multiple runs are included to provide a more representative picture of the convergence process. Since all optimization runs have identical computation processes with all individual parameters and the fitness value being calculated in every case, the presented curves are quite typical of all case studies that were examined. This means that the majority of the runs have attained high convergence to their final fitness value within 10 to 15 min, which can be deemed satisfactory for resource optimization of a project of that size (160 activities), considering also that this is an off-line analysis and does not require real-time decisions. It is interesting to note that the convergence pattern of the single RIO criterion is typically quite gradual within the convergence timespan. Instead, all other criteria gain much of their convergence at the beginning of the process. In the case of two sequential runs with different criteria (Cases 7 and 8), each run takes approximately the same amount of time as if it were executed independently.

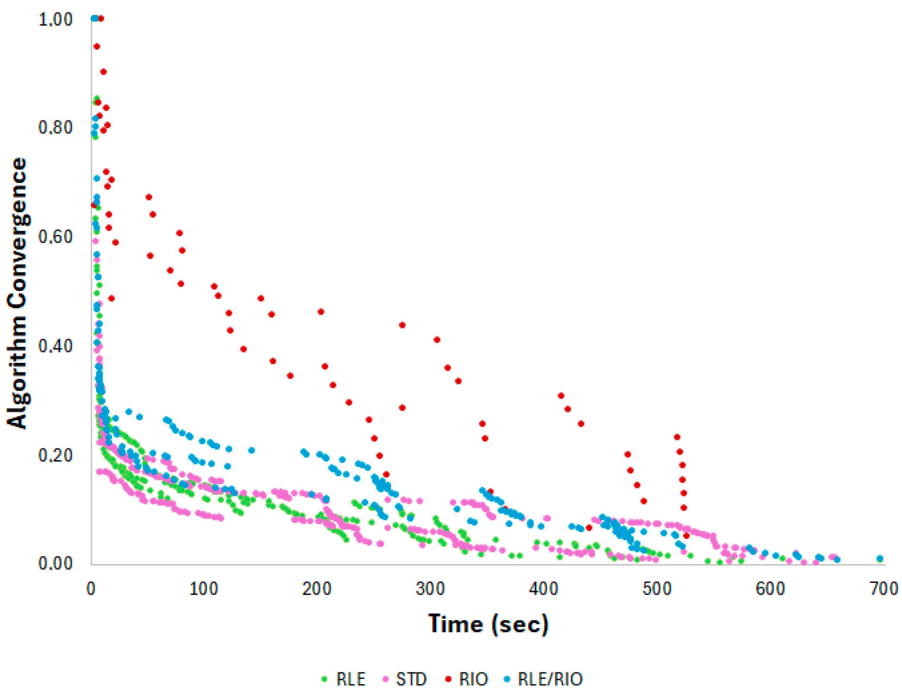

Figure 17. Algorithm convergence progress (200-day case).

5. Discussion

The resource-constrained scheduling problem is a multi-objective one with varying objective priorities in each project case and deployment environment. Thus, different optimization criteria should be developed in order to utilize those that better fit the individual problem objectives and characteristics. Focusing merely on a single sub-objective provides a partial view of the solution potential and typically prevents participants from making informed and concise decisions.

In this work, an optimization model is developed for multi-objective resource-constrained scheduling aimed at large and repetitive projects. Several specific parameters (decision criteria) have been integrated in a multi-objective model that attempts to minimize the negative impact from not meeting the project priorities and constraints. These parameters are associated with (a) the exceedance of the daily resource availability and (b) the day-by-day fluctuations of resource needs. The above deviations result in direct financial losses, which can be assessed depending on the specificities of each particular project. On the other hand, the above sub-objectives are partly conflicting, meaning that a holistic evaluation, including single- and multi-criteria analysis is performed in any potentially productive way. The present study examines different objectives and criteria, either separately or in combination, to evaluate the degree to which each optimization structure facilitates certain or prevailing objectives in actual projects.

To serve the needs for easy and practical application, which is important in scheduling operations of actual (construction) projects, the problem is structured in a spreadsheet form where scheduling and resource allocation calculations are made. Evolutionary algorithms have been found to perform well in such problem environments and the present work employs Genetic Algorithms in this direction. In addition, known general-purpose GA-solving commercial software has been employed. In construction industry practice, it is expected to use such type of assistance rather than developing an optimization model internally. Finally, due to the stochastic nature of evolutionary algorithms, it is recommended to implement multiple runs of a specific optimization setup to develop a clearer picture of the solution potential.

Optimization is performed in the form of a weighted-sum function of several parameters (criteria) that can potentially be used in combination for providing global decision-making assistance. The weighted-sum structure perfectly fits the problem nature. This is because the weights represent (at least in some of the included parameters, RLE and RIO) explicit actual unit cost values of the corresponding resource inefficiencies. As such, the objective function provides a pure cost value that can undisputably be used for assessing the specific project schedule, which minimizes the total resource allocation cost. Among the decision parameters that have been considered, the STD, RLE, and MaxR promote the goals of resource leveling and resource overallocation prevention. In another optimization direction, the RIO parameter can better represent the adverse effect of resource movement in and out of the project. The analysis reveals that all criteria can potentially end up with comparatively enhanced solutions and, therefore, it is advised to perform runs with all criteria, either individually or collectively. It is also advised to make multiple runs with each set of criteria as the output of an evolutionary type of search is not deterministic. The experimentation with the case study example shows that the employment of such a global analysis generally is in favor of improved results.

The case study results indicate that the optimization with the proposed strategy can provide rather effective resource allocation patterns, both in terms of resource overallocation and resource variations from day to day. In all cases that were examined, both the resource overallocation and the resource fluctuations in time are within a range of 3% to 7.5% of the total resources that are used in the project. This deviation appears to be acceptable for a project of such size (160 activities), especially if the exact solution (which is rarely known) also presents a degree of resource allocation inefficiencies. This means that the net deficit of the evolutionary algorithm solution, in comparison to the corresponding exact solution, is even lower than the percentages mentioned above.

6. Conclusions

The resource-constrained scheduling problem is a complex combinatorial one and includes several activity and resource input parameters as well as constraints that need to be met. As the project size (and complexity) rises, e.g., in the case of large and repetitive projects, the computational burden increases fast and the solution quality drops. In such cases, metaheuristic algorithms may be employed for solving the problem effectively (near-optimal solutions) and efficiently (reasonable time requirements).

The findings of the present work indicate that such methods can provide valuable and thorough decision support in multi-objective resource-constrained scheduling considering the specific priorities and objectives in every project case, including large and repetitive projects. In fact, although the optimization effectiveness may drop with the project's size and complexity, the importance of using formal optimization becomes more imperative in the case of large or repetitive projects.

This work demonstrates that evolutionary algorithms, particularly Genetic Algorithms, can be effective tools for solving the RCSP in large and repetitive projects. The integration of multiple optimization criteria allows for a more comprehensive evaluation of potential solutions, enhancing decision-making. The exploitation of user-friendly software tools and easy-to-implement solution strategies is essential for the broader adoption of these optimization methods in engineering practice. By addressing both theoretical and practical aspects, this study contributes to the advancement of optimization techniques in project management and paves the way for future research and practical applications in the field. In fact, future work should further focus on closing the gap between research and practice by incorporating practical aspects associated with the construction industry's needs and specificities.

Author Contributions: Conceptualization, V.L. and A.C.; Methodology, V.L. and A.C.; Software, V.L. and A.C.; Validation, V.L., A.C. and S.K.; Formal Analysis, V.L., A.C. and S.K.; Investigation, V.L. and A.C.; Resources, V.L., A.C. and S.K.; Data Curation, V.L. and A.C.; Writing—Original Draft Preparation, V.L. and A.C.; Writing—Review and Editing, V.L., A.C. and S.K.; Visualization, V.L. and A.C.; Supervision, A.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author/s.

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

References

1. Berthaut, F.; Grèze, L.; Pellerin, R.; Perrier, N.; Hajji, A. *Optimal Resource-Constraint Project Scheduling with Overlapping Modes*; Cirrelt: Montreal, QC, Canada, 2011; pp. 1–15.
2. Hariga, M.; Shamayleh, A.; El-Wehedi, F. Integrated Time–Cost Tradeoff and Resources Leveling Problems with Allowed Activity Splitting. *Int. Trans. Oper. Res.* **2019**, *26*, 80–99. [CrossRef]
3. Moukrim, A.; Quilliot, A.; Toussaint, H. An Effective Branch-and-Price Algorithm for the Preemptive Resource Constrained Project Scheduling Problem Based on Minimal Interval Order Enumeration. *Eur. J. Oper. Res.* **2015**, *244*, 360–368. [CrossRef]
4. Alcaraz, J.; Maroto, C.; Ruiz, R. Solving the Multi-Mode Resource-Constrained Project Scheduling Problem with Genetic Algorithms. *J. Oper. Res. Soc.* **2003**, *54*, 614–626. [CrossRef]
5. Kaiafa, S.; Chassiakos, A.P. A Genetic Algorithm for Optimal Resource-Driven Project Scheduling. *Procedia Eng.* **2015**, *123*, 260–267. [CrossRef]
6. Hegazy, T.; Kamarah, E. Schedule Optimization for Scattered Repetitive Projects. *Autom. Constr.* **2022**, *133*, 104042. [CrossRef]
7. Khalilzadeh, M. Resource Levelling in Projects Considering Different Activity Execution Modes and Splitting. *J. Eng. Des. Technol.* **2021**, *20*, 1073–1100. [CrossRef]
8. Bettemir, Ö.; Sonmez, R. Hybrid Genetic Algorithm with Simulated Annealing for Resource-Constrained Project Scheduling. *J. Manag. Eng.* **2015**, *31*, 04014082. [CrossRef]
9. Li, H.; Zhang, H. Ant Colony Optimization-Based Multi-Mode Scheduling under Renewable and Nonrenewable Resource Constraints. *Autom. Constr.* **2013**, *35*, 431–438. [CrossRef]

10. Myszkowski, P.B.; Skowroński, M.E.; Olech, Ł.P.; Oślizło, K. Hybrid Ant Colony Optimization in Solving Multi-Skill Resource-Constrained Project Scheduling Problem. *Soft Comput.* **2015**, *19*, 3599–3619. [CrossRef]
11. Geem, Z.W. State-of-the-Art in the Structure of Harmony Search Algorithm. In *Recent Advances in Harmony Search Algorithm*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 1–10. [CrossRef]
12. Kumar, N.; Vidyarthi, D.P. A Model for Resource-Constrained Project Scheduling Using Adaptive PSO. *Soft Comput.* **2016**, *20*, 1565–1580. [CrossRef]
13. Akbari, R.; Zeighami, V.; Ziarati, K. Artificial Bee colony for resource constrained project scheduling problem. *Int. J. Ind. Eng. Comput.* **2011**, *2*, 45–60. [CrossRef]
14. Ziarati, K.; Akbari, R.; Zeighami, V. On the Performance of Bee Algorithms for Resource-Constrained Project Scheduling Problem. *Appl. Soft Comput. J.* **2011**, *11*, 3720–3733. [CrossRef]
15. Eshraghi, A. A New Approach for Solving Resource Constrained Project Scheduling Problems Using Differential Evolution Algorithm. *Int. J. Ind. Eng. Comput.* **2016**, *7*, 205–216. [CrossRef]
16. Anagnostopoulos, K.P.; Koulinas, G.K. A Simulated Annealing Hyperheuristic for Construction Resource Levelling. *Constr. Manag. Econ.* **2010**, *28*, 163–175. [CrossRef]
17. Koulinas, G.; Kotsikas, L.; Anagnostopoulos, K. A Particle Swarm Optimization Based Hyper-Heuristic Algorithm for the Classic Resource Constrained Project Scheduling Problem. *Inf. Sci.* **2014**, *277*, 680–693. [CrossRef]
18. Christodoulou, S.; Ellinas, G.; Aslani, P. Entropy-Based Scheduling of Resource-Constrained Construction Projects. *Autom. Constr.* **2009**, *18*, 919–928. [CrossRef]
19. Roca, J.; Pugnaghi, E.; Libert, G. Solving an Extended Resource Leveling Problem with Multiobjective Evolutionary Algorithms. *Int. J. Comput. Intell.* **2008**, *4*, 289–300.
20. Beşikci, U.; Bilge, Ü.; Ulusoy, G. Multi-Mode Resource Constrained Multi-Project Scheduling and Resource Portfolio Problem. *Eur. J. Oper. Res.* **2015**, *240*, 22–31. [CrossRef]
21. Mathew, J.; Paul, B.; Dileeplal, J.; Mathew, T. Multi Objective Optimization for Scheduling Repetitive Projects Using GA. *Procedia Technol.* **2016**, *25*, 1072–1079. [CrossRef]
22. 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]
23. El-Abbasy, M.S.; Elazouni, A.; Zayed, T. Generic Scheduling Optimization Model for Multiple Construction Projects. *J. Comput. Civ. Eng.* **2017**, *31*, 04017003. [CrossRef]
24. Eid, M.S.; Elbeltagi, E.E.; El-Adaway, I.H. Simultaneous Multi-Criteria Optimization for Scheduling Linear Infrastructure Projects. *Int. J. Constr. Manag.* **2021**, *21*, 41–55. [CrossRef]
25. Samuel, B.; Mathew, J. Resource Allocation in a Repetitive Project Scheduling Using Genetic Algorithm. *IOP Conf. Ser. Mater. Sci. Eng.* **2018**, *330*, 012098. [CrossRef]
26. Salama, T.; Moselhi, O. Multi-Objective Optimization for Repetitive Scheduling under Uncertainty. *Eng. Constr. Archit. Manag.* **2019**, *26*, 1294–1320. [CrossRef]
27. García-Nieves, J.D.; Ponz-Tienda, J.L.; Ospina-Alvarado, A.; Bonilla-Palacios, M. Multipurpose Linear Programming Optimization Model for Repetitive Activities Scheduling in Construction Projects. *Autom. Constr.* **2019**, *105*, 102799. [CrossRef]
28. Kannimuthu, M.; Raphael, B.; Ekambaram, P.; Kuppuswamy, A. Comparing Optimization Modeling Approaches for the Multi-Mode Resource-Constrained Multi-Project Scheduling Problem. *Eng. Constr. Archit. Manag.* **2020**, *27*, 893–916. [CrossRef]
29. Abido, M.A.; Elazouni, A. Modified Multi-Objective Evolutionary Programming Algorithm for Solving Project Scheduling Problems. *Expert Syst. Appl.* **2021**, *183*, 115338. [CrossRef]
30. Yuan, Y.; Ye, S.; Lin, L.; Gen, M. Multi-Objective Multi-Mode Resource-Constrained Project Scheduling with Fuzzy Activity Durations in Prefabricated Building Construction. *Comput. Ind. Eng.* **2021**, *158*, 107316. [CrossRef]
31. Sharma, K.; Trivedi, M.K. Development of Multi-Objective Scheduling Model for Construction Projects Using Opposition-Based NSGA III. *J. Inst. Eng. Ser. A* **2021**, *102*, 435–449. [CrossRef]
32. Dai, G.; Liao, M.; Zhang, R. Resource Levelling in Repetitive Construction Projects with Interruptions: An Integrated Approach. *J. Civ. Eng. Manag.* **2023**, *29*, 93–106. [CrossRef]
33. He, Y.; Jia, T.; Zheng, W. Tabu Search for Dedicated Resource-Constrained Multiproject Scheduling to Minimise the Maximal Cash Flow Gap under Uncertainty. *Eur. J. Oper. Res.* **2023**, *310*, 34–52. [CrossRef]
34. Bredael, D.; Vanhoucke, M. A Genetic Algorithm with Resource Buffers for the Resource-Constrained Multi-Project Scheduling Problem. *Eur. J. Oper. Res.* **2024**, *315*, 19–34. [CrossRef]
35. Ding, H.; Zhuang, C.; Liu, J. Extensions of the Resource-Constrained Project Scheduling Problem. *Autom. Constr.* **2023**, *153*, 104958. [CrossRef]
36. Lazari, V.; Chassiakos, A.; Karatzas, S. Sensitivity Analysis for Different Size Resource-Constrained Scheduling Problems. In Proceedings of the 9th International Conference on Construction Engineering and Project Management, Las Vegas, NV, USA, 20–23 June 2022.

37. Gómez Sánchez, M.; Lalla-Ruiz, E.; Fernández Gil, A.; Castro, C.; Voß, S. Resource-Constrained Multi-Project Scheduling Problem: A Survey. *Eur. J. Oper. Res.* **2023**, *309*, 958–976. [CrossRef]
38. Lazari, V.; Chassiakos, A. Multi-Objective Resource-Constrained Scheduling in Construction Projects. In *Collaboration and Integration in Construction, Engineering, Management and Technology: Proceedings of the 11th International Conference on Construction in the 21st Century, 9–11 September 2019, London, UK*; Springer International Publishing: London, UK, 2021; pp. 583–588.