Journal of Zhejiang University-SCIENCE C (Computers & Electronics)
ISSN 1869-1951 (Print); ISSN 1869-196X (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

Combinatorial auction algorithm for project portfolio selection and scheduling to maximize the net present value

Yong-yi SHOU[†1], Yi-lun HUANG[2]
([1]Department of Management Science & Engineering, Zhejiang University, Hangzhou 310058, China)
([2]Department of Instrument Science & Engineering, Zhejiang University, Hangzhou 310027, China)
[†]E-mail: yshou@zju.edu.cn

Received Aug. 5, 2009; Revision accepted Mar. 15, 2010; Crosschecked May 31, 2010

Abstract: Scheduling projects at the activity level increases the complexity of decision making of project portfolio selection but also expands the search space to include better project portfolios. An integer programming model is formulated for the project portfolio selection and scheduling problem. An iterative multi-unit combinatorial auction algorithm is proposed to select and schedule project portfolios through a distributed bidding mechanism. Two price update schemes are designed to adopt either a standard or an adaptive Walrasian tâtonnement process. Computational tests show that the proposed auction algorithm with the adaptive price update scheme selects and schedules project portfolios effectively and maximizes the total net present value. The price profile generated by the algorithm also provides managerial insights for project managers and helps to manage the scarce resources efficiently.

Key words: Project management, Portfolio selection, Combinatorial auction, Project scheduling
doi:10.1631/jzus.C0910479 **Document code:** A **CLC number:** TP39; O22

1 Introduction

Project management has gained increased importance in modern organizations and needs to be analyzed from a systematic perspective due to its dynamic complexity (Yeo, 1993). One of the main emphases in project management is project selection and scheduling, as it must ensure an effective and efficient use of substantial resources. The complexity of project selection and scheduling comes mostly from the interconnection of multiple projects. However, up to 90 percent of all projects, by value, are undertaken in a multiple project context (Turner, 2008). Given a set of project proposals and constraints, the decision maker has to select a subset of proposals, maybe only one project or the whole set of projects, in order to optimize the performance of the selected portfolio. In case of research and development (R&D) project management, the decision-making is especially critical for a firm's innovative capability and sustainable competitive advantage (Dye and Pennypacker, 1999).

The popular methods for project selection in industries normally consist of two stages: evaluating single projects first and then selecting projects through a greedy algorithm. Projects are ranked and prioritized according to a predetermined set of criteria (Henriksen and Traynor, 1999; Linton *et al*., 2002; Meade and Presley, 2002). In the decision-making processes of R&D project investment, the net present value (NPV) is a widely accepted criterion (Nakamura and Tsuji, 2004). Then projects are selected one by one according to their priority values until the available resources are depleted. These methods are straightforward and hence widely adopted in practice. However, a combination of high priority projects may not form a good portfolio (Chien, 2002).

Mathematical models for project portfolio selection are also common in the literature. Schmidt (1993) established a non-linear integer programming model to examine the interdependency of candidate projects. Three types of interactions are included in his model, namely the benefit, outcome, and resource interactions. A branch-and-bound algorithm was proposed to solve the model. Badri *et al*. (2001) developed a goal programming model for information system project selection. Stummer and Heidenberger (2003) suggested an integer programming approach to search for Pareto optimal portfolios in a multi-stage decision making process. Gabriel *et al*. (2006) proposed a multi-objective integer optimization model with cost probability distributions. More recently, Carazo *et al*. (2010) established a comprehensive model for multi-objective project portfolio selection. Since the portfolio selection problem is NP-hard (Doerner *et al*., 2006), a number of metaheuristics have emerged in recent years, for example, the evolutionary algorithm (Medaglia *et al*., 2007), and the ant colony algorithm (Doerner *et al*., 2004; 2006).

The majority of project portfolio selection researches do not take into consideration the project scheduling issue (Coffin and Taylor, 1996). Some papers in the literature (e.g., Ghasemzadeh *et al*., 1999; Carazo *et al*., 2010) included scheduling into their models, but just like the above-mentioned studies they still regarded a single project as a basic indivisible unit in decision making. It is a common assumption in the previous literature that a single project has a fixed and unchangeable schedule. Hence, a decision maker confines himself/herself to select project portfolios according to these predetermined schedules and their corresponding contribution to the organization. However, project managers can negotiate to re-schedule their activities so as to better utilize the limited resources. In some cases, one more project can be included without hindering other selected projects merely by adjusting the resource allocation amongst these projects. Hence, an integrated view of project selection and scheduling provides more flexibility and is beneficial to the organization's overall performance even though it increases the complexity of decision making.

Several recent studies have noted the merit of this extension. Gutjahr *et al*. (2008) proposed a model on project portfolio planning. In their model, scheduling and staff assignment for a candidate set of selected projects is a subproblem of the project selection problem. Chen and Askin (2009) established a more generic integer programming model for project selection and scheduling with two sets of decision variables for project selection and project scheduling at the activity level respectively. An implicit enumeration algorithm was proposed to search for all possible project priority sequences with high profit.

It is noted that multi-project scheduling contributes to the quality of project portfolio decision. In the implicit enumeration algorithm proposed by Chen and Askin (2009), a priority rule based heuristic for the resource-constrained project scheduling problem (RCPSP) was adopted to schedule a single project. Chen and Askin (2009) acknowledged that their implicit enumeration algorithm is still heuristic unless all activities of all selected projects are considered simultaneously when constructing schedules, because the subproblem of scheduling a selected project portfolio is a multi-project scheduling problem in nature. The treatment of scheduling multiple projects individually overlooks interdependencies and synergy amongst the multiple projects and hinders effective resource allocation (Kurtulus and Davis, 1982). Exact methods are not realistic for the resource-constrained multi-project scheduling problem (RCMPSP) since a single project RCPSP is already NP-hard (Demeulemeester and Herroelen, 2002). Hence, priority rule based heuristics are widely adopted for the RCMPSP although their performance is not guaranteed (Kurtulus and Narula, 1985). These heuristics are fast and can be easily integrated into project portfolio selection and scheduling algorithms (e.g., Gutjahr *et al*., 2008).

2 Problem description

Suppose there are a set of N competing projects and a pool of K types of limited resources. Besides the resource sharing there is no relationship amongst these projects.

Consider an individual project i consisting of n_i single mode activities with precedence relations amongst them. Only renewable resources (Brucker *et al*., 1999) are considered in this problem. For every single project, the responsible project manager needs to solve a single-mode resource-constrained project scheduling problem to optimize its objective value, i.e., to determine the start (completion) time of every activity without violating the precedence relations or resource constraints.

Given a limited resource pool or overall budget, if these projects are submitted to higher management or a committee, normally a subset of projects is selected to form an optimal portfolio to maximize the return on investment. Obviously, the project schedules will influence the selection of projects. Thus, this problem is a project portfolio selection and scheduling problem (PPSSP).

The notations used in this paper are listed in Table 1.

Table 1 Notations for the project portfolio selection and scheduling problem

Symbol	Explanation
i	Project index, i=1, 2, …, N, where N is the number of candidate projects
j	Activity index, j=1, 2, …, n_i, where n_i is the number of activities in project i
t	Time index, t=0,1, …, T, where T is the upper bound of time periods
k	Resource index, k=1, 2, …, K, where K is the number of resource types
d_{ij}	Duration of activity (i, j), or activity j in project i
P_{ij}	Set of immediate predecessors of activity (i, j)
ST_{ij}	Start time of activity (i, j)
CT_{ij}	Completion time of activity (i, j), $\mathrm{CT}_{ij}=\mathrm{ST}_{ij}+d_{ij}$
I_t	Set of activities during execution at time period t
R_k	Capacity of renewable resource k
r_{ijk}	Quantity of resource k required by activity (i, j) for its execution
$\mathrm{NPV}_i(t)$	The net present value of project i if it completes at time period t

It is assumed that the cash inflow occurs only at the project's completion. Hence, the NPV is a function of the project completion time. For a PPSSP, two sets of decision variables are involved. One set of decision variables is for project selection, to indicate if a specific project is accepted or rejected:

$$y_i=\begin{cases}1, & \text{if project } i \text{ is accepted,}\\ 0, & \text{if project } i \text{ is rejected.}\end{cases} \quad (1)$$

Another set of decision variables is for project scheduling, indicating when an activity shall start or complete:

$$x_{ijt}=\begin{cases}1, & \text{if activity } j \text{ of project } i \text{ completes at time period } t,\\ 0, & \text{otherwise.}\end{cases} \quad (2)$$

The problem is formulated as a 0-1 integer programming model (Chen and Askin, 2009):

$$\max \sum_{i=1}^{N}\sum_{t=0}^{T}\mathrm{NPV}_i(t)\cdot x_{i,n_i,t} \quad (3)$$

s.t.

$$R_k-\sum_{i=1}^{N}\sum_{j=1}^{n_i} r_{ijk}\sum_{\tau=t}^{t+d_{ij}-1} x_{ij\tau}\geq 0,\quad \forall k,t, \quad (4)$$

$$\sum_{t=1}^{T}(t-d_{ij})\cdot x_{ijt}-\sum_{t=0}^{T} t\cdot x_{iht}\geq 0,\ \forall i,j,\ \forall (i,h)\in P_{ij}, \quad (5)$$

$$\sum_{t=0}^{T} x_{ijt}=y_i,\quad \forall i,j, \quad (6)$$

$$y_i\in\{0,1\},\quad \forall i, \quad (7)$$

$$x_{ijt}\in\{0,1\},\quad \forall i,j,t. \quad (8)$$

The objective (3) aims to maximize the total NPV of the project portfolio. Constraint (4) ensures that at any time period the demand on any resource from the portfolio does not exceed its capacity. Constraint (5) describes the precedence relations among activities, requiring an activity to start only after all its predecessors have completed. Constraint (6) ensures that every activity is non-preemptive; i.e., activity preemption is not allowed. Constraints (7) and (8) declare the decision variables.

The model in (3)–(8) is not a two-stage problem. The optimal selection of projects depends on the project schedules, yet the optimal schedules also depend on the project selection decision. To some extent, this PPSSP model is an extension of both the project scheduling problem and the project portfolio selection problem. If all project schedules are predetermined, it becomes a typical project selection problem. If the selection of projects is predetermined, it becomes a multi-project scheduling problem. If only one project is selected, it simplifies to an RCPSP.

3 Iterative combinatorial auction

An auction mechanism is useful in pricing project resources and in selecting and scheduling project portfolios. An auction is a market institution with an explicit set of rules determining resource allocation and prices on the basis of bids from the market participants (McAfee and McMillan, 1987). Hence, the auction can be interpreted as a way of allocating resources with difficult-to-determine values. An auction procedure is consistent with the dual level management structure of project portfolio selection where a higher level manager or committee is in charge of resource allocation and project managers are responsible for individual project planning and scheduling (Yang and Sum, 1997).

Due to the substitution effect among different bidding objects, bidders may have preferences not for a particular item but for a combination of multiple objects. Such types of auctions are known as combinatorial auctions (de Vries and Vohra, 2003; Abrache *et al*., 2007). In most cases of combinatorial auctions, objects have interdependent values and different combinations of objects generate different values. The advantage of combinatorial auctions is that it leads to economical and more efficient allocations of bidding objects to the bidders.

There is no lack of applications of combinatorial auctions reported in the literature. For example, combinatorial auction mechanisms were designed for allocation of airport time slots (Rassenti *et al*., 1982), course registration (Graves *et al*., 1993), machine scheduling (Kutanoglu and Wu, 1999), production coordination in a supply chain (Ertogral and Wu, 2000), assignment of bandwidths to network users (Dramitinos *et al*., 2007), and transportation procurement (Lim *et al*., 2008). In these applications, an auctioneer announces the prices of the bidding objects and the bidders submit their combined bids according to their preferences.

3.1 Combined bids

In an auction the bidders bid for bidding objects, or items. In this paper, the resource k at time period t is regarded as a single item, denoted as g_{kt}. Thus, there is a set of items:

$$G=\{g_{kt}\mid 1\le k\le K,\ 0\le t\le T\}. \tag{9}$$

Note that item g_{kt} is not indivisible. It is a multi-unit item with R_k units.

Every activity requires a subset of resources during its execution. The demand of activity (i, j) on resource k, during its specific execution time, corresponds to a bundle of items:

$$B_{ijk}=\{g_{kt}\mid \mathrm{ST}_{ij}\le t<\mathrm{CT}_{ij}\}. \tag{10}$$

For item g_{kt} the number of units requested by bid B_{ijk} is equal to r_{ijk}. A combined bid of an activity is

$$B_{ij}=\bigcup_{k=1}^{K}B_{ijk}. \tag{11}$$

The collection of combined bids of all activities in a project forms a package of items corresponding to a project schedule

$$B_{i}=\bigcup_{j=1}^{n_i}B_{ij}. \tag{12}$$

This project bid may not be feasible. If a project schedule is precedence feasible, the start time of all its activities shall conform to constraint (4). Let ST(B_{ij}) denote the lowest index of time periods of items in B_{ij}. Then ST(B_{ij}) indicates the start time of the activity corresponding to bid B_{ij}. Hence, constraint (4) can be rewritten as

$$\mathrm{ST}(B_{ij})\ge \mathrm{ST}(B_{ih})+d_{ih},\ \forall j,\ \forall (i,h)\in P_{ij}. \tag{13}$$

If a project schedule is to be resource feasible, the demand by all its activities on any resource at any time period must not exceed the resource capacity. Let $D_{kt}(B_i)$ be the total number of units of item g_{kt} in a single project bid B_i. The resource constraint for a single project i can be rewritten as

$$D_{kt}(B_i)\le R_k,\quad \forall k,t, \tag{14}$$

where $D_{kt}(B_i)$ is measured by

$$D_{kt}(B_i)=\sum_{(i,j)\in I_t}r_{ijk}. \tag{15}$$

In the context of a PPSSP, when considering a single project schedule's resource feasibility, it is assumed that all resources are available for this project; i.e., no other projects compete for these resources. In reality, the resource pool is shared by all competing projects, and normally the pool is sufficient for any single project to perform its activities as soon as possible. That means constraint (14) is not a must-have constraint for a single project bid.

The combination of projects forms a project portfolio. Without considering the resource constraints, a combined bid corresponding to a portfolio is

$$B = \bigcup_{i=1}^{N} B_i, \tag{16}$$

where B_i is a precedence feasible bid subject to constraint (13).

This portfolio bid is precedence feasible since all its included projects are precedence feasible and it is assumed that no precedence relations exist among projects. Yet this portfolio may not be resource feasible. Let $D_{kt}(B)$ be the total number of units of item g_{kt} demanded by a portfolio bid B:

$$D_{kt}(B) = \sum_{i=1}^{N} D_{kt}(B_i). \tag{17}$$

A resource feasible portfolio bid B shall conform to the following constraint:

$$D_{kt}(B) \le R_k, \quad \forall k, t. \tag{18}$$

3.2 Bidders' utility

At a certain round during an iterative auction, let λ_{kt} be the current price of item g_{kt}. Given the price vector $\boldsymbol{\lambda}$ at this round, the payment of a single project bid B_i is

$$P_i(B_i, \boldsymbol{\lambda}) = \sum_{k=1}^{K}\sum_{t=0}^{T} \lambda_{kt} D_{kt}(B_i). \tag{19}$$

Since a project bid B_i corresponds to a single project schedule S_i, the payment can be rewritten as

$$P_i(B_i, \boldsymbol{\lambda}) = \sum_{k=1}^{K}\sum_{t=1}^{T} \lambda_{kt} \sum_{j=1}^{n_j} r_{ijk} \sum_{\tau=t}^{t+d_{ij}-1} x_{ij\tau}. \tag{20}$$

The return of this project bid is the NPV of the corresponding project schedule. Hence, the utility of a project bid B_i is defined as

$$\begin{aligned} U_i(B_i, \boldsymbol{\lambda}) &= \mathrm{NPV}_i(B_i) - P_i(B_i, \boldsymbol{\lambda}) \\ &= \mathrm{NPV}_i(\mathrm{CT}_{in_i}) - \sum_{k=1}^{K}\sum_{t=1}^{T} \lambda_{kt} \sum_{j=1}^{n_j} r_{ijk} \sum_{\tau=t}^{t+d_{ij}-1} x_{ij\tau}. \end{aligned} \tag{21}$$

A project bid with a non-positive utility value is not acceptable and is not to be submitted. Therefore, the utility function of project bid B_i is modified as

$$\begin{aligned} U_i(B_i, \boldsymbol{\lambda}) = \mathrm{sig}\big(\mathrm{NPV}_i(B_i) - P_i(B_i, \boldsymbol{\lambda})\big) \\ \cdot\big(\mathrm{NPV}_i(B_i) - P_i(B_i, \boldsymbol{\lambda})\big), \end{aligned} \tag{22}$$

where sig(·) is a unit step function, defined as

$$\mathrm{sig}(x) = \begin{cases} 1, & x > 0, \\ 0, & x \le 0. \end{cases} \tag{23}$$

A bidder (a single project manager) responsible for project bid B_i aims to maximize its utility value defined in Eq. (22) subject to precedence constraint (13). This bidder's optimization problem is equal to the following scheduling problem of a single project:

$$\begin{aligned} \max U_i(B_i, \boldsymbol{\lambda}) = \mathrm{sig}\big(\mathrm{NPV}_i(B_i) - P_i(B_i, \boldsymbol{\lambda})\big) \\ \cdot\left(\mathrm{NPV}_i(\mathrm{CT}_{in_i}) - \sum_{k=1}^{K}\sum_{t=0}^{T} \lambda_{kt} \sum_{j=1}^{n_j} r_{ijk} \sum_{\tau=t}^{t+d_{ij}-1} x_{ij\tau}\right) \end{aligned} \tag{24}$$

s.t.

$$\sum_{t=0}^{T} (t - d_{ij}) \cdot x_{ijt} - \sum_{t=0}^{T} t \cdot x_{iht} \ge 0, \ \forall j, \forall (i,h) \in P_{ij}, \tag{25}$$

$$\sum_{t=0}^{T} x_{ijt} = y_i, \quad \forall j, \tag{26}$$

$$x_{ijt} \in \{0,1\}, \quad \forall j, t. \tag{27}$$

This is a project scheduling problem with no resource constraints. The objective of this problem is to allocate resources effectively to minimize the cash outflows and to maximize the cash inflows at the project's completion.

In the problem (24)–(27), let c_{ijt} be the cash flow at an activity's completion:

$$c_{ijt} = \begin{cases} -\mathrm{NPV}_i(t) + \sum_{k=1}^{K} r_{ijk} \sum_{\tau=t-d_{ij}}^{t-1} \lambda_{k\tau}, & \text{if } j = n_i, \\ \sum_{k=1}^{K} r_{ijk} \sum_{\tau=t-d_{ij}}^{t-1} \lambda_{k\tau}, & \text{if } j < n_i. \end{cases} \tag{28}$$

Hence, for a given project i, the objective function in (24) is equal to:

$$\min \sum_{j=1}^{n_i}\sum_{t=0}^{T} c_{ijt} x_{ijt} \tag{29}$$

subject to (25)–(27).

The problem in (29) aims to minimize the total cost subject to precedence relations. Such a project cost minimization problem with no resource constraints can be converted to a maximum flow problem (Mohring *et al*., 2003), and hence can be solved efficiently by various methods such as the push-relabel method (Cherkassky and Goldberg, 1997).

3.3 Auctioneer's return

An auctioneer normally aims to maximize his/her income from selling items. The total payment from all winning bidders is a common objective in a combinatorial auction winner determination problem. However, in a project portfolio selection problem, a higher manager shall aim to maximize the total NPV in lieu of maximizing the payment from individual projects; otherwise, an increase in cost will inevitably be encouraged. A decision maker of project portfolio selection is much like an auctioneer in a combinatorial allocation problem (CAP) who aims to maximize the overall social efficiency of the market (Abrache *et al*., 2007). Therefore, a reasonable objective function for an auctioneer in a PPSSP is:

$$\max_{B} \sum_{B_i \in B} \mathrm{NPV}_i(B_i) \tag{30}$$

subject to constraint (18).

The auctioneer needs to find an optimal portfolio bid through a pricing mechanism. It is noted that this is a multi-unit combinatorial auction (MUCA). However, the PPSSP is extraordinarily complicated because the project bid not only has its specific utility function but also has its own internal structure to restrict the feasibility of item bundles.

3.4 Price update scheme

For MUCAs, it is rare that the equilibrium of demand and supply is achieved in a single round. The equilibrium can be achieved through a Walrasian tâtonnement process, a typical non-monotone price update scheme (Abrache *et al*., 2007). In an iterative combinatorial auction, the Walrasian tâtonnement process works as follows: the auctioneer announces the prices, and then each bidder submits his/her combined bid indicating how much of each item they would demand. No transactions take place at disequilibrium prices. Instead, prices are lowered for items with positive prices and excess supply, and prices are raised for items with excess demand.

Suppose at round b the price of item g_{kt} is λ_{kt}^{b} in a standard Walrasian tâtonnement process. The price is to be updated as follows:

$$\lambda_{kt}^{b+1} = \max\left\{0, \lambda_{kt}^{b} + s\left(D_{kt}(B^b) - R_k\right)\right\}, \tag{31}$$

where B^b is the portfolio bid at round b, and s is a fixed step size for price adjustment.

A classical result of the general equilibrium theory establishes that Walrasian equilibrium prices exist under conditions of continuity, monotony, and concavity of preference functions (Arrow and Debreu, 1954). The equilibrium single-item prices can be derived from the dual of the Lagrangian relaxation for a combinatorial auction problem with indivisible single-unit items (Bikhchandani and Mamer, 1997). As an MUCA aiming to optimize the total NPV in objective function (30), a special tâtonnement process with an effective price update scheme is requested.

In the model (3)–(8), the resource constraint (4) is the only constraint involving multiple projects and the other constraints are independent. Given a non-negative Lagrangian multiplier $\boldsymbol{\lambda}$, the model can be turned into a Lagrangian relaxation (LR) model:

$$\max\left[\sum_{k=1}^{K}\sum_{t=0}^{T}\lambda_{kt}R_k + \sum_{i=1}^{N}\left(\sum_{t=0}^{T}\left(\mathrm{NPV}_i(t)\cdot x_{i,n_i,t}\right) - \sum_{k=1}^{K}\sum_{t=0}^{T}\lambda_{kt}\sum_{j=1}^{n_i} r_{ijk}\sum_{\tau=t}^{t+d_{ij}-1} x_{ij\tau}\right)\right] \tag{32}$$

subject to (5)–(8).

This LR model can be decomposed into a series of independent single project subproblems:

$$\upsilon(\mathrm{LR}_{\lambda}) = \sum_{i=1}^{N}\upsilon(\mathrm{LR}_{\lambda,i}) + \sum_{k=1}^{K}\sum_{t=1}^{T}\lambda_{kt}R_k, \tag{33}$$

where $\upsilon(\mathrm{LR}_{\lambda})$ denotes the optimal value of the above LR problem at a given Lagrangian multiplier $\boldsymbol{\lambda}$, and $\upsilon(\mathrm{LR}_{\lambda,i})$ is the optimal value of the subproblem of single project i, which can be formulated as

$$\upsilon(\mathrm{LR}_{\lambda,i}) = \max\left(\sum_{t=0}^{T}\left(\mathrm{NPV}_i(t)\cdot x_{i,n_i,t}\right) - \sum_{k=1}^{K}\sum_{t=0}^{T}\lambda_{kt}\sum_{j=1}^{n_i} r_{ijk}\sum_{\tau=t}^{t+d_{ij}-1} x_{ij\tau}\right) \tag{34}$$

subject to (25)–(27).

Comparing objective function (34) with (24), it is clear that the optimal value of LR_i is equal to the utility of an optimal single project bid:

$$\upsilon(\mathrm{LR}_{\lambda,i}) = \max U_i(B_i, \lambda). \tag{35}$$

For a given Lagrangian multiplier $\boldsymbol{\lambda}$, $\upsilon(\mathrm{LR}_{\lambda})$ provides an upper bound of the optimal NPV of the original PPSSP. The best upper bound corresponds to the solution of the following Lagrangian dual problem:

$$\min_{\boldsymbol{\lambda}\geq \mathbf{0}} \upsilon(\mathrm{LR}_{\lambda}). \tag{36}$$

The dual problem can be solved by a subgradient method (Fisher, 1981) by gradually lowering the upper bound. However, a subgradient method needs to solve all subproblems optimally and this could be time consuming. A surrogate subgradient method (Zhao *et al*., 1999) can be applied without solving all subproblems optimally. A surrogate subgradient is defined as

$$\tilde{g}_{kt} = -R_k + \sum_{i=1}^{N}\sum_{j=1}^{n_i} r_{ijk} \sum_{\tau=t}^{t+d_{ij}-1} x_{ij\tau}. \tag{37}$$

Thus, a price update scheme based on the surrogate subgradient method for the Lagrangian dual problem can be established. Suppose at round b the price of item g_{kt} is λ_{kt}^{b}. The price is to be updated in an adaptive Walrasian tâtonnement style:

$$\lambda_{kt}^{b+1} = \max\{0,\ \lambda_{kt}^{b} + s^{b}\tilde{g}_{kt}^{b}\}, \tag{38}$$

where $\tilde{g}_{kt}^{b}$ is the surrogate subgradient given by

$$\tilde{g}_{kt}^{b} = D_{kt}(B^{b}) - R_k, \tag{39}$$

and B^b is the portfolio bid at round b, which is a feasible or infeasible solution provided by an approximate optimization method.

In Eq. (38), a practical step size is (Fisher, 1981; Zhao *et al*., 1999)

$$s^{b} = \alpha^{b} \frac{\upsilon(\mathrm{LR}_{\lambda^{b}}) - \mathrm{LB}}{\sum_{k}\sum_{t}(\tilde{g}_{kt}^{b})^{2}}, \tag{40}$$

where LB is a target lower bound of the LR problem and α^b is a scalar. The scalar α^b is halved whenever $\upsilon(\mathrm{LR}_{\lambda})$ has failed to decrease in a fixed number of successive auction rounds (Fisher, 1981). Such a pricing policy helps to adjust the prices substantially at the early stages and to fine tune the prices in later stages.

For a non-equilibrium price vector $\boldsymbol{\lambda}\neq\boldsymbol{\lambda}^{*}$, the optimal solution of LR_{λ} may not be a feasible solution to the original PPSSP, since there may be a resource conflict amongst competing single project bids. Hence, a fast heuristic is necessary to convert a resource-infeasible portfolio to a feasible one.

A serial schedule generation scheme (Kolisch, 1996) can be adopted to convert an infeasible portfolio. The priority value of an activity is defined by the start time in the given portfolio bid plus a fraction of its duration. All activities in the portfolio are then scheduled serially and start time is assigned as early as possible without violating the resource constraints. A multi-project schedule after conversion tends to be longer and hence has a decreased NPV. When there is tight resource availability, one or more projects may have negative NPVs, and hence they shall be excluded from the portfolio.

3.5 Iterative auction process

An iterative multi-unit combinatorial auction (MUCA) process for the PPSSP is described by the following pseudocode:

Algorithm 1 Multi-unit combinatorial auction (MUCA) procedure

Input: d_{ij}, P_{ij}, r_{ijk}, R_k, $\mathrm{NPV}_i(t)$.

Output: y_i, x_{ijt}.

```
WHILE criteria NOT satisfied
    FOR every single project i
        SCHEDULE to maximize its utility;
        COMPUTE its utility;
        IF utility<0 THEN
            SUBMIT no bid;
        ELSE
            SUBMIT a project bid;
        ENDIF
    ENDFOR
    COMPUTE upper bound;
    IF portfolio bid is NOT resource feasible THEN
        CONVERT to a feasible portfolio bid;
    ENDIF
    COMPUTE lower bound;
    UPDATE price vector;
ENDWHILE
OUTPUT the best portfolio (y, x).
```

The algorithm reads the data and initializes parameters. The initial price vector is set at zero. For every single project, the schedule is optimized to maximize its utility at the current price vector. If its utility is negative, the project manager waives the project bid. After all bidders have submitted their bids, an upper bound of the PPSSP is computed.

In most cases, the combined portfolio bid is not resource feasible and shall be converted to a feasible one. A fast heuristic based on the serial schedule generation scheme is employed. A lower bound is then computed.

The price vector is then updated by the auctioneer. Two schemes are available here: the standard Walrasian tâtonnement process defined in Eq. (31), and the adaptive Walrasian tâtonnement process defined in Eq. (38).

The auction stops when: (1) the lower bound is epsilon close to the upper bound; (2) the scale of step sizes is too small to adjust the price vector effectively in an adaptive tâtonnement process; or (3) a preset number of auction rounds are reached.

4 An illustrative example

The application of the proposed MUCA algorithm is illustrated in this section. The example consists of four candidate projects (Fig. 1). Only one type of renewable resource is demanded by the projects. The capacity of the resource is three units.

For each project, its NPV is a function of its completion time. A maximal NPV is achieved at its critical path duration which is determined by the traditional critical path method (CPM) without considering resource constraints. A project's NPV decreases 5% per time period after its critical path duration. For the four projects in Fig. 1, the maximal NPVs are 6, 8, 11, and 9, respectively.

At the first auction round, the prices are set at zero, and each project manager submits a CPM schedule to maximize its utility. Due to the limited resource capacity, the combined portfolio bid including all four projects is infeasible. Hence, the auctioneer raises the prices of over-demanded items. In a standard Walrasian tâtonnement process with a step size of 0.01, the auction continues with the upper bound and lower bound converging gradually (Fig. 2).

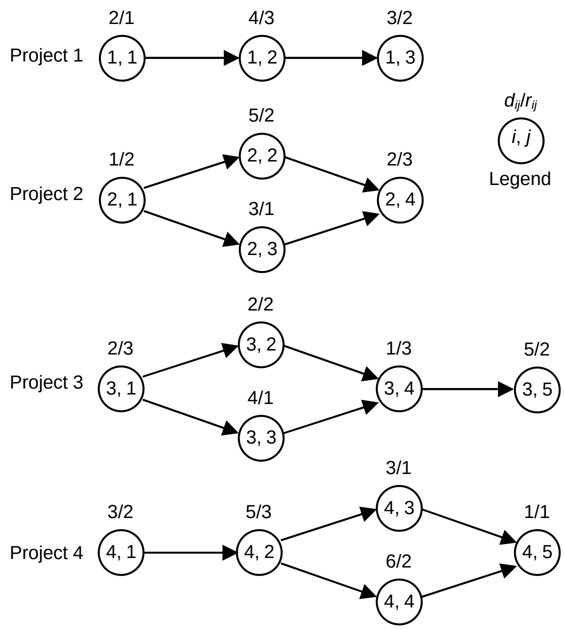

Fig. 1 Illustrative project selection problem

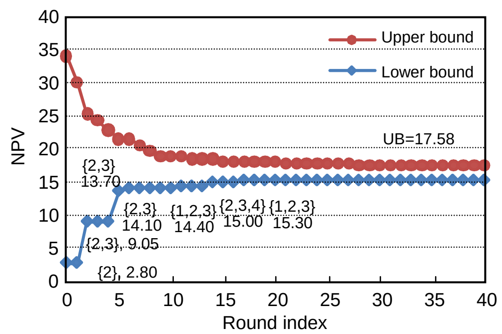

Fig. 2 Upper bound and lower bound values during the auction process

The tags along the lower bound curve show the projects selected, in brackets, and the corresponding NPVs

Initially, only one project is included in the feasible portfolio. With the auctioneer updating the price vector step by step, the bidders utilize the resource more effectively. At round 18, a portfolio including three projects is found. This portfolio includes projects 1, 2, and 3, and the corresponding multi-project schedule is shown in Fig. 3.

It is noted that a portfolio with the same three projects has been found in a previous round, as shown in Fig. 2. However, at that round the multi-project schedule of the portfolio is not optimized, and hence the portfolio's NPV is lower than 15.30. This illustrates that multi-project scheduling plays an important role in solving the PPSSP.

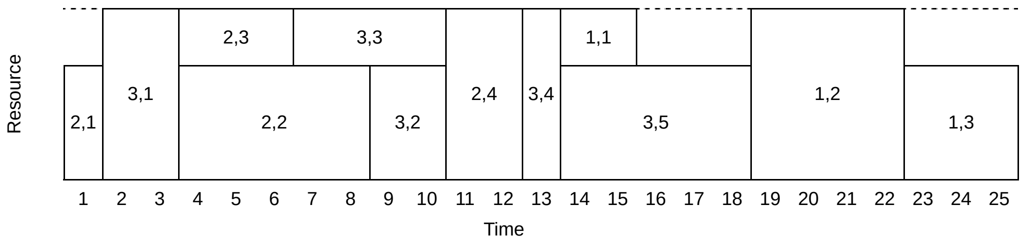

Fig. 3 The selected portfolio and its schedule

The final prices after the auction ends are presented in Fig. 4. It is obvious that competition for the resource is intense in the early time periods. Such an analysis can provide some managerial insights and help managers to monitor the resource closely in those periods. If the organization manages to increase the resource capacity at high price time periods, for example, through out-sourcing, it is possible that more projects can be included or the current portfolio can be accelerated to raise the total NPV.

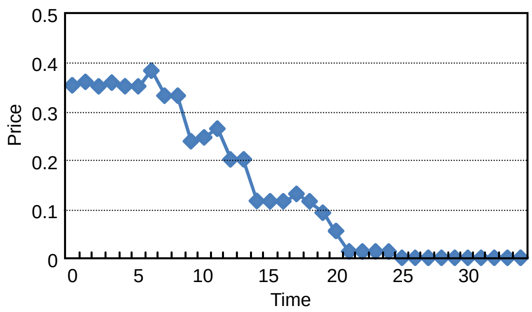

Fig. 4 The price profile after the auction ends

5 Computational testing

A computational test was conducted based on the problem set constructed by Chen and Askin (2009). The set uses RCPSP instances from the well-known Patterson set (Patterson, 1984). Seven project packages were generated. Each package had 10 projects with three types of renewable resources. A full factorial design with four experiment cells was realized by two levels of both resource availability and profit decreasing rate (Table 2). This experimental design, crossed with seven project packages, yielded 28 instances for project selection and scheduling. A detailed description on how to construct these instances is available in Chen and Askin (2009).

The iterative MUCA algorithm proposed in this paper was implemented in C language on a PC with duo CPUs at 2 GHz and 1 GB physical memory. The maximal number of auction rounds was set to be 30; a sequence of step sizes, from 0.0005, 0.01, 0.02, 0.04, to 0.08, were used for the standard Walrasian tâtonnement process.

Table 2 Experimental design

Cell	Resource availability	Profit decreasing rate
1	Low	Low
2	Low	High
3	High	Low
4	High	High

The MUCA algorithm was executed to solve the 28 instances, with two price update schemes, namely the standard Walrasian tâtonnement process and the adaptive Walrasian tâtonnement process. The results are listed in Table 3.

Wilcoxon signed ranks tests for data in Table 3 show that MUCA-A generated better portfolios than ENUM at a 0.001 significance level, and MUCA-S with step sizes of 0.01, 0.02, and 0.04 generated better portfolios than ENUM at a 0.05 significance level. MUCA-S with a step size of 0.08 generated portfolios with a slightly higher average NPV but the difference was not significant. MUCA-S with a step size of 0.005 generated worse portfolios than ENUM though the difference was still not significant. The tests showed that for MUCA-S the selection of a suitable step size is critical to its performance, while MUCA-A is more robust and is superior to MUCA-S with various step sizes at a 0.05 significance level.

The MUCA algorithms generated upper and lower bound values during the auction process. Table 4 presents the lower and upper bound values at the end of combinatorial auctions. For the MUCA algorithm with a standard Walrasian tâtonnement process, only MUCA-S with step size 0.02 is presented in Table 4 since it has the highest average NPV among all MUCA-S parameter settings in Table 3.

Table 3 Net present values achieved by different methods

Cell	Instance	Net present value						
		MUCA-S					MUCA-A**	ENUM
		s=0.005	0.01*	0.02*	0.04*	0.08		
1	1	2449	2423	2449	2455	2360	2449	2002
	2	3536	3498	3635	3635	3635	3635	3635
	3	1969	1969	1917	1969	1404	1969	1736
	4	2286	2582	2582	2509	2582	2582	2467
	5	3681	3681	3681	3681	3824	3856	3681
	6	2295	2295	2295	2338	2295	2295	2295
	7	2636	2542	2683	2763	2779	2779	2141
2	1	1975	2191	2191	2191	2191	2191	2218
	2	3635	3635	3635	3635	3635	3635	3635
	3	1969	1969	1969	1969	1969	1969	1922
	4	2467	2582	2582	2582	2582	2582	2467
	5	3681	3681	3681	3681	3681	3681	3681
	6	2295	2295	2295	2295	1906	2295	2295
	7	2372	2181	2171	2584	2171	2909	2454
3	1	5584	6133	6086	5842	5786	6050	5271
	2	8655	8719	8654	8271	8189	8621	7504
	3	5675	5598	5679	5647	5551	5756	5074
	4	6765	7124	7137	7012	6963	7268	6012
	5	8606	8565	8412	8444	8268	8606	7378
	6	6501	6448	6433	6360	6011	6520	5632
	7	6687	6836	6769	6743	6519	6824	6050
4	1	2396	5788	5788	5788	5670	5788	5454
	2	8128	8770	8154	8189	8189	8770	7924
	3	1665	5095	5084	5541	5216	5801	5259
	4	6908	6908	6488	6452	6682	6547	6464
	5	8240	8456	8056	7463	8113	8456	7799
	6	4989	4989	6065	6055	4827	6055	5787
	7	4439	4757	6751	6575	5769	6529	6267

s: step size. MUCA-S: MUCA algorithm with a standard Walrasian tâtonnement process; MUCA-A: MUCA algorithm with an adaptive Walrasian tâtonnement process; ENUM: the implicit enumeration algorithm in Chen and Askin (2009). * Better than ENUM at a 0.05 significance level; ** Better than ENUM at a 0.001 significance level

Table 4 shows that the proposed MUCA algorithms were effective for both Cell 3 and Cell 4 where the gaps between lower and upper bounds were lower than 10% on average. While for Cell 1 and Cell 2 the MUCA algorithms were relatively ineffective since the gaps were much larger. Therefore, it is clear that the resource availability is a significant determinant for the algorithm performance while the profit decreasing rate is not so significant.

To examine the quality of the proposed MUCA algorithms, the upper bounds provided by the mixed integer programming model (Chen and Askin, 2009) were adopted. The average gaps to the upper bound are shown in Table 5. The average gaps to the upper bound of MUCA-A were smaller than both MUCA-S and ENUM. It is noted that for Cell 3 and Cell 4, MUCA algorithms had much smaller gaps than ENUM, which means if resource availability is higher the MUCA algorithm has the potential to find close-to-optimal portfolios.

The average computation is listed in Table 6. Obviously the resource availability had a significant role in determining the computation time. If the resource availability is low, as in Cell 1 and Cell 2, the MUCA algorithm needs more time to search for resource-feasible portfolios.

Table 4 Lower and upper bound values of MUCA algorithms

Cell	Instance	LB		UB		Gap (%)	
		MUCA-S*	MUCA-A	MUCA-S*	MUCA-A	MUCA-S*	MUCA-A
1	1	2449	2449	2756	2789	11.14	12.18
	2	3635	3635	4557	4576	20.23	20.56
	3	1917	1969	2391	2427	19.82	18.88
	4	2582	2582	3165	3232	18.42	20.11
	5	3681	3856	4516	4515	18.49	14.60
	6	2295	2295	2907	2871	21.06	20.07
	7	2683	2779	3307	3335	18.88	16.66
	Average	2748.86	2795	3371.43	3392.13	18.47	17.60
2	1	2191	2191	2715	2724	19.29	19.57
	2	3635	3635	4564	4602	20.35	21.02
	3	1969	1969	2381	2377	17.29	17.17
	4	2582	2582	3216	3229	19.71	20.05
	5	3681	3681	4521	4535	18.59	18.84
	6	2295	2295	2869	2900	20.00	20.85
	7	2171	2909	3235	3271	32.90	11.07
	Average	2646.29	2751.71	3357.18	3377.03	21.18	18.52
3	1	6086	6050	6317	6158	3.66	1.75
	2	8654	8621	8766	8804	1.28	2.08
	3	5679	5756	5983	6127	5.09	6.06
	4	7137	7268	7385	7363	1.01	1.28
	5	8412	8606	9014	8658	3.35	0.60
	6	6433	6520	6522	6650	6.68	1.96
	7	6769	6824	7152	7215	5.36	5.41
	Average	7024.29	7092.14	7305.63	7282.12	3.85	2.61
4	1	5788	5788	6304	6320	8.19	8.41
	2	8154	8770	8818	8891	7.53	1.36
	3	5084	5801	6006	6088	15.35	4.71
	4	6488	6547	7404	7418	12.37	11.74
	5	8056	8456	8768	8826	8.12	4.19
	6	6065	6055	6792	6828	10.71	11.33
	7	6751	6529	7058	7181	4.34	9.08
	Average	6626.57	6849.43	7307.17	7364.47	9.31	6.99

* Step size s=0.02. Gap (%)=(UB−LB)/UB×100

Table 5 Average gap to the upper bound

Cell	Average gap to the upper bound (%)		
	MUCA-A	MUCA-S*	ENUM
1	13.93	15.35	21.00
2	22.42	25.40	24.80
3	5.15	6.06	18.00
4	11.16	14.05	16.70
Average	13.17	15.21	20.13

* Step size s=0.02

Table 6 Average computation time

Cell	Average computation time (s)	
	MUCA-S*	MUCA-A
1	45.241	46.330
2	44.970	46.167
3	22.392	24.877
4	24.386	24.939

* Step size s=0.02

6 Conclusions

An iterative multi-unit combinatorial auction algorithm is designed for project portfolio selection and scheduling in this paper. Two price update schemes are developed. The adaptive Walrasian tâtonnement process based on subgradient methods proves to be effective and robust in searching for high return project portfolios. For the standard Walrasian tâtonnement process, the selection of a suitable step size is critical for the quality of project portfolios. A case with four candidate projects is used to illustrate the operations of the auction procedure and the selected portfolio with its multi-project schedule. As demonstrated in the example case, the final price vector represents the resource demand situation and provides managerial insights to project managers.

Two directions deserve particular attention in future research. First, compared with classic project selection models, the integer programming model in this paper is simplified to some extent although it extends to schedule at the activity level. Interdependency among candidate projects is excluded, and only the NPV is included in performance criteria. If a more generic multi-objective model is established for the project portfolio and scheduling problem, the computational complexity will increase and more efficient algorithms are required. Second, the current model is a static one whereas information uncertainty prevails in practice. It is advisable to include some uncertainties, for example, stochastic activity durations and uncertain project benefit. Revised auction algorithms or other metaheuristics shall be designed for such stochastic models.

Acknowledgements

The authors thank Dr. Jia-qiong CHEN for providing the benchmark problem set and comments on the draft paper. The authors also thank the anonymous reviewers for their constructive suggestions.

References

Abrache, J., Crainic, T.G., Gendreau, M., Rekik, M., 2007. Combinatorial auctions. *Ann. Oper. Res.*, **153**(1):131-164. [doi:10.1007/s10479-007-0179-z]

Arrow, K., Debreu, G., 1954. Existence of equilibrium for a competitive economy. *Econometrica*, **22**(3):265-290. [doi:10.2307/1907353]

Badri, M.A., Davis, D., Davis, D., 2001. A comprehensive 0-1 goal programming model for project selection. *Int. J. Proj. Manag.*, **19**(4):243-252. [doi:10.1016/S0263-786399)00078-2]

Bikhchandani, S., Mamer, J., 1997. Competitive equilibrium in an exchange economy with indivisibilities. *J. Econ. Theory*, **74**(2):385-413. [doi:10.1006/jeth.1996.2269]

Brucker, P., Drexl, A., Mohring, R., Neumann, K., Pesch, E., 1999. Resource-constrained project scheduling: notation, classification, models, and methods. *Eur. J. Oper. Res.*, **112**(1):3-41. [doi:10.1016/S0377-2217(98)00204-5]

Carazo, A.F., Gomez, T., Molina, J., Hernandez-Diaz, A.G., Guerero, F.M., Caballero, R., 2010. Solving a comprehensive model for multiobjective project portfolio selection. *Comput. Oper. Res.*, **37**(4):630-639. [doi:10.1016/j.cor.2009.06.012]

Chen, J., Askin, R.G., 2009. Project selection, scheduling and resource allocation with time dependent returns. *Eur. J. Oper. Res.*, **193**(1):23-34. [doi:10.1016/j.ejor.2007.10.040]

Cherkassky, B.V., Goldberg, A.V., 1997. On implementing the push-relabel method for the maximum flow problem. *Algorithmica*, **19**(4):390-410. [doi:10.1007/PL00009180]

Chien, C.F., 2002. A portfolio-evaluation framework for selecting R&D projects. *R&D Manag.*, **32**(4):359-368. [doi:10.1111/1467-9310.00266]

Coffin, M.A., Taylor, B.W., 1996. Multiple criteria R&D project selection and scheduling using fuzzy logic. *Comput. Operat. Res.*, **23**(3):207-220. [doi:10.1016/0305-0548(96)81768-0]

Demeulemeester, E., Herroelen, W., 2002. Project Scheduling: a Research Handbook. Kluwer Academic Publishers, New York, p.203-206.

de Vries, S., Vohra, S., 2003. Combinatorial auctions: a survey. *INFORMS J. Comput.*, **15**(3):284-309. [doi:10.1287/ijoc.15.3.284.16077]

Doerner, K.F., Gutjahr, W.J., Hartl, R.F., Strauss, C., Stummer, C., 2004. Pareto ant colony optimization: a metaheuristic approach to multiobjective portfolio selection. *Ann. Oper. Res.*, **131**(1-4):79-99. [doi:10.1023/B:ANOR.0000039513.99038.c6]

Doerner, K.F., Gutjahr, W.J., Hartl, R.F., Strauss, C., Stummer, C., 2006. Pareto ant colony optimization with ILP preprocessing in multiobjective project portfolio selection. *Eur. J. Oper. Res.*, **171**(3):830-841. [doi:10.1016/j.ejor.2004.09.009]

Dramitinos, M., Stamoulis, G.D., Courcoubetis, C., 2007. An auction mechanism for allocating the bandwidth of networks to their users. *Comput. Networks*, **51**(18):4979-4996. [doi:10.1016/j.comnet.2007.08.007]

Dye, L.D., Pennypacker, J.S., 1999. Project Portfolio Management: Selecting and Prioritizing Projects for Competitive Advantage. Center for Business Practices, West Chester, PA, USA.

Ertogral, K., Wu, S.D., 2000. Auction-theoretic coordination of production planning in the supply chain. *IIE Trans.*, **32**(10):931-940. [doi:10.1023/A:1007660530806]

Fisher, M.L., 1981. The Lagrangian relaxation method for solving integer programming problems. *Manag. Sci.*, **27**(1):1-18. [doi:10.1287/mnsc.27.1.1]

Gabriel, S.A., Kumar, S., Ordonez, J., Nasserian, A., 2006. A multiobjective optimization model for project selection with probabilistic considerations. *Soc.-Econ. Plan. Sci.*, **40**(4):297-313. [doi:10.1016/j.seps.2005.02.002]

Ghasemzadeh, F., Archer, N., Iyogun, P., 1999. A zero-one model for project portfolio selection and scheduling. *J. Oper. Res. Soc.*, **50**(7):745-755. [10.1057/palgrave.jors.2600767]

Graves, R.L., Schrage, L., Sankaran, J., 1993. An auction method for course registration. *Interfaces*, **23**(5):81-92. [doi:10.1287/inte.23.5.81]

Gutjahr, W.J., Katzensteiner, S., Reiter, P., Stummer, C., Denk, M., 2008. Competence-driven project portfolio selection, scheduling and staff assignment. *Cent. Eur. J. Oper. Res.*, **16**(3):281-306. [doi:10.1007/s10100-008-0057-z]

Henriksen, A.D., Traynor, A.J., 1999. A practical R&D project-selection scoring tool. *IEEE Trans. Eng. Manag.*,

46(2):158-170. [doi:10.1109/17.759144]

Kolisch, R., 1996. Serial and parallel resource-constrained project scheduling methods revisited: theory and computation. *Eur. J. Oper. Res.*, **90**(2):320-333. [doi:10.1016/0377-2217(95)00357-6]

Kurtulus, I., Davis, E.W., 1982. Multi-project scheduling: categorization of heuristic rule performance. *Manag. Sci.*, **28**(2):161-172. [doi:10.1287/mnsc.28.2.161]

Kurtulus, I., Narula, S.C., 1985. Multi-project scheduling: analysis of project performance. *IIE Trans.*, **17**(1):58-66. [doi:10.1080/07408178508975272]

Kutanoglu, E., Wu, S.D., 1999. On combinatorial auction and Lagrangean relaxation for distributed resource scheduling. *IIE Trans.*, **31**(9):813-826. [doi:10.1023/A:1007666414678]

Lim, A., Rodrigues, B., Xu, Z., 2008. Transportation procurement with seasonally varying shipper demand and volume guarantees. *Oper. Res.*, **56**(3):758-771. [doi:10.1287/opre.1070.0481]

Linton, J.D., Walsh, S.T., Morabito, J., 2002. Analysis, ranking and selection of R&D projects in a portfolio. *R&D Manag.*, **32**(2):139-148. [doi:10.1111/1467-9310.00246]

McAfee, R.P., McMillan, J., 1987. Auctions and bidding. *J. Econ. Lit.*, **25**(2):699-738.

Meade, L.M., Presley, A., 2002. R&D project selection using the analytic network process. *IEEE Trans. Eng. Manag.*, **49**(1):59-66. [doi:10.1109/17.985748]

Medaglia, A.L., Graves, S.B., Ringuest, J.L., 2007. A multi-objective evolutionary approach for linearly constrained project selection under uncertainty. *Eur. J. Oper. Res.*, **179**(3):869-894. [doi:10.1016/j.ejor.2005.03.068]

Mohring, R.H., Schulz, A.S., Stork, F., Uetz, M., 2003. Solving project scheduling problems by minimum cut computations. *Manag. Sci.*, **49**(3):330-350. [doi:10.1287/mnsc.49.3.330.12737]

Nakamura, Y., Tsuji, M., 2004. Study on the allocation of R&D investment: method of evaluating and selecting R&D projects for investment. *Int. J. Innov. Technol. Manag.*, **1**(1):55-74. [doi:10.1142/S0219877004000027]

Patterson, J.H., 1984. A comparison of exact approaches for solving the multiple constrained resource, project scheduling problem. *Manag. Sci.*, **30**(7):854-867. [doi:10.1287/mnsc.30.7.854]

Rassenti, S.J., Smith, V.L., Bulfin, R.L., 1982. A combinatorial auction mechanism for airport time slot allocation. *The Bell J. Econ.*, **13**(2):402-417. [doi:10.2307/3003463]

Schmidt, R.L., 1993. A model for R&D project selection with combined benefit, outcome and resource interactions. *IEEE Trans. Eng. Manag.*, **40**(4):403-410. [doi:10.1109/17.257733]

Stummer, C., Heidenberger, K., 2003. Interactive R&D portfolio analysis with project interdependencies and time profiles of multiple objectives. *IEEE Trans. Eng. Manag.*, **50**(2):175-183. [doi:10.1109/TEM.2003.810819]

Turner, J.R., 2008. The Handbook of Project-Based Management. McGraw-Hill, London, p.323-326.

Yang, K.K., Sum, C.C., 1997. An evaluation of due date, resource allocation, project release, and activity scheduling rules in a multiproject environment. *Eur. J. Oper. Res.*, **103**(1):139-154. [doi:10.1016/S0377-2217(96)00266-4]

Yeo, K.T., 1993. Systems thinking and project management: time to reunite. *Int. J. Proj. Manag.*, **11**(2):111-117. [doi:10.1016/0263-7863(93)90019-J]

Zhao, X., Luh, P.B., Wang, J., 1999. Surrogate gradient algorithm for Lagrangian relaxation. *J. Optim. Theory Appl.*, **100**(3):699-712. [doi:10.1023/A:1022646725208]