International Journal of Hybrid Information Technology
Vol.8, No.7 (2015), pp.305-318
http://dx.doi.org/10.14257/ijhit.2015.8.7.28

Multi-objective Integration of Flexible Collaborative Planning and Fuzzy Flexible Lot-Splitting Scheduling Based on the Pareto Optimal

Zhenqiang Bao[1], *Xiaoqing Ren[1], Yulu Yang[1], Junwu Zhu[1], Cheng Wang[2], Richao Yin[1]

[1]Information Engineering College, Yangzhou University, China
[2]AsiaInfo-Linkage, Inc, Nanjing, China
yzbzq@163.com, xqrenyzu@163.com,1159080933@qq.com, jwzhu@yzu.edu.cn, wangchengyz@126.com, yrcyzu97@163.com

Abstract

Since that a job usually contains several working procedures in actual production, and it's hard to optimize the flexible collaborative planning, the flexible lot splitting and scheduling of the job simultaneously in batch production mode, a model for multi-objective integration of flexible collaborative planning and fuzzy lot-splitting scheduling is established. We take four performance indicators below which are the most common as standards to optimize the model: average delivery satisfaction, fuzzy total cost, fuzzy completion time and average credibility of job tardiness, and then a multi-objective algorithm based on the Pareto optimal is established. In this algorithm, we design the integrated coding scheme, which include collaboration chromosome, lot-splitting chromosome and scheduling chromosome, meanwhile the Pareto optimal scheme is designed. Finally, the efficiency of the model and algorithm is proved by the simulation.

Keywords: *Multi-objective optimization; Flexible collaborative planning; Fuzzy flexible lot splitting; integrated model.*

1. Introduction

Multi-objective integration of flexible collaborative planning and flexible lot-splitting scheduling is the extension of the traditional Job-shop Scheduling Problem. It's more complex and suitable for the practical production. In order to optimize this problem, we need to take another two factors into consideration: on one hand, the flexible collaborative planning whose tasks can be outsourced to partners is different from the traditional one, and its superiority is improved by literature [1-2]; on the other hand, the jobs finished by own often require batch production, the same job can be divided into several lots while being processed, and the number of each lot is variable, so we call it the flexible lot-splitting scheduling. Research of LOW [3-4] shows that in job shop, it's efficient for reducing the idle waiting time of the machine, raising the utilization rate of equipment and shortening the production cycle by batch processing. However the difficulty is how to determine the number of the optimal batch and the quantity of jobs in each lot. So the key to solving the model for multi-objective integration of flexible collaborative planning and flexible lot-splitting scheduling is how to integrate the flexible collaborative planning, the flexible lot splitting and scheduling of the job.

In this paper, we take into consideration that processing time and due date obeyed fuzzy time window distribution as well as the integration of inner and outer production unit. We establish a model for multi-objective integration of flexible collaborative planning and fuzzy lot-splitting scheduling. The four optimization objectives of the model are average delivery satisfaction, fuzzy total cost, fuzzy completion time and average

ISSN: 1738-9968 IJHIT

credibility of job tardiness, and then a multi-objective algorithm based on the Pareto optimal is established. Finally, the multi-objective integration of flexible collaborative planning and fuzzy lot-splitting scheduling based on the Pareto optimal will really come true.

2. Related Works

Nowadays, there are not too many researches on integrated optimization of batch production. LOW [3] and Pan Quanke [4] proved the efficiency of batch processing in job shop by simulation, it could reduce the idle waiting time of the machine and shorten the production cycle, but they didn't tell us how to split up the batch; JEONG [5] proposed an algorithm for dynamic lot-splitting scheduling. However this algorithm applied only to JSP with a single process route, and it was hard to solve the flexible JSP; CHAN [12-13] adopted two kinds of Genetic Algorithm to optimize the lot splitting and scheduling of the job separately, so the optimal solution was unpersuasive; Sun Zhijun [6] utilized the strategy of the same lot size to optimize the quantity and process sequence of each sub-job, but the batch couldn't be adjusted to the machine load flexibly; Bai Junjie [7] proposed a flexible lot-splitting method based on 'cursor'. At the same time, he adopted a particle coding scheme, which integrated the lot splitting and scheduling. In this way, he could optimize them simultaneously. However, he integrated them just from the point of view of the inner production unit, not take into account the collaborative planning and scheduling with their partners under the circumstance of supply chain, and he didn't introduce the study to the fuzzy production environment which is closer to the actual production. Model in this paper is designed to solve these problems.

3. Fuzzy Measure of Key Performance Indicators in the Model

This paper adopts the triangular fuzzy numbers represent fuzzy processing time, the trapezoidal fuzzy numbers represent fuzzy due date, which was introduced by literature [8]. We apply the addition operation, maximum operation and comparison operation to fuzzy numbers, which was defined by literature [9]. At the same time, we need to extend the corresponding traditional fuzzy scheduling theory. In this section, the fuzzy number operators and calculation methods for each objective are described.

3.1. Satisfaction Index

Satisfaction index which is proposed by Masatoshi Sakawa[9-10] means the ratio of the area which belongs to the membership degree function of the fuzzy completion time to the intersected area. Since the model includes the collaborative planning, the new formula for calculating is:

$$AI_i^{'} = \begin{cases} \dfrac{area\,\tilde{C}_i \cap area\,\tilde{D}_i}{area\,\tilde{C}_i}, & Co_i = 1 \\ 1, & Co_i = 0 \end{cases} \tag{1}$$

In formula (1), AI_i means the satisfaction index of job i , $area\,\tilde{C}_i$ means the area which formed by the membership degree function of fuzzy completion time; $area\,\tilde{D}_i$ means the area which formed by the membership degree function of fuzzy due date, $Co_i\left(Co_i \in \{0,1\}\right)$ is the collaborative planning of job i , if $Co_i = 0$, job i will be outsourced to partners; else if $Co_i = 1$, it will be finished by own.

3.2. Fuzzy Total Cost Index

The cost must be considered in the model for integration of collaborative planning and fuzzy scheduling. Most scholars regard the processing cost of each working procedure as a single definite number while studying on the fuzzy scheduling. In this paper, we regard it as a fuzzy number, which can be calculated by material cost, the corresponding processing cost of each machine and the fuzzy processing time, so we call it fuzzy processing cost.

Taking the collaborative planning into consideration, as the collaborative cost is a definite number while the processing cost of the task finished by own is a fuzzy number, we need to define the addition operation between a definite number and a fuzzy number firstly to calculate the total cost. Assume that P is the collaborative cost, we treat it as a special triangular fuzzy number $\tilde{P}=(P,P,P)$, and $\tilde{C}=(c_i^1,c_i^2,c_i^3)$ is the fuzzy processing cost. According to the addition operation on fuzzy numbers, we can conclude that:

$$P+\tilde{C}=(P,P,P)+(c_i^1,c_i^2,c_i^3)=(P+c_i^1,P+c_i^2,P+c_i^3) \tag{2}$$

)

3.3. Fuzzy Total Cost Index

To calculate the completion time, we must calculate the completion time of each working procedure orderly, and finally we can get the completion time of the last procedure. In the fuzzy flexible scheduling, the processing time of each working procedure is a fuzzy number, so the completion time we finally get is also a fuzzy number, we call it fuzzy completion time here. The formula for calculating fuzzy completion time of working procedure O_{ij} is:

$$\tilde{T}_{ij}=\begin{cases}\tilde{T}_{mp}+\tilde{t}_{ijk}, & j=1\\ \max\{\tilde{T}_{i(j-1)},\tilde{T}_{mp}\}+\tilde{t}_{ijk} & j\neq 1\end{cases} \tag{3}$$

In formula (3), $\tilde{T}_{mp}$ means the fuzzy completion time of its preceding activity in M_k (if M_k remains idle before process O_{ij}, $\tilde{T}_{mp}=(0,0,0)$); $\tilde{T}_{i(j-1)}$ means the fuzzy completion time of $O_{i(j-1)}$; $\tilde{t}_{ijk}$ is the fuzzy processing time of O_{ij} in M_k.

Similar to the fuzzy total time, the new formula for calculating fuzzy completion time of the working procedure O_{ij} with collaborative planning is:

$$\tilde{T}'_{ij}=\begin{cases}\tilde{T}_{ij}, & Co_i=1\\ 0, & Co_i=0\end{cases} \tag{4}$$

3.4. Tardiness Credibility Index

Tardiness credibility index means the credibility of the finished job with tardiness under the circumstance of fuzzy due date, and it can be used for measuring the possibility of tardiness happen to each job. Since the fuzzy processing time is a triangular fuzzy number while the fuzzy due date is a trapezoidal fuzzy number, to calculate the tardiness of each job, we need to define the subtraction operation between a triangular fuzzy number and a trapezoidal fuzzy number [11]: Assume that $C_i=(c_i^1,c_i^2,c_i^3)$ is the fuzzy completion time of job i and $D_i = (d_i^1, d_i^2, d_i^3, d_i^4)$ is the fuzzy due date, the fuzzy tardiness T_i of job i is $T_i = C_i - D_i = (c_i^1 - d_i^4, c_i^2 - d_i^3, c_i^2 - d_i^2, c_i^3 - d_i^1)$.

For the fuzzy tardiness T_i of job i, if $T_i > (0, 0, 0, 0)$, the tardiness will occur certainly; else if $T_i < (0, 0, 0, 0)$, the tardiness will not occur; else it's hard to judge whether the tardiness will happen. Literature [11] has concluded and proved the general method for solving the tardiness credibility of jobs in fuzzy scheduling. Assume that the fuzzy tardiness of job i is $T_i = (t_i^1, t_i^2, t_i^3, t_i^4)$, $t_i^1 < t_i^2 \le t_i^3 < t_i^4$ the tardiness credibility $Cr_{\tilde{T}_i}(t \ge 0)$ is:

$$Cr_{\tilde{T}_i}(t \ge 0) = \begin{cases} 0, & t_i^4 \le 0 \\ \lambda t_i^4 / (t_i^4 - t_i^3), & t_i^4 > 0, t_i^3 \le 0 \\ \lambda, & t_i^3 > 0, t_i^2 \le 0 \\ (t_i^2 - \lambda t_i^1) / (t_i^2 - t_i^1), & t_i^2 > 0, t_i^1 \le 0 \\ 1, & t_i^1 > 0 \end{cases} \tag{5}$$

Similar as above, the new formula for calculating the tardiness credibility with collaborative planning is:

$$Cr'_{\tilde{T}_i}(t \ge 0) = \begin{cases} Cr_{\tilde{T}_i}(t \ge 0), & Co_i = 1 \\ 0, & Co_i = 0 \end{cases} \tag{6}$$

4. A Model for Multi-Objective Integration of Flexible Collaborative Planning And Fuzzy Lot-Splitting Scheduling

4.1. Problem Description

N jobs ($J_i, i \in \{1, 2, 3..., n\}$) need to be finished in the production unit. Q_i Is the production lot size of job i, and every job is constrained by its own fuzzy due date $\tilde{D}_i$. The own processing capability of the production unit is insufficient, so part of the jobs must be outsourced to partners. Assume that the collaborative ratio of job i is r_i, and $r_i \in [0,1]$. If $r_i < 1$, job i can be divided into P_i ($1 \le P_i \le Q_i(1 - r_i)\, and P_i \in N$) sub-jobs. Every sub-job will be handled as a whole, and they have the same stating time. q_{ij} Means the number of job i in lot j. The collaborative cost of J_i is expressed in C_{PJi} under the hypothesis that all collaborative jobs can be finished within the stipulated time. The production unit has m machines ($M_k, k \in \{1, 2, ..., m\}$), the processing cost per unit time of which expresses in C_{Mk}. Every job contains one or several working procedures (O_{ij}, $j \in \{1, 2, ..., n_i\}$, n_i means the total working procedures of J_i). O_{ij} can be produced in different machines. M_{ij} means the set of available machines for number j working procedure of J_i, $M_{ij} \in \{1, 2, ..., m\}$. $\tilde{t}_{ijk}$ means the fuzzy processing time of the number j working procedure of J_i in the M_k. The material cost of J_i expresses in MC_{Ji}.

The goal of the integration of flexible collaborative planning and fuzzy lot-splitting scheduling is to optimize the flexible collaborative planning, the flexible lot splitting and scheduling of the job simultaneously. Meanwhile the model employs average delivery satisfaction, fuzzy total cost, fuzzy completion time and average credibility of job tardiness as optimization objectives to optimize them overall on the premise that they can satisfy all the constraints.

4.2. Objective functions

(1) Average delivery satisfaction

$$\max f_1 = \frac{1}{n}\sum_{i=1}^{n} AI_i \tag{7}$$

For the convenience of handling the function, we change formula (7) into formula (8):

$$\min f_1 = 1 - \frac{1}{n}\sum_{i=1}^{n} AI_i \tag{8}$$

(2) Fuzzy total cost

$$\min f_2 = \sum_{i=1}^{n}\left(\sum_{1}^{P_i} q_{ij}(MC_{Ji} + \sum_{j=1}^{n_i}\sum_{k=1}^{m} C_{Mk}\tilde{t}_{ijk} d_{ijk})\right) + \sum_{i=1}^{n} Q_i r_i C_{PJi} \tag{9}$$

(3) Fuzzy completion time

$$\min f_3 = \max\left\{\tilde{C}_i \mid i = 1, 2, \cdots, n\right\} \tag{10}$$

(4) Average credibility of job tardiness

$$\min f_4 = \frac{1}{n}\sum_{i=1}^{n} Cr_{\tilde{T}_i} \tag{11}$$

AI_i is the delivery satisfaction of job i, and $\tilde{C}_i$ is the fuzzy completion time. d_{ijk} is the decision variable. If the number j working procedure of J_i is processed in M_k, $d_{ijk} = 1$; otherwise, $d_{ijk} = 0$. $Cr_{\tilde{T}_i}$ is the tardiness credibility of job i.

4.3. Constraints

(1) Process constraint. Every job can be processed by only one process route:

$$\sum_{j=1}^{n_i}\sum_{k} d_{ijk} = n_i \tag{12}$$

(2) Working procedure constraint. The next procedure of the same job can't be processed until the preceding one is finished.

$$\tilde{T}_{ijk} - \tilde{T}_{i(j-1)l} \geq \tilde{t}_{ijk}, \quad j \neq 1 \tag{13}$$

$\tilde{T}_{ijk}$ is the completion time of the number j working procedure of J_i processed in M_k.

(3) Machine constraint. When a working procedure is being processed in M_k, the others should wait until it is finished.

$$\tilde{T}_{ijk} - \tilde{T}_{pqk} + MY_{ijpqk} \geq \tilde{t}_{ijk} \tag{14}$$

$$\tilde{T}_{pqk} - \tilde{T}_{ijk} + M\left(1 - Y_{ijpqk}\right) \geq \tilde{t}_{pqk} \tag{15}$$

M is a positive parameters which is large enough. $Y_{ijpqk} \in \{0,1\}$, if $Y_{ijpqk} = 1$, procedure j of J_i will be processed in M_k before procedure q of J_p , else $Y_{ijpqk} = 0$.

(4) The other constraints:

Procedures of different jobs are unrestrained. All jobs have the same priority level. All machines are available at time zero. At least one job should be processed by its own, and so on.

5. Algorithm for the Model Based On the Pareto Optimal

In this section, we propose an algorithm based on the improved NSGA-II. To make the algorithm better, we design the integrated coding scheme, which include collaboration chromosome, lot-splitting chromosome and scheduling chromosome, meanwhile the Pareto optimal scheme is designed.

5.1. Coding and Decoding

The collaborative ratio of each job, the lot splitting and their scheduling are included in the solutions of the model for integration of flexible collaborative planning and fuzzy lot-splitting scheduling. So the collaboration chromosome, lot-splitting chromosome and scheduling chromosome should be included in the chromosome coding scheme.

One gene of a collaboration chromosome represents one decision variable, and the random number between 0 and 1 represents the collaborative ratio. As shown in Figure.1, in the initialization, we generate n random numbers express in $Rand_i$ (n means the total number of jobs, $Rand_i \in \{0,1\}$, $i \in \{1,2,3\cdots,n\}$), which represents each gene on collaboration chromosome. The top three genes in Figure.1 represent that the collaborative ratio of the number of outsourced jobs 1, 2, 3 are 0, 0.4 and 0.5.

0	0.4	0.5	...	0.7	1

n

Figure 1. The Gene Segment of Collaboration Chromosome

In addition, since that in actual production, there may be a small number of jobs needed to be outsourced or finished by own. In this case, as constrained in formula (16), we will outsource all the jobs or finish by own.

$$Rand_i = \begin{cases} 0, & Rand_i \leq 0.1 \\ 1, & Rand_i \geq 0.9 \end{cases} \tag{16}$$

In the scheduling problem, there is a U-sharped relationship between the lot and the production cycle [7]: if the lot size is too small, the quantity of the sub-lot will increase. At the same time, with the increase of the quantity of the sub-lot, the search space of the problem will rise sharply, and the search efficiency of the algorithm will drop as well as the quality of the solutions; on the other hand, if the lot size is too big, part of the machines will be leaved idle for a long time, and the load of some machines will be overweight. So production efficiency will drop. Aiming at the disadvantage of the same batch proposed by Sun Zhijun [6], we adopt a flexible lot-splitting method based on 'cursor' which is proposed by Bai Junjie [7]: we need to set a certain amount of cursors in the working procedures of one job, and then divide the job into several lots. As shown in Figure.2, assume that the number of jobs need to be finished by own is 10, the position of the three cursors is 3, 7, 7, so the jobs are divided into 3 lots (3, 4, 3). During the running time of the algorithm, the cursors will adapt themselves based on the current state of the system. In addition, the number of the cursors can be appropriate changed according to the number of the jobs. In this way, the sub-lot and its lot size can be adjusted flexibly.

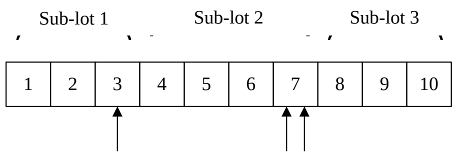

Figure 2. The Flexible Lot-Splitting Method of the Jobs

The lot-splitting chromosome represents the lot-splitting strategy of each job, and the scheduling chromosome represents the scheduling sequence of each lot. In order to identify the lot-splitting chromosome and the scheduling chromosome, we insert a meaningless '00' between them. Assume that there are two kinds of jobs, the production quantity and the collaborative ratio of which is 10, 10 and 0, 0.4, the number of the job finished by own is 10 and 6. As shown in Figure.3, the first part of the code means that the two kinds of jobs are divided into 3 and 2 lots, and the second part means the scheduling sequence of each lot. The scheduling sequence is still coded by the process. For example, '11' and '12' represent the first and the second lot of the first job. The first '11' represents the first working procedure in the first lot of the first job, and the second '11' represents the second working procedure in the first lot of the first job, and so on. If there are more than 10 kinds of jobs or sub-lots, they can be expressed in decimal numbers.

37	2	00	11	12	21	11	13	22	. . .

Figure 3. The Gene Segment of the Lot-Splitting Scheduling

As shown in Figure.4, an integrated chromosome is consisted of collaboration chromosome, lot-splitting chromosome and scheduling chromosome, which are separated with '00'.

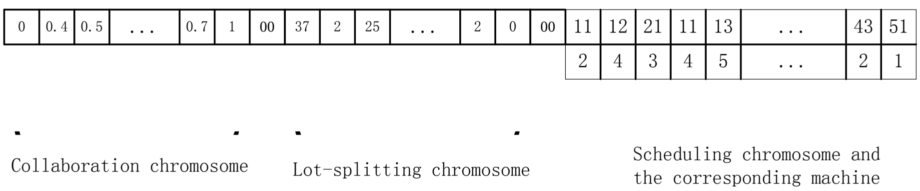

Figure 4. Chromosome Coding Scheme

While decoding, we need to decode the three part of the chromosome respectively to get a feasible solution of the problem.

5.2. Crossover and Mutation Operation

During the crossover and mutation operation, the lot size of each job can be changed, so we need to adjust the length of the chromosome. Aiming at this particularity, crossover and mutation operation of the three parts should be undertaken separately.

(1) Crossover operation of collaboration chromosome. We use the single-point crossover strategy in this operation. As shown in Figure.5, choose a crossover point randomly, collaboration chromosome X and its corresponding collaboration chromosome Y will be cut into two parts. We will get the new collaboration chromosomes by exchanging the second part of these two collaboration chromosome.

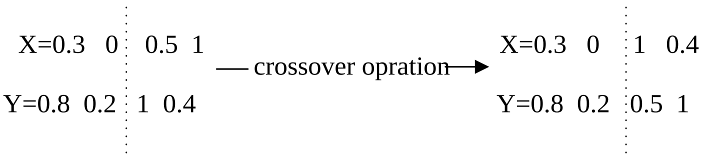

Figure 5. Crossover Operation of Collaboration Chromosome

Mutation operation of collaboration chromosome is as below: Choose a mutation bit randomly, replace it with a random number between 0 and 1, and this random number should meet the formula (16). Take the collaboration chromosome $[0.3\ 0.4\ 0.5\ 0.8]$ for example, if we choose bit 2 to do the mutation operation, the new collaboration chromosome is $[0.3\ 0.6\ 0.5\ 0.8]$.

(2) Crossover operation of lot-splitting chromosome. The number of the genes need to be crossed is random. Since that the number of each job isn't always the same, the cursor can beyond the actual quantity of processing, we need to repair the chromosome. As shown in Figure.6, the cursors of the first job in the first chromosome are appeared at bit 3 and 7, the quantity of the second job is 6, so we need to get rid of cursor 7. At the same time, the lot size of the jobs is changed. If there is an increase, we can insert the corresponding chromosome into a random bit of the second part of the chromosome; if there is a decrease, we need to remove the extra chromosome. For example, both of the two jobs have 3 working procedures. If the lot size of the first job reduce from 3 to 2, we should remove the 3 extra '13' in the second part of the chromosome; If the lot size of the second job increase from 1 to 4, we should insert 3 '22' and '23' randomly.

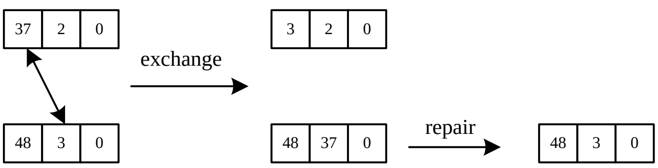

Figure 6. Crossover and Repair Operation of Lot-Splitting Chromosome

Mutation operation of lot-splitting chromosome is as below: Exchange the 2 random genes of the lot-splitting to generate a new chromosome. If the new chromosome is an illegal one, we should repair it with the same method above.

(3) Crossover operation of scheduling chromosome. In order to reduce the complexity of the program and speed the algorithm, we set a rule that the crossover operation of scheduling chromosome can be allowed only when the collaboration chromosomes are identical.

Mutation operation of scheduling chromosome is similar to lot-splitting chromosome.

5.3. Fast Non-Dominant Sorting and Selection Operation

The non-dominant sorting is based on the conception of Pareto optimum, the basic process is: To each individual, calculate its non-dominated level, which is expressed in $rank_i$; the non-dominated level of all non-dominated individuals is defined as 1, the others are equal to the number of individuals dominate it plus 1. If N_i is the number of individuals that dominate individual i, the non-dominated level of individual i is:

$$rank_i = 1 + N_i \tag{17}$$

We sort the non-dominated level of all individuals by the $rank_i$. When constructing the non-dominated set, the quick sort algorithm was adopted in this paper, which was proposed by ZHENG Jinhua [14]. Complexity of this algorithm is smaller than $O(rn\lg n)$, and it's better than traditional NSGA Ⅱ's $O(rN^2)$.

The crowding distance is also used in this paper to maintain the diversity of individuals. An individual's crowding distance can be calculated through summing the distance of each sub-objective between two neighboring individuals within the same non-dominated level after the normalization, which was expressed in $L[i]_d$. The calculation method in this paper is as follows: Sort the individuals at the same non-dominated level from smallest to largest according to its fitness value of objective j. Set the first and the last individual's crowding distance to infinity, so that they can be selected in each selection process, other individuals' crowding distance of objective j are calculated in this way:

$$L[i]_{dj} = \frac{fitness_j^{i+1} - fitness_j^{i-1}}{fitness_j^{n} - fitness_j^{1}} \tag{18}$$

$L[i]_{dj}$ Means the crowding distance of objective j of individual i; $fitness_j^{i+1}$ means the fitness value of objective j of individual $i+1$ at this non-dominated level; $fitness_j^{n}$ means the fitness value of objective j of the last individual at this non-dominated level. We get each individual's crowding distance after the sum of its crowding distance in each objective. The calculation of crowding distance is to ensure that our algorithm can converge to the evenly distributed Pareto-optimal front.

After the non-dominant sorting and the calculation of crowding distance, each individual has two properties: the non-dominant level $rank_i$ and the crowding distance $L[i]_d$. Define a partial order relation $\succ_n$: if $rank_i < rank_j$, or $rank_i = rank_j$ and $L[i]_d > L[j]_d$, then $i \succ_n j$. This partial order relation can be understood as individual i better than individual j. The selection operation is based on this partial order relation.

5.4. The Algorithm Flow

The procedures of the MOEA based on improved NSGA-II are as follows:

(1) Generate the initial population P_0 randomly, population size is $Popsize$, and set $t = 0$;

(2) Non-dominant sorting the P_0, and calculate its crowding distance. Then generate the first progeny generation Q_0 through the selection, crossover and mutation operation;

(3) Combine the father generation and the progeny generation: $R_t = P_t \cup Q_t$. Non-dominant sort the R_t, and construct the Pareto-optimal front F according to the conception of Pareto optimum, $F = \{F_1, F_2, ..., F_i\}$. F_i means the non-dominated front of level i;

(4) Clear the P_{t+1}, and set $i = 1$;

(5) If $|P_{t+1}| + |F_i| \le Popsize$, continue to step (6); otherwise, skip to step (8);

(6) Calculate the crowding distance $L[i]_d$ of individuals in the set of F_i, and put the individuals in the set of F_i to the next generation P_{t+1}, $P_{t+1} = P_{t+1} \cup F_i$;

(7) Set $i = i + 1$, back to step (5);

(8) Execute the selection, crossover and mutation operation on P_{t+1}, generate the new Q_{t+1};

(9) If $t < Gen_{\max}$ ($Gen_{\max}$ is the maximum evolution time), $t = t + 1$, and back to step (3); otherwise, end the evolution.

6. Simulation and Analysis

The efficiency of the model and the algorithm is proved by the data simulated from a manufacturing enterprise and its partner. There are 10 machines, 6 kinds of jobs, 10 batches and 6 working procedures. The restraints of the available machines are shown in Table 1. The fuzzy processing time, the quantity of each job, the fuzzy due date, the material cost and collaboration cost per unit time are shown in Table 2. "—" means this job cannot be outsourced. The processing cost per unit time of every machine is given in Table 3.

The algorithm parameters are set as below: the population size $Popsize = 100$, crossover probability $P_c = 0.45$, mutation probability $P_m = 0.05$, the maximum evolution time is 80, reliability coefficient $\lambda = 0.5$, and other parameters are omitted. The proposed algorithm is coded with Java. The running environment for the program is as follows: P4 CUP, 2.8GHz, and 1.25G RAM.

Table 1. Machine Constraints

J_i	Machine constraints					
J_1	3/10	1	2	4/7	6/8	3
J_2	2	3	5/8	6/7	1	4/10
J_3	3/9	4/7	6/8	1	2/10	5
J_4	4	1/9	3/7	2/8	5	6
J_5	5	2/7	3/10	6/9	1	4/8
J_6	2	4/7	6/9	1	5/8	3

Table 2. Fuzzy Processing Time, Production Quantity, Fuzzy Due Date, Material Cost and Collaborative Cost of Each Job

J_i	$\tilde{t}_{ijk}$						N	$\tilde{D}_i$	MC_{Ji}	C_{PJi}
	1	2	3	4	5	6				
J_1	(1,3,4)/(4,5,6)	(8,10,12)	(7,9,11)	(3,5,6)/(3,4,6)	(2,3,4)/(2,3,4)	(8,10,12)	10	(250,330, 420,510	240	780

								)		
J_2	(4,6,8)	(7,8,9)	(1,1,2)/ (3,4,5)	(4,5,6)/ (4,6,7)	(2,3,5)	(2,3,4)/ (1,3,4)	10	(240,280, 350,440)	270	870
J_3	(1,1,2)/ (3,4,6)	(4,5,7)/ (6,7,8)	(3,5,6)/ (4,6,8)	(4,5,6)	(7,9,11)/ (9,11,13)	(1,1,2)	10	(210,290, 380,450)	260	—
J_4	(5,7,9)	(3,4,5)/ (1,3,5)	(3,4,5)/ (4,6,8)	(1,3,4)/ (4,5,7)	(1,1,2)	(2,3,5)	10	(250,290, 360,450)	300	800
J_5	(5,6,7)	(8,10,12)/ (10,12,15)	(6,7,8)/ (7,9,12)	(6,8,10)/ (7,8,9)	(3,5,6)	(2,4,6)/ (6,7,8)	10	(320,370, 430,520)	230	790
J_6	(2,3,4)	(9,10,12)/ (9,11,13)	(7,8,10)/ (5,7,9)	(8,9,10)	(3,4,6)/ (3,5,7)	(8,9,10)	10	(340,380 430,520)	250	850

Table 3. Processing Cost per Unit Time of Every Machine

M_k	M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8	M_9	M_{10}
C_{Mk}	4	6	8	4	9	6	5	3	7	5

The Pareto optimal solution set generated by the program is shown in Table 4, and we just choose parts of the representative solutions according to our personal preference. In addition, to verify the efficiency of integration of flexible collaborative planning and lot-splitting scheduling, we solve the combinatorial scheduling problems respectively with the algorithm proposed in this paper. These problems are traditional collaborative planning, flexible collaborative planning, lot-splitting scheduling and flexible lot-splitting scheduling. The optimization results are shown in Table 5 to Table 9.

Table 4. The Pareto Optimal Solution of Integration of Flexible Collaborative Planning and Flexible Lot-Splitting Scheduling

No.	Optimization goal			
	f_1	f_2	f_3	f_4
1	0.139	(30528,31962,34484)	(363,436,528)	0.211
2	0.121	(30778,32562,34985)	(318,416,504)	0.183
3	0.113	(30890,32780,35170)	(300,396,482)	0.171
4	0.105	(31190,34386,37876)	(292,384,476)	0.168

Table 5. The Pareto Optimal Solution of Integration Of Flexible Collaborative Planning And Lot-Splitting Scheduling With The Same Batch

No.	Optimization goal			
	f_1	f_2	f_3	f_4
1	0.189	(31424,32848,35376)	(402,492,584)	0.265
2	0.166	(31678,33460,35885)	(358,456,540)	0.237
3	0.157	(31788,33682,36068)	(344,430,520)	0.223
4	0.146	(32086,35284,38774)	(338,415,510)	0.219

Table 6. The Pareto Optimal Solution of Integration of Flexible Collaborative Planning and Scheduling

No.	Optimization goal			
	f_1	f_2	f_3	f_4
1	0.217	(32218,33646,36178)	(430,522,612)	0.291
2	0.194	(32462,34254,36676)	(384,482,570)	0.269
3	0.186	(32688,34486,36868)	(378,468,556)	0.251
4	0.174	(32882,36078,40166)	(368,445,548)	0.237

Table 7. The Pareto Optimal Solution of Integration of Collaborative Planning and Flexible Lot-Splitting Scheduling

No.	Optimization goal			
	f_1	f_2	f_3	f_4
1	0.191	(31812,33218,35726)	(418,508,599)	0.269
2	0.168	(32064,33860,36216)	(373,476,554)	0.237
3	0.159	(32158,34048,36468)	(359,448,536)	0.226
4	0.147	(32482,35684,39128)	(356,432,528)	0.219

Table 8. The Pareto Optimal Solution of Integration of Collaborative Planning and Lot-Splitting Scheduling With the Same Batch

No.	Optimization goal			
	f_1	f_2	f_3	f_4
1	0.218	(32266,33698,36228)	(436,528,618)	0.293
2	0.196	(32512,34308,36724)	(388,486,578)	0.269
3	0.186	(32734,34526,36916)	(382,474,558)	0.252
4	0.174	(32938,36084,40216)	(374,446,548)	0.237

Table 9. The Pareto Optimal Solution of Integration of Collaborative Planning and Scheduling

No.	Optimization goal			
	f_1	f_2	f_3	f_4
1	0.268	(32872,34298,36834)	(462,584,644)	0.341
2	0.245	(33112,34918,37336)	(412,512,604)	0.319
3	0.237	(33348,35126,37522)	(412,504,582)	0.302
4	0.224	(33546,36694,40828)	(402,474,574)	0.296

We can see from Table 4 to Table 9 that, the combination of flexible collaborative planning and flexible lot-splitting scheduling can balance the capacity of the production unit effectively. At the same time, it can increase the utilization rate of the machine, shorten the production cycle, improve the satisfaction of the due date, and reduce the manufacturing cost. The scheduling Gantt of the third solution in Table 4 is shown in Figure.7. In this solution, collaborative ratio of each job is below: 0.5, 1, 0, 0.8, 0.4, and 0.2. The lot of J_3, J_5, J_6 is 2 (3, 7), 2 (3, 3), 2 (3, 5), others have no sub-lots, so there are 8 sub-lots. Numbers in Figure.7 represent the kinds of jobs, the lots and the corresponding working procedures. For example, ‘3, 2-1’ means the first working procedure of the second lot of J_3. Run-up time of the lot and waiting time of the machine are omitted. In addition, the processing time is fuzzy, so we just mark the corresponding position of each procedure and the fuzzy completion time of the last procedure.

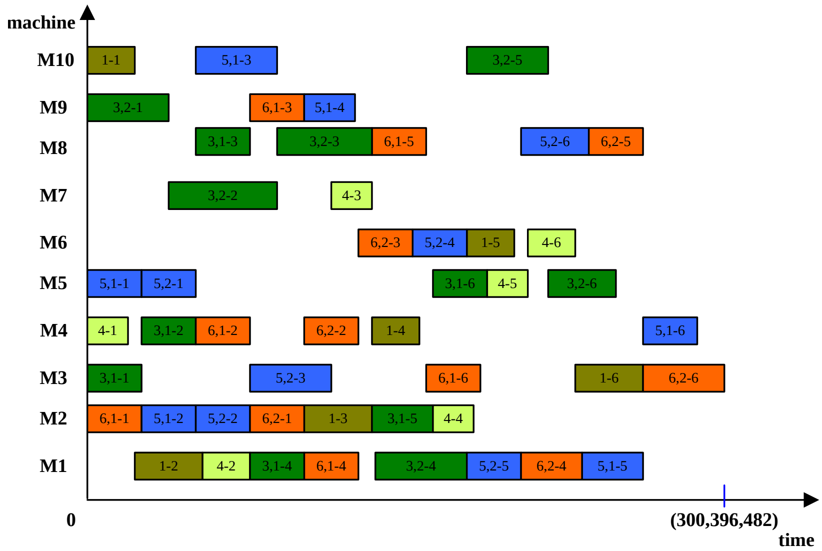

Figure 7. Gantt for Integration of Flexible Collaborative Planning and Flexible Lot-Splitting Scheduling

7. Conclusion

The main work of this paper was summarized as follows: First, we studied the fuzzy measure method of delivery satisfaction, fuzzy total cost, fuzzy completion time and credibility of job tardiness. At the same time, the calculation formulas of them were given. Second, a model for multi-objective integration of flexible collaborative planning and fuzzy lot-splitting scheduling of batch production was established. And then we designed a multi-objective algorithm based on the Pareto optimal, which include collaboration chromosome, lot-splitting chromosome and scheduling chromosome. Meanwhile the Pareto optimal scheme and the algorithm flow were given. Finally, the efficiency of the model and algorithm was proved by the simulation.

Since that the number of optimization objective is usually more than four, our further work is to design a high-dimension multi-objective evolutionary algorithm to solve the model for integration of collaborative planning and fuzzy scheduling.

8. Acknowledgements

This research was financially supported by the National Natural Science Foundation of China (Grant NO.61170201), Research and Innovation Project for College Graduates of High Education of Jiangsu Province of China No. CXZZ13_0919 and Research and Innovation Project for Undergraduates of High Education of Jiangsu Province of China No. 201411117041Z.

9. References

9.1 Journal Article

[1] Bao Zhenqiang, Wang Guijun, Bao Rong, Ding Quanxun and Pan Quanke. Journal of Computational Information Systems. 6, 12 (**2010**)
[2] Bao Rong, Wang Weiye, Gu Hanjie and Xu Yong'an. Computer Technology and LOW C Y, HSU C M and HUANG K I. International Journal of Advanced Manufacturing Technology, 24, 9/10 (**2004**)Development, 20, 10 (**2010**)

[3] LOW C Y, HSU C M and HUANG K I. International Journal of Advanced Manufacturing Technology, 24, 9/10 (**2004**)
[4] Pan Quanke, Zhu Jianying. Journal of Mechanical Engineering, 40, 4 (**2004**)
[5] JEONG H I, PARK J and LEACHMAN R C. International Journal of Production Research, 37, 15 (**1999**)
[6] Sun Zhijun, An Jing and Huang Weiqing. China Mechanical Engineering, 19,2 (**2008**)
[7] Bai Junjie, Gong Yiguang, Wang Ningsheng and Tang Dunbing. Computer Integrated Manufacturing System, 16, 2 (**2010**)
[8] Lotfi A. Zadeh. Information and Control. 8 (**1965**)
[9] Masatoshi Sakawa and Ryo Kubota. European Journal of Operational Research. 120, 2 (**2000**)
[10] Masatoshi Sakawa and Tetsuya Mori. Computers & Industrial Engineering. 36, 2 (**1999**)
[11] Yang Hongbing and Yan Hongsen. Computer Integrated Manufacturing System. 15, 7 (**2009**)

9.2 Conference Proceeding

[12] CHAN F T S, WONG T C and CH AN P L Y. Equal size lot streaming to Job-shop scheduling problem using genetic algorithms. Proceedings of 2004 IEEE International Symposium on Intelligent Control. (**2004**) 472- 476; Washington, D. C. USA
[13] CHAN F T S, WONG T C and CHAN L Y. Lot splitting under different Job Shop conditions. Proceedings of 2007 IEEE Congress on Evolutionary Computation. IEEE, (**2007**) 4722-4728; Washington, D. C. USA.

9.3 Books

[14] Zheng Jinhua, Editor. Multi-objective evolutionary algorithms and their applications, Science Publishing House, Beijing (**2007**)

Author

Zhenqiang Bao

