International Journal of
Information
and
Management
Sciences

International Journal of Information and Management Sciences
30 (2019), 143-167. DOI:10.6186/IJIMS.201906_30(2).0004

Solving Multi-Mode Resource-Constrained Multi-Project Scheduling Problem with Combinatorial Auction Mechanisms

Chi-Bin Cheng, Chiao-Yu Lo and Chih-Ping Chu

Tamkang University

Abstract

This study solves a multi-project, multi-mode, and resource-constrained project scheduling problem. Multi-mode means that the activities in a project can be accomplished in one out of several execution modes, each of which represents an alternative combination of resource requirement of the activity. The present study considers the case that the resources need to be allocated first to individual projects by the upper-level manager, and then the project manager of each project schedules the project to optimize its outcome. In view of such a hierarchical decision-making structure, this study uses bi-level decentralized programming to model the problem. The proposed solution procedure employs combinatorial auction mechanisms to determine resource allocations to projects. A regular combinatorial auction and a fuzzy combinatorial auction are used, respectively, for cases of hard and soft capacity constraints. The proposed solution procedure is evaluated by comparison with the results reported in the literature.

Keywords: Multi-project scheduling, resource dedication, bi-level decentralized programming, combinatorial optimization, fuzzy logic.

1. Introduction

As market competiveness becomes even more furious, the necessity for firms to simultaneously conduct multiple projects under scarce resources is more and more critical (see Can and Ulusoy [14]). A report by Economist Intelligence Unit showed that 80having project management as a core competency helped them remain competitive during recession (see Economist Intelligence Unit [21]). In particular, parallel to the global expansion of the IT sector and the growth of R& D and engineering service activities, project management is increasingly used as a management paradigm (see Can and Ulusoy [14]).

Resource allocation among projects is one of the most important issues in the multi-project management setting. In the literature, approaches to solving the multi-project scheduling problem generally assume a resource-sharing policy among all projects to form a shared pool of each resource, with the exception of the research by Besikei et al. [11], which will be discussed later in the literature review. The assumption of resource sharing cannot be applied to all multi-project environments due to various reasons (see Beşikci et al. [12]) as follows.

(1) Geographical limitations, e.g., projects are distributed across the world.

(2) Project characteristics may not allow resource sharing; e.g., the development process is highly technology-intensive.

(3) Resource characteristics may not allow resource sharing; e.g., it is not desirable to allocate software developers to different projects for learning curve concern.

(4) Resource sharing is too costly; e.g., moving heavy machinery equipment is uneconomic. A typical scenario is the tower crane used at a construction site; it is not economic to move and set up the crane between two different sites, and thus, the crane can be used for other projects only after the current project completes the use of the crane.

When resource sharing is allowed, the multi-project scheduling problem can be represented by a single project network formed by combining the networks of all individual projects, and available single-project solution methods can be employed to solve the problem. However, when the assumption of resource sharing is not applicable, the dedication of resources to individual projects is required to optimize the overall multi-project scheduling performance.

Project management includes activities ranging from the tactical level, such as determination of due dates and resource allocation, to the operational level, such as the scheduling of specific activities (see Speranza and Vercellis [51]). Such a decision-making process in the multi-project environment is often considered having a dual-level management structure (see Yang and Sum [63]), which contains an upper-level manager and a team of lower-level project managers. The upper-level manager is responsible for the management of all projects, while the project managers are responsible for scheduling the activities of individual projects. It is further noted that this decision-making process also involves sequential and interactive decisions. For example, individual project managers attempt to optimize the outcomes of their projects after the upper-level manger determine the resources allocated to each project; on the other hand, the expected outcome of each project would motivate the upper-level manager to reconsider his previous decision and make necessary changes. The present study uses bi-level decentralized programming to model the dual-level structure, as well as the sequential and interactive decision-making process of the problem. Bi-level programming is closely related to leader-follower (Stackelberg) games in economics. It describes the hierarchical decision structure in many real-world situations, such as strategic planning (see Bracken and McGill [14]), resource allocation (see Aiyoshi and Shimizu [4]), and water management (see Anandalingam and Apprey [6]). Bi-level programming is categorized as non-convex programming and proven to be NP-hard by Ben-Ayed and Blair [10]. Bi-level problems share the following features (see Shih et al. [48] and Wen and Hsu [60]): (1) interactive decision-making units exist within a predominantly hierarchical structure; (2) execution of decisions is sequential, from the top to a lower level; (3) each decision unit independently optimizes its own benefits, but is affected by the actions of other units; and (4) the external effect on a decision maker's problem can be reflected in both the objective function and the set of constraints.

This study also considers that activities in a project can be executed by alternative modes which are pre-determined recipes of resource usages with their corresponding activity durations. The scenario of the executable modes can be illustrated by a software development project which involves three types of manpower, designers, developers, and testers. A software module can be completed with different combinations of the three types of skilled staff, and each combination is referred to as a mode. Apparently, different combinations of these resources lead to different completion times, and the resource mode chosen for a certain software module would constraint the mode choice by other software modules.

Each execution mode represents a combination of resource requirements of the activity. The excess or the deficit of a constituent resource in an execution mode is undesirable; the excess denotes a resource waste, while the deficit of a certain resource in the mode fails to carry out the activity. Such a characteristic of multi-mode project scheduling coincides with the concept of combinatorial auction (CA), where bidders are allowed to submit bids on combinations of items. CA was proposed as early as 1976 (see Jackson [27]) for radio spectrum rights, and later Rassenti et al. [43] proposed such auctions to allocate airport time slots (see de Vries and Vohra [20]). With the combinational bids, CA enables bidders to express complementarities between items, and thus, the bidder does not have to speculate on an item's valuation under the risk of not getting other complementary items (see Sandholm [46]).

The bi-level decentralized formulation proposed by this study facilitates the use of CA in solving the problem. Each project manager bids for the desirable resource modes that are potentially to improve the outcome of the project, while the upper-level manager acts as the auctioneer to determine which resource combinations are granted to which project under the overall resource capacity constraint. An algorithm is formulated in this study to execute the CA mechanism iteratively to gradually improve the solution. In this algorithm, a bid generation procedure that only finds and retains most beneficial bids constrains the number of bids involved in the CA mechanism and hence alleviates the computational complexity of the CA problem. In particular, each bidder (i.e. project manager) also explores solutions that are slightly inferior to the optimum of his project outcome to increase the chance of winning the auction, and the auctioneer (i.e. upper-level manager) will also slack off the project completion time demands a little if the auction fails in the previous round. This solution strategy renders a chance for the auctioneer not only to find solutions that satisfy the capacity constraints but also obtain an acceptable overall performance. This study further suggests a fuzzy resource constraint in case failing to find a feasible project schedule in reasonable time. The introduction of fuzzy capacity enable the evaluation of the tradeoff between the completion time and the capacity expansion.

From the above discussion, the problem concerned in this study is a multi-mode resource-constrained multi-project scheduling problem (MRCMPSP) without resource sharing. Other characteristics of the multi-project environment considered in this study are as follows. All projects are assumed to start initially, and preemption of activity is not allowed. Following Slowinski [49], three types of resources are considered, namely, renewable, non-renewable, and doubly constrained. Renewable resources are available in limited quantities during each time period. A typical example of such a resource is skilled labor, where the labor hours available to work on the project each day is limited but the resource is renewed each day to a predetermined level. Non-renewable resources are limited for the entire project; money is the best example of such a resource. Doubly constrained resources are limited both periodically and for the entire project. This type of resources is not explicitly considered in our model, since it can be incorporated by treating it as a renewable as well as a non-renewable resource.

The remainder of this paper is organized as follows. Section 2 provides a literature review on the field of project scheduling. Presented in Section 3 is the MRCMPSP modeled as a bi-level decentralized programming problem. Section 4 presents the solution procedure which is formulated according to a regular auction mechanism and a fuzzy combinatorial auction mechanism. Computational comparison is carried out in Section 5 based on the problem instances provided by Besikei et al. [11]. Finally, conclusions are given in Section 6.

2. Literature Review

The MRCMPSP contains a set of single-project multi-mode resource-constrained project scheduling problems (MRCPSP) as its sub-problems. It is convenient to start the discussion with the MRCPSP.

2.1. Single-project MRCPSP

The MRCPSP is a NP-hard problem, and when there are at least two renewable resources, the problem of finding a feasible solution is already NP-complete (see Kolisch and Drexl [29]). Exact approaches to solving MRCPSP were first proposed by Talbot [54], followed by Sprecher et al. [53], Hartmann and Drexl [26], and Sprecher and Drexl [52]. Heuristic methods have been proposed to find near-optimum solutions in reducing computation time, such as Slowinski et al. [50]. Meta-heuristics, especially genetic algorithms, were widely used to solve the problem (see Mori and Tseng [39] and Alcaraz et al. [5]). Table 1 provides a classification and summary of literature for solving the multi-mode single project scheduling problem, in which the methods classified into exact method, heuristics, and meta-heuristics.

The MRCPSP model embedded in the MRCMPSP of this study is formulated according to Talbot [54]. Talbot [54] formulated MRCPSP as an integer programming problem with two alternative objectives; one is to minimize the project completion time, and another one emphasizes a monetary objective that minimize the usage of resources under an arbitrary long due date.

Table 1: Literature for solving single project with multiple modes (MRCPSP).

Approach type	Algorithms	Literature
Exact method	Two-stage solution	Talbot [54]
	Branch-and-bound	Sprecher et al. [53]
		Sprecher and Drexl [52]
		Hartmann and Drexl [26]
Meta-heuristics	Genetic algorithms	Mori and Tseng [39], Hartmann [25], Alcaraz et al. [5], Barrios et al [8], Bagherinejad and Majd [7]
	Hybrid GA	Lova et al. [36], Elloumi and Fortemps [22], Van Peteghem and Vanhoucke [57], Vartouni and Khanli [58]
	Particle swarm optimization	Chen and Sandnes [16]
Heuristics	Decision support system	Slowinski et al. [50]
	Empirical hardiness model	Messelis and Causmaecker [38]

2.2. Multi-project scheduling

One of the earliest studies on multi-project scheduling problems is the integer programming formulation by Pritsker et al. [42], in which important aspects of multi-project scheduling have been covered, such as renewable resource constraints and multiple modes of resource usages. Kurtulus and Narula [32] examined the tardiness cost performance of 10 scheduling rules for a multi-project scheduling problem. Multiple modes of resource usage were not considered in their study. Rule-based scheduling approaches were also proposed by Tsubakitani and Decro [55] and Lawrence and Morton [34].

Lova et al. [35] developed a two-phase multi-criteria heuristic to improve resource allocation in multi-project scheduling. In the first phase, it obtains a feasible schedule with a time criterion, minimizing the mean project delay or the multi-project duration. In the second phase, it improves the first-phase schedule with non-time criteria, such as project splitting, in-process inventory, resource leveling, or idle resources. Meta-heuristics such as genetic algorithm were also employed to solve the multi-project scheduling problems in some studies, e.g., Kim et al. [28], Goncalves et al. [24], and Lova et al. [36].

Dual-level management structure is a popular concept in multi-project scheduling literature; such a structure also facilitates the decomposition of the original problem. Yang and Sum [63] presented the dual-level project management where the project managers at the operational (lower) level are in charge of scheduling and controlling individual project activities, while the upper-level manager works on a more tactical level and is responsible for all projects. The upper-level manager schedules all projects as individual entities to generate the start times and due dates for each project. According to these start times and due dates, each project manager schedules his project activities according to the resource capacity imposed by the upper-level manager. Sperenza and Vercellis [51] also proposed a two-stage approach to multi-project scheduling which decomposes the problem into a hierarchy of integer programming models to reflect the dual-level project management structure. Yang and Sum [64] followed their prior work Yang and Sum [63] and examined the performance of due date, resource allocation, project release, and activity scheduling rules in a multi-project environment. Can and Ulusoy [15] have recently developed a two-stage decomposition approach from the concepts of macro-activity and macro-mode introduced by Sperenza and Vercellis [51] to solve the dual-level multi-project scheduling problem. In the first stage, each project is reduced to a macro-activity with macro-modes and the multi-project scheduling is solved as a single project problem. In the second stage, individual project is scheduled to minimize its makespan using the start time and resource profile obtained in the first stage. Table 2 classifies and summarizes the literature for solving the multiple project scheduling problems with respect to single/multiple modes, methods used, and resource sharing.

Table 2: Literature for solving multi-project scheduling problems.

Mode	Approach Type	Literature	Resource sharing
Single mode	Exact method	Speranza and Vercellis [51]	Yes
	Meta-heuristics	Kim et al. [28]	Yes
		Pérez et al. [41]	Yes
	Heuristics	Tsubakitani and Decro [55], Lawrence and Morton [34], Yang and Sum [62][63]	Yes
		Lova et al. [35]	Yes
		Chien et al. [19]	Yes
		Zheng et al. [64]	Yes/No*
		Wauters et al. [59]	Yes/No*
Multi-mode	Exact method	Pritsker et al. [42]	Yes
	Meta-heuristics	Goncalves et al. [24]	Yes
		Lova et al. [36]	Yes
		Besikei et al. [11][13]	No
		Can and Ulusoy [15]	Yes
	Heuristics	Kurtulus and Narula [32], Tseng [56]	Yes

*: Containing both sharable and non-sharable resources.

The literature discussed above generally assumed that resources are shared among projects. However, this assumption can be invalid when resource sharing is not allowed or preferred (see Besikei et al. [11, 13]). Zheng et al. [65] and Wauters et al. [60] considered both sharable and non-sharable resources, where global resources can be shared among projects while local resources can be used by individual projects only. Besikei et al. [11, 13] particularly considered the case of unshared resources in the multi-project environment as a resource dedication problem. Resources are first dedicated to individual projects. Once resources are dedicated, they are no longer allowed to be shared with other projects; the multi-project problem then becomes solving separately many MR-CPSPs with their dedicated resources. A genetic algorithm with a local improvement heuristic (referred to as a combinatorial auction by the authors) is developed to solve the problem. In the combinatorial auction heuristic of Besikei et al. [11], the preferences of projects for resources are calculated with a modified Lagrangian relaxation formulation of the single project scheduling problem. After a short episode of sub-gradient optimization, values of the Lagrangian coefficients are taken as the bids of projects for resources. The unused resources in the current solution are then distributed to projects according to these bids by means of a continuous knapsack model to achieve a more preferable resource dedication state, and then a new schedule for each project is calculated. The overall combinatorial search is carried out by the genetic algorithm, in which the above-mentioned bidding heuristic is used as a local improvement procedure.

Inspired by Besikei et al. [11], the present study also focuses on the case of non-shareable resources in a multi-project environment. This study proposes a novel formulation of the problem as a bi-level decentralized programming (BLDP), which facilitate the interactions between the higher manager and the project managers. We propose an interactive algorithm based on the concept of combinatorial auction to solve the problem. Unlike previous approaches using either exact methods or (meta-)heuristics to solve the problem, our algorithm integrates both exact methods and heuristics in the solution procedure. Moreover, we incorporate fuzzy constraints in our model to ease the capacity constraints and explore potential solutions to the problem.

3. Modeling of MRCMPSP

This study considers a MRCMPSP where resource sharing among projects is not allowed. The concept of dual-level management structure is also adopted in this study. The upper-level manager determines the assignments of resources and the due dates to all projects; project managers at the lower level in turn strive to optimize their project outcomes with the resources under due dates assigned by the upper-level manager. Such a decision-making process is not one-way but rather interactive. In other words, the expected outcome of each project would make the upper-level manager reconsider his earlier decision and come up with alternatives if necessary. This interactive decision-making process continues until a satisfactory solution is obtained and is characterized as a bi-level decentralized programming problem (BLDP) in this study.

The BLDP belongs to the domain of distributed decision making problem, which addresses an important and rapidly developing field in decision science and comprises many diverse areas and disciplines. Schneeweiss [47] presented a unified framework to identify the overlap among different areas so as to take advantage of synergies, and to discover and understand general principles of distributed decision making. Schneeweiss [47] classified the domain of distributed decision making with respect to weak/strong hierarchical characters and single/multiple decision-maker situations, and introduced the basic concepts and theories as well as general applications including production planning/design/implementation, supply chain management, service operations and costs, and leadership problems. In particular, Adhau et al. [2, 3] and Adhau and Mittal [1] proposed distributed multi-agent systems with auctions based negotiation to resolve the conflict and allocation of shared resources among multiple competing projects; however, their approaches considered only single mode resource usage.

The upper-level decision in our BLDP is a resource allocation problem, where the manager assigns the amounts of resources to individual projects under the constraints of resource capacities. Each lower-level project manager then schedules the activities and the usages of resources, attempting to optimize the outcome of the project. Minimization of the weighted total tardiness of all projects is one of the most adopted objectives both in the literature and in actual practice (see Besikei et al. [11, 13]), and is used as the objective of the upper-level manager in the proposed BLDP model. Resource allocation and due date assignment to individual projects are decisions of the upper-level manager. These two decisions are in fact interdependent, i.e., the completion time of a project may be improved if more resources are allocated to the project. Resource allocation decision is much more complicated than the due date decision since it generally involves the combinations of many different resources. The present study proposes treating due date assignment as the decision at the upper level and leaving resource demand as the objective of individual projects. In other words, each project manager minimizes the resource usage under the due date request of the upper-level manager, and uses the resulting solution as a suggestion to the upper-level manager for evaluation. This study formulates the lower-level problem based on the MRCPSP formulation by Talbot [54] with a monetary objective. The monetary objective can avoid solutions with extra resources when searching feasible solution with the due-date given by the upper-level, and as such, all the monetary conversion factors are set to 1.

The proposed BLDP model for MRCMPSP is presented as follows.

Upper-level decision variables

δ_p : due date assigned to project p.

Upper-level parameters

V : Total number of projects.

D_p : due date for project p.

ν_p : weight of project p.

Y_k : total capacity of the k-th renewable resource.

Z_i : total capacity of the i-th nonrenewable resource.

Lower-level decision variables

x_{jtm} : a binary variable, equals 1 if activity j is completed by mode m at time period t; and 0 otherwise.

s_{kp} : the amount of the k-th renewable resource dedicated to project p.

c_{ip} : the amount of the i-th nonrenewable resource dedicated to project p.

Lower-level parameters

γ_k : monetary conversion factor of resources k.

E_j : earliest finish time of activity j.

L_j : latest finish time of activity j.

R_{kt} : capacity of renewable resource k at time period t.

r_{jkm} : usage (consumption) of renewable resource k by activity j, operating on mode m.

W_i : capacity of nonrenewable resource i.

w_{jim} : usage (consumption) of nonrenewable resource i by activity j, operating on mode m.

d_{jm} : duration of activity j by mode m.

N : total number of activities in the project.

M_j : total number of resource modes for activity j.

K : total number of renewable resources.

I : total number of nonrenewable resources.

H : planning horizon.

P : the set of all pairs of activities with precedence relationships; $(a, b) \in P$ indicates that activity a precedes activity b.

$$\text{Minimize} \qquad \sum_{p=1}^{V} \nu_p(\delta_p - D_p)^+ \tag{3.1}$$

$$\text{Subject to: :} \quad \sum_{p=1}^{V} s_{kp} \le Y_k, \qquad k = 1, \ldots, K \tag{3.2}$$

$$\sum_{p=1}^{V} c_{ip} \le Z_i, \qquad i = 1, \ldots, I \tag{3.3}$$

$$\delta_p \ge 0 \quad \text{are integers,}$$

where s_{kp} and c_{ip} solve the p-th lower-level MRCPSP:

$$\text{Minimize} \qquad \sum_{k=1}^{K} \gamma_k s_{kp} + \sum_{i=1}^{I} \gamma_i c_{ip} \tag{3.4}$$

$$\text{Subject to: :} \quad \sum_{m=1}^{M_j} \sum_{t=E_j}^{L_j} x_{jtm} = 1, \quad j = 1, \ldots, N \tag{3.5}$$

$$-\sum_{m=1}^{M_a} \sum_{t=E_a}^{L_a} t x_{atm} + \sum_{m=1}^{M_b} \sum_{t=E_b}^{L_b} (t - d_{bm}) x_{blm} \ge 0, \text{ for all } (a, b) \in P \tag{3.6}$$

$$R_{kt}^0 = \sum_{j=1}^{N} \sum_{m=1}^{M_j} \sum_{q=t}^{t+d_{jm}-1} r_{jkm} x_{jqm}, \qquad k = 1, \ldots, K,\ t = 1, \ldots, H \tag{3.7}$$

$$s_{kp} \geq R_{kt}^0, \qquad k = 1, \ldots, K,\ t = 1, \ldots, H \tag{3.8}$$

$$c_{ip} = \sum_{j=1}^{N} \sum_{m=1}^{M_j} \sum_{t=E_j}^{L_j} w_{jim} x_{jim}, \qquad i = 1, \ldots, I \tag{3.9}$$

$$\sum_{m=1}^{M_N} \sum_{t=E_N}^{L_N} t x_{Ntm} \leq \delta_p \tag{3.10}$$

$$x_{jtm} \in \{0, 1\}, \qquad \forall\ j, t, m$$

$$s_{kp}, c_{ip} \geq 0 \quad \text{are integers,}$$

In the above BLDP formulation, the objective of the upper-level manager is to minimize the weighted total tardiness of all projects, i.e., objective function (3.1). The decision of the upper-level is to determine the due date δ_p for each project, which becomes the constraint for the project manager, i.e., constraint (3.10), to determine his optimum resource demand as described by the lower-level model. The objective function (3.4) expresses such resource usage decisions s_{kp} and c_{ip} to minimize the total monetary value of required resources. Constraint (3.5) ensures that each job is completed once by using one of the possible modes. Constraint (3.6) maintains the precedence relations between activities. The usage of a renewable resource may vary for different time periods; and the maximum amount among all periods is the required amount by the project, which is expressed by constraints (3.7) and (3.8), where R_{kt}^0 is the amount of resource k used in time t by project p. Constraint (3.9) is the usage of non-renewable resources. The total usages of resources of all projects must not exceed the capacity as expressed by constraints (3.2) and (3.3).

In this BLDP formulation, the interaction between the upper-level and the lower-level becomes the adjustment of due dates by the upper-level manager and the resource demands by project managers. This BLDP formulation for the MRCMPSP not only enforces the interaction between the upper-level and the lower-level managers, but also alleviates the difficulty of the upper-level manager in making the resource allocation decision.

4. Solution Procedure

The bi-level programming problem is usually solved by transforming the original problem into a regular single-objective problem by replacing the lower-level problems with their Karush-Kuhn-Tucker (KKT) optimality conditions. Such a transformation generally loses the interactive and distributed natures of the original problem, since the KKT optimality conditions for the lower-level problems restrict the upper-level's optimization search [33]. The transformation also merges the decentralized decisions to a single and large problem that makes the problem even more difficult to solve, considering each of the embedded MRCPSP is already NP-hard. In the present study, instead of employing the KKT transformation approach to solve BLDP, a bidding structure suggested by Cheng [17] is adopted. The solution structure of Cheng [17] solves the upper-level problem and each lower-level problem separately. The resulting lower-level solutions are treated as bids and submitted to the upper-level manager. The upper-level manager then determines the winning bids by solving an auction problem. This study adopts the same concept and proposes an algorithm based on the combinatorial auction mechanism to solve our multi-project scheduling problem. The proposed auction-based algorithm facilitate the interactions between two levels. Unlike the KKT transformation favoring the lower-level decision and restricting the upper-level optimality, the proposed algorithm allows the two levels to optimize their respective decisions during the interactive process. Our algorithm decomposes the lower-level problem to many individual sub-problems, which can be solved by an available solver for their small problem sizes. This decomposition enables the integration of heuristics and exact methods, and hence provides an efficient way to solve the BLDP.

4.1. Resource allocation by combinatorial auction

The mode to accomplish an activity is a combination of resources, and thus the resources required to complete a project is also a combination of the amounts of a set of resources. The ignorance of such a combinatorial nature of the MRCPSP would result in difficulties in searching desirable resource allocations to individual projects. To resolve this difficulty, the present study suggests using the combinatorial auction mechanism to determine resource allocation.

The winner determination problem in combinatorial auctions is an NP-complete problem (see Sandholm [45]), and thus the feasibility of applying combinatorial auctions to real-world problems had been debated (see McMillian [37]). To reduce the computational complexity of combinatorial auctions, Rothkopf et al. [44] and Park and Rothkopf [40] suggested limiting the number of bid combinations with some restriction strategies to make the winner determination problem computationally manageable. In 2008, the Federal Communications Commission (FCC) of USA formally adopted a combinatorial auction to sell the 700-MHz radio spectrum rights by utilizing the bid combination reduction approach of Rothkopf et al. [44] to constrain the number of bids, and obtained $19 billion revenue for the American government. Epstein et al. [23] also successfully implemented combinatorial auctions to help the Chile government improve the quality of school meal catering contract assignment. Their solution resulted in savings of around $40 million per year. Cheng et al. [18] also suggested a combinatorial auction mechanism to solve a television advertising time slot allocation problem.

When applying the combinatorial auction to BLDP, the upper-level manager acts as the auctioneer, while lower-level project managers are bidders. The lower-level decision is to find a set of preferred resource combinations as bids for submission to the upper-level manager by solving a bid formulation problem, while the upper-level decision is to determine resource allocation to projects by solving a winner determination problem according to the bids submitted by all project managers.

4.1.1. Bid formulation problem

In general, a mode that can accomplish the activity faster demands more resources. The set of optimum resource combinations are treated as bids submitted to the upper-level manager, where the completion time associated with each resource combination is considered as a bid price.

Assuming the maximum tolerable delay of a project by the upper-level manager is T, the upper-level manager can assign any due date that is within the tolerable tardiness to a project to explore resource usage combinations that are feasible to the capacity constraints. If an assigned due date is unable to render feasible resource usages, the upper-level manager would extend the due date with the expectation to find feasible solutions. In the proposed algorithm, project managers solve their MRCPSPs to find ideal resource usage plans and bid to the upper-level manager, while the upper-level manager gradually extends the due date by one time period if necessary. When the due date is extended, the bids generated in the previously assigned due dates together with the newly generated bid are submitted to the upper-level manager. When project managers solve the MRCPSP to find the ideal resource usage plans, not only the optimum solution but also a few inferior solutions to the optimum are obtained as well. The purpose is to increase the chance of satisfying the capacity constraints.

The number of solutions to explore is a choice of the upper-level manager. Solutions inferior to the optimum can be explored by solving the following MRCPSP.

MRCPSP-r:

$$\begin{aligned} \text{Minimize } & f(r) = (3.4) \\ \text{Subject to: } & (3.5)\text{-}(3.10),\ x_{jtm} \in \{0,1\},\ \forall j,t,m, s_{kp}, c_{ip} \geq 0 \text{ are integers} \\ & f(r) > f^*(r-1) \qquad (4.1) \end{aligned}$$

where $f^*(r-1)$ is the optimum objective obtained previously in solving the above MRCPSP the $(r-1)$-th times.

Table 3 presents the bid contents associated with different levels of tardiness of project p as discussed above. Let $B^*_{p\tau}(r) = [s^*_{1_{p\tau}}, \ldots, s^*_{K_{p\tau}}(r), c^*_{1_{p\tau}}, \ldots, c^*_{K_{p\tau}}(r)]$ denote the r-th order optimum solutions under τ tardiness, where $s^*_{kp\tau}(r)$ and $c^*_{ip\tau}(r)$ are the corresponding optimum resource usages, and let $\delta^*_{p\tau}$ be the associated completion time when the tardiness is τ, i.e., $\delta^*_{p\tau} = D_p + \tau$, $\tau = 0, \ldots, T$. Each bid is associated with a bid price; in this case, $\delta^*_{p\tau}$ is the bid price of $B^*_{p\tau}(r)$, $\forall r$. When τ is increased, the bids generated previously are added to the set of newly generated bids under current τ.

Table 3: Bids of resources by project p.

Tardiness(τ)	0	1	$\cdots$	T
Bid content ($b_{p\tau}$)	$b_{p0}=\{(B^*_{p0}(r),\delta^*_{p0}),\forall r\}$	$b_{p1}=\{(B^*_{p1}(r),\delta^*_{p1}),\forall r\}\cup b_0$	$\cdots$	$b_{pT}=\{(B^*_{pT}(r),\delta^*_{pT}),\forall r\}\cup b_{T-1}$

4.1.2. Winner determination problem

Lower-level project managers submit their bids under the current due dates assigned by the upper-level manager (or equivalently the tolerable tardiness notified by the upper-level manager). The upper-level manager then determines the winning bids according to all these submitted bids under the satisfaction of capacity constraints, in which each project is granted one and exactly one bid. Let $y_{p\tau r}$ denote the 0-1 decision variable that equals 1 if bid $(B^*_{p\tau}(r),\delta^*_{p\tau})$ is selected; and 0, otherwise. The upper-level winner determination problem is reformulated as:

$$\begin{aligned}
\text{Minimize}\ & \sum_{p=1}^{V}\sum_{\tau=0}^{T}\sum_{r=1}^{L}\nu_p(\delta^*_{pt}-D_p)y_{pbr} \\
& +\varphi\Big(\sum_{k=1}^{K}\sum_{p=1}^{V}\sum_{\tau=0}^{T}\sum_{r=1}^{L}\gamma_k s^*_{kp\tau}(r)y_{p\tau r}+\sum_{i=1}^{I}\sum_{p=1}^{V}\sum_{\tau=0}^{T}\sum_{r=1}^{L}\gamma_i c^*_{ip\tau}(r)y_{p\tau r}\Big) \quad (4.2) \\
\text{Subject to:}\ & \sum_{\tau=1}^{T}\sum_{r=0}^{L}y_{p\tau r}=1,\quad p=1,\ldots,V \quad (4.3) \\
& \sum_{p=1}^{V}\sum_{\tau=0}^{T}\sum_{r=1}^{L}s^*_{kp\tau}(r)y_{p\tau r}\le Y_k,\quad k=1,\ldots,K \quad (4.4) \\
& \sum_{p=1}^{V}\sum_{\tau=0}^{T}\sum_{r=1}^{L}c^*_{ip\tau}(r)y_{p\tau r}\le Z_i,\quad i=1,\ldots,I \quad (4.5) \\
& y_{p\tau r}\in\{0,1\},\ \forall p,\tau,r
\end{aligned}$$

where L is the repetition length of solving the MRCPSP. The second term in the objective function is added to exclude solutions that result in the same weighted total tardiness but use more resources, where φ is a very small constant to avoid the resource costs from dominating the objective function.

4.2. Fuzzy combinatorial auction

It is possible that the upper-level manager is unable to find a feasible set of winning bids even after the due dates of projects have been extended or when the manager is not satisfied with the resulting project completion times. Such a case implies the demand of additional resources to render feasible project schedules. Besikei et al. [11]

also pointed out the need of budget consideration for resource investment to expand the overall resource capacities. To enable the decision of capacity expansion, fuzzy capacities are considered in the formulation to allow the evaluation of possible capacity expansions; i.e., constraints (4.4) and (4.5) are replaced with:

$$\sum_{p=1}^{V}\sum_{\tau=0}^{T}\sum_{r=1}^{L} s^*_{kp\tau}(r)y_{p\tau r} \le \widetilde{Y}_k, \quad k=1,\dots,K \tag{4.6}$$

and

$$\sum_{p=1}^{V}\sum_{\tau=0}^{T}\sum_{r=1}^{L} c^*_{ip\tau}(r)y_{p\tau r} \le \widetilde{Z}_i, \quad i=1,\dots,I \tag{4.7}$$

respectively, where $\widetilde{Y}_k$ and $\widetilde{Z}_i$ are fuzzy capacities of the renewable and nonrenewable resources defined by fuzzy sets and are characterized by the membership functions depicted in Figure 1(a) and (b) respectively.

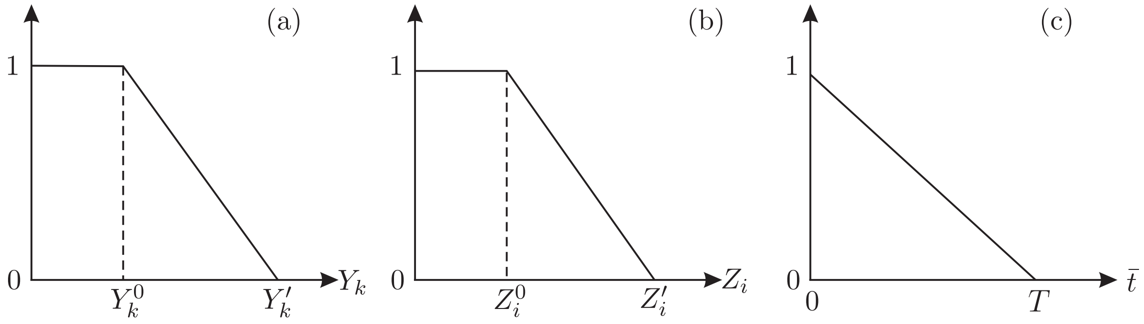

Figure 1: Membership functions of (a) renewable resource, (b) nonrenewable resource, and (c) tardiness.

In the membership function of Figure 1(a), when the usage of resource is less than Y_k^0, the upper-level DM has perfect satisfaction; this point is usually the current capacity. On the other hand, Y_k' indicates the maximum tolerable usage of the resource, which can be determined based on the available budget. Figure 1(b) describes the membership function of the nonrenewable resource in the same manner. The construction of the membership functions in Figure 1 is based on the decision-maker's experience and judgment.

The fuzzy combinatorial auction problem can be solved by the fuzzy approach of Werners [62]. The approach of Werners [62] is adopted to model the fuzzy capacity, which can be interpreted as the tradeoff between resource expansion and the cost. First, by the definition of the membership function in Figure 1(a), fuzzy constraint (4.6) can be rewritten as

$$\sum_{p=1}^{V}\sum_{\tau=0}^{T}\sum_{r=1}^{L} s^*_{kp\tau}(r)y_{p\tau r} \le Y_k^0 + (1-\mu_k)(Y_k' - Y_k^0), \qquad k=1,\dots,K \tag{4.8}$$

where μ_k is a membership value for indicating the degree of satisfaction of the upper-level manager regarding the expansion of resource k and $\mu_k \in [0,1]$. The right-hand side of the above constraint expresses that the increment of resource will reduce the degree of satisfaction. Similarly, according to the membership function in Figure 1(b), the fuzzy constraint (4.7) can be converted to

$$\sum_{p=1}^{V}\sum_{\tau=0}^{T}\sum_{r=1}^{L} c^*_{ip\tau}(r)y_{p\tau r} \leq Z_i^0 + (1-\nu_i)(Z_i' - Z_i^0), \qquad i = 1,\dots,I \tag{4.9}$$

with ν_i as the membership value to indicate the degree of satisfaction regarding the expansion of resource i and $\nu_i \in [0,1]$. The fuzzy optimization approach of Werners [61] is developed from the max-min fuzzy decision concept of Bellman and Zadeh [9]. Bellman and Zadeh [9] proposed that a fuzzy decision can be defined as a fuzzy set of alternatives resulting from the intersection of the objective and the constraints. The intersection of objective and constraints is obtained through the minimum operator. The optimum fuzzy decision is then obtained by maximizing the intersection of objective and constraints. This concept is referred to as a max-min approach. As a result, the fuzzy combinatorial auction problem can be rewritten as:

$$\text{Minimize} \quad \alpha - \varphi\Big(\sum_{k=1}^{K}\sum_{p=1}^{V}\sum_{\tau=0}^{T}\sum_{r=1}^{L}\gamma_k s^*_{kp\tau}(r)y_{p\tau r} + \sum_{i=1}^{I}\sum_{p=1}^{V}\sum_{\tau=0}^{T}\sum_{r=1}^{L}\gamma_i c^*_{ip\tau}(r)y_{p\tau r}\Big) \tag{4.10}$$

Subject to: (4.3), (4.8), (4.9)

$$\bar{t} = \frac{\sum\limits_{p=1}^{V}\sum\limits_{\tau=0}^{T}\sum\limits_{r=1}^{L}\nu_p(\delta^*_{p\tau} - D + p)y_{p\tau r}}{\sum\limits_{p=1}^{V}\nu_p} \tag{4.11}$$

$$\bar{t} \leq (1-\gamma)T \tag{4.12}$$

$$\alpha = \min_{\forall k,i}\{\mu_k, \nu_i, \gamma\} \tag{4.13}$$

$$y_{p\tau r} \in \{0,1\}, \ \forall p, \tau, r$$

Constraint (4.11) converts the weighted total tardiness into a weighted average tardiness; thus, the manager can express the preference in accordance with the membership function in Figure 1(c). Constraint (4.12) defines the satisfaction regarding the weighted tardiness according to the membership function in Figure 1(c), where the satisfaction (i.e. membership value γ) being 100% when the tardiness is 0 and the satisfaction reaches 0% when the tardiness exceeds the maximum tolerable delay T. Though the resource expansion explored by the above fuzzy combinatorial auction does not consider the overall resource budget explicitly, it provides additional information for the decision-maker to examine the possibility of performance improvement with a reasonable amount of resource expansion.

4.3. Algorithm

The flowchart of the solution procedure discussed above is depicted in Figure 2, and the detailed steps of the solution procedure are described in the algorithm below.

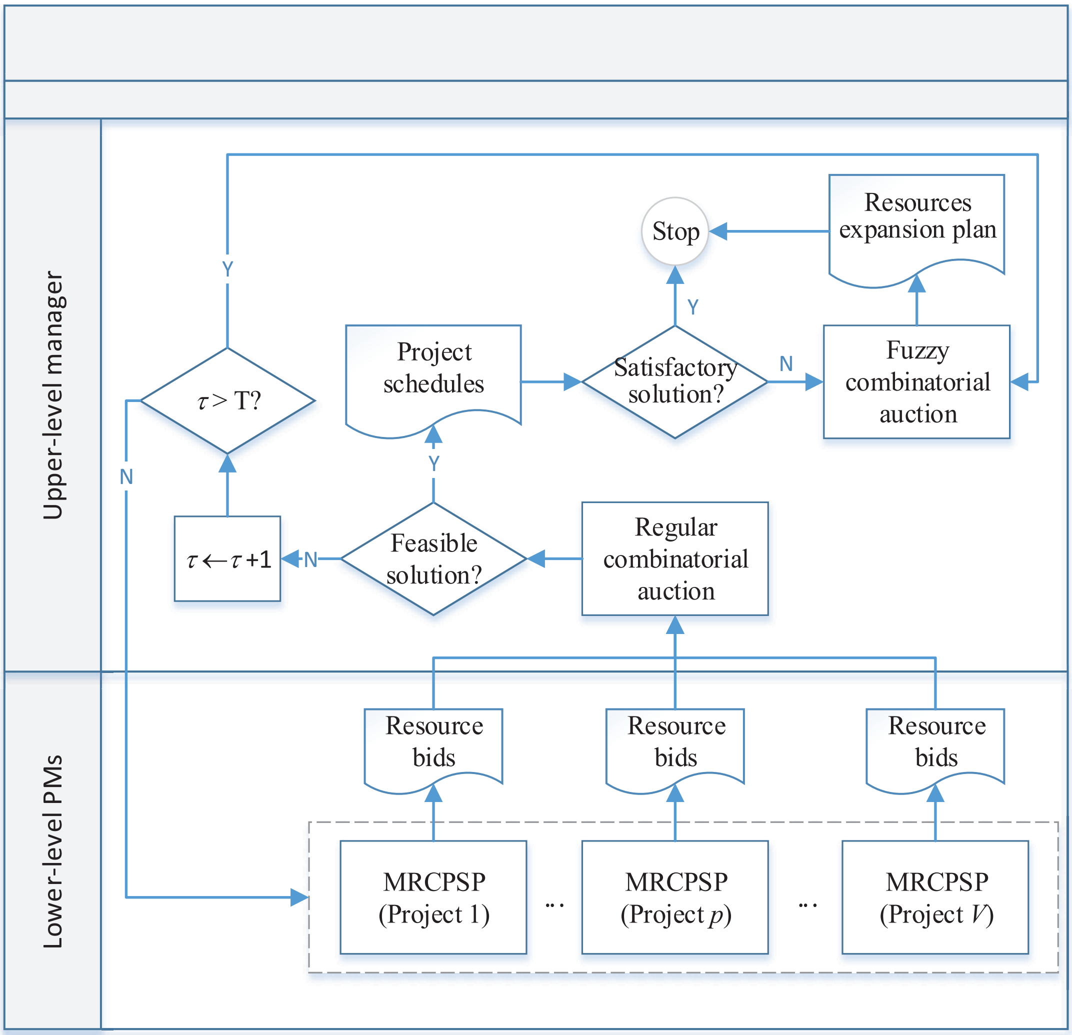

Figure 2: Flowchart of solution procedure.

Algorithm:

Step 0. Parameter settings of T, L, and the membership functions in Figure 1. Initialize $\tau = 0$

```
Step 1. For project $p = 1$ to $V$
            For $r = 1$ to $L$
                Solve MRCPSP: min{(3.4): (3.5)-(3.10) and (4.1), $x_{jtm} \in \{0, 1\}$, $\forall\, j, t, m$,
                $s_{kp}, c_{ip} \geq 0$ are integers}.
            Next $r$
            Submit bid set $b_{p\tau}$ to the upper-level manager.
        Next $p$
Step 2. Solve regular CA: min{(4.2): (4.3)-(4.5), $y_{p\tau r} \in \{0, 1\}$, $\forall\, p, \tau, r$}.
        If feasible, go to Step 4; otherwise go to Step 3.
Step 3. $\tau \leftarrow \tau + 1$ (increase the due date by one unit each time)
        If $\tau > T$, go to Step 5; otherwise, go to Step 1.
Step 4. If project completion times are satisfactory, go to Step 6; otherwise, go to Step
        5.
Step 5. Solve the fuzzy CA:
        max{(4.10): (4.3), (4.8), (4.9), (4.11)-(4.13), $y_{p\tau r} \in \{0, 1\}$, $\forall\, p, \tau, r$}.
        Obtain the resource expansion plan.
Step 6. Stop.
```

The parameters T and L set in the beginning of the algorithm together limit the maximum number of bids by a projector manager to be $L \times T$. With V projects in total, the upper-level manager solves a combinatorial auction problem with only $L \times T \times V$ bids in maximum. Such a mechanism permits the computation complexity of the combinatorial auction problem to grow moderately.

5. Computational Results

Beşikci et al. [11] created a set of problem instances of MRCMPSP based on the problem sets of Kolisch and Sprecher [31]. Problem instances are grouped according to their network complexity and maximum utilization factor. Network complexity is defined as the total number of arcs divided by the total number of nodes in the project network. Maximum utilization factor is the ratio defined as the resource requirement of no-delay schedule over the available capacity. When the ratio is less than or equal to 1, the project can be completed without delay. This study uses the problem instances of Beşikci et al. [11] and a benchmark problem dataset to evaluate the performance of the proposed approach.

5.1 Comparison with the approach of Beşikci et al. [11]

Two levels of network complexity, i.e., 1.4 and 1.8, and three levels of maximum utilization factor, i.e., 1.2, 1.4, and 1.5, are selected to construct a full factorial design with 10 problems in each combination. Each problem group comprises 10 problem instances, and each problem instance contains 6 projects with weights from 1 to 6, respectively;

each project consists of 22 or 32 activities with two renewable resources and two non-renewable resources. The due dates of projects are set following the approach of Beşikci et al. [11]. The due date of a project with the highest weight, i.e., 6, is set as its no-delay completion time without considering capacity constraints. As the weight decreases, the project is assigned a tighter due date as shown in Table 4. As can be seen, the minimum weighted tardiness possible for a problem instance is 35 according to the objective function (4.2). In the experiments, the maximum allowable tardiness T is set to be 10, and the repetition times (L) of solving the MRCPSP under each tardiness level is set to be 5. The monetary conversion factors γ_i, $\forall\, i$, are all set to be 1.

Table 4: Due dates assigned to projects with different weights.

Project	Weight	Due date	Tardiness
1	6	(No delay)	0
2	5	(No delay) - 1	1
3	4	(No delay) - 2	2
4	3	(No delay) - 3	3
5	2	(No delay) - 4	4
6	1	(No delay) - 5	5

The proposed solution procedure is programmed by JAVA with CPLEX library and is run on an Intel Core i5, 1.4Ghz processor with 4GB memory. Computational results are presented in Tables 5 and 6 and are compared with the results obtained by Beşikci et al. [11] which used an Intel Xeon X5492, 3.40 Ghz processor. AWT in the tables denotes the average weighted total tardiness and ART is the average run time (in minutes) for a problem instance, with "std" in the parentheses indicating their standard deviations. Beşikci et al. [11] compared the performances of four methods in their study, namely GA with Lagrangian relaxation (GA-LA), GA with linear relaxation, subgradient optimization, and exact method. Since the GA-LA outperformed the other three methods as reported by Beşikci et al. [11], this study compares the proposed algorithm with the GA-LA only.

In the problem group NC1.4-MUF1.2 (Table 5), 8 out of the 10 problem instances obtain the optima (i.e., weighted total tardiness = 35.0), and two instances close to the optimum (36 and 37 respectively), resulting in an AWT=35.3. This result is slightly inferior to those obtained by Beşikci et al. [11]; nevertheless, with a few expansion of the resources (3 units on average), the fuzzy combinatorial auction can obtain optima for all instances. The computation time of the fuzzy CA is negligible (usually a couple of seconds) and thus is not reported in the tables. The column *Extra resource* indicates the amount of additional resources needed to expand on the current capacities. For example, the average amount of resources needed to expand for the group NC1.4-MUF1.5 is 54.7, in which the resource requirements of the first problem instance in this group are presented in Table 7. In the example of Table 7, the resource usages suggested by the fuzzy CA for the renewable resources and the first non-renewable resource do not exceed their current capacities but improve the utilization of these resources; while the second non-renewable resource is suggested to expand by 50 units and is counted as the amount of extra resource demand for this case. Such results also provide the information about the insufficiency or abundance of a resource for the manager to adjust resource capacities.

Table 5: Comparison of results for problem instances containing 6 projects with 22 activities.

Problem	Beşikci et al. [11]		Regular combinatorial auction		Fuzzy combinatorial auction		
group	AWT	ART	AWT	ART		AWT	
(NC-MUF)			(std)	(std)	α	(std)	Extra resource
1.4-1.2	35.0	6.19	35.3	2.13	0.83	35.0	3.0
			(0.6)	(0.66)		(0.0)	
1.4-1.4	42.6	94.40	71.0	2.05	0.78	45.4	31.8
			(39.1)	(0.53)		(9.0)	
1.4-1.5	52.9	117.29	125.6	5.18	0.68	53.7	60.5
			(68.3)	(4.5)		(22.0)	
1.8-1.2	35.0	3.38	35.3	2.06	0.83	35.0	3.0
			(0.6)	(0.58)		(0.0)	
1.8-1.4	41.4	98.99	71.5	2.05	0.78	45.4	31.8
			(39.1)	(0.54)		(9.0)	
1.8-1.5	50.6	120.00	108.5	2.40	0.73	56.5	52.3
			(52.8)	(0.95)		(17.3)	

Table 6: Comparison of results for problem instances containing 6 projects with 32 activities.

Problem	Beşikci et al. [11]		Regular combinatorial auction		Fuzzy combinatorial auction		
group	AWT	ART	AWT	ART		AWT	
(NC-MUF)			(std)	(std)	α	(std)	Extra resource
1.4-1.2	35.0	14.81	35.3	11.24	0.83	35.0	11.0
			(0.9)	(8.43)		(0.0)	
1.4-1.4	56.0	74.07	57.6	10.72	0.79	43.1	27.6
			(23.6)	(8.22)		(8.1)	
1.4-1.5	181.0	110.08	100.6	22.94	0.70	48.6	58.3
			(51.9)	(17.46)		(13.6)	
1.8-1.2	35.0	8.73	35.3	10.78	0.83	35.0	11.0
			(0.9)	(8.63)		(0.0)	
1.8-1.4	82.2	88.45	57.6	13.84	0.79	43.1	27.6
			(23.6)	(10.75)		(8.1)	
1.8-1.5	148.0	110.15	100.9	9.85	0.75	52.4	48.5
			(44.9)	(7.42)		(14.9)	

As for the remainder problem groups in Table 5, the solutions found in this study are all inferior to those obtained by Beşikci et al. [11]; however, the computation times by our approach are much less than that by Beşikci et al. [11], and by further solving these problems with the fuzzy combinatorial auction, AWTs obtained by our regular CA are improved and approximate the solutions of Beşikci et al. [11]. A pairwise t-test on the AWTs in Table 5 between Beşikci et al. [11] and our regular combinatorial auction indicates that the approach of Beşikci et al. [11] is superior (with $t = -2.62$ and $p = 0.02$). When comparing between Beşikci et al. [11] and our fuzzy combinatorial auction, the difference is not significant (with $t = -2.29$ and $p = 0.07$).

Table 7: Resource demands solved by regular and fuzzy CAs (the first problem instance of group NC1.4-MUF1.5).

Resource		Capacity (A)	Resource usages by regular CA (B)	Resource usages by fuzzy CA (C)	Extra resources (C-A)
Renewable	1	118	104	116	-2
	2	135	131	128	-7
Non-renewable	1	456	407	434	-22
	2	530	529	580	50

Table 6 presents the computational results of problem groups with projects consisting of 32 activities. Except problem groups NC1.4-MUF1.2, NC1.4-MUF1.4, and NC1.8-MUF1.2, which are slightly inferior to the solutions by Beşikci et al. [11], the proposed algorithm outperforms on the remainder problem groups. Although the pairwise t-test does not evidence a significant difference on the AWT (with with $t = 1.84$ and $p = 0.06$), our algorithm dominates all problem groups on the computation time. Furthermore, by employing the fuzzy CA to solve the problems, AWTs are significantly improved and outperform those by Beşikci et al. [11] (with $t = 2.07$ and $p = 0.047$).

According to the computational results presented in Tables 5 and 6, it seems the proposed algorithm tends to perform better on harder problems. Problems with MUF values closer to 1 are regarded as relatively easy problems whereas problems with higher MUF values are harder ones (see Beşikci et al. [11]). The algorithm of Beşikci et al. [11] performed well on problems with lower dimension (i.e., 22 activities) and especially on problem groups whose problem structures commonly have MUF values equal to 1.2. However, for problems with higher dimension, i.e., problem instances with 32 activities, the proposed algorithm has better performance in general.

5.2. Performance evaluation based on a benchmark dataset

To evaluate the performance of our proposed approach on larger project networks, experiments on multiple projects with 52 activities for each project are also conducted. Problem instances are constructed based on the MMLIB dataset[1] by Van Peteghem and Vanhoucke [58], which was originally designed for single project MRCPSP. Five problem instances are tested, where each problem instance contains five projects, and each project

[1] MMLIB dataset can be downloaded at http://mmlib.eu/download.php.

Table 8: Description of problem instances containing 5 projects with 52 activities.

	Problem 1	Problem 2	Problem 3	Problem 4	Problem 5	Weights
Project ID in MMLIB (network complexity)	J502_1.mm (4.08)	J503_1.mm (4.19)	J504_1.mm (4.12)	J505_1.mm (4.63)	J506_1.mm (3.90)	5
	J502_2.mm (4.67)	J503_2.mm (3.96)	J504_2.mm (4.62)	J505_2.mm (4.15)	J506_2.mm (4.27)	4
	J502_3.mm (4.37)	J503_3.mm (4.15)	J504_3.mm (4.31)	J505_3.mm (4.40)	J506_3.mm (4.52)	3
	J502_4.mm (3.94)	J503_4.mm (4.00)	J504_4.mm (4.00)	J505_4.mm (4.87)	J506_4.mm (3.65)	2
	J502_5.mm (4.75)	J503_5.mm (4.38)	J504_5.mm (4.31)	J505_5.mm (4.23)	J506_5.mm (3.98)	1
Due date	23	18	19	17	20	
Resource capacity	R1: 153 R2: 147 N1: 1192 N2: 1244	R1: 151 R2: 138 N1: 665 N2: 686	R1: 146 R2: 149 N1: 1258 N2: 1382	R1: 144 R2: 137 N1: 939 N2: 1038	R1: 148 R2: 142 N1: 1461 N2: 1442	

R1: renewable resource 1; R2: renewable resource 2; N1: non-renewable resource 1; N2: non-renewable resource 2

Table 9: Computational results for problem instances containing 5 projects with 52 activities.

Problem Instance	Proposed approach		Lower bound
	AWT	Run-Time	AWT
1	126	6.30	27
2	74	6.03	23
3	113	3.45	14
4	66	8.33	32
5	172	3.98	21

has 52 activities and each activity uses three resource modes (except the source and the sink) with two renewable and two non-renewable resources. Descriptions of these problem instances are summarized in Table 8. The original IDs of the projects (in MMLIB) contained in each problem instance are given. The weights of projects are from 5 to 1 in each problem instance. The due date of each problem instance is determined based on the benchmark[2] provided by Van Peteghem and Vanhoucke [58] while solving the MRCPSP. The minimum of the completion times obtained by Van Peteghem and Vanhoucke [58] among each problem group (instance) is used as the due date of that problem instance. The resource capacity of each problem instance is formulated by taking the summation of the original resource capacities of all projects inside the problem. This experiment is run on a computer with Intel Core i5, 2.20Ghz processor and 12GB memory. The

[2] The benchmark can be found at http://mmlib.eu/solutions.php.

computational results are presented in Table 9, where the lower bound of each problem instance is computed based on the completion times of individual projects obtains by Van Peteghem and Vanhoucke [58]. It must be noted that such lower-bounds may be much lower than the true optima. The results in Table 9 show our approach performs well in terms of computational time for even larger problems, in which, the solution quality obtained for Problems 2 and 4 are better than the rest problems for their completion times are closer to their lower bounds.

6. Concluding Remarks

This study considers a multi-mode resource-constrained multi-project scheduling problem where resource sharing among projects is not allowed. The problem is modeled as a bi-level decentralized programming problem which contains an upper-level problem for due date assignment decision and a set of lower-level problems for individual project scheduling. The problem is solved using a regular combinatorial auction mechanism and a fuzzy combinatorial auction mechanism. The regular combinatorial auction is employed when only current resource capacities are considered, while the fuzzy combinatorial auction takes possible expansion of resources into account.

For performance evaluation, the problem instances created by Beşikci et al. [11] are adopted. Computational results show that the proposed algorithm outperforms the approach of Beşikci et al. [11] on problem instances with greater complexity in terms of both solution quality and computation time. The fuzzy combinatorial auction mechanism also provides information about the insufficiency or abundance of a resource for the manager to adjust resource capacities. To evaluate the performance of our proposed approach on larger project networks, experiments on multiple projects with 52 activities for each project are further conducted. Computational results show our approach performs well in terms of computational time for even larger problems.

Current studies of resource-constrained multi-project scheduling generally consider either full-shareable or non-shareable resources. However, there are cases that involve both situations. For example, the tower crane in construction projects is generally a non-shareable resource, but workers are easily to move and dynamically distribute between two construction sites and thus can be treated as a shareable resource. In our future study, the scheduling of resource-constrained multi-projects with both shareable and non-shareable resources will be investigated.

Acknowledgements

This work was supported by the Ministry of Science and Technology in Taiwan under Grant MOST103-2221-E-032-019. An initial work of this study was presented at the 3rd IEEE International Conference on Cybernetics, 2017. The authors are grateful for the audience's comments in improving this study.

References

[1] Adhau, S. and Mittal, M. L. (2012). *A Multiagent Based System for Resource Allocation and Scheduling of Distributed Projects*, International Journal of Modeling and Optimization, Vol.2, 524.

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

[3] Adhau, S., Mittal, M. L. and Mittal, A. (2013). *A multi-agent system for decentralized multi-project scheduling with resource transfers*, International Journal of Production Economics, Vol.146, 646-661.

[4] Aiyoshi, E. and Shimizu, K. (1981). *Hierarchical decentralized systems and its new solution by a barrier method*, IEEE Transactions on Systems, Man, and Cybernetics, Vol.11, 444-449.

[5] Alcaraz, J., Maroto, C. and Ruiz, R. (2003). *Solving the multi-mode resource-constrained project scheduling problem with genetic algorithms*, Journal of the Operational Research Society, Vol.54, 614-626.

[6] Anandalingam, G., and Apprey, V. (1991). *Multi-level programming and conflict resolution*, European Journal of Operational Research, Vol.51, 233-247.

[7] Bagherinejad, J. and Majd, Z. R. (2014). *Solving the MRCPSP/max with the objective of minimizing tardiness/earliness cost of activities with double genetic algorithms*, International Journal of Advanced Manufacturing Technology, Vol.70, 573-582.

[8] Barrios, A., Ballestin, F. and Valls, V. (2011). *A double genetic algorithm for the MRCPSP/max*, Computers & Operations Research, Vol.38, 33-43.

[9] Bellman, R. E. and Zadeh, L. A. (1970). *Decision-making in a fuzzy environment*, Management Science, Vol.17, B141-B164.

[10] Ben-Ayed, O. and Blair, C. E. (1990). *Computational difficulties of bilevel linear programming*, Operations Research, Vol.38, 556-560.

[11] Beşikci, U., Bilge, Ü. and Ulusoy, G. (2013). *Resource dedication problem in a multi-project environment*, Flexible Services and Manufacturing Journal, Vol.25, 206-229.

[12] Beşikci, U., Bilge, Ü. and Ulusoy, G. (2014). *Multi-mode resource constrained multi-project scheduling and resource portfolio problem*, working paper.

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

[14] Bracken, J. and McGill, J. M. (1974). *Defense applications of mathematical programs with optimization problem in the constraints*, Operations Research, Vol.22, 1086-1096.

[15] Can, A. and Ulusoy, G. (2014). *Multi-project scheduling with two-stage decomposition*, Annals of Operations Research, Vol.217, 95-116.

[16] Chen, R.-M. and Sandnes, F. E. (2014). *An efficient particle swarm optimizer with application to man-day project scheduling problems*, Mathematical Problems in Engineering, Vol.2014, 1-9.

[17] Cheng, C.-B. (2011). *Reverse auction with buyer-supplier negotiation using bi-level distributed programming*, European Journal of Operational Research, Vol.211, 601-611.

[18] Cheng, C.-B., Wu, H.-C. and Chan, C.-C. H. (2012). *Design of a combinational auction mechanism for television advertising market in Taiwan*, Research Journal of Applied Sciences, Engineering and Technology, Vol.4, 4105-4111.

[19] Chien, T.-T., Lin, Y.-I. and Tien, K.-W. (2013). *Agent-based negotiation mechanism for multi-project human resource allocation*, Journal of Industrial and Production Engineering, Vol.30, 518-527.

[20] de Vries, S. and Vohra, R. (2003). *Combinatorial auctions: A survey*, INFORMS Journal on Computing, Vol.15, No.3, 284-309.

[21] Economist Intelligence Unit (2009). *Closing the gap: The link between project management excellence and long-term success*, Economist Intelligence Unit Limited.

[22] Elloumi, S. and Fortemps, P. (2010). *A hybrid rank-based evolutionary algorithm applied to multi-mode resource-constrained project scheduling problem*, European Journal of Operational Research, Vol.205, 31-41.

[23] Epstein, R., Henrquez, L., Cataln, J., Weintraub, G. Y. and Martnez, C. (2002). *A combinational auction improves school meals in Chil*, Interfaces, Vol.32, 1-14.

[24] Goncalves, J. F., Mendes, J. J. M. and Resende, M. G. C. (2008). *A genetic algorithm for resource constrained multi- project scheduling problem*, European Journal of Operational Research. Vol.189, 1171-1190.

[25] Hartmann, S. (2001). *Project scheduling with multiple modes: a genetic algorithm*, Annals of Operations Research, Vol.102, 111-135.

[26] Hartmann, S. and Drexl, A. (1998). *Project scheduling with multiple modes: a comparison of exact algorithms*, Networks, Vol.32, 283-297.

[27] Jackson, C. (1976). *Technology for spectrum markets*, Ph.D. thesis, Department of Electronical Engineering, Massachusetts Institute of Technology, Cambridge, MA.

[28] Kim, K. W., Yun, Y., Yoon, J., Gen, M. and Yamazaki, G. (2005). *Hybrid genetic algorithm with adaptive abilities for resource-constrained multiple project scheduling*, Computers in Industry, Vol.56, 143-160.

[29] Kolisch, R. and Drexl, A. (1997). *Local search for nonpreemptive multi-mode resource-constrained project scheduling*, IIE Transactions, Vol.29, 987-999.

[30] Kolisch, R. and Padman, R. (2001). *An integrated survey of deterministic project scheduling*, OMEGA, Vol.29, 249-272.

[31] Kolisch, R. and Sprecher, A. (1996). *PSPLIB - a project scheduling problem library*, European Journal of Operational Research, Vol.96, 205-216.

[32] Kurtulus, I. S. and Narula, S. C . (1985). *Multi-project scheduling: analysis of project performance*, IIE Transactions, Vol.17, 58-66.

[33] Lai, Y.-J. (1996). *Hierarchical optimization: a satisfactory solution*, Fuzzy Seta and Systems, Vol.77, 321-335.

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

[35] Lova, A., Maroto, C. and Tormos, P. (2000). *A multicriteria heuristic method to improve resource allocation in multiproject environment*, European Journal of Operational Research, Vol.127, 408-424.

[36] Lova, A., Tormos, P., Cervantes, M. and Barber, F. (2009). *Efficient hybrid genetic algorithm for scheduling projects with resource constraints and multiple execution modes*, International Journal of Production Economics. Vol.117, 302-316.

[37] McMillian, J. (1994). Selling spectrum rights, Journal of Economic Perspectives, Vol.8, 145-162.

[38] Messelis, T. and De Causmaecker, P. (2014). *An automatic algorithm selection approach for the multi-mode resource-constrained project scheduling problem*, European Journal of Operational Research, Vol.233, 511-528.

[39] Mori, M. and Tseng, C. C. (1997). *A genetic algorithm for multi-mode resource constrained project scheduling problem*, European Journal of Operational Research, Vol.100, 134-141.

[40] Park, S. and Rothkopf, M. H. (2001). *Auctions with endogenously determined allowable combinations*, RUTCOR research report 3-2001, Rutgers University, New Brunswick, NJ.

[41] Prez, E., Posada, M. and Lorenzana, A. (2015). *Taking advantage of solving the resource constrained multi-project scheduling problems using multi-modal genetic algorithms*, Soft Computing, in press.

[42] Pritsker, A. A. B., Watters, L. J. and Wolfe, P. M. (1969). *Multiproject scheduling with limited resources: a zero-one programming approach*, Management Science, Vol.16, 93-108.

[43] Rassenti, S. J., Smith, V. L. and Bulfin, R. L., (1982). *A Combinatorial Auction Mechanism for Airport Time Slot Allocation*, The Bell Journal of Economics, Vol.13, 402-417.

[44] Rothkopf, M. H., Pekeč, A. and Harstad, R. M. (1998). *Computationally manageable combinatorial auctions*, Management Science, Vol.44, 1131-1147.

[45] Sandholm, T. (2000). *Approaches to winner determination in combinatorial auctions*, Decision Support Systems, Vol.28, 165-176.

[46] Sandholm, T. (2002). *Algorithm for optimal winner determination in combinatorial auctions*, Artificial Intelligence. Vol.135, 1-54.

[47] Schneeweiss, C. (2003). Distributed Decision Making, (2nd ed.) Springer-Verlag, Berlin.

[48] Shih, H.-S., Lai, Y.-J. and Lee, E. S. (1996). *Fuzzy approach for multi-level programming problems*, Computers and Operations Research Vol.23, 73-91.

[49] Slowinski, R. (1980). *Two approaches to problems of resource allocation among project activities: a comparative study*, Journal of Operations Research Society, Vol.31, 711-723.

[50] Slowinski, R., Soniewicki, B. and Weglarz, J. (1994). *DSS for multiobjective project scheduling*, European Journal of Operational Research, Vol.79, 220-229.

[51] Speranza, M. G. and Vercellis, C. (1993). *Hierachical models for multi-project planning and scheduling*, European Journal of Operational Research, Vol.64, 312-325.

[52] Sprecher, A. and Drexl, A. (1998). *Solving multi-mode resource-constrained project scheduling problem by a simple, general and powerful sequencing algorithm*, European Journal of Operational Research, Vol.107, 431-450.

[53] Sprecher, A., Hartmann, S. and Drexl, A. (1997). *An exact algorithm for project scheduling with multiple modes*, OR Spectrum, Vol.19, 195-203.

[54] Talbot, F. B. (1982). *Resource-constrained project scheduling with time- resource tradeoffs: the non-preemptive case*, Management Science, Vol.28, 1199-1210.

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

[56] Tseng, C.-C. (2008). *Two heuristic algorithms for a multi-mode resource-constrained multi-project scheduling problem*, Journal of Science and Engineering Technology, Vol.4, 63-74.

[57] Van Peteghem, V. and Vanhoucke, M. (2010). *A genetic algorithm for the preemptive and non-preemptive multi-mode resource-constrained project scheduling problem*, European Journal of Operational Research, Vol.201, 409-418.

[58] Van Peteghem, V. and Vanhoucke, M. (2014). *An experimental investigation of metaheuristics for the multi-mode resource-constrained project scheduling problem on new dataset instances*, European Journal of Operational research, Vol.235, 62-72.

[59] Vartouni, A. M. and Khanli, L. M. (2014). *A hybrid genetic algorithm and fuzzy set applied to multi-mode resource-constrained project scheduling problem*, Journal of Intelligent and Fuzzy Systems, Vol.26, 1103-1112.

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

[61] Wen, U.-P. and Hsu, S.-T. (1991), *Linear bi-level programming problems-a review*, Journal of Operational Research Society, Vol.42, 25-133.

[62] Werner, B. (1987). *An interactive fuzzy programming system*, Fuzzy Sets and Systems, Vol.23, 131-147.

[63] Yang, K.-K. and Sum, C.-C. (1993). *A comparison of resource allocation and activity scheduling rules in a dynamic multi-project scheduling environment*, Journal of Operations Management, Vol.11, 207-218.

[64] Yang K.-K. and Sum, C.-C. (1997). *An evaluation of due date, resource allocation, project release, and activity scheduling rules in a multiproject environment*, European Journal of Operational Research, Vol.103, 139-154.

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

Department of Information Management, Tamkang University, New Taipei City 25137, Taiwan (R.O.C.).

E-mail: cbcheng@mail.tku.edu.tw

Major area(s): Optimization, soft computing, machine learning.

Department of Information Management, Tamkang University, New Taipei City 25137, Taiwan (R.O.C.).

E-mail: silveryamanda@gmail.com

Major area(s): Project management, information systems.

Graduate Institute of Management Sciences, Tamkang University, New Taipei City 25137, Taiwan (R.O.C.).

E-mail: kenwa@mail.im.tku.edu.tw

Major area(s): Information security, national security, information management.

(Received February 2018; accepted December 2018)