International Conference
on Industrial Engineering and Systems Management
IESM'2013
October 28 - October 30
RABAT - MOROCCO

A Combination of Different Resource Management Policies in a Multi-Project Environment

Umut BESİKCİ [a], Gündüz ULUSOY [a], Ümit BİLGE [b]

[a] *Faculty of Engineering and Natural Sciences, Sabanci University, Istanbul*

[b] *Faculty of Engineering, Bogazici University, Istanbul*

Abstract

Multi-project problem environments are defined according to the way resources are managed in the problem environment which is called the resource management policy (RMP) in this study. Different resource management policies can be defined according to the characteristics of the projects and/or resources in the problem environment. The most common RMP encountered in the multi-project scheduling literature is the resource sharing policy (RSP) where resources can be shared among projects without any costs or limitations. This policy can be seen as an extreme case since there is a strong assumption of unconstrained resource sharing. Another RMP can be defined as the other extreme such that resources cannot be shared among projects which is called resource the dedication policy (RDP). The last RMP considered in this study is between these two policies where resources are dedicated but can be transferred among projects when a project finishes, the dedicated resources to this project can be transferred to another one starting after the finish of the corresponding project. This RPM is called the resource transfer policy (RTP). In this study we investigate a problem environment where all these three types of RPM are present. Additionally, the general resource capacities are taken as decision variables that are constrained by a given general budget. We call this multi-project environment as the Generalized Resource Portfolio Problem (GRPP). We have investigated this problem and proposed an iterative solution approach based on exact solution methods which determines the general resource capacities from the budget, resource dedications, resource sharing and resource transfer decisions and schedules the individual projects. Computational results for over forty test problems are reported.

Key words: Multi-project scheduling, resource management policies, resource sharing, resource dedication, resource transfer

1 Introduction

In multi-project problem environments, different projects are coupled with resource related constraints in the problem. Thus resource management policy (RMP) is a very important conceptual part of the problem environment since it defines how this coupling is realized. In resource sharing policy (RSP), the sharable resources couple the individual projects for their resource usages over the time periods. In resource dedication policy (RDP), dedicated resources couple the projects with their maximum resource usages. And finally in resource transfer policy (RTP), transferable resources not only couples individual projects with their maximum resource

⋆ This paper was not presented at any other revue. Corresponding author G. Ulusoy. Tel. ++(90) 216 4839503. ++(90) 216 4839550.

Email addresses: umut.besikci@boun.edu.tr (Umut BESİKCİ), gunduz@sabanciuniv.edu (Gündüz ULUSOY), bilge@boun.edu.tr (Ümit BİLGE).

usages but also with the transferred resources among themselves. Here, a resource transfer from a project to another one can only be achieved, if one of the projects finishes before the start of the other one.

RDP is extensively investigated in [3] and [2]. RDP is first introduced in [3] leading to the so- called resource dedication (RD) problem, where the objective is minimizing the total weighted tardiness over all projects. The authors present a mathematical formulation for RD problem and two different solution methodologies; a genetic algorithm (GA) based on combinatorial auction (CA) for RD heuristic and a sub gradient optimization approach. CA for RD is based on preferences of the projects for the resources and improves a given initial solution.

The study [3] is extended to deal with resource portfolio problem (RPP) under RDP by including general resource capacity decisions in [2]. The objective is the minimization of the total weighted tardiness over all projects. A two phased GA is proposed for the problem where in the first phase the RD space is searched and in the second phase RD and resource portfolio (RP) spaces are searched simultaneously. For the RP space search, the authors propose a new improvement move called the CA for RP. CA for RP employs general resource preferences to improve a given initial solution.

RTP is introduced in [1] where the general resource capacities are decisions. The resource transfer from a project to another one is only feasible for the condition that one of the projects finishes before the start of the other one. Thus, to allow resource transfers among projects, the release time of the projects are introduced as a new decision to the problem. A solution procedure based on modification for Branch and Cut (B&C) procedure of *CPLEX* is proposed. The modification includes new branching strategies, heuristic solution procedures and valid inequalities. Note that by including the general resource capacity decision in [2] and [1] a new decision level, which can be thought as rough-cut-capacity planning, is added to the problem environment.

This paper generalizes all the types of RMP in a multi-project problem environment. Sharable, dedicated and transferable resources are present in the multi-project problem environment and all these resource types bring different aspects to the problem environment. Additionally, a general budget is introduced to the problem which further emphasizes the importance of resource related decisions in multi-project problem environments. We call this problem as Generalized Resource Portfolio Problem (GRPP). The multi-project environment leading to GRPP and its mathematical programming formulation is given in Section 2. In iterative solution approach based on exact methods for GRPP is explained in detail in Section 3. Experimental results for over forty problems are provided in Section 4. Section 5 concludes together with suggestions for further research.

2 Generalized Resource Portfolio Problem

GRPP in a multi-project environment is determination of the general resource capacities for a given total resource budget, dedication of dedicated and transferable resources to the projects, setting of the sharable resource usages, determination of the project sequence relations and resource transfers between feasible projects (according to the project sequence relations) in such a way that individual project schedules would result in an optimal solution for a predetermined regular objective function. The objective is taken as the minimization of the total weighted tardiness of the projects, since the weighted tardiness is a relatively common objective function employed in the scheduling problems with due dates and has practical relevance as well. Note that this objective can be replaced by any regular objective function.

The multi-project environment in this paper has the following additional characteristics: The projects are ready to start at time zero but their release times are determined by the corresponding decision variables and each has an assigned due date. No precedence constraints among projects are defined. The due date constraints are soft in the sense that they can be exceeded at the expense of associated tardiness cost. The activities are non-preemptive and have finish-to-start zero time lag. The resource set includes both renewable and nonrenewable resources. As mentioned above, renewable resources are of three different types: i) sharable, ii) dedicated and iii) transferable. The activities can be executed by choosing one of the available resource usage modes and the corresponding activity durations. In this problem environment, uncertainty is not considered. Mathematical programming formulation for the GRPP is given below.

Mathematical Model GRPP:

Sets and Indices:

V	Set of projects, $v \in V$
J_v	Set of activities of project v, $j \in J_v$
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, $k \in K$
KD	Set of dedicated renewable resources
KS	Set of sharable renewable resources
KT	Set of transferable renewable resources
I	Set of nonrenewable resources, $i \in I$
T	Set of time periods
vN	Index of last activity of project 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_{vjkm}	Renewable resource k usage of activity j of project v operating on mode m
w_{vjim}	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
Ω	A big number calculated as the sum of time periods of all projects

Decision Variables:

X_{vjmt}	=	$\begin{cases} 1 \text{ if activity } j \text{ of project } v \text{ operating on mode } m \text{ is finished in period } t \\ 0 \text{ otherwise} \end{cases}$
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
$SR_{vv'k}$	=	Amount of renewable resource k transferred to project v' from project v
$Y_{vv'}$	=	$\begin{cases} 1 \text{ if project } v' \text{ is released after project } v \text{ is finished} \\ 0 \text{ otherwise} \end{cases}$

$$min.\ z = \sum_{v \in V} TC_v \tag{1}$$

Subject to

$$\sum_{m \in M_{vj}} \sum_{t=E_{vj}}^{L_{jv}} X_{vjmt} = 1 \quad \forall\ j \in N_v \text{ and } \forall\ v \in V \tag{2}$$

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

$$\sum_{m \in M_{v'1}} \sum_{t=E_{v'1}}^{L_{v'1}} t X_{v'1mt} - \sum_{m \in M_{vN}} \sum_{t=E_{vN}}^{L_{vN}} t X_{vNmt} \leq \Omega(Y_{vv'}) \quad \forall\ v, v' \in V \tag{4}$$

$$\sum_{m \in M_{vN}} \sum_{t=E_{vN}}^{L_{vN}} t X_{vNmt} - \sum_{m \in M_{v'1}} \sum_{t=E_{v'1}}^{L_{v'1}} t X_{v'1mt} \leq \Omega(1 - Y_{vv'}) \quad \forall\ v, v' \in V \tag{5}$$

$$SR_{vv'k} \leq \Omega(Y_{vv'}) \quad \forall\ v, v' \in V \text{ and } \forall\ k \in KT \tag{6}$$

$$\sum_{j \in N_v} \sum_{m \in M_{vj}} \sum_{q=min\{t,E_{vj}\}}^{max\{t+d_{vjm}-1,L_{vj}\}} r_{vjkm} X_{vjmq} \leq BR_{vk} + \sum_{v' \in V} SR_{v'vk} \quad \forall\ k \in KT\ \forall\ t \in T \text{ and } \forall\ v \in V \tag{7}$$

$$BR_{vk} + \sum_{v' \in V} SR_{v'vk} \geq \sum_{v' \in V} SR_{vv'k} \quad \forall\ k \in KT \text{ and } \forall\ v \in V \tag{8}$$

$$\sum_{j \in N_v} \sum_{m \in M_{vj}} \sum_{q=min\{t,E_{vj}\}}^{max\{t+d_{vjm}-1,L_{vj}\}} r_{vjkm} X_{vjmq} \leq BR_{vk} \quad \forall\ k \in KD\ \forall\ t \in T\ \forall \text{ and } v \in V \tag{9}$$

$$\sum_{v \in V} BR_{vk} \leq R_k \quad \forall\ k \in KT, KD \tag{10}$$

$$\sum_{v \in V} \sum_{j \in J_v} \sum_{m \in M_{vj}} \sum_{q=min\{t,E_{vj}\}}^{max\{t+d_{vjm}-1,L_{vj}\}} r_{vjkm} X_{vjmq} \leq R_k \quad \forall\ k \in KS \text{ and } \forall\ t \in T \tag{11}$$

$$\sum_{j \in N_v} \sum_{m \in M_{vj}} \sum_{t=E_{vj}}^{L_{vj}} w_{vjim} X_{vjmt} \leq BW_{vi} \quad \forall\ i \in I \text{ and } \forall\ v \in V \tag{12}$$

$$\sum_{v \in V} BW_{vi} \leq W_i \quad \forall\ i \in I \tag{13}$$

$$\sum_{i \in I} cw_i W_i + \sum_{k \in K} cr_k R_k \leq tb \tag{14}$$

$$TC_v \geq C_v \Big(\sum_{t=E_{vN}}^{L_{vN}} \sum_{m \in M_{vN}} t X_{vNmt} - dd_v \Big) \quad \forall\ v \in V \tag{15}$$

$$X_{vjmt} \in \{0,1\} \quad \forall\ j \in J, \forall\ t \in T, \forall\ m \in M_{vj} \text{ and } \forall\ v \in V \tag{16}$$

$$BR_{vk},\ BW_{vi},\ R_k,\ W_i,\ TC_v \in Z^+ \quad \forall\ v \in V, \forall\ k \in K \text{ and } \forall\ i \in I \tag{17}$$

$$Y_{vv'} \in \{0,1\} \text{ and } SR_{vv'k} \in Z^+ \quad \forall\ v, v' \in V \text{ and } \forall\ k \in K \tag{18}$$

The objective function (1) is the minimization of the total weighted tardiness cost for all projects. Constraint sets (2) and (3) impose activity finish and precedence relations. Constraint sets (4) and (5) set the decision variable $Y_{vv'}$ to 1, if project v is finished before project v' is released, and to 0 otherwise. Thus, the $SR_{vv'k}$ values will only have positive values, if project v is finished before project v' is released (constraint set (6)). Constraint set (7) limits the transferable renewable resources employed for each project with the dedicated transferable renewable resources and the transferred transferable renewable resources from the other projects. In constraint set (8), the total transferable renewable resource that can be transferred by a project is limited with the total resource dedicated to this project and the total resource it gained from transfers. Constraint set (9) limits the dedicated renewable resource usage of a project with renewable resource dedication decision variables. In constraint set (10) the general transferable and dedicated renewable resource capacities are determined according to the corresponding resource dedication values. Constraint set (11) is for general sharable renewable resource capacity where the corresponding resource usages are calculated for all projects over the shared time periods. Constraint sets (12) and (13) calculate the nonrenewable resource dedication values available for each project

and general nonrenewable resource capacities, respectively. Constraint (14) limits the sum of the total renewable and nonrenewable resource costs with the general resource budget. And finally, constraint set (15) calculates the weighted tardiness value for each project. Constraint sets (16)-(18) define the feasible ranges for the decision variables.

3 An Iterative Solution Approach Based on Exact Methods For Generalized Resource Portfolio Problem

GRPP is a conceptually and computationally complex problem. In our previous works, we have proposed solution approaches for RPP under different RMPs. In GRPP, all these different types of resource are present. The solution approach we propose for GRPP is an iterative approach and different steps of the algorithm are based on our previous studies. The basic rationale for the solution approach is obtaining a solution for a relaxed version of the GRPP and then iteratively make the solution feasible for the original problem. The basic steps of the algorithm are given below, where α is the step size for increasing the budget and Δ is the trash hold value for the closeness of the total budget:

Initialization: Set iteration counter, initialize α and Δ.

Step 1: Relax the problem by removing the corresponding constraints for sharable renewable resources, namely, constraint (11). Solve the relaxed problem with a modified version of the solution procedure proposed in [1] for RPP under RTP. If a feasible solution is obtained, go to *Step 3*; otherwise, go to *Step 2*.

Step 2: Increase the current resource budget by α and go to *Step 1*.

Step 3: Calculate the sharable renewable resource budget requirement (SRR) for the solution obtained in *Step 1* by using the activity finish times and selected modes. The budget requirement of the current solution is given by the dedicated resource requirement (DRR). Calculate δ which shows the budget excess as follows:

$$\delta = \frac{DRR + SRR - TotalBudget}{TotalBudget} \tag{19}$$

If this δ value is positive, then the solution obtained is infeasible with respect to the general budget. If so, then decrease the general budget by *SRR* and go to *Step 1*. If, on the other hand, δ is negative, then this means that there is a feasible solution for the problem and the general resource capacity budget requirement for this solution is less than the general budget. If $\delta \leq \Delta$ then terminate the solution procedure with the solution obtained. Otherwise, increase the current budget by $(\delta - \Delta)$. Finally, increase the iteration counter and check whether the iteration limit is reached. If that is the case, terminate the solution procedure with the best solution at hand. Otherwise go to *Step 1*.

Step 1 of the algorithm needs a detailed explanation since it provides a solution approach for the relaxed problem (without any sharable resource constraints). This solution procedure is based on the approach proposed in [1] with a modification to include both dedicated and transferable resources in the RPP. The next section gives detailed explanation for the solution approach.

3.1 A Modified Branch and Cut Procedure for Dedicated and Transferable Resources

Dedicated and transferable resources are similar to some extend such that during the planning horizon of the projects the resources cannot be shared among projects. The only difference is that a transferable resource can be transferred from a project to another one when one of them is finished before the start of the other one. To solve the RPP with dedicated and transferable resources the solution approach proposed in [1] will be modified accordingly. The solution procedure is based on B&C algorithm of *CPLEX* and some modification in the certain steps of the B&C procedure.

CPLEX employs the well known B&C procedure to solve integer problems [4]. The steps of B&C procedure can be modified by the user by employing special purpose algorithms developed specially for the problem at hand. The control of the above procedures is enabled with callback functions in the *CPLEX*. At every node, when a feasible LP relaxation solution is found for the problem, the associated callback functions for each of the above procedures can be used to control the details of the B&C procedure. Three basic steps will be modified to control the details of the B&C procedure in this study; branching, feasible solution generation and local cut generation. Explanations of these modifications are given in detail below.

3.1.1 Branching Strategies

Branching variable selection is an important part of the B&C procedure and can improve the execution of the procedure greatly. There are different sets of integer and binary decision variables which can be used for branching. In *CPLEX*, branching variable can be selected at each viable node. Here, we will select branching variables to a degree and then leave this procedure to *CPLEX*.

The proposed branching strategy is based on the structure of the mathematical model GRPP. Note that when $Y_{vv'}$ variables are known, the problem reduces to RPP with available resource transfer options according to the values of $Y_{vv'}$ decision variables for transferable resources. In other words, when $Y_{vv'}$ variables are set, the remaining decisions are general resource capacities, resource dedications for dedicated and transferable resources, resource transfers between feasible projects (for projects finishing before the start of another one) and finally scheduling of individual projects. This structure of the problem not only facilitates the search but also allows for different feasible solution generation approaches. To branch on $Y_{vv'}$ decision variables, the branch callback function of *CPLEX* is used and branches are generated from integer infeasible variables from the linear relaxation solution on the node explicitly and fed into B&C procedure of *CPLEX*.

The selection rule for $Y_{vv'}$ variable among the integer infeasible ones is given below:

Initialization: Select all the projects having integer infeasible $Y_{vv'}$ decision variables in a node and add selected projects to set *InP*.

Step 1: Select the projects from *InP* that do not have any sequence relations (i.e., all $Y_{vv'}$ and $Y_{v'v}$ variables are integer infeasible for project v) and add selected projects to set *SP*. If set *SP* is empty, then go to *Step 2*; else go to *Step 4*.

Step 2: Select the projects from *InP* that are the predecessor projects in their sequence relations (i.e. all $Y_{v'v}$ variables are integer feasible and 0 for project v) and add selected projects to set *SP*. If set *SP* is empty, then go to *Step 3*; else go to *Step 4*.

Step 3: Select the projects from *InP* that have integer infeasible $Y_{vv'}$ decision variables and add selected projects to set *SP*, go to *Step 4*.

Step 4: Select the project v from *SP* with maximum weight and select the project v' with minimum weight among the projects that have integer infeasible $Y_{vv'}$ values. Generate two branches with the selected $Y_{vv'}$ decision variable ($Y_{vv'} \leq 0$ and $Y_{vv'} \geq 1$).

This selection procedure gives first priority to projects without any sequence relations, and then to the projects that are predecessor in their sequence relations. Project weights are used for tie breaking.

3.1.2 Feasible Solution Generation Heuristic

Addition of a feasible solution can improve the execution of the B&C procedure since it provides an upper bound for the algorithm. We propose a feasible solution heuristic for the problem based on top of $Y_{vv'}$ branching strategy. Note that when all the $Y_{vv'}$ decision variables are integer feasible, the problem becomes RPP with transferable and dedicated resources. In [1], authors propose a feasible solution procedure for RPP with only transferable resources. We have modified this procedure to obtain a feasible solution heuristic for the problem. Please refer to [1] for the details of the original procedure. The feasible solution heuristic for RPP with dedicated and transferable resources is given below:

Step 1: Determine nonrenewable resource values for each project proportional to their no-delay resource requirements (resource requirements for the unconstrained case). The no-delay resource requirement is the amount of resource required for the no-delay schedule of the project. Calculate the remaining budget and go to *Step 2*

Step 2: Determine dedicated and transferable resource values for each project which does not have any predecessor relations (such that for project v all $Y_{v'v}$ decision variables are 0) proportional to their no-delay resource requirement from the remaining budget. Here for the projects that have successor relation to other projects (such that for project v there is at least one project v' with $Y_{vv'} = 1$) their transferable resource values are calculated considering their successor projects. Go to *Step 3*.

Step 3: Resource transfers among projects are determined with the following mathematical model.

$$min.z = \sum_{v}^{V} \sum_{k}^{K} (ND_{vk} - \sum_{v'}^{V} SR_{v'vk}) \tag{20}$$

Subject to

$$SR_{vv'k} \leq BR'_{vk} Y_{vv'} \quad \forall\ v, v' \in V \text{ and } \forall\ k \in K \tag{21}$$

$$\sum_{v'}^{V} SR_{vv'k} \leq BR'_{vk} \quad v \in V \text{ and } \forall\ k \in K \tag{22}$$

$$\sum_{v'}^{V} SR_{v'vk} \leq ND_{vk} \quad v \in V \text{ and } \forall\ k \in K \tag{23}$$

where BR'_{vk} is the renewable resource dedication value for predecessor projects calculated in *Step 2* and ND_{vk} is the no-delay renewable resource requirement for successor projects. This mathematical model tries to determine resource transfers among available projects such that the deviation between total resource transfers to a project and its no-delay resource requirement is minimized over all projects.

With this mathematical model resource transfer values for each sequential relation is calculated. Go to *Step 4*

Step 4: With dedicated, transferable resource values determined in previous steps, solve a multi-mode resource constrained project scheduling problem (MRCPSP) for each project employing an exact solution approaches and determine their corresponding schedule. Go to *Step 5*.

Step 5: Improve the solution obtained with CA for RD and CA for RP (refer to works [3] and [2] for a detailed explanation for CA for RD and CA for RP, respectively). Go to *Step 6*.

Step 6: Feed the *CPLEX* with the obtained solution.

3.1.3 Valid Inequalities

Valid inequalities can improve the B&C procedure and can have an important impact on the efficiency of the procedure. *CPLEX* generates various valid inequalities based on polyhedral theory. One additional valid inequality, namely sequence valid inequality, will be generated at every viable node whenever a $Y_{vv'}$ decision variable is set to 1:

$$F_{v'} \geq F_v + EP_v Y_{vv'} \qquad \forall (v, v') \in V : Y_{vv'} = 1 \tag{24}$$

where F_v si the release time of project v and EP_v is the possible earliest completion time of project v calculated from CPM.

Note that these valid inequalities can be applied a priori in the problem formulation with the following constraint set:

$$F_{v'} \geq F_v + EP_v + \Omega(y_{vv'} - 1) \qquad \forall\ v, v' \in V \tag{25}$$

This approach will result in different solutions for the LP relaxation at each node which can improve the B&C procedure.

4 Experimental Results

To test the proposed solution approach different multi-project problems are created from the *j20* and *j30* sets in PSPLIB [5]. For a problem 6 projects are used from the corresponding sets. Since there are only two renewable resources in the problem sets, we have introduced another renewable resource to have one resource for each renewable resource type. The resource usage for the added renewable resource in the modes of the

corresponding activities is calculated as the average resource usage of the present renewable resources. Every resource is given an associated cost and modes are slightly modified to achieve a mode set such that the more costly modes have faster execution times than the cheaper modes. Node that the network structure is kept the same.

The solution approach proposed for GRPP is tested with a group of problems. To categorize these problems we have used a project measure proposed by [6], namely Average Utilization Factor (AUF). AUF is defined as the ratio of the no-delay resource requirement of the project and available resource. Since this measure is proposed for sharable resources we have modified it for the transferable and dedicated resources. To calculate AUF for the corresponding resources, we sum up the no-delay resource requirements of projects individually and then calculate the ratio. For sharable resources we calculate the AUF from the combined project network. Note that the final values for AUF is calculated over the budget value of the resource requirements. If AUF value is less than or equal to one, this means that there is enough budget to obtain a no-delay schedule for the problem. When AUF value is greater than one, it can be thought as a measure for the tightness of the budget. We used two levels for AUF: 1.5 and 1.6 and generated 10 problems for each combination.

The parameters used in the iterative solution approach are selected as follows. $\Delta = 0.05$, $\alpha = 0.05$ and finally the iteration limit is selected as 10. Tables 1 and 2 report the results for the multi-project problems with 6 projects each having 22 activities. Tables 3 and 4 show the results for multi-project problems with 6 projects each having 32 activities. The #Iteration column shows the number of iterations that is observed to obtain the reported solution. The δ Value column shows the calculated δ value in the last iteration. Objective function value (OFV) and Lower Bound columns show the corresponding values for the solution of the relaxed problem in the last iteration. Finally, Not Available (NA) states that the procedure could not come up with a feasible solution in 240 minutes for that particular problem.

Solution approaches are coded with *Microsoft Visual Studio 2010 C#* and *CPLEX 11.2* is used. A total of 4096 Mb working memory is allocated and the hard drive is used when this allocation is exceeded. Test runs are carried out on an *Intel Xeon X 5492, 3.40 Ghz* processor

Table 1
Results for the multi-project problems with 6 projects each having 22 activities and AUF = 1.5

Problems	#Iterations	δ Value	OFV	Lower Bound
1	3	-0.00015	43	43
2	3	-0.000068	61	61
3	2	-0.002167	40	40
4	3	-0.00023	35	35
5	4	-0.000132	87	87
6	2	-0.003341	38	38
7	2	-0.000352	35	35
8	2	-0.003469	41	41
9	3	-0.00474	41	41
10	2	-0.007927	38	38

As it can be seen from the results, the iterative solution approach has solved most of the problems. When AUF is increased to 1.6, the solution approach could not find feasible solution for 2 and 3 cases in the problem groups of 22 and 32 activities, respectively. Note that as AUF increases, finding feasible solution becomes more difficult. Two paths to the solution can be observed from the detailed investigation of the results. The first one is a monotonic decrease in the infeasibility of sharable renewable resource usage at each iteration till a feasible solution is obtained. Note that this can be observed when the relaxed problem can obtain a feasible solution at each iteration with the reduced budget. Figure 1 shows the behavior of the δ values at each iteration for an example problem for such a case.

The second path to the solution for the case where feasible solution cannot be obtained in the relaxed problem. In this case, the general budget is increased by α and the relaxed problem is solved again, which might lead to a feasible solution. This behavior is displayed in Figure 2.

Table 2
Results for the multi-project problems with 6 projects each having 22 activities and AUF = 1.6

Problems	#Iterations	δ Value	OFV	Lower Bound
1	2	-0.016837	53	53
2	3	-0.005295	61	61
3	NA	NA	NA	NA
4	3	-0.000667	40	40
5	NA	NA	NA	NA
6	2	-0.011667	54	54
7	2	-0.001888	41	41
8	2	-0.000032	50	50
9	6	-0.003019	53	53
10	2	-0.005589	41	41

Table 3
Results for the multi-project problems with 6 projects each having 32 activities and AUF = 1.5

Problems	#Iterations	δ Value	OFV	Lower Bound
1	3	-0.000216	52	52
2	3	-0.000345	37	37
3	3	-0.000456	44	44
4	4	-0.000042	38	38
5	2	-0.000044	40	40
6	4	0	37	37
7	2	-0.00008	35	35
8	2	-0.000756	35	35
9	2	-0.006788	35	35
10	3	-0.002178	38	38

Table 4
Results for the multi-project problems with 6 projects each having 32 activities and AUF = 1.6

Problems	#Iterations	δ Value	OFV	Lower Bound
1	2	-0.001385	56	56
2	2	-0.003867	36	36
3	4	-0.008789	47	47
4	NA	NA	NA	NA
5	3	-0.006055	38	38
6	4	-0.007645	42	42
7	2	-0.006212	42	42
8	2	-0.009741	45	45
9	NA	NA	NA	NA
10	NA	NA	NA	NA

5 Conclusions and Further Research Topics

We have proposed a generalized multi-project scheduling problem with sharable, dedicated and transferable renewable resources which is called GRPP. The conceptual and computational complexity of the problem prevent the use of exact solution approaches for practical cases. We have proposed an iterative solution approach based on exact solution of the relaxed version of GRPP. To solve this relaxed problem a modified B&C procedure is

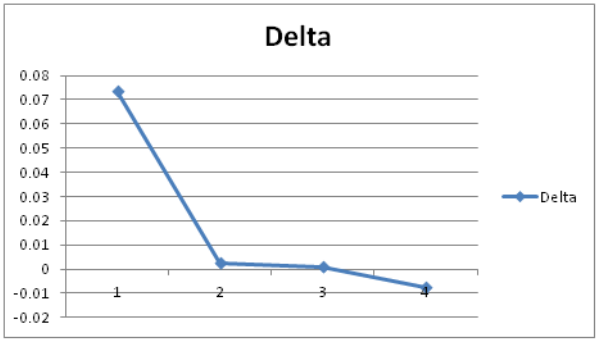

Fig. 1. δ value is decreasing monotonously while a feasible solution is found at each iteration for the relaxed problem

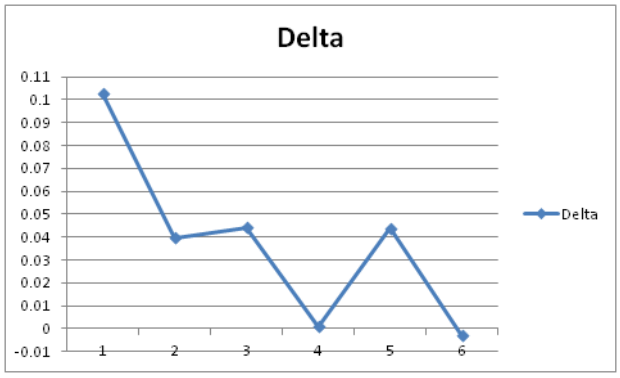

Fig. 2. δ value for the relaxed problem is increased to find a feasible solution

used. The test results show that the proposed solution approach can produce good results for the problems.

The first extension for the study can be the improvement of the solution procedure. When the sharable resources are accounted to the solution and an infeasible resource budget is resulted, feasible solution procedures can be used to search a solution near the neighborhood of the current solution of the relaxed problem. Another research direction would be the consideration of different RMP that we do not consider in this study.

Acknowledgements: We gratefully acknowledge the support given by the Scientific and Technological Research Council of Turkey (TUBITAK) through Project Number MAG 109M571.

References

[1] Umut Besikci, Gunduz Ulusoy, and Umit Bilge. A modified branch and cut procedure for resource portfolio problem under relaxed resource dedication policy. *Working Paper, Sabanci University, Istanbul, Turkey*, 2012.

[2] Umut Besikci, Gunduz Ulusoy, and Umit Bilge. Multi-mode resource constrained multi-project scheduling and resource portfolio problem. *Working Paper, Sabanci University, Istanbul, Turkey*, 2012.

[3] Umut Besikci, Gunduz Ulusoy, and Umit Bilge. Resource dedication problem in a multi-project environment. *Flexible Services and Manufacturing Journal doi: 10.1007/s10696-012-9140-9*, 25(1):206–229, 2013.

[4] ILOG. *ILOG CPLEX Callable Library Reference Manual.* IBM, New York, 2007.

[5] Reiner Kolish and Arno Sprecher. Psplib - a project scheduling problem library. *European Journal of Operaitonal Research*, 96:205–216, 1997.

[6] Ibrahim S. Kurtulus and Subhash C. Narula. Multi-project scheduling: Analysis of project performance. *IIE Transactions*, 17(1):58–66, 1985.