MULTI-PROJECT SCHEDULING WITH 2-STAGE DECOMPOSITION

Anıl Can

Sabancı University, Orhanlı, Tuzla 34956
Istanbul, Turkey
anilcan@sabanciuniv.edu

Gündüz Ulusoy

Sabancı University, Orhanlı, Tuzla 34956
Istanbul, Turkey
gunduz@sabanciuniv.edu

WORKING PAPER

REVISED

NOVEMBER 2012

MULTI-PROJECT SCHEDULING WITH 2-STAGE DECOMPOSITION

Anıl Can

Gündüz Ulusoy

Sabancı University, Orhanlı, Tuzla 34956
Istanbul, Turkey

ABSTRACT

A non-preemptive, zero time lag multi-project scheduling problem with multiple modes and limited renewable and nonrenewable resources is considered. A 2-stage decomposition approach is adopted to formulate the problem as a hierarchy of 0-1 mathematical programming models. At stage one; each project is reduced to a macro-activity with macro-modes. The macro-activities are combined into a single macro-activity network over which the macro-activity scheduling problem (MP) is defined, where the objective is the maximization of the net present value with positive cash flows and renewable resource requirements are time-dependent. An exact solution procedure and a genetic algorithm (GA) approach are proposed for solving MP. GA is also employed to generate an initial solution for the exact solution procedure. The first stage terminates with a post-processing procedure to distribute the remaining resource capacities. Using the start times and the resource profiles obtained in stage one each project is scheduled in stage two for minimum makespan. Three new test problem sets are generated with 81, 84 and 27 problems each and three different configurations of solution procedures are tested.

Keywords: Multiple projects, multiple modes, scheduling, decomposition, genetic algorithms.

1. INTRODUCTION

The resource constrained multi-project scheduling problem with multiple modes (MRCMPSP) is one of the more challenging problems in project management. Among other factors, as a result of the global expansion of the IT sector and the increase in research and development (R&D) and engineering services activities project based management finds more use in practice as a management paradigm. R&D organizations in particular (Liberatore and Titus, 1983) and large construction companies (Liberatore et al., 2001) execute multi-project scheduling procedures regularly. It has been suggested by Payne (1995) that up to 90%, by value, of all projects occurs in a multi-project context. As markets become more competitive, firms' obligation to simultaneously carry out multiple projects by managing the scarce resources becomes even more critical increasing the need to build appropriate management structures accordingly so as to increase their chances to avoid the failures resulting from the decisions taken at different managerial levels. The frequencies, time horizons and details of these decisions make it suitable for a hierarchical management scheme as the one presented by Hans et al. (2007).

One of the arrangements frequently used for managing multiple projects is the dual level management structure (Yang and Sum, 1993), which consists of a higher level manager and a number of project managers. While the project managers work at an operational level and are responsible for scheduling and controlling the activities of individual projects, the higher level manager works on a more tactical level and is responsible for all the projects and project managers. At the higher level, the projects are scheduled as individual entities so as to generate the start times and due dates for each project. Then based on these start times and due dates, each project is scheduled individually employing renewable and non-renewable resource capacities imposed by the higher level. Dual level managerial mechanism also grants a more beneficial position to blend decision approaches with different performance criteria. This also motivated researchers to exploit a similar approach by introducing dual level decomposition methodologies to multi-project planning and scheduling as in Speranza and Vercellis (1993).

This paper is organized as follows. Section 2 provides a brief description of the problem environment and a survey on the related work in the literature. The mathematical models and the solution methodology are presented in section 3. In section 4, a genetic algorithm (GA) for solving multi-mode resource constrained project scheduling problems with discounted cash flows (MRCPSPDCF) and time dependent renewable resource requirements is introduced. Section 5 provides the computational study and the results. In section 6, summary and some suggestions for future work are presented.

2. PROBLEM DESCRIPTION AND RELATED LITERATURE

In this study, a non-preemptive, zero time lag multi-project scheduling problem with multiple modes and limited renewable and non-renewable resources is considered. Each project network is of activity-on-node type with finish-to-start zero time lag type precedence relations. There are no due dates for the projects as well as no precedence relations among the projects. Although the problem is not formulated as a multi-objective programming problem, two different objectives are considered in two consecutive stages. The first stage corresponds to the tactical level aiming to determine the start times of the projects and resource allocation to them such that the net present value (NPV) of cash flows involved is minimized. The second stage corresponds to the operational level of activity scheduling with the objective of minimizing the makespan values of the individual projects employing the results of the first stage. Hence, the tactical and operational levels are both treated within the same model.

Three types of cash flows are employed in this study. *Revenues*: A lump sum payment is made at the completion of each project. *Fixed Costs*: The project fixed costs are resource independent and are incurred initially for each project. *Variable Costs*: The resource usage costs for the renewable and non-renewable resources are incurred periodically throughout each activity. It is assumed that an activity's consumption of the non-renewable resources as well as the variable cost distribution associated with this consumption are uniform over the execution period of that activity. The resource usage cost for a resource is taken to be the same over all projects and over all periods.

The resource constrained multi-project scheduling problem (RCMPSP) consists of a collection of projects which are to be scheduled sharing limited resources. The output consists of the start times of the projects and their activities and the allocation of resources to activities. A large body of literature existing for RCMPSP with or without multiple modes reflects implicitly or explicitly a single level management scheme for the planning and scheduling of multiple projects. A 0-1 linear programming formulation of this problem was first introduced by Pritsker et al. (1969) and three possible objective functions including minimizing total throughput time for all projects, minimizing the time by which all projects are completed, and minimizing the total lateness or lateness penalty for all projects were discussed. Some heuristic sequencing rules introduced by different researchers have been categorized by Kurtulus and Davis (1982). Considering the penalties due to project delays, Kurtulus and Narula (1985) analyzed six penalty functions together with four priority rules and determined that MAXPEN (Maximum Penalty First) rule performed best for minimizing the weighted project delay. Kim and Schniederjans (1989) presented a heuristic framework for RCMPSP and demonstrated a practical application. Bock and Patterson (1990) studied setting due dates by a rule-based heuristic approach and the preemption of resources from one project to another in a multi-project environment. A scheduling heuristic together with an update routine for control purposes is developed by Tsubakitani and Deckro (1990) based on actual housing data. For RCMPSP with the objective of minimizing weighted tardiness costs, Lawrence and Morton (1993) developed a cost-benefit scheduling policy with resource pricing. Lova and Tormos (2001) analyzed the effect of the schedule generation schemes, and some priority rules in multi-project and single-project environments. Kumanan et al. (2006) established a heuristic and a GA for scheduling a multi-project environment with an objective of minimizing the makespan of the projects. Gonçalves et al. (2008) presented a GA for RCMPSP with the chromosome representation based on the random keys and chromosome evaluation using a parameterized active schedule generating heuristic based on the priorities, delay times and release times. Zapata et al. (2008) presented three models that attempt to overcome the limitations caused by the indexing of the task execution modes, the indexing of time periods and the discrete nature of resources. In Mittal and Kanda (2009), new two-phase heuristics for RCMPSP are developed and compared with the existing methods.

Hans et al. (2007) proposed a positioning framework to distinguish between different types of project-driven organizations to aid project management in the choice between the various existing planning approaches. In line with the approach taken here, a group of papers deals with the dual level management approach for planning and scheduling multiple projects. Speranza and Vercellis (1993) suggested a decomposition of the problem into a hierarchy of integer programming models reflecting the dual level project management structure. Yang and Sum (1997) following their work mentioned above (Yang and Sum, 1993) examined the performance of due date, resource allocation, project release, and activity scheduling rules in a multi-project environment. For the decentralized version of RCMPSP, in which local and autonomous decision makers (project managers) contribute to decision making, some multi-agent system based solution procedures are discussed as in Lee et al. (2003), Confessore et al. (2007), Homberger (2007), and Homberger (2010).

Here, we aim to develop an effective and viable 2-stage decomposition approach reflecting the dual level project management structure and based on the concepts of macro-activity and macro-mode introduced by Speranza and Vercellis (1993).

3. SOLUTION APPROACH

Due to the complexity of the problem at hand, a 2-stage decomposition approach is applied as an approximation to it. The problem is formulated as a hierarchy of 0-1 mathematical programming models in two stages. In the first stage, each project is transformed into a macro-activity and different macro-modes are formed by evaluating various combinations of resource allocation through solving single project multi-mode resource constrained project scheduling problems (MRCPSP) with a budget based on the resource usage cost involved. After the macro-modes are determined, a proper time horizon is generated to build a macro-activity model with the objective of NPV maximization. The macro-activities representing individual projects are scheduled subject to the general resource capacities with the objective of maximizing NPV of the discounted cash flows involved. The problem of scheduling of macro-activities is a special kind of MRCPSP with discounted cash flows (MRCPSPDCF), where the cash flows are positive and renewable resource requirements are time dependent. A GA approach is designed for solving this problem. In the computational studies, this GA approach is also employed for generating starting solutions for the exact solution procedure. The result of the first stage is subjected to a post-processing procedure to distribute the remaining resource capacities. As a result of the first stage, the start times and the resource allocations for the projects are determined by the start times of the macro-activities and by the selection of the macro-modes. Using the start times and resource profiles obtained in stage one, each project is scheduled in stage two for minimum makespan. These two objectives employed separately in two consecutive stages reflect a multi-objective environment. For the single project scheduling problems the resource availabilities may differ from period to period. The resource constraints in stage two are tight constraints making the problems computationally easier to solve. The flow of the proposed 2-stage decomposition procedure is summarized in Figure 1.

Place Figure 1 about here

The sets, indices and parameters used in the models presented are listed below.

Sets and Indices:S = set of all projects

s : project indices

V_s : set of activities in project s

i, k : activity indices

z_s : finishing activity of project s; $z_s \in V_s$

P_s : set of precedence relations between all activities $i \in V_s$ in project s

M_{si}: set of modes of activity i of project s

j : activity execution mode indices; $j \in M_{si} = \{1, \dots, |M_{si}|\}$

$\widetilde{M}_s$: set of the macro-modes for project s

v : macro-mode indices; $v \in \widetilde{M}_s = \{1, \dots, |\widetilde{M}_s|\}$

R : set of renewable resources

r : renewable resource indices; $r \in R = \{1, \dots, |R|\}$

N : set of non-renewable resources

n : non-renewable resource indices; $n \in N = \{1, \dots, |N|\}$

$\mathcal{T}$: set of periods

$\mathcal{T}_s$: set of periods for project s

t, θ : period indices

Parameters:

α : discount rate

d_{ij} : processing time for activity i performed employing mode j

$\tilde{d}_{sv}$: processing time for macro-activity s performed employing macro-mode v

e_i : early start period for activity i

l_i : late start period for activity i

$\tilde{e}_s$: early start period for macro-activity s

$\tilde{l}_s$: late start period for macro-activity s

κ_s : artificial budget for project s employed in its macro-mode generation process

W_r : amount of renewable resource r available

W_{rt} : amount of renewable resource r available in period t

Q_n : amount of non-renewable resource n available

w_{ijr} : amount of renewable resource r utilized by activity i performed in mode j

ω_{svrt} : amount of renewable resource r utilized by macro-activity s performed in mode v in period t

q_{ijn} : amount of non-renewable resource n consumed by activity i performed in mode j

η_{svn} : amount of non-renewable resource r utilized by macro-activity s performed in mode v

C_s^R: lump sum payment made at the completion time of project s

C_s^I : project fixed cost to be incurred initially in order to start project s

a_r : unit resource usage cost of utilizing one unit of renewable resource r for one period

b_n : resource usage cost of consuming one unit of non-renewable resource n

g_{ij}: resource usage cost for activity i performed in mode j

3.1 Macro-Mode Generation

When generating the macro-modes, it is extremely significant to balance the trade-off between the diversity of the macro-modes and the size of the macro-activity scheduling model. Although increasing the number of macro-modes increases the number of possible outcomes and thus may lead to a better solution, it also increases the required computational effort. For each project $s\varepsilon S^a$, the corresponding macro-mode generation is performed by solving two interacting mathematical programming models. The first model employed for this purpose, MMG_s^1, is adopted from the shrinking model introduced by Speranza and Vercellis (1993). The second model, MMG_s^2, is introduced as a search systematic for generating the representative macro-modes. The interaction between these two models is explained later in this section.

In the following formulations, e_i and l_i for activity $i \in V_s$ are calculated using the critical path method. For that purpose, the length of the time horizon $\mathcal{T}_s$ for that purpose is determined using the time horizon setting method explained in section 3.2.

Model $\boldsymbol{MMG_s^1}$ ($\forall\ s \in S$)

$$min \quad T_{s,z_s} \tag{1}$$

$$s.t. \quad \sum_{j \in M_{sk}} \sum_{t=e_k}^{l_k} t x_{kjt} \geq \sum_{j \in M_{si}} \sum_{t=e_i}^{l_i} (d_{ij} + t) x_{ijt} \qquad (i,k) \in P_s \tag{2}$$

$$\sum_{i\in V_s}\sum_{j\in M_{si}}\sum_{\theta=\max(e_i,\ t-d_{ij}+1)}^{\min(l_i+d_{ij}-1,\ t)} w_{ijr}x_{ij\theta} \le W_r \qquad r\in R, t\in \mathcal{T}_s \qquad (3)$$

$$\sum_{i\in V_s}\sum_{j\in M_{si}} q_{ijn}\sum_{t=e_i}^{l_i} x_{ijt} \le Q_n \qquad n\in N \qquad (4)$$

$$\sum_{j\in M_{si}}\sum_{t=e_i}^{l_i} x_{ijt} = 1 \qquad i\in V_s \qquad (5)$$

$$\sum_{i\in V_s}\sum_{j\in M_{si}}\sum_{t=e_i}^{l_i} g_{ij}x_{ijt} \le \kappa_s \qquad (6)$$

$$x_{ijt} = \begin{cases} 1, & \text{if activity } i \text{ starts in period } t \text{ using mode } j \\ 0, & o/w \end{cases} \qquad i\in V_s, j\in M_{si}, t\in\mathcal{T}_s \ (7)$$

The objective (1) is the minimization of the makespan for project s denoted by T_{s,z_s}. Constraint sets regarding precedence relations within project s (2), renewable resource capacities (3), nonrenewable resource capacities (4) and assignments (5) are included in Model MMG_s^1. The resource usage costs, g_{ij}, are calculated as in (8) and are constrained by a budget κ_s (6).

$$g_{ij} = \sum_{r\in R} d_{ij}w_{ijr}a_r + \sum_{n\in N} q_{ijn}b_n \qquad i\in V_s, j\in M_{si} \qquad (8)$$

Model MMG_s^1 can be classified as an MRCPSP but with a budget constraint on resource usage costs. The resource constraints are not very tight since the capacities W_r and Q_n are bounds for the whole set of projects.

Model $\boldsymbol{MMG_s^2}$ ($\forall\ s\in S$)

$$\min \quad \kappa_s \qquad (9)$$

$$s.t. \qquad \sum_{i\in V_s}\sum_{j\in M_{si}}\sum_{t=e_i}^{l_i} g_{ij}x_{ijt} = \kappa_s \qquad (10)$$

$$T_{z_s} \le T_s^h \qquad (11)$$

$$(2)-(5), (7) \text{ from Model } MMG_s^1$$

In Model MMG_s^2, the budget κ_s is taken as the objective function (9). Constraint (10) provides the definition of κ_s in terms of the variable resource usage costs and the decision variables. Constraint (11) sets a parametric upper bound, T_s^h, on the makespan of the project. The way T_s^h is specified is explained below. Note that there is a negative relation between the project makespan and the budget consisting of the resource usage costs g_{ij} for the selected activity modes which are by definition positive. Macro-mode generation procedure is initialized by calculating the mode costs as expressed in(8).

A mode j of an activity i is called inefficient, if there exists another mode j^* for activity i with $d_{ij} \geq d_{ij^*}$ and $w_{ijr} \geq w_{ij^*r}$ for each renewable resource $r \in R$ and $q_{ijn} \geq q_{ij^*n}$ for each non-renewable resource $n \in N$ (Kolisch et al., 1995). Inefficient modes are removed from further consideration.

The maximum budget required, κ_s^{max}, is computed by determining the highest mode cost g_i^{max} for each activity $i \in V_s$ and adding them up. The bounds on the duration range $[D_s^{min}, D_s^{max}]$ for T_s^h are computed by solving Model MMG_s^1 once for $\kappa_s = 0$ and once for $\kappa_s = \kappa_s^{max}$, respectively. Duration range for T_s^h signifies the durations for possible macro-modes to be generated. Solving Model MMG_s^2 results in a schedule with a makespan equal to or less than T_s^h and mode selections resulting in the least budget requirements. Starting with D_s^{min}, T_s^h is increased by one at each step until D_s^{max} is reached. At each step, Model MMG_s^2 is solved and if κ_s value is lower than the previous solution, it is concluded that a new macro-mode v is generated based on the optimal solution of MMG_s^1 expressed by $x_{ij\theta}^*$ and added to the macro-mode set $\widetilde{M}_s$ of project s. Note that v is one of several macro-modes which might have been generated for the same T_s^h value. The duration, the renewable resource profile (12) and the non-renewable resource consumption (13) obtained in the solution of the Model MMG_s^2 define the new macro-mode v.

$$\omega_{svrt} = \sum_{i \in V_s} \sum_{j \in M_{si}} \sum_{\theta=\max(e_i,\ t-d_{ij}+1)}^{\min(l_i+d_{ij}-1,\ t)} w_{ijr} x_{ij\theta}^* \quad s \in S, v \in \widetilde{M}_s, r \in R, t \in \{1, \dots, T_s^h\} \qquad (12)$$

$$\eta_{svn} = \sum_{i \in V_s} \sum_{j \in M_{si}} q_{ijn} \sum_{t=e_i}^{l_i} x_{ij\theta}^* \qquad s \in S, v \in \widetilde{M}_s, n \in N \qquad (13)$$

The cash flow associated with a macro-activity s (project s) and a macro-mode $v \in \widetilde{M}_s$ is denoted by C_{sv} and is defined in (14). C_{sv} is obtained by subtracting from the lump sum payment received at the completion of the macro-activity s the expenditures incurred for the corresponding project fixed cost and the resource usage costs all being discounted to the start of macro-activity s using a discount factor α.

$$C_{sv} = C_s^R(1+\alpha)^{-\tilde{d}_{sv}} - C_s^I - \left(\sum_{\theta=0}^{\tilde{d}_{sv}-1} (1+\alpha)^{-\theta} \left(\sum_{r\in R} a_r \omega_{svr\theta} + \sum_{n\in N} b_n \frac{\eta_{svn}}{\tilde{d}_{sv}} \right) \right) \quad s \in S, v \in \widetilde{M}_s \quad (14)$$

3.2 Macro-Activity Scheduling

The macro-activity scheduling problem is defined as Model MP.

Model MP

$$max \quad NPV = \sum_{s\in S} \sum_{v\in \widetilde{M}_s} \sum_{t=\tilde{e}_s}^{\tilde{l}_s} (1+\alpha)^{-t+1} C_{sv} \tilde{x}_{svt} \qquad (15)$$

s.t.

$$\sum_{s\in S} \sum_{v\in \widetilde{M}_s} \sum_{\theta=\max(\tilde{e}_s,\ t-\tilde{d}_{sv}+1)}^{min(\tilde{l}_s+\tilde{d}_{sv}-1,\ t)} \omega_{svr(t-\theta+1)} \tilde{x}_{sv\theta} \le W_r \quad r \in R, t \in \mathcal{T} \ (16)$$

$$\sum_{s\in S} \sum_{v\in \widetilde{M}_s} \eta_{svn} \sum_{t=\tilde{e}_s}^{\tilde{l}_s} \tilde{x}_{svt} \le Q_n \qquad n \in N \qquad (17)$$

$$\sum_{v\in \widetilde{M}_s} \sum_{t=\tilde{e}_s}^{\tilde{l}_s} \tilde{x}_{svt} = 1 \qquad s \in S \qquad (18)$$

$$\tilde{x}_{svt} = \begin{cases} 1, \text{if macro} - \text{activity } s \text{ starts in period } t \\ \quad \text{using macro mode } v \\ 0, \quad o/w \end{cases} \qquad s \in S, v \in \widetilde{M}_s, t \in \mathcal{T} \qquad (19)$$

The cash flows C_{sv} in the objective function are defined earlier (14) and represent the NPV of the return and all the costs involved for macro-activity *s* and macro-mode $v \in \widetilde{M}_s$ discounted to the start time of macro-activity *s*. Hence, the objective function is the total discounted NPV of all the cash flows over all macro-activities (i.e., projects). Constraint set (16) is the capacity constraint for the renewable resources determined based on the schedules evaluated in the macro-mode generation step. Constraint set (17) is the capacity constraint for the non-renewable resources. Constraint set (18) ensures that for each project a macro-mode alternative is selected and is started at some point in the interval $[\tilde{e}_s, \tilde{l}_s\,]$.

The time horizon $\mathcal{T}$ employed in Model MP is obtained through a heuristic procedure developed here for this purpose and called the Relaxed Greedy Heuristic (RGH). In RGH, a simple binary integer programming model with non-renewable resource capacity and macro-mode assignment constraints is solved to obtain the non-renewable resource feasible list of macro-mode selections with the greatest sum of cash returns. Then these macro-modes are listed in non-decreasing order of cash flows and are scheduled using a serial scheduling scheme (see e.g., Kolisch, 1995; Kolisch, 1996) according to this ordered list yet this time taking the renewable resource capacities into consideration. In addition, an initial feasible solution, which is a lower bound for the actual problem, is obtained while determining the time horizon value.

3.3 Post-Processing for Macro-Activity Scheduling

In this section, a post-processing procedure is introduced to redistribute to the projects the renewable resources expressed by W'_{rt} (20) and non-renewable resources expressed by Q'_n (21) that are left over after the macro-activity scheduling where $\tilde{x}^*_{sv\theta}$ represents the best solution obtained for Model MP.

$$W'_{rt} = W_r - \sum_{s\in S}\sum_{v\in\tilde{M}_s}\sum_{\theta=\max(\tilde{e}_s,\ t-\tilde{d}_{sv}+1)}^{min(\tilde{l}_s+\tilde{d}_{sv}-1,\ t)} \omega_{svr(t-\theta+1)}\tilde{x}^*_{sv\theta} \qquad r\in R, t\in\mathcal{T} \quad (20)$$

$$Q'_n = Q_n - \sum_{s\in S}\sum_{v\in\tilde{M}_s}\eta_{svn}\sum_{t=\tilde{e}_s}^{\tilde{l}_s}\tilde{x}^*_{svt} \qquad n\in N \quad (21)$$

In order to benefit from the left-over capacities, for each project s, a new macro-mode v_s^+ is generated by solving Model MMG_s^3 . When trying to improve the NPV of the schedule one can either change the macro-mode selection or advance the start time of projects or do both. Here, the start time for each project is kept the same as before in order to keep the search limited since we seek local improvement resulting in relatively small computational burden. Model MMG_s^3 is an MRCPSPDCF with variable capacities for the renewable resources and with the positive and negative cash flows. The new macro-mode v_s^+ is generated so as to maximize the project NPV_s (22) assuming all of the extra resource capacities along with the currently assigned resource capacities are made available for the project s as expressed in the constraint sets (23) and (24). The objective function is defined by including the project fixed cost, the lump sum payment at the completion of the project and the variable resource usage costs, which are incurred on a periodic basis and are calculated as in (25). The NPV of the newly created alternative v_s^+ is at least as large as that of the macro-mode v_s^*, which was selected by solving Model MP.

Model $\boldsymbol{MMG_s^3}$ ($\forall\, s \in S$)

$$\max \quad NPV_s = \sum_{\theta=\tilde{e}_{z_s}}^{\tilde{l}_{z_s}} (1+\alpha)^{\theta-1} C_s^R - C_s^I - \sum_{i \in e_s} \sum_{j \in M_{si}} \sum_{t=e_i}^{l_i} \left(\sum_{\theta=0}^{d_{ij}-1} (1+\alpha)^{-\theta} \left(\sum_{r \in R} a_r w_{ijr} + \sum_{n \in N} b_n \frac{q_{ijn}}{d_{ij}} \right) \right) x_{sijt} \qquad (22)$$

$$s.t. \quad \sum_{i \in V_s} \sum_{j \in M_{si}} q_{ijn} \sum_{t=e_i}^{l_i} x_{ijt} \le Q'_n + \eta_{sv^*n} \qquad n \in N \qquad (23)$$

$$\sum_{i \in V_s} \sum_{j \in M_{si}} \sum_{\theta=\max(e_i,\ t-d_{ij}+1)}^{\min(l_i+d_{ij}-1,\ t)} w_{ijr} x_{ij\theta} \le W'_{r(T_s^*+t-1)} + w_{sv^*rt} \qquad r \in R, t \in \{1, \dots, \tilde{d}_{sv^*}\} \qquad (24)$$

(2), (5) and (7′) from Model MMG_s^1

where T_s^* is the start time of project s obtained in the solution of the Model MP, $\tilde{d}_{sv^*}$ is the duration of the macro-mode v_s^* and (7′) differs from (7) in that x_{ijt} is defined in (24) over $t \in \{1, \dots, \tilde{d}_{sv^*}\}$ rather than over $t \in \mathcal{T}_s$.

Once the new macro-mode v_s^+ is formed for each actual project s, the resulting changes in NPV and resource capacities due to macro-mode shifts are calculated. C''_s, the benefit gained on NPV due to the macro-mode shift in project s is calculated as in (25). Changes in renewable resource capacities, W''_{srt} and in non-renewable resource capacities, Q''_{sn} are defined in (26) and (27), respectively.

$$C''_s = (C_{sv^+} - C_{sv^*})(1+\alpha)^{(T_s^*-1)} \qquad s \in S^a \qquad (25)$$

$$Q''_{sn} = Q_{sv^+n} - Q_{sv^*n} \qquad s \in S^a, n \in N \qquad (26)$$

$$W''_{srt} = W_{sv^+rt} - W_{sv^*rt} \qquad s \in S^a, r \in R, t \in \{1, \dots, d_{sv^*}\} \qquad (27)$$

It may not be possible to shift the macro-modes for all projects at the same time because of the conflicting needs for the common left-over capacities. On the other hand, making a macro-mode shift for project s may assign some left-over capacities to project s but it may also release some of the resources that are no longer required once the shift is realized. This means that these possible macro-mode shifts are linked with each other. Hence, decisions on macro-mode shifts should be made considering the projects simultaneously by solving Model *MMS*.

In Model MMS, the aim is to maximize the total NPV gain by selecting the projects to apply the macro-mode shift (28). Model MMS is a knapsack type formulation with varying renewable resource capacities over time.

Model $\boldsymbol{MMS}$

$$max \quad \sum_{s \in S} C_s'' y_s \tag{28}$$

$$s.t. \quad \sum_{s \in S} Q_{sn}'' y_s \leq Q_n' \qquad n \in N \tag{29}$$

$$\sum_{s \in S} \sum_{\theta = T_s^*}^{T_s^* + \tilde{d}_{sv^*} - 1} W_{sr\theta}'' y_s \leq W_{rt}' \qquad r \in R, t \in \mathcal{T} \tag{30}$$

$$y_s = \begin{cases} 1, & \text{if project } s \text{ is selected for macro} - \text{mode shift} \\ 0, & o/w \end{cases} \qquad s \in S \tag{31}$$

Constraint sets (29) and (30) ensure that the total resource availability bounds are not violated. Variable y_s as defined in (31) indicates whether a macro-mode shift is applied to a project or not.

After applying the macro-mode shifts in the selected projects, the scheduling of each individual project follows.

3.4 Scheduling Each Individual Project

After setting the resource capacities and the start times of the projects, each project is individually scheduled for minimizing the project makespan. The problem is formulated as an MRCPSP with varying renewable resource capacities over time. Model S_s is given below:

Model $\boldsymbol{S_s}$ ($\forall\ s \in S$)

$$min \quad T_{z_s} \tag{32}$$

$$s.t. \quad \sum_{i \in V_s} \sum_{j \in M_{si}} q_{ijn} \sum_{t = e_i}^{l_i} x_{ijt} \leq \tilde{Q}_n^s \qquad n \in N \tag{33}$$

$$\sum_{i \in V_s} \sum_{j \in M_{si}} \sum_{\theta = \max(e_i,\ t - d_{ij} + 1)}^{\min(l_i + d_{ij} - 1,\ t)} w_{ijr} x_{ij\theta} \leq \widetilde{W}_{rt}^s \qquad r \in R, t \in \mathcal{T}_s \tag{34}$$

$(2), (5)$ and (7) from Model MMG_s^1

The time dependency of resource capacity levels was expected to cause a significant increase in computation time but it was not experienced in this particular problem because resource capacities are quite tight. Recall that they are determined by the selection of macro-modes, which were generated through solving a very similar model repeatedly.

4.A GENETIC ALGORITHM APPROACH FOR THE MACRO-ACTIVITY SCHEDULING PROBLEM

4.1 Representation

Since the problem is a version of multi-component combinatorial optimization problem with sequencing and selection components, a common chromosome structure including two serial lists is used to represent a solution for the problem as in Şerifoğlu (1997). First list is a permutation of non-dummy activities representing the priority order of activities for scheduling and the second one is a list of mode selections for activities. Another list representation based GA for RCPSP is given by Hartmann (1998), which he later extended to the multi-mode case (Hartmann, 2001).

4.2 Evaluation of the Chromosomes

Fitness of a chromosome is determined by calculating NPV values considering the positive cash flows incurred at the start of each activity. Start times are determined by obtaining the specific schedule represented by the lists stored in the chromosome. Since all cash flows are positive, starting the activities as early as possible is more desirable to achieve higher NPVs. A serial scheduling scheme is used to schedule the activities based on the priority sequence in the first list and the mode selections in the second list of the chromosome.

4.3 Operators

4.3.1 Crossover Operator

Considering that there is no precedence feasibility issues among the activities each corresponding to a project, 2-point crossover method is employed. In 2-point crossover procedure, two random genes from the first parent are picked and then genes before the first randomly selected gene and after the second randomly selected gene are directly passed on to the child. Then the genes associated with the activities missing in child's priority order list are acquired from the second parent following the order in its priority order list together with the associated modes.

4.3.2 Mutation Operators

There are two mutation operators used to randomly modify the newborn and reproduced chromosomes: *Swap mutation:* It is executed on the priority order list to obtain different sequences, which may or may not lead to a different schedule, by swapping the locations of two activities randomly selected. The activities are swapped preserving their already assigned mode.

Bit mutation: An activity is selected randomly on the priority order list and its mode is replaced with another randomly chosen mode value. Bit mutation is not allowed to lead to a non-renewable resource infeasible solution.

4.4 Population Management

Initial population is formed as follows: First, a mode selection list is generated by selecting a random mode for each activity and if the mode selections are not feasible considering the non-renewable resource capacities, it is formed again from scratch. Note that Kolisch and Drexl (1997) have proven that the feasibility problem for $|N| \geq 2$ is NP-complete. The non-renewable resource feasible mode selection list is then combined with a random sequence of activities created. In addition, if there are any existing solutions at hand, they can be also included in the initial solution.

At each iteration, a new population is created as follows: A number of new members, which corresponds to a ratio r_{new} of the population size n_{pop} are created by using the 2-point crossover with members randomly selected from the current population and are added to the new population along with two elite individuals. The additional number of individuals needed to increase the population size to n_{pop} is then reproduced from the current population with the elite individuals deleted using the roulette wheel selection method. Finally, each individual except the elite ones are considered first for a swap mutation with probability p_{swap} and then for a bit mutation with probability p_{bit}. New population generation scheme is given in Figure 2.

<u>Place Figure 2 about here</u>

4.5 Restart

In order to avoid the possibility of early convergence and to refresh the population, after each n_{res} number of generations restart is applied, if the ratio of identical individuals in the population exceeds 30%. If this is not the case, then the algorithm is run for another n_{res} number of generations. In each restart, all the members in the population except the elites are replaced by randomly generated new members.

4.6 Termination

The procedure is carried out for a predetermined number of generations and once this maximum generation limit n_{gen} is reached, the procedure is terminated.

4.7 Fine Tuning the Design Parameters

A series of experiments are performed to fine tune the design parameters for the proposed GA algorithm. Various values for the design parameters shown in Table 1 are tested to arrive at a combination of design parameter values, which will result in a relatively better performance. Number of elitesis set to 2 and for each remaining design parameter, representative values are tried to be selected for testing.

<u>Place Table 1 about here</u>

A test data set is formed consisting of 17 instances for which optimal values are determined using an MIP solver. These instances are sampled from the main data set, which is described in section 5 in detail, and tested for various design parameter value combinations. For each test data set and parameter combination, five replications are executed and the average best solutions and the average computation times are calculated. Considering that the primary intention is to obtain solutions that are as good as possible and the computational time required for GA application is relatively small, the combination performances are evaluated mainly based on the closeness of the best solution obtained to the optimal. The computational time is used as a secondary performance measure.

Parameter value combinations are tested in two phases. In the first phase, 324 combinations regarding parameters n_{pop}, n_{gen}, r_{new}, p_{swap} and p_{bit} are analyzed and set. Then using the parameter values fixed previously, 3 combinations regarding the number of generations per restart check are tested in the second level.

Comparing the performances of the parameter value combinations obtained excluding restart possibility, it has been observed that $n_{pop} = 100$ and $n_{gen} =$

500 perform better just as expected since larger values allow for more computation, which cannot have a negative effect on the objective value function. However, it was realized that there was not any significantly dominant set of values for the parameters r_{new}, p_{swap} and p_{bit} and various combinations worked quite well with small differences between each other. Therefore, a small best performing segment of parameter combinations for each data instance is taken and the frequency of combinations are considered resulting in a combination with $r_{new} = 0.6$, $p_{swap} = 0.5$, $p_{bit} = 0.2$ performed better. Fixing the parameter values determined so far, $\mathrm{n}_{\mathrm{res}}$ is tested. $n_{res} = 100$ performed better for the majority of data instances. Hence, it is decided to use the combination $n_{pop} = 100$, $n_{gen} = 500$, $r_{new} = 0.6$, $p_{swap} = 0.5$, $p_{bit} = 0.2$ and $n_{res} = 100$ for all the computations following.

5. COMPUTATIONAL STUDY

In order to analyze the performance of the proposed 2-stage decomposition method for the multi-project scheduling problem, a series of computational experiments are carried out. These experiments are meant to observe and examine the effects of various factors, which shape the problem environment, on the results obtained and the required computational effort.

Since currently there are no benchmark problem sets with the required structure available, new problem sets are generated using the single project instances taken from PSBLIB (Kolisch and Sprecher, 1996). Various instances with different number of jobs from PSPLIB are combined into multi-project problems by assigning cash flow values, general resource capacities, and resource utilization costs.

5.1 Resource Conditions

Resource Factor (RF_τ), which measures the usage/consumption, and Resource Strength (RS_τ) which measures the availability, are defined to represent the resource based conditions of resource categories $\tau \in \{R, N\}$ and shown to exercise (Kolisch et al.,1995) a strong effect on the behavior of RCPSP solution procedures, are adapted here for multi-project scheduling environment. RF_R is given by (35) and (36); and RF_N is given by (37) and (38).

$$\mathrm{y}_{\mathrm{ijr}} = \begin{cases} 1, & \mathrm{w}_{\mathrm{ijr}} > 0 \\ 0, & \mathrm{o/w} \end{cases} \qquad i \in V_s, j \in M_{si}, r \in R \qquad (35)$$

$$RF_R = \frac{1}{|R|} \frac{1}{|S|-2} \sum_{s=2}^{|S|-1} \frac{1}{|V_s|-2} \sum_{i=2}^{|V_s|-1} \frac{1}{|M_{si}|} \sum_{j \in M_{si}} \sum_{r \in R} y_{ijr} \tag{36}$$

$$z_{ijn} = \begin{cases} 1, & q_{ijn} > 0 \\ 0, & o/w \end{cases} \qquad i \in V_s, j \in M_{si}, n \in N \tag{37}$$

$$RF_N = \frac{1}{|N|} \frac{1}{|S|-2} \sum_{s=2}^{|S|-1} \frac{1}{|V_s|-2} \sum_{i=2}^{|V_s|-1} \frac{1}{|M_{si}|} \sum_{j \in Msi} \sum_{n \in N} z_{ijn} \tag{38}$$

The resource availability for each renewable resource $r \in R$ is given as:

$$K_r = K_r^{min} + round\{RS_R(K_r^{max} - K_r^{min})\} \qquad r \in R \tag{39}$$

where $K_r^{min} = \max_i\{\min_j\{w_{ijr}\}\}$ and the maximum level K_r^{max} is determined by the peak per period usage of the renewable resource r required in the early finish schedule obtained through forward recursion and selecting the activity modes with the greatest requirements for the renewable resource r.

The resource availability for each non-renewable resource $n \in N$ is given as:

$$K_n = K_n^{min} + round\{RS_N(K_n^{max} - K_n^{min})\} \qquad n \in N \tag{40}$$

where $K_n^{max} = \sum_i max_j\{q_{ijn}\}$ and $K_n^{min} = \sum_i min_j\{q_{ijn}\}$.

5.2 Financial Parameters

Discount rate (α) is selected to be 0.05 per period for all instances and constant throughout the time horizon. a_r and b_n are both assumed to be 3. Due to the nature of the problem and the solution procedure, cash flows for macro-modes cannot be initially known but they can only be calculated considering the lump sum payments at the completion time of projects, c_s^R; fixed cost to be invested in order to start a project, c_s^I; and resource based variable costs, a_r and b_n as the macro-modes are created one by one. This condition arises from a necessity for seeking a sensible approach to set c_s^R and c_s^I for each project $s \in S$. They are determined by using (42) and (43), where CR_s, a base cost related with resource usages as expressed in (41), is multiplied by a factor drawn from the uniform distribution $U \sim (0, 1)$, and the factors f^R for lump sum payments and f^I for investment costs. $f^R = 18$ and $f^I = 0.2$ are used here for all problem instances so that positive cash flows are ensured at the macro-mode generation process.

$$CR_s = \sum_{i \in V_s} \frac{1}{|M_{si}|} \sum_{j \in M_{si}} \left(\sum_{r \in R} d_{ij} a_r w_{ijr} + \sum_{n \in N} b_n q_{ijn} \right) \tag{41}$$

$$c_s^R = \quad CR_s f^R\big(1 + (U\sim(0,1))\big) \tag{42}$$

$$c_s^I = \quad CR_s f^I\big(1 + (U\sim(0,1))\big) \tag{43}$$

5.3 Problem Sets

Three problem sets denoted by A,B,C are created to represent a variety of different environmental factors.

Problem set A is formed to analyze the effect of resource based factors by fixing other factors. It includes multi-project instances all having the same number of projects consisting of the same number of activities but different resource requirements and resource availability levels, categorized by RS and RF values for renewable and non-renewable resources. Each instance includes 14 projects consisting of 10 activities each as shown in the first two columns of Table 2. Three levels are selected for each factor including RF_R, RF_N, RS_R and RS_N as given in the last four columns of Table 2. To avoid any infeasibilities due to insufficient non-renewable resources, a minimum value for RS_N, RS_N^{min}, is determined by simple testing and a medium level is also calculated by $RS_N^{mid} = RS_N^{min} + \big(1 - RS_N^{min}\big)/2$. Combinations of these four variable factors with three levels each results in problem set A with 81 instances in total.

<u>Place Table 2 about here</u>

Problem set B focuses on the effects of different number of projects and activities. In these multi-project instances, three levels are set for the number of projects and seven levels are set for the number of activities as provided in the first two columns of Table 3 RF values for renewable and non-renewable resources are fixed to be 0.5 as shown in the third and fourth columns of Table 3. Two levels are determined for RS_R and RS_N values as shown in the last two columns of Table 4. Levels for RS_Nvalues are set using $RS_N^{mid1} = RS_N^{min} + \big(1 - RS_N^{min}\big)/3$ and $RS_N^{mid2} = RS_N^{min} + 2 * \big(1 - RS_N^{min}\big)/3$. Combinations of these four variable factors with different levels results in problem set B with 84 instances in total.

<u>Place Table 3 about here</u>

In problem set C, a multi-project environment that is heterogeneous in terms of project sizes is emphasized by grouping projects consisting of different number of activities (Table 4). Basically, three multi-project groups are formed and different levels of resource strengths are assigned. In the first group; equal number of projects of relatively small, medium and large sizes are brought together. In group two, a few larger projects are handled together with a collection of smaller sized projects. In the third group, a few smaller projects are thrown into a bunch of relatively larger sized projects. The levels for RS_N values are set as for problem set A. Combinations of these three multi-project groups with three resource strength levels result in 27 instances.

Place Table 4 about here

5.4 Software and Hardware Information

All codes are written in GNU C# and the MIP solver is CPLEX 12.1. All experiments were performed on a HP Compaq dx 7400 Microtower with a 2.33 GHz Intel Core 2 Quad CPU Q8200 processor and 3.46 GB of RAM.

5.5 2-Stage Decomposition Method Performance Analysis

For assessing the performance of the 2-stage decomposition procedure three configurations of the methods employed are designed. Besides the GA approach presented in section 4, which is employed for solving the macro-activity scheduling model (Model MP), all of the mathematical programming models presented as part of the proposed 2-stage decomposition procedure are solved using an MIP solver. Model MP is also solved using the same MIP solver besides GA. In the first configuration, Model MP is solved by the GA approach whereas in the second configuration it is solved by the MIP solver. A third configuration is created by using the GA approach for generating an initial solution for the MIP solver in the second configuration.

5.5.1 Results

In this section, the general results obtained for the problem sets A, B and C by running the algorithm with all three configurations are shared. A two hours time limit is set for the MIP solver to run. For some of the instances in problem sets B and C, the computation time limit of two hours for the MIP solver is reached before an optimal solution is obtained. Such instances are not reported in the results presented. CPU_{Total} reported in Tables 5, 7 and 8 corresponds to the average CPU time required to solve both stages of the solution procedure. In Table 5, besides CPU_{Total}, average objective function value for stage 1, NPV $_{Ave}$, is reported for all three configurations.

Place Table 5 about here

Examining Table 5, it can be concluded that employing GA as a stand-alone routine for macro-activity scheduling Model MP performs quite well considering the good objective function values obtained with small computational effort spent. Table 5 also shows that Configuration 3 performs slightly better than Configuration 2 for the problem sets B and C in terms of the computational effort required.

Place Table 6 about here

Post-processing procedure improves the objective function value considerably with quite little computational effort as it can be seen in Table 6.

5.5.2 General Observations

In this section, some general observations made on the results obtained with Configuration 3 (employing both GA and MIP solver) are reported.

Place Table 7 about here

Table 7 shows that RS has a significant effect on the computational effort required for macro-project scheduling step. The computational effort required increases up to a maximum level as RS_R, which indicates the level of renewable resource availabilities, increases up to a certain medium level and afterwards computational effort required seems to decrease dramatically as the renewable resource availabilities climb to a higher level.

Place Table 8 about here

Table 8 presents the average CPU_{Total} required to solve the instances from problem set B and having different number of projects. Column 2 includes the average values including only the instances for which macro-project scheduling problem is solved to optimality within the time limit. The fact that the values in column 2 increase as the number of projects increases, coincides with the expectation that the number of projects in the problem environment has a significant impact on the problem difficulty.

6. SUMMARY AND FUTURE WORK

An operationally effective and viable 2-stage decomposition approach reflecting the dual level project management structure and based on the concepts of macro-activity and macro-mode introduced by Speranza and Vercellis (1993) is presented. For that purpose several different formulations and solution procedures have been introduced.

The macro-mode generation procedure in the first stage of the decomposition is applied with the introduction of a new search systematic for the macro-modes. The budget introduced is based on the different types of costs involved. The use of such a budget enables the generation of representative modes via M_s^1 and M_s^2.

In order to reduce the number of variables in the formulation for MRCPSPDCF with positive cash flows three different time horizon setting methods are developed and tested.

A GA approach is adopted for solving MRCPSPDCF with time dependent renewable resource requirements. The GA is employed as a standalone solution procedure as well as for generating initial solutions for the exact solution procedure.

An efficient post-processing procedure is introduced to distribute the resources that are left over after stage one to the projects to search for any improvements.

In order to analyze the performance and behavior of the proposed 2-stage decomposition method, new data sets are formed using the single project instances taken from PSBLIB compiled by Kolisch and Sprecher (1996) and a series of computational experiments are carried out.

Although this study deals with MRCMPSP, some specific versions of MRCPSP are directly dealt with as well due to the nature of the decomposition based approach applied such as, e.g., an MRCPSP with time-dependent renewable resource capacities.

There are several extension possibilities such as the following, which can be studied in the future.

- Precedence relations between projects can also be included considering that in practice some projects need to precede others because of technological reasons.
- Project termination deadlines can be specified and penalty costs for violating these deadlines can be included in the cost structure or a just in time environment can be considered.

Considering the relevance of the problem treated here to manufacturing firms as well as for project based firms and these future research possibilities, it can be concluded that resource constrained multi-project scheduling with hierarchical decomposition based approaches is a rich topic still requiring further investigation.

ACKNOWLEDGEMENT:

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

REFERENCES

Bock, D. B. and Patterson, J. H. (1990), “A comparison of due date setting, resource assignment and job preemption heuristics for multi-project scheduling problem”, *Decision Sciences*, 21, 387-402.

Confessore, G., Giordani, S. and Rismondo, S. (2007), “A market-based multi-agent system model for decentralized multi-project scheduling”, *Annals of Operations Research* 150, 115-135.

Gonçalves, J. F., Mendes, J. J. M. and Resende, M. G. C. (2008), “A genetic algorithm for the resource constrained multi-project problem”, *European Journal of Operational Research* 189, 1171-1190.

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. (1998), “A competitive genetic algorithm for resource-constrained project scheduling” *Naval Research Logistics* 45,733-750.

Hartmann, S. (2001), “Project scheduling with multiple modes: A genetic algorithm” *Annals of Operations Research* 102,733-750.

Homberger, J. (2007), “A multi-agent system for the decentralized resource-constrained multi-project scheduling problem”, *International Transactions in Operational Research*, 14, 565-589.

Homberger, J. (2012), “A (μ, λ)–coordination mechanism for agent–based multi–project scheduling”, OR Spectrum, 34,

Kim, S. O. and Schniederjans, M. J. (1989), “Heuristic framework for the resource constrained multi-project scheduling problem” *Computers and Operations Research* 16, 541-556.

Kolisch, R. (1995), *Project Scheduling under Resource Constraints-Efficient Heuristics for Several Problem Classes*, Physica-Verlag, Heidelberg.

Kolisch, R., Sprecher, A. and Drexl, A. (1995), “Characterization of generation of a general class of resource-constrained project scheduling problems”, *Management Science* 41, 1693-1703.

Kolisch, R. (1996), Serial and parallel resource-constrained project scheduling methods revisited: Theory and computation, *European Journal of Operational Research* 90, 320-333.

Kolisch, R. and Sprecher, A. (1996), “PSPLIB – A project scheduling problem library”, *European Journal of Operational Research* 96, 205-216.

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

Kumanan, S., Jegan Jose G. and Raja K. (2006), “Multi-project scheduling using an heuristic and a genetic algorithm”, *International Journal of Advanced Manufacturing Technology* 31, 360-366.

Kurtulus, I. and Davis, E. W. (1982), “Multi-project scheduling: categorization of heuristic rules performance”, *Management Science* 28, 161-172.

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

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

Lee, Y., Kumara, S. R. T. and Chatterjee, K. (2003), “Multiagent based dynamic resource scheduling for distributed multiple projects using a market mechanism”, *Journal of Intelligent Manufacturing* 14, 471-484.

Liberatore, J. M., Pollack-Johnson, B. and Smith, A. S. (2001), "Project Management in construction: software use and research directions", *Journal of Construction Engineering and Management* 127, 101-107.

Liberatore, J. M. and Titus, G. J. (1983), "The practice of management in R&D project management", *Management Science* 29, 962-974.

Lova, A. and Tormos, P. (2001), "Analysis of scheduling schemes and heuristic rules performance in resource-constrained multi-project scheduling", *Annals of Operations Research* 102, 263-286.

Mittal, M. L. and Kanda, A. (2009), "Two-phase heuristics for scheduling of multiple projects", *International Journal of Operational Research* 4, 159-177

Payne, J. H. (1995), "Management of multiple simultaneous projects: a state-of-the-art review", *International Journal of Project Management* 13, 163-168.

Pritsker, L. J., Walters, A. A. B. and Wolfe, P. M. (1969), "Multi-project scheduling with limited resources: a zero one programming approach", *Management* Science 16, 93-108.

Speranza, M. G. and Vercellis, C. (1993), "Hierarchical models for multi-project planning and scheduling", *European Journal of Operational Research* 64, 312-325.

Şerifoğlu, F. (1997), *A new uniform order-based crossover operator for genetic algorithm applications to multi-component combinatorial optimization problems,* Unpublished PhD Dissertation, Boğaziçi University, Istanbul.

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

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

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

Zapata, J. C., Hodge, B. M. and Reklaitis, G. V. (2008), "The multimode resource constrained multi-project scheduling problem: Alternative formulations", *AIChE Journal* 54, 2101-2119.

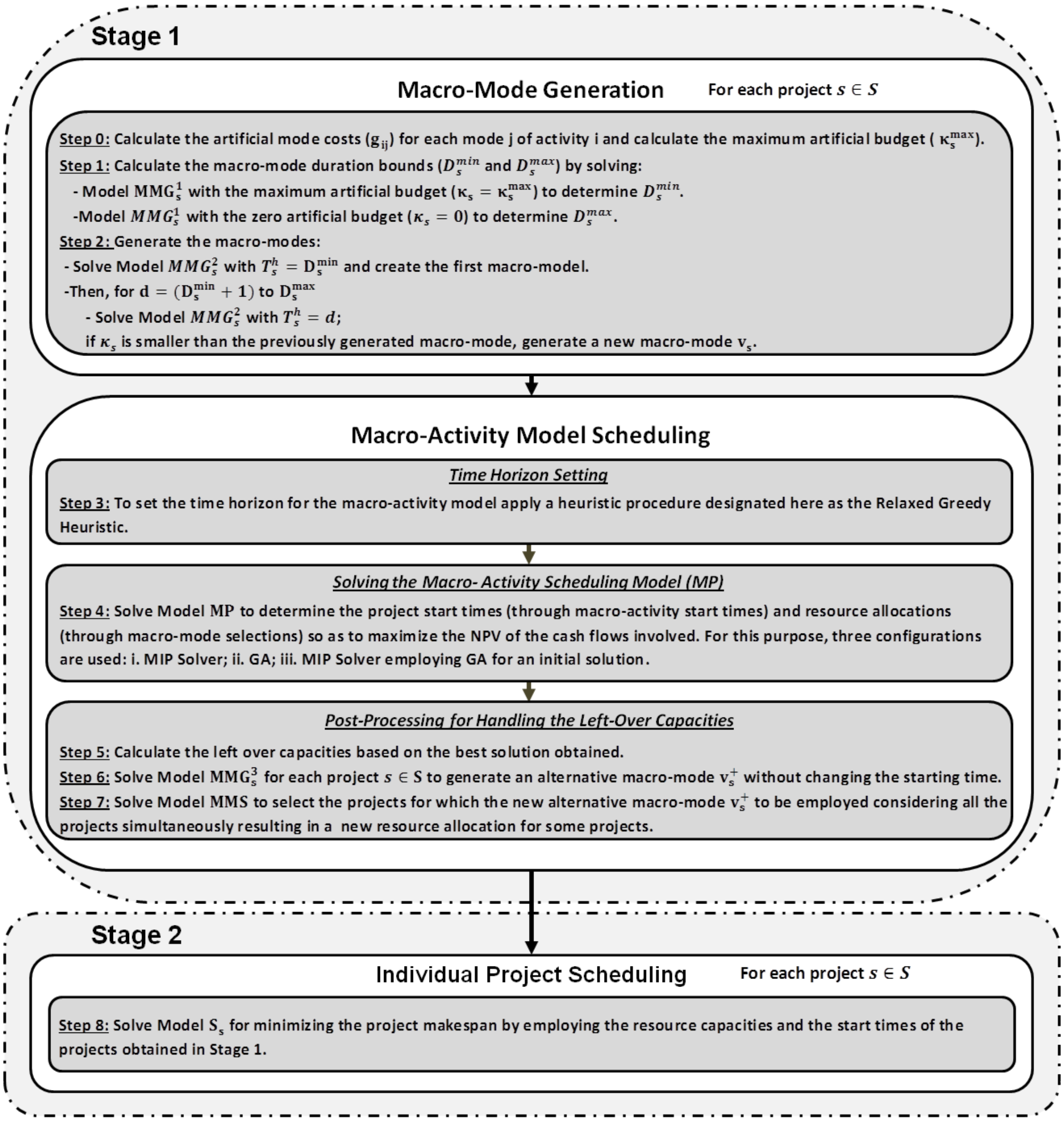

Figure 1. 2-Stage decomposition procedure flow

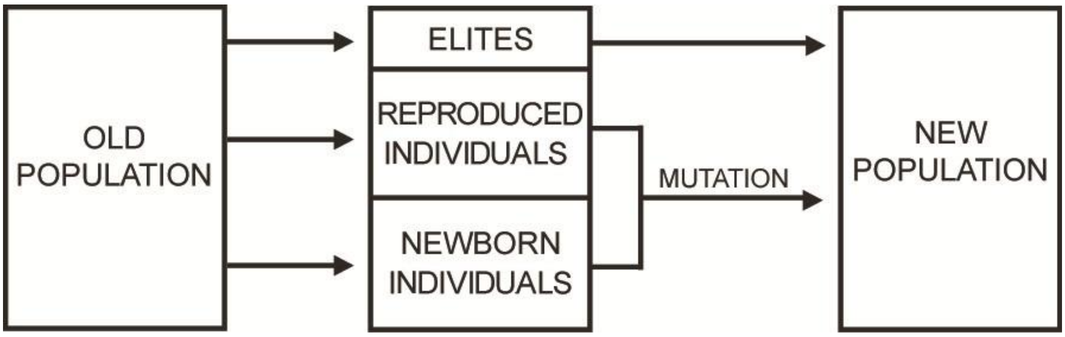

Figure 2. New population generation scheme

Table 1. Design parameters and their range of values for fine-tuning

Design Parameters	Identifier	Values
Number of elites	n_{elite}	{2}
Population size	n_{pop}	{50, 75, 100}
Number of generations	n_{gen}	{200, 300, 400, 500}
Ratio of newborn	r_{new}	{0.4, 0.6, 0.8}
Probability of swap mutation	p_{swap}	{0.2, 0.5, 0.8}
Probability of bit mutation	p_{bit}	{0.2, 0.5, 0.8}
Number of generations per injection check	n_{res}	{0, 50, 100}

Table 2. Problem set A

noProj	noAct	RF_R	RF_N	RS_R	RS_N
14	10	{0.5, 0.75, 1}	{0.5, 0.75, 1}	{0.3, 0.6, 0.9}	$\{RS_N^{min}, RS_N^{mid}, 1\}$

Table 3. Problem set B

noProj	noAct	RF_R	RF_N	RS_R	RS_N
{10, 15, 20}	{10, 12, 14, 16, 18, 20, 30}	0.5	0.5	{0.4, 0.7}	$\{RS_N^{mid1}, RS_N^{mid2}\}$

Table 4. Problem set C

noProj & noAct	RF_R	RF_N	RS_R	RS_N
{(5 * J10, 5 * J20, 5 * J30); (8 * J10, 8 * J12, 2 * J30); (3 * J10, 7 * J18, 7 * J20)}	0.5	0.5	{0.3, 0.6, 0.9}	$\{\{RS_N^{min}, RS_N^{mid}, 1\}$

Table 5. General results for problem sets A, B and C

Config. Employed in MP	NPV_{Ave}					
	Problem Set A		Problem Set B		Problem Set C	
	Average	Std. Dev.	Average	Std. Dev.	Average	Std. Dev.
1	101839.35	44425.41	98733.52	46676.62	131821.20	20483.18
2	101912.70	44312.06	99175.99	46923.91	134200.40	20386.24
3	101906.88	44310.16	99171.57	46905.64	134200.40	20386.24
Config. Employed in MP	CPU_{Total} (sec)					
	Problem Set A		Problem Set B		Problem Set C	
	Average	Std. Dev.	Average	Std. Dev.	Average	Std. Dev.
1	20.69	18.42	29.57	18.81	29.92	4.72
2	211.46	419.24	904.66	1939.85	801.12	1400.50
3	231.84	628.98	797.25	1533.19	747.16	1400.58

Table 6. Performance of post-processing

Config. Employed in MP	Average Post-Processing NPV Improvement (%)		
	Problem Set A	Problem Set B	Problem Set C
1	4.23	0.90	1.36
2	4.20	0.66	1.16
3	4.19	0.60	1.17
Config. Employed in MP	Average CPU (sec)		
	Problem Set A	Problem Set B	Problem Set C
1	0.60	0.43	0.92
2	0.52	0.40	0.65
3	0.51	0.41	0.65

Table 7. Effects of RS factor on computational effort required – Problem set A with configuration 3

RS_N	RS_R	Average CPU_{Total} (sec)
0.3	RS_R^{min}	237.24
0.3	RS_R^{mid}	181.44
0.3	1	187.57
0.6	RS_R^{min}	488.12
0.6	RS_R^{mid}	413.11
0.6	1	406.54
0.9	RS_R^{min}	49.09
0.9	RS_R^{mid}	61.72
0.9	1	61.73

Table 8. Effect of number of projects – Problem set B with configuration 3

noProj	Average CPU_{Total} (sec)	Number of instances solved to optimality
10	104.72	28 out of 28
15	1124.59	26 out of 28
20	1608.88	16 out of 28