A Bi-objective Genetic Algorithm Approach to Project Scheduling under Risk*

Kılıç M[1], Ulusoy G[2], and Sivrikaya Şerifoğlu F[3]

[1]Doğuş University, Istanbul
e-mail: mkilic@dogus.edu.tr

[2]Sabancı University, Istanbul
e-mail: gunduz@sabanciuniv.edu

[3]Abant Izzet Baysal University, Bolu
e-mail: serifoglu_f@ibu.edu.tr

Keywords: project scheduling, risk, multiobjective optimization, genetic algorithms.

1. Problem definition

In this paper, a mixed integer programming model and heuristic solution approaches based on genetic algorithms (GAs) are proposed to attack the problem of project scheduling under risk. This problem has been relatively rarely addressed in the literature (Ulusoy, 2002).

A set of risks is defined associated with each task (activity), where each risk has an impact and a probability of occurrence associated with it. Risks only affect the duration of the related task when they occur. A project manager can decrease the probability of occurrence and impact of each risk by taking some preventive measures at a predefined cost. The model has no resource constraints. It is assumed that the risks are independent and their impacts are additive at the activity level. It is further assumed that all the risks associated with an activity are identified and the risks are static throughout the project life. The problem is represented on an activity-on-node (AON) network with one starting and one ending node.

Notation:

$\{J\}$: Set of activities $j=1,\ldots,J$;
$\{P_j\}$: Set of immediate predecessors of activity j;
$\{L_j\}$: Set of resource types for activity j;
$\{N_j\}$: Set of risks n assigned to activity j;
d_j: Duration of activity j with no risks involved;
d_j : Expected duration of activity j under risk;
C_p: Unit penalty cost of being late;
C_o: Unit cost of overhead;
C_{lj}: Unit cost of resource type l_j;
C_{jnk} : Cost of reducing the risk level from state 1 to state k for risk n at activity j;
K_{jn}: Number of states for the probability of occurrence of risk n on activity j;
P_{jnk}: Probability of the occurrence of risk n for activity j at state k;
I_{jnk}: Impact of risk n, if it occurs, for activity j at state k;
W_{lj}: Number of workers of type l_j assigned to activity j;
T_{plan}: Due date set for the project;
$E(TC)$: Expected total cost;
EST_j: Earliest start time of activity j;
EFT_j: Earliest finish time of activity j;
EFT_J: Expected makespan of the project;
y: Expected lateness of project;

$$X_{jnk} = \begin{Bmatrix} 1, & \text{if the } k^{th} \text{ state is chosen for } n^{th} \text{ risk of } j^{th} \text{ activity} \\ 0, & \text{otherwise} \end{Bmatrix};$$

* Published in the "*Proceedings of the Ninth International Workshop on Project Management and Scheduling*", A. Oulamara, M-C. Portman (Editors), 58-61, Nancy, France, 2004.

Model:

$$Min\,E(TC) = y * C_p + \sum_{j=1}^{J}\sum_{n=1}^{N_j}\sum_{k=1}^{K_{jn}} C_{jnk} * X_{jnk} + C_o * EFT_J + \sum_{j=1}^{J}\sum_{l_j=1}^{L_j} W_{l_j} * d'_j * C_{l_j} \quad (1.1)$$

$$Min\,E(C_{\max}) = EFT_J \quad (1.2)$$

subject to:

$$EST_1 = 0 \quad (1.3)$$

$$EST_j = Max\{EFT_i \mid i \in P_j\} \qquad j = 2,...,J \quad (1.4)$$

$$EFT_j = EST_j + d'_j \qquad j = 1,...,J \quad (1.5)$$

$$d'_j = d_j + d_j * \sum_{n=1}^{N_j}\sum_{k=1}^{K_{jn}} X_{jnk} * I_{jnk} * P_{jnk} \qquad j = 1,...,J \quad (1.6)$$

$$C_{jn1} = 0 \qquad j = 1,...,J;\ n = 1,...,N_j \quad (1.7)$$

$$\sum_{k=1}^{K_{jn}} X_{jnk} = 1 \qquad j = 1,...,J;\ n = 1,...,N_j \quad (1.8)$$

$$y = \begin{Bmatrix} EFT_J - T_{plan}, & \text{if } EFT_J > T_{plan} \\ 0, & \text{otherwise} \end{Bmatrix} \quad (1.9)$$

$$X_{jnk} \in \{0,1\} \qquad j = 1,...,J;\ n = 1,...,N_j;\ k = 1,...,K_{jn} \quad (1.10)$$

This model aims to minimize expected total cost (1.1) of the project and the expected makespan (1.2). These two objectives are conflicting. The expected total cost is represented as the sum of four cost components: the penalty cost for lateness formulated as the product of unit penalty cost and lateness, the cost of risk reductions, the overhead cost formulated as the product of unit overhead cost and makespan, and the labor cost of the project.

Equations 1.3 to 1.5 are the critical path method (CPM) equations for forward recursion. Equation 1.6 is used to calculate the expected duration of an activity by adding the additional risk related durations to normal activity duration. Equation 1.7 defines C_{jn1}. Equation 1.8 assures that one and only one state for each one of the risks is selected

There are ($\sum_{j=1}^{J}\sum_{n=1}^{N_j}\sum_{k=1}^{K_{jn}} 1$) number of 0-1 decision variables. Small sized problems can easily be solved by a mathematical programming solver. But for large problems, computational costs become prohibitive. Hence, heuristic approaches are proposed.

Table 1 illustrates an activity for which there is only one risk involved with three states. TU and MU stand for time unit and monetary unit, respectively. The first state of the risks, named as the base case, has a zero cost. This corresponds to the real life situation of taking no preventive measures against a risk. Thus no cost is incurred. If necessary measures are taken to reduce the risk level from state 1 to 2, the expected duration of the activity becomes d'_x= 20+0.6*0.5*20=26 TU.

Table 1. Risk states for the example activity

Activity X Duration (d): 20 (TU) $\lvert L_X \rvert = 1$ W_{1X}= 3			
State	Probability of Occurrence (P_{jnk})	Impact (I_{jnk})	Cost (MU) (C_{jnk})
1	0.7	0.5	0
2	0.6	0.5	150
3	0.6	0.4	300

2. Solution approach

Multiobjective optimization with posteriori preference articulation is a developing topic in the OR literature. GAs constitute a popular solution procedure for this group of problems;

approximately 70% of the metaheuristic approaches suggested and published between 1991 and 2000 are GAs (Jones et al., 2002). Since GA uses parallel search techniques and multiobjective optimization problems have several nondominated solutions, this problem class and the solution procedure make a good match. Two improvement heuristics are proposed for further improving the solutions found by the GA. They try to decrease the expected total cost while keeping the critical path fixed.

2.1. The GA approach

Direct representation is used for encoding a solution. Each gene corresponds to a risk. The number in the gene represents the state that will be chosen for the corresponding risk. An extra three-bit portion is added to the chromosome to display the expected makespan, expected total cost and fitness values.

Roulette wheel selection is used as the selection mechanism. One point crossover is used to generate two offspring from two parent chromosomes. Bit mutation is used to replace the value on the randomly chosen gene of the chromosome with another randomly generated value.

The first population is generated randomly. For generating the subsequent populations, crossover, mutation and reproduction operators are applied in a parallel fashion contrary to the serial application in traditional GAs. First an operator is chosen: crossover with a probability of P_c, mutation with a probability of P_m and reproduction with a probability of $(1-(P_c+P_m))$. Then the chromosome(s) is (are) chosen according to the operator. Finally, the chosen operator is applied to the chosen chromosome(s).

Fitness is computed by the Formula (1.11). N_{dom} represents the number of individuals dominated by the current individual (chromosome) in the Pareto domination tournament involving the entire population. N_{pop} is the population size. *NNR* (nearest neighbourhood radius) gives the ratio of nearest individual's distance ($d_{nearest}$) to the maximum distance in the population (d_{maxpop}). *NNR* behaves like a sharing function; if an individual is closer to its nearest neighbour, it is assumed that it is in a crowded region. By multiplying R_{dom} and *NNR,* fitness of an individual in the nondomination sense and the sharing concept are combined.

$$\text{Fitness} = NNR * R_{dom} \qquad (1.11)$$
$$R_{dom} = (N_{dom} + 1) / (N_{pop} + 1) \qquad (1.12)$$
$$NNR = d_{nearest} / d_{maxpop} \qquad (1.13)$$

Elitism is applied by transferring the nondominated individuals in each population to the next one.

2.2. Improvement heuristics

Improvement heuristics are applied to the solutions obtained by the GA. They aim to avoid investing more money than needed to the non-critical activities to reduce their risks, while not changing the risk structure of the activities on the critical path and hence the makespan.

First a multi-mode project scheduling problem is formed by assigning every possible combination of states of all risks associated with an activity to different modes. For an activity with two risks having two states each, there will be four modes. A domination search over the modes is performed to eliminate the dominated modes. Within the modes of activities, the duration represents the expected durations when the corresponding risk states are chosen. The cost represents the sum of expected labor cost and the risk reduction costs, which constitute a local trade-off with the expected duration. The resulting problem is a discrete time/cost trade-off problem. A limit on expected project duration is specified, modes of activities on the critical path are fixed, and the expected cost is minimized while preserving the critical path. Exact solution approaches provided by Demeulemeester *et al*. (1996) may become computationally very costly, so heuristics are tried.

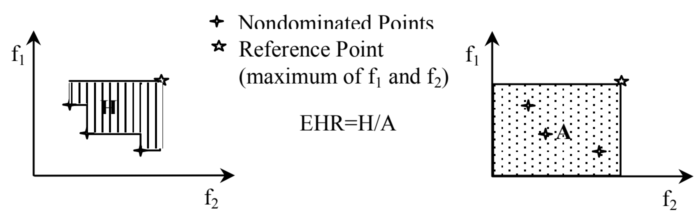
In the continuous cost vs. duration model based (CCDM) improvement heuristic, a linear curve is tried to fit to the time/cost scatter of the nondominated modes. By doing so, the problem is transformed to a continuous project crashing problem, which is easier to solve. For the cases, where a linear approximation is not adequate, a continuous curve is fitted, which is then approximated by a piecewise linear function. GAMS is used to solve the models, and modes are assigned to noncritical activities using the durations given by the GAMS solution.

In the GA results based (GAB) improvement heuristic, solutions provided by the GA themselves are used. If a solution involves dominated modes for noncritical activities, the nondominated mode with a lower duration is found.

The starting point solution obtained by either one of the above methods is subjected to the improvement routine. For each noncritical activity, slacks and the earning per duration value that will result if the activity is performed at its next higher duration mode are computed. Starting with the highest earning per duration activity, the durations are expanded without violating the slacks until slacks diminish to zero or there is no further mode to expand to. The earning per duration ratio is the ratio of the expected cost decrease to expected duration increase between the respective nondominated modes of the activity.

3. Computational study

A performance metric, "extreme hyperarea ratio (EHR)" is developed based on the idea of hyperarea ratio. This metric is the ratio of the hyperarea of the front (Figure 1(a)) to the area bounded by the origin and the so-called reference point defined by the maximum values of the two objective functions (Figure 1(b)).



This metric is used to compare GAs with different parameter settings in the finetuning process and to compare performances of different heuristic algorithms on test problems involving 15, 25 and 35 jobs.

To provide a comparison base, the true Pareto front is approximated by employing GAMS. The makespan objective is added to the model as a constraint to obtain a single objective cost minimization model to be repeatedly solved by GAMS. Limit on the makespan in the first such model is the maximum makespan. In a series of runs, the limit on the makespan resulting from the last GAMS run is decreased by increments of 0.01 until the minimum makespan is reached. Results from 60 GA runs are plotted together with the approximate Pareto front. For small problems with 15 activities, GA's performance is very good. For larger problems with 35 activities, the deviations are larger, and there is room for improvement. Indeed, the percentage of solutions improved by the application of improvement heuristics increases with problem size. But the average percentages of improvements in expected cost values are low. But this should be less of a concern, as the primary job of improvement heuristics is avoiding a trivially inferior solution.

4. Ongoing and future research topics

Other metaheuristic approaches may be tried. A priori and progressive preference articulation may be used, in case, real problem data and decision maker preference data are available. The problem may be recast in a form, where modes are to be decided upon rather than the risk states. The problem formulation may be made more realistic by allowing for dependent risks and/or resource constraints. The impacts and probability of occurrences of risks may be formulated using continuous functional forms.

References

Demeulemeester, E.L., W.S. Herroelen and S.E. Elmaghraby (1996). Optimal procedures for the discrete time/cost tradeoff problem in project networks. *European Journal of Operational Research*, 88, 50-68.

Jones, D.F., S.K. Mirrazavi, M. Tamiz (2002). Multiobjective metaheuristics: An overview of the current state-of-the-art. *European Journal of Operational Research*, 137-1, 1-9.

Ulusoy, G. (2002). Resource contrained scheduling in project planning., in Yöneylem Araştırması Halim Doğrusöz'e Armağan (Erkip, N. and M. Köksalan, eds); *METU Basım İşliği, Ankara*.