Scheduling with Fuzzy Processing Times in a Flow Shop

Milos Seda, Jiri Dvorak and Petr Majer
Technical University of Brno, Faculty of Mechanical Engineering
Institute of Automation and Computer Science
Technická 2, CZ-616 69 Brno, Czech Republic
Phone: +420-5-41143342, Fax: +420-5-744979
email:{mseda, jdvorak, ymajer00}@kinf.fme.vutbr.cz

ABSTRACT: This paper deals with a permutation flow shop scheduling problem where processing times are given by trapezoidal fuzzy numbers and a makespan is minimised. It describes possible approaches to representing and solving this problem as a multicriteria deterministic problem. Proposed approaches included the use of stochastic heuristic methods (simulated annealing, tabu search, genetic algorithms). Computational results for a selected case of multi-objective problem are presented.

KEYWORDS: flow shop scheduling, fuzzy processing times, heuristic methods

1. INTRODUCTION

Scheduling problems we can meet in many practical applications including *production management* [Blazewicz *et al*. (1996)] (*flow shop scheduling*, *job shop scheduling*), *project management*, *timetabling*, etc. The common feature of these problems is their combinatorial character. In practical problems the space of feasible solutions is so large that it is impossible to search the optimal solution by testing each feasible solution. From this reason heuristic methods are preferred to searching a solution, and among them especially stochastic methods (genetic algorithms, simulated annealing and tabu search) are used.

A scheduling problem is more complex when some data (processing times of operations, due dates of jobs) are uncertain. Many approaches to scheduling under uncertainty are based on the use of fuzzy set theory. Processing times can be represented by fuzzy intervals [Cury *et al*. (1998), Dubois *et al*. (1995), McCahon and Lee (1992)], due dates can be specified by membership functions expressing grades of satisfaction of a decision maker.

In [Dubois *et al*. (1995), Romero and Yamakami (1997)], a fuzzy approach is applied to job shop scheduling. The articles [Cury *et al*. (1998), Ischibuchi, Yamamoto *et al*. (1994), Ischibuchi, Tanaka and Misaki (1994)] describe fuzzy set applications to flow shop scheduling. The report [Kerr and Slany (1994)] provides a large survey of the application of fuzzy sets to scheduling.

This paper is concentrated on a flow shop scheduling problem where processing times are given by trapezoidal fuzzy numbers. In the next section, a mathematical model of this problem is described and possible approaches to constructing and solving a corresponding multicriteria deterministic problem are proposed. The section 3 presents computational results of stochastic heuristic methods applied to one of these multi-objective problems.

2. FLOW SHOP SCHEDULING WITH FUZZY PROCESSING TIMES

The aim of the manufacturing scheduling is to find the best sequence of jobs on machines under given constraints (technological conditions, available amount of production resources, etc.). Various decision criteria can be used including makespan (i.e. the maximum of the completion times of all jobs), mean flowtime, total tardiness and so on. These criteria are of the minimisation type.

In this section, we deal with *permutation flow shop scheduling* where n jobs must be processed in the same order on m machines satisfying the following constraints: (i) Each job comprises a set of m operations which must be done on a different machines; (ii) All jobs have the same processing operation order through the machines; (iii) Operations cannot be interrupted; (iv) Each machine can process only one operation at a time; (v) There is no precedence constraint among operations of different jobs. The aim is to find a permutation of jobs which optimises the specified objective function. In this paper we consider the makespan as the objective.

Let $J = \{1,\ldots, n\}$, $M = \{1,\ldots,m\}$, J_i = ith job in permutation, p_{ik} = processing time for job i on machine k, C_{ik} = completion time for job i on machine k, $C_{\max}(p)$ = completion time for job permutation p. If a job permutation $\pi = \{J_1, J_2, \ldots, J_n\}$ is given and processing times of jobs J_i are given as fuzzy numbers $p^F_{J_i,k}$, then we calculate the fuzzy completion times $C^F_{J_i,k}$ as follows:

$$C^F_{J_1,1} = p^F_{J_1,1} \quad (1)$$

$$\forall i \in J - \{1\}: C^F_{J_i,1} = C^F_{J_{i-1},1} \oplus p^F_{J_i,1} \quad (2)$$

$$\forall k \in M - \{1\}: C^F_{J_1,k} = C^F_{J_1,k-1} \oplus p^F_{J_1,k} \quad (3)$$

$$(\forall i \in J - \{1\})(\forall k \in M - \{1\}): C^F_{J_i,k} = \widetilde{\max}\{C^F_{J_{i-1},k}, C^F_{J_i,k-1}\} \oplus p^F_{J_i,k} \quad (4)$$

$$C^F_{\max}(p) = C^F_{J_n,m} \quad (5)$$

where $\oplus$ and $\widetilde{\max}$ are operations over fuzzy numbers.

Definition: A *fuzzy number* A is a fuzzy set represented by 4-tuple (a_1, a_2, a_3, a_4) and a piecewise continuous membership function with the following properties [Novák(1989)]: (i) $(a_1 \le a_2 \le a_3 \le a_4)$; (ii) $\mu_A(x) = 0$ for $x \le a_1$, $x \ge a_4$; (iii) $\mu_A(x) = 1$ for $a_2 \le x \le a_3$; (iv) $\mu_A(x)$ is increasing on $[a_1, a_2]$ and decreasing on $[a_3, a_4]$.

In this section we consider *trapezoidal fuzzy numbers* (see Fig. 1) given by the membership function (6). A special case, when $a_2 = a_3$, is called a *triangular fuzzy number*.

$$m_A(x) = \begin{cases} 0 & , \text{ if } 0 < x \le a_1 \\ \dfrac{x - a_1}{a_2 - a_1} & , \text{ if } a_1 < x \le a_2 \\ 1 & , \text{ if } a_2 < x \le a_3 \\ \dfrac{a_4 - x}{a_4 - a_3} & , \text{ if } a_3 < x \le a_4 \\ 0 & , \text{ if } x > a_4 \end{cases} \quad (6)$$

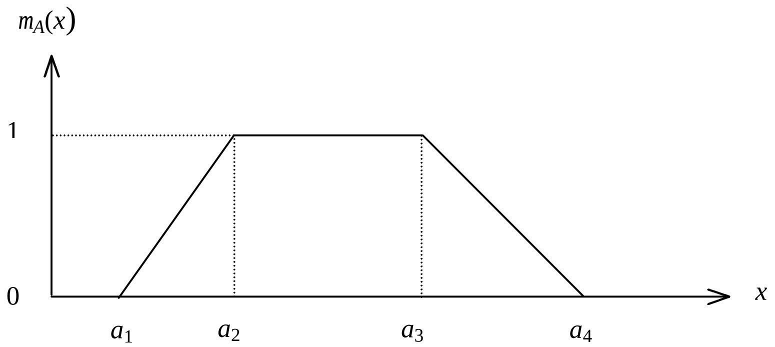

Figure 1: Trapezoidal fuzzy number

The addition of fuzzy numbers can be derived using the extension principle and it is determined as follows [Dubois and Prade (1988)]:

$$X^F \oplus Y^F = (x_1, x_2, x_3, x_4) \oplus (y_1, y_2, y_3, y_4) = (x_1 + y_1, x_2 + y_2, x_3 + y_3, x_4 + y_4) \quad (7)$$

When the maximum operation is derived in the same way, then its results need not be trapezoidal fuzzy numbers. Therefore we approximate this operation as follows [Dubois and Prade (1988)]:

$$\widetilde{\max}(X^F, Y^F) = (\max(x_1, y_1), \max(x_2, y_2), \max(x_3, y_3), \max(x_4, y_4)) \tag{8}$$

To find a job permutation p which minimises the fuzzy makespan, we must compare fuzzy numbers in some way. This is a difficult problem. The *ordering relation* $\preceq$ can be defined e.g. as follows:

$$X^F \preceq Y^F \Leftrightarrow (x_1 \le y_1) \wedge (x_2 \le y_2) \wedge (x_3 \le y_3) \wedge (x_4 \le y_4) \tag{9}$$

Defined relation, however, is not complete ordering relation, as fuzzy numbers X^F, Y^F satisfying

$$(\exists i, j \in \{1, 2, 3, 4\}) : (x_i < y_i) \wedge (x_j > y_j) \tag{10}$$

are not in the $\preceq$ relation. If $C^F_{\max}(p_1)$ and $C^F_{\max}(p_2)$ are not in the $\preceq$ relation then we say that the solutions p_1, p_2 of our scheduling problem are *non-dominated*.

It is straightforward that studied problem of the minimisation of $C^F_{\max}(p)$ can be replaced by the following four-criteria problem:

$$\text{Minimise } C_{\max,1}(p) \text{ and } C_{\max,2}(p) \text{ and } C_{\max,3}(p) \text{ and } C_{\max,4}(p) \tag{11}$$

where objective functions $C_{\max,\,j}(p)$ are deterministic. There are various techniques of the multicriteria optimization. In [Murata *et al*. (1996)] the genetic algorithm for searching non-dominated solutions of the two-criteria problem is described (that paper deals with a flow shop scheduling problem, where processing times are given as crisp intervals). It is possible to search a compromise solution of our problem on the basis of its transformation into the single-criterion problem where objective function can be designed as a weighted sum of criteria:

$$C_{\max}(p) = \sum_{j=1}^{4} w_j C_{\max,\,j}(p) \tag{12}$$

or as a distance between the vector of these criteria and a vector of ideal or given values. Values of such objective function then can be used in a common way in simulated annealing, tabu search or genetic algorithms.

In [McCahon and Lee (1992)] another approach to the comparison of fuzzy numbers is proposed. It uses generalised mean values and spread values for this reason. These values are defined for the fuzzy number A in the universe U as follows:

$$m(A) = \frac{\int_U x\,\mu_A(x)\,dx}{\int_U \mu_A(x)\,dx}, \qquad s(A) = \sqrt{\frac{\int_U x^2\,\mu_A(x)\,dx}{\int_U \mu_A(x)\,dx} - [m(A)]^2} \tag{13}$$

The fuzzy number with higher generalised mean value (GMV) is ranked higher than the fuzzy number with the lower GMV. If the GMVs happen to be equal, the spread $s(A)$ is calculated for each fuzzy number and the one with the smaller spread is judged the smaller. This way of comparison corresponds to a lexicographic approach of multicriteria programming with criteria $m(A)$ and $s(A)$. We can also apply a goal programming approach and minimise a distance between the vector $(m(C^F_{\max}), s(C^F_{\max}))$ and a goal vector (G_1, G_2). This problem can be stated e.g. as follows:

$$\text{Minimise } \left(w_1\left|G_1 - m(C^F_{\max})\right| + w_2\left|G_2 - s(C^F_{\max})\right|\right) \tag{14}$$

The goal G_1 can be obtained as a value of the $C_{\max}(p^H)$ where p^H is a heuristic solution of the deterministic problem with crisp processing times p_{ik} derived from fuzzy processing times $p^F_{ik} = (p_{i,k,1}, p_{i,k,2}, p_{i,k,3}, p_{i,k,4})$. It may be $p_{ik} = p_{i,k,1}$ or $p_{ik} = p_{i,k,2}$ or $p_{ik} = m(p^F_{ik})$. The goal G_2 can be equal to zero or it can be determined as the minimum of $s(p^F_{ik})$.

3. COMPUTATIONAL RESULTS

In this section, we present selected computational results for the two-criteria problem with the criteria (13). This problem was solved by the lexicographic approach combined with tabu search, simulated annealing, and genetic algorithm. Assuming a trapezoidal fuzzy number $A = (a_1, a_2, a_3, a_4)$ with membership function by (6), we get for $m(A)$ and $s(A)$ these expressions:

$$m(A)=\frac{a_4{}^2+a_4a_3+a_3{}^2-a_2{}^2-a_1a_2-a_1{}^2}{3(a_4+a_3-a_2-a_1)}$$

$$s(A)=\sqrt{\frac{a_4{}^3+a_4{}^2a_3+a_4a_3{}^2+a_3{}^3-a_2{}^3-a_2{}^2a_1-a_2a_1{}^2-a_1{}^3}{6(a_4+a_3-a_2-a_1)}-[m(A)]^2}$$

Tested benchmarks were created by symmetric fuzzification of the standard deterministic flow shop scheduling benchmarks from [Beasley (1996)]. Tables 1 and 2 summarize performance statistics of heuristic methods for a subset of fuzzified benchmark flow shop scheduling problems. Table 1 contains fuzzy makespans and Table 2 corresponding mean and spread values. All the methods were programmed in C++ and run on the IBM compatible computer with Pentium processor, 150 MHz and an operational memory 16 MB.

	TS				SA				GA			
n/m	a_1	a_2	a_3	a_4	a_1	a_2	a_3	a_4	a_1	a_2	a_3	a_4
10/6	7110	7458	8297	8620	7091	7381	8238	8533	7131	7471	8339	8658
	7014	*7296*	*8216*	*8435*	*7014*	*7296*	*8216*	*8485*	*7014*	*7296*	*8216*	*8485*
11/5	6368	6713	7368	7708	6368	6713	7368	7708	6470	6816	7512	7859
	6368	*6713*	*7368*	*7708*	*6368*	*6713*	*7368*	*7708*	*6368*	*6713*	*7368*	*7708*
12/5	6507	7077	7725	8306	6553	7099	7743	8308	6551	7104	7738	8315
	6524	*7042*	*7620*	*8272*	*6493*	*7067*	*7731*	*8305*	*6524*	*7042*	*7620*	*8272*
13/4	6668	7009	7560	7905	6557	6932	7416	7814	6686	7046	7582	7944
	6528	*6928*	*7404*	*7805*	*6528*	*6928*	*7404*	*7805*	*6528*	*6928*	*7404*	*7805*
14/4	7263	7637	8415	8790	7240	7617	8391	8769	7380	7753	8564	8938
	7240	*7616*	*8390*	*8766*	*7240*	*7616*	*8390*	*8766*	*7240*	*7616*	*8390*	*8766*
20/5	1170	1226	1376	1434	1231	1290	1431	1491	1175	1231	1382	1441
	1138	*1188*	*1337*	*1395*	*1195*	*1264*	*1374*	*1447*	*1131*	*1186*	*1332*	*1395*
20/10	1491	1557	1689	1756	1535	1610	1736	1815	1490	1555	1693	1759
	1448	*1510*	*1658*	*1723*	*1481*	*1558*	*1728*	*1805*	*1448*	*1510*	*1658*	*1720*
20/15	1853	1937	2078	2161	1946	2028	2184	2272	1847	1929	2071	2152
	1832	*1915*	*2012*	*2104*	*1888*	*1871*	*2148*	*2230*	*1804*	*1889*	*2027*	*2105*
30/10	1954	2053	2234	2334	2014	2114	2304	2407	1920	2020	2199	2293
	1916	*2010*	*2191*	*2278*	*1966*	*2055*	*2243*	*2335*	*1957*	*1972*	*2174*	*2261*
30/15	2457	2577	2813	2950	2517	2642	2887	3029	2411	2528	2752	2883
	2399	*2537*	*2773*	*2916*	*2492*	*2598*	*2845*	*2967*	*2363*	*2471*	*2719*	*2837*
50/10	3026	3159	3429	3592	2946	3071	3336	3487	2939	3066	3331	3480
	2996	*3115*	*3373*	*3532*	*2926*	*3047*	*3311*	*3474*	*2903*	*3015*	*3279*	*3436*
75/20	5124	5387	5826	6107	4768	4998	5438	5696	4820	5064	5493	5764
	4991	*5265*	*5706*	*5988*	*4731*	*4971*	*5380*	*5639*	*4743*	*4993*	*5431*	*5669*

Table 1: Results of benchmarks problems (n jobs/m machines) from 30 runs: Average fuzzy makespans and in italics fuzzy makespans with the best mean values from these runs; SA= simulated annealing, TS=tabu search, GA=genetic algorithm.

Comparing normalised mean values, we can see that results of GA are slightly better than results of other methods, but these differences are not statistically significant. This was verified using the *Kruskal-Wallis test* on the confidence level 0.95.

As an example we only briefly describe the used genetic algorithm with these parameters settings:

- Population size: N_{POP}=50.
- Number of generations: $t_{\max} \approx 10 \times n^2$.

The pseudo-Pascal skeleton of this GA is designed as follows:

$P(0) := \{P_1, P_2, \ldots, P_{Npop}\}$; { initial population of permutations }
$P_{\min}$: = Permutation_minimizing_makespan $\in P(0)$; { the best permutation }
$t := 0$;
while $t <$ number_of_generations **do**
begin **repeat** BinaryTournamentSelection($P(t)$, *parent*1, *parent*2);
offspring : = ModifiedTwoPointCrossover(*parent*1, *parent*2);
offspring : = ShiftMutation(*offspring*)
until not (*offspring* **in** population $P(t)$);
$P_{\max}$: = Permutation_maximizing_makespan $\in P(t)$; { the worst permutation }
$P(t+1) := P(t) - \{P_{\max}\} \cup \{offspring\}$; { SteadyStateReplacement ($P_{\max}$, *offspring*) }
if Makespan(*offspring*) < Makespan($P_{\min}$)
then $P_{\min}$: = *offspring*; { update the best permutation }
$t := t + 1$
end;

A description of other two methods (simulated annealing and tabu search) can be found in [Šeda and Dvoøák (1998)] where the deterministic version of this problem was studied.

			TS		SA		GA	
benchmark	*n*/*m*	*d_opt*	*m*(*A*)	*s*(*A*)	*m*(*A*)	*s*(*A*)	*m*(*A*)	*s*(*A*)
car5	10/6	7720	7870.7	352.6	7810.9	342.4	7899.3	358.5
			7739.1	*345.7*	*7752.5*	*354.2*	*7752.5*	*354.2*
car1	11/5	7038	7039.1	304.5	7039.1	304.5	7164.3	317.1
			7039.1	*304.5*	*7039.1*	*304.5*	*7039.1*	*304.5*
car3	12/5	7312	7404.2	390.3	7426.5	381.6	7427.9	382.6
			7370.1	*376.3*	*7399.0*	*393.9*	*7370.1*	*376.3*
car2	13/4	7166	7285.6	276.4	7180.6	275.0	7314.6	279.1
			7166.3	*278.2*	*7166.3*	*278.2*	*7166.3*	*278.2*
car4	14/4	8003	8026.3	349.8	8004.3	349.8	8158.8	358.5
			8003.0	*349.3*	*8003.0*	*349.3*	*8003.0*	*349.3*
reC01	20/5	1247	1301.5	62.0	1360.8	60.4	1307.3	62.4
			1264.7	*60.6*	*1320.1*	*56.1*	*1261.2*	*61.6*
reC07	20/10	1566	1623.3	60.4	1674.1	62.7	1624.3	61.7
			1584.8	*63.7*	*1643.0*	*74.7*	*1584.0*	*63.2*
reC13	20/15	1930	2007.2	69.1	2107.7	73.8	1999.7	68.7
			1966.1	*59.0*	*2035.1*	*91.0*	*1956.0*	*67.6*
reC23	30/10	2011	2143.8	85.9	2209.8	89.1	2107.8	84.5
			2098.6	*82.6*	*2149.8*	*84.5*	*2092.2*	*75.2*
reC25	30/15	2513	2699.7	111.6	2769.2	115.9	2643.9	106.7
			2656.4	*116.0*	*2725.9*	*109.3*	*2597.8*	*109.2*
reC33	50/10	3114	3302.4	128.1	3210.7	123.0	3204.6	123.0
			3255.2	*121.6*	*3190.7*	*124.3*	*3159.5*	*121.6*
reC27	75/20	4951	5611.6	219.8	5225.8	209.7	5286.1	211.7
			5487.8	*222.5*	*5180.8*	*203.3*	*5208.6*	*209.1*

Table 2: Mean and spread values corresponding to data from Table 1;
d_opt – optimal makespan for deterministic FSS.

4. CONCLUSION

This paper examined the permutation flow shop scheduling problem with fuzzy processing times and makespan objective. Possible solution approaches are based on transforming this problem into a deterministic multi-objective problem and using stochastic heuristic methods (genetic algorithms, simulated annealing, tabu search). The proposed approaches can be simply generalised for a fuzzy job shop scheduling problem by means of disjunctive graph-based representation and fuzzy PERT method. A future research will include a case where fuzzy processing times are combined with fuzzy due dates.

REFERENCES

Beasley, J.E., 1996, "OR-Library", Report, The Management School Imperial College, London, http://mscmga.ms.ic.ac.uk/pub/flowshop1.txt.

Blazewicz, J.; Ecker, K.H.; Schmidt, G.; Weglarz, J., 1996, "Scheduling Computer and Manufacturing Processes", Springer-Verlag, Berlin.

McCahon, C.; Lee, E.S., 1992, "Fuzzy Job Sequencing for a Flow Shop", European Journal of Operational Research, Vol. 62, pp. 294-301.

Cury, R.M; Khator, S.K.; Gauthier, F.A.O.; Barcia, R.M., 1998, "Scheduling Under Uncertainty: A Fuzzy Approach", Proceedings of the 7th Annual Industrial Engineering Research Conference, Banff, Canada, 11 pp.

Dubois, D.; Fargier, H.; Prade, H., 1995, "Fuzzy Constraints in Job-Shop Scheduling", Journal of Intelligent Manufacturing,, Vol. 6, pp. 215-234.

Dubois, D.; Prade, H., 1988, "Theorie des possibilites. Applications a la representation des connaissances en informatique", MASSON, Paris, (Russian translation "Teorija vozmožnostìj. Priloženie k predstavleniju znanij v informatike", Moskva, 1990).

Ishibuchi, H.; Yamamoto, N.; Murata, T.; Tanaka, H., 1994, "Genetic Algorithms and Neighborhood Search Algorithms for Fuzzy Flowshop Scheduling Problems", Fuzzy Sets and Systems, Vol. 67, pp. 81-100.

Ischibuchi, H.; Tanaka, H.; Misaki, S., 1994, "Fuzzy Flow Shop Scheduling Problems by Simulated Annealing", Physica-Verlag, Heidelberg, In: Delgado, M.; Kacprzyk, J.; Verdegay, J.L.; Vila, M.A. (eds.), "Fuzzy Optimization", pp. 351-363.

Kerr, R.M.; Slany, W., 1994, "Research Issues and Challenges in Fuzzy Scheduling", Technical University of Vienna, CD-Technical Report, No. 94/68, 17 pp.

Murata, T.; Ischibuchi, H.; Lee, K.H., 1996, "Application of Two-Objective Genetic Algorithm to Flowshop Scheduling Problems with Interval Processing Time", Proceedings of the Fourth European Congress on Fuzzy and Intelligent Technologies EUFIT '96, Aachen, Vol. 1, pp. 443-447.

Novák, V., 1989, "Fuzzy Sets and their Applications", Adam Hilger, Bristol.

Romero, P.R.; Yamakami, A., 1997, "Job-Shop Scheduling Problem: A General Linear Programming Approach", Proceedings of the Seventh IFSA World Congress, Prague, Vol. III, pp. 43-47.

Šeda, M.; Dvoøák, J., 1998, "An Application of Heuristic Techniques to Permutation Flow Shop Scheduling Problem", Proceedings of the 4th International Conference on Genetic Algorithms, Optimization Problems, Fuzzy Logic, Neural Networks and Rough Sets MENDEL '98, Brno, Czech Republic, pp. 132-139.

Tsujimura, Y. and Gen, M. and Kubota, E., 1995, "Solving Job-Shop Scheduling Problem with Fuzzy Processing Time Using Genetic Algorithm", Journal of Japan Society for Fuzzy Systems, Vol. 7, No. 5, pp. 1073-1083.