CMS 2: 279–293 (2005)

DOI: 10.1007/s10287-005-0035-z

Bilevel programming approach applied to the flow shop scheduling problem under fuzziness

Samir A. Abass

Department of Mathematics and Theoretical Physics, Nuclear Research Center, Atomic Energy Authority, P.O. Box 13759, Cairo, Egypt
(e-mail: samir_abdou@egypt.tv)

Abstract. This paper presents a fuzzy bilevel programming approach to solve the flow shop scheduling problem. The problem considered here differs from the standard form in that operators are assigned to the machines and imposing a hierarchy of two decision makers with fuzzy processing times. The shop owner considered higher level and assigns the jobs to the machines in order to minimize the flow time while the customer is the lower level and decides on a job schedule in order to minimize the makespan. In this paper, we use the concepts of tolerance membership function at each level to define a fuzzy decision model for generating optimal (satisfactory) solution for bilevel flow shop scheduling problem. A solution algorithm for solving this problem is given.

Keywords: Flow shop, Bilevel programming, Fuzzy decision model

AMS classification: 90C70, 90B36, 90C99

1 Introduction

Bilevel programming problem viewed as a problem with two DMs at two different hierarchical levels. The higher level decision maker (HLDM), the leader, selects his or her decision vector first and the lower level decision maker (LLDM), the follower, select his or her decision afterward based on the decisions of the higher level. The leader knows the functions of the followers, who may or may not know the functions of the leader.

The Stackelberg solution has been employed as a solution approach to bilevel programming problems, and a number of solution algorithms for obtaining the solution have been developed [3,4,15,16]. Recently Lai [10] and Shih, Lai and Lee [13] have proposed a solution concept different from the concept of Stackelberg

CMS Computational Management Science

concept. Brad and Moore have been proposed a branch and bound algorithm for treating the bilevel programming problem [5].

Karlof and Wang [8] developed two level branch and bound algorithm to solve an altered form of the standard flow shop scheduling problem modeled as a bilevel programming problem in a deterministic case. In this paper, we will use the concept of tolerance membership function to obtain the compromise of the various objective functions of the crisp programming problem. In fact the fuzzy decision making which allows various different degree of control is ideally suited for a manufacturing problems like the flow shop scheduling problem. In addition, this approach has the following advantages [17]:

1. The problem is simplified and the representation is more realistic and practical. This is because we are treating a fuzzy and not well defined problem as it is.
2. The use of membership functions to represent the goals of the DMs in different levels offers exceptional flexibility for the decisions proposed.

The flow shop scheduling problem is to process n jobs by m machines where each job has the same ordering of machines. The objective function is to minimize the makespan or others. The structure of the paper is as follows, Sect. 2 contains a brief introduction to bilevel programming. The description of the flow shop scheduling problem with bilevel programming and a solution algorithm to solve it is introduced in Sect. 3. In Sect. 4, we use the concepts of membership functions as well as the bilevel programming flow shop scheduling problem at each level, to develop a fuzzy decision model. In Sect. 5, we develop a solution algorithm to solve the problem of concern. An illustrative example is provided in Sect. 6 to demonstrate the efficiency of the proposed algorithm. Section 7 groups conclusions.

2 Bilevel programming

Bilevel programming (BLP) problem involves two optimization problems where the constraints region of the first problem is implicitly determined by another optimization problem. Bilevel programming, a tool for modeling decentralized decisions, consists of the objective of the leader at its first level and that of the follower at the second level. BLP has been proved to be NP-hard problem.

According to the formulation of Karlof and Wang [8] for the BLP problem, we have decision vectors, x_1 and x_2, where the higher level DM has control over the vector x_1 and the lower level DM has control over the vector x_2. Let S be the set of feasible choices $\{(x_1, x_2)\}$. Let f_1 and f_2 denotes the performance functions for the DMs. The BLP problem can be stated as:

$$\max_{x_1} f_1(x_1, x_2) \text{ higher level} \tag{1a}$$

where x_2 solves

$$\max_{x_2} f_2(x_1, x_2) \text{ lower level} \tag{1b}$$

subject to

$$(x_1, x_2) \in S.$$

Then $S_1 = \{(x_1, x_2^*) \,|\, f_2\ (x_1, x_2^*) = \max_{x_2} f_2(x_1, x_2)\}$is the level one feasible region and S is the level two feasible region.

3 The flow shop scheduling problem

In standard flow shop scheduling problem, n jobs $i_1, \ldots, i_n$ are processed in a shop containing m machines $(j = 1, \ldots, m)$ where each job contains m operations and every job follows the same ordering of machines as it is processed. In this problem, we have many objective functions such as makespan, number of idle machines or total flow time as stated in [1,2,6,11,12]. In this paper we consider makespan and flow time. The makespan is the time from when the first job begins on the first machine until when the last job finishes on the last machine. The flow time of each job is the time from when the first job begins on the first machine until the time when that job finishes on the last machine. The total flow time is the sum of the flow times of the jobs.

Now suppose there are m operators in the shop. The shop owner has to pay the operators based on the total flow time of the jobs, while the customer's charges based on the makespan of the jobs. Thus the objective of the shop owner is to minimize the total flow time while the objective of the customer is to minimize makespan. According to the time schedule, the customer has to decide the jobs ordering. The shop owner is the higher level and the customer (reacting to the shop owner decision) is the lower level as stated in Karlof and Wang [8].

Now let us define some necessary variables to formulate the problem

$mk_{ij} =$ Makespan associated with operator schedule i and job schedule j,
$ft_{ij} =$ Flow time associated with operator schedule i and job schedule j,
$t_{ij} =$ Processing time of the job in the i^{th}position on the j^{th}machine,
$x_{i+1,j} =$ The idle time on the j^{th}machine between the end of the job in the i^{th}position on the j^{th}machine and the start of the job $(i+1)^{th}$position on the j^{th}machine.
$k_i = 1,$ if operator schedule i is chosen; 0, otherwise,
$r_j = 1,$ if job schedule j is chosen; 0, otherwise.

Now we formulate the problem of concern as a bilevel programming problem

$$\min \sum_{i,j} ft_{ij}.r_j.k_i \text{ higher level} \tag{2a}$$

where r_j solves

$$\min \sum_{i,j} mk_{ij}.r_j.k_i \text{ lower level} \tag{2b}$$

subject to

$$\sum_{j=1}^{m!} r_j = 1$$

$$\sum_{j=1}^{n!} k_i = 1$$

$$k_i \in \{0, 1\}, i = 1, ..., n!$$

$$r_j \in \{0, 1\}, j = 1, ..., m!$$

3.1 Determining makespan

For each operator schedule and given n jobs, we have $n!$ orderings to arrange these jobs. We have to find an ordering which minimize the makespan. For any sequence $w = i_1, ..., i_n$, of jobs, let $C_{i_q,j}$ be the completion time of job i_q on machine $m_j, q = 1, ..., n; j = 1, ..., m$. Then since the start time of job i_q on the first machine is the same as the completion time of job i_{q-1} on that machine, $q = 2, ..., n$ and the job i_q can be processed on m_j as soon as possible after both are completed on m_{j-1} and the job i_{q-1} completes on $m_j (q = 2, ..., n; j = 2, ..., m)$, we have the next relations

$$\begin{aligned} &C_{i_q,1} = C_{i_{q-1},1} + t_{i_q,1} \\ &C_{i_q,j} = \max\left[C_{i_q,j-1}, C_{i_{q-1},j}\right] + t_{i_q,j} \\ &q = 1, ..., n; \ j = 2, ..., m \qquad (3) \\ &\text{where} \\ &C_{i_0,j} = 0, \ j = 1, ..., m. \end{aligned}$$

hence the makespan which is denoted by mk is equal to $C_{i_n,m}$ and is calculated by using (3) successively for $j = 1, .., m$ for each $q (q = 1, ..., n)$ in increasing order.

3.2 Determining flow time

The flow time of each job is the time from when the first job begins on the first machine until the time when that job finishes on the last machine. Then the total flow time is the sum of the flow times of each job. We can see that:
the flow time of the job in the first position is

$$ft_1 = x_{1m} + t_{1m}$$

the flow time of the job in the second position is

$$ft_2 = x_{1m} + t_{1m} + x_{2m} + t_{2m}$$

the flow time of the job in the n^{th} position is

$$ft_n = x_{1m} + t_{1m} + x_{2m} + t_{2m} + ... + x_{nm} + t_{nm}$$

therefore the total flow times is given by:

$$ft = \sum_{i=1}^{n} (n + 1 - i)(x_{im} + t_{im}). \tag{4}$$

Let k be the job sequence and then $k + 1$ represents the next sequence. For a particular sequence k, let mk_k and ft_k be the makespan and flow time of that sequence respectively. To determine the idle times, we have the following three cases:

(a) When $C_{i_q,j-1} > C_{i_{q-1},j}$ holds there arises an idle time of the machine m_j of the amount

$$x_{i_q,j-1} \equiv C_{i_q,j-1} - C_{i_{q-1},j}, q = 2, ..., n; \quad j = 2, ..., m$$

(b) When $C_{i_q,j-1} < C_{i_{q-1},j}$ holds there arises a waiting time for processing on machine m_j of the job i_q of the amount

$$W_{i_q,j} \equiv C_{i_{q-1},j} - C_{i_q,j-1}, q = 2, ..., n; \quad j = 2, ..., m$$

(c) When $C_{j-1,i_q} = C_{i_{q-1},j}$ holds they become

$$x_{i_q,j-1} = 0, q = 2, ..., n; j = 2, ..., m.$$

Note that $x_{1i} = 0, 1 \leq i \leq n$. Let l denotes the operator schedule, the notation $l = l + 1$ means go to the next sequence of operator schedule. Now we summarize an algorithm steps to determine the makespan and its corresponding flow time.

Algorithm 1

Step 1. Set l equal to the first operator schedule.
Step 2. Set k equal to the first job sequence.
Step 3. Determine the makespan mk_k by using the relations (3).
Step 4. Calculate the idle time as follows
Do $q = 2, n$
(a) If $C_{i_q,j-1} > C_{i_{q-1},j}$ then set

$$x_{i_q,j-1} \equiv C_{i_q,j-1} - C_{i_{q-1},j}, j = 2, ..., m.$$

(b) If $C_{i_q,j-1} < C_{i_{q-1},j}$ holds, then there is no idle time.
(c) If $C_{j-1,i_q} = C_{i_{q-1},j}$ holds then set

$$x_{i_q,j-1} = 0, j = 2, ..., m.$$

End do

Step 5. Determine the total flow time by using equation (4).
Step 6. Set $k = k + 1$, if k exceeds last sequence go to step 7; otherwise go to step 3.
Step 7. Set $l = l + 1$, if $l \leq$ the last operator schedule go to step 2.
Step 8. Stop.

4 Fuzzy bilevel programming

Shih, Lai and Lee [13] introduced the concept of compensatory operators for adjusting the decision making process between the different levels and also between the decision makers of the same level. Sinha [14] suggested a fuzzy mathematical programming approach to obtain the solution of multi-level linear programming problem by using linear membership functions. In our approach, we will use the concept of tolerance membership functions and apply it to the flow shop scheduling problem as follows: Consider the fuzzy decision making process applied to a bilevel programming. The HLDM specifies the preferred values of his or her variables and goals with certain amount of tolerance. This information is represented by the use of membership functions and passed to the LLDM. LLDM obtains his or her optimum based on goals and preferences of the higher level and then presents the results to the higher level. If the higher level agrees with the proposed solutions, a final decision is reached and this decision or solution is referred as a satisfactory solution. If he or she rejects this proposal, the DMs in both levels will need to re-evaluate and changes the goals and decision as well as their corresponding tolerances. This process is continued until the satisfactory solution is reached. This strategy is very flexible. Since the DMs in both levels first seek their optimal solutions in isolation, it does not violate the non-cooperative idea [7,9].

4.1 HLDM problem

First, HLDM solves the following flow shop scheduling problem:

$$\min_{\bar{t}} ft(\bar{t}) = \sum_{i=1}^{n} (n + 1 - i)(x_{im} + \bar{t})r_j k_i \tag{5}$$

subject to

$$\sum_{j=1}^{m!} r_j = 1,$$

$$\sum_{j=1}^{n!} k_i = 1,$$

$$k_i \in \{0, 1\}, i = 1, ..., n!$$

$$r_j \in \{0, 1\}, j = 1, ..., m! \quad \bar{t} > 0,$$

where $\bar{t} = (t_{im}, i = 1, ..., n), \bar{t} \in R^n$.

To build membership function, goals and tolerances should be determined first. However, they could hardly be determined without meaningful supporting data. For the flow time objective function, we should first find individual best solution ft^* and individual worst solution ft^- where

$$ft^* = \min_{\bar{t}} ft(\bar{t}),\ ft^- = \max_{\bar{t}} ft(\bar{t}). \tag{6}$$

Goals and tolerances can then be reasonably set for best solution. This data can then be formulated as the following membership function of fuzzy set theory

$$\mu_{ft}[ft\left(\bar{t}\right)] = \begin{cases} 1,\ \text{if } ft(\bar{t}) < ft^* \\ \dfrac{\left(ft^- - ft(\bar{t})\right)}{\left(ft^* - ft^-\right)}, \quad \text{if } ft^* \leq ft(\bar{t}) \leq ft^- \\ 0,\ \text{if } ft(\bar{t}) \geq ft^*. \end{cases} \tag{7}$$

Now, we can get the solution of the HLDM problem by solving the following Tchebycheff problem [17]:

$$\max \lambda \tag{8}$$

subject to

$$\begin{aligned} &\mu_{ft}[ft(\bar{t})] \geq \lambda, \\ &\lambda \in [0, 1], \\ &\sum_{j=1}^{m!} r_j = 1, \\ &\sum_{j=1}^{n!} k_i = 1, \\ &k_i \in \{0, 1\}, i = 1, ..., n! \\ &r_j \in \{0, 1\}, j = 1, ..., m! \end{aligned}$$

whose solution is assumed to be $[\bar{t}^H, ft^H, \lambda^H]$.

4.2 LLDM problem

Second, in the same way, the LLDM independently solves:

$$\min_{\bar{t}} mk(\bar{t}) = \sum_{i,j} (x_{im} + \bar{t}) r_j k_i \tag{9}$$

subject to

$$\sum_{j=1}^{m!} r_j = 1,$$
$$\sum_{j=1}^{n!} k_i = 1,$$
$$k_i \in \{0, 1\}, i = 1, ..., n!$$
$$r_j \in \{0, 1\}, j = 1, ..., m!$$
$$\bar{t} > 0, i = 1, ..., n$$

where $\tilde{t} = (t_{im}, i = 1, ..., n), \tilde{t} \in R^n$.

For the makespan objective function, the individual best solution mk^* and individual worst solution mk^- where

$$mk^* = \min_{\bar{t}}\ mk,\ \ mk^- = \max_{\bar{t}} mk. \tag{10}$$

This information can then be formulated as the following membership functions of fuzzy theory:

$$\mu_{mk}[mk(\bar{t})] = \begin{cases} 1,\ \text{if}\ mk(\bar{t}) < mk^* \\ \dfrac{\left(mk^- - mk(\bar{t})\right)}{\left(mk^* - mk^-\right)}, & \text{if}\ mk^* \leq mk(\bar{t}) \leq mk^- \\ 0,\ \text{if}\ mk(\bar{t}) \geq mk^-. \end{cases} \tag{11}$$

We can now get the solution of LLDM problem by solving the following Tchebycheff problem:

$$\max \beta \tag{12}$$

subject to

$$\mu_{mk}[mk(\bar{t})] \geq \beta, i = 1, ..., n$$
$$\beta \in [0, 1],$$
$$\sum_{j=1}^{m!} r_j = 1,$$
$$\sum_{j=1}^{n!} k_i = 1,$$
$$k_i \in \{0, 1\}, i = 1, ..., n!$$
$$r_j \in \{0, 1\}, j = 1, ..., m!.$$

Whose solution is assumed to be $[\bar{t}^L, mk^L, \beta^L]$.

Now the above solution of HLDM and LLDM are disclosed. However, two solutions are usually different because of conflicts of nature between two levels objective functions [15]. The HLDM knows that using the optimal decision $\bar{t}^H$ as a control factors for LLDM is not practical. It is more reasonable to have some tolerance that gives the LLDM an extent feasible region to search for his or her optimal solution, and also reduce searching time or interactions. In this way, the range of the decision $\bar{t}$ should be around $\bar{t}^H$ with its maximum tolerance $\bar{t}^1$ and the following membership function can be stated as:

$$\mu'_{\bar{t}}(\bar{t}) = \begin{cases} \dfrac{\bar{t} - (\bar{t}^H - \bar{t}^1)}{t^1_{im}}, \text{ if } \bar{t}^H - \bar{t}^1 \leq \bar{t} \leq \bar{t}^H; \\ \dfrac{(\bar{t}^H + \bar{t}^1) - \bar{t}}{\bar{t}^1}, \text{ if } \bar{t}^H \leq \bar{t} \leq \bar{t}^H + \bar{t}^1; \\ 0, \text{ otherwise.} \end{cases} \tag{13}$$

where $\bar{t}^H$ is the most preferred solution, the $(\bar{t}^H - \bar{t}^1)$ and $(\bar{t}^H + \bar{t}^1)$ is the worst acceptable decision and that satisfaction is increasing within the interval of $[\bar{t}^H - \bar{t}^1, \bar{t}^H]$ and decreasing within $[\bar{t}^H, \bar{t}^H + \bar{t}^1]$, and other decisions are not acceptable.

In order to supervise the LLDM to search for solutions in the right direction, the HLDM should define his or her goal with tolerance to the LLDM. The HLDM's goals may reasonably consider that $ft < ft^H$ is absolutely acceptable and $ft \geq ft^H$ is absolutely unacceptable. This is due to the fact that the LLDM obtained the optimum at $(\bar{t}^L)$, which in turn provides the HLDM the objective function value ft', makes any $ft > ft'$ unattractive in practice. The following membership function of the HLDM can be stated as:

$$\mu'_{ft}[ft(\bar{t})] = \begin{cases} 1, \text{ if } ft(\bar{t}) < ft^H; \\ \dfrac{\left(ft' - ft(\bar{t})\right)}{\left(ft^H - ft'\right)}, \text{ if } ft' \leq ft(\bar{t}) < ft^H; \\ 0, \text{ if } ft(\bar{t}) \geq ft'. \end{cases} \tag{14}$$

For each possible solution available to the HLDM, the LLDM may be willing to build a membership function for his or her objective function so that he or she can rate the satisfaction of each solution. The LLDM has the following membership function for his/her goals:

$$\mu'_{mk}[mk(\bar{t})] = \begin{cases} 1, \text{ if } mk(\bar{t}) < mk^L; \\ \dfrac{\left(mk' - mk(\bar{t})\right)}{\left(mk^L - mk'\right)}, \text{ if } mk' \leq mk(\bar{t}) < mk^L; \\ 0, \text{ if } mk(\bar{t}) \geq mk'. \end{cases} \tag{15}$$

where $mk' = mk[(\bar{t})]$. Because of mk^L is the best solution of (12), $mk(\bar{t}) < mk^L$ is impossible while the HLDM gives more constraints to the LLDM. The LLDM will not accept any $mk(\bar{t}) > mk^L$ for some reason as the HLDM, stated above [13].

Now, in order to generate the satisfactory solution for both DM's, we can solve the following Tchebycheff problem:

$$\max \delta \tag{16}$$

subject to

$$\begin{aligned}
&[(\bar{t}^H + \bar{t}^1) - \bar{t}]/\bar{t}^1 \geq \delta,\\
&[\bar{t} - (\bar{t}^H - \bar{t}^1)]/\bar{t}^1 \geq \delta,\\
&\mu'_{ft}[ft(\bar{t})] = \frac{[ft' - ft(\bar{t})]}{\left[ft^H - ft'\right]} \geq \delta,\\
&\mu'_{mk}[mk(\bar{t})] = \frac{[mk' - mk(\bar{t})]}{\left[mk^L - mk'\right]} \geq \delta,\\
&\delta \in [0, 1],\\
&\sum_{j=1}^{m!} r_j = 1,\\
&\sum_{j=1}^{n!} k_i = 1,\\
&k_i \in \{0, 1\}, i = 1, ..., n!\\
&r_j \in \{0, 1\}, j = 1, ..., m!.
\end{aligned}$$

Where δ is the overall satisfaction. By solving problem (16), if the HLDM is satisfied with this solution, then a satisfactory solution is reached. Otherwise, he or she should provide new membership functions for the fuzzy variables and objectives to LLDM until a satisfactory solution is reached.

5 The bilevel programming algorithm

First, we have to solve the HLDM and LLDM separately. Two solutions are usually different because of conflicts of nature between two levels objective functions. In our proposed algorithm, if the HLDM satisfied with the solution of problem (16), (we denote to this solution by $(\bar{t}^0, ft^0, mk^0)$) a satisfactory solution is reached. If not satisfied, the high level should provide a new membership functions for the decision variables and the objective to the lower level. This process is continued until a satisfactory solution is reached. Combined with the set of decisions and the tolerances, this solution becomes a satisfactory solution of problems (2a) and (2b). The proposed algorithm can be summarized in the following steps.

Algorithm 2.

Step 1. The HLDM and the LLDM solves his or her problem (2a) and (2b) respectively by using Algorithm 1.

Step 2. The HLDM solves his or her problem as follows:

(2a). Find individual solution by solving (5) and (6), we get ft^* and ft^-.

(2b). By using (7), build the membership functions $\mu_{ft}(\bar{t})$, then solve (8) whose solution is $[\bar{t}^H, ft^H, \lambda^H]$.

Step 3. The LLDM solves his or her problem as follows:

(3a). Find individual solution by solving (9) and (10), we get mk^* and mk^-.

(3b). By using (11), build the membership functions $\mu_{mk}(\bar{t})$, then solve (12) whose solution is $[\bar{t}^L, mk^L, \beta^L]$.

Step 4. If $[\bar{t}^H, ft^H, \lambda^H] = [\bar{t}^L, mk^L, \beta^L]$, the optimal or preferred solution of the system is obtained. Stop. Otherwise, go to Step 5.

Step 5. The HLDM decides his or her tolerances on the goal and the decisions in terms of membership functions by using (13) and (14). Meanwhile, the LLDM also decides his or her tolerance on the goal in terms of the membership functions by using (15). These membership functions will serve as extra constraints in forming problem (16).

Step 6. Solve problem (16). If the DMs at each level are satisfied with the solution, an optimal (satisfactory) solution is reached. Stop. Otherwise, go to step 5 to obtain new membership functions.

6 Illustrative example

Consider a simple example with three machines (and three operators) and three jobs. The time table of the three operators is as follows:

		Job 1	Job 2	Job 3
Operator 1	Machine 1	1	6	5
	Machine 2	4	2	7
	Machine 3	10	8	4
Operator 2	Machine 1	9	7	3
	Machine 2	3	9	5
	Machine 3	2	3	6
Operator 3	Machine 1	6	8	2
	Machine 2	7	8	5
	Machine 3	6	6	1

Let $l = 123$ be the first operator schedule and $k = 123$ be the first job schedule. Makespan of this sequence is calculated as in the following table

i_q	1	2	3
$C_{i_q,1}$	1	7	12
$C_{i_q,2}$	4	16	21
$C_{i_q,3}$	10	16	23

Then the Makespan at this sequence which is denoted by mk_1 is equal to 23. From the above table we get $x_{13} = 4$, $x_{23} = 6$, $x_{33} = 0$, then

$$ft = \sum_{i=1}^{3} (3 + 1 - i)(x_{i3} + t_{i3}) = 3(4 + 6) + 2(6 + 6) + 1 = 55.$$

By the same way, we calculate the makespan of the sequences $k = 132, 213, 231, 312$ and 321 as in the following tables

$k = 132$

i_q	1	3	2
$C_{i_q,1}$	1	6	12
$C_{i_q,2}$	4	11	21
$C_{i_q,3}$	10	12	27

$mk_2 = 27$ and $ft_2 = 65$

$k = 213$

i_q	2	1	3
$C_{i_q,1}$	6	7	12
$C_{i_q,2}$	15	18	23
$C_{i_q,3}$	21	27	28

$mk_4 = 29$ and $ft_4 = 100$

$k = 231$

i_q	2	3	1
$C_{i_q,1}$	6	11	12
$C_{i_q,2}$	15	20	23
$C_{i_q,3}$	21	22	29

$mk_3 = 28$ and $ft_3 = 87$

$k = 312$

i_q	3	1	2
$C_{i_q,1}$	5	6	12
$C_{i_q,2}$	10	13	22
$C_{i_q,3}$	11	19	28

$mk_5 = 28$ and $ft_5 = 76$

$k = 321$

i_q	3	2	1
$C_{i_q,1}$	5	11	12
$C_{i_q,2}$	10	20	23
$C_{i_q,3}$	11	26	32

$mk_6 = 32$ and $ft_6 = 100$

and by the same way we can calculate the rest of job schedules for each operator schedules ($l = 132, 213, 231, 312$ and 321). We find that minimum flow time is equal to 55 and minimum makespan equal to 23 which are achieved at $l = 123$ and $k = 123$. Then go to Algorithm 2.

First, the HLDM solves his or her problem as follows:

1. Finds the optimal solution by solving (5) and (6), we get

$$(ft^*, ft^-) = (55, 109)$$

2. By using (7), build membership function $\mu_{ft}(\bar{t})$ then solve (8) as follows:

$$\max \lambda$$

subject to

$$\begin{aligned}
&-3t_{13} - 2t_{23} - t_{33} + 54\lambda \geq -85,\\
&\lambda \in [0, 1],\\
&t_{13}, t_{23}, t_{33} > 0
\end{aligned}$$

whose solution is $(t_{13}^H, t_{23}^H, t_{33}^H) = (1, 1, 1)$, $ft^H = 30$ and $\lambda^H = 1$.

Second, the LLDM solves his or her problem as follows:

1. Finds the optimal solution by solving (9) and (10), we get

$$(mk^*, mk^-) = (23, 32)$$

2. By using (11), build membership function $\mu_{mk}(\bar{t})$ then solve (12) as follows:

$$\max \beta$$

subject to

$$\begin{aligned}
&-t_{13} - t_{23} - t_{33} + 9\beta \geq -22,\\
&\beta \in [0, 1],\\
&t_{13}, t_{23}, t_{33} > 0
\end{aligned}$$

whose solution is $(t_{13}^L, t_{23}^L, t_{33}^L) = (1, 1, 2)$, $mk^- = 14$ and $\beta^L = 1$.

Third,

1. Assume the HLDM's control decision variables $t_{i3}^H, i = 1, ..., 3$ is around 0 with the tolerances 1, 2, and 3 respectively.
2. By (13), (14) and (15), build membership functions $\mu'_{\bar{t}}(\bar{t})$, $\mu'_{ft}(\bar{t})$, $\mu'_{mk}(\bar{t})$. The LLDM then solves the following problem of (16):

$$\max \delta$$

subject to

$$
\begin{aligned}
&t_{13} - \delta \geq 0,\\
&-t_{13} - \delta \geq -2,\\
&t_{23} - 2\delta \geq -1,\\
&-t_{23} - 2\delta \geq -3,\\
&t_{33} - 3\delta \geq -2,\\
&-t_{33} - 3\delta \geq -4,\\
&-3t_{13} - 2t_{23} - t_{33} + \delta \geq -7,\\
&-t_{13} - t_{23} - t_{33} - \delta \geq -3,\\
&t_{13}, t_{23}, t_{33} > 0,\\
&\delta \in [0, 1].
\end{aligned}
$$

Whose solution is

$(t^0_{13}, t^0_{23}, t^0_{33}) = (0.86, 0.71, 0.57)$, $(ft^0, mk^0) = (28.57, 12.14)$ and $\delta = 0.86$.

If the HLDM is satisfied with the above solution, then a satisfactory solution is reached. Otherwise, he or she should provide new membership functions for the control variable and objectives to the LLDM until a satisfactory solution is reached.

7 Conclusions

This paper has proposed a bilevel flow shop scheduling model and a solution algorithm for solving this problem. This solution uses the concepts of tolerance membership functions at each level to develop a fuzzy decision model for generate satisfactory solution for the problem of concern.

Based on Lai's satisfactory solution concepts, the proposed solution algorithm proceeds from higher level (shop owner) to lower level (customer) in a natural and straightforward manner. The HLDM specifies his or her objectives and decisions with possible leeway, which are described by membership function of fuzzy set theory. This information then constraints the LLDM feasible space. The novelties of our approach are mainly concentrated in the use of tolerance membership functions at each level and the introduction of an algorithm to solve a bilevel flow shop scheduling problem using these functions. An illustrative example has been given to demonstrate the efficiency of the proposed algorithm.

However, there are some open points in this area such as:

1. Fuzzy approach is needed for multi-level flow shop scheduling problem.
2. On the basis of the proposed method, other membership functions such as piecewise, exponential, hyperbolic, hyperbolic-inverse or some specific power functions may be needed for practical reasons.

References

[1] Achugbue J, Chin F (1982) Scheduling the open shop to minimize mean flow time. SIAM J Comput 11: 709–720.
[2] Ahmadi R, Bagchi U (1992) Minimizing job idleness on deadline constrained environments. Opns Res 40: 972–985.
[3] Bard JF (1983) An efficient point algorithm for a linear two-stage optimization problem. Opns Res 38: 556–560.
[4] Bialas WF, Karwan MH (1984) Two-level linear programming. Manag Sci 30: 1004–1020.
[5] Brad J, Moore J (1990) A branch and bound algorithm for the bilevel programming problem. SIAM J Scient Statist Comput 11: 281–292.
[6] Deckro RJ Hebert and Winkofsky E (1982) Multiple criteria job-shop scheduling. Computers Ops Res 9: 279–285.
[7] Hsu-Shih Shih E, Lee ES (2000) Compensatory fuzzy mutiple level decision making. Fuzzy Sets and Systems 114: 71–87.
[8] Karlof John K, Wang W (1996) Bilevel programming applied to the flow shop scheduling problem. Computers Ops Res 23: 443–451.
[9] Lee ES, Hsu-Shih Shih (1999) Fuzzy and multi-level decision making. London Berlin Heidelberg: Springer
[10] Lai YJ (1996) Hierarchical optimization: a satisfactory solution. Fuzzy Sets and Systems 77: 321–335.
[11] Liao C (1993) Minimizing the number of machine idle intervals with minimum makespan in a flow shop. J Opl Res Soc 44: 817–824.
[12] Selen W, Hott D (1986) A mixed-integer goal-programming formulation of the standard flow-shop scheduling problem. J Opl Res Soc 37: 1121–1128.
[13] Shih HS, Lai YJ, Lee ES (1996) Fuzzy approach for multi-level mathematical programming problems. Computers Ops Res 23: 73–91.
[14] Sinha S (2003) Fuzzy programming approach to multi-level programming problems. Fuzzy Sets and Systems 136: 189–202.
[15] Wen UP, Bialas WF (1991) The hybrid algorithm for solving the three-level linear programming problem. Computers Ops Res 13: 367–377.
[16] White DJ, Anandalingam G (1993) A penalty function approach for solving bi-level linear programs. J Global Optim 3: 397–419.
[17] Zimmermann HJ (1987) Fuzzy sets, decision making, and expert systems. Boston: Kluwer Academic.