Case Study

Project Scheduling problem using metaheuristic algorithm

Amir Seyedan A
Department of industrial management,
Firoozkooh branch,
Islamic Azad University, Firoozkooh, Iran

ABSTRACT
In this paper we solve Time-Cost trade-off project, in a way that sees decision maker preferences and cares about resource restriction simultaneously. Cause of being a part of NP-hard questions we use a meta-initiative multi index genetic algorithm method for model making and solving it. This paper estimates completing project time watching both time and cost elements, also initiative SJS method is being used for resource attribution. In this order with caring aforementioned purposes we calculate project completing time, and decision maker could run to ideal answers by changing time and cost indexes importance. This question was coded in Matlab software and cause of being complicated it reaches to an ideal answer in acceptable time.

Key words: Multi-attribute decision making, time-cost trade-off, resource constraint, multi-attribute genetic algorithm, critical path method.

INTRODUCTION
For determining project scheduling program one of the common methods is CPM (Critical Path Method). Changing project time is impossible in this method. On the other hand, determining project time is based on resources boundless assumption and vice versa to the real world. Many jobs have been done to eliminate mentioned defects; the most considerable one is time-cost trade-off method initiation and resources attribution. Project administration resources including manpower, machinery, etc in real construction world are limited. For counting resources restriction construction scheduling has to have contained resources attribution. Lots of math and probing methods have been employed to solve time-cost trade-off question and attribute resources. Generally, all probing methods are dependent on question and do not guarantee optimum answer [1]. On the other hand, by question dimensions extension and complication math methods lose their performance, although they guarantee optimum answer [2]. By employing math method, changing probing methods principles to restrictions and purpose function is necessary and has some lapses possibly [3].

Subject Discuss
Numbered common probing models for solving TCT question are Fondahl [4], Parger structural method [5], Moslehi stiffness structural model [6] and Siemens model [7]. Having math methods more punctual than probing ones, they gained more attention. For instance we mention critical path planning [8], integer planning model [9] and dynamic programming [10] and also integer and linear planning hybrid model [11] to you.In Resource Allocation questions, linear and dynamic planning models are some employed mathematic methods [12]. For avoiding complex optimizing question, intuitive principles are used to solve these kinds of questions. Due to above-mentioned items and both crucial time and cost indexes, need to use Carter algorithm is clear. GA algorithm, due to its natural characteristic to search randomly based on purpose and in possible space, it has the ability to make multi-criterion question model in this field, has been in sight. For the first time in 1997 some researchers named Li and Love used GA in solving TCT (time-cost trade-off) [14]. Feng did the same also [15]. Other examples could be GA application in
construction time with or without resources limitation [16] and machine learning and GA application in time-cost trade-off question solving [17]. But still in all models optimizing has been kept as a single-criterion. So, need to use multi-criterion algorithm was totally clear. Some multi-criterion genetic algorithm applications are multi-purpose GA application in time-cost optimizing [3] and multi-criterion optimizing model for construction scheduling by using GA [2]. But still in all mentioned models decision maker preferences in using the best item has not been counted.

**Multi-criterion genetic algorithm model for TCT question solving supposing resources limitation**

Genetic algorithm simulates living organisms' natural genetic process. Noticeable GA power and its ability in one item random using are like a device to lead searching up in the unknown air to the better answer. This searching has an exact leading for parents searching by using exchange and mutation footsteps resulting children be better than fathers [18].

**GA Model Properties**

A: Time-cost relation

For making GA model more like real world, time and cost relation has been counted separately.

B: Chromosome Appearance

Due to resource limitation, doing activities and their way of doing ordering are decision variants, so we face to two chromosome appearances. In this way, chromosome is in two parts, first one the activities doing ordering, and second one information round their method.

<table>
<thead>
<tr>
<th>( A_1 )</th>
<th>( A_2 )</th>
<th>( \cdots )</th>
<th>( A_{n-1} )</th>
<th>( A_n )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( m_1 )</td>
<td>( m_2 )</td>
<td>( \cdots )</td>
<td>( m_{n-1} )</td>
<td>( m_n )</td>
</tr>
</tbody>
</table>

n: activities number \( A_i \): activity name

\( m_i \): activity implementation method number

This kind of answer could be known as mutation chromosome. In every block of this chromosome there's a natural number. Naturally this chromosome has \( 2n \) blocks that \( n \) blocks are in right and \( n \) blocks are in left. Left blocks chromosome number show activities doing priority ordering, that means the activity in block 1 has the highest priority and activity in block \( n \) has the least priority. Right side chromosome blocks number also respectively show activities \( A_1 \) to \( A_n \), and each activity \( m_i \) shows that activity method which costs money, time and specific resources. With this foundation for a set of \( m \) are defined for each \( A_i \) activity and \( m_i \) is one of them.

C: Resources Allocation

In this paper SGS method, an initiative method has been used to estimate project time regardless of resources limitation. SGS method uses a half complete scheduling program to build a practical scheduling program based on resources limitation criterion. Half complete scheduling program is a kind of program which is created by random numbers producer algorithm in an incipient manner and the total project time is unknown but shows the activities priority due to each other. SGS method is split to two methods itself. They are named serial SGS and parallel SGS. Next, we'll explain both[23].In most scientific studies to solve resources attribution question which have used meta-initiative methods, serial SGS has been used for project time calculating [19, 20]. Some others have used parallel SGS [21]. The reason serial SGS has been used in most scientific researches is that in parallel method not all optimizing answers fit in question room; but in serial method they all do [20]. Many experimental researches have shown that serial method have better results than parallel method at the time many scheduling are done for a project [22]. Possibly in low algorithm repetition parallel method gives us better answers than serial method. Totally we have no idea for sure which method is better, serial or parallel. So, in this paper this matter has been seen from a different perspective. This different angle of sight is using both serial and parallel, because that's impossible to foresee which method is better.

**Fitness Function**

In single-index optimizing there's an explanation of solving method, but not in multi-indexes.
First DM (Decision Making) puts its preferences due to time and cost indexes, then indexes weight due to DM preferences are calculated by entropy method. Suppose decision maker preferences matrix (DM) on indexes is like this:

\[ \text{Preference Matrix} = \begin{bmatrix} T_{ij} & C_{ij} \end{bmatrix} \]

Then, supposing DM preferences compatible with function indexes, weight function is like this, using entropy method:

\[ \text{fitness}(x) = \frac{w_i T(x) - T_{\text{min}} + \gamma + w_c C(x) - C_{\text{min}} + \gamma}{w_i + w_c} \]

Which variants in aforementioned relations are:

- \( T_{\text{max}}, T_{\text{min}} \) are time criterion maximum.
- \( C_{\text{max}}, C_{\text{min}} \) are cost criterion maximum.

\( C(x) \) and \( T(x) \) are time and cost criteria value \( X \) order item. Also parameter is a very small random number which is for blocking zero to zero division. DM decision maker could reach a group of ideal answers by changing its preferences for time and cost indexes.

**Selection Operator**

This operator randomly chooses two chromosomes from present group as parents till breeding happens like them. Employed method in this paper is so-called genetic algorithm method, roulette-wheel selection.

**Crossover Operator**

Chromosome structure has two parts in this paper; cause of having two parts, Crossover operator chromosome model used in this research has two different parts, too. In this way we explore operator which operates on first part:

![Crossover point](image)

Figure 1. Crossover operator on first part

In figure 1 example as it shown two parent chromosome are chosen and ready for crossover operation. From this operation we have two children born. By making crossover point specific, first child is born which in there father genes are before intersection point. That is exactly copied in child chromosome. Now for making genes from child chromosome which is in the right we must use mother genes. But mother right chromosome could not be copied in child, because some action will be repeated two times in gene or may some prerequisite relations lose their order. To overcome this problem we use mother chromosomes order to fill child right side chromosomes. This method in chart 1 is shown also, first genes 1, 2 and 5 from father are copied into child chromosome and then to fill the right side genes from child chromosome we use activities that are not used in child chromosome, which those are in mother
Project Scheduling problem using metaheuristic algorithm

chromosome order shown in example as 4, 6, and 3. For second child, this process is done vice versa. This means the left side genes are exactly copied from mother chromosome and the right side genes are taken respectively from father chromosome.

**Figure 2.** Crossover operator on chromosome second part.
In chart 2 crossover operator which operates on parent chromosome second part is shown. This operator is a two-point crossover operator which is not different from other two-point operators. To make new children, genes from both sides of crossover point are kept fixed and genes between chromosome crossover points are replaced by each other. Like fig 2, crossover operator is not certainly operated on all answers as parents, but rather its operating quantity is controlled by one of the genetic algorithm parameters called $P_c$. Also crossover does not certainly happen on the both chromosome parts, but rather this operation is completely independent on both parts. So, crossover operation could happen on both parts, one of them or none of them.

**Mutation Operator**
This operator makes little differences in child answers to block algorithm trouble in areal optimizing. So, it is tried to spread chromosomes in all answer space till algorithm have the chance to find real optimizing answer. For this algorithm, mutation operator in one gene from each chromosome layer make a difference. Reason to mutate in chromosome first part is to change the place of two chromosomes beside each other, in a way that activity prerequisite relations do not lose their order. In this project mutation operator has the ability to make answers in use and practical, and do not disturb activities prerequisite relations.

**Child chromosome before mutation**

| 1 | 2 | 5 | 3 | 4 | 6 |

**Mutation Point**

| 1 | 2 | 3 | 5 | 4 | 6 |

**Chart 3.** Mutation operator on chromosome first part.
In second part it's necessary for mutation operator to make healthy children; it means form of mutated chromosomes must be in a way that the mutated activity way of doing be in real too. Mutation of chromosome second part is shown in chart 4:

**Child chromosome before mutation**

| 1 | 2 | 3 | 2 | 1 | 4 |

**Mutation point**

| 1 | 2 | 1 | 2 | 1 | 4 |

**Chart 4.** Mutation operator on second part chromosome.
Mutation operation certainly is not done on all child answers, but rather it is controlled by a genetic algorithm parameter called $P_m$. Also
mutation operation is not definitely done on every part of the chromosome, but rather this operation is completely independent on every part. So, mutation operation could be done on every part, one of the parts or none of them.

**Case Study:** To study the present genetic algorithm performance in solving these kinds of questions, this time we have an example question to test presented algorithm. To do this, a project with 15 activities is chosen. These activities respectively are shown with numbers 1 to 15, which in these question prerequisite relations is shown in Table 2, maximum resources volume in Table 3 and project activities doing possible times in Table 1.

<table>
<thead>
<tr>
<th>Activity name</th>
<th>time</th>
<th>cost</th>
<th>R1</th>
<th>R2</th>
<th>R3</th>
<th>R4</th>
<th>R5</th>
<th>R6</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6</td>
<td>20000</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>5</td>
<td>22000</td>
<td>3</td>
<td>3</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>30000</td>
<td>7</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>15000</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>16000</td>
<td>4</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>19000</td>
<td>6</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>11000</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>7</td>
<td>10000</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>12000</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>14000</td>
<td>3</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>17000</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>10000</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>8000</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>8000</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>10000</td>
<td>2</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>18000</td>
<td>5</td>
<td>5</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>7000</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>7</td>
<td>6500</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>10000</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>12000</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>17000</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>9000</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>8000</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>6000</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>5000</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>4</td>
<td>4500</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>35000</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>38000</td>
<td>4</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>41000</td>
<td>7</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>4</td>
<td>30000</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>6</td>
<td>26000</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>6000</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td>5500</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Project Scheduling problem using metaheuristic algorithm

<table>
<thead>
<tr>
<th>Activity name</th>
<th>predecessors</th>
<th>Activity name</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>-</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>-</td>
<td>2</td>
</tr>
<tr>
<td>14</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>9,10,3</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>6,9,4</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>7,8,9</td>
<td>5.2</td>
<td>6</td>
</tr>
<tr>
<td>11</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>11,12</td>
<td>6</td>
<td>8</td>
</tr>
<tr>
<td>12,13</td>
<td>6,4,5</td>
<td>9</td>
</tr>
<tr>
<td>13,14</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>15</td>
<td>7.8</td>
<td>11</td>
</tr>
<tr>
<td>15</td>
<td>8,9</td>
<td>12</td>
</tr>
<tr>
<td>15</td>
<td>9,10</td>
<td>13</td>
</tr>
<tr>
<td>15</td>
<td>3,10</td>
<td>14</td>
</tr>
</tbody>
</table>

Table 2. Project prerequisite activities relations.
Table 3. Resources maximum supply.

<table>
<thead>
<tr>
<th></th>
<th>R6</th>
<th>R5</th>
<th>R4</th>
<th>R3</th>
<th>R2</th>
<th>R1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>5</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>infinite</td>
</tr>
</tbody>
</table>

**Solving example question by suggested algorithm**

To schedule and program case question by multi-index genetic algorithm first question information are created in program as question data and are put in software coding part (algorithm 1)

Algorithm 1: pseudocode of proposed algorithm

```matlab
for it=1:MaxIt
    % Selection Probabilities
    Costs=[pop.Cost];
    P=exp(-SelectionPressure*Costs/WorstCost);
    P=P/sum(P);
    % Crossover
    pop2=repmat(individual,nCrossover/2,2);
    for k=1:nCrossover/2
        i1=RouletteWheelSelection(P);
        i2=RouletteWheelSelection(P);
        p1=pop(i1);
        p2=pop(i2);
        [ch2.Cost ch2.Sol]=CostFunction(ch2.Position);
        pop2(k,1)=ch1;
        pop2(k,2)=ch2;
    end
    pop2=pop2(:);
    % Mutation
    pop3=repmat(individual,nMutation,1);
    for k=1:nMutation
        i=randi([1 nPop]);
        q.Position=Mutate(pop(i).Position,model);
        [q.Cost q.Sol]=CostFunction(q.Position);
        pop3(k)=q;
    end
    % Merge Populations
    pop=[pop
         pop2
         pop3]; %#ok
    % Sort Population
    Costs=[pop.Cost];
    [Costs SortOrder]=sort(Costs);
    pop=pop(SortOrder);
    WorstCost=max(WorstCost,Costs(end));
    % Delete Extra Individuals
    pop=pop(1:nPop);
    Costs=Costs(1:nPop);
```
% Save Results
BestSol=pop(1);
BestCost(it)=Costs(1);
BestT(it)=BestSol.Sol.T;
BestC(it)=BestSol.Sol.C;

% Show Information
disp(['Iteration ' num2str(it) ':   ' ...
     'Best Cost = ' num2str(BestCost(it)) ' , ' ...
     'Best T = ' num2str(BestT(it)) ' , ' ...
     'Best C = ' num2str(BestC(it))]);

End

To solve this question, algorithm parameters are put in this way:
- Population size = 100
- Crossover possibility \( P_c = 0/8 \)
- Mutation possibility \( P_m = 0/3 \)
- Maximum algorithm repetition = 100

Now, supposing cost and time indexes importance equal to each other, and ideal answer (optimized answer) is: Time = 25 and Cost = 173600, results gained from different performances are shown in this chart:

\[
\begin{align*}
T_{\text{min}} &= 17 \\
T_{\text{max}} &= 97 \\
C_{\text{min}} &= 154100 \\
C_{\text{max}} &= 322500 \\
W_T &= 0.5 \\
W_C &= 0.5
\end{align*}
\]

Table 4: Answers frequency table

<table>
<thead>
<tr>
<th>Time</th>
<th>Cost</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>173600</td>
<td>58</td>
</tr>
</tbody>
</table>

Table 5: Activity implementation priority

<table>
<thead>
<tr>
<th>Activity name</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>
<th>13</th>
<th>14</th>
</tr>
</thead>
<tbody>
<tr>
<td>duration</td>
<td>5</td>
<td>7</td>
<td>5</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>6</td>
<td>5</td>
<td>12</td>
<td>7</td>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td>Implementation method</td>
<td>2</td>
<td>5</td>
<td>5</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>

Activity implementation and time method

<table>
<thead>
<tr>
<th>Time</th>
<th>Cost</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>36</td>
<td>156000</td>
<td>70</td>
</tr>
</tbody>
</table>

Table 5: Activity implementation priority

<table>
<thead>
<tr>
<th>Activity name</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>
<th>13</th>
<th>14</th>
</tr>
</thead>
<tbody>
<tr>
<td>duration</td>
<td>5</td>
<td>7</td>
<td>5</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>6</td>
<td>5</td>
<td>12</td>
<td>7</td>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td>Implementation method</td>
<td>2</td>
<td>5</td>
<td>5</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>

Activity implementation and time method
Project Scheduling problem using metaheuristic algorithm

Figure 6. Question answer when $W_f=0/2$ and $W_c=0/8$

$$T_{\text{max}} = 97$$
$$T_{\text{min}} = 17$$
$$C_{\text{min}} = 154100$$
$$C_{\text{max}} = 322500$$
$$W_f = 0.6$$
$$W_c = 0.4$$

Table 10: Activity implementation priority

<table>
<thead>
<tr>
<th>Time</th>
<th>Cost</th>
<th>Consecutive frequencies</th>
</tr>
</thead>
<tbody>
<tr>
<td>24</td>
<td>176600</td>
<td>90</td>
</tr>
</tbody>
</table>

Table 11: Activity implementation priority

<table>
<thead>
<tr>
<th>activity name</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>
<th>13</th>
<th>14</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>duration</td>
<td>5</td>
<td>7</td>
<td>5</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>12</td>
<td>7</td>
<td>8</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>Implementation method</td>
<td>2</td>
<td>5</td>
<td>5</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>3</td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 12: Activity implementation and time method

Figure 7. Question answer when $W_f=0/6$ and $W_c=0/4$

We could say with opposite criteria in time-cost trade-off question, it’s impossible to know one method answer better than the other one. In time-cost trade-off question due to obtained answers DM could choose its ideal answers.

CONCLUSION

Presented algorithm model in this paper has been discussed to solve time-cost trade-off in limited resources condition by using a multi-index decision making method. Presented GA model has these advantages:

1. Could see both resources attribution question and time-cost trade-off question purposes

Amir Seyedan Anby
Project Scheduling problem using metaheuristic algorithm

together, which that results in solving time-cost trade-off question in limited resources. It
is clear that for common probing and math models it is hard to see both mentioned
purposes.

2: It could solve the question due to decision maker preferences in time and cost indexes.
3: Presented GA model in this paper is very flexible to solve the question with a
comparison to other methods, because of using experimental principles to solve it, limitations and purpose functions
formulization is not necessary if needed.

For more studies it’s possible to have a dynamic time-cost question modeling supposing having
needed resources to do activities on time; and also mention using probability theory concepts
in doing activities time cause of not having exact activities doing time. We suggest using
other optimizing algorithms which can work combined with present algorithm to raise
algorithm performance and have a resulted hybrid genetic algorithm.

REFERENCES
construction Scheduling ", J.Constr.Engrg and Mgmt, ASCE, Vol.125, No.6,
algorithm- based multi objective approach for Time-Cost optimization" , J.Constr.Engrg and Mgmt, ASCE,Vol.130, No.2
construction industry, Teach.Rop.No.9. The
construction Inst. Dept of Civ Eng., Stanford univ
9(3).396-320.,
programming ".,Civ.Engrg.Const.Res.Series No.2, University of Illinois, Urban,Champaign,
12.Davis E W (1973).Project scheduling under resource constraints historical review and
categorization of procedure. AIIE
algorithms to solve construction time-cost
16.Cengiz Toklu Y(2002). Application of
a genetic algorithm to construction scheduling
with or without resource constraints Can.J.Civ.Engrg, No. 29, 421-429.,
learning and GA to solve time-cost trade off
18.Goldberg D E(1989).Genetic algorithms in
search , optimization and machine learning 
.Addison Wesley publishing Co. Reading
Mass.


