

# Multi-Mode Resource Constrained Multi-Project Scheduling and Resource Portfolio Problem<sup>1</sup>

Umut Beşikci<sup>a</sup>, Ümit Bilge<sup>a</sup>, Gündüz Ulusoy<sup>b,\*</sup>

<sup>a</sup>*Boğaziçi University, Department of Industrial Engineering, Bebek, 34342, Istanbul, Turkey*

*umut.besikci@boun.edu.tr, bilge@boun.edu.tr*

<sup>b</sup>*Sabancı University, Faculty of Engineering and Natural Sciences, Orhanlı, Tuzla, 34956 Istanbul, Turkey*

*gunduz@sabanciuniv.edu*

---

## Abstract

This paper introduces a multi-project problem environment which involves multiple projects with assigned due dates; activities that have alternative resource usage modes; a resource dedication policy that does not allow sharing of resources among projects throughout the planning horizon; and a total budget. Three issues arise when investigating this multi-project environment. First, the total budget should be distributed among different resource types to determine the general resource capacities, which correspond to the total amount for each renewable resource to be dedicated to the projects. With the general resource capacities at hand, the next issue is to determine the amounts of resources to be dedicated to the individual projects. The dedication of resources reduces the scheduling of the projects' activities to a multi-mode resource constrained project scheduling problem (MRCPSP)

---

\*Corresponding Author. Tel: ++(90) 216 483-9503 Fax: ++(90) 216 483-9550

<sup>1</sup>NOTICE: This is the authors' version of a work that was accepted for publication in European Journal of Operational Research. Changes resulting from the publishing process, such as peer review, editing, corrections, structural formatting, and other quality control mechanisms may not be reflected in this document. Changes may have been made to this work since it was submitted for publication. A definitive version will be published in European Journal of Operational Research, 2014. DOI 10.1016/j.ejor.2014.06.025.

for each individual project. Finally, the last issue is the efficient solution of the resulting MRCPSPs. In this paper, this multi-project environment is modeled in an integrated fashion and designated as the Resource Portfolio Problem. A two-phase and a monolithic genetic algorithm are proposed as two solution approaches, each of which employs a new improvement move designated as the combinatorial auction for resource portfolio and the combinatorial auction for resource dedication. A computational study using test problems demonstrated the effectiveness of the solution approach proposed.

*Keywords:* Project scheduling, resource portfolio problem, multi-project scheduling, resource dedication, resource preference.

---

## 1. Introduction

Multi-project problem environments define the nature of business in most manufacturing and service companies. Different studies also point out this fact. Lova et al. (2000) states that 84% of the companies work with multiple, simultaneous projects and Payne (1995) notices that 90%, by value, of all projects occur in the multi-project context. Large construction companies (Liberatore et al., 2001), for example, regularly execute multi-project scheduling procedures. Resource related decisions are one of the prominent aspects of multi-project environments, since the resource based relations define the environment as a multi-project problem by coupling individual projects. The characterization of resource use by the individual projects in a multi-project environment is called the resource management policy (Besiki et al., 2013). Resource management policy can differ with respect to the environment characteristics (e.g., geographical distribution of projects, specific resource characteristics, etc.). A common approach for solving the multi-project scheduling problems in the literature assumes a resource sharing (RS) policy among the different projects forming a shared pool for each renewable resource. This policy cannot be applied in certain multi-project environments where resource sharing is not applicable because of various

reasons. The most common ones are listed as follows:

- The project characteristics may not allow RS. This scenario can be seen in various Research and Development (R&D) projects where the development process is highly technology intensive. Another example can be geographical limitations, where projects are so distributed across the world that resource sharing becomes impractical.
- The resource characteristics may not allow RS. For example, in software development projects, it is not desired to allocate developers to different projects because of the learning curve concept. An additional example can be offered in situations in which the sharing of heavy machinery equipment amongst projects is not possible due to the prohibitive installation costs.

These cases require a different resource management policy, which identifies the unique characteristics of a particular environment. To define a problem environment with the aforementioned characteristics, Besikci et al. (2013) propose the resource dedication (RD) policy. Under the RD policy, resources are not shared from a common pool but dedicated in certain amounts to individual projects throughout their durations.

In RS policy, the general resource capacity corresponds to the size of the shared pool for each renewable resource. Under RD policy, on the other hand, general resource capacity corresponds to the amount of each renewable resource to be distributed among the projects. In multi-project scheduling problems, the general resource capacity is taken as a given. But here, determining the set of general resource capacities becomes a decision in itself to be made according to a given total budget for a given set of projects. The determination of the general resource capacities subject to a total budget and the solution of the multi-project scheduling problem subject to these capacities under the RD policy is referred to as the Resource Portfolio Prob-

lem (RPP) in this study. Both the mathematical model and the proposed solution approach for RPP are the original contributions of this study.

The remainder of the paper continues with a brief literature review of resource considerations in project scheduling literature. In section 3, RPP is defined and its mathematical formulation is presented. In section 4, both the proposed two-phase genetic algorithm (GA) and a so-called monolithic GA approach are explained in detail. The computational study and its results are given in section 5. The conclusions and further research directions are presented in section 6.

## 2. Literature Review

Recall that RPP deals with the determination of resource levels (capacities) to provide for the resource requirements of the activities under RD policy subject to a given total budget. Hence, the literature review is restricted to studies addressing the determination of resource requirements of projects and different resource management policies. Interested readers can refer to Hartmann and Briskorn (2010) and Weglarz et al. (2011) for a survey of RCPSP and its extensions.

Herroelen (2005) introduces a hierarchical framework for project scheduling and control that identifies three different hierarchical levels (strategic, tactical and operational) and three functional planing areas (technological, capacity and material coordination). With respect to this categorization, our paper deals with problems related to resource capacity planning. Herroelen (2005) distinguishes four resource capacity functions: strategic resource planning, rough-cut capacity planning, resource constrained project scheduling, and detailed scheduling.

Möhring (1984) investigates the resource requirement problem for project scheduling with a graph theoretical approach. The author defines two important problem classes for project scheduling. The first problem, *problem A*, is defined as a problem of scarce resources where the objective is finding

the shortest project duration with a given amount of resources. The second problem, *problem B*, aims to find the least cost resource requirements within a given project duration, which is defined as problem of scarce time. The author uses graph theory to represent and solve these project scheduling problems where a special “dual” relation is employed between the scarce resource and scarce time problems.

Another study relating to the problem under consideration here and of the form of the problem of scarce time is presented by Demeulemeester (1995), which is defined as the resource availability cost problem (RACP). The general idea is to determine the least cost resource requirements for a single project scheduling problem. The proposed formulation for RACP is similar to the formulations for the resource constrained project scheduling problem (RCPSP) with the basic differences in the areas of objective function and the constraint for project duration. The solution methodology proposed for RACP is based on decision problems defined with resource limits.

An approach addressing the determination of resource requirements of projects is the rough-cut capacity planning (RCCP) for a multi-project environment. A recent example is provided by the Gademann and Schutten (2004). Two variants of the RCCP problem are defined in line with *problem of scarce resources* and *problem of scarce time* as presented by Möhring (1984), namely resource driven and time driven. The authors propose a linear program for the former, the time driven RCCP problem.

Speranza and Vercellis (1993) propose a two stage decomposition approach for multi-project scheduling where the projects can have precedence relations among themselves. The first stage of the approach aims to define the projects as “macro-activities” with multiple modes, where each mode is determined by solving a mathematical model for the project within a given budget that yields a finish time for the project. The authors assume that a set of possible budget limitations for a project can be estimated; such limitations are identified as “macro-modes”, constituting different modes for a project.

The single project network formed using the macro-activities with multiple modes is expressed as a multi-mode resource constrained project scheduling problem (MRCPSP) where the objective is the maximization of the net present value. The solution to this problem provides the required start and finish times of the projects, as well as the total nonrenewable and renewable resource capacities that a project can use within a given period. In the second stage of the approach, this information is used for detailed scheduling of individual projects for makespan minimization, within the given amount of renewable and nonrenewable resources.

An example of a different resource management policy in a multi-project environment is presented in Besikci et al. (2013), namely the Resource Dedication Problem (RDP). RDP is defined as the optimal dedication of resource capacities to different projects within the overall limits of the resources and with the objective of minimizing a predetermined objective function. Besikci et al. (2013) present a mathematical formulation for RDP and two different solution methodologies. The first solution approach is a genetic algorithm (GA) employing a new improvement move called the combinatorial auction for resource dedication (CA for RD), which is based on project preferences for resources. The second solution approach is a Lagrangian relaxation based heuristic employing subgradient optimization.

Krüger and Scholl (2009) investigate another resource management policy in their study, namely resource sharing with sequence dependent transfer times. The multi-project environment consists of multiple projects with single modes and a resource transfer time requirement when a resource is shared between different projects or when a resource flow occurs among the activities of the same project. The problem is referred to as the modified resource constrained multi-project scheduling problem with transfer times (RCMP-SPTT). The authors propose different heuristic rules for the parallel and serial scheduling schemes modified for the resource transfer concept. The RD policy studied in this paper corresponds to the transfer reducing ap-

proach of Krüger and Scholl (2009), where transfer of resources are strictly prohibited.

### **3. A Mathematical Programming Formulation for the Resource Portfolio Problem**

The RPP in a multi-project environment determinates the general resource capacities for a given total resource budget and the dedication of a set of resources to a set of projects with assigned due dates according to the determined general resource capacities in such a way that individual project schedules would result in an optimal solution for a predetermined objective. All projects are assumed to be ready to start. Uncertainties are thus not considered. The projects involve finish to start zero time lag and non-preemptive activities. There are both renewable and nonrenewable resources with limited capacities. Finally, each activity in each project has a set of execution modes with different time-resource alternatives. The projects in the problem environment are coupled with the general resource capacity decisions and not subject to precedence relations among themselves. The objective for the problem environment is taken as the minimization of the total weighted tardiness of the projects. The general problem studied in this paper can be thought of as an integrated capacity planning and multi-project scheduling problem under RD policy.

The multi-project environment of RPP has a high internal dependency among the projects in the sense expressed by Hans et al. (2007) because of the general resource budget. On the other hand, for a given RD within RDP, in the resulting MRCPPs, a low internal dependency among the projects exists. The proposed mathematical formulation is given below.

*Sets and Indices:*

$V$	Set of projects, $v \in V$
$J_v$	Set of activities of project $v$ , $j \in J_v$
$vN$	Last activity (indexed with $N$ ) of project $v$
$P_v$	Set of all precedence relationships of project $v$
$M_{vj}$	Set of modes for activity $j$ of project $v$ , $m \in M_{vj}$
$K$	Set of renewable resources, $k \in K$
$I$	Set of nonrenewable resources, $i \in I$
$T_v$	Set of time periods of project $v$ , $t \in T_v$

*Parameters:*

$E_{vj}$	Earliest finish time of activity $j$ of project $v$
$L_{vj}$	Latest finish time of activity $j$ of project $v$
$d_{vjm}$	Duration of activity $j$ of project $v$ , operating on mode $m$
$r_{vjk m}$	Renewable resource $k$ usage of activity $j$ of project $v$ , operating on mode $m$
$w_{vji m}$	Nonrenewable resource $i$ usage of activity $j$ of project $v$ , operating on mode $m$
$dd_v$	Assigned due date for project $v$
$c_v$	Relative weight of project $v$
$cr_k$	Unit cost of renewable resource $k$
$cw_i$	Unit cost of nonrenewable resource $i$
$tb$	Total resource budget

*Decision Variables:*



$$x_{vjmt} = \begin{cases} 1 & \text{if activity } j \text{ of project } v, \text{ operating on mode } m, \text{ is finished} \\ & \text{at period } t \\ 0 & \text{otherwise} \end{cases}$$

$BR_{vk}$  = Amount of renewable resource  $k$  dedicated to project  $v$

$BW_{vi}$  = Amount of nonrenewable resource  $i$  dedicated to project  $v$

$TC_v$  = Weighted tardiness cost of project  $v$

$R_k$  = Total amount of required renewable resource  $k$

$W_i$  = Total amount of required nonrenewable resource  $i$

*Mathematical Model RPP-RD*

$$\min. z = \sum_{v \in V} TC_v \quad (1)$$

Subject to

$$\sum_{m \in M_{vj}} \sum_{t=E_{vj}}^{L_{vj}} x_{vjmt} = 1 \quad \forall j \in J_v \text{ and } \forall v \in V \quad (2)$$

$$\sum_{m \in M_{vb}} \sum_{t=E_{vb}}^{L_{vb}} (t - d_{vbm}) x_{vbm} \geq \sum_{m \in M_{va}} \sum_{t=E_{va}}^{L_{va}} t x_{vam} \quad \forall (a, b) \in P_v \text{ and } \forall v \in V \quad (3)$$

$$\sum_{j \in J_v} \sum_{m \in M_{vj}} \sum_{q=\max\{t, E_{vj}\}}^{\min\{t+d_{vjm}-1, L_{vj}\}} r_{vjkm} x_{vjmq} \leq BR_{vk} \quad \forall k \in K, t \in T, v \in V \quad (4)$$

$$\sum_{j \in J_v} \sum_{m \in M_{vj}} \sum_{t=E_{vj}}^{L_{vj}} w_{vjim} x_{vjmt} \leq BW_{vi} \quad \forall i \in I \text{ and } \forall v \in V \quad (5)$$

$$\sum_{v \in V} BR_{vk} \leq R_k \quad \forall k \in K \quad (6)$$

$$\sum_{v \in V} BW_{vi} \leq W_i \quad \forall i \in I \quad (7)$$

$$\sum_{i \in I} c w_i W_i + \sum_{k \in K} c r_k R_k \leq tb \quad (8)$$

$$TC_v \geq c_v \left( \sum_{t=E_{vN}}^{L_{vN}} \sum_{m \in M_{vN}} x_{vNmt} - dd_v \right) \quad \forall v \in V \quad (9)$$

$$x_{vjmt} \in \{0, 1\} \quad \forall j \in J_v, \forall t \in T_v, \forall m \in M_{vj} \text{ and } \forall v \in V \quad (10)$$

$$BR_{vk}, BW_{vi}, R_k, W_i, TC_v \in Z^+ \quad \forall v \in V, \forall k \in K \text{ and } \forall i \in I \quad (11)$$

Objective function (1) minimizes the total weighted tardiness over all projects. Constraint set (2) ensures that all activities are scheduled once and only once for all projects. Constraint set (3) reflects the precedence relationships among the activities of all projects. Constraint set (4) sets the maximum level for the renewable resource usage over all projects and resource types. Constraint set (5) imposes the maximum level for the nonrenewable resource usage over all projects and resource types. Constraint sets (6) and (7) calculate the required renewable and nonrenewable resource capacities according to the dedicated renewable and nonrenewable resources, respec-

tively. Constraint set (8) ensures that the total cost of the general renewable and nonrenewable resources does not exceed the total available budget. Constraint set (9) calculates the weighted tardiness values for each project. Constraint sets (10)-(11) specify the feasible ranges for the decision variables. Constraints (5) and (7) can be represented with  $X_{vjmt}$  decision variables, but for accounting purposes and model clarity we have used nonrenewable resource dedication decision variables ( $BW_{vi}$ ). Note that, this is not the case for the renewable resource dedication decision variables ( $BR_{vk}$ ), since they are calculated as the maximum renewable resource allocation throughout the planning horizon.

When the problem formulation is interpreted according to the hierarchical framework of Herroelen (2005) it can be seen that RPP includes different levels of resource capacity functions. The strategic part of the problem definition includes the resource capacity planning for the projects under consideration by determining the capacity levels of the resources. Here, the important part of the problem is the determination of the general resource capacities according to the requirements of the projects from a general budget. The dedication of the resources to the individual projects forms the rough-cut capacity planning part of the hierarchical framework. Finally operational considerations are covered by solving resource constrained project scheduling problems and generating detailed schedules for the projects. RPP approaches the resource capacity planning problem of the multi-project environments holistically. Nevertheless, as the following sections describe in detail, the solution procedure distinguishes various levels of the problem and approaches these different levels by conceptually decomposing the problem.

#### 4. The Proposed Solution Methodologies

In the paper by Besikci et al. (2013), efficient solution approaches for RDP are developed including a GA application employing a new local improvement heuristic called CA for RD. Basically, CA for RD uses the preferences

of projects for resources and moves the current solution to a more “preferable” RD state. The insights gained from the studies related with RDP constitute the basis for the proposed solution approaches presented here for RPP. During the development of GA, special emphasis is given to the local improvement routines, since as indicated by several authors (e.g., Peteghem and Vanhoucke (2014), Lova et al. (2006), and Hartmann (2001)) problem specific improvement routines contribute extensively to the success of the particular metaheuristic employed.

The RPP formulation includes capacity allocation dimension beyond RDP, namely, determination of the general resource capacities to be dedicated to the individual projects under a total budget constraint. In other words, in addition to the RD space (RDS), the whole solution space of RPP possesses another dimension: general resource capacities. Thus, a search algorithm for RPP should explore resource capacities space (RCS) (different general resource capacity instances) and RDS (different RD instances, which are constrained by a general resource capacity instance), and further, for each general resource capacity and corresponding dedication instance, an MRCPSP should be formulated and solved for each project. A two-phase GA algorithm is proposed for searching this complex solution space. A so-called monolithic GA approach is also suggested in this paper, which simultaneously applies different space search algorithms with the purpose of comparative evaluation with the results of the two-phase GA.

#### *4.1. A Two-Phase Genetic Algorithm for the Resource Portfolio Problem*

Recognizing the hierarchical nature of RPP, a two-phase GA is proposed, where the first phase in GA is the RD phase; the second, the resource portfolio (RP) phase. In the first phase of the GA, the RDS is explored. The search algorithms (GA operators and CA for RD) used in this phase operate only on the RD part of the individuals as defined in 4.1.1. In the second phase of the GA, individuals are subject to RP space search (RPS) in addition to the RDS search. For RPS search new GA operators and an improvement heuristic,

CA for RP, are used. The rationale behind this two-phase approach is to facilitate RPS search with a widely explored RDS.

#### 4.1.1. Individual Representation and Fitness Calculation

The representation of an individual is shown in Table 1 for four projects and two renewable (R1,R2) and two nonrenewable (W1,W2) resources. The general resource capacities and RDs are combined into a single chromosome. The RD part of the individual is represented with the values under the project columns. These dedication values are feasible according to the general renewable and nonrenewable resource capacities presented in the resource capacities column. On the other hand, the general resource capacities constituting the RP part are feasible for the total resource budget.

Table 1: Representation of an individual

Resources	Projects				Resource Capacities
	Project1	Project2	Project3	Project	
R1	30	40	25	60	155
R2	40	45	40	50	175
W1	50	60	30	50	190
W2	15	40	50	40	145

The fitness value for an individual is the total weighted tardiness value for all projects. The weighted tardiness for each project is calculated by solving an MRCPSPP for each project when a new RD is selected. Although an individual has general resource capacities feasible according to the total budget and resource dedications feasible according to the general resource capacities, the individual projects can have infeasible schedules because of the RDs of the projects. This infeasibility is reflected in a penalty to the fitness calculation, which is taken as 1.5 times the time horizon of the corresponding project.

#### 4.1.2. Initial Population Generation

To generate an individual, first, general resource capacities are determined and afterwards RD values are generated. To generate the general resource capacities part of the individuals in the initial population according to the total budget, three different methods are used. The first method generates general resource capacities directly from the total budget proportional to the no-delay resource requirement (the required amount of resources for the no-delay schedule of the project) totals of the projects. One instance is generated in this manner. The second approach starts by satisfying the no-delay resource requirement of a given resource from the total budget and generates the general resource capacities for the remaining resources from the remaining total budget, proportional to the no-delay resource requirement totals of the projects. Instances as many as the number of resources ( $|R| + |W|$ ) are generated in this manner. Finally, the last approach randomly generates as many general resource capacity instances to obtain a total number of  $Z$  general resource capacity instances.

The RD parts of the individuals are obtained by three different methods. The first method generates one individual by dedicating the available general resource capacities proportional to the no-delay resource requirements of the projects. For each project, as many individuals as the number of projects ( $|V|$ ) are generated using the second method. For each project the no-delay resource requirement is satisfied, and the remaining general resource capacities are dedicated to the remaining projects proportional to the no-delay resource requirements of the corresponding projects. In the last approach, RDs are randomly generated for each project so as to end up with  $H$  number of RD parts in total for each general resource capacity instance under consideration. Thus, as a result, an initial population of  $ZxH$  individuals is created.

Table 2 gives an example for the generation of an individual in the initial solution, which is obtained using the first method introduced above. To

obtain the resource capacities, note that for renewable resources R1 and R2, the no-delay requirements are 100 and 200 units, respectively and for the nonrenewable resources W1 and W2, it is 600 units each. Hence, 1000 units are distributed among these resource types proportionally as 67, 133, 400 and 400 units resulting in the respective resource capacities. The RD values are obtained as follows. In the R2 row of Table 2, for example, the distribution of 133 units of R2 among the projects P1, P2, P3 and P4 follows the ratios obtained from the no-delay requirements as 0.25, 0.40, 0.15 and 0.20, respectively.

Table 2: Sample individual generation one for the initial solution for four projects (P1, P2, P3, P4), two renewable (R1, R2) and two nonrenewable (W1, W2) resources with 1,000 unit general budget using method 1.

Resources	No-delay Requirements	Resource Capacities	Resource Dedications			
			P1	P2	P3	P4
R1	P1:20 P2:15 P3:30 P4:35	67	14	10	20	23
R2	P1:50 P2:80 P3:30 P4:40	133	33	53	20	27
W1	P1:150 P2:80 P3:130 P4:240	400	100	53	87	160
W2	P1:200 P2:80 P3:100 P4:220	400	133	53	67	147

#### 4.1.3. Resource Dedication Space Search

RDS search is executed on the RD part of the individuals with different crossover and mutation operators in addition to CA for RD. Here only a summary of CA used for RD and the GA operators will be given. A detailed explanation can be found in Besikci et al. (2013).

#### Genetic Algorithm Operators

There are three mutation operators defined for the RDS search. The first mutation operator swaps RD values of two different projects of a randomly selected resource in an individual. The second mutation operator swaps two randomly selected RD values within a randomly selected project in an indi-

vidual. Finally, in the last mutation operator, two RD values are randomly swapped without any resource or project selection.

Three different crossover operators are defined for RDS search. The first crossover operator creates two new individuals from two randomly selected individuals by changing RDs of two randomly selected projects. The second crossover operator swaps RD values of selected resources of two different individuals. And finally, the last crossover operator swaps strings of RDs without any project or resource selection between two individuals.

The individuals generated from GA operators can have resource infeasibility caused by the corresponding RD values. These infeasibilities (if any exist) are corrected by decreasing the RDs of the projects according to the general resource capacities.

CA for RD is a local improvement heuristic which utilizes the preference concept. The preference of a resource can be thought of as the value of the resource for the project or a criterion for the amount of improvement that will be seen in the objective, if an additional unit of that resource is employed. The basic rationale behind CA for the RD procedure is moving the resource state of the project to a more preferable state, which will result in a solution at least as good as the solution of the previous one. Preferences of project for the resources are represented with  $p_{vk}$  and  $p_{vi}$  values for renewable and non-renewable resources, respectively. There are two different methods proposed by Besikci et al. (2013) to obtain the preferences: the first one is based on linear relaxation of the MRCPSp model; the other one, on a Lagrangian relaxation of the model. In this paper, the Lagrangian relaxation based preference calculation is used, since it is shown to lead to overall better preferences of projects for the resources (Besikci et al., 2013). The Lagrangian relaxation based preference calculation employs a modified Lagrangian relaxation formulation of MRCPSp, where the renewable and nonrenewable resource constraints are relaxed. The values of the Lagrangian coefficients, after a one step sub-gradient optimization, are taken as the preferences of



the projects for the resources.

### *Solving MRCPSP for Each Project*

*CPLEX 11.2* is used to solve the MRCPSP for each project. Since CA for RD is applied repeatedly through the execution of GA, some modifications are needed to reduce the run time of MRCPSP for each project. There are two basic tasks when solving an MRCPSP with *CPLEX*: model generation and executing the solution procedure.

When solving MRCPSP repeatedly for different RD instances for a given project, the basic difference is the renewable and nonrenewable resource capacity values. Thus, a generic model can be used to represent a project and when a solution is needed for a given RD instance for the corresponding project, then the only required task is changing the right hand side of the renewable and nonrenewable resource constraints. This modification can easily be achieved by defining a decision variable for each renewable and nonrenewable resource and updating them before solving the model. Note that to use generic models, a long time horizon ( $T_v$  for each project), which can cover all the possible renewable and nonrenewable resource capacity values, is needed. This time period is calculated at initialization phase by determining the minimum renewable and nonrenewable resource requirements for each resource and applying the Simulated Annealing approach proposed by Bouleimen and Lecocq (1998). The minimum renewable resource requirement is calculated by finding the minimum renewable resource requirement for each project over all modes and then taking the maximum of these values over all projects. Then the minimum nonrenewable resource requirement can be calculated from the selected modes in the aforementioned procedure. However, this long time horizon results in a very large set of activity finish decision variables. To compensate for the solution time inefficiency caused by this large decision variable set, some modifications are made. First of all, all solved cases for each project are stored, and whenever the same RD instance

is encountered for a project, then the stored solution is used for that case. In addition, the number of the activity finish decision variables are reduced by giving upper bounds to the corresponding generic model employing resource dominance cases. When an RD instance for a given project has higher resource capacity values for all renewable and nonrenewable resources than one of the stored solved problems, which we call a resource dominance case, then the makespan of the corresponding solution is given as an upper bound to the generic model. This bounding is achieved by limiting the makespan of the project with this upper bound, and *CPLEX* eliminates the redundant decision variables in the pre-processing phase.

#### *4.1.4. General Resource Capacities Space Search*

The RPS search is implemented on the general resource capacities part of the individual by employing GA operators defined for RPP and a new improvement heuristic, CA for the RP, which are explained below in detail.

##### *Genetic Algorithm Operators*

To search through the general resource capacities space a mutation and a crossover operator are defined. The mutation operation randomly swaps two general resource capacity values for a given individual. In crossover operation two different individuals are selected and between these individuals, two random general resource capacity values are swapped. After the application of GA operators, an individual can become infeasible according to the general resource budget and the RD totals. The general resource budget infeasibility is corrected by changing the general resource capacities of the resource that are not swapped in the first place. If the infeasibility of the general resource capacities cannot be corrected by only decreasing the general resource capacities of the unchanged resources, then the resources affected by the GA operators are accordingly decreased. Similarly, when there is an RD infeasibility, the RD values are adjusted in proportion to the general resource capacities. The probabilities for each crossover and mutation type operators

as well as CA for RD are set to 0.1.

*Combinatorial Auction for Resource Portfolio Problem*

CA for RP is based on preferences for the general resources. The application of CA for RP is similar to CA for RD. With the preferences and slack budget (difference between given general budget and budget value of the general resource capacity instance) at hand, the slack budget is distributed among the general resource capacities according to the preferences. The amount of budget that will be used for different resources is determined using a bounded knapsack model similar to the one used in CA for RD where preferences are profits and slack budget is the knapsack capacity. The key point of the algorithm is the calculation of preferences for the general resource capacities.

To determine the preferences for the general resources, the preferences obtained from the CA for RD are utilized. The preference for a general resource is calculated as the sum of the preferences of the individual projects for that resource as follows.

$$g_k = \sum_{v=1}^V p_{vk} \quad \text{Preference for general renewable resource } k \quad (12)$$

$$g_i = \sum_{v=1}^V p_{vi} \quad \text{Preference for general nonrenewable resource } i \quad (13)$$

where  $p_{vk}$  and  $p_{vi}$  are preferences of project  $v$  for renewable resource  $k$  and nonrenewable resource  $i$ , respectively.

The slack for general resource capacities can easily be calculated from the solution of individual MRCPSP for each project. The difference between the general budget and the total amount of resource used by all the projects (corrected with resource prices) will give the slack budget. Combining all this information provides the following knapsack model.

$$\max z = \sum_{i=1}^I y_i g_i + \sum_{k=1}^K y_k g_k \quad (14)$$

Subject to

$$\sum_{i=1}^I cw_i y_i + \sum_{k=1}^K cr_k y_k \leq b \quad (15)$$

$$y_i \leq a_i \quad (16)$$

$$y_k \leq a_k \quad (17)$$

$$y_i \text{ and } y_k \in Z^+ \quad (18)$$

where  $y_i$  ( $y_k$ ) is the positive decision variable for the amount of resource capacity allocated to nonrenewable resource  $i$  (renewable resource  $k$ );  $a_i$  and  $a_k$  are the upper limits for the transferred nonrenewable and renewable resources, respectively, calculated from the current general resource capacities and no-delay general resource requirements and  $b$  is the slack budget. The knapsack model is solved using CPLEX, which has a small variable set (total number of renewable and nonrenewable resources) and easy to solve.

With the results of the above knapsack model the general resource capacities are updated so that the unused general resource budget is transferred to the resources with high preference values. The RD values for each project are also updated according to the new general resource capacities with respect to the preferences of projects for resources. A new solution is obtained by solving the MRCPSPP for each project with the new RD values.

#### 4.1.5. Execution of the Two-Phase GA

The GA initially starts with  $ZxH$  individuals generated as described in Section 4.1.2. RD sub-populations generated from a general resource capacity instance are subject to only RDS search until convergence. In other words, only RD parts of individuals in the sub-populations are evolved. A specific RD sub-population is said to have converged if no improvement is observed for a specific number of generations for an RD sub-population. Whenever an RD sub-population converges, the distinct individuals within it are migrated

into the RP population for phase two. The rationale behind applying RPS search after the convergence of an RD population is the possibility to obtain better individuals with RPS search operators applied on a evolved RD part. For CA for RP heuristic, it is very important to have “good” preferences of projects for resources since those values are used to calculate general resource preferences. With a converged RD part, the preferences of general resource capacities will be able to value the value of a general resource with respect to the needs of the projects. Thus, especially for CA for RP, it is important to have a converged RD part, which would have useful and evolved information that the heuristic can use. Note that until all RD sub-populations converge, RD sub-populations and RP population run parallel.

The evolution in the RP population involves RPS search as well as RDS search. In the second phase, along with GA operators for both RD and RP, CA for RP is employed every time CA for RD is employed.

For the selection strategy, the best individuals among the current population and the newly generated individuals according to their fitness values are included in the next generation. To compensate for the intensifying effect of this selection strategy, the individuals for crossover, mutation and CA operators are randomly selected as a diversification strategy.

The GA parameters in the test runs are as follows. As defined above, the size of the population is  $ZxH$ . In the preliminary runs, it is observed that when  $Z$  is lower than 8 and  $H$  is lower than 10, GA does not give good overall results and when corresponding parameters are greater than 8 and 10, respectively, the algorithm does not improve. Thus the population size is determined as 80. Based on preliminary test runs, the probabilities for each crossover and mutation operator and as well as CA operators are taken as 0.1. GA approaches are terminated when the best solution in the population does not change for 10 iterations or within 180 minutes, whichever is reached first.

The execution of the two-phase GA is summarized below and the general

structure of the two-phase GA is depicted in Figure 1.

*Initialization of the Sub-populations:* Initialize the general resource capacity instances according to the general resource budget and RD instances according to the general resource capacity instances. This procedure is shown as the *Initialization* in Figure 1.

*Step 1:* Apply steps 1.1-1.3 to generate general resource capacity instances.

*Step 1.1:* Set general resource capacities proportional to no-delay resource requirement totals of the projects.

*Step 1.2:* For each resource, determine the capacity equal to the no-delay requirement totals of all projects, set capacities for remaining resources proportional to no-delay resource requirements of the projects.

*Step 1.3:* Generate random general resource capacity instances such that the budget is not exceeded.

*Step 2:* Apply steps 2.1-2.3 for each general resource capacity instance created to obtain individuals for RD sub-populations.

*Step 2.1:* Set RDs proportional to no-delay resource requirements of the projects.

*Step 2.2:* For each project set the project as the selected project, dedicate no-delay resource requirements to the selected project. For the remaining projects generate RDs proportional to no-delay resource requirements of the projects.

*Step 2.3:* Generate random RDs.

*Search:* Run until allowed execution time of 180 minutes is reached or the best solution in the population does not change for 10 iterations

*Phase-I RD Space Search:* Apply the RDS search operators in RD populations and RP population.

*Step 1:* Apply crossover operations for RD

*Step 2:* Apply mutation operations for RD

*Step 3: Apply CA for RD*

*RD Space Convergence Check:* For each RD population check convergence. If an RD population has converged, move distinct instances to RP population after applying CA for RP to each individual.

*Phase-II General Research Capacity Space Search:* For each individual in the RP population, execute the following steps.

*Step 1: Apply crossover operation for general resource capacities*

*Step 2: Apply mutation operation for general resource capacities*

*Step 3: Apply CA for RP, if CA for RD has been applied to the individual*

*Report:* Report the resulting schedules when the algorithm terminates.

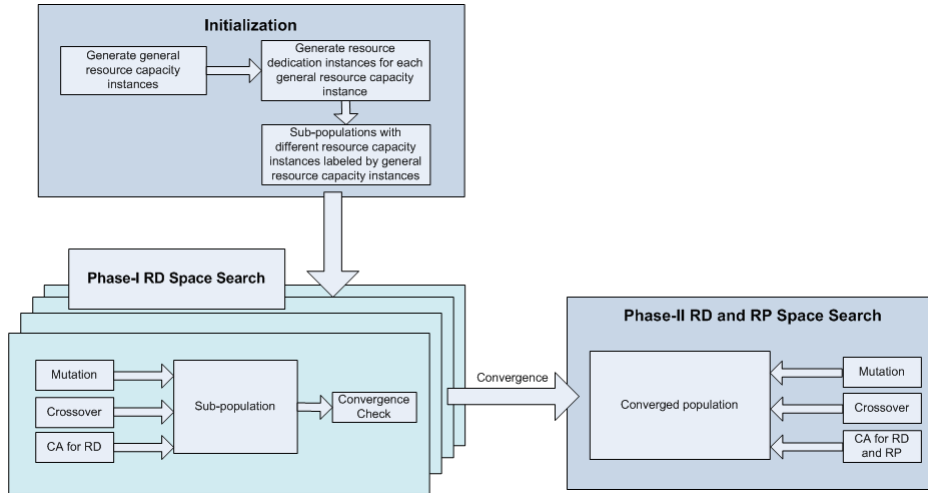


Figure 1: General structure of the two-phase GA

#### 4.2. A Monolithic GA with Simultaneous RD and RP Space Search

In adopting a monolithic view to RPP, a simultaneous execution mode for the proposed GA operators and the CA for RP is introduced. In this execution mode, the individuals in the population are evolved applying general resource capacities space search (mutation, crossover and CA for RPP)

simultaneously with RDS search. In this approach, CA for RP is applied to individuals, which have completed a CA for RD iteration.

## 5. Experimental Results

The solution approaches for RPP are tested using a series of test problems. Two different measures characterize and group test problems: network complexity (NC) and maximum utilization factor (MUF) (Kurtuluş and Narula , 1985).

When an activity-on-node representation is used, NC is defined as the number of arcs divided by the total number of nodes. MUF is calculated as the ratio of the no-delay schedule resource requirement and the available resource. If MUF value is less than or equal to one, then the resource capacity is sufficient to obtain a no-delay schedule. To have a meaningful MUF measure for the RD policy, the calculation of no-delay resource requirement of the multi-project problem is calculated as the sum of no-delay resource requirements of the individual projects. In addition, in the proposed multi-project scheduling environment, the resource capacities are not readily present but determined from a given resource budget. Thus, MUF calculation is modified by calculating the budget value of no-delay resource requirement divided by the general resource budget. If MUF value is equal to 1, then the general resource budget is sufficient to construct a general resource capacity instance that will allow a no-delay schedule for all projects. Similarly when MUF value is increased, then the general resource budget becomes tight.

Multi-project problems are generated by combining 6 projects either from j20 or j30 sets of PSPLIB (<http://129.187.106.231/psplib/>) (Kolish and Sprecher , 1996). The modes of the activities of the projects are modified to obtain a mode set that does not have dominated modes, such that a mode with higher total resource cost has always a lower duration, but the general network structure is kept as it is. The problem sets have four resources, two being renewable and two nonrenewable. From this resource set, one of each



resource type is selected as more costly and leading to a faster processing time but their prices are higher than those of the other resources of the same type.

Four different modes are generated with different durations and resource usages as follows. The fastest mode has the highest cost and it has resource consumption from all of the resources. The second fastest mode has a major amount of resource consumption of only costly resources. The third fastest mode uses only a moderate amount of the costly resources. Lastly, the slowest mode employs only the cheap resources. The mode generation is depicted in Table 3. The values of  $a$ ,  $b$ ,  $c$ , and  $d$  are selected as the minimum resource requirements and  $f$  is the minimum duration over all modes of the corresponding activity of the original data set taken from PSPLIB. The durations and resource usages are corrected according to the total resource costs of modes but have random components  $e$  and  $f$  with a relatively small magnitude, where  $e$  is uniformly distributed between 0 and corresponding component (for example while calculating the resource requirement of R2 for M2  $e$  is  $U[0-a]$ ) and  $f$  is uniformly distributed between  $U[0-x/2]$ . The modifications in the resource usage and durations of the modes are made to obtain a nondominated mode set with respect to the total resource costs. Bold letters indicate the relatively costly resources.

Table 3: General resource usage and duration characteristics of the modes

Modes	Resource Usages				Duration
	<b>R1</b>	R2	<b>W1</b>	W2	
M1	2a	b	2c	d	x
M2	2a±e	0	2c±e	0	x+f
M3	a±e	0	c±e	0	2x-f
M4	0	b±e	0	c±e	3x

The test problems are grouped according to different levels of number of activities, NC and MUF values. Two levels of NC (1.4 and 1.8) and four

different levels of MUF (1.4, 1.5, 1.6 and 1.7) are defined and a full factorial design with 10 problems in each combination generated for projects with 22 and 32 activities. Test runs demonstrate that when MUF values are lower than 1.4 the problem approaches the unconstrained case, a consequence which is of no interest here. MUF values higher than 1.7 lead to infeasibility for most of the cases for exact solution approaches. So as to compare different solution approaches for different problem characteristics, a base problem group with 10 problems is generated with NC value of 1.8 and MUF value of 1 for multi-project problems with 22 and 32 activities. From this base set, problems with different NC values are generated by randomly deleting precedence relations between different activities. Similarly problems with different MUF values are generated accordingly by adjusting the general resource budget. For example, the problem sets with activity count 22-NC 1.4-MUF 1.5 and activity count 22-NC 1.8-MUF 1.5 have the same mode structure and general resource budget but the previous combination has some of its precedence relations deleted to achieve an NC value of 1.4.

The due dates for projects are calculated to achieve a positive weighted tardiness value. The project with the highest weight has its due date as the calculated makespan of the unconstrained case using critical path methodology, which is named as the no-delay due date. The due dates become less than the no-delay due date as the weight of the projects decreases. The total weighted tardiness value calculated with no-delay due dates and assigned due dates gives a lower bound for the problem. Table 4 reports due date and weight assignments of projects (Besikci et al., 2013).

The data set used for this study can be downloaded from the following link: "<http://www.bufaim.boun.edu.tr/rpp-rd-dataset.zip>"

Results are presented in Tables 5 and 6 and Figures 2 and 3 for projects with 22 and 32 activities, respectively. The Two-Phase GA column refers to the GA that employs a two-phase search for RD and RP spaces. The Monolithic GA column refers to the monolithic approach. The Exact column is for

Table 4: Possible minimum weighted tardiness values for individual projects

Project	Due date	Weight	Possible Min. WT
Project1	No-delay due date	6	0
Project2	No-delay due date - 1	5	5
Project3	No-delay due date - 2	4	8
Project4	No-delay due date - 3	3	9
Project5	No-delay due date - 4	2	8
Project6	No-delay due date - 5	1	5
Possible Min. Total WT			35

the exact solution approach for the given mathematical formulation employing CPLEX 11.2. Every row in the tables represents a problem group and is identified with the corresponding entry in the NC-MUF column, which shows the corresponding network complexity and maximum utilization factors used. There are 10 problem instances in a problem group. The following notation is employed in Tables 3 and 4. AWT stands for the average weighted tardiness for a problem group. NS indicates the case where no feasible solution can be reached when employing the exact solution approach. Values under the ART column report the average execution time of the solution approaches in minutes for a problem group. All of the solution approaches have a run time limit of 180 minutes. Furthermore, the GA approaches terminate, if the best solution is not improved for 10 iterations. The OS column shows the number of instances in a problem group for which the optimal solution is found. For the exact solution approach, if  $OS + NS \neq 10$ , then the difference is the number of incumbent solutions (a feasible solution that is not proven to be optimal). Note that when a solution approach cannot find any solutions for a problem group, then AWT is labeled as not available (NA).

The results can be examined according to the solution quality and time and then compared using the paired t-test. It is intuitive and obvious that MUF is the most significant factor for the difficulty of the problem instances. For the relatively easy problems (with MUF values 1.4), all solution approaches can find the optimal solution in a reasonable time. With moderate

Table 5: Results for problem groups containing 6 projects with 22 activities

NC-MUF	Monolithic GA				Two-Phase GA				Exact			
	AWT	ART	OS	NS	AWT	ART	OS	NS	AWT	ART	OS	NS
1.4-1.4	42.60	49.18	7	0	36.00	36.11	8	0	36	3.31	8	0
1.4-1.5	61.60	96.14	2	0	47.70	84.58	8	0	48.81	108.48	8	1
1.4-1.6	98.8	157.3	0	0	67.00	127.7	1	0	88.75	180	2	6
1.4-1.7	138.5	162.4	0	0	104	135.5	1	0	109	180	2	8
1.8-1.4	44.50	38.51	7	0	39.40	32.31	9	0	39.40	21.12	9	1
1.8-1.5	71.40	91.39	0	0	51.80	76.96	7	0	51.80	105.55	7	3
1.8-1.6	105.50	171.4	0	0	72.90	144.90	1	0	92.20	180	2	7
1.8-1.7	130.80	167.60	0	0	99.70	144.40	1	0	119.70	180	1	9

Table 6: Results for problem groups containing 6 projects with 32 activities

NC-MUF	Monolithic GA				Two-Phase GA				Exact			
	AWT	ART	OS	NS	AWT	ART	OS	NS	AWT	ART	OS	NS
1.4-1.4	35.00	19.55	10	0	35.00	18.56	10	0	35	14.8	10	0
1.4-1.5	53.30	111.8	0	0	35.60	98.42	7	0	41.35	114.73	6	4
1.4-1.6	93.00	170.6	0	0	62.40	144.80	0	0	NA	180	0	10
1.4-1.7	106.2	163.8	0	0	73.80	133.80	0	0	NA	180	0	10
1.8-1.4	51.55	47.33	8	0	35.00	49.65	10	0	35	62.13	10	0
1.8-1.5	82.17	85.69	4	0	39.20	73.17	7	0	43.03	118.09	6	4
1.8-1.6	100.30	173.3	0	0	63.30	135.70	0	0	NA	180	0	10
1.8-1.7	114.70	166	0	0	82.60	146.90	0	0	NA	180	0	10

MUF values (1.5), the monolithic GA with simultaneous RD and RP search and the exact solution approach start to fall back, when compared to the two-phase GA. For the hardest problems of the set (with MUF values 1.6 and 1.7), the gap widens between the two-phase GA and the other solution approaches. Note that the exact solution approach cannot find results in the given time limit for a significant number of problems for projects with 22 activities and for all of the problem instances for projects with 32 activities. To summarize the results for solution quality, one can say that two-phase GA gives significantly better results than does the monolithic GA, a finding which shows the benefit of exploring the problem space in different phases.

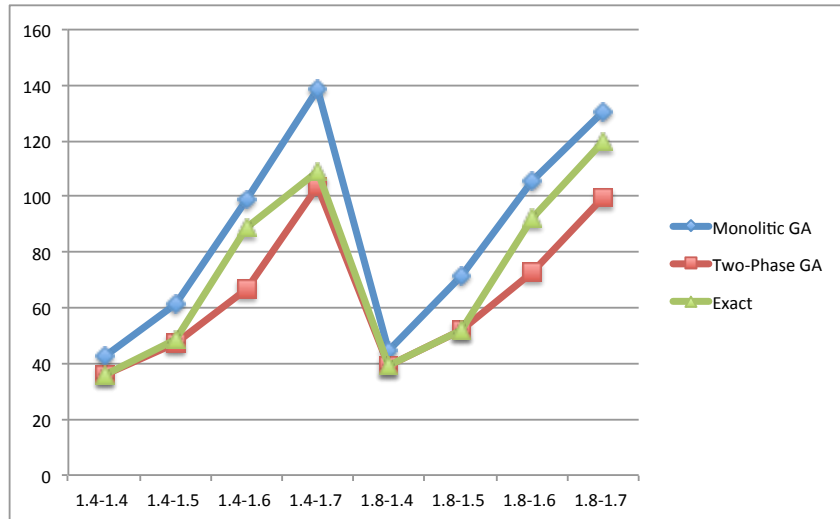


Figure 2: Results in graphical format for problem groups containing 6 projects with 22 activities (AWT for each problem group)

Furthermore, for relatively easy problems, the two-phase GA gives competitive results when compared with the exact solution approach. For problem instances with higher MUF values, the two-phase GA has a clear solution quality advantage over the exact solution approach, which even fails to find feasible solutions for most of the cases.

When the results are compared according to the solution times for the relatively easy problems (with MUF values 1.4), it can be seen that the exact solution approach is significantly better than the GA approaches. This can be explained by the heavy initialization cost for the GA approaches, which does not pay off for the easy cases. For MUF values 1.5, 1.6 and 1.7, the two-phase GA is significantly better than the remaining approaches with respect to solution times. This finding indicates that the two-phase GA approach explores the solution spaces more effectively than the monolithic GA with simultaneous space search. In other words, searching RP space after the RDS search has converged improves the execution time of GA. Additionally, note that problems with 32 activities have a higher solution time in general as

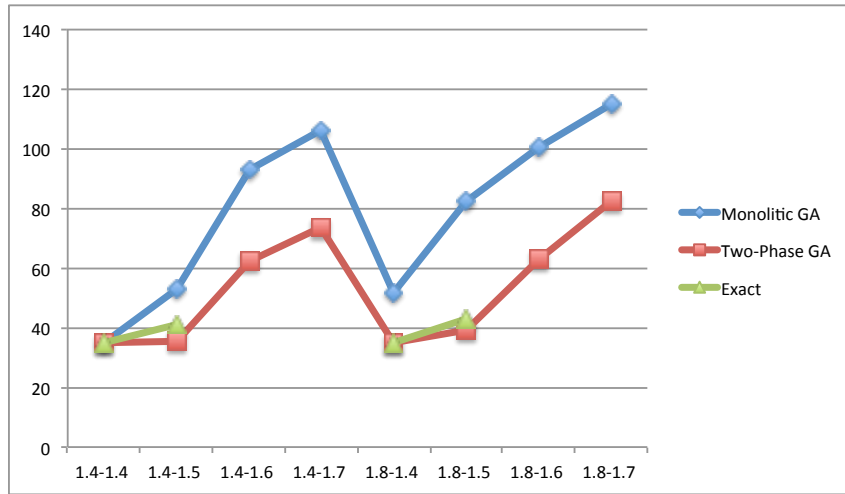


Figure 3: Results in graphical format for problem groups containing 6 projects with 32 activities (AWT for each problem group)

expected, because of the problem size.

## 6. Summary and Conclusions

This paper presents a new approach for multi-project scheduling environments where general resource capacities are determined from a general resource budget and the resources in the multi-project environment cannot be shared among projects and must be dedicated. The general mathematical formulation of RPP is proposed and a new improvement heuristic is defined and employed in two different GA approaches for the solution of RPP. The difference between the proposed GA approaches is related with the way of exploring the RD and RP spaces. As the name implies, the simultaneous GA approach searches RD and RP spaces simultaneously. The two-phase GA approach first searches the RDS for different general resource capacity instances until the RD part of the individuals converge. After convergence, the individuals are also subject to RP space search. The rationale behind this two-phase approach for GA is to obtain better preferences for general resources from converged RD values of the individual projects. The experi-

mental results show that, the two-phase GA approach gives overall the best results among the proposed solution approaches with respect to solution time and quality.

An important aspect of the proposed solution procedure lies in its usefulness to the decision maker at various managerial levels. The procedure yields different types of information related with the resource capacity decisions in a multi-project problem environment. First of all, the general resource capacity values correspond to the strategic resource planning decisions. Additionally, other implicit information related with these general resource capacities is produced in the solution procedure. For instance, the preference values of general resource capacities show the relative importance of resources with respect to the projects at hand, and hence, indicate the criticality of the resources in case of a disruption in availability of these resources. Another important result is that the RD values correspond to rough-cut capacity planning for our problem environment. Finally, the solution also gives the detailed scheduling for each project, which can be used as a baseline schedule.

With Besikci et al. (2013) and this paper, the RD concept is extensively studied. RD is a resource management policy, which needs further attention. Obviously, the RD concept does not cover all the possible ways of using resources thus, different resource management approaches such as a combination of RD and RS approaches can be the first extension. Another future research would be using different objective functions such as the net present value. Additionally, although the due dates are assumed to be given in RPP, the procedure presented here can be adopted to support the process of due date determination. This can be achieved by solving RPP under different due date scenarios. In the RPP considered here, the projects are given. A further decision component can be added to the RPP by introducing project portfolio selection to the problem environment.

*Acknowledgements:* This work is supported by Scientific and Technological Research Council of Turkey (TUBITAK-MAG 109M571) and Bogazici

University Scientific Research Projects (BAP-09HA302D)

## References

- Besikci, U., Bilge, U., and Ulusoy, G. (2013). Resource dedication problem in a multi-project environment. *Flexible Services and Manufacturing Journal*, **25(1)**: 206-229.
- Bouleimen, K. and Lecocq, H. (2003). A new efficient simulated annealing algorithm for the resource constrained project scheduling problem and its multiple mode version. *European Journal of Operational Research*, **149(2)**: 268-281.
- Demeulemeester, E., (1995). Minimizing resource availability costs in time-limited project networks. *Management Science*, **41(10)**:, 1590-1598.
- Gademann, N. and Schutten, M. (2004). Linear programming based heuristics for project capacity planning. *IIE Transactions*, **37**: 153-165.
- Hans, E. W., Herroelen, W., Leus, R. and Wullink, G. (2007). A hierarchical approach to multi-project planning under uncertainty. *Omega*, **35**: 563-577.
- Hartmann, S. (2001). Project Scheduling with Multiple Modes: A Genetic Algorithm, *Annals of Operations Research*, **102**: 111-135.
- Hartmann, S., Briskorn, D., (2010). A survey of variants and extensions of the resource-constrained project scheduling problem. *European Journal of Operational Research*, **207**: 1-14.
- Herroelen, W., (2005). Project scheduling - theory and practice. *Production and Operations Management*, **14(4)**: 413-432.
- Kolisch, R. and Sprecher, A. (1996). PSPLIB - A project scheduling problem library. *European Journal of Operational Research*, **96**: 205-216.



- Krüger, D. and Scholl, A. (2009). A heuristic solution framework for the resource constrained (multi-)project scheduling problem with sequence dependent transfer times. *European Journal of Operational Research*, **197**: 492-508.
- Kurtuluş, I. S. and Narula S. C. (1985). Multi-project scheduling: Analysis of project performance. *IIE Transactions* **17**: 58-66.
- Liberatore, J. M., Pollack-Johnson, B., Smith, A. S. (2001). Project Management in construction: software use and research directions. *Journal of Construction Engineering and Management*, **127**: 101-107.
- Lova, A., Maroto C., Tormos P. (2000). A multi criteria heuristic method to improve resource allocation in multi project scheduling. *European Journal of Operational Research*, **127**: 408-424.
- Lova, A., Tormos, P., Cervantes, P., Barber, F. (2006). An efficient hybrid genetic algorithm for scheduling projects with resource constraints and multiple execution modes. *International Journal of Production Economics*, **117(2)**: 302-316.
- Möhring, R. H. (1984). Minimizing costs of resource requirements in project networks subject to a fixed completion time. *Operations Research*, **32**: 89-120.
- Payne, J. H. (1995). Management of multiple simultaneous projects: a state-of-the-art review. *International Journal of Project Management*, **13(3)**: 163-168.
- Peteghem, V. 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*, **235**: 62-72.

- Speranza, M. G. and Vercellis C. (1993). Hierarchical Models for Multi-Project Planning and Scheduling. *European Journal of Operational Research*, **64**: 312-325.
- Talbot, F. B. (1982). Resource-constrained project scheduling with time-resource tradeoffs: the nonpreemptive case. *Management Science*, **28(10)**: 1199-1210.
- Weglarz, J., Jozefowska, J., Mika, M., Waligora, G. (2011). Project scheduling with finite or infinite number of activity processing modes - A survey. *European Journal of Operational Research*, **208(10)**: 177-205.