Journal of Applied Sciences

ISSN 1812-5654

science alert

ANSInet
an open access publisher
http://ansinet.com

Journal of Applied Sciences 8 (14): 2553-2560, 2008
ISSN 1812-5654

Flexible Flow-Lines Model at m Stage in an Inexact Environment with Interval Coefficients

[1]M.B. Fakhrzad, [2]Mehdi Heydari and [1]H. Khademi Zare
[1]Department of Industrial Engineering, Yazd University, Yazd, Iran
[2]Department of Industrial Engineering, University of Science and Technology, Tehran, Iran

Abstract: In this study, we consider the tardiness and earliness-minimizing inexact flexible flow line problem with n jobs and m stages and uncertain processing times, setup times and due-dates. For each operation, an uncertainty interval is given and it is assumed that the processing time of each operation can take on any value from the corresponding uncertainty interval, regardless of the values taken by processing times of other operations. For most of scheduling problems, processing times, setup times and due-dates are treated as certain values, but that is not proper to all actual situations. Processing times and setup times are not constant because of measurement errors in the data sets for deciding them and/or human actions in the manufacturing process. In some cases, a decision maker may prefer using interval numbers as coefficients of an inexact relationship. As a coefficient an interval assumes an extent of tolerance or a region that the parameter can possibly take. A model mixed integer design of the problem is formulated in inexact environment. On the basis of a comparative study on ordering interval numbers, inequality constraints involving interval coefficients are reduced in their satisfactory crisp equivalent forms and a satisfactory solution of the problem is defined.

Key words: Flexible flow, line, interval number, inequality relation

INTRODUCTION

The objective of a flow-shop sequencing problem is to find the sequence of jobs that minimizes the maximum flow time. Johnson published the first paper on the flow-shop problem in 1954. This problem has held the attention of many researchers and has been extensively studied in the literature (Kurz and Askin, 2003; Logendran *et al*., 2005). A survey paper about the flow-shop problem was presented by Ruiz and Maroto (2006), Ruiz *et al.* (2008) and Coffman (2002). Scheduling jobs in flexible flow-lines is considered an NP-complete problem (Liu and Chang, 2000; Janiak *et al*., 2007), which prompted the development of many heuristics to provide a valuable and quick solution (Gupta *et al*., 2002; Vob and Witt, 2007). Most of the methods proposed in the literature assume that all of the time parameters and relevant data are already exactly known. However, in the practical sense, this assumption is unrealistic since there are many vaguely formulated relationships between jobs and many imprecisely quantified job processing time values in a real world description of the flow-shop problem (Pugazhendhi *et al*., 2004; Jin *et al*., 2006).

In conventional mathematical programming, coefficients of problems are usually determined by the experts as crisp values. But in reality, in an imprecise and uncertain environment, it is an unrealistic assumption that the knowledge and representation of an expert are so precise. Hence, in order to develop good Operations Research methodology fuzzy and stochastic approaches are frequently used to describe and treat imprecise and uncertain elements present in a real decision problem. In fuzzy programming problems the constraints and goals are viewed as fuzzy sets and it is assumed that their membership functions are known (Sakawa and Kubota, 2000; Hong and Wang, 2000; Litoiu and Tadei, 2001) On the other hand, in stochastic programming problems the coefficients are viewed as random variables and it is also assumed that their probability distributions are known (Kurz and Askin, 2004; Lin and Liao, 2003; Allaoui and Artiba, 2006). These membership functions and probability distributions play important roles in their corresponding methods. However, in reality, to a Decision Maker (DM) it is not always easy to specify the membership function or the probability distribution in an inexact environment. Therefore, it is much easier for a Decision-Maker (DM) to specify a value range than to give an exact value for the processing time of each job. Accordingly, use of an interval coefficient may serve the purpose better. Though by using α-cuts, fuzzy numbers

Corresponding Author: M.B. Fakhrzad, Department of Industrial Engineering, Yazd University, Yazd, Iran
Tel: +98 (21) 77240128 Fax: +98 (21) 77240482

can be degenerated into interval numbers (Jiang *et al.*, 2008), deliberately we keep this concept out of the scope of this paper. Here, an interval number is considered as an extension of a real number and as a real subset of the real line $\Re$ (Jiang *et al.*, 2007). As a coefficient an interval also signifies the extent of tolerance (or a region) that the parameter can possibly take. However, in decision problems its use is not much attended as its merits. In this study, we concentrate on a satisfactory solution approach based on DM's interpretation of inequality relations and objective of the problem with respect to the inexact environment.

THE MATHEMATICAL MODEL

A hybrid flow-shop system is defined by the set M = {1, …, j, …, m} of m processing stages (machine center). Each stage j, j ∈ M is a set composed of k identical machines. Set I = {1, …, i, …, n} composed of n independent jobs has to be processed at the M stages and one of K identical machines at each stage. Each job i, i ∈ I is considered as a sequence of m operations with processing times $P_{ij} \geq 0$, j ∈ M) $P_{ij} = (P_{ija}, P_{ijc})$.

The Jth operation of a job at the jth stage can commence only after the completion of (j-1) previous operations from the sequence. Processing of each job can not be started before its release date and each machine can process only one operation at a time. All jobs are available at time zero. Most of the parameters used in this problem are inexact and perhaps appropriately given in terms of simple intervals. In reality inexactness of this kind can be cited in countless numbers.

Thus, the optimization hybrid flow-shop mathematical model is defined as follows:

Decision variables and parameters: In this study, the assumptions and notations are described as follows:

Assumptions:

- Each machine can process only one job at a time;
- All jobs are available for machine processing simultaneously at time zero;
- Jobs are not pre-emptive;
- Each job has m tasks to be executed in sequence on m machine centers;
- All machine centers have the same number of identical machines;

Decision variables and parameters are defined as follows:

i = Index of jobs i = 1, 2,...., N
j = Index of stages j = 1, 2,...., m
k = Index of machines k = 1, 2,...., K
$C_i = (C_{ia}, C_{ic})$ completion time of job i
$C_{ij} = (C_{ija}, C_{ijc})$ completion time of job i in stage j
$T_i = (T_{ia}, T_{ic})$ tardiness of job i
$E_i = (E_{ia}, E_{ic})$ earliness of job i
$d_i = (d_{ia}, d_{ic})$ due date of job i
$R_i = (R_{ia}, R_{ic})$ ready time of job i
$P_{ij} = (P_{ija}, P_{ijc})$ processing time of job i in stage j
$S_{ij} = (S_{ija}, S_{ijc})$ set-up time of job i in stage j
$H_i = (H_{ia}, H_{ic})$ holding cost of job i per time unit
$\beta_i = (\beta_{ia}, \beta_{ic})$ shortage cost of job i per time unit
$U_{ij} = (U_{ija}, U_{ijc})$ starting time of job i in stage j

$$X_{ijk} = \begin{cases} 1 & \text{if job i in stage j is allocated to machine k} \\ 0 & \text{otherwise} \end{cases}$$

Objective function: The objective in to minimize the total holding and shortage cost associated with earliness and tardiness.

$$\text{Min } Z = \sum_{i=1}^{N}\left[(H_{ia}, H_{ic})E_i + (\beta_{ia}, \beta_{ic})T_i\right] \quad (1)$$

S. t.

$$C_i + E_i - T_i = (d_{ia}, d_{ic}) \quad i = 1,2,...,n \quad (2)$$

$$\sum_{k=1}^{K} X_{ijk} = 1 \quad i = 1,2,....N, \; j = 1,2,....,m \quad (3)$$

$$\sum_{i=1}^{N} X_{ijk} = k \quad k = 1,...,K \quad j = 1,...,m \quad (4)$$

$$C_{i,[j+1]} - C_{i,[j]} + L(1 - X_{ijk}) \geq (P_{ija}, P_{ijc}) + (S_{ija}, S_{ijc}) \quad i = 1,2,...,N, \; j = 1,2,...,m, \; k = 1,2,...,K \quad (5)$$

$$C_i \geq \sum_{j=1}^{m} (P_{ija}, P_{ijc}) + (S_{ija}, S_{ijc}) \quad i = 1,2,...,N \quad (6)$$

$$U_{i,j+1} \geq U_{ij} + (P_{ija}, P_{ijc}) + (S_{ija}, S_{ijc}) \quad i = 1,2,....,N, \; j = 1,2,....,m \quad (7)$$

$$U_{i1} \geq (R_{ia}, R_{ib}) \quad i = 1,2,...., N \quad (8)$$

$$X_{i,j,k} = 0,1 \quad i = 1,2,...,N, \; j = 1,2,....,m, \; k = 1,2,....,K \quad (9)$$

$$(E_i, T_i, U_{ij}) \geq 0 \quad i = 1,2,....,N, \quad j = 1,2,...,m \quad (10)$$

Relation (2) reflects the earliness or tardiness for each part with respect to the defined due date. Relation (3) indicates that job i in stage j requires only one machine. Relation (4) guarantees that a machine can process at most one job at a time. Relation (5) assures that completion time of each job that immediately precedes another job is greater than or equal to the sum of processing and set-up times of that job on all machines. In this relation L denotes larger positive number. Relation (6) ensures that completion time of each job in stage j is greater than or equal to the sum of processing and set-up times on all machines. Relation (7) shows that total starting time of each job at stage j is greater than or equal to its starting time in the previous stage and its processing and set-up times at that stage. Relations (8) represents the starting time constraint for job i and Relations (9) and (10) represent the state of the variables.

However, for solution, techniques of classical linear programming cannot be applied if and unless the above interval-valued structure of the problem be reduced into a standard linear programming structure and for that we have to clear up the following main issues:

- First, regarding interpretation and realization of the inequality relations involving interval coefficients.
- Second, regarding interpretation and realization of the objective `Min' with respect to an inexact environment.

THE BASIC INTERVAL ARITHMETIC

All lower case letters denote real numbers and the upper case letters denote the interval numbers or the closed intervals on $\Re$.

3.1 $$A = [a_L, a_R] = \{a : a_L \le a \le a_R, a \in \Re\},$$

where, a_L and a_R are left and right limit of the interval A on the real line $\Re$, respectively. If $a_L = a_R$, then A = [a; a] is a real number.

Interval A is alternatively represented as $A = \langle m(A), w(A)\rangle$ where, m (A) and w (A) are the mid-point and half-width (or simply be termed as 'width') of interval A, i.e.,

$$m(A) = \frac{1}{2}(a_L + a_R), \qquad w(A) = \frac{1}{2}(a_R - a_L)$$

3.2. Let * ∈ {+, −,., ÷} be a binary operation on the set of real numbers.

Then, $A \underset{\sim}{*} B = \{a * b, a \in A, b \in B\}$ defines a binary operation on the set of closed intervals. In case of division it is assumed that $0 \notin B$.

If λ is a scalar, then

$$\lambda.A = \lambda[a_L, a_R] = \begin{cases} \lambda[a_L, a_R] & \text{for } \lambda \ge 0, \\ \lambda[a_R, a_L] & \text{for } \lambda < 0. \end{cases}$$

The extended addition $\underset{\sim}{+}$ and extended subtraction $\underset{\sim}{-}$, are defined as follows:

$$A \underset{\sim}{+} B = [a_L + b_L, a_R + b_R],$$
$$A \underset{\sim}{-} B = [a_L - b_R, a_R - b_L],$$

The following equations also hold for $A \underset{\sim}{+} B$ and $A \underset{\sim}{-} B$:

$$m(A \underset{\sim}{+} B) = m(A) + m(B)$$
$$m(A \underset{\sim}{-} B) = m(A) - m(B),$$
$$w(A \underset{\sim}{+} B) = w(A \underset{\sim}{-} B) = w(A) + w(B)$$

INEQUALITY RELATION WITH INTERVAL COEFFICIENTS

An extensive research and wide coverage on interval arithmetics and its applications can be found in Wei and Chen (2007). Here we find two transitive order relations defined over intervals: the first one as an extension of '<' on the real line as A < B iff $a_R < b_L$ and the other as an extension of the concept of set inclusion i.e.,

$$A \subseteq B \text{ iff } a_L \ge b_L \text{ and } a_R \le b_R.$$

These order relations cannot explain ranking between two overlapping intervals. The extension of the set inclusion here only describes the condition that the interval A is nested in B; but it cannot order A and B in terms of value. We need to develop a definition of comparing two interval numbers.

Sevastianov (2007) approached the problem of ranking two interval numbers more prominently. In their approach, in a maximization problem if intervals A and B are two, say, profit intervals, then maximum of A and B can be defined by an order relation $\le_{LR}$ between A and B as follows:

$$A \le_{LR} B \text{ iff } a_L \le b_L \text{ and } a_R \le b_R, \; A <_{LR} B \text{ iff } A \le_{LR} B \text{ and } A \ne B.$$

Sevastianov (2007) suggested an another order relation $\le_{mw}$ where, $\le_{LR}$ cannot be applied, as follows:

$$A \le_{mw} B \text{ iff } m(A) \le m(B) \text{ and } w(A) \ge w(B),$$

$A <_{mw} B$ iff $A \le_{mw} B$ and A≠B Both of the above order relations $\le_{LR}$ and $\le_{mw}$ are antisymmetric, reflexive and transitive and hence, define partial ordering between intervals. Sevastianov (2007) showed that both of the order relations never conflict in the sense that there exists no such pair of A and B) A ≠ B (so that $A \le_{LR} B$ and $B \le_{mw}$ hold.

Sengupta *et al.* (2001) showed that there exists a set of pairs of intervals for which both of $\le_{LR}$ and $\le_{mw}$ do not hold. They proposed a simple and efficient index for comparing any two interval numbers on the real line through decision maker's satisfaction.

Tong's approach: Tong (1999) deals with interval inequality relations in a separate way.

For a minimization problem as follows:

$$\text{Minimize } Z = \sum_{j=1}^{n} [c_{Lj}, c_{Rj}] x_j,$$

Subject to

$$\sum_{j=1}^{n} [a_{Lij}, a_{Rij}] x_j \geq [b_{Li}, b_{Ri}]$$
$$\forall i = 1, 2, \ldots, m,$$
$$x_j \geq 0, \quad \forall j,$$

each inequality constraint is first transformed into 2^{n+1} crisp inequalities to yield

$$D_i = \{D_i^k / k = 1, 2, \ldots, 2^{n+1}\},$$

which are the solutions to the ith set of 2^{n+1} inequalities.

On the other hand, Tong defines a characteristic formula (CF)

$$\sum_{j=1}^{n} a_{ij} x_j \geq b_i$$

of the ith inequality relation, $\forall i$, where $a_{ij} \in [a_{Lij}, a_{Rij}]$ and $b_i \in [b_{Li}, b_{Ri}]$.

Now, if the ith CF generates solution D_i such that

$$D_i = \bigcup_{k=1}^{2^{n+1}} D_i^k,$$

then CF is called maximum-value range inequality and if CF generates solution D_i such that

$$D_i = \bigcap_{k=1}^{2^{n+1}} D_i^k,$$

then it is called minimum-value range inequality. Tong (1999) then defines minimum and maximum optimal objective value of the problem using max and min value inequalities, respectively.

Discussion: Let us take a very simple inequality relation with a single variable, [10, 20] x ≤ [5, 35]. According to Gupta *et al.* (2002) the interval inequality generates 2^{1+1} crisp inequalities:

$$\left.\begin{array}{l} 10x \leq 5 \Rightarrow x \leq 0.5 \\ 10x \leq 35 \Rightarrow x \leq 3.5 \\ 20x \leq 5 \Rightarrow x \leq 0.25 \\ 20x \leq 35 \Rightarrow x \leq 1.75 \end{array}\right\} D = \{D^k / k = 1, 2, 3, 4\}$$

$$\bar{D} = \bigcup_{k=1}^{2^2} D^k \Rightarrow x \leq 3.5\text{: max value range inequality;}$$

$$\bar{D} = \bigcap_{k=1}^{2^2} D^k \Rightarrow x \leq 0.25\text{: min value range inequality;}$$

Here we would like to raise a question on Tong's approach (1999): how does one interpret the use of the operators union and intersection in defining max and min-value range inequalities, respectively?

Using the union operator in defining the crisp equivalent form of the ith original constraint may be interpreted as at least one element of the interval $A_i x$ is less than or equal to at least one element of interval B_i which clearly does not validate the original constraint condition. Using ϑ-index it can be shown that $\vartheta(B_i \precsim A_i x) = 1$, i.e., $A_i x$ is definitely greater than B_i.

On the other hand, using the intersection operator in defining the crisp equivalent form may be interpreted as all elements of $A_i x$ is less than or equal to all elements of B_i which is merely an oversimplification. Using ϑ-index, it can be shown that $\vartheta(B_i \precsim A_i x) = 1$.

In actual practice, for a wide range of feasibility of the decision variable vector, DM may allow $A_i x$ even to be nested in B_i, i.e., some/all elements of $A_i x$ may even be allowed to be greater than or equal to some elements of B_i and that how much to be allowed will be decided by the DM and this will depend on his optimistic attitude, on his risk versus benefit assessment and as a whole, on the level of satisfaction the DM tries to achieve from the decision-making process.

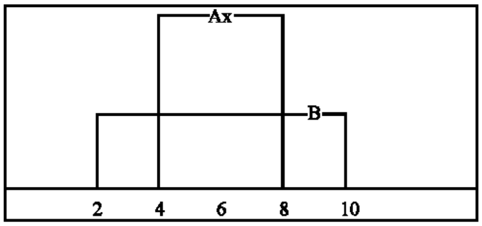
Hence, in our opinion, some sort of conditions indicating DM's satisfaction/utility requirement has to be incorporated in generation of a crisp equivalent structure of the inequality constraint with interval coefficients. Using the properties of ϑ-index we develop a satisfactory crisp equivalent structure of an inequality constraint with interval coefficients.

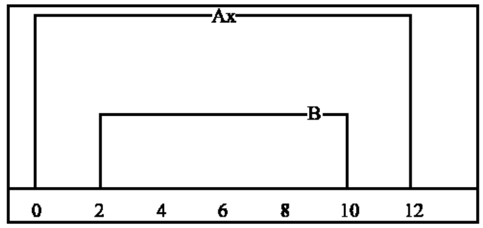
A satisfactory crisp equivalent system of Ax≤B: Let $A = [a_L, a_R]$, $B = [b_L, b_R]$ and x is a singleton variable.

According to ϑ-index the acceptability condition of Ax≤B may be defined as $\vartheta(Ax \precsim B) \geq 0$,
i.e., m (Ax) ≤ m (B).

Now, let us take the condition m (Ax) = m(B), then, for a given value of x, we may have two different possible setups.

Case-I: When interval A is relatively narrower than interval B: Ax may be nested in B. For example, for x = 2, the relation [2, 4] x ≤ [2, 10] may be viewed as given:





Case-II: When interval A is relatively wider than it was in case I: B may be nested in Ax. For example, for x = 2, the relation [0, 6] x ≤ [2, 10] may be viewed as shown:

From the examples given above, for both of the cases, the following remarks may be made:

- Case I definitely satisfies the original interval inequality for x ≤ 2 because $\vartheta(Ax \precsim B) \geq 0$. However, an optimistic DM may remain under-satisfied with the optimal constraint condition and for getting higher satisfaction, he may like to increase the value of x to such an extent that $\vartheta(Ax \precsim B)$ does not pass over a threshold assumed and fixed by him.
- On the other hand, by case II, the original interval inequality condition is not denied even for x ≤ 2 because $\vartheta(Ax \precsim B) \geq 0$. But a pessimistic DM may not be satisfied if the right limit of Ax spills over the right limit of B. To attain his required level of satisfaction the DM may even like to reduce the value of x so that $a_R x \leq b_R$.

To shed more light on interval inequality relation from a different angle, let us refer an equivalent form of a deterministic inequality where:

$$ax \leq b \text{ is } az \in]-\infty, b]$$

Let us extend this concept to an inexact environment: if the real numbers a and b are allowed to be replaced by intervals A and B, respectively, that one's possible reaction is as much as similar to Wei and Chen's concept of set-inclusion, i.e.,

$$Ax \leq B \Rightarrow Ax \subset D$$

where $D =]-\infty, b_R$.

Keeping in view the two remarks stated above and the Wei and Chen's concept (2007), we propose a satisfactory crisp equivalent form of interval inequality relation as follows:

$$Ax \leq B \Rightarrow \begin{cases} a_R x \leq b_R, \\ \vartheta(Ax \precsim B) \leq \alpha \in [0,1] \end{cases}$$

where, α may be interpreted as an optimistic threshold assumed and fixed by the DM.

Similarly, for Ax ≥ B, we have the satisfactory crisp equivalent form by the following pair:

$$a_L x \geq b_L,$$

$$\vartheta(Ax \precsim B) \leq \alpha \in [0,1]$$

AN INTERVAL LINEAR PROGRAMMING PROBLEM AND ITS SOLUTION

Let us consider the following problem:

$$\text{Minimize} \quad Z = \sum_{j=1}^{n} [c_{Lj}, c_{Rj}] x_j$$

$$\text{Subject to} \quad \sum_{j=1}^{n} [a_{Lij}, a_{Rij}] x_j \geq [b_{Li}, b_{Ri}]$$

$$\forall i = 1, 2, ..., m,$$

$$x_j \geq 0, \quad \forall j$$

As is described in the previous section a satisfactory crisp equivalent system of constraints of the ith interval constraint can be generated as follows:

$$\sum_{j=1}^{n} a_{Lij} x_j \geq b_{Li}, \quad \forall i,$$

$$b_{Li} + b_{Ri} - \sum_{j=1}^{n} (a_{Lij} + a_{Rij}) x_j \leq$$

$$\alpha (b_{Ri} - b_{Li}) + \alpha \sum_{j=1}^{n} (a_{Rij} - a_{Lij}) x_j$$

The working of ϑ-index may be summarized by the following principle:

The position (of mean) of an interval compared to that of another reference interval results in whether the former is superior or inferior to the later. On the other hand, the width of a superior (inferior) interval compared to that of the reference interval specifies the grade to which the DM is satisfied with the superiority (inferiority) of the former compared to the later.

The objective of a conventional linear programming problem is to maximize or minimize the value of its (one only, single-valued) objective function satisfying a given set of restrictions. But, a single-objective interval linear programming problem contains an interval-valued objective function. As an interval can be represented by any two of its four attributes (viz., left limit, right limit, mid-value and width), an interval linear programming, by using attributes mid-value and width (say) can be reduced into a linear biobjective programming problem as follows (Sengupta *et al*., 2001):

Max/Min {mid-value of the interval objective function},
Min {width of the interval objective function},
sub: to {set of feasibility constraints}.

From this problem naturally one may get two conflicting optimal solutions:
$x^* = \{x_j^*\}$ from max/min {mid value}
sub: to {constraints},
$x^{**} = \{x_j^{**}\}$ from min {width}
sub: to {constraints}
and from there two optimal interval values Z^* and Z^{**}.

If $x^* = x^{**}$ then there does not exist any conflict and x^* is the solution of the problem:

But if $x^* \neq x^{**}$, for the maximization problem, $m(Z^*) > m(Z^{**})$ and $w(Z^*)>w(Z^{**})$ (because, Z^* is obtained through maximizing m(Z) and Z^{**} is obtained not by maximizing m (Z), but through another goal, by minimizing w (Z)).

Similarly, for minimizing problem, if $x^* \neq x^{**}$, then, $m(Z^*) < m(Z^{**})$ and $w(Z^*) > w(Z^{**})$ (because Z^* here is obtained by minimizing m(Z) and Z^{**} by minimizing w(Z)).

Therefore, if $x^* \neq x^{**}$, Z^* and Z^{**} are said to be two non-dominated alternative extreme interval objective values Jiang *et al.* (2008). On the other hand, the principle of ϑ-index indicates that for the maximization (minimization) problem, an interval with a higher mid-value is superior (inferior) to an interval with a lower mid-value.

Therefore, though Z^* and Z^{**} are two non-dominated alternatives from the viewpoint of a biobjective problem, as two interval values of the interval-valued objective function of the original problem they can be ranked.

Hence, in order to obtain max/min of the interval objective function, considering the mid-value of the interval-valued objective function is our primary concern. We reduce the interval objective function its central value and use conventional LP techniques for favour of its solution.

We also consider width but as a secondary attribute, only to confirm whether it is within the acceptable limit of the DM. If it is not, one has to reduce the extent of width (uncertainty) according to his satisfaction and thus to obtain a less wide interval from among the non-dominated alternatives accordingly.

The following LP problem is the necessary equivalent form of the original problem:

$$\text{Minimize } m(Z) = \frac{1}{2}\sum_{j=1}^{n}(c_{Lj} + c_{Rj})x_j$$

$$\text{subject to } \sum_{j=1}^{n} a_{Lij}x_j \geq b_{Li}, \quad \forall i,$$

$$b_{Li} + b_{Ri} - \sum_{j=1}^{n}(a_{Lij} + a_{Rij})x_j$$

$$\leq \alpha(b_{Ri} - b_{Ri}) + \alpha\sum_{j=1}^{n}(a_{Rij} - a_{Lij})x_j,$$

$$x_j \geq 0, \quad \forall j.$$

It is only when there exists the possibility of multiple solutions, that comparative widths are required to be calculated and then in favour of a minimum available width, we get the solution.

COMPUTATIONAL EXPERIMENTS

The efficiency of the proposed algorithms is verified by choosing 220 random instances with the following characteristics:

- Dimensions of the problems are between (N.M.K) = (25×5×3) and (N.M.K) = (100×20×12)
- Set-up times for each job at each stage are chosen interval numbers (1, 15) and crisp numbers (5, 10).
- Holding (earliness) and shortage (tardiness) costs for each job at each stage are chosen interval numbers (1, 15) and crisp numbers (5, 10).
- Operation processing time for each job at each stage is chosen fuzzy numbers (10, 40) and crisp numbers (20, 30).
- Due date for each job at each final stage is chosen interval numbers (150, 450) and crisp numbers (250, 350).

Accordingly, two categories of problems have been discussed with exact and interval numbers. As described in section 5, interval number problems are first converted to an exact model and then are solved using Lingo 7 software. Also, exact number problems directly can be solved by Lingo 7 software and results are provided in Table 1. According to the provided results in Table 1, mean values of the objective functions for interval number and exact number match. This is shown in Fig. 1.

Table 1: Comparison of objective function mean values for exact and interval number problems

		Objective function of value			
	No. of	Crisp No.	Interval No.		
Problem size (N.M.K)	problems solved	b	a	c	Ave. interval numbers
20×5×3	5	22.65	15.21	35.42	25.32
20×8×3	8	24.12	16.11	37.18	26.65
20×10×3	10	26.15	17.25	39.41	28.33
20×12×3	12	31.18	19.41	41.12	30.27
20×15×3	15	35.27	21.15	42.15	31.65
30×5×5	5	36.45	25.41	48.11	36.76
30×8×5	8	39.75	27.11	50.12	38.62
30×10×5	10	42.84	28.19	61.17	44.68
30×12×5	12	45.27	32.18	62.11	47.15
30×15×5	15	48.44	35.41	65.19	50.30
50×5×8	5	51.28	41.15	69.11	55.13
50×8×8	8	52.18	42.17	70.15	56.16
50×10×8	10	54.29	46.11	71.12	58.62
50×12×8	12	56.44	52.15	74.19	63.17
50×15×8	15	58.99	56.11	75.11	65.61
80×5×10	5	61.85	59.15	76.19	67.67
80×8×10	8	63.24	61.12	76.89	69.00
80×10×10	10	65.41	62.17	77.34	69.76
80×12×10	12	66.81	62.85	79.41	71.13
80×15×10	15	68.85	63.11	80.11	71.61
100×10×12	5	69.21	65.15	81.19	73.17
100×15×12	8	71.95	67.18	82.25	74.72
100×20×12	10	72.45	69.15	85.18	77.17

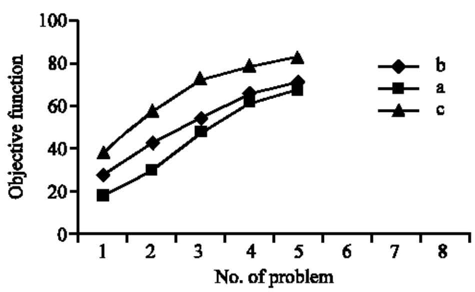


Fig. 1: Comparison of objective function. a, c: Interval numbers; b: Crisp numbers

CONCLUSION

Appropriate scheduling not only reduces manufacturing costs but also reduces the possibility of violating due dates. Finding good schedules for given set of jobs can thus help factory supervisors control job flows and provide for nearly optimal job sequencing. Scheduling jobs in flexible flow-lines has long been known to be an NP complete problem. Since task processing times, set-up times, due dates, holding costs and shortage costs in real applications are usually uncertain, in this study; we have defined a satisfactory crisp equivalent system of an inequality constraint with interval coefficients. The approach defined here has come out as an application of ϑ-index for comparing two intervals through DM's satisfaction. Once the crisp equivalent structure of the constraint set is defined, solution to a problem with maximizing or minimizing objective function practically turns to be maximization or minimization of the central value of the interval-valued objective function. In this regard, a point worth mentioning is: If the DM is not satisfied with the extent of uncertainty (width) involved in the optimal objective value, he can achieve his required level of satisfaction by adjusting allowable width of the optimal objective value and/or by redefining satisfying conditions for generating crisp equivalent set of constraints.

REFERENCES

Allaoui, H. and A. Artiba, 2006. Scheduling two-stage hybrid flow shop with availability constraints. Comput. Operat. Res., 33: 1399-1419.

Coffman, G., 2002. Computer and job-shop scheduling theory. Int. J. Approx. Reason, 29: 215-234.

Gupta, J.N.D., K. Krüger, V. Lauff, F. Werner and Y.N. Sotskov, 2002. Heuristics for hybrid flow shops with controllable processing times and assignable due dates. Comput. Operat. Res., 29: 1417-1439.

Hong, T.P. and T.T. Wang, 2000. Fuzzy flexible flow shop at two center for continuous fuzzy domains. Inform. Sci., 129: 227-237.

Janiak, A., E. Kozan, M. Lichtenstein and C. Oguz, 2007. Metaheuristic approaches to the hybrid flow shop scheduling problem with a cost-related criterion. Int. J. Prod. Econ., 105: 407-424.

Jiang, C., X. Han, F.J. Guan and Y.H. Li, 2007. An uncertain structural optimization method based on nonlinear interval number programming and interval analysis method. Eng. Struct., 29: 3168-3177.

Jiang, C., X. Han, G.R. Liu and G.P. Liu, 2008. A nonlinear interval number programming method for uncertain optimization problems. Eur. J. Operat. Res., 188: 1-13.

Jin, Z., Z. Yang and T. Ito, 2006. Metaheuristic algorithms for the multistage hybrid flow shop scheduling problem. Int. J. Prod. Econ., 100: 322-334.

Kurz, M.E. and R.G. Askin, 2003. Comparing scheduling rules for flexible flow lines. Int. J. Prod. Econ., 85: 371-388.

Kurz, M.E. and R.G. Askin, 2004. Scheduling flexible flow lines with sequencing-dependent setup times. Eur. J. Operat. Res., 159: 66-82.

Lin, H.T. and C.J. Liao, 2003. A case study in a two-stage hybrid flow-shop with setup time and dedicated machines. Int. J. Prod. Econ., 86: 133-143.

Litoiu, M. and R. Tadei, 2001. Fuzzy scheduling with application to real-time systems. Fuzzy Set Syst., 121: 523-535.

Liu, C.Y. and S.C. Chang, 2000. Scheduling flexible flow shops with sequence-dependent setup effects. IEEE Trans. Robotics Autom., 16: 408-419.

Logendran, R., S. Carson and E. Hanson, 2005. Group scheduling in flexible flow shops. Int. J. Prod. Econ., 96: 143-155.

Pugazhendhi, S., S. Thiagarajan, C. Rajendran and N. Anantharaman, 2004. Generating non-permutation schedules in flowline based manufacturing systems with sequence-dependent setup times of jobs: A heuristic approach. IJAMT., 23: 64-78.

Ruiz, R. and C. Maroto, 2006. A genetic algorithm for hybrid flowshops with sequence dependent setup times and machine eligibility. Eur. J. Operat. Res., 169: 781-800.

Ruiz, R., F. Sivrikaya-Serifoglu and T. Urlings, 2008. Modeling realistic hybrid flexible flowshop scheduling problems. Comput. Operat. Res., 35: 1151-1175.

Sakawa, M. and R. Kubota, 2000. Fuzzy programming for multi objective job shop scheduling with fuzzy processing time and fuzzy due date through genetic algorithms. Eur. J. Operat. Res., 120: 393-407.

Sengupta, A., T. Kumar Pal and D. Chakraborty, 2001. Interpretation of inequality constraints involving interval coefficients and a solution to interval linear programming. Fuzzy Set Syst., 119: 129-138.

Sevastianov, P., 2007. Numerical methods for interval and fuzzy number comparison based on the probabilistic approach. Inform. Sci., 77: 4645-4661.

Tong, S., 1999. Interval number and fuzzy number linear programming. Fuzzy Set Syst., 66: 301-306.

Vob, S. and A. Witt, 2007. Hybrid flow shop scheduling as a multi-mode multi-project scheduling problem with batching requirements: A real-word application. Int. J. Prod. Econ., 105: 445-458.

Wei, S.H. and S.M. Chen, 2007. Fuzzy risk analysis based on interval-valued fuzzy numbers. Expert Syst. Applied, In Press, Available online 28 December.