

Abstract number: 015-0408

Abstract title: A Decision support system for resource constrained multi project scheduling problem using genetic algorithm

M. Suresh, Karuna Jain and Pankaj Dutta,

Shailesh J. Mehta School of Management

Indian Institute of Technology, Bombay

Powai, Mumbai-400076, India

(sureshmarappan@iitb.ac.in, kjain@iitb.ac.in, pdutta@iitb.ac.in)

Phone number: + 91-22-2576 7785/7781; Fax (91)022 25722872/257234

POMS 21st Annual Conference

Vancouver, Canada

May 7 to May 10, 2010

A Decision support system for resource constrained multi project scheduling problem using genetic algorithm

Abstract

Companies usually manage various projects simultaneously, sharing a pool of renewable resources. Often, the availabilities of common resources are limited and not sufficient to execute all the concurrent activities. In such circumstances, decision making about their schedule is necessary. A comprehensive review of the literature on resource constrained multi project scheduling problem (RCMPSP) reveals that an important issue of total cost minimization with cyclic resources vacation in parallel project environment has not received due attention. The purpose of this paper is to develop a decision support system (DSS) for RCMPSP using genetic algorithm in the context of a cyclic off days for labours, cyclic preventive maintenance of machines, overtime allowed for critical activities, scheduling the casual workers with the objective to minimization total cost of all projects and mean project delays. As an application an extensive computational experiment is reported.

Keywords: Multi-projects, genetic algorithm, decision support system, total project cost.

1. Introduction

Project planning and scheduling is a complex and challenging assignment in order to meet the project's objectives. Managing the multiple projects in simultaneously manner is quite common in modern day industry. Multiple projects are processed parallel and compete for scarce resources. Resource Constrained Multi Project Scheduling Problem (RCMPSP) concerns with

assigning common resources to activities over time. In addition, processing every activity requires a predefined amount of resources, which are available in limited quantities in every time period. The resources are not available at all times during the project planning horizon, and the periods of resource unavailability are known in advance and we refer to such resource unavailability as resource vacations. Workers can become unavailable due to planned cyclic vacations and for machines, predictable absences often take the form of scheduled preventive maintenance. In this paper, we capture the characteristics of resource variability with respect to time.

2. Literature review

Fendley (1968) is the first author who explicitly discussed the modeling of multi project scheduling system. Kurtulus and Davis (1982) have used a single-project approach for the multi-project scheduling problem, whereas Kurtulus and Narula (1985) have extended the multi-project approach. Payne (1995) has estimated that up to 90% (measured by their value), of all projects are carried out globally in the multi-project context. Lova and Tormos (2001) have conducted survey of 202 Spanish companies and assured that the 84% of them executed multiple projects in parallel. For RCMPSP, research has been mainly focused on the development of heuristics and the effectiveness of a heuristics mainly depends on the problem context. In this section we present the literature on multi objective RCMPSP with resource vacation and employee's overtime.

Dumond and Mabert (1988) have delineated the multi-project scheduling in context of the static and the dynamic project environment. In static environment all projects are summarized in to a super big project and scheduled once. Project scheduling problems in static environments are studied by Fendley (1968), Pritsker et al. (1969), Kurtulus and Davis (1982), Kurtulus and

Narula (1985), Lawrence and Morton (1993), Lova and Tormos (2001), Vercellis (1994), Patterson (1973), Mohanty and Siddiq (1989), Wiley et al. (1998), Lova et al. (2000), Kruger and Scholl (2008). Project scheduling problems in dynamic environments are studied by Dumond and Mabert (1988), Bock and Patterson (1990), Yang and Sum (1993, 1997), Ash and Smith-Daniels (1999), Anavi-Isakow and Golany (2003). In this paper we consider static multi-projects scheduling problems.

The majority of project scheduling models are single objective (makespan minimization). Normally the real project scheduling problems are multi objective. The literature on multi objective project scheduling is surprisingly scant. Some of the references are Slowinski (1989), Nagar et.al (1995), Hansen (1997), Hapke et.al (1998), Viana and Sousa (2000), Fawzan and Haouari (2004), Abbasi et.al (2006), Azaron and Moghaddam (2007), Belfares et.al (2007), Senouci and Derham (2008), Elloumi and Fortemps (2010). In this model we have considered two objective functions namely total project cost (to be minimized), the mean of the project completion time (to be minimized). Incorporating the resource vacations generally results in initial availability of resources capacities varies over time [Drexl and Grunewald (1993), Hartmann (1999), Bottcher et al., (1999)]. In this paper, we have considered predetermined specific resource vacation. The knowledge and familiarity with the project possessed by the existing resources are a valuable asset to increasing capacity is that of working overtime (Payne, 1995). However, the working of excessive amounts of overtime causes a drop in efficiency of employees, so better method is to limit the taking on of additional work (Payne, 1993). In this paper, we have considered overtime only for critical activities and allowed skilled workers on their predetermined vacation period.

Davis (1985) has used Genetic Algorithms (GA) in the job shop scheduling problem. Since then, a vast literature on the application of GAs to RCPSP has emerged [Hartmann (1998), Kim et al.(2003), Kumanan et al. (2006), Goncalves et al. (2008), Valls et al. (2008), Long and Ohsato (2009)]. GA has become popular solution approach in solving time-cost trade-off problems. Feng et al. (1997) proposed a model using the genetic algorithm and the Pareto front approach to solve time-cost trade-off problems in construction projects. Chen and Shahandashti (2009) have proposed a hybrid GA and simulated annealing based for generic multi-project scheduling problems with multiple resource constraints model, in which the generalized precedence relationship between the activities and makespan minimization are taken into account. Senouci and Derham (2008) have proposed a GA for scheduling of construction projects to minimize the project time, cost and maximize resource utilization. Although, developments of decision support system (DSS) for RCPSP with the objectives to minimize total cost and makespan in resource vacation environment has not been addressed in the literature.

In this paper, we have proposed a DSS to solve RCPSP using GA in the context of cyclic-off-days for labours, cyclic preventive maintenance of machines, overtime for critical activities and non-renewable resource utilization with the objective to minimize both the cost and makespan. The remainder of the paper is organized as follows. In Section 2 the general formulation of the problem and the development of proposed GA. Section 3 we present our computational experience with the GA. Finally, In Section 4 we provide the conclusions and future research directions.

2. RCMPSP model development using genetic algorithm

There are a number of projects to be scheduled. Every project consists of a given set of activities. There is cost associated with the use of every resource and is directly proportional to the amount of resource used. There may be cost associated with keeping resource idle. In general, one would like to keep promised delivery dates, avoid penalties for late completion and to keep good-will. It is advantageous to minimize the overall length of the schedule so that resources can be used for other activities and also reducing the project cost. In RCMPSP, the input to a decision support system (DSS) includes information about activities, resources and constraints. In order to design the framework of the model, the following assumptions are made:

- The RCMPSP consists of multiple projects which are a number of activities with known processing time and multiple resources.
- Activities are subject to finish-start precedence relations with zero time lags, meaning an activity can be started if and only if all of its predecessors have been completed.
- All activity's duration, project due date, cost of the resources, project penalty cost, etc. are predefined.
- In the resource pool various type of renewable resources are available with predetermined vacation period.
- Activities cannot be interrupted; there is only one execution mode for each activity.
- Total amount of available nonrenewable resources are in hand at the starting of the project.
- Overtimes are allowed to critical activities only.
- Only skilled workers are allowed to perform the overtime on their predetermined vacation period.

- Outsourcing is considered only for casual workers.
- Penalty cost is incurred for late completion of the project.

The notations used to develop the model for the problem are given in Table 1.

Table1. Notations

T	total planning horizon	K	number of resources
EF_{ij}	earliest finish time of activity j of project i	P_{ij}	set of immediate predecessors of activity j of project i
LF_{ij}	latest finish time of activity j of project i	S_{ij}	set of immediate and all successors of activity j of project i
β_k	constant per period availability of resource type k	\bar{S}_j	set of all total successors of activity j of project i
b_{ijk}	per period resource usage of resource type k by activity j of project i	EFT_{ij}	earliest finish time of activity j of project i
EST_{ij}	earliest Start Time for activity j of project i	LFT_{ij}	latest finish time for activity j of project i
LST_{ij}	latest Start Time for activity j of project i	$v(ij)$	priority value of activity j of project i
β_{rt}	left over capacity of the renewable resource k in period t .	AS_i	set of activities already scheduled from project i
r_{ijk}	Amount of resource k required by activity j of project i	S_n	schedule set
SFT_{ij}	scheduled finish time of activity j of project i	D_n	decision set
SST_{ij}	scheduled start time of activity j of project i	k	resource type, $k = 1, \dots, K$
α	interest rate	C_{\max}	largest project completion time
A_t	set of activities being in progress in period t	C_i	completion time of project i
T_H	occurrence time of last payment point ($= C_{\max}$)	T_P	occurrence time of payment point P
γ_{kt}^S	number of skilled workers k working /unit time	$\gamma_{kt}^{S,OT}$	number of skilled workers k overtime / unit time
γ_{kt}^C	number of casual workers k working / unit time	γ_{kt}^{SC}	number of special casual workers k working / unit time
γ_{kt}^M	number of machines k operating / unit time	γ_{kt}^{MI}	number of machines k ideal / unit time
γ_{kt}^{MM}	number of machines k under maintenance / unit time	γ_{kt}^{NR}	amount of nonrenewable resource k consumption / unit time of project i
γ_{kt}^{NRI}	amount of nonrenewable resource k ideal / unit time	PTI_k^S	payment time interval for skilled workers k
PTI_k^C	payment time interval for casual workers type k	PTI_k^M	payment time interval for machines k
PTI_k^{NR}	payment time interval for non renewable resources k	A_{kt}^S	maximum availability of skilled workers k / unit time
A_{kt}^C	maximum availability of casual workers k / unit time	A_{kt}^{NR}	maximum availability of non renewable resource k / unit time
W_{ikt}^{SOT}	skilled workers k overtime cost / unit time /worker of project i	w_{kt}^S	skilled workers k wage / unit time /worker
W_{kt}^{MI}	machines k ideal cost / unit time /unit	W_{kt}^{MO}	machines k operating cost / unit time /unit
W_{kt}^{MM}	machines k maintenance cost / unit time /unit	W_{kt}^{CI}	casual workers k service cost / unit time /worker

W_{kt}^{CO}	causal workers k working cost including service cost / unit time / worker	W_{kt}^{SCO}	special causal workers k working cost including service cost / unit time /worker
W_{kt}^{NRI}	non renewable resource k ideal storage cost /unit / unit time	W_{kt}^{NR}	non renewable resource k consumption cost /unit / unit time
H	number of payment points	P_p^s	payment point for skilled workers payment
P_p^{MM}	payment point for machines operations payment	P_p^C	payment point for causal workers payment
P_p^{NR}	payment point for non renewable resources payment	ϕ_{ip}	promising due date for project i
δ_{pt}	penalty incurred at time period t , if project is not completed by due date ϕ_{ip}	O_{it}^H	over head cost / unit time of project i
OT_{ij}	overtime activity j of project i	CP_j	critical path activity of project i
TC	total cost of all projects	OT_t	overtime period t
M	total number of projects in the multi project	CP_i	resource unconstrained critical path length of project i .
A	cost of skilled workers	B	cost of casual workers
C	cost of Machines	D	cost of non-renewable resources
E	overhead cost	F	penalty cost

2.1 Development of framework

The project i consists of $j = 1, \dots, J$ activities with duration of d_{ij} periods. The non-preemptable activities are numbered from 0 to J , where the dummy activities 0 and J mark the beginning and the end of the project i . We consider an upper bound T_i on the project i 's makespan, and sum of maximum (T_i) upper bound on all project is T . B_r units of renewable resource r ($r \in R$) are commonly available at resource pool in each period t , $t = 1, \dots, T$. Let the activity ij requires b_{ijr} units of resource r during every period of its duration and A_t be the set of activities being in progress in period t , then a conceptual structure of RCMPSP can be put forth as follows:

Objective function: Min (Mean time delay) and Min (Total cost)

$$\text{Minimize mean project delay} = \text{Min.} \sum_{i=1}^M \sum_{t=1}^T \frac{tS_{i,J,t} - CP_i}{M} \quad (1)$$

$$\text{Where, } S_{i,J,t} \in \begin{cases} 1, & \text{if activity } J \text{ of project } i \text{ start at time } t \\ 0, & \text{otherwise} \end{cases}$$

$$\text{Minimize total cost } (TC) = \text{Min. } [A+B+C+D+E+F] \quad (2)$$

Subject to,

$$SFT_{il} \leq SFT_{ij} - d_{ij}; ij = i2, \dots, iJ, il \in P_{ij} \mid SFT_{ij} \geq 0 \quad (3)$$

$$\sum_{i=1}^M \sum_{j \in A_i} b_{ijk} \leq \beta_{kt} \quad k \in R, t = 1, \dots, T \quad (4)$$

$$\sum_{i=1}^M \gamma_{kti}^{S,OT_{ij}} \leq \sum_{k=1}^K t \cdot x_{kt}^{S,off} \mid OT_{ij} \in CP_{ij} \quad (5)$$

Where, $x_{kt}^{S,off} = \begin{cases} 1, & \text{Skilled worker } k \text{ cyclic off day on period } t \\ 0, & \text{Otherwise} \end{cases}$

The objective function (1) minimizes the mean project delay of all projects, i.e., its equivalent to minimizing the mean resource-constrained completion time of the projects. The objective function (2) minimizes total cost of the projects. Constraint (3) takes into consideration the precedence relations between each pair of activities (il, ij) , where il immediately proceeds ij . Total resource usages at time t of any resource type k is less than its availability and is given by resource constraint (4). Finally, over time constraint set (5) limits the skilled worker requirement is not exceed the number of skilled workers available on their predetermined off day.

Expression for TC

TC is also used as the fitness function. The fitness level is used to associate a probability of selection with each individual schedule. The total cost function of the model covers six costs that may occur during the execution of the project.

Therefore,

TC of the all projects = sum of the [cost of skilled workers + cost of casual workers + cost of machines + cost of non-renewable resources + overhead cost + penalty cost]

In each payment point the interest rate α is included to find the total cost of individual resources and the sum of the total cost is calculated at the end of project.

Cost of skilled workers (A)

The cost of skilled workers can be classified as cost of regular workers and cost of overtime workers.

$$\text{Sum of skilled workers cost: } \sum_{P_p^s}^H \left[\left(\sum_{k=1}^K \sum_{t=1}^{PTI_k^s} w_{kt}^s \right) + \left(\sum_{i=1}^M \sum_{k=1}^K \sum_{t=1}^{PTI_k^s} w_{ikt}^{SOT} \cdot x_{cp_{ijkt}}^s \right) \right] \cdot (1 + \alpha)^{(C_{\max} - T_p)}$$

where,

$$x_{cp_{ijkt}}^s = \begin{cases} 1, & \text{if skilled workers type } k \text{ performing overtime at time point } t \text{ for critical activity } j \text{ of project } i \\ 0, & \text{Otherwise} \end{cases}$$

Cost of casual workers (B)

The cost of casual workers can be classified as cost of regular casual workers which belongs to cost of utilizing as well as keeping casual workers idle and cost of special casual workers. In each time period limited number of casual workers available without vacation. Work allotment to the casual workers basis on first come first serve and wages to be paid on every day. We also pay the minimum service charge to ideal casual workers on that particular day. Payment occurs on every day. In case casual workers shortages on any particular day, we immediately hire few special casual workers temporarily on the basis of high wage, which also limited availability per time period.

$$\text{Sum of casual workers cost: } \sum_{t=1}^{C_{\max}} \sum_{k=1}^K \left[\left(\gamma_{kt}^C \cdot w_{kt}^C \right) + \left(\left(A_{kt}^C - \gamma_{kt}^C \right) \cdot w_{kt}^{CI} \right) + \left(\gamma_{kt}^{SC} \cdot w_{kt}^{SCO} \right) \right] \cdot (1 + \alpha)^{(C_{\max} - t)}$$

Cost of Machines (C)

The cost of machines can be classified as cost of operating, preventive maintenance cost and ideal cost of machines.

Sum of the machines cost:

$$\sum_{p=1}^H \left[\sum_{i=1}^M \sum_{k=1}^K \sum_{t=1}^{PTI_k^M} w_{kt}^{MO} \cdot x_{kit}^o \right] + \left[\sum_{k=1}^K \sum_{t=1}^{PTI_k^M} w_{kt}^{MI} \cdot x_{kt}^I \right] + \left[\sum_{k=1}^K \sum_{t=1}^{PTI_k^M} w_{kt}^{MM} \cdot [1 - (x_{kt}^o + x_{kt}^I)] \right] \cdot (1 + \alpha)^{(C_{\max} - T_p)}$$

$$x_{kt}^o = \begin{cases} 1, & \text{if machine type } k \text{ working at time point } t \text{ of project } i \\ 0, & \text{otherwise} \end{cases}$$

$$x_{kt}^I = \begin{cases} 1, & \text{if machine type } k \text{ ideal at time point } t \\ 0, & \text{otherwise} \end{cases}$$

Cost of non-renewable resources (D)

The nonrenewable resources are available on a total project basis, with limited consumption availability for the entire project. A nonrenewable resource is called redundant if the sum of the maximal requests of the activities for this resource does not exceed its availability. Let us assume nonrenewable resources are available at start of the project.

$$\text{Sum of the non-renewable resources cost: } \sum_{p=1}^H \left[\sum_{k=1}^K \sum_{t=1}^{PTI_k^{NR}} \left(\sum_{i=1}^M \gamma_{kit}^{NR} \cdot w_{kt}^{NR} \right) + \left(\gamma_{kt}^{NRI} \cdot w_{kt}^{NRI} \right) \right] \cdot (1 + \alpha)^{(C_{\max} - T_p)}$$

Overhead cost (E)

The project overhead cost associated with the well being and comfort for project exertion process. The overhead cost linear with project duration, and also consider that the overhead cost payment occurring on daily.

$$\text{Overhead cost: } \sum_{i=1}^M \sum_{t=1}^{C_i} O_{it}^H (1 + \alpha)^{(C_{\max} - t)}$$

Penalty cost (F)

If the project is delayed beyond its due date, lump sum penalty δ_{pt} is incurred per day. Penalty cost per unit time is independent on the number of activities delayed. When there are only penalties for late completion and no reward for early completion. In project p the dummy end activity J starts beyond due date ϕ_{ip} , penalty δ_{pt} will incur at time $t = \phi_{ip} + 1$.

$$\text{Penalty cost: } \sum_{i=1}^M \delta_{pt} \cdot \sum_{t+\phi_{ip}=t}^T x_{iJt}$$

$$x_{iJt} = \begin{cases} 1, & \text{if activity } J \text{ not started at time } t \text{ of project } i \\ 0, & \text{otherwise} \end{cases}$$

The initial aim of the cyclic resource vacation is to prepare a resource calendar based on the limited resource availability. Resources like machines which consider that routine preventive maintenance / predetermined services, similar like employees predetermined/cyclic holidays to find actual staffing levels on each time period. We allocate vacation to each resource on different combination days from holiday list and extend the cyclic holidays allocation on each resource up to the maximum time period T. The preparation step of the vacation calendar is shown in Fig.1.

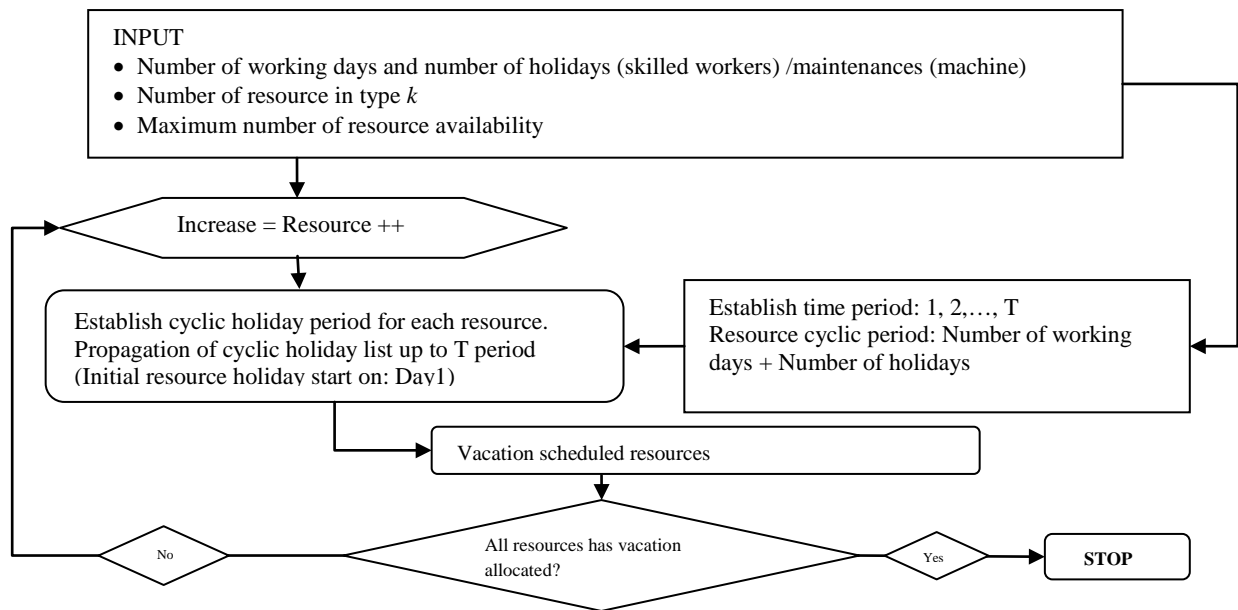


Figure 1. Propagation of the resource vacation calendar

2.2. Serial schedule generation algorithm for generating initial feasible schedules

In the section, we adopt the serial schedule generation (SSG) algorithm to build the initial population for GA. The schedule representation is based on precedence feasible of the set of activities. In GA, the genotypes are transformed into schedules using a serial scheduling scheme. In order to generate the initial 30 schedules, we incorporate convex combination of three effective priority rules, namely, minimum latest finish time (LFT), and minimum slack (MINSLK) and maximum total work content (MAXTWK) using multi-pass priority rule based method. The serial scheduling consists of $n = \sum_{i=1}^M \sum_{j=1}^J ij$ stages, in each of which one activity is selected and scheduled. Two disjoint sets S_n and D_n contains the activities that are either scheduled or partially scheduled and the unscheduled activities with every predecessor being in the scheduled set, respectively. Therefore, a serial schedule generation scheme algorithm can be described as follows:

/* Initialization

Set $n=1$, $S_n = \varphi$

While $|S_n| < J$ Do stage n

Compute D_n and $\beta_n, t = 1, \dots, T, r \in k$

$$D_n = \{ij \mid ij \notin S_n, P_{ij} \subseteq S_n\}; \quad \beta_n = \beta_r - \sum_{i=1}^M \sum_{j \in A_i} b_{ij}$$

/* Convex combination of priority rules

$$\text{LFT}, v_1(ij) = \text{Min.} LFT_{ij}$$

$$\text{MINSLK}, v_2(ij) = \text{Min.} (LST_{ij} - EST_{ij})$$

$$\text{MAXTWK, } v_3(ij) = \left(\sum_{k=1}^K \sum_{ij \in AS_i} d_{ij} r_{ijk} + d_{ij} \sum_{k=1}^K r_{ijk} \right)$$

Here,

$$\sum_{k=1}^K \sum_{ij \in AS_i} d_{ij} r_{ijk} : \text{Total work scheduled for the activities already scheduled (} AS_i \text{) of the project } i$$

$$d_{ij} \sum_{k=1}^K r_{ijk} : \text{Total work consumed by activity } j \text{ of project } i$$

/* Combination method

$$v(ij) = \sum_{m=1}^3 w_m v_m(ij) \mid w_m \geq 0, \sum_{m=1}^3 w_m = 1$$

/* Activity selection based on priority value

$$ij^* = \min_{ij \in D_n} \{ij \mid v(ij)\}$$

$$EFT_{ij^*} = \max\{SFT_{il} \mid il \in P_{ij^*}\} + d_{ij^*}$$

$$SFT_{ij^*} = \min\{t \mid EFT_{ij^*} \leq t \leq LFT_{ij^*}, b_{ij^*k} \leq \beta_{k\tau}, \tau = t - d_{ij^*} + 1, \dots, t, r \in k\}$$

$$S_{n+1} = S_n \cup \{ij^*\}$$

$$n = n + 1$$

End

Stop

The algorithm terminates at stage $n = J$ when entire activities in all projects are in the scheduled set. Overtime allowed for skilled workers on their vacation period and limited to work progress on the critical activities. Also we outsource the special casual labours, when casual labours are shortages on that particular time period. Fig.2 depicts the serial schedule generation scheme for total cost project model.

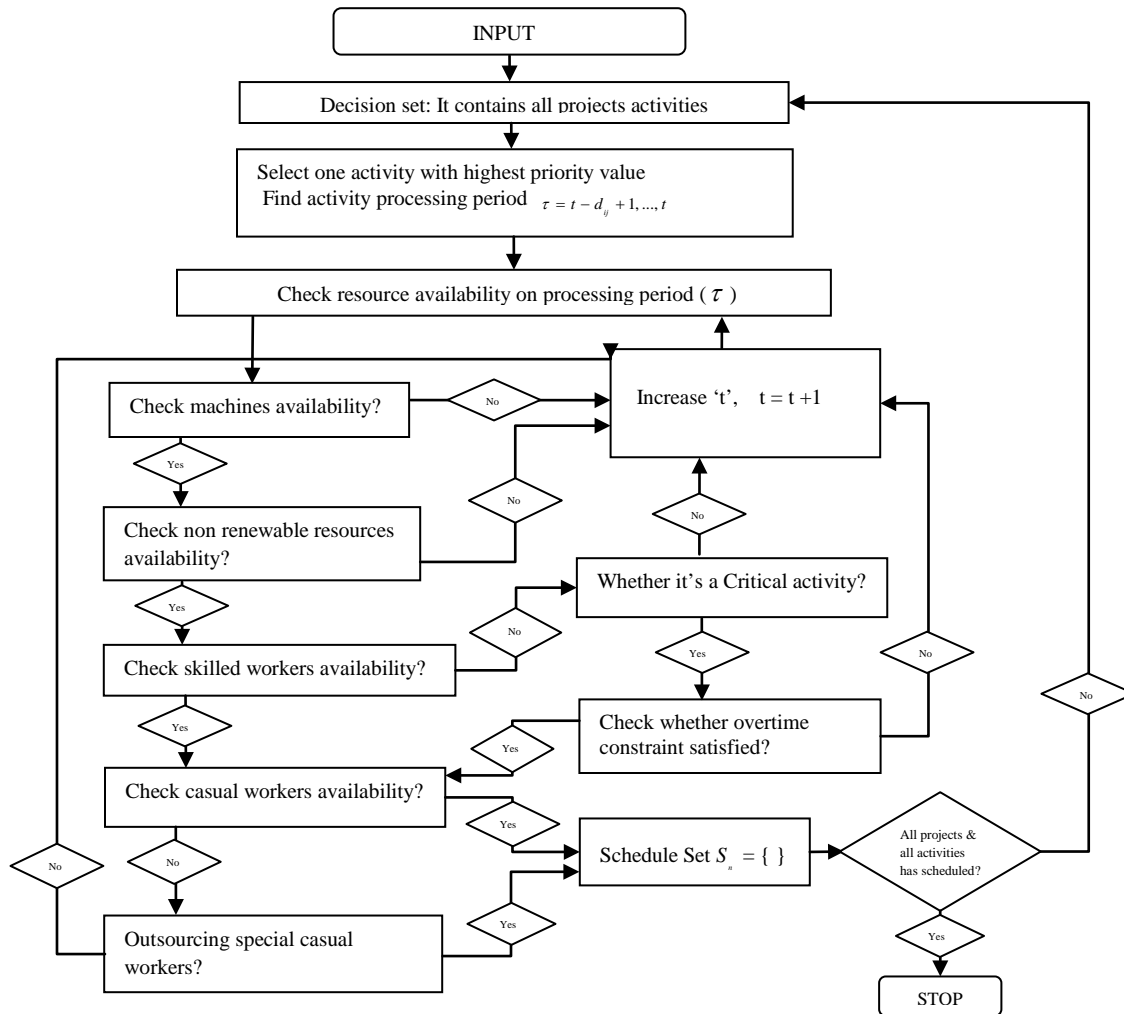


Figure 2. Flowchart of the serial schedule generation scheme

2.3. A decision support systems using genetic algorithm for the RCMPSP

GA is, in essence, an optimization search procedures inspired by the logical system's improved fitness through evolution. A solution for the RCMPSP is represented by a chromosome that contains an activity sequence for the problem. The genotypes are transformed into schedules using a serial scheduling scheme. Each gene in a chromosome stands for an activity number. Technique for Order Preference by Similarity to Ideal Solution (TOPSIS) is a suitable method to measure the relative performance of each schedule alternative and thus ranks the preference

order to obtain the best compromise one [Hwang and Yoon (1981)]. We use TOPSIS method in the selection processes for sorting the schedules. Given that m alternatives schedules are evaluated in terms of makespan and cost.

We select the five best schedules for the purpose of upgrading the quality of initial population. We have used repetitive mutation operator to upgrade the initial population. The following repetitive mutation algorithm is applied 30 times to each selected schedule individually and we save all resulting schedules in repetitive mutation pool.

/ Repetitive mutation algorithm*

Initialize $n = 1$,

Initial Schedule $\lambda_1 = (ij_1, \dots, ij_J)$

While $n \leq 30$

Randomly select one activity j_l from project i , between $j_3 \leq j_l \leq j_{J-1}$

Find all immediate predecessors and select one activity (j_h) which has highest schedule finish time from same project i .

$\lambda_n = (j_1, \dots, j_h, \dots, j_l, \dots, j_J) \mid j_h \in P_l \ \forall h \in \text{high index value}$

That is, activity j_l is left shifted within the activity list and inserted immediately after the activity j_h of same project i . Clearly shown, this resulting activity list is still precedence feasible.

$\lambda'_n = (j_1, \dots, j_h, j_l, j_{h+1}, \dots, j_{l-1}, j_{l+1}, \dots, j_J)$

Assign, $\lambda_{n+1} := \lambda'_n ; \forall n$

$n = n + 1$

End

Stop

In the next, we select the best 5 schedules from 150 resulting schedules and add them into the initial population pool created by SSGS. Then we sort-out best 30 schedules for GA initial population. Its split into three equal sets based on the ranking: top 10 schedules in Set-A, moderately fit 10 schedules in Set-B and remaining 10 schedules falling in Set-C. We apply Hartmann's (1998) two point, uniform crossover strategy where precedence feasible parents' activity lists are proved to produce precedence feasible offsprings. For two point crossover, the father is from the fittest Set-A and the mother is randomly selected from Set-C. In the uniform crossover father is from Set-B and mother is chosen randomly from the Set-C. The two point crossover produces two off springs; a son and a daughter, while uniform crossover produces just one son.

In each generation 30 offsprings are produced and then added with parents to sort out the best 24 schedules for the next generation. The precedence based swap mutation operator is used here, which simply selects two activities at random and if precedence relationship is satisfied perform swaps, otherwise selects another position randomly until swapping is possible. We use swap mutation probability 0.034; one schedule from the offspring pool is randomly selected for performing the swap mutation and is added to the next generation parent pool. Next 5 schedules from the repetitive mutation pool are randomly selected and added to the next generation's initial population. We have used satisfactory termination criterion, 100 schedules and 500 schedules for the purpose of the performance measure of the algorithm. The Fig.3 depicts the flowchart of the proposed GA.

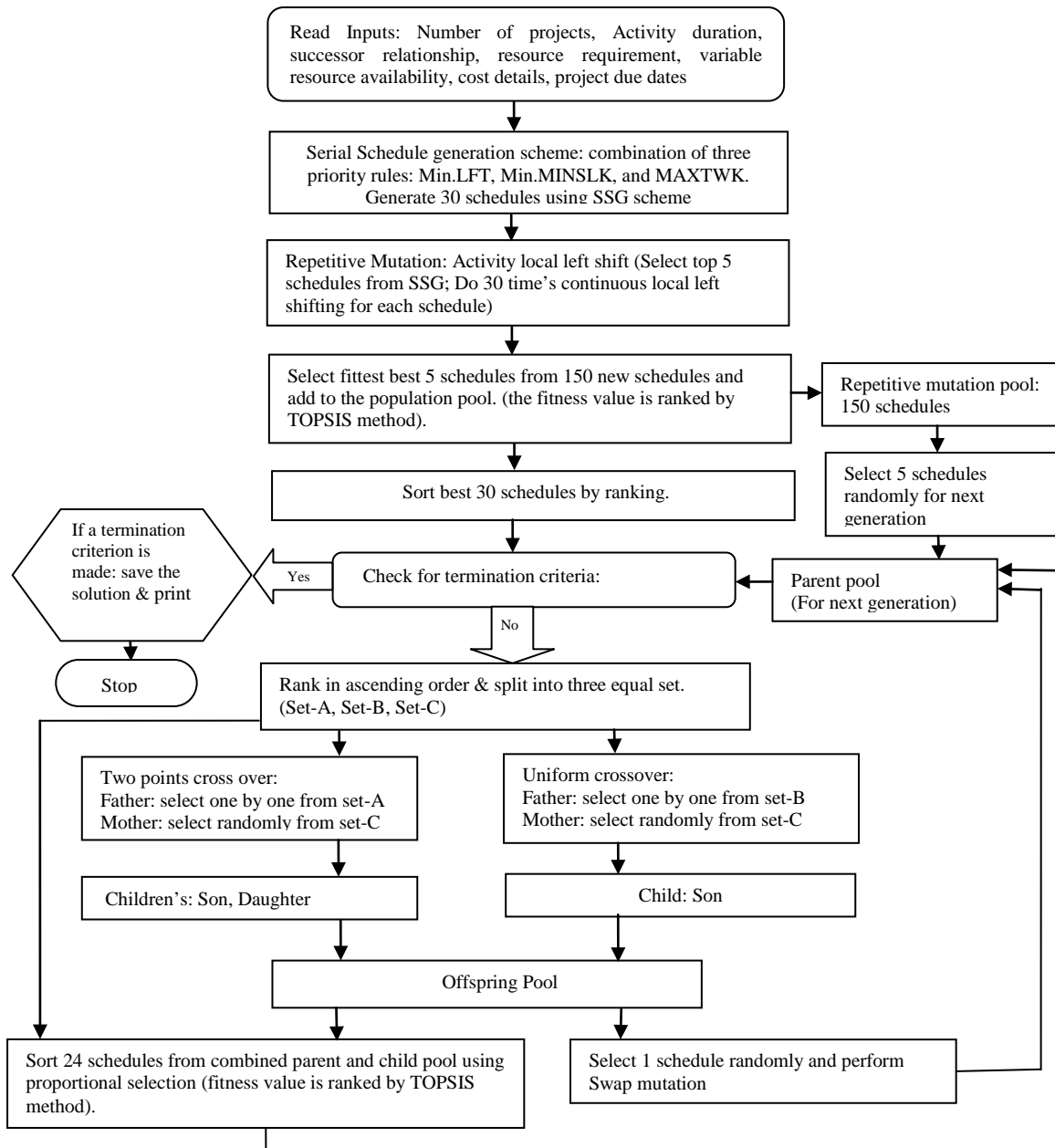


Figure 3. Flowchart of the genetic algorithm

3. Computational experiments

In this section, we investigate the effectiveness of the proposed evolutionary strategy. The experiment is performed on a Pentium 4 CPU at 2.67 GHz and 512 MB of RAM. The algorithm has been coded in C #.

3.1 RCMPSP test project

We have used the standard network model from existing literature. These instances have been modified according to our model. The precedence network diagram of the test project-1(P-1), test project-2(P-2), test project-3(P-3) are shown in Fig.4, 5, 6 respectively.

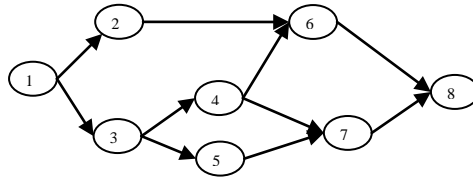


Figure 4. Network diagram of test project-1

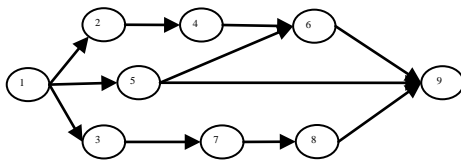


Figure 5. Network diagram of test project-2

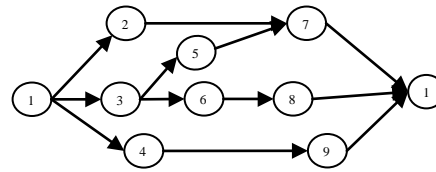


Figure 6. Network diagram of test project-3

We perform the test projects with three renewable resources and one non renewable resource, the duration and resource requirements are shown in table 2 to 4.

Table 2. Activity details and resource usage for test projects -1, 2, 3

Activity	Successor			Duration (days)			Renewable resource requirement, units /day									Non renewable resources requirement, units / day		
							Skilled worker			Casual Labour			Machines			Materials		
	P-1	P-2	P-3	P-1	P-2	P-3	P-1	P-2	P-3	P-1	P-2	P-3	P-1	P-2	P-3	P-1	P-2	P-3
1	2,3	2,3,5	2, 3, 4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	6	4	7	3	2	2	5	2	2	3	2	2	2	1	4	5	5	2
3	4,5	7	5, 6	2	6	3	2	5	1	4	4	2	2	4	2	2	3	3
4	6, 7	6	9	3	3	1	5	1	3	2	2	3	3	1	3	4	3	4
5	7	6,9	7	4	4	2	6	2	2	3	2	2	2	2	4	5	4	3
6	8	9	8	2	5	2	2	3	1	4	4	2	3	3	2	3	2	2
7	8	8	10	5	5	3	5	2	2	3	2	3	4	2	3	5	4	1
8	-	9	10	0	4	3	0	3	3	0	2	3	0	2	4	0	3	2
9	-	-	10	--	0	5	--	0	2	--	0	2	--	0	2	--	0	3
10	--	--	--	--		0	--	--	0	--	--	0	--	--	0	--	--	0
Upper resource limit / day							18			10 & (Special casual workers : 5)			15			228		

Table 3. Cyclic resource vacation and cost details for test projects-1, 2, 3

Resource details	Renewable resources			Non renewable resources
	Skilled workers	Casual labours	Machines	Materials
				P-1
Continuously working / available days	6	All days	10	All days
Cyclic holiday /preventive maintenance day	1	0	1	0
Cost / day/ unit , \$	110	85	150	90
Ideal charge / day, \$	0	15	50	1
Maintenance cost / day	0	0	95	0

Table 4. Specific project, resource details for test projects-1, 2, 3

Specific project / resource details	Project-1	Project-2	Project-3
Project due date, (in days)	15	19	11
Penalty cost / day, \$	500	300	500
Overhead cost, \$ / day	150	120	130
Special casual worker cost, \$ /day	125	125	125
Overtime cost for skilled worker	220	220	220
Interest Rate, % / day	0.007	0.007	0.007
Payment period for skilled worker (in days)	30	30	30
Payment period for casual / special casual workers,(in days)	1	1	1
Payment period for machines, (in days)	7	7	7
Payment period for materials, (in days)	10	10	10
Payment period on overhead cost	Daily	Daily	Daily
Payment period on penalty cost	c_1	c_2	c_3

The optimum results obtained those projects are shown in table 5. Fig.7 and 8 shows the progress of improvement of objective functions with number of total schedules generated. After conducting extensive experiments, it is found that the increase in the number of schedules generated beyond 360 does not change the mean project delay, projects total cost value of the proposed GA.

Table 5. Results of the test projects-1, 2, 3 and their % deviations from critical path value

Objective weights (TOPSIS Method)		Max. no. of schedules									
		100					500				
Mean projects delay w_1	Total cost w_2	Makespan Duration (in days)			Mean projects delay	Total cost Amount (in \$)	Makespan Duration (in days)			Mean projects delay	Total cost Amount (in \$)
		P1	P2	P3			P1	P2	P3		
0.0	1.0	17	22	12	5.66	36836.68	20	21	15	7.33	21968.84
0.1	0.9	17	22	12	5.66	36836.68	16	21	18	7.00	21787.66
0.2	0.8	17	22	12	5.66	36836.68	19	21	14	6.66	21121.91
0.3	0.7	17	22	13	6.00	36895.34	16	19	21	7.33	21096.33
0.4	0.6	17	22	12	5.66	36836.68	20	20	15	7.00	20074.59
0.5	0.5	17	22	12	5.66	36836.68	20	20	12	6.00	21234.54
0.6	0.4	17	22	12	5.66	36836.68	20	20	15	7.00	21105.04
0.7	0.3	17	22	12	5.66	36836.68	20	20	16	7.33	22068.19
0.8	0.2	17	22	12	5.66	36836.68	20	21	11	6.00	22171.49
0.9	0.1	17	22	12	5.66	36836.68	20	20	15	7.00	22840.38
1.0	0.0	18	23	11	6.00	38222.07	18	20	13	5.66	23063.38

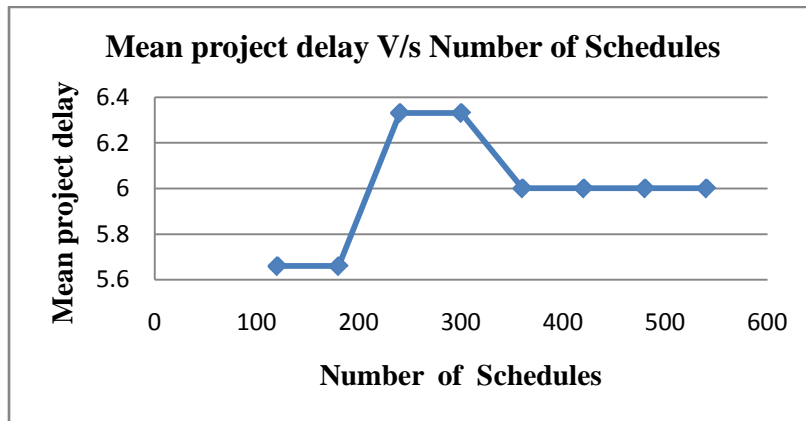


Figure 7. Mean project delay over number of schedules for test projects 1, 2, 3 [Given, $w_1 = 0.5$; $w_2 = 0.5$]

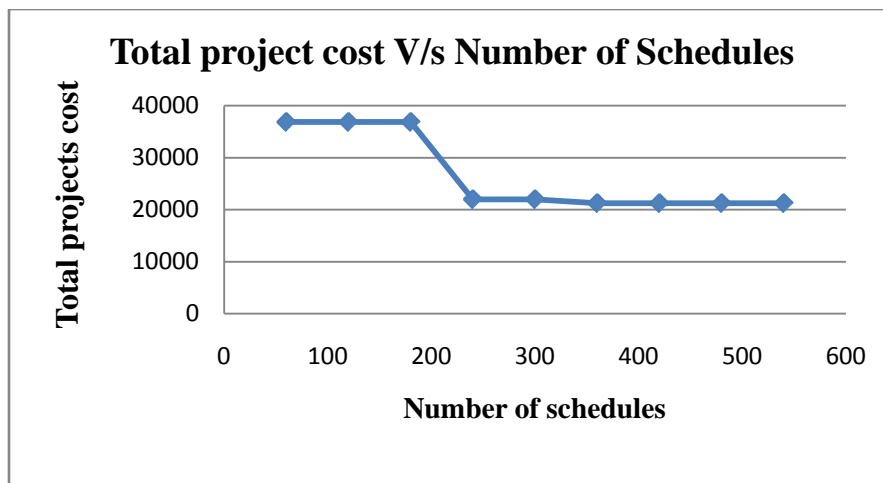


Figure 8. Total projects cost over number of schedules for test projects 1, 2, 3 [Given, $w_1 = 0.5$; $w_2 = 0.5$]

4. Conclusions

In this paper, we have proposed a decision support system for multi projects scheduling under limited resources to minimize both the total cost, the constraints cyclic-off-days for workers, cyclic preventive maintenance of machines, overtime for critical activities and non-renewable resource utilization. In order to improve the performance of the objectives of the model, we used TOPSIS method for sort-out the best compromise schedule. The proposed algorithm is coded in C#. For future research, at least two issues are worth investigating. Firstly, it would be interesting to generalize the present model to include uncertainties in resources vacation. Secondly, setup time and the resource transfer time between the projects models can be considered as a promising direction to develop further heuristics that can be built as a powerful DSS in the RCMPSP.

References

- Abbasi, B., S.Shadrokh, J.Arkati. 2006. Bi-objective resource-constrained project scheduling with robustness and makespan criteria. *Applied Mathematics and Computation*, 180,146-152.
- Al-Fawzan, M.A., M. Haouari. 2004. A bi-objective model for robust resource-constrained project scheduling. *International Journal of Production Economics*, 18,1-13.
- Anavi-Isakow, S.A., B. Golany. 2003. Managing multiple-project environments through constant work-in-progress. *International Journal of Project Management*, 21(1), 9-18.
- Ash, R., D.E. Smith-Daniels. 1999. The effects of learning, forgetting, and relearning on decision rule performance in multiproject scheduling. *Decision Sciences*, 30 (1), 47-82.
- Azaron, A., R.T.Moghaddam. 2007. Multi-objective time–cost trade-off in dynamic PERT networks using an interactive approach. *European Journal of Operational Research*, 180(3), 1186-1200.

- Belfares,L., W. Klibi, N. Lo, A. Guitouni. 2007. Multi-objectives Tabu Search based algorithm for progressive resource allocation. *European Journal of Operational Research*, 177, 1779-1799.
- Bottcher, J., A. Drexl, R. Kolisch, F.Salewski. 1999. Project scheduling under partially renewable resource constraints. *Management Science*, 45, 543-559.
- Bock, D.B., J.H.Patterson. 1990. A comparison of due date setting, resource assignment, and job preemption heuristics for the multiproject scheduling problem. *Decision Sciences*, 21(2), 387-402.
- Chen,P.H., S.M.Shahandashti. 2009. Hybrid of genetic algorithm and simulated annealing for multiple project scheduling with multiple resource constraints. *Automation in Construction*, 18, 434-443.
- Davis,L. 1985. Job shop scheduling with genetic algorithms, in: *Proceedings of the 1st international conference on genetic algorithms*.
- Drexl, A., J.Grunewald.1993.Nonpreemptive multi-mode resource-constrained project scheduling. *IIE Transactions*, 25 (5), 74-81.
- Dumond, J., V.A. Mabert. 1988. Evaluating project scheduling and due date assignment procedures: An experimental analysis. *Management Science*, 34 (1), 101-118.
- Elloumi,S., P. Fortemps. 2010. A hybrid rank-based evolutionary algorithm applied to multi-mode resource-constrained project scheduling problem. *European Journal of Operational Research*, 205(1), 31-41.
- Fendley, L.G. 1968. Towards the development of a complete multiproject scheduling system. *Journal of Industrial Engineering*, 19 (10), 505-515.
- Feng, C., L. Liu, S. Burns. 1997. Using genetic algorithms to solve construction time-cost trade-off problems. *J. Comput. Civ. Eng., ASCE* 11 (3), 184-189.
- Goncalves,J.F., J.J.M. Mendes, M.G.C. Resende. 2008. A genetic algorithm for the resource constrained multi-project scheduling problem. *European Journal of Operational Research*, 189(3), 1171-1190.
- Hansen,P. 1997. Tabu search for multiobjective optimization: MOTS. Technical Report, Institute of Mathematical Modelling, Technical University of Denmark.
- Hapke, M., A. Jaszkiwicz, R. Slowinski. 1998. Interactive analysis of multiple-criteria project scheduling problems, *European Journal of Operational Research*, 107, 315-324.
- Hartmann, S. 1998. A competitive genetic algorithm for resource constrained project scheduling. *Naval Research Logistics*, 45, 279-302.

- Hartmann, S. 1999. Project Scheduling under Limited Resources: Models, Methods, and Applications. Springer-Verlag, Berlin, Germany.
- Hwang, C.L., K. Yoon. 1981. Multiple Attribute Decision Making Methods and Applications. Springer, Berlin, Germany.
- Kim, K.W., G. Mitsu, G. Yamazaki. 2003. Hybrid genetic algorithm with fuzzy logic for resource-constrained project scheduling. *Applied Soft Computing*, 2, 174-188.
- Kruger, D., A. Scholl. 2009. A heuristic solution framework for the resource constrained (multi)project scheduling problem with sequence-dependent transfer times, *European Journal of Operational Research*, 197(2), 492-508.
- Kumanan, S., J. J. Jose, K. Raja. 2006. Multi-project scheduling using an heuristic and a genetic algorithm. *International Journal of Advanced Manufacturing technology*, 31, 360-366.
- Kurtulus, I.S., E.W. Davis. 1982. Multi-project scheduling: Categorization of heuristic rules performance. *Management Science*, 28 (2), 161-172.
- Kurtulus, I.S., S.C. Narula. 1985. Multi-project scheduling: Analysis of project performance. *IIE Transactions*, 17 (1), 58-66.
- Lawrence, S.R., T.E. Morton. 1993. Resource constrained multi-project scheduling with tardy costs: Comparing myopic, bottleneck, and resource pricing heuristics. *European Journal of Operational Research*, 64 (2), 168-187.
- Long, L.D., A. Ohsato. 2009. A genetic algorithm-based method for scheduling repetitive construction projects. *Automation in Construction*, 18(4), 499-511.
- Lova, A., P. Tormos. 2001. Analysis of scheduling schemes and heuristic rules performance in resource-constrained multiproject scheduling. *Annals of Operations Research*, 102, 263-286.
- Lova, A., C. Maroto, P. Tormos. 2000. A multicriteria heuristic method to improve resource allocation in multi-project scheduling. *European Journal of Operational Research*, 127 (2), 408-424.
- Mohanty, R.P., M.K. Siddiq. 1989. Multiple projects-multiple resources-constrained scheduling: Some studies. *International Journal of Production Research*, 27 (2), 261-280.
- Nagar, A., J. Haddock, S. Heragu. 1995. Multiple and bi-criteria scheduling: a literature survey, *European Journal of Operational Research*, 81, 88-104.
- Patterson, J.H. 1973. Alternate methods of project scheduling with limited resources. *Naval Research Logistics Quarterly*, 20 (4), 767-783.

Payne, J.H. 1993. Introducing formal project management into a traditional, functionally structured organization. *International Journal of Project Management*, 11 (4), 239-243.

Payne, J.H. 1995. Management of multiple simultaneous projects: a state-of-the-art review, *International Journal of Project Management*, 13(3), 163-168.

Pritsker, L.J., A.A.B.Watters, P.M.Wolfe. 1969. Multiproject scheduling with limited resources: A zero-one programming approach. *Management Science*, 16(1), 93-108.

Senouci, A., H.R.A.Derham. 2008. Genetic algorithm-based multi-objective model for scheduling of linear construction projects. *Advances in Engineering Software*, 39, 1023-1028.

Slowinski, R. 1989. Multiobjective project scheduling under multiple-category resource constraints, in: R. Slowinski, J. Weglarz (Eds.), *Advances in Project Scheduling*, Elsevier, Amsterdam.

Viana, A., J.P.D.Sousa. 2000. Using metaheuristics in multiobjective resource constraints project scheduling. *European Journal of Operational Research*, 120, 359-374.

Valls, V., F.Ballestin, S. Quintanilla. 2008. A hybrid genetic algorithm for the resource-constrained project scheduling problem. *European Journal of Operational Research*, 185(2), 495-508.

Vercellis, C. 1994. Constrained multi-project planning problems: A Lagrangian decomposition approach. *European Journal of Operational Research*, 78 (2), 267-275.

Wiley, V.D., R.F.Deckro, J.A. Jackson. 1998. Optimization analysis for design and planning multi-project programs. *European Journal of Operational Research*, 107 (2), 492-506.

Yang, K.K., C.C.Sum. 1993. A comparison of resource allocation and activity scheduling rules in a dynamic multi-project environment. *Journal of Operations Management*, 11 (2), 207-218.

Yang, K.K., C.C.Sum. 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(1), 139-154.