Constrained Renewable Resource Allocation in Fuzzy Metagraphs

B.Mirza Masoumzadeh [#1], S.S.Hashemin [*2]

[#] Department of Industrial Engineering, Ardabil Branch, Islamic Azad University, Ardabil, Iran
*Department of Industrial Engineering, Ardabil Branch, Islamic Azad University, Ardabil, Iran

***Abstract*— *Optimal constrained resource allocation causes decrement in the delay of project. In this research, a project has been shown with a Metagraph. It is assumed that the duration of activities is a positive trapezoidal fuzzy number. A certain type of constrained renewable resource is necessary to execute the activities. The aim is to schedule the activities such that, the delay of the project decreases. Latest start times (LS) and latest finish times (LF) have been computed utilizing the forward and backward computation. Then, activities were ordered ascendingly based on the two above-mentioned criteria. A new heuristic algorithm has been developed to allocate the renewable constrained resource. Five examples were solved using the proposed algorithm based on LS and LF criteria. Finally, start time and finish time of the activities were defined as positive trapezoidal fuzzy numbers. Also, completion time of the project was computed.*

Keywords—*fuzzy Metagraph, constrained renewable resource allocation, project completion time*.

I. Introduction

If the constrained resources are allocated to project activities optimally, then projects can be completed in the shortest time. In real world, completion time of activities is not deterministic. So, they can be defined by random variables or fuzzy numbers. When resources are constrained, the completion time of project will increase. Metagraphs were introduced as tools for project planning and control in [1]. Based on forward and backward computation for deterministic metagraphs forward and backward computations were generalized for fuzzy metagraphs in [2]. A method has been developed for non-renewable constrained resource allocation in fuzzy metagraphs in [3]. A method has been innovated for non-renewable constrained resource allocation in fuzzy metagraphs in [4] in which time parameters depend on resources allocation. Non-renewable constrained resource allocation in stochastic metagraphs has been presented in [5]. Renewable constrained resource allocation in fuzzy metagraphs via min slack has been studied in [6]. Renewable constrained resource allocation in fuzzy metagraphs based on min LF has been proposed in [7]. Based on one of the common criteria min LF, a method has been presented for renewable constrained resource allocation in stochastic metagraphs in [8]. In this paper, a new method is developed for renewable constrained resource allocation based on the two common criteria (latest finish time and latest start time). In addition, this method schedules the activities in a way that the delay in the project completion time decreases. Some examples were solved by the new method.

II. Metagraph

A metagraph is identified with $F = (X, E, D)$. $X = x_i, i = 1, \dots, I$ is called the generating set. x_i is called the element of X. $E = e_j, j = 1, 2, \dots, J$ is the set of edges. Each edge is a ordered pair as V_j, W_j. $V_j \subset X$ is called the invertex of e_j and $W_j \subset X$ is called the outvertex of e_j such that $V_j \cap W_j = \varnothing$. It is supposed that the duration of edge e_j is a known positive trapezoidal fuzzy number and represented by d_j. Fig. 1 shows a metagraph.

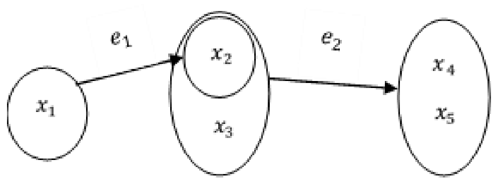

Fig. 1 Metagraph

A. Simple path

An element $x \in X$ is connected to element $x' \in X$ if a sequence of edges exists $e'_k, k = 1, 2, ..., k'$ such that $x \in V'_1$, $x' \in W'_{K'}$ and $W'_k \cap V'_{k+1} \neq \varnothing$. This sequence of edges is called a simple path from x to x'. x is called source and x' is called target. k' is called the length of simple path. In figure 1 $e_1 e_2$ is a simple path from x_1 to x_5 and from x_1 to x_4 .

B. Metapath

A metapath is the set of edges that is shown with $M(B,C)$. B is the source set and C is the target set. B is the set of invertex elements which are not outvertex elements. C is the set of outvertex elements which are not invertex elements.

III. CRITICAL PATH COMPUTATIONS IN FUZZY METAGRAPH

For the first time, critical path computations for fuzzy metagraphs have been developed in [2]. In this paper, First, critical path computations have been presented for deterministic metagraphs. Then, critical path computations have been generalized for fuzzy metagraph as follows:

A. Forward computation

For each element $x_i \in B$, set $\tilde{Q}_i = (0,0,0,0)$ and mark x_i. let $\tilde{Q}_i = (0,0,0,0)$ for all other elements. Let $E = M(B,C)$. while $E = M(B,C) \neq \varnothing$ for each edge e_j, mark all elements in the invertex of e_j and set $ES_j = \max\{Q_i\}$ such that ES_j is the earliest start time of edge e_j. Then, for each $x_k \in W_j$ set $\tilde{Q}_k = \max\{\tilde{Q}_k, (ES_j + \tilde{d}_j)\}$, $E \leftarrow E - e_j$ and mark it.

Repeat the above operations while $E = \varnothing$. Then, $\tilde{T}^e$ (the earliest completion time of metapath) is obtained. Set $\tilde{T}^e = \tilde{T}^L = \max \tilde{Q}_i, x_i \in X$.

B. Backward computation

For each element $x_i \in C$, set $\tilde{L}_i = \tilde{T}^l$ and mark x_i. Let $\tilde{L}_i = \tilde{T}^l$ for all other elements. Let $E_0 = M(B,C)$. While $E_0 \leftarrow M(B,C) \neq \varnothing$ for each edge mark all elements in the outvertex of e_j and set $LF_j = \min\{L_i\}, x_i \in W_j$ such that LF_j is the latest finish time of edge. Then, for each $x_k \in V_j$ set $\tilde{L}_k = \min\{\tilde{L}_k, (LF_j - \tilde{d}_j)\}$, and mark it.

Repeat the above operations while $E_0 = \varnothing$.

So, ordered pair $(\tilde{Q}_i, \tilde{L}_i)$ can be obtained for each element of the metagraph.

In backward computation, the subtraction of two positive trapezoidal fuzzy numbers is computed. This subtraction must be a positive trapezoidal fuzzy number. However, the result of common subtraction for trapezoidal fuzzy numbers may be non-positive. Thus, we apply the other suitable subtraction operator that has been proposed in [9]. This subtraction operator is defined as follows:

$$\tilde{LS}_J = \tilde{LF}_J - \tilde{d}_j = (ls_{j1}, ls_{j2}, ls_{j3}, ls_{j4})$$

$$ls_{j4} = \max(0, \min(lf_{j4} - d_{j4}))$$

$$ls_{j3} = \max(0, \min(ls_{j4}, \min(lf_{j3} - d_{j3})))$$

$$ls_{j2} = \max(0, \min(ls_{j3}, \min(lf_{j2} - d_{j2})))$$

$$ls_{j1} = \max(0, \min(ls_{j2}, \min(lf_{j1} - d_{j1})))$$

Completion time, ES, LS, EF, LF are computed. LS and LF are two common criteria in defining the activities sequence for resource allocation.

Based on the above-mentioned criteria, the activities are ordered ascendingly to solve the next section examples, using the proposed algorithm.

IV. THE PROPOSED ALGORITHM

In real world, the available resources for executing the projects are often limited. Therefore, the activities can not start at the earliest start time (ES).The project can be completed in the shortest time when the available resources are unconstrained. Completion time of the project increases when the available resources are constrained. The purpose is to schedule the activities in such order that the delay of the project decreases.

A. Assumption

A project can be shown with a metagraph in which the execution time of activities is positive trapezoidal fuzzy numbers. It is assumed that the available resources are constrained and renewable.

B. Notations

Notations have been defined in table I.

TABLE I. NOTATIONS

$\tilde{T}$	The current time of the metagraph
RS	Amount of Available resources
s_i	Amount of needed resources for execution of the activity
EAS	Eligible activity set.
OSS	Ordered activity set.
$XOSS$	Subset of OSS activities in which the available resources are enough to start them.
$S_{\tilde{T}}$	Set of activities that finish at $\tilde{T}$
SS	Set of activities from which the members of EAS are chosen.
S	Union of the remaining activities and ongoing activities at each stage
B	Set of ongoing activities that are on different paths.

C. Steps of the algorithm

Step 1. Set $\tilde{T} = \{0,0,0,0\}$

Step2. Determine EAS .

Step 3. Determine OSS .

Step 4. Determine $XOSS$. if $XOSS \neq \varnothing$ then $RS \leftarrow RS - \sum_{i \in xoss} s_i$

Step 5. $\tilde{T} = \min\left(\tilde{T} + \min_{i \in xoss} e_i , \tilde{T}_{b \in B}\right)$

If $XOSS = \varnothing$ then $\tilde{T} \leftarrow \min_{b \in B} \tilde{T}_b$

Step 6. Determine $S_{\tilde{T}}$.

$$RS \leftarrow RS + \sum_{i \in S_{\tilde{T}}} s_i$$

Step 7. $S \leftarrow S - S_{\tilde{T}}$

$$B \leftarrow S - SS$$

Step 8. If $s \neq \varnothing$ then go back to step2, otherwise $\tilde{T}$ is the completion time.

V. EXAMPLES

In these two examples, activities are allocated by new algorithm. They are allocated for minimum resource that project is available to do, to maximum resource that completion time of project equals to completion time of project without constrained resource.

A. *Example1*

A project has been shown by a metagraph in fig. 2 which consists of 5 activities. The project information has been given in table II. The example has been solved using the proposed algorithm when the available resources are $RS = 3$, $RS = 4$, $RS = 5$. Completion time of the project, start and finish times of the activities have been given in table III, table IV and table V.

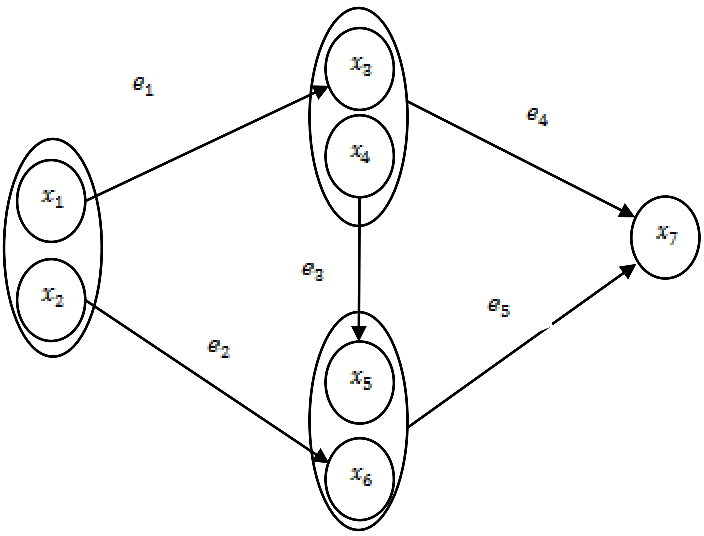

Fig. 2 Metagraph of example 1

TABLE II. INFORMATION oF EXAMPLE

activities	Duration of activities	Needed resources for execution of activities
e_1	(25,28,32,35)	$s_1=1$
e_2	(40,55,65,70)	$s_2=2$
e_3	(20,25,35,40)	$s_3=2$
e_4	(32,37,43,48)	$s_4=2$
e_5	(42,45,55,60)	$s_5=3$
Activities are sorted based on min LF		$e_2 - e_3 - e_1 - e_5 - e_4$
Activities are sorted based on min LS		$e_2 - e_3 - e_4 - e_1 - e_5$

TABLE III. THE RESULTS OF THE EXAMPLE WHEN THE CONSTRAINED RESOURCE IS RS=3

Criteria	*LF*		*LS*	
times / Activities	Start time	Finish time	Start time	Finish time
e_1	(0,0,0,0)	(25.28,32,35)	(0,0,0,0)	(25,28,32,35)
e_2	(0,0,0,0)	(40,55,65,70)	(0,0,0,0)	(40,55,65,70)
e_3	(40,55,65,70)	(60,80,100,110)	(40,55,65,70)	(60,80,100,110)
e_4	(102,125,155.170)	(134,162,198,218)	(60,80,100,110)	(92,117,143,158)
e_5	(60,80,100,110)	(102,125,155,170)	(92,117,143,158)	(134,162,198,218)
Completion time	(134,162,198,218)		(134,162,198,218)	

TABLE IV. THE RESULTS OF THE EXAMPLE WHEN THE CONSTRAINED RESOURCE IS RS=4

Criteria	*LF*		*LS*	
Times / Activities	Start time	finish time	Start time	Finish time
e_1	(0,0,0,0)	(25,28,32,35)	(0,0,0,0)	(25,28,32,35)
e_2	(0,0,0,0)	(40,55,65,70)	(0,0,0,0)	(40,55,65,70)
e_3	(0,0,0,0)	(20,25,35,40)	(0,0,0,0)	(20,25,35,40)
e_4	(20,25,35,40)	(52,62,78,88)	(20,25,35,40)	(52,62,78,88)
e_5	(40,55,65,70)	(82,100,120,130)	(40,55,65,70)	(82,100,120,130)
Completion time	(82,100,120,130)		(82,100,120,130)	

TABLE V. THE RESULTS OF THE EXAMPLE WHEN THE CONSTRAINED RESOURCE IS RS=5

Criteria	*LF*		*LS*	
times / Activities	Start time	Finish time	Start time	Finish time
e_1	(20,25,35,40)	(45,53,67,75)	(20,25,35,40)	(45,53,67,75)
e_2	(0,0,0,0)	(40,55,65,70)	(0,0,0,0)	(40,55,65,70)
e_3	(0,0,0,0)	(20,25,35,40)	(0,0,0,0)	(20,25,35,40)
e_4	(82,100,120,130)	(114,137,163,178)	(82,100,120,130)	(114,137,163,178)
e_5	(40,55,65,70)	(82,100,120,130)	(40,55,65,70)	(82,100,120,130)
Completion time	(114,137,163,178)		(114,137,163,178)	

B. *Example2*

A project has been shown by a metagraph in fig. 3 which consists of 7 activities. The project information has been given in table VI. The example has been solved using the proposed algorithm when the available resources are $RS = 4$, $RS = 5$, $RS = 6$, $RS = 7$. Completion time of the project, start and finish times of activities have been given in table VII, table VIII, table IX and table X.

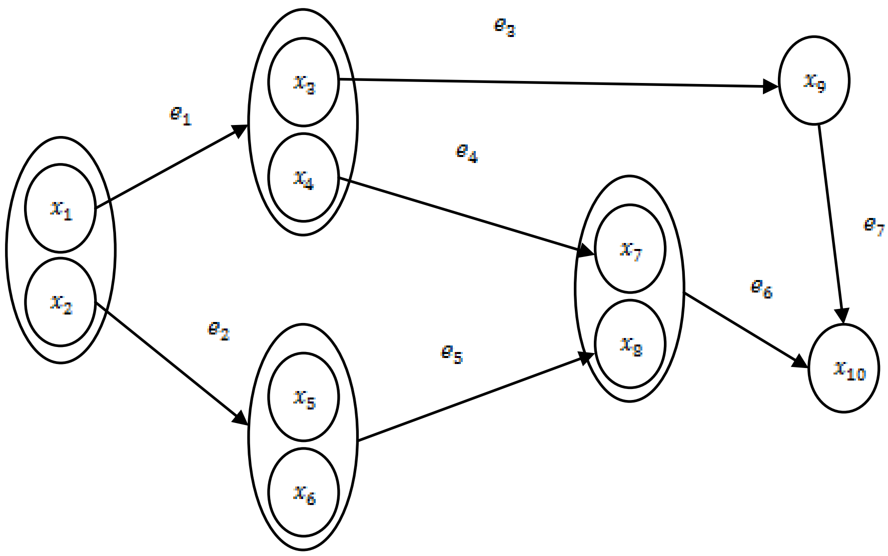

Fig. 3 Metagraph of example2

TABLE VI. INFORMATION OF EXAMPLE2

activties	Duration of activities	Needed resource for execution of the activities
e_1	(25,28,32,35)	$s_1=2$
e_2	(40,55,65,70)	$s_2=2$
e_3	(35,38,42,45)	$s_3=3$
e_4	(32,37,43,48)	$s_4=2$
e_5	(20,25,35,40)	$s_5=1$
e_6	(60,65,75,85)	$s_6=4$
e_7	(65,78,85,90)	$s_7=2$
Activities are sorted based on min LF		$e_1-e_2-e_3-e_4-e_5-e_6-e_7$
Activities are sorted based on min LS		$e_2-e_1-e_3-e_4-e_5-e_7-e_6$

TABLE VII. THE RESULTS OF THE EXAMPLE WHEN THE CONSTRAINED RESOURCE IS RS=4

Criteria	LF		LS	
Times / Activities	Start time	Finish time	Start time	Finish time
e_1	(0,0,0,0)	(25,28,32,35)	(0,0,0,0)	(45,53,67,75)
e_2	(0,0,0,0)	(40,55,65,70)	(0,0,0,0)	(40,55,65,70)
e_3	(57,65,75,83)	(92,103,117,128)	(57,65,75,83)	(92,103,117,128)
e_4	(25,28,32,35)	(57,65,75,83)	(25,28,32,35)	(57,65,75,83)
e_5	(40,55,65,70)	(60,75,100,110)	(40,55,65,70)	(60,75,100,110)
e_6	(92,103,117,128)	(152,168,192,213)	(157,181,202,218)	(217,246,277,303)
e_7	(152,168,192,213)	(217,246,277,303)	(92,103,117,128)	(157,181,202,218)
Completion time	(217,246,277,303)		(217,246,277,303)	

TABLE VIII. THE RESULTS OF THE EXAMPLE WHEN THE CONSTRAINED RESOURCE IS RS=5

criteria	*LF*		*LS*	
times / Activities	Start time	Finish time	Start time	Finish time
e_1	(0,0,0,0)	(25,28,32,35)	(0,0,0,0)	(45,53,67,75)
e_2	(0,0,0,0)	(40,55,65,70)	(0,0,0,0)	(40,55,65,70)
e_3	(25,28,32,35)	(60,66,74,80)	(25,28,32,35)	(60,66,74,80)
e_4	(40,55,65,70)	(72,92,108,118)	(40,55,65,70)	(72,92,108,118)
e_5	(60,66,74,80)	(80,91,109,120)	(60,66,74,80)	(80,91,109,120)
e_6	(125,144,159,170)	(185,209,234,255)	(125,144,159,170	(185,209,234,255)
e_7	(60,66,74,80)	(125,144,159,170)	(60,66,74,80)	(125,144,159,170)
Completion time	(185,209,234,255)		(185,209,234,255)	

TABLE IX. THE RESULTS OF THE EXAMPLE WHEN THE CONSTRAINED RESOURCE IS RS=6

criteria	*LF*		*LS*	
times / Activities	Start time	Finish time	Start time	Finish time
e_1	(0,0,0,0)	(25,28,32,35)	(0,0,0,0)	(45,53,67,75)
e_2	(0,0,0,0)	(40,55,65,70)	(0,0,0,0)	(40,55,65,70)
e_3	(25,28,32,35)	(60,66,74,80)	(25,28,32,35)	(60,66,74,80)
e_4	(40,55,65,70)	(72,92,108,118)	(40,55,65,70)	(72,92,108,118)
e_5	(40,55,65,70)	(60,80,100,110)	(40,55,65,70)	(60,80,100,110)
e_6	(72,92,108,118)	(132,157,183,203)	(72,92,108,118)	(132,157,183,203)
e_7	(60,66,74,80)	(125,144,159,170)	(60,66,74,80)	(125,144,159,170)
Completion time	(132,157,183,203)		(132,157,183,203)	

TABLE X. THE RESULTS OF THE EXAMPLE WHEN THE CONSTRAINED RESOURCE IS RS=7

criteria	*LF*		*LS*	
times / Activities	Start time	Finish time	Start time	Finish time
e_1	(0,0,0,0)	(25,28,32,35)	(0,0,0,0)	(45,53,67,75)
e_2	(0,0,0,0)	(40,55,65,70)	(0,0,0,0)	(40,55,65,70)
e_3	(25,28,32,35)	(60,66,74,80)	(25,28,32,35)	(60,66,74,80)
e_4	(25,28,32,35)	(57,65,75,83)	(25,28,32,35)	(57,65,75,83)
e_5	(40,55,65,70)	(60,80,100,110)	(40,55,65,70)	(60,80,100,110)
e_6	(60,80,100,110)	(120,145,175,195)	(60,80,100,110)	(120,145,175,195)
e_1	(60,66,74,80)	(125,144,159,170)	(60,66,74,80)	(125,144,159,170)
Completion time	(120,145,175,195)		(120,145,175,195)	

VI. CONCLUSIONS AND RECOMMENDATIONS

A new algorithm has been innovated for renewable constrained resource allocation in fuzzy metagraph. The activities are ascended based on min LF and min LS criteria for solving the examples. The algorithm is flexible; it can solve the problem using the arbitrary criterion and arbitrary ranking method. Completion time of the project is computed as a trapezoidal fuzzy number.

This algorithm can be generalized to allocate the several kinds of constrained resources. Also, it can be generalized to allocate the several kinds of constrained resources when some of them are renewable and others are non-renewable.

REFERENCES

[1] Basu, A., & Blanning, R. W. (1992, December). Metagraphs and Petri nets in model management. In *Proc. Workshop Inform. Tech Systems* (Vol. 64, p. 73).

[2] Hashemin S. S., (2008) Computing of completion Time for Fuzzy Metagraphs, *2nd Joint Congress on Fuzzy and Intelligent Systems*, Iran, Tehran.

[3] Hashemin S. S., (2010) Constrained Resource Allocation in Fuzzy metagraph, *10th Iranian Conference on Fuzzy Systems*, Shahid Beheshti University, 13-15.

[4] Vahedi, H. & Hshemin, S. S. (2015). A New Method for Allocation of Constrained Non-Renewable Resource in Fuzzy Metagraphs, *International Journal of Industrial Engineering*, 2(3), 14-19.

[5] Akhound Shiviyari, N. & Hashemin, S. S. (2017). Constrained Consumable Resource Allocation in Stochastic Metagraphs with Discrete Random Times, *International Journal of Industrial Engineering*, 4(4), 6-10.

[6] Hashemin, S. S. (2011). Constrained Renewable Resource Allocation in Fuzzy Metagraphs via Min-Slack. *International Journal of Applied*, 1(1), 7-17.

[7] Hashemin, S. S. (2012). Application of Fuzzy Latest Finish Time for Constrained Renewable Resource Allocation in Fuzzy Metagraph,*8th International Symposium Statistics Anadolu university*.

[8] Hakimzadeh, E. & Hashemin, S. S. (2017). Constrained Renewable Resource Allocation in Stochastic Metagraphs, *International Journal of Industrial Engineering*, 4(4), 1-5.

[9] Soltani, A., & Haji, R. (2007). A Project Scheduling Method Based on Fuzzy Theory. *Journal of Industrial and SystemEngineering*,1(1),70-80.