

A Memetic Algorithm for Maximizing Net Present Value in Resource-Constrained Project Scheduling Problem

Angela H. L. Chen and Chiuh-Cheng Chyu

Abstract—In this study, we develop a model that considers monetary issues in resource-constrained environments, and involves scheduling project activities to maximize net present value. This problem is recognized as the “resource-constrained project scheduling problem with discounted cash flows (RCPSPDCF),” which is strongly NP-hard. All resources considered are both types of renewable and nonrenewable; the duration of each activity depends on the amount of resources allocated to its execution. Efforts are made by considering a two-stage method applying mode selection rules at the first stage and the memetic algorithm at the second stage. Results are shown in a comparative study which demonstrates the effectiveness of using memetic algorithm in maximizing project net present value; as well as, a combination of mode selection rules which provide a high probability of giving the best solution.

I. INTRODUCTION

IN the project management and scheduling a number of different objectives can be investigated. While three most commonly used objectives are: to minimize project makespan in [1]–[3], to minimize the overall project cost in [4], [5], and to maximize project present value in [6]–[10], most studies have focused more on makespan problem but discussed less on cost and present value problems. In practices, while meeting the project deadline is equivalently important, most contractors consider maximizing financial returns the main objective in commercial projects. Thus only relatively recently, this topic is being investigated extensively under the title of maximizing project Net Present Value (NPV) and recognized as the “Resource-Constrained Project Scheduling Problem with Discounted Cash Flows (RCPSPDCF),” a strongly NP-hard problem.

Since the introduction on scheduling to maximize project net present value by Russell in [10], several contributions have been published and we can therefore roughly divided problems based on network, resource, and payment types. For instances, while unconstrained Problems are Russell in [10], Etgar *et al.* in [17], Smith-Daniels and Aquilano in [14], Vanhoucke *et al.* in [18], constrained problems are Talbot in

[5], Icmeli and Erenguc in [19], and Ulusoy *et al.* in [20].

So far many methods have been developed to obtain good solutions but none dominates in all project environments in [8], [9], [13], [14], [19], [21]. Methods using priority rules to maximize the NPV are: for example, Sepil and Ortaç in [13], and Pinder and Maruchek in [9]. Also, methods concerning exact procedures are as follows: Patterson *et al.* in [8], Icmeli and Erenguc in [19], Shtub and Etgar in [22], De Reyck and Herroelen in [23], and Vanhoucke *et al.* in [24]. Though metaheuristics have been applied in solving maximization of makespan in project scheduling mostly, but haven’t been widely applied in solving NPV optimization. Few articles found in our literature review are such Etgar *et al.* in [25] proposed a simulated annealing approach, Ulusoy *et al.* in [20] used a genetic algorithm with a special crossover operator, Mika *et al.* in [26] implemented local search in their simulated annealing and tabu search.

Our study implements memetic algorithms, which have differentiated itself from the Genetic algorithm classification in [33], and [34]. Genetic algorithms try to emulate biological evolution where individuals cannot choose, modify and improve their own genes in its natural process; yet, memetic algorithms try to mimic cultural evolution where individuals can intentionally acquire, modify, and improve their memes [35]. This metaheuristic has been proven to be very effective in solving many hard combinatorial optimization problems, such as Traveling salesman problem in [27]–[30], Quadratic assignment problem in [31], and Job shop scheduling problem in [32], yet has not been applied in RCPSPDCF.

The RCPSPDCF problem in this study assumes an individual activity may be performed in one of several ways, each with its unique time and cost requirements. Then, all the activities should be completed by a common due date. A series of cash flows may occur over the course of a project in two forms: cash outflows refer to expenditures for labor, equipment, materials, etc., and cash inflows incur in the form of payment. Time value of money is taken into consideration through discounting the cash flows. To generate feasible solutions for our multi-mode RCPSPDCF, we employ the memetic algorithm (MA) to calculate and assign priority to each activity for scheduling. In addition to deal with mode assignment problem, several mode selection rules (MSR) have been developed. Few of these rules are modifications of heuristic procedures developed for the RCPSP with renewable resources only in [1], [36]; other rules are developed based on the cost concept.

The next section introduces the mathematical model of the RCPSPDCF. Section 3 describes the scheduling heuristics tested in this work. Section 4 presents the experimental design

A. H. L. Chen is a Ph. D. candidate at the Industrial Engineering and Management Department, Yuan-Ze University, Taoyuan County, R. O. C. She also teaches at the Finance Department, Nanya Institute of Technology, Taoyuan County, R. O. C. (corresponding author to provide phone: 886-92077-6682; e-mail: achen@nanya.edu.tw).

C. C. Chyu is an Associated Professor at the Industrial Engineering and Management Department, Yuan-Ze University, Taoyuan County, ROC. (e-mail: iehshsu@saturn.yzu.edu.tw).

and computational results in this study. Then conclusions are presented in Section 5.

II. MODEL FORMULATION

As known, the multi-mode RCPSPDCF comprises a series of cash inflows and cash outflows over the duration of the project. Cash outflows occur as project expenditures, which are determined by the product of resource unit costs and usages plus other costs. Cash inflows, on the other hand, occur as payments for completed activities. Furthermore, in this model, cash inflows occur at the beginning of each activity while cash outflows at the completion of each activity. The objective is to maximize the net present value and can be represented as follows:

$$\max \sum_{m=1}^{M_i} \sum_{i=1}^I \sum_{t=EF_i}^{LFT_i} F_{im}^{in} \cdot e^{-\alpha(t-d_{im})} \cdot (x_{im} - d_{im}) - \sum_{m=1}^{M_i} \sum_{i=1}^I \sum_{t=EF_i}^{LFT_i} F_{im}^{out} \cdot e^{-\alpha t} \cdot x_{im}$$

$$\sum_{m=1}^{M_i} \sum_{t=EF_i}^{LFT_i} x_{im} = 1, \quad i = 1, \dots, I \quad (1)$$

$$\sum_{m=1}^{M_j} \sum_{t=EF_j}^{LFT_j} t x_{jmt} \leq \sum_{m=1}^{M_j} \sum_{t=EF_j}^{LFT_j} (t - d_{jm}) x_{jmt}, \quad j \in P_i, i = 1, \dots, I \quad (2)$$

$$\sum_{i=1}^I \sum_{m=1}^{M_i} r_{imk}^p \cdot \sum_{q=\max\{t, EF_i\}}^{\min\{t+d_{im}-1, LFT_i\}} x_{imq} \leq R_k^p \quad \text{for } k = 1, 2, \dots, K^p; t = 1, 2, \dots, T \quad (3)$$

$$\sum_{i=1}^I \sum_{m=1}^{M_i} r_{imk}^v \cdot \sum_{t=EF_i}^{LFT_i} x_{imt} \leq R_k^v \quad \text{for } k = 1, 2, \dots, K^v; t = 1, 2, \dots, T \quad (4)$$

$$\sum_{t=EF_i}^{LFT_i} t \cdot x_{imt} \leq T, \quad i = 1, \dots, I \quad (5)$$

$$x_{imt} \in \{0, 1\}, \quad \forall i, \forall m, \forall t \quad (6)$$

The indices in this model are:

i, j = activity index ($i, j = 1, \dots, I$)

k = resource type index

m = mode index of each activity ($m = 1, \dots, M_j$)

The parameters are:

I = total number of activities in the project

K^p = total number of renewable resource types in the project

K^v = total number of nonrenewable resource types in the project

M_j = number of modes for activity j

P_i = set of direct predecessors

T = deadline for the project

F_{im}^{out} = cash outflow for activity i if it is executed in mode m

F_{im}^{in} = cash inflow for activity i if it is executed in mode m

α = discount rate

d_{im} = duration of activity i if it is executed in mode m

r_{imk}^p = requirement of resource type k per period by activity i executed in mode m

R_k^p = amount of renewable resource type k available per period

r_{imk}^v = requirement of nonrenewable resource type k by activity i executed in mode m

R_k^v = amount of nonrenewable resource type k available for

the entire project

And the decision variables are:

$x_{imt} = 1$: if activity i finishes at time t and executes with mode m ; 0: otherwise.

Constraints (1) and (5) ensure that each activity starts at only one time period, with only one mode. The precedence relationships among activities are defined in constraint (2). Constraint (3) and (4) represents the renewable and nonrenewable resource utilization requirements and prevents resource overuse. Constraint (5) enforces the project due date by setting a target project due date T . Constraint (6) defines the decision variable as a binary one.

III. METHODOLOGY

This section describes the general framework of the combinatorial procedure for solving this problem. We first use a mode selection rule to find a feasible mode and then we employ the memetic algorithm to find the corresponding schedule. This combinatorial procedure consists of two-phase: the mode selection rules (MSR) and the memetic algorithm (MA). Details are presented next.

A. The Mode Selection Rules (MSR)

Current mode selection rules in literatures are mainly designed for renewable resources for makespan problem in [1], [3], [36], [37]. Therefore, we develop other mode selection rules for non-renewable and cost-oriented resources. *Shortest Feasible Mode (SFM)* and *Least Resource Proportion (LRP)* adopted directly (referred to [1] for implementation details) while the remaining nine rules are either modified or developed to take into account the duration of each activity mode alternative and the cost of each resource type. Details are described below:

Least Product Sum of Resource and Duration (LPSRD): For each activity, choose the execution mode which has the minimum product sum of resource usage to its corresponding mode duration, $\min_m \sum_{k=1}^K (r_{j,k,m} \times d_{j,m})$.

Least Product Sum of Resource Cost and Duration (LPSRCD): Differently from the previous one, choose the execution mode which has the minimum product sum of resource cost to its corresponding mode duration, $\min_m \sum_{k=1}^K (r_{j,k,m} \times c_k \times d_{j,m})$, for each activity.

Least Total Resource Usage (LTRU): Choose the execution mode which requires the least total resource usage, $TRU_{j,m} = \sum_{k=1}^K r_{j,k,m}$.

Least Criticality Ratio (LCR): Choose the execution mode which requires the least amount of the most critical resource type determined by $\min_m \left\{ \frac{\max_k \{r_{j,k,m}\}}{NR_k} \right\}$.

The Least Ratio Sum of Non-renewable Resource (LRS): Choose the execution mode which requires the least sum of the ratio of the non-renewable resource type to its corresponding resource limitation.

Least Total Resource Cost (LTRC): Select the mode which

contains the least total resource cost.

Least Relative Resource Consumption (LRRC, LRRC^{}):* The highest priority to modes whose assigning leaves the lowest relative resource consumption to the involved activities is determined by the following equations: $\min_m \sum_{k=1}^K (r_{jmk} / NR_k^{remain})$, and $NR_k^{remain} = NR_k - \sum_{j \in A} r_{jmk}$, where the set A includes all activities j that have already been assigned a mode m . However, for $LRRC^*$, the same set A includes only successors that have already been assigned a mode m .

Maximum Ratio of the Average Remaining Capacity (MARCR, MARCR^{}):* Give the highest priority to modes whose assigning leaves the highest average remaining capacity ratio (ARCR), $\max_m \left\{ \sum_k (NR_k^{remain} / NR_k) / K \right\}$, where the set A in $MARCR$ and $MARCR^*$ has been defined earlier in $LRRC$ and $LRRC^*$.

B. The Memetic Algorithm (MA)

The first use of the term *Memetic Algorithms (MA)* in the computing literature has appeared in [38]-[39], and is defined as an evolutionary-based search augmented by hybridizing with one or more phases of local search procedures or by the use of problem-specific information. Below, we show a pseudo-code representation of our MA and describe these elements as follows.

The initialization phase which begins at **InitializePop** and ends just before the **Repeat** command is responsible for generating an initial population. The initial population is generated by user-specified priority rules (e.g. one half of population by the weighted earliest finish time (EFT) rule and the other half by the weighted latest finish time (LFT) rule). Then, the local search is employed to all members in the population. After checking the convergence rate of the population, mutation is performed if the rate is greater than the threshold. Again, local search is applied to the offspring generated in mutation operation. If the population passes the convergence check, i.e. a certain level of diversity is maintained, recombination operation is employed, and local search is implemented to improve the offspring. Lastly, the procedure continues until the termination criteria are reached.

Algorithm MA:

Begin

Set population size = $psize$; #generations $GEN = 1$;

InitializePop P_0 ;

For each individual $i \in P_0$ **do**

$i := \text{Local-Search}(i)$;

$P = P_0$;

Repeat

If P converges then

For each $i \in P$, $i \neq \text{Best}(P)$ **do**

$i_c := \text{Mutate}(i)$;

$i_c := \text{Local-Search}(i_c)$;

Endfor;

Else

For $n := 1$ to #recombinations **do**

Repeat

 Select two parents $i_a, i_b \in P$ by tournament;

$i_c := \text{Recombine}(i_a, i_b)$;

$i_c := \text{Local-Search}(i_c)$;

 Perform similarity test for offspring i_c ;

Until pass-similarity-test;

 Add i_c to P ;

Endfor;

$P = \{\text{best } psize \text{ individuals in } P\}$;

$GEN++$;

until terminate = true;

End;

Next, we will survey all the main elements of the MA we have implemented and specify the solution representation, recombination, mutation, and local search.

1) Solution Representation

A feasible solution in this study is represented by the two arrays: one is a list of execution modes for the corresponding activities called the mode assignment and the other one is a precedence-feasible combination of activities called the activity list.

2) Selection

The selection operator ensures that the best individuals have a higher probability of surviving; for that, we use a binary tournament selection for selecting parent solutions.

3) Recombination

The main idea of the recombination operator is to generate new diversified and potentially promising offspring; therefore, a modified order-based procedure is implemented. First, a number of activities are randomly selected from one parent, and then the order of activities in the selected position is imposed onto the corresponding activities in the other parent. Then the offspring is checked for the precedence feasibility, and activities that violate their precedence relations are shifted leftward one by one to their first feasible positions.

4) Mutation

As the recombination implemented will lead to a quick loss of diversity, mutation we use here will preserve the diversity of the population. We implement a traditional mutation strategy, called adjacent swapping when the population fails to pass the convergence check. First, two adjacent positions are randomly picked; then the alleles in these two adjacent positions swap their values while the precedence feasibility is sustained. Mutation operation is applied to all members but the best one in the population. This can be considered as an Elitist strategy to preserve the best solution obtained.

5) Local Search (LS)

The local search operator is used to improve solutions in the configuration space before inserting into the population. From our preliminary testing, we applied four local searches: Forward-Backward Improvement (FBI), Two-Swap, Enhanced Left, and Insertion, to refine a constructed schedule to local optimum. And the result has revealed the FBI method gives the best refinement of a constructed schedule to local optimum in NPV maximization. Hence, the FBI method is

used here.

For FBI method, the backward pass is first applied to a feasible schedule, and the activities that are rescheduled in decreasing order of their scheduled completion times are shifted to the latest feasible position in their forward free slack. Afterward, the forward pass is applied to the feasible schedule obtained by the backward pass. The activities here, however, are rescheduled in increasing order of their scheduled start times and are shifted left to the earliest feasible position in their backward free slack. We can obtain further details of FBI method in Tormos and Lova in [40], Valls *et al.* in [41], [42], and Klein in [43].

C. The Decoding Schemes (DS)

Commonly known two different schemes are parallel list scheduling (PLS) and serial list scheduling (SLS), which transform an activity list (AL) into a feasible schedule. From our preliminary testing, we have concluded that, the forward pass predominantly performs better than the backward pass and SLS performs better in finding #best; thus, we decide to implement SLS in forward pass. The necessary steps for implementing both decoding schemes can be referred in Kolisch in [52].

IV. RESULTS AND DISCUSSIONS

The implementations were coded and compiled in Visual C++ and tested on problem instances with known optimal makespan from ProGen developed by Kolisch *et al.* in [45]. These problems consist of projects with activities (j_{10} to j_{20}) and with 2 renewable and 2 non-renewable. Also, the duration of activity j in mode m varies from 1 to 10. In each problem size, 640 instances were generated, but for some instances no feasible solutions exist; therefore, these instances will be excluded from our consideration. In all, we consider 12 different problem parameter settings for activity j_{10} to j_{20} . The result shows the proposed MA has solved exactly 72 instances of the considered problems.

The original ProGen data set, however, does not contain data on cash flows for the activities such as unit costs and arbitrary project deadlines. Thus, additional data were generated to modify the original data set. The costs per unit time for renewable and nonrenewable resources were set to be at 100 monetary units, respectively. The other costs are determined by a multiplier of 10%~20% from a total resource costs. The target project due date (T) for each problem instances was generated from a uniform distribution [$1.1T^*$, $1.4T^*$] where T^* is the best makespan. The experiment was performed at a discount rate 0.1% per period and a profit margin 10%, both were constant over the entire planning horizon.

Below we present the computational results and in each case the following numbers are shown:

- #Best – the number of instances for which the algorithm found a solution equal to the best solution known.
- Avg. Dev. – the average absolute deviation from the best solution known.
- Max. Dev. – the maximum absolute deviation from the

best solution known.

- Min. Dev. – the minimum absolute deviation from the best solution known.
- #Infeasible – the number of instances for which the algorithm is not able to find a solution.

For the sake of comparison, we implement a priority rule-based heuristic, whose framework is described as follows: we find a feasible mode from each mode selection rule and therefore establish a list of feasible modes. Then from this list, we employ a priority rule with a local search and a decoding scheme to find the corresponding schedule. We consider 22 commonly used priority rules in solving makespan problem, and 4 popular local searches and 2 effective decoding schemes (see Table I). Further detailed implementation can be found in [1], [2], [43], and [44]. As a result, each priority rule-based procedure will generate its best solution out of 96 combinations.

For the MA, the procedure terminates when the number of evaluations reaches 300. Computational results indicate that our proposed MA consistently performs better than other priority rules in terms of finding the number of best solutions in Table II. The second best method is EFT. As well, the proposed MA also consistently performs well in obtaining the lowest value of average deviation for j_{10} ~ j_{20} in Table III. But few priority rules such as ROT and EFT show good performance under certain activity numbers. Similar results appear in obtaining the lowest value of maximum deviation for j_{10} ~ j_{20} in Table IV.

Table VI shows performance ranking for the MA and 22 priority rules over j_{10} ~ j_{20} . The ranking for j_{10} ~ j_{20} was done by the three criteria: number of best solutions, average deviation, and number of infeasible solutions (provided in Table V). Then we sum up their scores (rankings) for j_{10} ~ j_{20} ; the lowest score indicates the best method among all. Again, our results indicate that the proposed MA is ranked first in the list, the EFT rule is second, and the ROT rule is third.

V. CONCLUSION

In this paper we presented a combinatorial approach for the multi-mode RCSPDCF problem. Assuming cash inflows at the beginning of each activity and cash outflows at the completion of each activity, we scheduled the activities to maximize the net present value of all cash flows. We first established the list of feasible modes via mode selection rules and then implemented a MA to find the corresponding schedule. Such method has been compared on a set of standard test problems – PSPLIB, where additional data were generated randomly with the uniform distribution.

The computational results allow distinguishing our MA from the other heuristics. Our MA consistently performs well in obtaining the highest number of best solutions found, the lowest value of average deviation and maximum deviation. In the future, we plan to continue the research on the application of the metaheuristics to different multi-mode RCSPDCF problems. In particular, it will be interesting to consider different payment model as well other financial criteria.

REFERENCES

- [1] F. F. Boctor, "heuristics for scheduling projects with resource restrictions and several resource-duration modes," *Int'l J. Prod. Res.*, vol. 31, no. 11, pp.2547-2558, 1993.
- [2] E. W. Davis and J. H. Patterson, "A comparison of heuristic and optimum solutions in resource-constrained project scheduling," *Mang. Sc.*, vol. 21, pp. 944-955, 1975.
- [3] R. Kolisch, "Efficient priority rules for the resource-constrained project scheduling problem," *J. Oper. Mang.*, vol. 12, pp. 179-192, 1996.
- [4] R. Slowinski, "Two approaches to problems of resource allocation among project activities: a comparative study," *J. Oper. Res.*, vol. 31., no. 8, pp.711-723, 1980.
- [5] B. F. Talbot, "Resource constrained project scheduling with time-resource tradeoffs: The nonpreemptive case," *Mang. Sc.*, vol. 28, pp. 1197-1210, 1982.
- [6] R. H. Doersch and J. H. Patterson, "Scheduling a project to maximize its net present value: A zero-one programming approach," *Mang. Sc.*, vol. 23, pp.882-889, 1977.
- [7] R. G. Grinold, "The payment schedule problem," *Nav. Res. Log.*, vol. 19, pp. 123-136, 1972.
- [8] J. H. Patterson, P. Slowsinki, F. B. Talbot, and J. Weglarz, "An algorithm for a general class of precedence and resource constrained scheduling problems," in *Advances in Project Scheduling*, R. Slowinski and J. Weglarz, Eds. Amsterdam: Elsevier.
- [9] J. P. Pinder and A. S. Maruchek, "Using discounted ash flow heuristics to improve project net present value," *J. Oper. Mang.*, vol. 14, pp.229-240.
- [10] R. A. Russell, "Cash flows in networks," *Mang. Sc.*, vol. 16, pp.357-373, 1970.
- [11] S. E. Elmaghraby and W. S. Herroelen, "The scheduling of activities to maximize the net present value of projects," *Eur. J. Oper. Res.*, vol. 49, pp.35-49, 1990.
- [12] W. S. Herroelen and E. Gallens, "Computational experience with an optimal procedure for the scheduling of activities to maximize the net present value," *Eur. J. Oper. Res.*, vol. 65: pp. 274-277, 1993.
- [13] C. Sepil and N. Ortac, "Performance of the heuristic procedures for constrained projects with progress payments," *J. Oper. Res. Soc.*, vol. 48, no. 11, pp.1123-1130, 1997.
- [14] D. E. Smith-Daniels and N. J. Aquilano, "Using a late start resource constrained project to improve project ne present value," *Decsion. Sc.*, vol. 17, pp.617-630, 1987.
- [15] D. E. Smith-Daniels and V. L. Smith-Daniels, "Maximizing the net present value of a project subject to material and capital constraints," *J. Oper. Mang.*, vol. 7, pp. 33-45, 1987.
- [16] M. Vanhoucke, E. Demeulemeester, and W. Herroelen, "Progress payments in project scheduling problems," *Eur. J. Oper. Res.*, vol. 148, pp.604-620, 2003.
- [17] R. Etgar, A. Shtub, and L. J. LeBlanc, "Scheduling projects to maximize net present value the case of time-dependent, contingent cash flows," *Eur. J. Oper. Res.*, vol. 96, pp. 90-96, 1997.
- [18] M. Vanhoucke, E. Demeulemeester, W. Herroelen, "On maximizing the net present value of a project under renewable resource constraints," *Mang. Sc.*, vol. 47, pp. 113-1121, 2001.
- [19] O. Icmeli and S. S. Erengue, "The resource constrained time/cost tradeoff project scheduling problem with discounted cash flow," *J. Oper. Mang.*, vol. 14, pp.255-275, 1996.
- [20] G. Ulusoy, F.Sivrik a-Şerifoğlu, S. Şahin, "Four payment models for the multi-mode resource constrained project scheduling problem with discounted cash flows," *Ann. Oper. Res.*, vol. 102, pp. 237-261, 2001.
- [21] P. Brucker, A. Mohring, K. Neumann, E. Pesch, "Resource-constrained project scheduling: Notation, classification, models and methods," *Eur. J. Oper. Res.*, vol. 112, pp.3-41, 1999.
- [22] A. Shtub, R. Etgar, "A branch-and-bound algorithm for scheduling projects to maximize net present value: The case of time dependent, contingent cash flows," *Int'l. J. Prod. Res.*, vol. 35, pp. 3367-3378, 1997.
- [23] B. De Reyck and W. Herroelen, "An optimal procedure for he resource-constrained project scheduling problem with discounted cash flows and generalized precedence relations," *Comp. Oper. Res.*, vol. 25, no. 1, pp. 1-17, 1998.
- [24] M. Vanhoucke, E. Demeulemeester, W. Herroelen, "Scheduling projects with linearly time-dependent cash flows to maximize the net present value," *Int'l. J. Prod. Res.*, vol. 39, pp. 3159-3181, 2001.
- [25] R. Etgar, A. Shtub, L. J. LeBlanc, "Scheduling projects to maximize net present value – the case of time-dependent, contingent cash flows," *Eur. J. Oper. Res.*, vol. 96, pp. 90-96, 1996.
- [26] M. Mika, G. Waligóra, and J. Weglarz, "Simulated annealing and tabu search for multi-mode resource-constrained project scheduling with positive discounted cash flows and different payment models," *Eur. J. Oper. Res.*, vol. 164, no. 3, pp. 639-668, 2005.
- [27] P. Moscato and M. G. Norman, "A 'memetic' approach for the traveling salesman problem. Implementation of computational ecology on message passing systems," in *Parallel Computing and Transputer Applications*. Amsterdam: IOS Press, 1992, pp.187-194.
- [28] H. K. Tsai, J. M. Yang and C. Y. Kao, "Solving traveling salesman problems by combining global and local search mechanisms," in *Proc. of the 2002 International Congress of Evolutionary Computation (CEC'02)*, IEEE Press, pp. 1290-1295, 2002.
- [29] N. Krasnogor and J. Smith, "A memetic algorithm with self-adaptive local search: TSP as a case study," in *GECCO-2000: Proc.s of the Genetic and Evolutionary Computation Conference*, Morgan Kaufman, pp. 987-994, 2000.
- [30] P. Merz and B. Freisleben, "Memetic algorithms for the traveling salesman problem," *Complex Systems*, vol. 14, pp. 297-345, 2001.
- [31] P. Merz and B. Freisleben, "Fitness Landscape analysis and memetic algorithms for the quadratic assignment problem," *IEEE Trans. on Evol. Comput.*, vol. 4, pp. 337-352, 2000.
- [32] A. Yin, "A new neighborhood structure for the job shop scheduling problem," in *Proc. 3rd Int. Conf. on Machine Learning and Cybernetics*, August 26-29, Shanghai, vol. 4, pp.2098-2101, 2004.
- [33] J. H. Holland, *Adaptation in Natural and Artificial Systems*, Ann Arbor, MI: Univ. Michigan Press, 1975.
- [34] K. A. De Jong, "Adaptive system design: A genetic approach," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-10, pp. 566-574, 1980.
- [35] J. Digalakis and K. Margaritis, "Performance comparison of memetic algorithms," *App. Math. Comput.*, vol.158, pp. 237-252, 2004.
- [36] J. Buddhakulsomsiri and D. S. Kim, "Priority rule-based heuristic for multi-mode resource constrained project scheduling problems with resource vacations and activity splitting," *Eur. J. Oper. Res.*, vol. 178, pp. 374-390, 2007.
- [37] F. Salewski, A. Schirmer, and A. Drexl, "Project scheduling under resource and mode identity constraints: Model, complexity, methods, and application," *Eur. J. Oper. Res.*, vol.102, pp. 88-110, 1997.
- [38] P. Moscato, On evolution, search, optimization, genetic algorithms and martial arts: toward memetic algorithms, Caltech Concurrent Computation Program, California Institute of Technology, Pasadena, Technical Report 790, 1989.
- [39] L. Davis, "Aplying adaptive algorithms to epistatic domains," in *Proc. IJCAI Conf.*, Morgan Kaufmann, 1985, pp. 162-164.
- [40] P. Tormos and A. Lova, "A competitive heuristic solution technique for resource-constrained project scheduling," *Ann. Oper. Res.*, vol. 102, pp. 65-81, 2001.
- [41] V. Valls, F. Ballestin, and S. Quintanilla, "Justification and RCPSP: A technique that pays," *Eur. J. Oper. Res.*, vol.165, pp. 375-386, 2005.
- [42] V. Valls, F. Ballestin, and S. Quintanilla, "A population-based approach to the resource constrained project scheduling problem," *Ann. Oper. Res.*, vol. 131, pp.305-324, 2004.
- [43] R. Klein, "Bidirectional planning: Improving priority rule-based heuristics for scheduling resource-constrained projects," *Eur. J. Oper. Res.*, vol. 127, pp. 619-638, 2000.
- [44] R. Kolisch, "Serial and parallel resource-constrained project scheduling methods revisited: Theory and computation," *Eur. J. Oper. Res.*, vol. 90, pp. 320-333, 1996.
- [45] R. Kolisch, A. Sprecher, and A. Drexl, "Characterization and generation of a general class of resource-constrained project scheduling problems," *Mang. Sc.*, vol. 41, pp.1693-1703, 1995.

TABLE I
COMBINATION ON A COMBINATORIAL PRIORITY RULE-BASED HEURISTIC

Priority Rules	Abbreviation
Select Jobs Randomly	RAN
Shortest Processing Time	SPT
Longest Process Time	LPT
Most Immediate Successors	MIS
Most Total Successors	MTS
Least Non-related Jobs	LNRJ
Greatest Rank Positional Weight	GRPW
Minimum Slack First	MSLK
Earliest Start Time	EST
Earliest Finish Time	EFT
Latest Start Time	LST
Latest Finish Time	LFT
Greatest Resource Demand	GRD
Greatest Cumulated Resource Demand	GCUMRD
Weighted Resource Utilization Ratio and Precedence	WRUP
Resource Scheduling Method	RSM
Activity-Time	ACTIM
Activity-Resource	ACTRES
Time-Resource	TIMRES
General Resource (Combination of Activity- and Time-Resource)	GENERES
Resource Over Time	ROT
Local Searches	Abbreviation
Forward Backward Improvement	FBI
Enhanced Left Move	Enhance-Left
Insertion	Insert
Two-Swap	2-swap
Decoding Schemes	Abbreviation
Forward Serial List Scheduling	FSLS
Forward Parallel List Scheduling	FPLS

TABLE II
PERFORMANCE COMPARISON BETWEEN THE MA AND PRIORITY RULES ON #BEST (J10~J20)

Method	<i>j10</i>	<i>j12</i>	<i>j14</i>	<i>j16</i>	<i>j18</i>	<i>j20</i>
<i>MA</i>	11	8	11	10	8	9
<i>RAN</i>	11	8	8	7	7	8
SPT	7	7	8	6	7	8
LPT	9	7	6	7	7	7
MIS	8	8	9	7	7	9
MTS	9	7	8	6	6	9
LNRJ	10	7	9	6	8	9
GRPW	8	6	7	7	7	8
GRPW*	7	7	6	7	6	7
MSLK	10	6	7	7	7	10
EST	7	8	6	8	6	8
<i>EFT</i>	9	8	11	8	8	8
LST	7	7	8	8	6	8
LFT	8	6	8	7	7	8
GRD	8	7	8	8	6	8
GCUMRD	9	7	8	6	8	8
WRUP	9	8	8	8	7	7
RSM	8	8	9	8	7	7
ACTIM	8	7	8	8	6	8
ACTRES	8	6	7	8	6	8
TIMRES	9	7	6	8	6	7
GENERES	8	6	6	8	6	7
<i>ROT</i>	11	8	8	9	7	8

TABLE III
PERFORMANCE COMPARISON BETWEEN THE MA AND PRIORITY RULES ON AVG. DEV. (J10-J20)

Method	<i>j10</i>	<i>j12</i>	<i>j14</i>	<i>j16</i>	<i>j18</i>	<i>j20</i>
<i>MA</i>	0.002	0.021	0.000	0.156	0.058	0.049
RAN	0.249	3.029	0.139	4.942	0.182	0.578
SPT	3.339	3.090	14.292	5.096	0.545	0.408
LPT	2.945	3.232	0.394	5.174	15.584	9.082
MIS	2.923	3.085	0.036	4.370	0.210	0.361
MTS	3.052	3.176	0.067	5.000	0.228	0.257
LNRJ	2.797	3.131	0.097	0.514	0.340	0.248
GRPW	3.147	3.249	15.725	0.953	0.797	0.568
GRPW*	3.601	3.122	14.394	0.795	0.283	0.508
MSLK	3.381	3.208	14.289	0.705	4.239	0.891
EST	3.162	3.061	0.144	0.563	0.607	0.726
<i>EFT</i>	2.903	3.029	0.000	0.601	0.453	0.302
LST	3.237	3.168	0.125	0.522	0.779	0.569
LFT	0.583	3.286	0.102	0.576	0.332	0.538
GRD	0.824	3.139	15.555	0.699	1.129	1.104
GCUMRD	0.355	3.065	15.464	1.182	0.495	7.652
WRUP	0.390	3.085	0.081	0.789	0.256	0.823
RSM	0.771	3.029	0.077	1.076	0.910	0.885
ACTIM	0.824	3.122	0.111	0.626	0.975	0.462
ACTRES	0.414	3.255	15.589	0.695	15.545	0.682
TIMRES	0.466	3.216	14.403	0.643	0.845	0.831
GENERES	0.491	3.212	15.665	0.607	0.738	1.635
<i>ROT</i>	0.086	3.029	0.055	0.199	0.344	0.782

TABLE IV
PERFORMANCE COMPARISON BETWEEN THE MA AND PRIORITY RULES ON MAX. DEV. (J10-J20)

Method	<i>j10</i>	<i>j12</i>	<i>j14</i>	<i>j16</i>	<i>j18</i>	<i>j20</i>
<i>MA</i>	0.020	0.190	0.000	1.870	0.630	0.560
RAN	3.240	32.870	0.780	45.000	0.850	5.300
SPT	36.000	32.870	169.990	45.000	3.220	3.390
LPT	36.000	32.870	1.990	45.000	126.940	81.600
<i>MIS</i>	36.000	32.870	0.310	45.000	1.160	2.930
MTS	36.000	32.870	0.370	45.000	1.410	1.640
LNRJ	36.000	32.870	0.860	2.390	3.570	1.870
GRPW	36.000	32.870	169.990	7.250	4.670	6.250
GRPW*	36.000	32.870	169.990	4.420	1.850	4.890
MSLK	36.000	32.870	169.990	4.370	41.300	10.690
EST	36.000	32.870	0.760	4.420	3.570	5.560
<i>EFT</i>	36.000	32.870	0.000	4.420	2.640	2.770
LST	36.000	32.870	1.000	3.870	5.930	5.560
LFT	5.340	32.870	1.000	3.270	1.550	3.810
GRD	7.950	32.870	169.990	4.420	9.060	6.940
GCUMRD	3.240	32.870	169.990	8.530	4.670	81.600
WRUP	3.690	32.870	0.780	7.340	1.450	5.530
RSM	7.950	32.870	0.850	9.520	3.600	8.740
ACTIM	8.400	32.870	0.830	4.420	5.970	4.270
ACTRES	2.620	32.870	169.990	4.370	126.940	6.010
TIMRES	2.620	32.870	169.990	3.820	4.110	7.680
GENERES	3.690	32.870	169.990	3.830	5.020	9.820
<i>ROT</i>	1.120	32.870	0.500	2.190	2.160	6.050

TABLE V
PERFORMANCE COMPARISON BETWEEN THE MA AND PRIORITY RULES ON #INFEASIBLE. (J10-J20)

Method	j10	j12	j14	j16	j18	j20
MA	0	2	1	1	2	1
RAN	0	2	2	3	2	1
SPT	0	2	1	2	2	1
LPT	0	2	3	2	2	1
MIS	0	2	1	2	2	1
MTS	0	2	1	2	2	1
LNRJ	0	2	1	2	2	1
GRPW	0	2	2	2	2	1
GRPW*	0	2	1	2	2	1
MSLK	0	2	1	2	2	1
EST	0	2	3	2	2	1
EFT	0	2	1	2	2	1
LST	0	2	1	2	2	1
LFT	0	2	2	2	2	1
GRD	0	2	2	2	3	1
GCUMRD	0	2	2	2	2	1
WRUP	0	2	2	2	2	1
RSM	0	2	2	2	2	1
ACTIM	0	2	1	2	2	1
ACTRES	0	2	2	2	2	1
TIMRES	0	2	1	2	2	1
GENERES	0	2	2	2	2	1
ROT	0	2	2	2	2	1

Table VI
PERFORMANCE RANKING OF THE MEMETIC ALGORITHM AND PRIORITY RULES
BASED ON THREE CRITERIA: AVG. DEV., NO. BEST, AND NO. INFEASIBLE

Method	Activity j						Total
	j10	j12	j14	j16	j18	j20	
MA	1	1	1	1	1	2	7
EFT	9	3	2	5	3	6	28
ROT	2	5	6	2	9	15	39
LNRJ	4	13	5	20	2	3	47
RAND	3	2	12	18	5	12	52
MIS	18	7	3	17	6	5	56
WRUP	7	8	8	11	7	19	60
MSLK	5	16	16	14	13	1	65
RSM	14	4	4	12	12	21	67
GCUMRD	6	9	14	21	4	17	71
ACTIM	16	12	10	7	21	8	74
MTS	11	16	7	22	15	4	75
LFT	13	23	9	13	8	9	75
EST	20	6	19	4	17	14	80
LST	21	15	11	3	19	11	80
GRPW	19	8	18	16	11	10	82
SPT	22	10	13	23	10	7	85
GRD	15	14	15	10	22	16	92
ACTRES	11	22	17	9	23	13	95
TIMRES	8	17	22	8	20	20	95
LPT	10	14	20	19	14	23	100
GENERES	12	20	23	6	18	22	101
GRPW*	23	11	21	15	16	18	104