MODELING AND ANALYSIS OF JOINT WORK PACKAGE SIZING AND PROJECT SCHEDULING CONSIDERING RESOURCE CONSTRAINTS

Nianmin Zhang[1] and Xiao Li[1]
[1]Department of Civil Engineering, The University of Hong Kong SAR, China

Abstract

Effective work package sizing and project scheduling are essential problems in construction project management. In current studies, they are often treated as separate optimization problems. However, in practical situations where resources are limited, there is a growing recognition of the need to integrate these two processes to achieve more efficient and effective project management. This research aims to fill this gap by developing an integer non-linear programming model incorporating work package sizing and project scheduling while considering resource limitations. To evaluate the effectiveness of our model, we utilized Gurobi optimization software to solve nine project instances obtained from RanGen, each characterized by its unique complexities. Furthermore, we analyzed the impact of resource utilization on work package sizing and the project schedule, and we found that both excessive and inadequate resource availability can increase project costs.

Introduction

Effective project scheduling is a crucial problem in project management, as highlighted by several studies (Li et al., 2018; Tsz Wai et al., 2023). This problem becomes even more challenging in a resource-constrained environment, where project tasks exhibit interdependencies in two key aspects. Firstly, tasks compete for limited resources, and sufficient availability of resources is essential for initiating and executing construction tasks. Secondly, the project workflow imposes precedence constraints, whereby certain tasks must be performed based on the completion of specific scheduled tasks. These two interdependencies are encompassed within the resource-constrained project scheduling problem (RCPSP), which is widely recognized and formalized in the project scheduling literature (Hartmann and Briskorn, 2022).

From a perspective of scheduling scope and complexity, RCPSP can be categorized into two groups. The first group is the resource-constrained single project scheduling problem (RCSPSP), which aims to optimize the schedule and resource allocation for a single project to ensure timely project completion. In RCSPSP, the scheduling process is task-based, and each task to be scheduled has specified work content, such as processing duration and resource requirement, as well as an execution scope, such as precedence relationships among tasks. The second group is the resource-constrained multiple project scheduling problem (RCMPSP), which involves coordinating and prioritizing schedules across multiple projects that share common resources. The objective is to balance the workload across projects, ensure the timely completion of each individual project, and achieve the overall goals of the project portfolio (Issa and Tu, 2020).

With the implementation of the work breakdown structure (WBS) method in the construction industry, the RCPSP for construction projects has evolved into a work package-based scheduling problem, which can be considered as a complex extension of the RCPSP (Li et al., 2022). For example, the production of a modular room can be decomposed hierarchically into six work packages, including structure production, door/window production, wall production, print, electric service, and test, using WBS tools for planning or scheduling (Liu et al., 2023). Each work package typically consists of one or more elemental tasks that are not further decomposed (Li and Hall, 2019). Specifically, the work package for structure production may include elemental tasks such as board production, 2D assembly, and concrete curing. Therefore, in construction projects, not only should the resource and precedence constraints be considered in the RCPSP, but the size of the work package also has a significant impact on project performance. Specifically, smaller work packages tend to increase project complexity and workload while diminishing economies of scale. Conversely, larger work packages require a higher allocation of resources, potentially straining resource availability and leading to bottlenecks or delays. Furthermore, as demonstrated by Blazewicz et al. (1983), the RCPSP belongs to the category of strongly NP-hard problems, presenting a significant challenge in obtaining the optimal solution for RCPSP and acting as a bottleneck for further improvements in project performance. By considering a more comprehensive set of influential factors in the RCPSP, there is potential to enhance project management capabilities and improve the efficiency of resource utilization.

Work package sizing has garnered considerable attention, with studies demonstrating the significance of optimizing work package size to reduce project costs (Li and Hall, 2019; Li et al., 2021b). This optimization encompasses various factors, including minimizing costs associated with inaccurate estimation, monitoring, and control of work packages. Moreover, the sizing of work packages has implications for the project's economies of scale and discounted cash flow, further emphasizing the importance of this optimization in cost reduction endeavors (Liu et al., 2023).

While the optimization of work package sizing has enabled a trade-off among multiple objectives, there are still limitations in current studies that hinder the improvement of project performance. One limitation is that the work package sizing process often takes place during the initial planning stage of a construction project, without considering detailed task information such as task durations, resource requirements, and precedence relations among tasks (Li et al., 2023). However, resource constraints are common in construction projects and significantly impact the feasibility and effectiveness of work packages (Li et al., 2021a). This can lead to the creation of unsatisfactory project schedules during the project scheduling stage, ultimately resulting in poor project performance. Moreover, the work packages in RCPSP are predetermined and do not allow for splitting or recombination, limiting the flexibility of project managers to schedule work packages and effectively utilize resources in the project scheduling stage (Du et al., 2021).

To summarize, work package sizing and construction project scheduling are interdependent activities that should be considered simultaneously to mitigate project delays, cost overruns, and disruptions, particularly in resource-constrained and uncertain project scenarios. Work package sizing involves grouping tasks into cohesive work packages, which determine the resource requirements and processing time. These factors then influence scheduling decisions and the overall project cost. Project scheduling decisions, in turn, have a significant impact on resource availability, which affects the feasibility of work package splitting and recombination (Pellerin et al., 2020). By integrating work package sizing and project scheduling, project stakeholders can better manage resources, optimize project timelines, and mitigate potential bottlenecks or resource conflicts (Zaman et al., 2021).

However, most studies treat work package sizing and project scheduling as separate problems, neglecting the interrelationship between them in resource-constrained construction projects. To the best of our knowledge, no research to date provides an integrated resource-constrained project scheduling and work package sizing method while considering the interrelationship between them. Therefore, it is crucial to propose a new integrated method that combines work package sizing and project scheduling under resource and precedence constraints to reduce project costs while improving resource efficiency (Zhang et al., 2024).

To address the existing research gap, we present a joint model for project scheduling and work package sizing that takes into account both precedence and resource constraints. Our proposed approach makes the following contributions: 1) For the first time, we introduce an integrated model that considers the interplay between project scheduling and work package sizing. 2) To validate the effectiveness of our model, we conducted various experiments using Gurobi with nine project instances obtained from RanGen. 3) We analyze the impact of resources on project schedule and work package size by changing availability.

Methodology

Assumptions

- We assume that the project has been planned into a given set of tasks that are elemental. That is, no further breakdown of those tasks is possible for reasons that may be technical, logistical, financial, administrative, or cultural.
- Only the active tasks can be grouped into a work package due to the specific role in the construction project.
- Considering the mass production in real scenarios (e.g., reduction in set-up time, material preparation, mass workforce, and new production tech), lag time is considered among tasks that are grouped into the same package, while the package-to-package is still regarded to be hard precedence-constrained.

Problem Modeling

Consider an task-on-node (TON) project network described by $G(N,A,W)$, where N denotes a set of tasks in the project ($N = \{0,1,2,\ldots,\bar{n},\bar{n}+1,\ldots,\bar{n}+m,\bar{n}+m+1\}$). Each activity has a given work duration $d_i \geq 0, i \in N$ and given work content $x_i \geq 0, i \in N$. We assume that there are K types of renewable resources, and each resource $k \in \{1,2,3,\ldots,K\}$ has a limited capacity a_k available. When a task i is executed, its requirement for resource type $k \in \{1,\ldots,K\}$ at each time period is a fixed integer $r_{i,k} \geq 0$. Tasks 0 and $\bar{n}+m+1$ are the dummy start and end of the project $((Task_0, Task_{\bar{n}+m+1}) \in N_{Dummy})$, whose process duration, work content and required resource are both 0. More specifically, let $N_A = \{1,\ldots,\bar{n}\}$ denotes the active tasks that can be grouped into work packages, while $N_I = \{\bar{n}+1,\ldots,\bar{n}+m\}$ is the inactive task set, which can only be regarded as single work package due to several fac-tors, such as the responsibility assignment, internal cohe-sion or risk control (Li and Hall, 2019); thus we can get that $N = N_A \cup N_I \cup N_{Dummy}$. A represents a set of precedence relationships among tasks, $(A \subseteq N \times N)$, if $(i,j) \in A$ that means $Task_j$ cannot start before $Task_i$ is finished. W is the set of weights on the arcs that denotes the minimum time lags between tasks grouped into a single work package, $W = \{\delta_{i,j} \geq 0 | (i,j) \in A\}$. When $s_i - s_j > \delta_{i,j}, (i,j) \in A$ is satisfied, the $Task_i$ can be conducted, where s_i and s_j are the start times of tasks i and j,respectively. More specifi-cally, (1) if $\delta_{i,j} = 0$, then tasks i and j can be completely overlapped, i.e. they can be executed in parallel; (2) if $d_i \geq \delta_{i,j}$, then tasks i and j can be partially overlapped, i.e. activity j can only begin after activity i has been in ex-ecution for at least $\delta_{i,j}$ time units; (3) if $\delta_{i,j} \geq d_i$, then tasks i and j are executed sequentially, i.e. activity j can only begin after activity i is completed. Figure 1 presents an example illustrating the above three cases of activity overlap, namely, complete overlap, partial overlap, and no overlap with respect to two tasks. These variables and additional notation are summarized in Table 1.

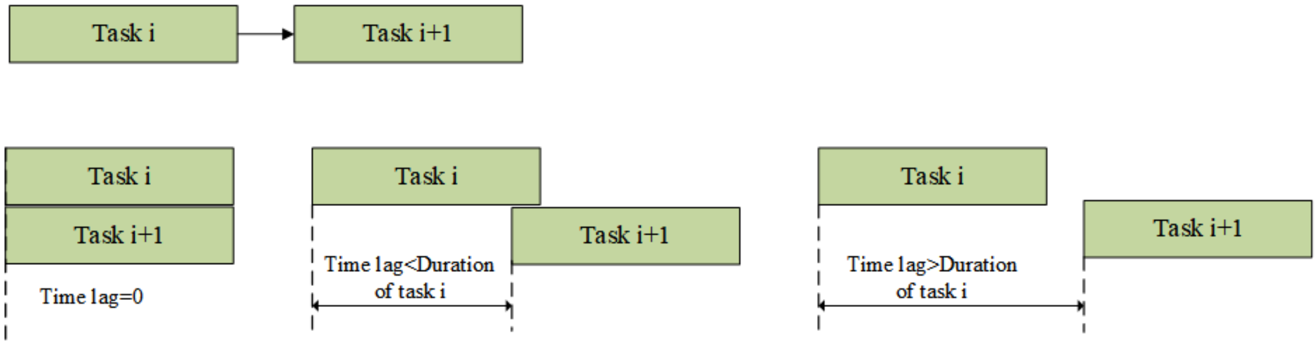

Figure 1: The lag time of the tasks.

The Objective Function

In this study, we adopted a combination of objectives (minimize the total project makespan and project cost) to optimize the work package sizing and work package scheduling in a resource-constrained environment (Li and Hall, 2019). Assuming that all the tasks are ready to be allocated to a total of $\bar{n}+m$ work packages, the objective function can be expressed as follows:

$$Minimize \qquad J = \lambda F_t + (1-\lambda)F_c, \tag{1}$$

$$F_c = \omega(p+m) + \sum_{j=1}^{\bar{n}+m} F(X_{w_j}) + \xi \sum_{j=1}^{\bar{n}+m} X_{w_j}(1-e^{-\alpha C_{w_j}}), \tag{2}$$

$$F_t = Max(C_{w_j}), \qquad \forall j \in \{1,2,...,\bar{n},...\bar{n}+m\} \tag{3}$$

$$F(X_{w_j}) = f(X_{w_j}) + g(X_{w_j}) + h(X_{w_j}),$$
$$\forall j \in \{1,2,...,\bar{n},...\bar{n}+m\} \tag{4}$$

where F_c and F_t are the project cost and makespan, and λ is the weight of the time part. For the most part, $p \in \{\bar{p},\bar{p}+1,...,\bar{n}+m\}$ is the number of work packages that contain at least one task, where $\bar{p}$ is the lower bound for any feasible solutions that can be derived from the theory of critical path. ω is the fixed cost of the work package and X_{w_j} is the j_{th} work package's work content. C_{w_j} is the complete time of the j_{th} work package. We model the cost of inaccuracy in time estimation of a work package as function $f(X_{w_j})$, the cost for monitoring and controlling the progress of the work package as function $g(X_{w_j})$ and the economies of scale from repetition and similarity of tasks within a work package as function $h(X_{w_j})$, which are functions of the work packages' work content. Additionally, work package sizing poses impacts on cash flow, which is modeled as $\xi \sum_{j=1}^{\bar{n}+m} X_{w_j}(1-e^{-\alpha C_{w_j}})$, where ξ and α are parameters for cash flow cost function.

Work Package Sizing Constraints

work package sizing is conducted based on the structure of the given activity network and relies on the requirement that the network of work packages cannot contain cycles (Lambrechts et al., 2008). Additionally, the active tasks need to be allocated to the work package $P_A \in \{1,2,3,...,\bar{n}\}$, while the inactive tasks should be allocated to $P_I \in \{\bar{n},\bar{n}+1,...,\bar{n}+m\}$. $b_{i,j}$ is the decision variable, which is 1 if activity $i \in N$ is allocated to work package $j \in P_A \cap P_I$, and 0 otherwise. The work package sizing constraints are as follows:

$$b_{i,j} + b_{i',j} \leq 1, \qquad \forall i,i' \in N_A,$$
$$\forall i' \in \bigcup_{i'' \in Succ(i) \cap N_I} Succ(i''), \forall j \in \{1,2,3...,\bar{n}\} \tag{5}$$

$$b_{i,j} + b_{i',j} \leq 1, \qquad \forall i,i' \in N_A,$$
$$\forall i' \in \bigcup_{i'' \in Pre(i) \cap N_I} Pre(i''), \quad \forall j \in \{1,2,3...,\bar{n}\} \tag{6}$$

$$b_{i',j} b_{i'',j} \leq b_{i,j}, \qquad \forall i,i',i'' \in N_A,$$
$$\forall i' \in Succ(i), \quad \forall i'' \in Pre(i), \quad \forall j,j' \in \{1,...,\bar{n}\} \tag{7}$$

$$\sum_{j \in P_A} b_{i,j} = 1, \qquad \forall i \in N_A \tag{8}$$

$$\sum_{j \in P_I} b_{i,j} = 1, \qquad \forall i \in N_I \tag{9}$$

$$\sum_{i \in N_I} b_{i,j} = 1, \qquad \forall j \in P_I \tag{10}$$

$$z_j = max\{b_{i,j}\}, \qquad \forall j \in P_A, \quad \forall i \in N_A \tag{11}$$

$$p = \sum_{j \in P_A} z_j, \tag{12}$$

$$X_{w_j} = \sum_{i \in N} b_{i,j} x_i, \qquad \forall j \in \{1,...,\bar{n}+m\} \tag{13}$$

$$r_{w_j,k} = \sum_{i \in N} b_{i,j} r_{i,k}, \qquad \forall j \in \{1,...,\bar{n}+m\}, \quad \forall k \in K \tag{14}$$

$$b_{i,j} \in \{0,1\}, \qquad \forall i \in N, \quad \forall j \in \{1,...,\bar{n}+m\} \tag{15}$$

$$z_j \in \{0,1\}, \qquad \{\forall j \in 1,2,...,\bar{n}+m\} \tag{16}$$

where constraints (5)-(6) refer to that if an inactive task i'' is a predecessor/successor of activity i, then no predecessor/successor of task i'' should be grouped into the same work package with activity i. Constrain (7) shows that any pairs of a predecessor and a successor activity of active activity i are not allowed to be grouped into a single work package while without activity i. Otherwise, there will be a cycle in the work packages network. constraints (8)-(10) represent that an active/inactive activity can only be allocated to an active/inactive work package, and each inactive work package only contains a single inactive task. We il-

Table 1: Nomenclature

Notaion	Explanation
N	The set of project tasks.
N_A	The set of active tasks.
N_I	The set of inactive tasks.
N_{Dummy}	The set of dummy tasks.
K	The set of resource types.
T	The set of time.
P_A	The set of work packages for active tasks,$P_A = \{1,2,3,...,\bar{n}\}$.
P_I	The set of work packages for inactive packages, $P_I = \{\bar{n}+1,\bar{n}+2,\bar{n}+3,...,\bar{n}+m\}$.
d_i	Duration of task $i \in N$.
$r_{i,k}$	The quantity of required resource $k \in K$ for task $i \in N$.
$\delta(i',i)$	Time lag between task $i \in N_A$ and task $i' \in N_A$.
a_k	The initial quantity of resource $k \in K$.
c_k	The cost of each unit of resource $k \in K$.
x_i	Work content of task $i \in N$.
$b_{i,j}$	1 if task $i \in N_A$ is allocated to active work package $j \in P_A$; 0 otherwise.
$q_{i,t}$	1 if task $i \in N$ starts at time t; 0 otherwise.
$a_{j,t}$	1 if work package $j \in P_A \cap P_I$ is processed at time t; 0 otherwise.
z_j	1 if active work package $j \in P_A$ is not empty; 0 otherwise.

lustrate the relationship between the active tasks and the active work packages in Constraint (11) and Constraint (16). If some active tasks are grouped into active work package j, z_j is 1, which means that the active work package is not empty, and 0 otherwise. Constraint (12)-(14) shows that we can get p effective active work packages that are not empty, and the work content and the resource requirement of these work packages are the sum of the tasks in the corresponding work package. Constrain (15) shows that $b_{i,j}$ is a binary decision variable.

Work Package Scheduling constraints

Assuming that the project is conducted within $T \in (T_{lower}, T_{upper})$ time units, where T_{lower} and T_{upper} are the lower bound and upper bound of a project. They can be estimated by the project's activity graph. Taking into account the presence of precedence and resource constraints both among work packages and tasks within the same work package, we outline the following work package scheduling constraints:

$$\sum_{t\in T} t q_{i,t} = s_i, \qquad \forall i \in N \tag{17}$$

$$\sum_{t\in T} q_{i,t} = 1, \qquad \forall i \in N \tag{18}$$

$$s_i \geq b_{i',j} C_{w_j}(1-b_{i,j}), \qquad \forall i \in N, \quad \forall j \in P_A \cup P_I, \quad \forall i' \in Pre(i) \tag{19}$$

$$s_i \geq b_{i,j} b_{i',j}(s_{i'} + \delta(i',i)), \qquad \forall i \in N_A, \quad \forall j \in P_A, \quad \forall i' \in Pre(i) \cap N_A \tag{20}$$

$$c_i = d_i + s_i \qquad \forall i \in N, \tag{21}$$

$$S_{w_j} = min\{s_i b_{i,j}\}, \qquad \forall i \in N, \quad \forall j \in \{1,...,\bar{n}+m\} \tag{22}$$

$$C_{w_j} = max\{c_i b_{i,j}\}, \qquad \forall i \in N, \quad \forall j \in \{1,...,\bar{n}+m\} \tag{23}$$

$$a_{j,t} = \sum_{i\in N_I} \sum_{t'=t-d_i+1}^{t} b_{i,j} q_{i,t'}, \qquad \forall j \in P_I \tag{24}$$

$$\sum_{i\in N_A} \sum_{t'=t-d_i+1}^{t} b_{i,j} q_{i,t'} - M a_{j,t} \leq 0, \qquad \forall j \in P_A \tag{25}$$

$$a_k \geq \sum_{j=1}^{\bar{n}+m} r_{w_j,k} a_{j,t}. \qquad \forall t \in T \tag{26}$$

$$q_{i,t} \in \{0,1\}, \qquad \forall i \in N \tag{27}$$

$$a_{j,t} \in \{0,1\}, \qquad \forall j \in P_A \cap P_{I]} \quad \forall t \in T \tag{28}$$

where the variables s_i and c_i represent the starting time and completion time of activity i, while S_{w_j} and C_{w_j} represent the starting time and completion time of work package j, respectively. The decision variable $q_{i,t}$ in constrain (17) corresponds to the start processing time of the i_{th} activity, taking a value of 1 if activity i starts at time t, and 0 otherwise. Constraint (17) indicates that task execution is non-preemptive. Once a task starts at time t, it will continue without interruption until the task is complete. Constraints (19)-(20) enforce precedence constraints on the start time of task i. More specifically, constraint (19) ensures that task i can only begin after all the predecessor work packages have been completed. Constraint (20) specifies that the start time of activity i within work package j must not exceed the start time of the preceding activity within the same work package, and any required lag time must be satisfied prior to execution. Constraints (22)-(23) indicate that the starting and completion times of the work package are determined based on the earliest and latest starting and completion times of the tasks within that work package. Specifically, constraint (22) specifies that the starting time of the work package cannot be later than the earliest starting time among its tasks. On the other hand, constraint (23) ensures that the completion time of the work package is not earlier than the latest completion time among its tasks. After that, constraints (24)-(26) give the resource constraints for the work packages. $a_{j,t}$ is 1 if any tasks in work package w_j is processing at time t, 0 otherwise. Constrain (24) gives $a_{j,t}$ for inactive work packages while constrain (25) gives it for active work packages using the big-M method. Constraint (26) illustrates that the amount of resources $\sum_{j=1}^{\bar{n}+m} r_{w_j,k} a_{j,t}$ needed by all the executing work packages at any time t should not exceed the current amount of available resources a_k.

Experiments and Analysis

This section validates the effectiveness of the proposed work package sizing and scheduling model with nine projects from Rangen (Demeulemeester et al., 2003; Vanhoucke et al., 2008). These nine projects are categorized into three types, each consisting of a different number of tasks: 10, 20, and 30, respectively. The experiments are conducted using Gurobi 11.0 optimization software and executed on a PC platform running the Windows 11 64-bit operating system, Intel i9-13900KF processor.

Validation of Our Model

To obtain the nine projects, we first generate the task networks using RanGen2 with the following network measures:

- Network size indicator I_1: the number of the tasks including the active tasks and inactive tasks.
- Serial or parallel indicator I_2: $I_2 = (\eta - 1)/(I_1 - 1)$, where η is the depth of the network, and $0 \le I_2 \le 1$.
- Resource factor RF: an indicator of average resource availability

$$\mathrm{RF} = \frac{1}{nK} \sum_{i=1}^{n} \sum_{k=1}^{K} \begin{cases} 1, & \text{if } r_{ik} > 0 \\ 0, & \text{otherwise} \end{cases}$$

- Resource strength (RS): an indicator of resource availability at the peak time, where $r_k^{\min} = max\{r_{i,k}\}, \forall i \in N$ and $r^{\max}$ is the peak requirement of resource $k \in K$ in a schedule with the precedence constraints

$$\mathrm{RS}_k = \frac{a_k - r_k^{\min}}{r_k^{\max} - r_k^{\min}} \tag{29}$$

If the value of I_2 approaches 1, the generated network exhibits characteristics similar to a serial network, where tasks are sequentially executed. Conversely, if I_2 approaches 0, the network resembles a parallel network, where all tasks can be executed simultaneously. RF represents the typical distribution of resource types that are requested by each activity, while RS serves as a metric for assessing the level of complexity involved in scheduling a project that is constrained by limited resources. As for the remaining network measures of RanGen, we utilize their default settings to ensure consistency and comparability in our analysis. We have defined the following parameters for the three groups of project instances: I_i is set to $\{10, 20, 30\}$, I_2 is set to $\{0.8, 0.8, 0.8\}$, RF is set to $\{0.4, 0.4, 0.4\}$, RS is set to $\{0.2, 0.2, 0.2\}$ and $K = 4$. To evaluate the performance of our model, we have generated three test instances for each group, resulting in a total of nine instances. In accordance with the work by Li and Hall (2019), we have set the discount rate as $\alpha = 0.00025$, and used values of $\lambda = 0.5, \omega = 50$, $\xi = 50$, $f(x) = 3x^{0.8}$, $g(x) = h(x) = x^{1.2}$, to define our objective function. Although Gurobi is more proficient in solving linear models, our problem includes a nonlinear objective term. However, by introducing auxiliary variables, we can still utilize Gurobi to solve it.

The results of the nine projects from RanGen are presented in Table 2. In Group 1, each project instance consists of 10 tasks (nodes), with 8 active tasks and 2 inactive tasks. All experiments for this group were completed within 520 seconds, yielding objective values of 317.31, 325.29, and 304.84 for Project 1, Project 2, and Project 3, respectively. In Group 2, each instance comprises 20 tasks, with 14 active tasks and 6 inactive tasks. However, the running time for each experiment significantly increased, with Project 4, Project 5, and Project 6 taking 15,007, 24,428, and 23,674 seconds, respectively. The corresponding objective values for these projects are 516.16, 579.00, and 532.21. Projects in Group 3 consist of 21 active tasks and 9 inactive tasks, with a total of 30 tasks packaged into approximately 18 packages. Although we were able to achieve objective values of 815.70, 869.14, and 765.43 for Project 7, Project 8, and Project 10, respectively Balouka and Cohen (2021).

Overall, the results of these experiments provide compelling evidence to support the effectiveness of our modeling method that integrates work package sizing and scheduling in a resource-constrained environment. Additionally, our model takes into account the significance of striking a balance between timely completion, reasonable management cost, and maintaining a healthy cash flow in real-world construction projects. This is demonstrated in Table 2, where the makespan is included as part of the objective. By incorporating these relevant practical considerations, our model offers a more comprehensive and realistic solution for effectively managing construction projects in real-world scenarios.

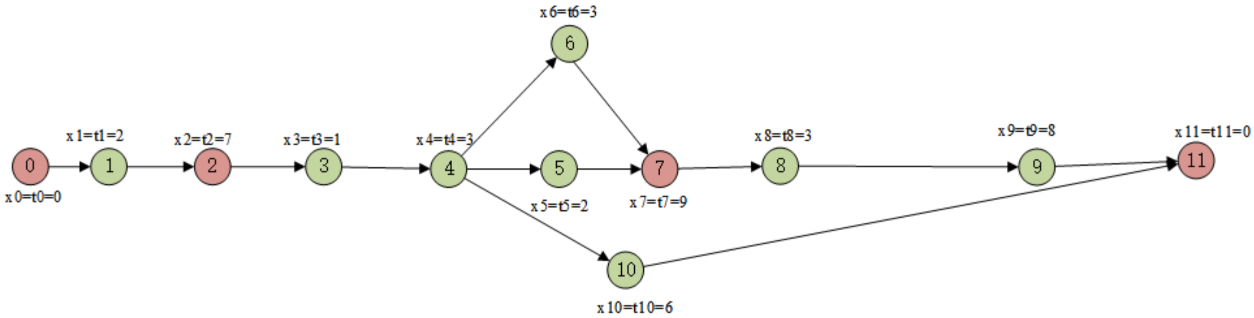

Figure 2: The initial structure of project 1 in group 1.

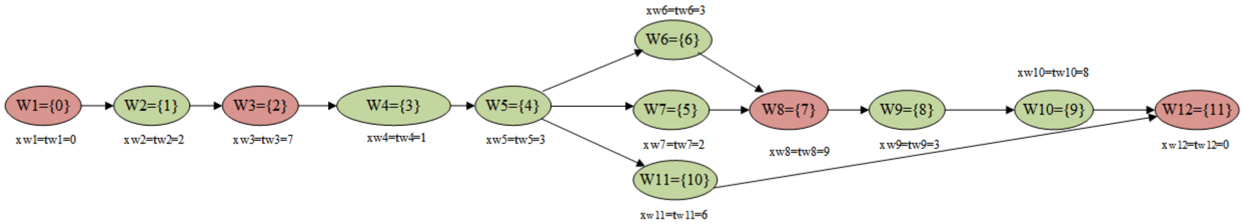

Figure 3: The work packages in the environment with 10 units of resource.

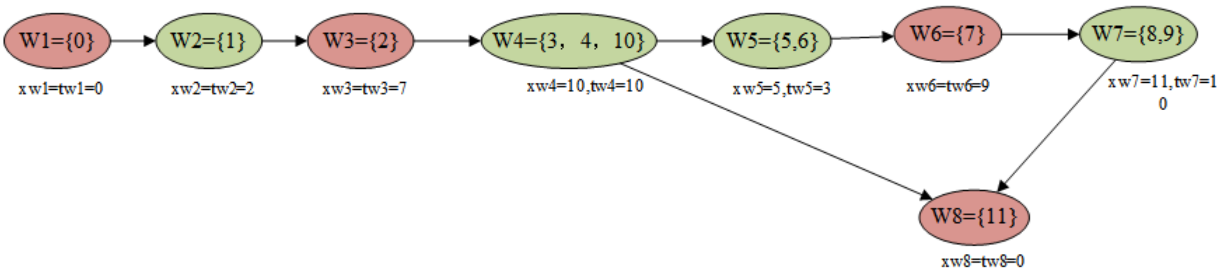

Figure 4: The work packages in the environment with 20 units of resource.

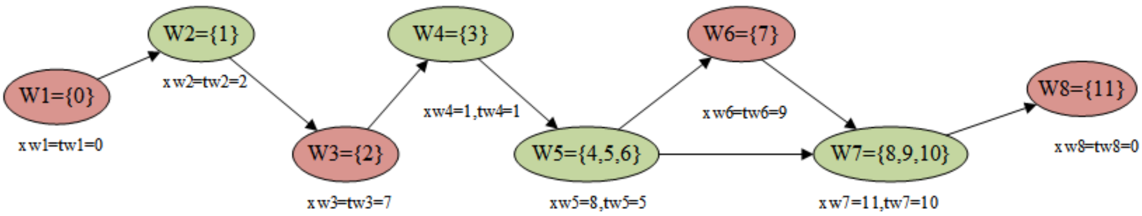

Figure 5: The work packages in the environment with 30 units of resource.

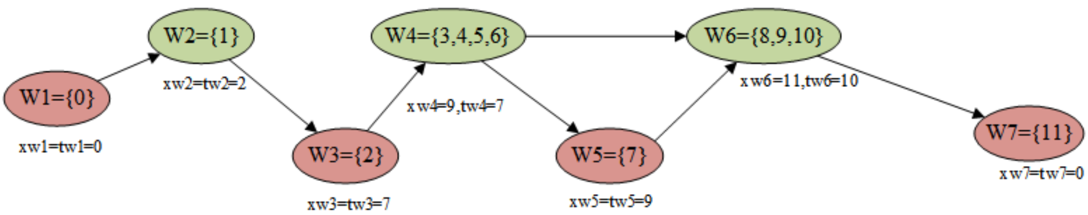

Figure 6: The work packages in the environment with 50 units of resource.

Table 2: *Solutions of the nine projects via Gurobi*

Indicator	Group 1			Group 2			Group 3		
	P 1	P 2	P 3	P 4	P 5	P 6	P 7	P 8	P 9
Node	10	10	10	20	20	20	30	30	30
$Node_A$	8	8	8	14	14	14	21	21	21
$Node_I$	2	2	2	6	6	6	9	9	9
Runtime	131	520	473	15007	24428	23674	98078	107645	188763
Packages	8	8	8	11	12	12	18	17	18
Makespan	41	39	30	75	88	56	105	89	98
Objective	317.31	325.29	304.84	516.16	579.00	532.21	815.70	869.14	765.43

Sensitivity Analysis

Given the significant impact of resource availability on the work package sizing and scheduling process in our problem, we conducted a sensitivity analysis to further investigate this relationship. During the sensitivity analysis, we systematically varied the resource availability levels with project 1 in group 1 (with resource {10,10,10,10}, {20,20,20,20}, {30,30,30,30} and {50,50,50,50}) and observed the corresponding changes in work package sizes. Figure 2 illustrates the initial structure of the tasks, while Figure 3 shows the work package structure under an environment with 10 units of each type of resource. Due to the limited resource availability, it is not feasible to group tasks into a single package. As a result, the work package structure remains the same as the initial one.

As the resource availability increases, more tasks can be grouped into packages. In Figure 4 and Figure 5, where there are 20 and 30 units of resources respectively, we observe the presence of 8 packages, with each package accommodating a maximum of 3 tasks. Figure 6 demonstrates a scenario where resource availability is no longer a constraint. This allows tasks to be packaged as long as the precedence constraints are satisfied. As a result, the number of packages is minimized. Overall, these figures highlight the impact of resource availability on the work package structure. With limited resources, fewer tasks can be grouped together, resulting in a larger number of packages. However, as resource availability increases, more tasks can be efficiently packaged, leading to a decrease in the number of packages.

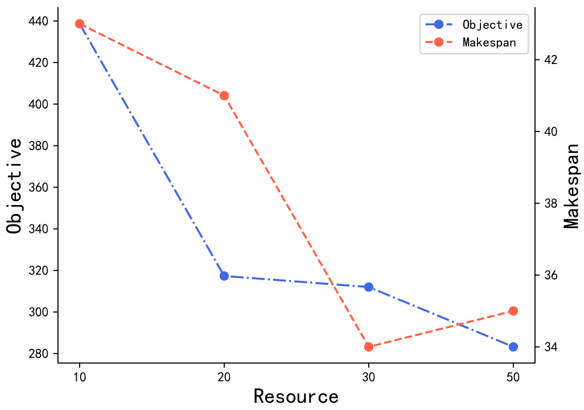

Figure 7: The comparisons of the makespan and objection.

Figure 7 presents a line graph showcasing the relationship between the objective function and makespan as the resource availability varies. As the resource availability increases from 10 to 30, the objective function shows a declining trend, decreasing from 440 to 290. This indicates that with more resources allocated, the total project cost decreases, reflecting improved efficiency and cost-effectiveness. However, when the resource availability reaches 50, the objective function starts to increase, reaching a value of 305. This suggests that excessive resource allocation beyond a certain point can lead to resource waste and increased project costs. Therefore, it is crucial to determine the appropriate level of resource availability that strikes a balance between cost reduction and resource utilization. In contrast, the makespan shows a declining trend as resource availability increases. This means that with more resources, the project duration or completion time decreases. This aligns with the intuition that additional resources enable tasks to be completed more quickly, resulting in a shorter overall project duration.

Conclusions

This paper presents an integrated model that addresses the challenges of work package sizing and scheduling in resource-constrained construction projects. To validate the effectiveness of our model, we utilized Gurobi, a powerful optimization solver, to search for optimal solutions, which demonstrates our model's effectiveness. Furthermore, we conducted an analysis to investigate the influence of resource availability on the work package sizing and scheduling process. By considering different resource levels, we find that proper resources are crucial for reducing the project's cost. However, solving our problem using Gurobi is time-consuming and not suitable for practical applications. In the future, we plan to develop algorithms that can efficiently solve our problem.

Acknowledgments

The work described in this paper is supported by grants from the Research Grants Council of the Hong Kong SAR of China, the National Natural Science Foundation of China, The University of Hong Kong (RGC Project No.15219422 G-HKU502/22, NSFC Project No. 72201228, HKU Project No. 2201100548 109000053)

References

Balouka, N. and Cohen, I. (2021). A robust optimiza-tion approach for the multi-mode resource-constrained project scheduling problem. European Journal of Oper-ational Research, 291(2):457–470.

Blazewicz, J., Lenstra, J. K., and Kan, A. R. (1983). Scheduling subject to resource constraints: classification and complexity. Discrete applied mathematics, 5(1):11–24.

Demeulemeester, E., Vanhoucke, M., and Herroelen, W. (2003). Rangen: A random network generator for activity-on-the-node networks. Journal of scheduling, 6:17–38.

Du, B., Tan, T., Guo, J., Li, Y., and Guo, S. (2021). Energy-cost-aware resource-constrained project scheduling for complex product system with activity splitting and recombining. Expert Systems with Applications, 173:114754.

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

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

Lambrechts, O., Demeulemeester, E., and Herroelen, W. (2008). Proactive and reactive strategies for resource-constrained project scheduling with uncertain resource availabilities. Journal of scheduling, 11(2):121–136.

Li, C.-L. and Hall, N. G. (2019). Work package sizing and project performance. Operations Research, 67(1):123–142.

Li, C. Z., Xu, X., Shen, G. Q., Fan, C., Li, X., and Hong, J. (2018). A model for simulating schedule risks in prefabrication housing production: A case study of six-day cycle assembly activities in hong kong. Journal of cleaner production, 185:366–381.

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

Li, H., Zhang, X., Sun, J., and Dong, X. (2023). Dynamic resource levelling in projects under uncertainty. International Journal of Production Research, 61(1):198–218.

Li, X., Chi, H.-l., Lu, W., Xue, F., Zeng, J., and Li, C. Z. (2021b). Federated transfer learning enabled smart work packaging for preserving personal image information of construction worker. Automation in Construction, 128:103738.

Li, X., Wu, C., Xue, F., Yang, Z., Lou, J., and Lu, W. (2022). Ontology-based mapping approach for automatic work packaging in modular construction. Automation in Construction, 134:104083.

Liu, Z., Li, X., Wu, C., Ma, J., Yang, Z., and Guo, Y. (2023). Automatic work package sizing for cost-effective modular construction. Automation in Construction, 154:105003.

Pellerin, R., Perrier, N., and Berthaut, F. (2020). A survey of hybrid metaheuristics for the resource-constrained project scheduling problem. European Journal of Operational Research, 280(2):395–416.

Tsz Wai, C., Wai Yi, P., Ibrahim Olanrewaju, O., Abdelmageed, S., Hussein, M., Tariq, S., and Zayed, T. (2023). A critical analysis of benefits and challenges of implementing modular integrated construction. International Journal of Construction Management, 23(4):656–668.

Vanhoucke, M., Coelho, J., Debels, D., Maenhout, B., and Tavares, L. V. (2008). An evaluation of the adequacy of project network generators with systematically sampled networks. European Journal of Operational Research, 187(2):511–524.

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

Zhang, Y., Li, X., Teng, Y., Shen, G. Q., and Bai, S. (2024). A heuristic rule adaptive selection approach for multi-work package project scheduling problem. Expert Systems with Applications, 238:122092.