求解模糊作业车间调度问题的群体邻域搜索算法

郑友莲[1,2],李元香[2],雷德明[3]
(1.湖北大学数学与计算机科学学院,湖北武汉 430062; 2.武汉大学软件工程国家重点实验室,湖北武汉 430072;
3.武汉理工大学自动化学院,湖北武汉 430070)

摘 要: 本文提出一种群体邻域搜索算法(Swarm-based Neighborhood Search,SNS),用于最小化模糊作业车间调度问题(Fuzzy Job Shop Scheduling Problem,FJSSP)的模糊 makespan.该算法使用基于有序工序的编码,通过锦标赛选择和概率为 1 的动态调整互换操作更新群体.对调度结果的理论分析表明,模糊 makespan 能反映解的优劣.理论分析及大量实验证明,SNS 具有较强的全局和局部优化能力,以及较快的收敛速度,在求解 FJSSP 方面具有较强的优势.

关键词: 模糊作业车间调度; 群体邻域搜索; 互换
中图分类号: TP301 **文献标识码:** A **文章编号:** 0372-2112 (2011) 10-2454-05

Solving Fuzzy Job Shop Scheduling Problems Through Swarm-Based Neighborhood Search Algorithm

ZHENG You-lian[1,2], LI Yuan-xiang[2], LEI De-ming[3]
(1. *Faculty of Mathematics & Computer Science*, *Hubei University*, *Wuhan*, *Hubei* 430062, *China*;
2. *State Key Laboratory of Software Engineering*, *Wuhan University*, *Wuhan*, *Hubei* 430072, *China*;
3. *School of Automation*, *Wuhan University of Technology*, *Wuhan*, *Hubei* 430070, *China*)

Abstract: This paper presents a swarm-based neighbourhood search algorithm(SNS) to minimize the maximum completion time of fuzzy job shop scheduling problem(FJSSP). SNS uses an ordered operation-based representation, tournament selection and swap operation in which probability is 1 and swap-time is adjusted dynamically. The theoretical analyses on scheduling results show that fuzzy makespan can be used to evaluate the quality of solution. Theoretical analysis and a large number of experiments demonstrate that SNS has strong global and local optimization capabilities, faster convergence speed, and promising performance on FJSSP.

Key words: fuzzy job shop scheduling; swarm-based neighborhood search; swap

1 引言

不确定性是制造过程的基本特性,其主要描述工具包括随机理论和模糊理论等,其中模糊理论在不确定性建模[1]、优化[2]、控制[3]与调度[4~12]等中的应用较广泛.近十多年来,模糊调度理论与方法受到国内外学者的广泛关注.

模糊作业车间调度问题(FJSSP)突破了传统作业车间调度加工时间固定这一约束,采用模糊数描述加工条件和参数,自 Sakawa 等[4,5]最早利用遗传算法(GA)求解 FJSSP 以来,FJSSP 研究取得了较大进展[6~12].只是现有调度算法以 GA 和粒子群算法为主,表示方法以基于工序的编码和随机键编码为主,这样使得调度算法全局优化能力较强,而局部优化能力不足,从而影响求解质量.

为此,本文提出了基于有序工序的编码,并设计群体邻域搜索(SNS)算法,该算法通过锦标赛选择和概率为 1 的动态调整互换更新群体,具有结构简单、实现容易和局部优化能力较强等特点;另外,现有研究很少对调度结果进行理论分析,本文的理论分析表明,模糊目标值能反映解的优劣.

2 模糊数及其操作

三角模糊数(TFN)由于其参数少,求和、取大和比较等操作较简单,而广泛应用于模糊加工时间的描述.在模糊调度建立过程中,TFN 求和操作用来计算模糊完成时间,TFN 取大用来计算模糊加工开始时间,而不同的调度解之间的比较则利用 TFN 的比较过程.

对 TFN $A=(a_1,a_2,a_3)$和 $B=(b_1,b_2,b_3)$,

$$A+B=(a_1+b_1,a_2+b_2,a_3+b_3) \tag{1}$$

在模糊场合,常用如下过程[4~7,9~12]比较不同的 TFN:对于 TFN $A=(a_1,a_2,a_3)$和 $B=(b_1,b_2,b_3)$,首先用指标 c_1 进行比较;如果两个 TFN 的 c_1 值相等,则选

收稿日期:2010-09-27;修回日期:2011-07-11
基金项目:国家 863 计划(No.2007AA01Z290);国家自然科学基金(No.60773009)

用 c_2 比较 A 和 B;如果 A 和 B 的 c_1 和 c_2 值都相等,则选用指标 c_3 比较.

对于 A: $c_1(A)=(a_1+2a_2+a_3)/4$, $c_2(A)=a_2$, $c_3(A)=a_3-a_1$.

通常 $A\vee B$ 不再是 TFN,而是普通模糊数,这将导致求和更加复杂,比较无法进行,从而使得调度建立过程无法完成,为此,计算近似 $A\vee B$,使其始终具有三角特性.采用如下准则[12]计算近似 $A\vee B$:如果 $A>B$,则令 $A\vee B=A$;否则令 $A\vee B=B$.

3 求解 FJSSP 的群体邻域搜索算法

3.1 问题描述

$n\times m$ FJSSP 由 n 个工件 $J_i(i=1,2,\cdots,n)$和 m 台机器 $M_k(k=1,2,\cdots,m)$组成,每个工件包含 m 道工序,o_{ij}表示工件 J_i 的第 j 道工序,其加工时间 p_{ij}为 TFN,即 $p_{ij}=(a_{ij}^1,a_{ij}^2,a_{ij}^3)$.另外,FJSSP 还需满足一些约束[11],如每一时刻每台机器只能加工一个工件;工序加工不容许中断等.

问题目标函数为

$$\min \qquad C_{\max}=\max_{i=1,2,\cdots,n} C_i \tag{2}$$

其中 C_i 是 J_i 的模糊完成时间,$C_{\max}$是最大模糊完成时间.

3.2 群体邻域搜索算法

本文基于有序工序的编码和解码,提出了一种有效的群体邻域搜索算法,下面详细描述 SNS 的主要步骤.

3.2.1 编码、解码及理论分析

与基于工序的编码和随机键编码相比,基于有序工序的编码其解码过程更简单,这是因为染色体本身就是有序工序表,而有序工序表基本决定了加工顺序,而且容易设计出有效的邻域搜索算子.

采用基于有序工序的编码方法,该方法用二元组串$((p_1,r_1),(p_2,r_2),\cdots,(p_i,r_i),\cdots,(p_{nm},r_{nm}))$表示 $n\times m$ FJSSP 的解,其中基因(p_i,r_i)表示工序 $o_{p_i r_i}$,而整个串为有序工序表,其中有 m 个基因的第一个值取为 1(对应第 1 个工件),m 个基因的第一个值取为 2(对应第 2 个工件),依此类推,有 m 个基因的第一个值取为 n(对应第 n 个工件).具有相同工件号的基因称作“等工件基因”,整个染色体由 n 组等工件基因组成.

解码过程描述如下:从有序工序表的第一道工序开始;从左到右依次安排所有工序的加工,为每道工序在其加工机器确定最合适的模糊加工开始时间.

设 TFN $B_{ij}=(b_{ij}^1,b_{ij}^2,b_{ij}^3)$和 $E_{ij}=(e_{ij}^1,e_{ij}^2,e_{ij}^3)$分别表示工序 o_{ij}的模糊开始时间和模糊完成时间,o_{ij}的加工机器为 M_k,如果 $j>1$,则 $B_{ij}=E_{i(j-1)}\vee\eta_k$;否则 $B_{ij}=\eta_k$,其中 η_k 表示机器 M_k 可用于 o_{ij}加工的最早开始时间.$E_{ij}=B_{ij}+p_{ij}$.

表 1　一个 3×3 FJSSP 的加工数据

工件	加工工序					
	1	2	3	1	2	3
	机器号(模糊加工时间)			实际加工时间		
J_1	1(2,3,4)	2(3,4,5)	3(1,2,3)	3	3	2
J_2	2(1,2,3)	3(3,5,7)	1(2,3,5)	1	5	3
J_3	3(2,3,5)	1(2,3,4)	2(2,4,6)	3	2	3

对于表 1 所示的 3×3 FJSSP,设其染色体为((3,1),(2,1),(3,2),(1,1),(1,2),(3,3),(2,2),(1,3),(2,3)),对应的有序工序表为 $o_{31},o_{21},o_{32},o_{11},o_{12},o_{33},o_{22},o_{13},o_{23}$,对应的模糊调度如下:

M_1:{o_{11},(0,0,0)→(2,3,4)},{o_{32},(2,3,5)→(4,6,9)},{o_{23},(5,8,12)→(7,11,17)}

M_2:{o_{21},(0,0,0)→(1,2,3)},{o_{12},(2,3,4)→(5,7,9)},{o_{33},(5,7,9)→(7,11,15)}

M_3:{o_{31},(0,0,0)→(2,3,5)},{o_{22},(2,3,5)→(5,8,12)},{o_{13},(5,8,12)→(6,10,15)}

上述模糊调度中,每道工序所涉及的两个 TFN 分别为对应的 B_{ij}和 E_{ij}.利用表 1 所示的实际加工时间和上述调度中各机器上的工序加工顺序,得到对应的实际调度为

M_1:{o_{11},0→3},{o_{32},3→5},{o_{23},8→11}

M_2:{o_{21},0→1},{o_{12},3→6},{o_{33},6→9}

M_3:{o_{31},0→3},{o_{22},3→8},{o_{13},8→10}

为了进行理论分析,定义模糊集 A 的 $\alpha\in[0,1]$截集 $A_\alpha=\{x\in R\mid\mu_A(x)\geqslant\alpha\}$,对模糊集 A_i, $i=1,2,\cdots,n$,

$$\Big(\sum_{i=1}^{n}A_i\Big)_\alpha=\sum_{i=1}^{n}(A_i)_\alpha \tag{3}$$

模糊调度中,对每一个工件 J_i,其模糊完成时间 $C_i=\sum_{o_{hl}\in\theta_i}p_{hl}$,其中 θ_i 是用于计算 C_i 的工序的集合.上述模糊调度中,$C_1=p_{13}+p_{22}+p_{31}$.

对任何工序 $o_{hl}\in\theta_i$,其实际加工时间 $\hat{p}_{hl}$满足 $\hat{p}_{hl}\in(p_{hl})_{\alpha_{hl}}$,定义工件 J_i 的原始实际完成时间为 $\hat{C}_i=\sum_{o_{hl}\in\theta_i}\hat{p}_{hl}$,它满足如下关系:

$$\hat{C}_i\in\sum_{o_{hl}\in\theta_i}(p_{hl})_{\alpha_{hl}}\subset\sum_{o_{hl}\in\theta_i}(p_{hl})_{\alpha_i}=\Big(\sum_{o_{hl}\in\theta_i}p_{hl}\Big)_{\alpha_i}$$

$$\hat{C}_i\in\Big(\sum_{o_{hl}\in\theta_i}p_{hl}\Big)_{\alpha_i}=(C_i)_{\alpha_i}$$

其中,$\alpha_i=\min\{\alpha_{hl}\mid o_{hl}\in\theta_i\}$.

可见,原始实际完成时间总是属于 C_i 的截集.实际完成时间 $\bar{C}_i=\hat{C}_i+\beta_i$, $\beta_i\geqslant 0$.如果在机器 M_k 上加工的工序 $o_{hl}\in\theta_i(l>1)$满足条件 $B_{hl}>E_{h(l-1)}$和 $b_{hl}^3<$

$e^3_{h(l-1)}$,则可能在某些情况下,对于任何 $\alpha'_i < \alpha_i$,$\bar{C}_i$ 都不属于$(C_i)_{\alpha'_i}$,而在大多数情况下,总能找到一个 $\alpha'_i < \alpha_i$,使得 $\bar{C}_i \in (\bar{C}_i)_{\alpha'_i}$;如果没有工序 $o_{hl} \in \theta_i$ 满足 $B_{hl} > E_{h(l-1)}$和 $b^3_{hl} < e^3_{h(l-1)}$,则总能找到一个合适的 $\alpha'_i < \alpha_i$,使得 $\bar{C}_i \in (\bar{C}_i)_{\alpha_i}$.因此,对于工件的大多数可能实际完成时间 $\bar{C}_i$,能得到 C_i 的一个截集,使得 $\bar{C}_i \in (C_i)_{\alpha'_i}$,从而表明模糊 makespan 能反映解的优劣,模糊 makespan 越小,对应的解越优,从而保证优化计算是有意义的.

3.2.2 群体更新策略及算法描述

采用如下精英策略:在 SNS 搜索过程,如果新解优于精英个体,则更替精英个体;否则精英个体保持不变.

由于最大完成时间是模糊数,轮盘赌复制和比例复制不再适用,故采用二元锦标赛选择[13].

本文的邻域搜索为互换算子,其过程如下:每次随机选择 1 对基因(p_i, r_i)和(p_j, r_j),$i \neq j$ 且 $p_i \neq p_j$ 进行互换,一共选择 t 次,每次选中的基因均与前面所选基因不同;然后对每一组等工件基因的第二个值重新赋值以构成新的有序工序表.

SNS 的伪代码如下:

(1)随机产生初始群体 S,将其中目标值最小的个体定义为精英个体 e,$g = 1$;

(2)while($g <$ max_ge){//max_ge 为最大迭代次数;

(3)S = Tournament_selection(S, e);

(4)根据互换概率 p 对 S 中的每个个体实施互换;

(5)计算 S 中每个个体的目标值,并更新精英个体 e;

(6)$g = g + 1$;}

其中,S = Tournament_selection(S, e)表示将精英个体 e 加入到群体 S 后,对 S 执行二元锦标赛选择,产生新的群体 S,对于个体 i,其目标函数 $F(i) = C^i_{\max}$, $C^i_{\max}$为个体 i 的模糊 makespan.

4 计算实验

本文选用 16 个实例,分别是 ABZ5-6、ORB1-5 和 LA16-24 的扩展,其中 LA21-24 为 15 × 10 的实例,其他的全为 10 × 10 的实例.对实例的每个工序,$p_{ij} = (\delta_{ij}, \underline{p}_{ij}, \underline{p}_{ij} + \gamma_{ij})$,其中 $\underline{p}_{ij}$表示确定性实例的相应的加工时间,整数 $\delta_{ij} \in [0.85\ \underline{p}_{ij}, 0.94\ \underline{p}_{ij}]$,$\gamma_{ij} \in [0.1\ \underline{p}_{ij}, 0.19\ \underline{p}_{ij}]$,如果 $\gamma_{ij} < 1$,则让 $\gamma_{ij} \in [1,2]$.为简单起见,这些扩展的实例仍称作 ABZ5-6、ORB1-5 和 LA16-24.

下面首先测试并确定 SNS 的参数,然后将 SNS 和其他算法进行比较.所有实验在 Microsoft Visual C++ 6.0 环境下实现且执行于 PC(2.5GHz CPU,2GB RAM).

4.1 SNS 参数测试

SNS 的参数为:群体大小为 100,10 × 10 问题的最大代数均取为 600,15 × 10 问题的最大代数均取为 1000.对互换概率 p 和互换次数 t 进行测试,互换概率 p 在 0.5 ~ 1.0 之间变化,互换次数 t 在 1 ~ 4 之间变化.对每个实例随机执行 30 次,图 1 给出了 ORB1 和 LA21 的计算结果三维图,avg-c_1 表示 30 个最好解的平均值的指标 c_1,其他各个实例也得到了类似形状的三维图.从图中可知,当互换概率和互换次数均取为 1 时,能得到最好的效果,互换概率越小效果越差,变异次数越大效果也越差.这是因为,若互换概率取小了,则达不到进化的效果,所以很难进化到最优值;若互换次数取大了,搜索前期能够进行大范围探索,进行一定的进化,但后期却不能很好地进行邻域搜索,从而难以逼近最好解.

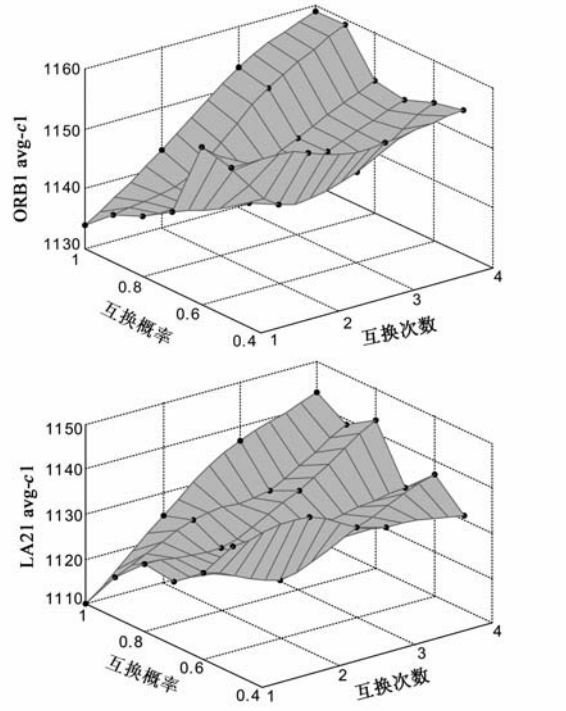

图1 SNS参数测试结果

将互换概率确定为 1,互换次数以 1 为主.进一步实验证实,t 进行如下动态调整能获得更好的寻优结果:1 ~ 50 代,t = 3;50 ~ 100 代,t = 2;100 代以后,t = 1.这是因为,当 SNS 在进化的早期,较短时间进行较大范围的探索,能够更有利地发现最优解区域,而在后期较长时间进行邻域搜索能更好逼近最优解.图 2 给出了这种参数设置下的 ORB1 和 LA24 的某一次进化示意图,图中"目标值-c_1"表示目标值的指标 c_1,其他实验也有类似的进化图,可见最大代数的设置是合理的.

4.2 结果比较与讨论

将 SNS 与 RKGA[11]和 GPSO[8]进行比较.GPSO 的参

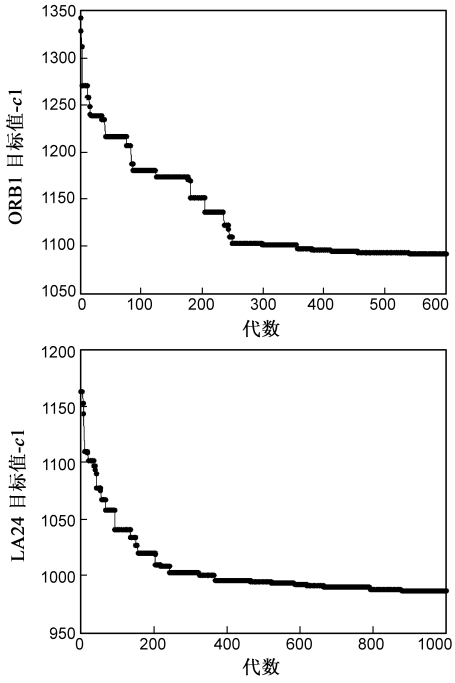

图2　SNS进化进程

数为:种群规模为 20. RKGA 的参数为:交叉概率为 0.9,变异概率为 0.1,种群规模为 100. SNS 的参数为:群体大小为 100,互换概率为 1,互换次数动态调整:1 ~ 50 代,t = 3;50 ~ 100 代,t = 2;100 代以后,t = 1. 对于三种算法,10 × 10 问题的最大代数均取为 600,15 × 10 问题的最大代数均取为 1000. 三种算法对同样的问题其目标函数计算次数相同.

对每一个实例,每种算法随机执行 20 次,表 2 和表 3 分别描绘了 20 次执行中的平均解与最好解的对比情况,平均解表示 20 个最好解的平均值. 每种算法的执行时间如表 4 所示,对于"工件数 × 机器数"相同的实例,每个算法都有相似的计算时间,因此,表 4 记录了两类实例的平均执行时间.

如表 2 所示,SNS 对于 16 个例子均获得了优于 RKGA 和 GPSO 的平均解. 如表 3 所示,SNS 有 14 个实例能收敛于最好解,而 RKGA 只有实例 ABZ6 收敛于最好解,GPSO 只有实例 LA20 收敛于最好解,SNS 对于 ABZ6 和 LA20 的计算结果与最好解接近. 另外,SNS 所花费的计算时间远远少于 RKGA 和 GPSO. 因此,SNS 能在短时间内获得明显优于 RKGA 和 GPSO 的计算结果.

群搜索的引入使得 SNS 具有较好的全局搜索能力;互换概率取为 1 又保证了算法的探索力度和进化速度;通过动态调整互换次数,使得 SNS 在进化早期进行较大范围的探索,这又在一定程度上增强了其全局搜索能力,而在进化后期较长时间的邻域搜索又保证了其局部搜索能力. RKGA 的局部搜索能力较差,所以性能不如 SNS. 另外,GPSO 的主要操作是交叉和变异,但是其交叉效率较低,而且一些父代的交叉可能无效,正是交叉的低效性或无效性导致了 GPSO 的性能较差.

表 2　三种算法的平均解对比

实例 \ 算法	RKGA	GPSO	SNS
ABZ5	1121,1262,1442	1118,1257,1437	1117,1254,1431
ABZ6	848,965,1094	851,966,1094	855,960,1085
ORB1	997,1128,1278	1011,1144,1295	1004,1122,1270
ORB2	814,920,1045	818,930,1055	813,914,1045
ORB3	962,1085,1229	970,1094,1246	946,1062,1203
ORB4	931,1047,1188	940,1059,1200	932,1042,1186
ORB5	817,921,1042	820,922,1048	814,913,1035
LA16	867,978,1114	870,974,1107	860,971,1113
LA17	700,789,901	712,795,904	702,787,897
LA18	775,874,1000	780,873,1002	765,859,980
LA19	780,878,1003	778,875,996	765,870,993
LA20	826,931,1061	815,921,1045	811,917,1042
LA21	978,1109,1264	1005,1135,1286	972,1104,1255
LA22	870,982,1116	876,1000,1146	872,979,1111
LA23	927,1045,1188	934,1056,1199	917,1045,1183
LA24	890,999,1142	898,1009,1152	883,998,1140

表 3　三种算法的最好解对比

实例 \ 算法	RKGA	GPSO	SNS
ABZ5	1109,1252,1436	1117,1250,1428	1103,1242,1425
ABZ6	824,948,1074	834,960,1081	839,948,1066
ORB1	986,1107,1266	985,1132,1268	965,1087,1236
ORB2	805,911,1034	798,911,1046	787,893,1022
ORB3	935,1059,1193	952,1066,1222	922,1033,1180
ORB4	925,1032,1172	929,1038,1179	919,1026,1161
ORB5	805,915,1033	813,905,1034	788,889,1012
LA16	856,956,1094	848,945,1076	838,945,1085
LA17	698,784,892	687,785,895	697,784,887
LA18	769,859,973	770,859,982	761,848,954
LA19	760,865,990	763,861,976	749,860,981
LA20	805,909,1027	790,891,1008	802,907,1030
LA21	952,1085,1241	962,1092,1231	935,1075,1235
LA22	861,968,1102	841,968,1103	850,955,1085
LA23	918,1032,1175	914,1030,1179	900,1032,1167
LA24	878,982,1114	877,978,1113	870,977,1115

表 4　三种算法的计算时间

算法	时间(秒)	
	10 × 10 问题	15 × 10 问题
RKGA	7.1	21.7
GPSO	7.8	22.7
SNS	4.7	13.5

5　结论

本文提出一种基于 SNS 的新型调度算法,对调度结果的理论分析表明,模糊 makespan 能反映解的优劣. 所提出的 SNS 因使用基于有序工序的编码和自身独特的结构,从而具有较强的全局和局部优化能力.

参考文献

[1] 林树宽,支力佳,等.基于组合 SVR 的非平稳序列的模糊建模方法[J].电子学报,2006,34(10):1929 – 1932.
Lin S K, Zhi L J, et al. A Multi-SVR Based fuzzy modeling method for non-stationary time series[J]. Acta Electronica Sinica, 2006, 34(10): 1929 – 1932. (in Chinese)

[2] 陈得宝,赵春霞.复数自适应进化规划及模糊规则基的自动提取[J].电子学报,2007,35(2):341 – 344.
Chen D B, Zhao C X. An plurality adaptive evolutionary programming and extracting fuzzy rule bases automatically[J]. Acta Electronica Sinica, 2007, 35(2): 341 – 344. (in Chinese)

[3] 杨勇.液压伺服系统自适应模糊变结构控制[J].电子学报,2008,36(1):86 – 89.
Yang Y. Adaptive fuzzy variable structure control for hydraulic servo system[J]. Acta Electronica Sinica, 2008, 36(1): 86 – 89. (in Chinese)

[4] Sakawa M, Mori T. An efficient genetic algorithm for job shop scheduling problems with fuzzy processing time and fuzzy due date[J]. Computers & Industrial Engineering, 1999, 36(2): 325 – 341.

[5] Sakawa M, Kubota R. Fuzzy programming for multi-objective job shop scheduling with fuzzy processing time and fuzzy due date through genetic algorithm[J]. European Journal of Operational Research, 2000, 120(2): 393 – 407.

[6] Song X Y, Zhu Y L, et al. Study on the combination of genetic algorithms and ant colony algorithms for solving fuzzy job shop scheduling problems[A]. Proceedings of IMACS Multi-conferences on Computational Engineering in Systems Applications [C]. Piscataway: IEEE, 2006. 1904 – 1909.

[7] Wu C S, Li D C, Tsai T I. Applying the fuzzy ranking method to the shifting bottleneck procedure to solve scheduling problems of uncertainty[J]. International Journal of Advanced Manufacturing Technology, 2006, 31(1 – 2): 98 – 106.

[8] Niu Q, Jiao B, Gu XS. Particle swarm optimization combined with genetic operators for job shop scheduling problem with fuzzy processing time[J]. Applied Mathematics and Computation, 2008, 205(1): 148 – 158.

[9] Lei D M. Pareto archive particle swarm optimization for multi-objective fuzzy job shop scheduling problems[J]. International Journal of Advanced Manufacturing Technology, 2008, 37(1 – 2): 157 – 165.

[10] González-Rodríguez I, Puente J, et al. Semantics of schedules for the fuzzy job-shop problem[J]. IEEE Transactions on Systems, Man, and Cybernetics, 2008, 38(3): 655 – 666.

[11] Lei D M, Guo X P. Solving fuzzy flexible job shop scheduling problems using genetic algorithm[A]. Proceeding of International Conference on Machine Learning and Cybernetics[C]. Piscataway: IEEE, 2008. 1014 – 1019.

[12] Lei D M. A genetic algorithm for flexible job shop scheduling with fuzzy processing time[J]. International Journal of Production Research, 2010, 48(10): 2995 – 3013.

[13] Goldberg D E, Deb K. A comparative analysis of selection schemes used in genetic algorithms[A]. Rawlins G. Foundations of genetic algorithms[C]. San Mateo, California: Morgan Kaufmann, 1991. 69 – 93.

作者简介

郑友莲 女,1972 年 10 月生于湖北省石首市.副教授、武汉大学软件工程国家重点实验室博士生.主要研究方向:智能计算、生产调度.
E-mail: arren1@qq.com

李元香 男,1962 年生于湖北省监利县.武汉大学教授、博士生导师.主要研究方向:智能计算、并行计算.
E-mail: yxli@whu.edu.cn

雷德明 男,1968 年 5 月出生于湖北省石首市.武汉理工大学副教授.主要研究方向:生产调度、智能优化.
E-mail: deminglei11@163.com