Solving a multi-objective multi-mode resource-constrained project scheduling problem with discounted cash flows

F.S. Kazemi
M.Sc in Department of Industrial Engineering, College of Engineering, University of Tehran, Tehran, IRAN
f.s.kazemi@ut.ac.ir

R. Tavakkoli-Moghaddam
Professor in Department of Industrial Engineering College of Engineering, University of Tehran, Tehran, IRAN
tavakoli@ut.ac.ir

Keywords:
Multi-objective project scheduling, Resource-constraint, Discounted cash, NSGA-II.

Abstract
This paper presents a multi-objective multi-mode resource-constrained project scheduling problem (MRCPSP) with positive and negative cash flows. We consider the objectives of the net present value (NPV) maximization and robustness maximization along by makespan minimization. Furthermore, to make a model close to the real world situations, two types of commonly used payment models are considered. We present a mathematical model for the given problem and solve it by the Lingo 8 software. In addition, we use the nondominated sorting genetic algorithm II (NSGA-II) in order to solve the proposed model. Finally, the computational results for a set of test problems taken from the project scheduling problem library (PSPLIB) are presented and discussed.

1. Introduction
The resource-constrained project scheduling problem (RCPSP) is to schedule project activities in order to complete a project in the minimum possible time under the presence of precedence and resource constraints. Furthermore, precedence constraints are defined between activities (i.e., no activity can be started before finishing all its predecessors). In certain situations, it is possible to execute each activity in one of several alternative modes that usually represent relations between the used and consumed resources, and the duration of the activity. In such a case, the resulting problem is called the multi-mode resource-constrained project scheduling problem (MMRCPSP). The objective is to find an assignment of modes to activities, as well as precedence and resource-feasible starting times.
of all activities such that the project duration (i.e., makespan) is minimized. The problem is known to be strongly NP-hard. For a comprehensive survey on project scheduling, see Brucker et al [1].

In general, the RCPSP is an important and challenging problem that has been widely studied over the past few decades. For example, Brucker et al. [1] provided an extensive review of the RCPSP literature, quoting more than 200 relevant papers. They discussed both exact and heuristic approaches, and presented a new notation and classification scheme, similar to the one traditionally used in the machine scheduling literature. Herroelen et al. [2] presented an exhaustive survey of exact algorithms (e.g., branch-and-bound procedures) for several variants of the RCPSP. Most of these heuristics are extensively evaluated in Hartmann and Kolisch [3]. Herroelen et al. [4] presented a survey paper exclusively focusing on project scheduling problems with discounted cash flows. Moreover, the reader can refer to the comprehensive monograph of Kolisch [5]. Viana and Pinho de Sousa [6] applied multi-objective versions of simulated annealing (SA) and tabu search (TS) to the RCPSP in order to minimize the makespan, the "weighted" lateness of activities and the violation of resource constraints.

Al-Fawzan and Haouari [7] addressed the issue of designing a project schedule that is not only short in time, but also less vulnerable to disruptions due to reworks and other undesirable conditions. To that aim, they introduced the concept of schedule robustness and developed a bi-objective resource-constrained project scheduling model. They considered the objectives of robustness maximization along with the makespan minimization. Heilmann [8] investigated a branch-and-bound procedure for the MMRCPS with minimum and maximum time lags. Shadrokh and Kianfar [9], presented a genetic algorithm (GA) for solving a class of project scheduling problems, called resource investment problem. In their research, tardiness of a project is permitted with the defined penalty. Rabbani et al [10] investigated a newly developed method for the RCPSP in stochastic networks by merging the new and traditional resource management methods. The objective of their presented model was defined as minimizing the multiplication of the expected project duration and its variance. Kobylanski and Kuchta [11], demonstrated a note on the paper by Al-Fawzan and Haouari [7] presented two new criteria for a bi-objective problem for robust resource-constrained project scheduling. Mendes et al [12] presented a GA for the RCPSP. The chromosome representation of the problem was based on random keys. The schedule was constructed using a heuristic priority rule, in which the priorities of the activities were given by the GA.

Pinto et al [13] introduced a heuristic that is based on a unique GA to solve the resource-sharing and scheduling problem (RSSP). This problem considers the use of a set of resources for the production of several products, and selects a single mode for each operation and accordingly to schedule the resources, while minimizing the makespan time. Abbasi et al [14] addressed bi-objective resource-constrained project scheduling that the first objective is makespan to be minimized, and the second one, a recently developed measure, is robustness maximization aimed at the floating time maximization to make scheduling more reliable. Mika et al [15] presented a TS method for the RCPSP with schedule-dependent setup times. Alvarez-Valdes et al [16] developed some preprocessing techniques and several heuristic algorithms based on GRASP and path relinking for project scheduling under the partially renewable resource. Damak et al [17] considered the RCPSP with multiple execution modes for each activity and minimization of the makespan. To solve this problem, they proposed a differential evolution (DE) algorithm. Goncalves et al [18] presented a GA for the resource constrained multi-project scheduling problem. The chromosome representation of the problem was based on random keys. The schedules are constructed using a heuristic that builds parameterized active schedules based on priorities, delay times, and release dates defined by the GA. Valls et al [19] proposed a hybrid genetic algorithm (HGA) for the RCPSP. This algorithm introduces several changes in the GA paradigm: a crossover operator specific for the RCPSP; a local improvement operator that was applied to all generated schedules; a new way to select the parents to be combined; and a two-phase strategy by which the second phase re-starts the evolution from a neighbour’s population of the best schedule found in the first phase.

Hence, the proposed model has not been solved by any researcher. In this paper to make the model close to the real world situation, we address a multi-objective multi-mode resource-constrained project scheduling problem with the discounted cash flow (MMRCPSPDCF), positive and negative cash flows. We also consider activities require resources and all resources are assumed to be renewable,
which are available in limited amounts. Objective functions in our problem are the net present value (NPV) maximization and robustness maximization along by the makespan minimization. We solve the given problem under the four different payment models using the Lingo 8 software and NSGA-II.

In the following, we first describe our problem and its mathematical model in Section 2. In Section 3, we explain our proposed and implemented NSGA-II. In Section 4, we show the experimental results obtained by the Lingo 8 software and NSGA-II. Finally, the last section concludes this study.

2. Proposed Model
2.1. Objective function

Objective functions in our presented problem are the NPV maximization and robustness maximization along by the makespan minimization. In the real world, there are two parties involved in the practical project, namely client and contractor. In addition, there are two types of loops: 1) the inner loop (i.e., the contractor) corresponds to search for the schedule of activities maximizing his/her NPV and 2) the outer loop (i.e., the client) regarding the distribution of the budget over the event nodes. In this paper, the NPV is influenced by four types of payment scheduling models. The objective is to maximize the NPV for two types of payments given by the following relationships.

1. Lump-Sum Payment (LSP): In this model, when the project is terminated successfully, the whole payment is paid by the client to the contractor. This method is one of the most common applied payment structures. The NPV is computed by:

\[
NPV = CF_{\text{LSP}} (1 + \alpha)^{-FT_j} - \sum_{j=1}^{n} \sum_{t=1}^{LF_j} \sum_{m=1}^{M} CF_{jm}^{-} \times x_{jmt} \quad (1)
\]

\[
CF_{\text{LSP}} = \sum_{j=1}^{n} CF_{j}^{+} \quad (2)
\]

2. Payments at Event Occurrence (PEO): In this case, payments are paid at the occurrence of events. This model is very practical and frequently used. The related NPV is computed by:

\[
NPV = \sum_{j=1}^{n} CF_{j}^{+} \times (1 + \alpha)^{-FT_j} - \sum_{j=1}^{n} \sum_{t=1}^{LF_j} \sum_{m=1}^{M} CF_{jm}^{-} \times (1 + \alpha)^{-t} \times x_{jmt} \quad (3)
\]

Where, \( FT_j \) is completion time of activity \( j \).

Other objective can be regarded as the makespan minimization. The makespan is the completion time of a project that equals the completion time of activity \( n \). This objective is computed by:

\[
C_{\text{max}} = \max \left( \sum_{m=1}^{M} \sum_{t=1}^{LF_j} t \times x_{jmt} \right) \quad (4)
\]

Al-Fawzan and Haourai [7] developed a robustness concept in their study and used a TS method to find good solutions. Robustness of the scheduling means that if some activities take more time than estimated in the scheduling phase, the project finish date will not change without any cost. This concept relates to the project floating time. The value of \( \lambda \) in Equation (5) presents the importance of the makespan versus the floating time. In this paper, we consider this criterion as an objective computed bellow.

\[
Robustness = \lambda \times C_{\text{max}} - (1 - \lambda) \times TF \quad (5)
\]

\[
TF = \sum_{j=1}^{n} (LF_j - EF_j) \quad (6)
\]

Where, \( \lambda \) is one value in interval \((0, 1]\).
2.2. Mathematical model

In this section, we present a mathematical model for the multi-objective multi-mode resource-constrained project scheduling problem with the discounted cash flow (MMRCPSPPDCF).

Indexes and parameters:
- **J**: Index of activity
- **m**: Index of mode
- **k**: Index of resource
- **t**: Index for period of time
- **G**: Directed activity on node graph
- **V**: Set of activities
- **M_j**: Number of execution modes of activity **j**
- **n**: Number of activities
- **P_j**: Set of direct predecessors of activity **j**
- **S_j**: Set of direct successor of activity **j**
- **R**: Number of type's renewable resource
- **d_{jm}**: Duration of activity **j** executed in mode **m**
- **R_k**: Availability of each resource type **k** in each time period
- **r_{jm}**: Request for resource **k** by activity **j** executed in mode **m**
- **CF_j^+**: Positive cash flows for activity **j**
- **CF_j^-**: Negative cash flows for completion activity **j** in mode **m**
- **\alpha**: Discounted rate
- **T**: Project time window
- **\lambda**: A real value in (0, 1) range

Variables:
- **ES_j**: Earliest start of activity **j**
- **LS_j**: Latest start of activity **j**
- **EF_j**: Earliest finish of activity **j**
- **LF_j**: Latest finish of activity **j**
- **TF_j**: Float of activity **j**
- **TF**: Total float.
- **C_j**: Completion time of activity **j**
- **C_{max}**: Maximum completion time
- **NPV**: Net present value
- **x_{jmt}**: \begin{cases} 
1 & \text{if activity } j \text{ is completed in mode } m \text{ in period } t \\
0 & \text{otherwise} 
\end{cases}

Proposed mathematical model:
- \text{max } NPV \tag{7}
- \text{min } (\lambda \times C_{max} - (1 - \lambda) \times TF) \tag{8}

s.t.
- **ES_j** = 0 \tag{9}
- **EF_j = ES_j + d_{jm} ; m = 1, 2, \ldots, M_j , i = 1, 2, \ldots, n** \tag{10}
- **ES_j = \max \{EF_i\} ; \forall i \in P_j , j = 1, 2, \ldots, n** \tag{11}
- **LF_j = T** \tag{12}
- **LS_j = LF_j - d_{jm} ; j = 1, 2, \ldots, n** \tag{13}
In the above model, the objective functions in (7) and (8) calculate two defined objectives. Constraints (9) to (14) calculate the earliest and latest finish time for all activities. Constraint (15) establishes the predecessor relations. Constraint (16) states if activity \( j \) started in mode \( m \in 1,2,...,M_j \) in time \( t \), then it must be completed in the same mode. Constraint (17) calculates the completion time of activity \( j \). Constraint (18) calculates the maximum completion time of the project. Constraint (19) calculates the time window of the project, and Constraint (20) says \( T \) is the upper bound of \( C_{max} \). Constraint (21) ensures that the renewable resource constraint is met. Constraint (22) calculates the floating time for activity \( j \). Constraint (23) calculates the total float. Finally, Constraint (24) defines the domain of the decision variables.

3. Non-dominated sorting genetic algorithm (NSGA-II)

Deb et al [20] proposed a multi-objective evolutionary algorithm (MOEA), namely non-dominated sorting genetic algorithm II (NSGA-II), which an enhanced version of the NSGA. They addressed all of these issues and proposed an improved version of the NSGA. From the simulation results on a number of difficult test problems, they found that the NSGA-II outperforms two other contemporary MOEAs: namely Pareto-archived evolution strategy (PAES) and strength- Pareto EA (SPEA) in terms of finding a diverse set of solutions and in converging near the true Pareto-optimal set.

3.1. Proposed NSGA-II

In this paper, we design the NSGA-II to solve the proposed model. We use the peak crossover [19] and local based mutation in our design. Parents are selected by using the 2-tournament selection procedure. In this selection, two solutions are chosen randomly, and then the candidate solution with the best fitness function value is selected. Afterwards, we determine randomly whether \( i \) or \( j \) represents one parent as the father. The other parent represents the mother.
3.2. Solution representation and initialization

One important decision in designing a meta-heuristic method is to decide how to represent and relate solutions in an efficient way to the searching space. Representation should be easy to decode to reduce the cost of the algorithm. The first task is to represent a solution of a problem in solving the MRCPSPDCF. The chromosome is represented by two vectors, the first is a precedence-feasible permutation of activities, in which each activity \( j (j = 1, \ldots , n) \) has to occur after all its predecessors and before all its successors. The second is a list of execution modes for all activities, the \( j \)-th of this vector defines the execution mode of activity in \( j \)-th cell of the first vector.

In this paper, to generate \( N \) trial solutions in the initial population, serial and two-way serial schedule generation schemes have been used. Each scheme generates \( N \) trial solutions. We put these solutions in a pool and select \( N \) solutions among generated \( 2N \) solutions by non-dominated sorting and crowded comparison operator [20].

4. Experimental Results

4.1. Mathematical model

We run the mathematical model for four small-sized problems by the Lingo 8 software. Number of activities in these problems are 12, in which Activities 1 and 12 are defined as dummy activities with duration zero. The required data of these problems are shown in Tables 1 to 3.

<table>
<thead>
<tr>
<th>Prob.</th>
<th>( \alpha ) (discounted rate)</th>
<th>TH</th>
<th>( R_k )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.01</td>
<td>22</td>
<td>16</td>
</tr>
<tr>
<td>2</td>
<td>0.05</td>
<td>16</td>
<td>24</td>
</tr>
<tr>
<td>3</td>
<td>0.01</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>4</td>
<td>0.05</td>
<td>20</td>
<td>16</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Activity</th>
<th>d_j</th>
<th>r_k</th>
<th>successor</th>
<th>d_j</th>
<th>r_k</th>
<th>successor</th>
<th>d_j</th>
<th>r_k</th>
<th>successor</th>
<th>d_j</th>
<th>r_k</th>
<th>successor</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>3,2,4</td>
<td>0</td>
<td>0</td>
<td>3,2,4</td>
<td>0</td>
<td>0</td>
<td>3,2,4</td>
<td>0</td>
<td>0</td>
<td>3,2,4</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>6</td>
<td>5,6,7</td>
<td>3</td>
<td>9</td>
<td>10</td>
<td>4</td>
<td>6</td>
<td>6,8,11</td>
<td>2</td>
<td>5</td>
<td>5,6</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>9</td>
<td>6,7,8</td>
<td>5</td>
<td>8</td>
<td>5,7,8</td>
<td>2</td>
<td>10</td>
<td>7,9</td>
<td>3</td>
<td>6</td>
<td>6,7,8</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>2</td>
<td>8,9</td>
<td>5</td>
<td>7</td>
<td>6,7,8</td>
<td>7</td>
<td>10</td>
<td>5,6,7</td>
<td>3</td>
<td>8</td>
<td>9,10,11</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>7</td>
<td>8,9</td>
<td>2</td>
<td>7</td>
<td>6,9,11</td>
<td>1</td>
<td>4</td>
<td>8,9,11</td>
<td>3</td>
<td>9</td>
<td>7,8</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>3</td>
<td>10,11</td>
<td>2</td>
<td>8</td>
<td>10</td>
<td>2</td>
<td>6</td>
<td>9,10</td>
<td>2</td>
<td>9</td>
<td>10,11</td>
</tr>
<tr>
<td>7</td>
<td>9</td>
<td>6</td>
<td>10,11</td>
<td>1</td>
<td>5</td>
<td>9,11</td>
<td>1</td>
<td>7</td>
<td>10,11</td>
<td>1</td>
<td>3</td>
<td>9,10,11</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>4</td>
<td>10,11</td>
<td>3</td>
<td>8</td>
<td>9,10,11</td>
<td>1</td>
<td>5</td>
<td>10</td>
<td>6</td>
<td>5</td>
<td>9</td>
</tr>
<tr>
<td>9</td>
<td>4</td>
<td>6</td>
<td>12</td>
<td>1</td>
<td>7</td>
<td>12</td>
<td>1</td>
<td>6</td>
<td>12</td>
<td>3</td>
<td>5</td>
<td>12</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>7</td>
<td>12</td>
<td>2</td>
<td>4</td>
<td>12</td>
<td>3</td>
<td>10</td>
<td>12</td>
<td>1</td>
<td>6</td>
<td>12</td>
</tr>
<tr>
<td>11</td>
<td>3</td>
<td>8</td>
<td>12</td>
<td>2</td>
<td>9</td>
<td>12</td>
<td>6</td>
<td>5</td>
<td>12</td>
<td>3</td>
<td>8</td>
<td>12</td>
</tr>
<tr>
<td>12</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
Table 3. Negative and positive cash flows

<table>
<thead>
<tr>
<th>Activity</th>
<th>Cash flow</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>CF&lt;sub&gt;j&lt;/sub&gt;</td>
<td>0</td>
<td>2125</td>
<td>1808</td>
<td>1721</td>
<td>1909</td>
<td>1200</td>
<td>1931</td>
<td>1810</td>
<td>1864</td>
<td>2011</td>
<td>2077</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>CF&lt;sub&gt;j&lt;/sub&gt;</td>
<td>0</td>
<td>980</td>
<td>726</td>
<td>657</td>
<td>807</td>
<td>240</td>
<td>825</td>
<td>728</td>
<td>771</td>
<td>889</td>
<td>941</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

The main assumptions are:
- Number of renewable resources and execution modes are set at 1.
- The range of \( \lambda \) is between 0.95 to 1.

Test problems have been run for four types of payment scheduling models with positive cash flows in this study. The required data for these models is reported as follows. In the LSP case, the whole payments are paid the completion time of project \( (FT_n) \). Valued as 17, 11, 14 and 14 for any four test problems, respectively. The computational results for different types of payment models types are reported in Table 4.

Table 4. Computational results obtained by Lingo 8

<table>
<thead>
<tr>
<th>Prob.</th>
<th>( \lambda )</th>
<th>( f(C_{max},FT) )</th>
<th>( NPV_{LSP} )</th>
<th>( NPV_{PEO} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.95</td>
<td>14.95</td>
<td>8855.261</td>
<td>9896.814</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>17</td>
<td>8855.261</td>
<td>9896.814</td>
</tr>
<tr>
<td>2</td>
<td>0.95</td>
<td>9.7</td>
<td>5785.215</td>
<td>7688.164</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>11</td>
<td>5785.215</td>
<td>7688.164</td>
</tr>
<tr>
<td>3</td>
<td>0.95</td>
<td>12.5</td>
<td>9186.605</td>
<td>10001.48</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>14</td>
<td>9186.605</td>
<td>10001.48</td>
</tr>
<tr>
<td>4</td>
<td>0.95</td>
<td>12.6</td>
<td>4388.025</td>
<td>7385.925</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>14</td>
<td>4388.025</td>
<td>7385.925</td>
</tr>
</tbody>
</table>

4.2. Validation of the mathematical model and NSGA-II

The validation of the proposed mathematical model is proven by solving one of small-sized problems given in Section 4.1. Therefore, Problem 1 with \( \lambda = 0.95 \) is selected. The optimal solution for each objective function is generated by a complete enumeration method. Then, the problem the proposed mathematical model is solved by the Lingo 8 software. The optimal results of the mathematical model are 8855.261 for maximizing the NPV with the LSP, and 14.95 for minimizing the \( f(C_{max},FT) \). The optimal solutions found by each of which methods are exactly the same. Hence, the constraints given in the mathematical model work accurately. This problem has been solved by the NSGA-II. Since, the results of test problems with the NSGA-II and the mathematical model are exactly similar; our proposed and implemented algorithm is convergence.

4.3. Experimental results for NSGA-II

For test instances, we use a set of problems from the project scheduling problem library [21]. The project in this set consists of 10, 12, 14, 16, 18 and 20 activities. The negative cash flows are generated from the uniform distribution of U[1, 1000]. Then, these values are sorted based on the processing time of the activity in each mode. The larger value is assigned to the mode with a smaller processing time. To generate positive cash flows, we use Eq. 25.

\[
CF_j^+ = CF_{jm} \times RR
\] (25)
Where, RR is the income rate. There are four data sets for each test problem generated from the combination of the RR and discount rate. Values of the RR are 1.25 for the first two problems and 1.5 for two other problems. The problems uses discounted rates (i.e., $d$) of 0.01, 0.05, 0.01 and 0.05. The main assumption of the NSGA-II are as follows:

- Population size (i.e., $N$) is set at 100.
- The algorithm is terminated after 200 iterations.
- Local iteration is fixed to 10.
- Crossover and mutation rates are fixed to 0.8 and 0.2, respectively.
- The value of $\lambda$ is selected from set \{0.97, 0.99\}

The results of the algorithm are shown in Table 5. In this table, not only the makespan and NPV are appropriate, but also are robust.

<table>
<thead>
<tr>
<th>Prob. set</th>
<th>Data set</th>
<th>Number of non-dominate</th>
<th>$F(C_{\text{max}}, TF)$</th>
<th>$F(C_{\text{max}}, TF)$</th>
<th>$NPV_{\text{LSP}}$</th>
<th>$NPV_{\text{PEO}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$J_{10}$</td>
<td>1</td>
<td>28</td>
<td>118.9</td>
<td>203</td>
<td>4290.7</td>
<td>3931.8</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>32</td>
<td>148.7</td>
<td>150.5</td>
<td>1832.3</td>
<td>2849.3</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>21</td>
<td>103</td>
<td>110</td>
<td>3602.6</td>
<td>2530.1</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>59</td>
<td>112.4</td>
<td>117.2</td>
<td>2990.4</td>
<td>2816.4</td>
</tr>
<tr>
<td>$J_{12}$</td>
<td>1</td>
<td>93</td>
<td>209.1</td>
<td>214</td>
<td>1751.1</td>
<td>2214.5</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>67</td>
<td>156</td>
<td>161.3</td>
<td>1857.1</td>
<td>2319.6</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>33</td>
<td>149</td>
<td>152</td>
<td>1951.9</td>
<td>3763.6</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>19</td>
<td>162</td>
<td>164.9</td>
<td>2269.3</td>
<td>2504.7</td>
</tr>
<tr>
<td>$J_{14}$</td>
<td>1</td>
<td>71</td>
<td>213.02</td>
<td>216.21</td>
<td>1804.7</td>
<td>2923.1</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>25</td>
<td>129</td>
<td>133.56</td>
<td>2790.7</td>
<td>3522.1</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>38</td>
<td>227.91</td>
<td>230</td>
<td>2934.5</td>
<td>2823.6</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>38</td>
<td>257</td>
<td>261</td>
<td>4262.4</td>
<td>2936.6</td>
</tr>
<tr>
<td>$J_{16}$</td>
<td>1</td>
<td>96</td>
<td>275</td>
<td>277.8</td>
<td>3593.8</td>
<td>2951.9</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>57</td>
<td>395.1</td>
<td>399.4</td>
<td>4179.7</td>
<td>3001.5</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>83</td>
<td>300</td>
<td>302.09</td>
<td>2239.9</td>
<td>2781.8</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>54</td>
<td>299.5</td>
<td>304</td>
<td>2007.5</td>
<td>4268.1</td>
</tr>
<tr>
<td>$J_{18}$</td>
<td>1</td>
<td>51</td>
<td>294.3</td>
<td>304</td>
<td>2163.1</td>
<td>3245.7</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>13</td>
<td>275.2</td>
<td>278.7</td>
<td>2255.5</td>
<td>4995.8</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>68</td>
<td>334</td>
<td>339</td>
<td>2571.6</td>
<td>3293.5</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>83</td>
<td>260.9</td>
<td>264.3</td>
<td>3907.8</td>
<td>4622.7</td>
</tr>
<tr>
<td>$J_{20}$</td>
<td>1</td>
<td>93</td>
<td>345.6</td>
<td>348.91</td>
<td>3867.3</td>
<td>2905.3</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>59</td>
<td>401</td>
<td>405</td>
<td>4137.4</td>
<td>4325.2</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>76</td>
<td>379</td>
<td>382</td>
<td>4261.5</td>
<td>3550.8</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>58</td>
<td>298.4</td>
<td>310.7</td>
<td>4856.7</td>
<td>4768.2</td>
</tr>
</tbody>
</table>

5. Conclusions

In this paper, we have presented a mathematical model for the multi objective multi-mode resource-constrained project scheduling problem (RCPSP) with positive and negative cash flows (MMRCPSPDCF). In addition, we have considered two types of payment models with positive cash flows. Four different test problems have been solved with the Lingo 8 software. Further, we have
presented a well-known multi-objective evolutionary algorithm (MOEA), namely non-dominated sorting genetic algorithm II (NSGA-II), with the peak crossover and the local search based mutation to solve the MMRCPSPDCF with three criteria, namely makespan, robustness and NPV. A data set taken from the project scheduling problem library are solved by the NSGA-II.

References


