

Splittable stochastic project scheduling with constrained renewable resource

S.S. Hashemin*

Assistant Professor, Department of Industrial Engineering, Islamic Azad University, Science and Research Branch, Tehran, Iran

S.M.T. Fatemi Ghomi

Professor, Department of Industrial Engineering, Amirkabir University of Technology, Tehran, Iran

Abstract

This paper discusses the problem of allocation of constrained renewable resource to splittable activities of a single project. If the activities of stochastic projects can be split, these projects may be completed in shorter time when the available resource is constrained. It is assumed that the resource amount required to accomplish each activity is a discrete quantity and deterministic. The activity duration time is assumed to be a discrete random variable with arbitrary experimental distribution. Solving stochastic mathematical programming model of problem is very hard. So, here some existing methods for deterministic problems have been generalized for stochastic case. Solutions of generalized methods are relatively better than random solutions. However, the authors developed the new algorithm that may improve the solutions of generalized methods and project Completion Time Distribution Function (CTDF). Comparison of solution of a method with random solutions is a common assessment method in literature research. Hence, the efficiency of the proposed algorithm represented using this method.

Keywords: Project scheduling; Constrained resource; Renewable; Stochastic; Splittable; Completion time; Distribution function

1. Introduction

Sometimes some of stochastic project activities are splittable. Splitting of these activities increases the computations, which are necessary for activities scheduling and determining the CTDF. However, doing this is valuable because when part of the constrained resource has been allocated to the accomplishment of some activities, the rest of the resource is not sufficient to the accomplishment of a complete activity but it may be sufficient to the accomplishment of the part of splittable activity. Consequently, project can be completed in shorter time. Above saying is the different aspect of the previous researchers. Furthermore we developed the new algorithm for solving this problem.

There are many studies in literature regarding the resource allocation, in general. The estimation of completion time in PERT-type networks (PTN) is closely related to the constrained resource allocation

problem. This subject has been surveyed in some papers [1,7,11,32].

Brucker et al. [4] have reviewed notation, classification, models and methods in resource-constrained project scheduling. Herroelen et al. [22] have presented a new classification compatible to machine scheduling. Herroelen et al. [21] have surveyed the recent development in resource-constrained project scheduling. Herroelen and Leus [23] have surveyed and discussed the scheduling problem under uncertainty and research potentials of approaches.

Igelmund and Radermacher [24] have assumed that activities durations are realized according to some joint probability distributions. They have introduced a wide class of strategies, as preselective strategies. The expected overall project cost has been minimized using these strategies. Furthermore they have shown that the class of preselective strategies is very important because it can illustrate the stable behavior [25]. Mohring et al. [29] have presented some develop-

* Corresponding author. E-mail: s_s_hashemin@yahoo.com

ments in the theory of non-preemptive stochastic scheduling problem.

A complete characterization of set strategies has been given by Mohring et al. [30]. Golenko-Ginzburg [13] has developed a two-level decision-making model for controlling stochastic projects. He has also introduced upper and lower levels decision-making and heuristic procedures to solve this stochastic optimization problem. When the limited resources are consumable, Elmaghraby [6], using dynamic programming has minimized the project completion time. His proposed method solves the deterministic resource allocation problem. Martel and Ouellette [27] have examined the problem of allocating available amount of a particular resource among partially interchangeable activities. The problem has been formulated as a stochastic program with complete resource and it has been reduced to a deterministic convex allocation problem through parametric programming. Wan [35] has given sufficient condition to minimize the expected length of a stochastic CPM-type network. He has shown that a class of problems can be estimated efficiently through simulating at one, not different from value of parameters. Fernandez et al. [12] have considered nonanticipativity constraints because the constraints have not been considered by previous methods. They have provided potentially unattainable solutions. This approach has been used in commercial software for stochastic project scheduling. Golenko-Ginzburg et al. [14] have developed a hierarchical three-level decision making model. These levels are upper (company level), medium (project level) and subnetwork level. The main goal has been to develop a unified three-level decision making model and to indicate planning and control action and optimization problems for all levels. Bowers [3] has shown examining the probabilities of each activity lying on a critical path can identify the source of risk. A quantitative test has been developed and applied. This test has indicated that the critical probability may provide valuable management information. When the constrained resources are non-consumable, Golenko-Ginzburg and Gonik [15], using a zero-one integer programming, have maximized the total contribution of the accepted activities to the expected project duration. The contribution of each activity is the product of the average duration of the activity and its probability of being on the critical path. Golenko-Ginzburg and Gonik [16] have presented a new heuristic control algorithm for stochastic network projects. The developed control algorithm is essentially more efficient than the step-by-step control procedures. This algorithm has reduced computa-

tional time and has provided better solutions than would be attained using on-line sequential statistical analysis. Golenko-Ginzburg and Gonik [17] have developed a look over heuristic algorithm for resource-constrained in PTN. Each activity is of random duration depending on the resource amounts assigned to that activity. The aim has been to minimize the expected project duration. Tsai and Gemmill [34] have proposed Tabu search to solve stochastic resource-constrained projects. Superiority of Tabu search over the other heuristics has been shown. Mohring and Stork [31] have introduced the linear preselective policies. They have combined the benefits of preselective and priority policies. Efficient algorithms have been derived. Gokbayrak and Cassandras [19] have transformed stochastic discrete resource allocation problem into a on-line surrogate continuous optimization problem and have proceeded to solve the latter using standard gradient based approaches. Then surrogate problem methodology has been generalized [20]. Golenko-Ginzburg et al. [18] have developed an optimization procedure to maximize the probability confidence for project due-dates under budget constraints or to minimize the project budget under due-dates chance constraints. Chance-constrained programming model has been reviewed from the points of view of accuracy and validity by Elmaghraby et al. [9]. They have obtained a lower bound for cumulative distribution function of project completion time. Elmaghraby [8] has proposed a dynamic programming approach for problem that n jobs have been processed by single and multiple processors. It has some similarities with constrained resource allocation problem. Where a large number of candidate new products must undergo a set of tests for certification, Choi et al. [5] have developed a new way to combine heuristic solution through dynamic programming in the state space generated by heuristics. Azaron and Memariani [2] have developed a bicriteria model for the resource allocation problem in PTN, in which the total direct costs of the project as first objective, and the mean of project completion time as the second objective is minimized. The goal attainment method has been used to solve this problem. Resource allocation problem under stochastic condition for multimodal activity networks has been presented by Tereso et al. [33]. This problem has been solved by dynamic programming. Furthermore approximation schemes have been suggested. Elmaghraby [10] has shown that gross errors may be committed in cost estimates by replacing the random variables by their averages.

In this paper, by splitting the activities, the CTDF

of project has been improved. First, the authors transform the original network of project to another network to make the possibility of activity splitting (Section 3). Then some of existing methods for deterministic problems have been generalized for stochastic case (Section 4). Comparison of a method's solution with random solutions is a common assessment method in literature research [28]. Generalized methods have been evaluated in some examples using the above method (Section 5). Since the results are not satisfactory, the authors developed a new algorithm (Section 6).

This algorithm will improve the solutions of previous generalized methods and CTDF of the project. The efficiency of proposed new algorithm is represented in the same examples. Section 7 is devoted to the conclusions and recommendations for future studies.

2. The problem

In this paper, it is supposed that:

- a) Project can be formulated as PERT-type network.
- b) Activity implementation requires only one kind of renewable resource.
- c) The amount of available resource is limited and deterministic.
- d) Allocation of resource to activities performed, discretely. In the other word, the amount of the resource allocated to each activity denoted with a positive integer number which is deterministic.
- e) Duration of network activities are arbitrary discrete random variables. Its probability or distribution function may be obtained using experiments in real world.
- f) Some of network activities are splittable such that even by one unit of limited resource, we can accomplish a part of an activity.

Defining the sequence of limited resource allocation to activities is the main aim such that CDTF of project be maximized within some specified period of time. If this time be the due date, in fact we want to minimize the probability of the delay. These two aims are completely the same because if the due date and random variable of project completion time are represented with t and T , respectively, then:

$$P(T \leq t) = 1 - P(T > t)$$

and

$$Max P(T \leq t) \sim Min P(T > t).$$

3. Transformation of original network of project to another network to make the possibility of activity splitting

Suppose the activity j of project requires k_j units of constrained resource to be accomplished. This activity is transformed to k_j parallel activities, in such a way the random variable of completing these parallel activities would be the same as the random variable of completion time of activity j in the original network of project, but each of these parallel activities requires only one unit of constrained resource to be accomplished.

In the original network of project, each activity is defined by an ordered pair. The first element represents the random variable of activity duration time and the second element represents the required amount of resource to perform the activity. With this definition, the following figure suitably illustrates the transformation.

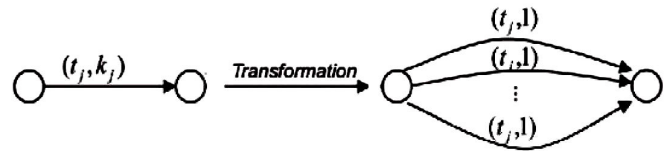


Figure 1. A perspective to illustrate the transformation.

In the above figure, the number of parallel activities is k_j .

In order to have no change in project CTDF when simulation is done, the equality of random variable t_j with random variable $\max(t_j, t_j, \dots)$, where the number of t_j in the parenthesis is k_j , should be guaranteed. But we know that this equality does not hold in general cases. But if we account the generated random variable of t_j equally for all transformed parallel activities in each run of simulation, completion time of both projects (original and transformed projects) would be the same. This operation is logically correct, because activity j , as an activity, has a random time to be accomplished and we can perform it at most in k_j non-overlapping portions (split).

4. Generalization of some of existing heuristics

Some of scheduling methods for allocation of limited renewable resources in deterministic networks have been introduced in Table 1.

Table 1. Heuristic methods for allocation of renewable resources in deterministic networks.

Rule	Notation	Operating Features
Minimum Activity Slack	MINSLK (MINLST)	Schedules first those activities with lowest activity slack time (total float). This means that activities schedules based on latest start time.
Minimum Late Finish Time	LFT	Schedules first those activities with the earliest values of late finish time.
Resource Scheduling Method	RSM	Priority index calculated on basis of pair-wise comparison of activity early finish and late start times. Gives priority to activities roughly in order to increase late finish time.
Greatest Resource Demand	GRD	Schedules first those activities with greatest resource demand in order to complete potential bottleneck activities.
Greatest Resource Utilization	GRU	Gives priority to that group of activities which results in the minimum amount of idle resources in each scheduling interval. Involves an integer linear programming logarithm.
Shortest Imminent Operations	SIO	Schedules first those activities with shortest durations in an attempt to complete the greatest number of activities within a given time-span.
Most Jobs Possible	MJP	Gives priority to the largest possible group of jobs which can be scheduled in an interval. Involves an Integer linear programming logarithm.
Random Activity Selection	RAN	Priority given to jobs selected at random, subject to resource availability limits.

Among these methods it has been shown that the first two methods (MINLS and MINLF) are more efficient and give better results in comparison with the other methods [28].

Here, the authors generalize the two above-mentioned efficient methods for stochastic networks.

4.1. Generalization of Min LS and Min LF methods

The delay of critical activities in deterministic projects is to be minimized. In other words, they have the highest preference. Other activities also have been arranged based on total floating time, ascensionally. However, total floating time of activities in PERT-type networks are random variables. Based on the same reasoning we generalized the Min LS method as two different forms.

In algorithm (*G*), the average of total floating time of activities are being estimated by simulation. Then activities are being arranged based on average total floating time, ascensionally. This method is noted by $\overline{Min LS}$.

In algorithm (*H*), in each run of simulation, activities are being arranged based on total floating time, ascensionally. Then each sequence is being saved. After the completion of simulation the sequence which has the highest frequency has been selected. This method is noted by $\hat{Min LS}$.

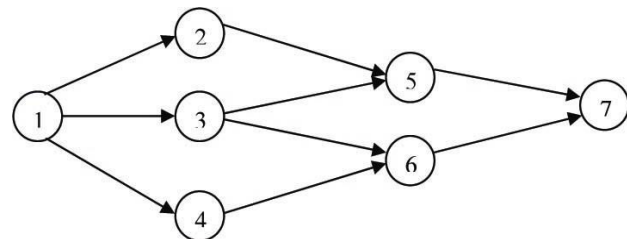
It is evident that activity with smallest LFT (Latest finish time) must be completed as soon as possible.

Hence, the activities can be arranged based on LFT. In algorithms (*I*) and (*J*) we generalized these methods for PERT-type networks the same as algorithms (*G*) and (*H*), respectively. In these algorithms LFT has been utilized as criteria instead of LST. Applied methods in algorithms (*I*) and (*J*) are noted by $\overline{Min LF}$ and $\hat{Min LF}$, respectively. Steps of these algorithms have been described in the appendix A.

5. Numerical examples

It has been attempted to have different configurations for structures of networks considered in the examples. Regarding the reference [26], four different types of structures have been considered for networks of examples. They are as follows:

Type A. The number of parallel activities is more in the initial portion of network and less in the final portion, like:

**Figure 2.** Network (A)

Type B. The number of parallel activities is less in the initial portion of network and more in the final portion, like:

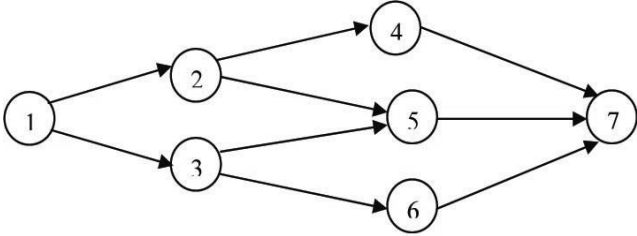


Figure 3. Network (B)

Type C. The number of parallel and independent paths are high in the network, like:

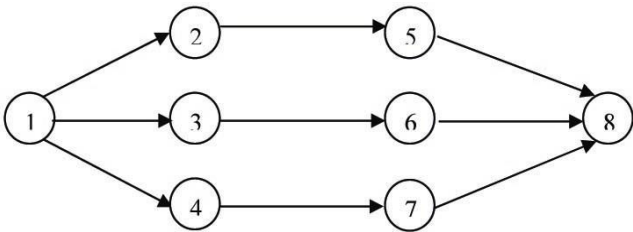


Figure 4. Network (C)

Type D. The number of parallel activities, in the initial or final portion of network, does not make so much difference. The paths are not independent of each other and have common activities, like:

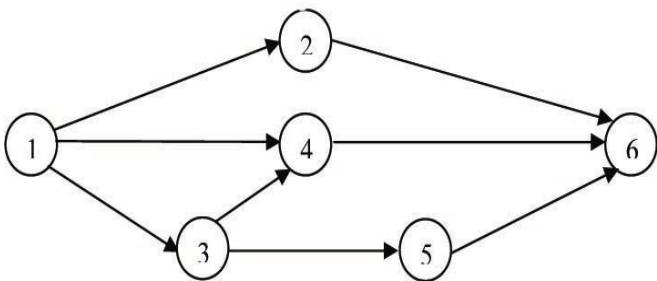


Figure 5. Network (D)

In relation to the distribution of activity duration time, it has been attempted to account the real world situation as much as possible. Usually, in real world,

the information regarding duration time of an activity is available in the following table:

i	t_i	f_i
1	t_1	f_1
2	t_2	f_2
\vdots	\vdots	\vdots
n	t_n	f_n

Fitting a known distribution to the above random variable not only necessitates a tedious statistical computations, but also it follows a remarkable error. Hence, the distribution of activity duration time is assumed to be discrete in this research, the values of cumulative frequency distribution for each class i

$$(CF_i) \text{ can be computed as } CF_i = \sum_{u=1}^i f_u .$$

The random generation of duration time of activity j , using the above assumptions and information, may be performed as the following:

$$P = RND \cdot \sum_{i=1}^n f_i . \tag{1}$$

If $CF_{m-1} < P \leq CF_m$ then, $t_j = t_m$ where t_j is the generated random time for duration time of activity j . Note that $CF_0 = 0$. However, three different types of distribution have been considered for project activities, their characteristics are shown in the Figure 6.

Each type of network has random activity duration time of types 1, 2 and 3 respectively. The numerical values of each distribution have been the same for each type of network.

To distinguish the superior method (or methods) in every example, it is vital to have a suitable criterion for comparison. For this purpose, let:

$$F_{LS}^-(t), F_{LS}^+(t), F_{LF}^-(t), F_{LF}^+(t)$$

would be the values of project CDF achieved by the different methods for given t . These values are obtained using the algorithms (G), (H), (I) and (J), respectively.

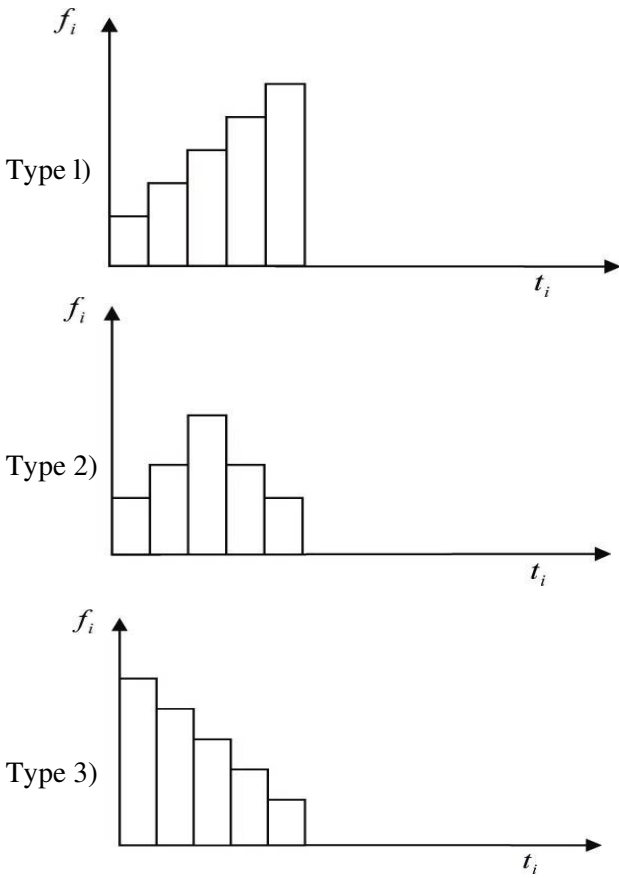


Figure 6. Different types of activity duration distributions.

The method corresponding the greatest gained $F(t)$ is named the superior method at point t . For example, for given t , if the value of distribution function by $Min \overline{LF}$ is greatest between four values of distribution functions gained by different methods, the point $(t, F_{\overline{LF}}(t))$ would be a superior point for $Min \overline{LF}$ method.

In Table 2, the number written in the opposite of each method represents the number of superior points.

Table 2. Results of numerical examples.

Network type Distribution type		A	B	C	D
		1	$\overline{LS} : 13$ $\hat{LS} : 6$ $\overline{LF} : 1$ $\hat{LF} : 0$	$\overline{LS} : 3$ $\hat{LS} : 3$ $\overline{LF} : 3$ $\hat{LF} : 9$	$\overline{LS} : 1$ $\hat{LS} : 1$ $\overline{LF} : 12$ $\hat{LF} : 12$
2	$\overline{LS} : 15$ $\hat{LS} : 15$ $\overline{LF} : 0$ $\hat{LF} : 0$	$\overline{LS} : 7$ $\hat{LS} : 7$ $\overline{LF} : 8$ $\hat{LF} : 8$	$\overline{LS} : 9$ $\hat{LS} : 13$ $\overline{LF} : 2$ $\hat{LF} : 8$	$\overline{LS} : 13$ $\hat{LS} : 12$ $\overline{LF} : 2$ $\hat{LF} : 2$	
3	$\overline{LS} : 16$ $\hat{LS} : 16$ $\overline{LF} : 3$ $\hat{LF} : 3$	$\overline{LS} : 5$ $\hat{LS} : 11$ $\overline{LF} : 6$ $\hat{LF} : 6$	$\overline{LS} : 11$ $\hat{LS} : 8$ $\overline{LF} : 1$ $\hat{LF} : 8$	$\overline{LS} : 6$ $\hat{LS} : 10$ $\overline{LF} : 1$ $\hat{LF} : 1$	

In 61% of points, \overline{LS} and \hat{LS} have been superior methods. In 23% of points, \overline{LF} has been superior method. In 33% of points, \hat{LF} has been superior method. The sum of above percents is more than 100%, because more than one method have been commonly superior in some points.

5.1. Evaluation of generalized methods

Among the generalized methods $Min \overline{LS}$ and $Min \hat{LS}$ are better than others. However, quality of their solutions compared with optimal solution are unknown. Generation of random solutions and obtaining better random solutions percent (BRSP) can be suitable criteria for defining the quality of the solution of methods. Utilizing this method is common in research literature [28] because when feasible solutions are being generated randomly and uniformly, then any kind of solutions will be existed among them. Therefore $(100\% - BRSP)$ will present the quality of the solution of methods relatively. Table 3 represents the $(100\% - BRSP)$ of the best generalized method in previous examples.

Table 3. Results of evaluation of generalized methods.

Network type Distribution type		A	B	C	D
		1	%34	%64	%82
2	%47	%68	%74	%65	
3	%58	%72	%73	%56	

Above results show that the best solution among the solutions of generalized methods is better than random solutions in %61.75 of cases (average percentage). Hence they are relatively better than random solutions but this average percentage (61.75%) is not satisfactory. Therefore, the authors developed the new algorithm which may improve the generalized methods solution.

6. Development of new algorithm

Obtaining the valuable information using better random solutions is the main aim of the algorithm creation. To define the better random solutions, sum of the differences of CTDF of solutions (random solution and generalized method solution) must be computed. Using the obtained information, some of adjacent activities can be swapped in existing sequence. The new obtained sequence can be significantly better than the previous sequence, mostly.

Consider the matrix $A_{N \times N}$ (N= number of the activities network). If $a_{ij} > a_{ji}$, the activity i is prior to activity j . To determine the elements of A, we use the better random solutions. In each better random solution, if activity i is prior to activity j , then we add the positive number to a_{ij} . This added positive number is proportion to the sum of differences of CTDF of solutions. Then we can obtain the new sequence by swapping the adjacent activities (using a_{ij} elements). By computing the BRSP for new sequence, we can be sure that new sequence is better than previous one. These operations can be repeated while the existing sequence may not be improved.

6.1. New algorithm

First we define the following notations:

- $F_e(t)$: CTDF for exiting sequence.
- $F_q(t)$: CTDF for random sequence, generated in q -th simulation run.
- $X_{[1], \dots, X_{[N]}}$: Arranged activities in existing sequence.
- $X'_{[1], \dots, X'_{[N]}}$: Arranged activities in improved sequence.
- M : Number of generated random sequences.
- $X^q_{[1], \dots, X^q_{[N]}}$: Generated random sequence in q -th run of simulation.

Figure 7 shows the new algorithm as a flowchart. After the execution of algorithm, if BRSP is not satisfactory, we can change the primary sequence of algorithm to one of the better random sequences. This task causes that algorithm to improve the solution of problem again.

6.2. Improvement of examples solutions using new algorithm

Table 4 represents the (100% - BRSP) of solutions, generated by the new algorithm. The asterisk sign (*) show problems which their (100% -BRSP) were not satisfactory (smaller than %90). In these problems primary sequence has been surrogated with one of the better random sequences. Then we execute the algorithm again. Consequently, previous solution can be improved, significantly. The results of Table 4 shows that the new algorithm is more efficient than all of generalized methods.

Table 4. (100% - BRSP) of improved solutions, using the new algorithm.

Network type \ Distribution type	A	B	C	D
1	%99	%100*	%98*	%100*
2	%99	%93	%99*	%90
3	%94	%94*	%98*	%99*

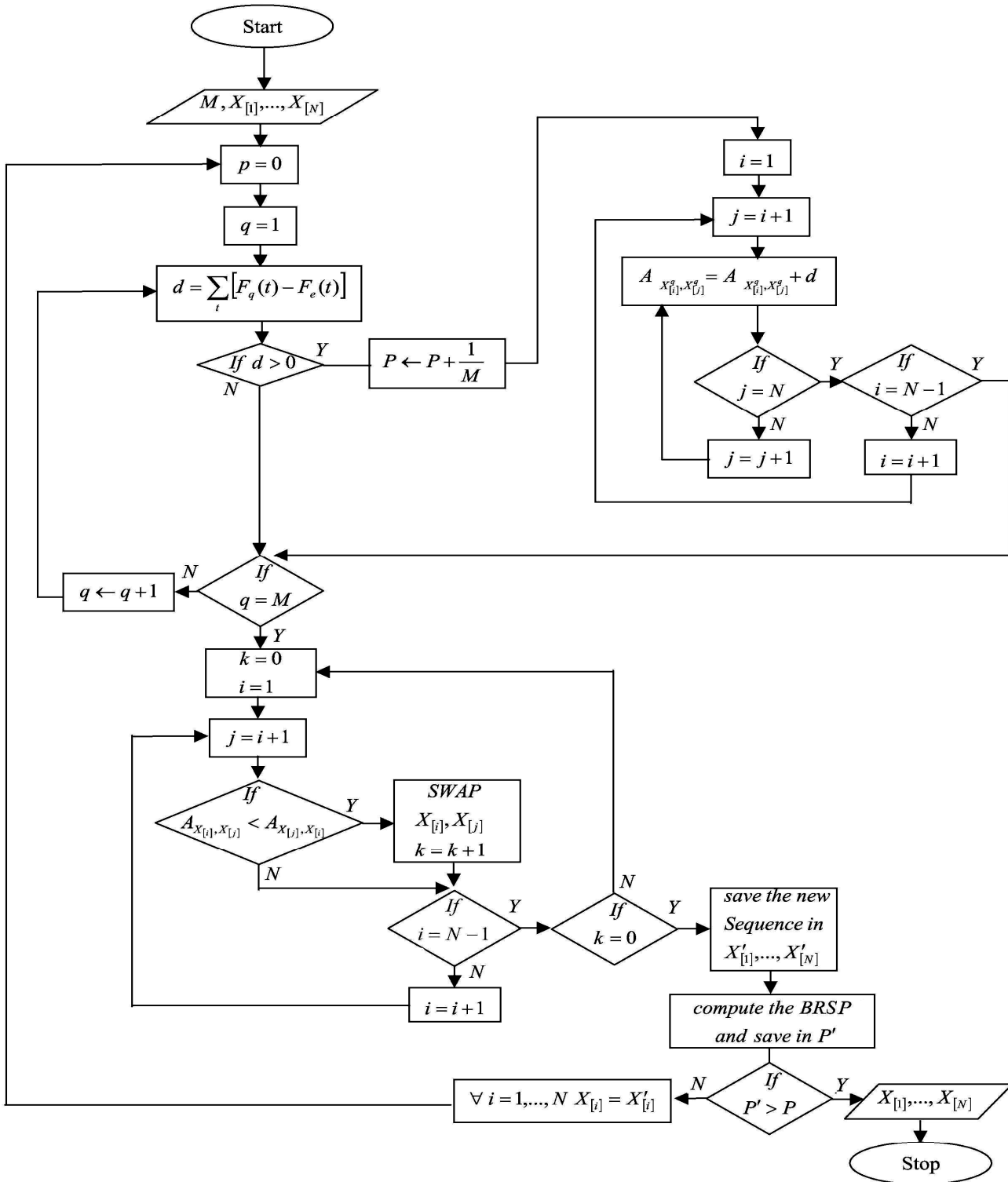


Figure 7. Flowchart of new algorithm.

7. Conclusion and recommendations for future research

The following remarks are some conclusions of this research:

- Min \overline{LS} and Min \hat{LS} methods act better than other generalized methods in the project that the activities are split.
- In the case of activities are split, the project, undoubtedly, will have the better CDF than that of the case the activities cannot be split.
- One of the outstanding characteristics of the new algorithm which was clarified in practice is that this algorithm improves the primary solution by a few number of random solutions (almost 100 solutions) and a few number of simulations (almost 10 run).
- Some limitations may exist in the activity splitting. This study has split the activities, in such a way that each portion needs one unit of resource for each unit of time. But the planner can split the activities with the resource requirement of more than one unit for each unit of time for the portions. If this kind of splitting happens, the new algorithm developed by authors still has the capability of scheduling the activities of project.

The following recommendations are made for future studies:

- Similar research can be made on the projects, which the resource is constrained, but not renewed.
- The research can be made for the case more than one kind of constrained resource is concerned.
- The proposed algorithm may be generalized for the projects having stochastic activities with continuous random variables for duration time.
- In some projects the matter of resource leveling is an interesting and practical subject. So, a new research can be directed from the viewpoint of resource leveling. In this respect, the methods can be compared and also some new methods may be suggested to facilitate the operation of resource leveling.
- The research can be made for the case that allocation of resource to activities is done continuously.

References

- [1] Abdelkader, Y. H., 2004, Evaluating project completion times when activity times are Weibull distributed. *European Journal of Operational Research*, 157(3), 704-715.
- [2] Azaron, A. and Memariani, A., 2004, Bicriteria resource allocation problem in PERT networks. *International Journal of Engineering Transactions A: Basics*, 17(2), 157-168.
- [3] Bowers, J., 1996, Identifying critical activities in stochastic resource constrained networks. *Omega, International Journal of Management Science*, 24(1), 37-46.
- [4] Brucker, P., Drexl, A., Mohring, R., Neumann, K. and Pesch, E., 1999, Resource-constrained project scheduling: Notation, classification, models and methods. *European Journal of Operational Research*, 112(1), 3-41.
- [5] Choi, J., Realff, M. J. and Lee, J. H., 2004, Dynamic programming in a heuristically confined state space: A stochastic resource-constrained project scheduling application. *Computers and Chemical Engineering*, 28, 1039-1058.
- [6] Elmaghraby, S. E., 1993, Resource allocation via dynamic programming in activity networks. *European Journal of Operational Research*, 64, 199-215.
- [7] Elmaghraby, S. E. and Agrawal, M. K., 2000, On computing the distribution function of the sum of independent random variables. *Computers and Operations Research*, 28, 473-483.
- [8] Elmaghraby, S. E., 2001, On the optimal release time of jobs with random processing times, with extensions to other criteria. *International Journal of Production Economics*, 74(1-3), 103-113.
- [9] Elmaghraby, S. E., Soewandi, H. and M. J. Yao, 2001, Chance constrained programming in activity networks: A critical evaluation. *European Journal of Operational Research*, 131(2), 440-458.
- [10] Elmaghraby, S. E., 2005, On the fallacy of averages in project risk management. *European Journal of Operational Research*, 165(2), 307-313.
- [11] Fatemi Ghomi, S. M. T. and Hashemin, S. S., 1999, A new analytical algorithm and generation of Gaussian quadrature formula for stochastic network. *European Journal of Operational Research*, 114(3), 610-625.
- [12] Fernandez, A. A., Armacost, R. L. and Pet-Edwards, J. J. A., 1996, The role of the nonanticipativity constraint in commercial software for stochastic project scheduling. *Computers and Industrial Engineering*, 31(1-2), 233-236.
- [13] Golenko-Ginzburg, D., 1993, A two-level decision making model for controlling stochastic projects. *International Journal of Production Economics*, 32(1), 117-127.

- [14] Golenko-Ginzburg, D. Gonik, A. and Kesler, S., 1996, Hierarchical decision making model for planning and controlling stochastic projects. *International Journal of Production Economics*, 46-47, 39-54.
- [15] Golenko-Ginzburg, D. and Gonik, A., 1997, Stochastic network project scheduling with non-consumable limited resources. *International Journal of Production Economics*, 48(1), 29-37.
- [16] Golenko-Ginzburg, D. and Gonik, A., 1998, High performance heuristic algorithm for controlling stochastic network projects. *International Journal of Production Economics*, 54(3), 235-245.
- [17] Golenko-Ginzburg, D. and Gonik, A., 1998, A heuristic network project scheduling with random activity duration depending on the resource allocation. *International Journal of Production Economics*, 55, 149-162.
- [18] Golenko-Ginzburg, D., Gonik, A. and Sitniakorsky, S., 2000, Resource supportability model for stochastic network projects under a chance constraint. *Communication in Dependability and Quality Management*, 3, 89-102.
- [19] Gokbayrak, K. and Cassandras, C. G., 2001, Online surrogate problem methodology for stochastic discrete resource allocation problems. *Journal of Optimization Theory and Applications*, 108(2), 349-376.
- [20] Gokbayrak, K. and Cassandras, C. G., 2002, Generalized surrogate problem methodology for online stochastic discrete optimization. *Journal of Optimization Theory and Applications*, 114(1), 97-132.
- [21] Herroelen, W., DeReyck, B. and Demeulemeester, E., 1998, Resource constrained project scheduling: A survey of recent developments. *Computers and Operations Research*, 25(4), 279-302.
- [22] Herroelen, W., Demeulemeester, E. and DeReyck, B., 2001, A note on the paper "Resource-constrained project scheduling: Notation, classification, models and methods" by Brucker et al., *European Journal of Operational Research*, 128(3), 679-688.
- [23] Herroelen, W. and Leus, R., 2003, Project scheduling under uncertainty: survey and research potentials. *European Journal of Operational Research*, 165(2), 289-306.
- [24] Igelmund, G. and Radermacher, F. J., 1983, Preselective strategies for the optimization of stochastic project network under resource constraints. *Networks*, 13, 1-28.
- [25] Igelmund, G. and Radermacher, F. J., 1983, Algorithmic approaches to preselective strategies for stochastic scheduling problems. *Networks*, 13, 29-48.
- [26] Kurtulus, I. and Davis, W., 1982, Multi-project scheduling: categorization of heuristic rules performance. *Management Science*, 28, 161-172.
- [27] Martel, A. and Ouellet, J., 1994, Stochastic allocation of a resource among partially interchangeable activities. *European Journal of Operational Research*, 74, 528-539.
- [28] Moder, J. J., Philips, C. R. and Davis, E. W., 1983, *Project Management with CPM, PERT and Precedence Diagramming*, Third edition, Van Nostrand Reinhold International Company Limited.
- [29] Mohring, R. H., Radermacher, F. J. and Weiss, G., 1984, Stochastic scheduling problems I- general strategies. *Zeitschrift für Operations Research*, 28, 193-260.
- [30] Mohring, R. H., Radermacher, F., J. and Weiss, G., 1985, Stochastic scheduling problems II- set strategies. *Zeitschrift für Operations Research*, 29, 65-104.
- [31] Mohring, R. H. and Stork, F., 2000, Linear preselective policies for stochastic project scheduling. *Mathematical Methods of Operations Research*, 52(3), 501-515.
- [32] Pontrandolfo, P., 2000, Project duration in stochastic network by the PERT-path technique. *International Journal of Project Management*, 18(3), 215-222.
- [33] Tereso, A. P., Madalena, M., Araujo, T. and Elmaghraby, S. E., 2004, Adaptive resource allocation in multimodal activity networks. *International Journal of Production Economics*, 92(1), 1-10.
- [34] Tsai, Y. W. and Gemmill, D. D., 1998, Using tabu search to schedule activities of stochastic resource-constrained projects. *European Journal of Operational Research*, 111(1), 129-141.
- [35] Wan, Y. W., 1994, Resource allocation for a stochastic CPM-type network through perturbation analysis. *European Journal of Operational Research*, 79, 239-248.

Appendix A

- I) The steps of algorithm(G) based on \overline{LS} :
- Step 1.* Transform the original network of project to a network, where the activity splitting has been

defined. Perform this operation according to the descriptions given in Section 3.

Step 2. Generating randomly the activities duration times, the network is simulated M' times and compute:

$$\overline{LS}_j = \frac{\sum_{q=1}^{M'} LS_j^{(q)}}{M'} \quad j = 1, 2, 3, \dots, N.$$

Step 3. Set $q = 1$.

Step 4. Set $T = 0$ and perform the random generation of $t_j, j = 1, 2, 3, \dots, N$.

Step 5. Determine the eligible activity set (EAS), i.e., those activities with all predecessor activities scheduled.

Step 6. From among the members of the current EAS, determine the ordered scheduling set (OSS) of activities, i.e., those ordered according to \overline{LS}_j with smallest value first.

Step 7. Consider the activities in OSS in the order listed and schedule those for which sufficient resources are available for the duration of the activity. As activities are scheduled update the level of resources available. Increase ES_j of those activities-in-process the resource allocated to them, as one unit, and decrease their t_j 's as one unit as well. Take out those activities their t_j 's reach zero, from EAS. Build new EAS. The resources allocated to those activities freed (because the resources are renewable), are added to remaining resources and consider available resources for $T = T + 1$.

Step 8. If EAS is an empty set, stop, record the project completion time for this simulation run, i.e., $T^{(q)}$ and go to step 10. Otherwise, go to step 9.

Step 9. $T = T + 1$ and return to step 5.

Step 10. If $q = M'$, go to step 11. Otherwise, Set $q = q + 1$, return to step 4.

Step 11. Considering $T^{(q)}, q = 1, 2, \dots, M'$ as values of random variables of project completion time, construct the following frequency distribution function table.

L	T_l	f_l	CF_r
1	$T_1 = \underset{q}{\text{Min}}(T^{(q)})$	f_1	CF_1
2	T_2	f_2	CF_2
\vdots	\vdots	\vdots	\vdots
K	$T_k = \underset{q}{\text{Max}}(T^{(q)})$	f_k	CF_k

$$T_{l+1} = T_l + 1$$

$$f_l = \text{Absolute frequency of } T_l$$

$$CF_r = \sum_{l=1}^r f_l.$$

With the above manner, an estimation of cumulative distribution function of project completion time will be resulted.

The steps of algorithm (H) based on $\text{Min } \overline{LF}$ would be the same as the steps of algorithm based on $\text{Min } \overline{LS}$; except that this point activities are ordered according to \overline{LF}_j with smallest values first to determine OSS set in step 6. \overline{LF}_j can be computed by the following equality

$$\overline{LF}_j = \frac{\sum_{q=1}^{M'} LF_j^{(q)}}{M'}. \quad (2)$$

II) The steps of algorithm (I) based on $\text{Min } \hat{LS}$

Step 1. Transform the original network of project to a network, where the activity splitting has been defined. Perform this operation according to the descriptions given in Section 3.

Step 2. Set $q = 1$.

Step 3. Generating randomly the activities duration time, simulate the network and order the activities set according to LS_j with smallest values first. Then store it in an ordered N-tuple with the name of $OSS^{(q)}$.

Step 4. If $q = M'$, go to step 5. Otherwise set $q = q + 1$ and return to step 3.

Step 5. Now we have M' number of ordered N-tuple ($OSS^{(q)}, q = 1, 2, \dots, M'$). Among them, determine the ordered N-tuple with the highest frequency and name it OSS.

Step 6. Set $q = 1$.

Step 7. Set $T = 0$ and perform the random generation of $t_j, j = 1, 2, \dots, N$.

Step 8. Determine the eligible activity set (EAS), i.e., those activities with all predecessor activities scheduled.

Step 9. Order the members of EAS according to the achieved ordered N-tuple of OSS in step 5.

Step 10. Consider the activities in OSS in the order listed and schedule those activities for which sufficient resources are available for the duration of the activity. As activities are scheduled update the level of resources available. Increase ES_j of those activities-in-process the resource allocated to them, as one unit and decrease their t_j 's as one unit. Take out these activities their t_j 's reach zero, from EAS. Build new EAS. The resources allocated to these activities freed (because the resources are renewable), are added to remaining resources and available resources for $T = T + 1$ are considered.

Step 11. If EAS is an empty set, stop; record the project completion time for this simulation run, i.e., $T^{(q)}$ and go to step 13. Otherwise, go to step 12.

Step 12. $T = T + 1$ and return to step 8.

Step 13. If $q = M'$, go to step 14. Otherwise, Set $q = q + 1$, return to step 7.

Step 14. This step is the same as step 11 of algorithm *G*.

The steps of algorithm (*J*) based on $\text{Min } \hat{LF}$ would be the same as the steps of algorithm based on $\text{Min } \hat{LS}$; except that this point activities are ordered according to LF_j with smallest values first to determine $OSS^{(q)}$ in step 3.