Location and Job Shop Scheduling Problem with Minimizing Maximum Completion Time Goal in Fuzzy Environment Simultaneously Through Ranking Fuzzy Numbers and α-cut Fuzzy

Mohsen Dastpak
Department of Financial Engineering
University of Economic Sciences, Tehran, Iran

Saeed Poormoaied
Department of Industrial Engineering
Ferdowsi University of Mashhad, Iran

Masoud Naderpoor
Department of Computer Engineering
Ferdowsi University of Mashhad, Iran

Abstract

In this paper we consider job shop scheduling problem (JSSP) and quadratic assignment problem (QAP) simultaneously. We assume that each job has several fuzzy parameters such as due date, products transportation time, Processing time, and set-up time. Our objective is to specify the location of machines and the scheduling of jobs so that minimize both maximum completion time and transportation cost, simultaneously. We name this new problem Fuzzy Scheduling and Location Problem in the Job shop Environment (FSLPJE). To solve FSLPJE, we utilize genetic algorithm and simulation annealing approaches as well as ranking fuzzy presented by Jing-Shing Yao.

Keywords
JSSP, QAP, Fuzzy parameters, Completion time, Transportation time, Genetic algorithm, Simulation annealing, ranking fuzzy number

1. Introduction
Nowadays, along with progress of technology, big industrials tend to be converted into private and little industrials. Problems such as productivity and production management are considerable in competitive environment in the real world. Among all parameters, production time of products is a very important parameter for successfulness so that we can find important function of scheduling in productivity environment. Job shop scheduling problem (JSSP) is a very important problem in production planning due to much similarity among the assumptions of JSSP and the real industrial environments. In JSSP, there are m machines and n jobs. Each job has got a particular processing and sequencing for production that is predetermined. Each operation of each job has a particular processing time. The aim of JSSP is specifying the sequencing of jobs on machines so that criteria performance such as make span, penalty of tardiness and be minimized. It is a long time that JSSP has appeared; JSSP is one the most famous and hardest combinational problems (Garey 1976). JSSP is important in aspect of production management and combinational optimization and many researchers are attracted to JSSP for decade. JSSP was introduced as criteria for evaluation of new optimization approach in 1960. Along this time, many papers about JSSP published and a lot of methods for solving JSSP introduced that one of the most successful methods was Dileep's method (Dileep, 1997), but an effective algorithm that can solve JSSP in polynomial time have not yet been found, so that problems with 10 machines and 10 jobs wasn't solved for one-fourth century until 1986, and nowadays problems with 20 machines and 20 jobs have not yet been solved.

Many researchers who dealt with JSSP [3, 4, 5, 6, 6, 8] assumed that time parameters, i.e. processing times, are fixed and deterministic. This assumption may be realistic if the operations under considerations are fully automated. However, whenever there is human interaction, this assumption may present difficulties in applying the schedule or even invalidating it. Recently, researchers start to address the uncertainty of the data in the real world (i.e. processing times, due dates) and use fuzzy numbers to address this uncertainty. The first significant application that considers the uncertainty in time parameters is Fortemps (1997). In this application the author used six-point fuzzy numbers to represent fuzzy durations. He used simulated annealing (SA) as an optimization technique and the optimization criterion was minimizing the fuzzy make span (FM). To test the approach, he fuzzified the FT 6 × 6 problem (Fisher 1963) and other famous problems such as La11, La12, La13, and La14 (Lawrence 1984). The produced solutions are flexible, since they are able to cope with all possible durations within the specified range. (Sakawa and Kubota 2001) presented a two-objective genetic algorithm to minimize the maximum fuzzy completion time and maximize the average agreement index. (Tsujimura and et.al 1995) present job shop scheduling with fuzzy processing time using Gas. (Sakawa and Mori 1997) proposed a job shop scheduling with both fuzzy processing time and fuzzy due date through GAs incorporating the concept of similarity among individuals have been proposed. (Ghrayeb 2000) presented a genetic algorithm approach to optimizing fuzzy JSSP, in which imprecise processing times are modeled as triangular fuzzy numbers (TFNs). This approach relies on using three-point fuzzy numbers to represent the imprecision in processing times. In that paper, the strength of his approach is that the produced schedule is flexible; it stays valid and can cope with all possible durations within the specified ranges. (Sakawa and Kubota 2001) in your paper considered the imprecise or fuzzy nature of the data in real-world problems. They introduced job shop scheduling with fuzzy processing time by a triangular fuzzy number (TFN) and fuzzy due date by a doublet number and they proposed a genetic algorithm for solving the formulated problems.

Assignment problem (AP) is a one of the most applicable problem in both manufacturing and service systems by which we can assign *n* jobs to *n* workers so that total assignment costs are minimized. After present AP, this problem was changed and was extended, one of this changing was formulation of AP with fuzzy parameters (Bogomolnaia 2001 and Lin 2004). Quadratic assignment problem (QAP) as a special kind of AP is a considerable problem for years. In QAP, each machine can locate in each place similar to AP (Lawler 1963). The number of machines is equal to the number of places; on the other hand, each job has predetermined sequencing on machines.
Ranking of fuzzy numbers is not a simple process. Unlike real numbers, fuzzy numbers have no natural order. Fuzzy numbers are not in linear order; they are usually in partial order. Since the study of fuzzy ranking began, various ranking methods that yield a totally ordered set have been proposed. However, there is no best method agreed. All the proposed ranking methods have advantages as well as disadvantages. In this paper, for ranking the fuzzy number $\tilde{C}_{max}$, we utilize the method presented by Jing-Shing Yao and Wu (2000).

Literature view represents that QAP and JSSP are considered by researchers independently. In this paper, we present a new model named Fuzzy Scheduling and Location Problem in the Job-shop Environment (FSLPJE) that merges QAP and JSSP. This paper is organized as follows: In Section 2, we shortly describe the Problem definition. Then, the mathematical formulation of FSLPJE is presented in Section 3. The proposed approach for solving fuzzified problem is described in Section 4. In section 5, we present a numerical example. In Section 6, genetic algorithm and simulated annealing are described. Finally, the last section deals with concluding remarks and shows the efficiency of proposed model.

2. Problem Definition

Assume that there are *n* jobs that each job has a series of predetermined operation, i.e. the operation sequencing on each job is predetermined. Each job only process on each machine once. We have *m* machine that operate the operation that are related to jobs. Each machine operates only one operation; on the other hand, there are *m* place for locating machines. The place of depot and machines are predetermined and fix. At first we assume that all of jobs are in depot. The completion time is varies for each job .i.e. each job have only one completion time. The aim of us is specifying the scheduling of jobs and location of machines, simultaneously so that maximum completion time is minimized.

2.1 Problem assumptions

The assumptions of this problem are as following: 1- Each machine process only one job at a time. 2- At first all of jobs are ready. 3- Transportation time between two machines is depended to distance between them. Transportation isn't automatically, so transportation time is variable parameters, therefore we consider it as triangular fuzzy number. 4- Set up time for operation each job is variable, so we consider it as triangular fuzzy number. 5- Disruption of jobs is not permitted. 6- Processing time is variable, so we consider it as triangular fuzzy number. 7- Each job can is located in every place.

2-2. Parameters

$\tilde{P}_{Ki}$: Processing time of job *i* on machine *k*. $\tilde{s}_{ijk}$: Set up time of job *i* to *j* on machine *k*. i_{last}: Last operation of job *i*. n: Quantity of jobs. m: Quantity of machines. As each machine operates only one operation, so quantity of operation is the same quantity of machines. Therefore Index of operation demonstrates machines. As processing time, set up time and transportation time are fuzzy parameters, so start time is fuzzy parameter as well. z_{kr}: If machine *k* is assigned to place *r*, z_{kr} is equal to 1 else 0. X_{kij}: If job *i* is operated earlier than job *j* on machine *k*, X_{kij} is equal to 1 else 0. $\tilde{C}_i$: Completion time of job *i* that is as follows: $\tilde{C}_i = \tilde{y}_{i\ last,i} + \tilde{P}_{i\ last,i} \quad ; \quad \forall\, i_{last}$
AS processing time is a fuzzy parameter, completion time is fuzzy parameter as well.

3. Fuzzy Preliminaries

Fuzzy set theory has emerged as a powerful tool to quantitatively represent and manipulate the imprecision that sometimes governs the decision-making process. Fuzzy sets or fuzzy numbers can be used to encounter the imprecision by setting the values of the input parameters to be functions of triangular or trapezoidal shapes. Some basic definitions, taken from [7, 16], that are related to fuzzy set theory are briefly reviewed below for the interest of the readers.

Definition 1: $\tilde{A}$ is a fuzzy set in a universe of discourse X. It is characterized by a membership function, $\mu_{\tilde{A}}(x)$, which is associated with each element x, where x is a real number in the interval [0, 1]. The function value $\mu_{\tilde{A}}(x)$ is termed as the grade of membership of x in $\tilde{A}$**.**

Definition 2: The fuzzy set $\tilde{A}$ of the universe of discourse X is convex, where $\mu_{\tilde{A}}(\lambda x_1 + (1-\lambda)x_2) \geq \min\big(\mu_{\tilde{A}}(x_1), \mu_{\tilde{A}}(x_2)\big)$ for all $x_1, x_2 \in$ X and for $\lambda \in [0, 1]$.

Definition 3: The fuzzy set $\tilde{A}$ of the universe of discourse X is called a normal fuzzy set when $\exists \in x_i$ X, $\mu_{\tilde{A}}(x_i) = 1$.

Definition 4: A fuzzy number is a fuzzy subset in the universe of discourse X that is both convex and normal. $\tilde{A}$ is said to be a trapezoidal fuzzy number represented by the crisp numbers (a_1,a_2,a_3,a_4), where $a_1 < a_2 < a_3 < a_4$, when its membership function is denoted as:

$$\mu_{\tilde{A}}(x) = \begin{cases} \dfrac{x-a_1}{a_2-a_1} & , \ a_1 \leq x \leq a_2 \\ 1 & , \ a_2 \leq x \leq a_3 \\ \dfrac{x-a_4}{a_3-a_4} & , \ a_3 \leq x \leq a_4 \\ 0 & \ otherwise \end{cases} \qquad (1)$$

When $a_2 = a_3$, the trapezoidal fuzzy number described in Eq. (1) becomes a triangular, which is a special case of the first. In this paper, the Function Principle Method, from [4], is used to simplify the model calculations. Now, define $\tilde{A} = (a_1, a_2, a_3, a_4)$ and $\tilde{B} = (b_1, b_2, b_3, b_4)$ as two trapezoidal fuzzy numbers with the following properties:

1. $\tilde{A} + \tilde{B} = (a_1 + b_1, a_2 + b_2, a_3 + b_3, a_4 + b_4)$.

If a_1, a_2, a_3, a_4 and b_1, b_2, b_3, b_4 are all positive real numbers, then: 1. $\tilde{A}.\tilde{B} = (a_1.b_1, a_2.b_2, a_3.b_3, a_4.b_4)$.
Let λ be a real number, then for $\lambda \geq 0$, $\lambda\tilde{A} = (\lambda a_1, \lambda a_2, \lambda a_3, \lambda a_4)$ and $\lambda < 0$, $\lambda\tilde{A} = (\lambda a_4, \lambda a_3, \lambda a_2, \lambda a_1)$.
2. $-\tilde{B} = (-b_4, -b_3, -b_2, -b_1)$, $\tilde{A} - \tilde{B} = (a_1 - b_4, a_2 - b_3, a_3 - b_2, a_4 - b_1)$.

If a_1, a_2, a_3, a_4 and b_1, b_2, b_3, b_4 are all positive real numbers, then 3. $\frac{1}{B}=\left(\frac{1}{b_4},\frac{1}{b_3},\frac{1}{b_2},\frac{1}{b_1}\right)$, $\frac{A}{B}=\left(\frac{a_1}{b_4},\frac{a_2}{b_3},\frac{a_3}{b_2},\frac{a_4}{b_1}\right)$. The presented principle also holds for trapezoidal fuzzy number.

4. FSLPJE Modeling

We have *n* jobs that must be operated on *m* machines, and there are *m* places for locating machines. The place of depot is fixed and predetermined. We assume that at first all of jobs are in depot. Each job is demonstrated with index *i* (*i*=1, 2… *n*) and each job has predetermined operation (O_{ki}), and *k* is the index of machines (*k*=1, 2… *m*). For example, O_{23} means the operation of job 3 that is operated on machine 2. In addition, each job has a specific path for operating; this path is dnoted as A_i.

Processing time of job *i* on machine *k* is denoted as $\tilde{P}_{Ki}$; it must be note that $\tilde{P}_{0i}=0$, *i*=1… *n*. Transportation time consists of the transportation between depot and machine as well as between two machines. This time depends on the place of machines. For example, if we assume that job *i* must be carried between machine *k* and *l*, and these machines are located in places *r* and *s*; the transportation time between two places is denoted as $\tilde{t}_{rs}$. The problem objective is the total cost minimizing; the cost includes the transportation cost. Model 1 is the mathematical modeling of FJSSLP.

Model 1:

$\min \tilde{C}_{Max}$

Subject to:

$$\tilde{C}_{max} \geq \tilde{y}_{i\ last\ ,i} + \tilde{p}_{i\ last\ ,i} \quad , \qquad \forall\, i_{last} \tag{1-1}$$

$$\sum_{k=0}^{m} Z_{kr} = 1 \quad ; \quad r = 0,1,\ldots,m \tag{2-1}$$

$$\sum_{r=0}^{m} Z_{kr} = 1 \quad ; \quad k = 0,1,\ldots,m \tag{3-1}$$

$\forall\,(k,i) \to (l,i) \in A_i \quad ; \quad i = 1,\ldots,m$ and $k,l,r,s = 0,1,\ldots,m \;; r \neq s\,, r < s\,, l \neq k$:

$$\tilde{y}_{li} \geq \tilde{y}_{ki} + \tilde{P}_{ki} + \tilde{t}_{rs} - M\big((1-Z_{kr}) + (1-Z_{ls})\big) \tag{4-1}$$

$$\tilde{y}_{kj} \geq \tilde{y}_{ki} + \tilde{P}_{ki} + \tilde{S}_{ijk} - M\big(1-X_{kij}\big) \quad ; \quad \forall (K,i),(K,j) \;;\; k = 1,\ldots,m\,;\; i,j = 0,1,\ldots,n \tag{5-1}$$

$$\tilde{y}_{ki} \geq \tilde{y}_{kj} + \tilde{P}_{kj} + \tilde{S}_{jik} - M\, X_{kij} \quad ; \quad \forall (K,i),(K,j) \;;\; k = 1,\ldots,m \;;\; i,j = 0,1,\ldots,n \tag{6-1}$$

$$Z_{00} = 1 \tag{7-1}$$

$$\tilde{y}_{0i} = 0 \quad ; \quad i = 1,\ldots,n \tag{8-1}$$

$$\tilde{y}_{ij} \in R^{+} \quad ; \quad i,j = 1,\ldots,n \tag{9-1}$$

$$Z_{kr} \in \{0,1\} \quad ; \quad k,r = 0,1,\ldots,m \tag{10-1}$$

$$X_{kij} \in \{0,1\} \;;\; \forall (K,i),(K,j) \quad ; \quad k = 1,\ldots,m \;;\; i,j = 0,1,\ldots,n \tag{11-1}$$

In Eq (4-1), (5-1), (6-1), M is a big number. In this model, objective function minimizes maximum completion time. This object is formulated as $\min \tilde{C}_{Max}$. Eq (2-1), that demonstrate each machine must be located one palce. Eq (3-1), demonstrate that each machine assigns to only one place. Eq (4-1), express that if job *i* processed on machine l before machine *k*, start time of job *i* on machine l must be more than start time this job on machine *k* plus processing time of that job on machine *k* plus transportation time between machine *k* and *l*. In addition, we assume that process sequencing demonstrated as (k,i)$\rightarrow$(l,i) where $\rightarrow$ present sequencing. Eq (5-1) and (6-1), demonstrate that two jobs cannot process on one machine simultaneously, i.e. each machine process one job at a time. Eq (7-1), sure that place of depot is fixed. Eq (8-1), indicate that start time all jobs are fixed in depot, i.e. at first all jobs are in depot and but isn't processed by any machine. Eq (9-1), express $\tilde{y}_{ij}$ are nonnegative. Eq (10-1) and (11-1) demonstrate that Z_{kr} and X_{kij} are 0 or 1.

5. Solving Method (The Ranking System of Fuzzy Numbers on R)

Yao and Wu [21] used signed distance to define ranking of fuzzy numbers. The signed distance used for fuzzy numbers has some similar properties to the properties induced by the signed distance in real numbers. The signed distance method for ranking fuzzy numbers can be explained briefly as follows:

Let F be the family of the fuzzy numbers on R. The sign distance is defined as d*(a, 0) = a on R. Then for a, b ∈ R, d*(a, b) = a − b. For $\tilde{D}$, $\tilde{E}$ ∈ F, with α-cut ($0 \leq a \leq 1$), there is a closed interval $D(\alpha) = [D_L(\alpha), D_R(\alpha)]$. Then, the signed distance of $\tilde{D}$, $\tilde{E}$ is defined as [21]: $d(\tilde{D}, \tilde{E}) = \frac{1}{2}\int_0^1 [D_L(\alpha), D_R(\alpha) - E_L(\alpha) - E_R(\alpha)]d\alpha$

It can be proved that d is an extension of d*. And:

$d(\tilde{D}, \tilde{E}) > 0$ iff $d(\tilde{D},0) > d(\tilde{E},0)$ iff $\tilde{E} < \tilde{D}$

$d(\tilde{D}, \tilde{E}) < 0$ iff $d(\tilde{D},0) < d(\tilde{E},0)$ iff $\tilde{D} < \tilde{E}$

$d(\tilde{D}, \tilde{E}) = 0$ iff $d(\tilde{D},0) = d(\tilde{E},0)$ iff $\tilde{E} \approx \tilde{D}$

According to these definitions, the signed distance of a triangular fuzzy number $\tilde{A} = (\underline{a}, a, \overline{a})$ is defined as:

$d(\tilde{A}, 0) = \frac{1}{2}\int_0^1 [\underline{a} + (a - \underline{a})\alpha + \overline{a} - (\overline{a} - a)\alpha]d\alpha = \frac{1}{4}(2a + \underline{a} + \overline{a})$

The signed distance of a trapezoidal fuzzy number $\tilde{A} = (\underline{a}, a_1, a_2, \overline{a})$ is defined as

$d(\tilde{A}, 0) = \frac{1}{2}\int_0^1 [\underline{a} + (a_1 - \underline{a})\alpha + \overline{a} - (\overline{a} - a_2)\alpha]d\alpha = \frac{1}{4}(a_1 + a_2 + \overline{a} + \underline{a})$

Let $\tilde{A}$ and $\tilde{B}$ are two triangular or trapezoidal fuzzy numbers, their ranking relation is defined as:

$\tilde{A} \leq \tilde{B} \Leftrightarrow d(\tilde{A},0) \leq d(\tilde{B},0)$.

In this study, we use signed distance to define ordering, which is easier to catch upon its meaning. This signed distance we use here has some similar properties to the properties induced by the signed distance in real numbers. We discuss many properties concerning ordering, we can compare the two signed distances of two fuzzy numbers, therefore the method is feasible for minimizing maximum completion time because this method compares two signed distances of two maximum completion time and selects minimum maximum completion time.

6. Numerical Example

As illustrative numerical examples, consider 6×6 utilized for solving proposed model. And assume that we have six jobs and six machines that can process jobs. At first all of jobs are in depot. The objective is specifying machine lactation and jobs sequencing on machines so that maximum completion time be minimized. Note that place 0 and machine 0 are the same depot.

Table 1: Problem 1 of 6×6 MOFJSSP

	Processing machines (fuzzy processing time)					
Job 1	4 (9, 13, 17)	3 (6, 9, 12)	1 (10, 11, 13)	5 (5, 8, 11)	2 (10, 14, 17)	6 (9, 11, 15)
Job 2	4 (5, 8, 9)	2 (7, 8, 10)	5 (3, 4, 5)	3 (3, 5, 6)	1 (10, 14, 17)	6 (4, 7, 10)
Job 3	5 (3, 5, 6)	4 (3, 4, 5)	3 (2, 4, 6)	1 (5, 8, 11)	2 (3, 5, 6)	6 (1, 3, 4)
Job 4	6 (8, 11, 14)	3 (5, 8, 10)	1 (9, 13, 17)	4 (8, 12, 13)	2 (10, 12, 13)	5 (3, 5, 7)
Job 5	3 (8, 12, 13)	5 (6, 9, 11)	6 (10, 13, 18)	2 (4, 6, 8)	1 (3, 5, 7)	4 (4, 7, 9)
Job 6	2 (8, 10, 13)	4 (8, 9, 10)	6 (6, 9, 12)	3 (1, 3, 4)	5 (3, 4, 5)	1 (2, 4, 6)

Table 2: Transportation time between place r and s ($\tilde{t}_{rs}$) – (unit of time)

	Place 0	Place 1	Place 2	Place 3	Place 4	Place 5	Place 6
Place 0	0	(2.4,4.3,5.4)	(2.1,4,4.5)	(3,4,4.5)	(2.3,4,5.1)	(1.3,2,4.1)	(2.1,3.5,4.1)
Place 1	(2.4,3.3,4.4)	0	(1.3,2.2,4)	(2.3,4,5.1)	(2,3,4)	(2.4,4.2,5)	(2.3,4,5.1)
Place 2	(1.3,3,5.1)	(2.3,4,5.1)	0	(2,3,4)	(1.3,2,4.1)	(1.5,2,4.3)	(2.4,4.3,5.4)

Place 3	(2.4,4.3,5.4)	(1.3,2,4.1)	(2.3,4,5.1)	0	(2.1,3.5,4.1)	(2.3,4,5.1)	(2.3,4,5.1)
Place 4	(2.1,4.2,4.9)	(2.3,4,5.1)	(1.3,2,4.1)	(2.3,4,5.1)	0	(1.1,2,4)	(2,3,4)
Place 5	(2.3,4,5.1)	(2.1,3.5,4.1)	(2,3,4)	(2.3,4,5.1)	(2.3,4,5.1)	0	(2.3,3,4.1)
Place 6	(1,2,4)	(2,4,5)	(2.2,4,4.4)	(2.3,4,5.1)	(2,4.1,5)	(2.4,4.3,5.4)	0

Table 3: Set-up time from job i to j on each machine ($\tilde{S}_{ij1}$) – (unit of time)

		Job 1	Job 2	Job 3	Job 4	Job 5	Job 6
Mach 1	Job 1	0	(1,2,3.1)	(1.3,2,3.1)	(2,3,3.3)	(1.3,2,3.1)	(2.1,3,4.1)
	Job 2	(2,3.1,4.1)	0	(2.2,4.1,5)	(3,4,4.4)	(2.3,4,5.1)	(1.1,2,3.1)
	Job 3	(1.4,2.5,3)	(1.3,2,3.1)	0	(2.3,4,5.1)	(2,3,3.5)	(3,4,5.1)
	Job 4	(2,3,4)	(2.2,4.1,5)	(2.3,4,5.1)	0	(1,3,4.1)	(2.3,4,5.1)
	Job 5	(2,3,3.5)	(3,4,5.1)	(2,4.1,5)	(3,4,5.1)	0	(1.3,2,3.1)
	Job 6	(1,3,4.1)	(2,4,5)	(2.3,4,5)	(2,4,5)	(2.3,4.5,5)	0
Mach 2	Job 1	0	(1,3,4.1)	(1.3,2,3.1)	(2,2.9,3.3)	(1.3,2,3.1)	(1.1,2,3.3)
	Job 2	(2,3.1,4.1)	0	(2.2,4.1,5)	(3,4,5.1)	(2.2,4,4.5)	(1.1,2,3.1)
	Job 3	(2,3,3,4.1)	(1.3,2,3)	0	(2,4,5)	(2,3.2,3.5)	(3,4,5.1)
	Job 4	(2,4,4.6)	(2,4.1,5)	(2.3,4,5.1)	0	(1,3,4)	(2.3,4,5.1)
	Job 5	(2,3.1,3.5)	(3,4.1,5.1)	(2,3,3,4.1)	(3.1,4,5.1)	0	(2.1,3,4.1)
	Job 6	(2,3,4.1)	(2,4.1,5)	(2,2.5,3)	(2.2,4.1,5)	(2,4,5)	0
Mach 3	Job 1	0	(1,2,3.1)	(1.3,2.5,3.1)	(2,3,3.3)	(1.3,2,3.1)	(2.1,3,4.1)
	Job 2	(2,3.1,4.1)	0	(2.2,4.1,5)	(3,4,4.4)	(2.3,4,5.1)	(1.1,2,3.1)
	Job 3	(1.1,2.2,3.1)	(1.3,2,3.1)	0	(2.3,4,5.1)	(2,3,3.5)	(3,4,5.1)
	Job 4	(2.3,4,5.1)	(2.2,4.1,5)	(2.3,4,5.1)	0	(1,3,4.1)	(2.3,4,5.1)
	Job 5	(2,3,3.5)	(3,4,5.1)	(2,4,5)	(3,4,5.1)	0	(1.3,2,3.1)
	Job 6	(1,3,4.1)	(2,4,5)	(1,2,3)	(2,4,5)	(2.3,4.3,5)	0
Mach 4	Job 1	0	(1,2,3.1)	(1.3,2,3.1)	(2,3,3.3)	(1.3,2,3.1)	(2.1,3,4.1)
	Job 2	(2,3.1,4.1)	0	(2.2,4.1,5)	(3,4,4.4)	(2.3,4,5.1)	(1.1,2,3.1)
	Job 3	(1.4,2.5,3)	(1.3,2,3.1)	0	(2.3,4,5.1)	(2,3,3.5)	(3,4,5.1)
	Job 4	(2,3,4)	(2.2,4,5.1)	(2.3,4,5.1)	0	(1,3,4.1)	(2.3,4,5.1)
	Job 5	(2,3,3.5)	(3,4,5.1)	(2,4.1,5)	(3,4,5.1)	0	(1.3,2,3.1)
	Job 6	(1,3,4.1)	(2,4,5)	(2.3, 4, 5)	(2,4,5)	(2.3,4.5,5)	0
Mach 5	Job 1	0	(1,3, 4.1)	(1.3,2,3.1)	(2, 2.9, 3.3)	(1.3,2,3.1)	(1.1,2,3.3)
	Job 2	(2,3.1,4.1)	0	(2.2,4.1,5)	(3,4,5.1)	(2.2,4,4.5)	(1.1,2,3.1)
	Job 3	(2,3,3,4.1)	(1.3,2,3)	0	(2, 4, 5)	(2, 3.2, 3.5)	(3,4,5.1)
	Job 4	(2,4,4.6)	(2,4.1,5)	(2.3,4,5.1)	0	(1,3,4)	(2.3,4,5.1)
	Job 5	(2,3.1,3.5)	(3, 4.1, 5.1)	(2,3, 3, 4.1)	(3.1,4,5.1)	0	(2.1, 3, 4.1)
	Job 6	(2,3,4.1)	(2, 4.1, 5)	(2, 2.5, 3)	(2.2, 4.1, 5)	(2, 4, 5)	0
Mach 6	Job 1	0	(1,2, 3.1)	(1.3,2.5,3.1)	(2, 3, 3.3)	(1.3,2,3.1)	(2.1,3,4.1)
	Job 2	(2,3.1,4.1)	0	(2.2,4.1,5)	(3,4,4.4)	(2.3,4,5.1)	(1.1,2,3.1)
	Job 3	(1.1,2.2, 3)	(1.3,2,3.1)	0	(2.3, 4, 5.1)	(2, 3, 3.5)	(3,4,5.1)
	Job 4	(2.3,4,5.1)	(2.2,4.1,5)	(2.3,4,5.1)	0	(1,3,4.1)	(2.3,4,5.1)
	Job 5	(2,3,3.5)	(3, 4, 5.1)	(2, 4, 5)	(3,4,5.1)	0	(1.3, 2, 3.1)
	Job 6	(1,3,4.1)	(2, 4, 5)	(1, 2, 3)	(2, 4, 5)	(2.3, 4.3, 5)	0

Simulated Annealing:

Simulated annealing algorithm runs with 400 degree as temperature parameter and 0.999 as cooling rate parameter.

	1	2	3	4	5	6
Machine 1	1	3	4	2	5	6
Machine 2	6	2	5	1	3	4
Machine 3	5	1	3	4	2	6
Machine 4	1	3	2	6	4	5
Machine 5	3	5	1	2	6	4
Machine 6	4	5	6	1	3	2

Job order in machines

Machine	Place
1	4
2	3
3	2
4	5
5	1
6	6

Machines location

Figure 1: Best completion time: (96, 104, 121)

Genetic Algorithm

Genetic algorithm runs by 100 as population, 0.85 as Crossover parameter and 0.05 as mutation parameter.

	1	2	3	4	5	6
Machine 1	1	3	2	4	6	5
Machine 2	2	6	1	3	4	5
Machine 3	1	2	4	3	5	6
Machine 4	1	2	3	6	4	5
Machine 5	2	1	3	5	6	4
Machine 6	4	2	6	1	3	5

Job order in machines

Machine	Place
1	1
2	4
3	3
4	2
5	5
6	6

Machines location

Figure 2: Best completion time: (123, 187, 254)

7. Conclusion

In this paper, we consider JSSP and QAP in fuzzy environment simultaneously, so our aim is sequencing of jobs on machines and locating machines simultaneously. We assume that transportation between two machines contains time. Processing time, transportation time, and set-up time defines as fuzzy parameters. The modeled problem is a fuzzy programming, to solve this problem; we utilize ranking fuzzy number. Since proposed model is a NP-hard problem, we utilize genetic and simulated annealing algorithm for solving proposed model. Finally, solving problem with other meta-heuristic methods can be a good idea for future researches. As mentioned Simulated annealing has a better answer in compare of Genetic algorithm. On the other hand, we can consider this problem with two depots, depot 1 for raw material and another for completed products; this idea can alter the jobs sequencing and other costs in proposed model.

References

[1] Bogomolnaia, A., A new solution to the random assignment problem, *J. Econo. Theor.*, vol. 100, no. 295–328, 2001.

[2] Lin, C.J., and Wen, U.P., A labeling algorithm for the fuzzy assignment problem, *Fuzzy Sets Syst*, vol. 142, pp. 373–391, 2004.

[3] Croce F. D., Tadei R., and Volta, G., A genetic algorithm for the job shop problem: *Computers & Operations Research*, vol. 22, pp. 15-24, 1995.

[4] Sule, D. R., Industrial Scheduling, *PWS Publishing Company*, pp. 37-49, 1998.

[5] Garey, E., Johnson, D., and Sethi, R., The complexity of flow shop and job-shop scheduling: *Mathematics of Operations Research*, vol. 1, pp. 117-29, 1976.

[6] Fisher, H., and Thompson, G.L., Probabilistic-learning combinations of local job-shop scheduling rules, in: J. Muth, G. Thompson (Eds.), *Industrial Scheduling, Prentice-Hall, Englewood Cliffs, NJ*, 1963.

[7] Essafia, I., Mati, Y., and Dauzere-Peres, S., A genetic local search algorithm for minimizing total weighted tardiness in the job-shop scheduling problem*: Computers & Operations Research,* vol. 35, pp. 2599 – 2616, 2008.

[8] Li, J. Q., Pan, Q.-K., and Liang, Y.-C., An effective hybrid tabu search algorithm for multi-objective flexible job-shop scheduling problems, *Computers & Industrial Engineering*, vol. 59, pp. 647–662, 2010.

[9] Jackson, J. R., An extension of Johnson's results on job lot scheduling: *Navel Research Logistics Quarterly*, vol. 3, pp. 201-203, 1956.
[10] Lawler, E., Quadratic assignment problem, *Management Science*, vol. 9, pp. 586-599, 1963.
[11] Lawrence, S., Resource constraint project scheduling: An experimental investigation of heuristic scheduling techniques, *Carnegie-Mellon University*, pp. 412-435, 1993.
[12] Sakawa, M., and Kubota, R., Two-objective fuzzy job shop scheduling through genetic algorithms, *Electron. Commun. Jpn*., vol. 84, no. 4, p. 3, 2001.
[13] Sakawa, M., and Mori, T., Job shop scheduling with fuzzy due date and fuzzy processing time through genetic algorithms (in Japanese), *Journal of Japan Society for Fuzzy Theory and Systems*, vol. 9, no. 2, pp. 231-238. 1997.
[14] Sakawa, M., and Kubota, R., Fuzzy programming for multi objective job shop scheduling with fuzzy processing time and fuzzy due date through genetic algorithms, *European Journal of Operational Research*, vol. 120, pp. 393-407, 2000.
[15] Ghrayeb, O., An efficient genetic algorithm for JSSP with fuzzy durations, *Proceedings of Industrial Engineering Research Conference*, Cleveland, OH, 2000.
[16] Fortemps, P., Job shop scheduling with imprecise durations: a fuzzy approach, *IEEE Trans. Fuzzy Syst*., vol. 5-4, pp. 557– 569, 1997.
[17] Huang, R.-H., Multi-objective job-shop scheduling with lot-splitting production, *International Journal of Production Economics*, vol. 124, pp. 206–213, 2010.
[18] Lawrence, S., Resource constrained project scheduling: an experimental investigation of heuristic scheduling techniques, *Carnegie Mellon University, Technical Report, Pittsburg*, PA, 1984.
[19] Tsujimura, Y., Gen, M., and Kubota, E., Solving job-shop scheduling problem with fuzzy processing time using genetic algorithm, *Journal of Japan Society for Fuzzy Theory and Systems*, vol. 7, no. 5, pp. 1073-1083, 1995.
[20] Yao, J.-S., and Wu, K., Ranking fuzzy numbers based on decomposition principle and signed distance, *Fuzzy Sets Systems*, vol. 116, pp. 275–88, 2000.

Appendix

Code: Simulated annealing algorithm runs with 400 degree as temperature parameter and 0.999 as cooling rate parameter.

```
Begin
m: number of machines or locations;
n: number of jobs;
max_c, tmax_c: Maximum Completion time among jobs. (The second one is temporary variable);
earl_Y[][], temp_earl_Y[][]: earliest time that each job can be done in each machine.(The second one is temporary
array);
mach_job[][], temp_mach_job[][]:Priority of jobs that should be done in each machine.(The second one is
temporary array);
mach_place[], temp_mach_place[]:Indicates place of each machine.(The second one is temporary array);
While temperature>epsilon do
    N1 = Next_Sol();
    tmax_c = fitness (N1);
    delta = tmax_c – max_c;
    proba = double random number between 0 and 1;
    If delta<0 then
        mach_job = temp_mach_job;
        earl_Y = temp_earl_Y;
        max_c = tmax_c;
    Else If proba< exp(-delta/temperature) then
        mach_job = temp_mach_job;
        earl_Y = temp_earl_Y;
        max_c = tmax_c;
    Else
        swap(temp_mach_place[loc2], temp_mach_place[loc1]);
        swap(temp_mach_job[M_mach][J_job2], temp_mach_job[M_mach][J_job1]);
    End If;
    If max_c < B_max_c Then
        B_mach_job = mach_job;
        B_earl_Y = earl_Y;
        B_max_c = max_c;
    End If;
    temperature = temperature * alpha;
End While.
End.

Next_Sol Procedure ():
  loc1= random integer number between 1 to m;
  loc2= random integer number between 1 to m;
  swap(temp_mach_place[loc1], temp_mach_place[loc2]);

  M_mach= random integer number between 0 to m;
  J_job1= random integer number between 0 to n-1;
  J_job2= random integer number between 0 to n-1;
  swap(temp_mach_job[M_Mach][J_Job1],temp_mach_job[M_Mach][J_Job2]);
```

Code: Genetic algorithm runs by 100 as population, 0.85 as Crossover parameter and 0.05 as mutation parameter.

```
Begin
m:number of machines or locations;
n: number of jobs;
crossoverP: percentage of Crossover
mutationP: percentage of Mutation
populationSize: number of chromosomes
maxGeneration: nember of generations

 population ← Create first population;
generation ←1;
While generation < MaxGeneration do

    Evaluate( poulatoin);
    If Best_Fitness_in_Genaration>Best_Fitness then
        Best_Fitness = Best_Fitness_in_generation;
        Best_Completion_Time = Best_Completion_Time_in_generation;
    End If.
    P ← a ranodom number between 0 and 1;
        If crossoverP>P then
                Crossover(population);
        End if.
    If mutationP >P then
                Mutation(population);
    End if.
    generation = generation+1;
End While.
Return Best_completio_time;
End.

Evaluate procedure:
Begin
For i:1 to populationSize
Decode chromosome number i;
Calculate completion_time of each job;
Calaculate chromosome_Fitness based on completion_time;
If Chromosome_Fitness> Best_Fitness_in_Generation then
    Best_Fitness_in_Generation = Choromosome_Fitness;
    Best_Completion_Time_in_generation = completion_time
End if.
End.
```