Solving scheduling problem by multi-index genetic algorithm

Seyed Ahmad Sheibatolhamdy¹*, Mostafa Samady¹ and Ardeshir Shaabani²

¹Department of Industrial Management, Firoozkooh Branch, Islamic Azad University, Firoozkooh, Iran.
²Department of Public Management, Payamenoor University, P.O.B.19395-3697, Tehran, Iran.

Accepted 26 August, 2011

In this paper, we solve time-cost trade-off project in a way that the decision maker preferences and cares about resource restriction are sorted out simultaneously. For analysis of a part of NP-hard questions, we used a meta-initiative multi index genetic algorithm method for modeling and solving it. This paper estimates the complete project time by both time and cost elements. Also, the initiative SJS method was used for resource attribution. In this order, with regards to the aforementioned purposes, we calculated the project's completion time, and the decision maker could run to ideal answers by changing time and cost indexes importance. This question was coded in Matlab software and because of its complication, it reaches 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 the critical path method (CPM). 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 (Azaron et al., 2005). On the other hand, by question dimensions extension and complication math methods lose their performance, although they guarantee optimum answer (Sou-Sen and Chung-heui, 1999). By employing math method, changing probing methods principles to restrictions and purpose function is necessary and has some lapses possibly (Daisy et al., 2004).

AN OVERVIEW OF SOLVING SCHEDULING PROBLEM BY MULTI-INDEX GENETIC ALGORITHM

Numbered common probing models for solving TCT question are Fondahl (Fondahl, 1961), Parger structural method (Parger, 1963), Moslehi stiffness structural model (Moselhi, 1993) and Siemens model (Siemens, 1961). Having math methods more punctual than probing ones, they gained more attention. For instance we mentioned critical path planning [8], integer planning model (Mayer and Shaffer, 1963) and dynamic programming (Robinson, 1995), and also integer and linear planning hybrid model (Liu and Burns, 1995).

In resource allocation questions, linear and dynamic planning models are some employed mathematical methods (Davis, 1973). For avoiding complex optimizing question, intuitive principles are used to solve these kinds of questions. Due to previous-mentioned items and both crucial time and cost indexes, the need to use Carter’s...
algorithm is clear.

GA algorithm, due to its natural characteristic to search randomly based on purpose and in possible space, has the ability to make multi-criterion question model in this field. For the first time in 1997, some researchers named Li and Love used GA in solving time-cost trade-off (TCT) (Li, 1997). Feng did the same also (Feng and Burns, 1996). Other examples could be GA application in construction time with or without resources limitation (Cengiz, 2002) and machine learning and GA application in time-cost trade-off question solving (Heng and Love, 1999). 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 (Daisy et al., 2004) and multi-criterion optimizing model for construction scheduling by using GA (Sou-Sen and Chung-heui, 1999). However, in all the mentioned models, the decision maker preferences of using the best item were not accounted for.

MULTI-CRITERION GENETIC ALGORITHM MODEL

Genetic algorithm simulates living organisms' natural genetic process. Noticeable GA power and its ability in one random item are used 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 (Goldberg, 1989).

GA MODEL PROPERTIES

Time-cost relation

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

Chromosome appearance

Due to resource limitation, doing activities and their way of doing ordering are decision variants, so we are faced with two chromosome appearances. In this way, chromosome is in two parts: the first part comprises the activities used for carrying out ordering, and the second part comprises information about their method. This structure is shown in Figure 1.

This kind of answer could be known as mutation chromosome. In every block of this chromosome, there is a natural number. Naturally, this chromosome has 2n blocks; 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 A1 to An, and each activity mi shows that the activity method costs money, time and specific resources. Based on this, a set of m is defined for each Ai activity of which mi is one of them.

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. Subsequently, we will explain both.

In most scientific studies to solve resources attribution question which have used meta-initiative methods, serial SGS has been used for project time calculating (Hartmann, 1998). Some others have used parallel SGS (Lee and Kim, 1996). 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 (Hartmann, 1998).

Many experimental researches have shown that serial method have better results than parallel method at the time many scheduling are done for a project (Hartmann and Kolisch, 2000). 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 it is impossible to foresee which method is better.

FITNESS FUNCTION

In single-index optimizing, there is an explanation of solving method, but not in multi-indexes. First decision
making (DM) 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:

\[
\text{Preference Matrix} = [T_{ij}, C_{ij}]
\]

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

\[
W = [w_t, w_c]
\]

In this method, every items adequacy is obtained by:

\[
\text{fitness value}(x) = \frac{\sum W_j \times r_{ij}}{\sum_j W_j}
\]

In this formula, \( r_{ij} \) is non-scale fitness i order item and j order index.

If fuzzy method is used for non-scaling, then, last items fitness will be obtained by:

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

\( w_t \) and \( w_c \) are non-scale fitness values. \( T_{\text{max}}, T_{\text{min}} \) are time criterion maximum. \( C_{\text{max}}, C_{\text{min}} \) are cost criterion maximum. \( T(x), C(x) \) 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.

**PROBLEM SOLVING**

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.

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, Appendix).

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 Figures 2, 3 and 4.

In Figure 2, cost and time indexes importance equal to each other and in Figures 3 and 4 cost and time indexes importance is different.
CONCLUSION

The 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. From this study, it could be observed that both resources attribution question and time-cost trade-off question 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. It could solve the question due to decision maker preferences in time and cost indexes.

Presented GA model in this paper is very flexible to solve the question with a comparison to other methods,
because using experimental principles to solve it, limitations and purpose functions formulization is not necessary if needed. We could say with opposite criteria in time-cost trade-off question, it is 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.

For more studies, it is possible to have a dynamic time-cost question modeling, supposing there is need for resources to be used to perform activities on time. Also, it should be mentioned that the probability theory concepts of doing activities does not explain the exact time of doing activities. We suggest that other optimizing algorithms which can work in combination with the present algorithm should be used to raise algorithm performance and should be able to result to a hybrid genetic algorithm.

ACKNOWLEDGEMENTS

This research work was done at Firoozkooh branch, Islamic Azad University, Firoozkooh, IRAN. The authors would like to thank them for their financial support.

REFERENCES

APPENDIX

for it = 1: MaxIt
    Percent selection probabilities
    Costs = [pop. Cost];
    P = exp (-Selection pressure*costs/worstcost);
    P = P/sum (P);
    Percent crossover
    pop2 = repmat (individual,nCrossover/2,2);
    for k = 1: ncrossover/2
        i1 = Roulette wheel selection (P);
        i2 = Roulette wheel selection (P);
        p1 = pop (i1);
        p2 = pop (i2);
        [ch2. Cost ch2.Sol] = Cost Function (ch2.Position);
        pop2 (k, 1) = ch1;
        pop2 (k, 2) = ch2;
    end
    pop2 = pop2(:);
    Percent mutation
    pop3 = repmat(individual,nMutation,1);
    for k = 1:nMutation
        l = randi ([1 nPop]);
        q. Position = Mutate (pop (i). Position, model);
        [q. Cost q. Sol] = Cost Function (q.Position);
        pop3 (k) = q;
    end
    Percent merge populations
    pop = [pop
            pop2
            pop3]; %#ok
    Percent sort population
    Costs = [pop. cost];
    [Costs sort order] = sort (Costs);
    pop = pop (sortorder);
    Worst cost = max (Worst cost, costs (end));
    Percent delete extra individuals
    pop = pop (1:nPop);
    Costs = Costs (1:nPop);
    Percent save results
    Best Sol = pop (1);
    Best cost (it) = Costs (1);
    BestT (it) = BestSol.Sol.T;
    BestC (it) = BestSol.Sol.C;
    Percent show Information
    disp ('Iteration ' num2str(it) ': ' ...

Algorithm 1. Pseudo code of proposed algorithm.