

文章编号: 1001-0920(2007)11-1228-07

一种求解双目标 job shop 问题的混合进化算法

师瑞峰^a, 周一民^a, 周 泓^b

(北京航空航天大学 a. 计算机学院, b. 经济管理学院, 北京 100083)

摘要: 提出一种求解双目标 job shop 排序问题的混合进化算法. 该算法采用改进的精英复制策略, 降低了计算复杂性; 通过引入递进进化模式, 避免了算法的早熟; 通过递进过程中的非劣解邻域搜索, 增强了算法局部搜索性能. 采用该算法和代表性算法 NSGA-II, MOGLS 对 82 个标准双目标 job shop 算例进行优化对比, 所得结果验证了该算法求解双目标 job shop 排序问题的有效性.

关键词: 多目标优化; 递进进化; job shop; 进化算法

中图分类号: F406.6

文献标识码: A

A hybrid evolutionary algorithm for bi-objective job shop scheduling problems

SHI Rui-feng^a, ZHOU Yi-min^a, ZHOU Hong^b

(a. School of Computer Science and Engineering, b. School of Economics and Management, Beihang University, Beijing 100083, China. Correspondent: SHI Rui-feng, E-mail: shi.ruifeng@buaa.edu.cn)

Abstract: Aiming at solving bi-objective job shop scheduling problems, a hybrid evolutionary algorithm is proposed. An improved elite duplication strategy is applied, which reduces computational cost of the algorithm. An escalating evolutionary strategy is introduced into the algorithm, which is designed to overcome premature convergence. Besides, by applying a variable neighborhood search strategy to achieve Pareto solutions during the population escalation, the algorithm's local search ability is enhanced. Numerical experiments, which employ the proposed algorithm, together with other two typical algorithms NSGA-II and MOGLS, is made to solve 82 bi-objective job shop scheduling problems. The optimization results show the effectiveness of the algorithm proposed here on solving bi-objective job shop scheduling problems.

Key words: Multi-objective optimization; Escalating evolution; Job shop; Evolutionary algorithm

1 引言

多目标 job shop 排序问题是一类典型的组合优化难题, 对这类问题的传统解法有以瓶颈转移法为代表的启发式算法和以分支定界法为代表的精确解法两大类^[1]. 精确解法尽管从理论上可以获得问题的最优解, 但解空间往往随问题规模的增长呈指数增加, 因此难以在可接受的时间内获得较大规模问题的最优解^[2]. 启发式算法尽管能在很短的时间内给出求解问题的一个可行解, 但通常无法保证解的最优性, 并且启发式算法通常只对特定的单目标优化问题有效, 难以推广应用于多目标优化问题^[3]. 因此, 利用现代启发式算法对 job shop 问题进行多

目标优化, 已成为该领域所关注的焦点^[4].

多目标进化算法是 20 世纪 80 年代中期发展起来的一类优化技术, 自它诞生起就受到排序研究领域的广泛关注^[5-7]. Murata^[8] 首先提出一种采用变系数加权的遗传算法求解多目标 flow shop 问题; Bagchi^[9] 对采用多目标进化算法求解生产排序问题进行了大量的算例分析和比较研究.

本文在现有研究基础上, 提出一种求解双目标 job shop 问题的混合进化算法 (HEMEA): 通过精英复制, 既有效保留算法已搜索的非劣解集信息, 又降低了计算复杂性; 通过递进进化策略, 避免了算法早熟, 增强了算法收敛性. 采用本文算法对 82 个标准双目标 job shop 算例进行优化, 所得结果验证了

收稿日期: 2006-08-07; 修回日期: 2006-10-18.

基金项目: 国家自然科学基金项目 (70771003, 70521001); 新世纪优秀人才支持计划项目 (NCET).

作者简介: 师瑞峰 (1977—), 男, 山西河津人, 博士后, 从事多目标优化、进化算法的研究; 周泓 (1965—), 男, 武汉人, 教授, 博士生导师, 从事进化算法、生产计划等研究.

算法的有效性.

2 双目标 job shop 问题描述

Job shop 问题可描述为: n 项待加工作业 $J = \{1, 2, \dots, n\}$ 需通过 m 台机器 $M = \{1, 2, \dots, m\}$ 进行加工, 各作业分别通过各机器加工且仅加工一次, 各作业按照各自指定的工艺路线进行加工, 同一时刻每台机器上最多加工一个作业, 且作业加工不允许中断. 通常, 单目标 job shop 问题的优化目标有最小化 makespan, total flowtime, total tardiness 等, 这些目标分别反映了工件的完工指标及作业安排的紧致程度. 本文双目标 job shop 排序的优化目标为同时最小化 makespan 和 total tardiness, 优化模型为

$$(\min f_1, \min f_2) = (\min(C_{\max}), \min_j T_j) = (\min_{FS}(\max_j C_j), \min_{FS}(\max_{j=1}^n(\max\{0, c_j - d_j\}))), \quad (1)$$

$$s. t. \quad t_{ja}^S + p_{ja} \leq t_{jb}^S + G(1 - q_{jab}), \quad a, b = 1, 2, \dots, m, \quad j = 1, 2, \dots, n; \quad (2)$$

$$t_{gi}^S + p_{gi} \leq t_{hi}^S + G(1 - x_{ghi}), \quad i = 1, 2, \dots, m, \quad g, h = 1, 2, \dots, n; \quad (3)$$

$$t_{ji}^S \geq 0, \quad i = 1, 2, \dots, m, \quad j = 1, 2, \dots, n; \quad (4)$$

$$x_{ghi} = 0 \text{ 或 } 1, \quad i = 1, 2, \dots, m, \quad g, h = 1, 2, \dots, n. \quad (5)$$

模型中的参数含义为: FS 为可行排序集合; o_{ji} 为作业 j 的第 i 道工序的加工操作; p_{ji} 为操作 o_{ji} 的加工时间; G 为任意大的正数;

$$q_{jab} = \begin{cases} 1, & \text{作业 } j \text{ 在机器 } a \text{ 上加工} \\ & \text{后紧接着在 } b \text{ 上加工;} \\ 0, & \text{否则.} \end{cases}$$

模型中的变量含义为: t_{ji}^S 为操作 o_{ji} 的开始时间; C_j 为作业 j 的完工时间, 因此有 $C_j = \max_{i \in M} \{t_{ji}^E\}$, 其中 t_{ji}^E 为操作 o_{ji} 的结束时间, 且 $t_{ji}^E = t_{ji}^S + p_{ji}$;

$$x_{ghi} = \begin{cases} 1, & \text{作业 } g \text{ 在机器 } i \text{ 上的紧后作业为 } h; \\ 0, & \text{否则.} \end{cases}$$

约束(2)表示了作业 j 的两个连续操作 o_{ja} 和 o_{jb} 的优先关系: 操作 o_{jb} 的开始时间大于等于操作 o_{ja} 的结束时间. 约束(3)表明, 在某一时刻, 机器 i 上最多只能加工一项作业.

3 递进多目标进化算法

3.1 进化编码及其操作

在已有求解 job shop 问题的遗传算法中, 主要有 9 种较为成熟的编码方案^[10]. 本文选用基于操作的编码方式, 这种编码的优点是可将某一机器加工不同工件的先后次序信息以及各工件的工艺路线信

息同时体现于编码中.

基于操作的编码可描述为: 对于工件数为 n , 机器数为 m 的 job shop 问题, 用一列长度为 $n \times m$, 且每个位置元素取值为 $1 \sim n$ 自然数序列排列的数字串表示一个排序解, 即 $[a_1, a_2, \dots, a_i, \dots, a_{n \times m}]$, 其中 $a_i \in \{1, 2, \dots, n\}$, 且满足

$$\sum_{i=1}^{n \times m} i(j) = m, \quad j = 1, 2, \dots, n. \quad (6)$$

这里

$$i(j) = \begin{cases} 1, & a_i = j; \\ 0, & \text{否则;} \end{cases} \quad i = 1, 2, \dots, n \times m, \quad j = 1, 2, \dots, n. \quad (7)$$

式(6)和(7)表明, 编码中重复出现每个字符 $j(j = 1, 2, \dots, n)$ 的次数恰好等于 m 次, 从而保证了解码后的工序与工艺路线一一对应, 即保证了编码的合法性.

Gen 和 Cheng^[10,11]总结了现有 job shop 各编码对应的交叉和变异操作算子. 本文选用基于顺序的交叉算子和移位变异算子作为进化操作算子. 关于这两种操作算子的实现方法详见文献[10,11].

3.2 精英复制

本文提出一种不需计算个体适应值(仅依据目标值)、仅通过非劣解等级优先原则与同一等级内稀疏个体优先原则, 引导搜索的精英复制策略. 该策略的思想为: 在每代进化结束时合并新旧群体, 剔除混合群体中的重复个体; 然后按照混合群体中个体非劣解等级由低到高逐级复制到下一代进化群体. 若某一等级群体复制进入下一代群体后进化群体规模超过规定规模, 则按个体稀疏优先原则将该等级个体逐一复制到下一进化群体, 直至进化群体达到规定规模.

精英复制策略实现过程可描述如下:

1) 设当前群体为 P , 群体规模为 N , 对 P 进化操作生成的群体为 P' , 新旧群体合并后的混合群体为 P , 下一代进化群体为 Q ;

2) 令 $P = P \cup P'$, 对 $\forall x_i \in P$, 若 $\exists x_k \in P, k \neq i$, 且满足 $x_i = x_k$, 则 $P = P \setminus x_k$;

3) 依据目标值对 P 进行 Pareto 分级, 得到

$S_i = P$, 且 $S_1 \supset S_2 \supset \dots \supset S_r$, 其中 r 为 P 的非劣解等级数, 每一等级的个体数为 n_1, n_2, \dots, n_r , 令 $t = 1$;

4) 若 $\sum_{k=1}^t n_k < N$, 则 $Q = Q \cup S_t, t = t + 1$, 重复 4); 否则转 5);

5) 计算 S_t 内任意个体 x_j 与其他个体的距离

$$d_j = \sqrt[n]{d_{jk}^n}, \quad (8)$$

并按 d_j 从大到小的顺序重新排列各 x_j , 得到 $x_{[1]}, x_{[2]}, \dots, x_{[n]}$, 令 $l = 1$; 其中

$$d_{jk} = \sqrt[m]{[f_l(x_j) - f_l(x_k)]^2}, \quad (9)$$

$x_{[l]} \in \{x_j\}$, 且满足 $d_{[1]} > d_{[2]} > \dots > d_{[n]}$;

6) 若 $\sum_{k=1}^{l-1} n_k + l \leq N$, 则令 $Q = Q \cup x_{[l]}, l = l + 1$, 重复 6); 否则, 选择完毕.

上述步骤 2) 中, 新旧群体合并后按照非劣解等级优先复制, 保证了已搜索到的精英个体不会在其后进化中遗失; 剔除混合群体中的重复个体, 旨在保证进化群体的多样性, 避免精英个体在后代群体中大量繁殖而导致算法早熟.

3.3 递进进化结构

Coello 等^[12] 于 2001 年提出一种基于微小群体、重复精英保留策略的小生境群体遗传算法 (micro GA), 并采用该方法成功地求解了一些多目标函数优化问题. 基于文献^[13] 的研究, 本文提出一种改进的递进进化算法: 将整个进化过程“分层”优化, 每次递进都以一定策略对群体加以重构, 以避免算法早熟; 通过外部伴随群体对不同递进进化层间的非劣解信息进行保留和交换, 以保证算法的收敛性; 通过递进过程中的非劣解局部搜索, 加速非劣解前沿的进化.

此处, 群体递进过程首先从外部伴随群体中选取一定比例的个体进入下一层进化群体, 以保证算法进化过程的延续性; 然后初始化新进化群体中的其余个体, 避免算法可能出现的早熟收敛. 本文的递进进化算法如图 1 所示.

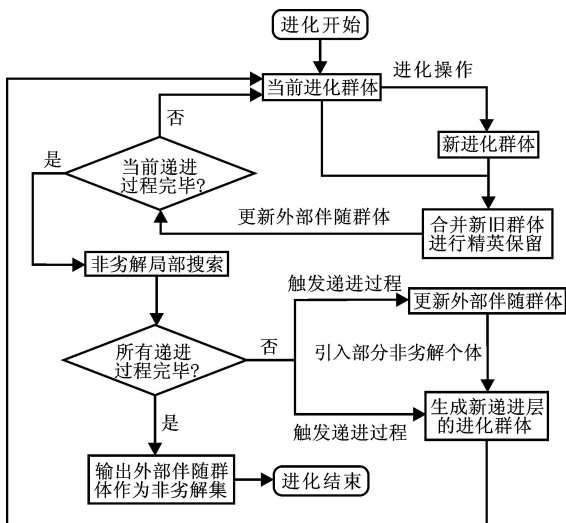


图 1 递进进化算法框架

3.4 可变邻域局部搜索

由于现有进化算法在求解问题时难以对“潜在最优解”进行有效的局部搜索, 本文提出一种非劣解可变邻域局部搜索策略. 可变邻域搜索的思想最早由 Hansen 和 Neland 等^[14] 提出, 本文将这一思想引入群体递进过程, 对递进过程中的外部伴随群体非劣解进行局部搜索. 该搜索算法可描述如下 (设外部伴随非劣解集为 A):

1) 将移位变异算子应用于当前非劣解 x_l , 得到邻域新解 x_l' , 若 $x_l' > x_l$, 则转 2); 若 $x_l' \sim x_l$, 则转 3); 否则, 转 4).

2) 令 $x_l = x_l'$, 采用文献^[10] 中的邻位置换变异算子对 x_l 进行指定步长的微小局部搜索. 若微小邻域解 $x_l' > x_l$, 则 $x_l = x_l'$, 继续剩余的微小局部搜索, 并转 4).

3) $A = A \cup x_l$, 对于 $\forall y_i \in A$, 若 $\exists y_k \in A, k \neq i$, 且满足 $y_i > y_k$, 则 $A = A \setminus y_k$.

4) 若 x_l 未达最大邻域搜索步数, 则转 1).

5) 若 A 中所有个体局部搜索完毕, 则算法停止; 否则转 1).

本文可变邻域局部搜索算法的特点是: 有效利用了进化算法的求解效率, 即仅当局部搜索发现更好解时, 才对其邻域进行精细搜索, 避免了邻域搜索的盲目性; 仅对递进过程中的伴随非劣解集进行邻域搜索, 避免对大量个体进行邻域搜索, 在保证进化搜索效率的前提下, 最大程度地提高局部搜索效率.

3.5 混合进化算法流程

图 1 的混合进化算法流程如下:

- 1) 令外部伴随非劣解集为空集;
- 2) 随机生成初始进化群体, 评价个体的各目标值;
- 3) 通过交叉和变异, 生成新群体, 评价新群体中个体的各目标值;
- 4) 合并新、旧群体, 在目标空间内进行非劣解分级, 按照精英复制策略确定下一代进化群体;
- 5) 更新外部伴随非劣解集, 若当前进化层进化完成, 转 6); 否则转 3);
- 6) 对外部伴随群体进行可变邻域局部搜索;
- 7) 若所有递进层进化完成, 则输出外部伴随非劣解集, 算法结束; 否则, 重构进化群体, 转 3).

3.6 算法复杂性分析

本文算法复杂性如表 1 所示.

本文混合进化算法的复杂度为

$$o(\text{HEMEA}) \cong$$

$$G_{\max} E_{\max} \left(N + \left(N + \frac{n_L n}{G_{\max}} \right) f_k + (2N)^2 (1 + f_d) + N_A^2 \right) \quad (10)$$

因个体目标评价时间 f_k 与同一非劣解等级内

表 1 混合进化算法复杂性分析

步骤	复 杂 性
1)	N
2)	$N + N f_k$
3)	$G_{\max} E_{\max} N + G_{\max} E_{\max} N f_k$
4)	$G_{\max} E_{\max} (2N)^2 + G_{\max} E_{\max} (2N)^2 f_d$
5)	$G_{\max} E_{\max} N_A^2$
6)	$E_{\max} (n_L n) f_k$
7)	$o(0)$

注： E_{\max} 为总递进层数， G_{\max} 为每层进化代数， N 为群体规模， N_A 为外部伴随群体规模， f_k 为评价单个个体目标值的时间花费， f_d 为计算个体之间距离的时间花费， n_L 为邻域搜索最大步长， n 为 job shop 问题的工件数。

两个体间距离评价时间 f_d 的数量级高于其他时间花费，故本文算法复杂性可进一步简约为

$$o(\text{HEMEA}) \cong G_{\max} E_{\max} \left((N + n_L n / G_{\max}) f_k + (2N)^2 (1 + f_d) \right) \cong G_{\max} E_{\max} (N f_k + (2N)^2 f_d) \cong G_{\max} E_{\max} (2N)^2 f_d \cong G_{\max} E_{\max} (4N^2) \quad (11)$$

式(11)表明，本文算法在最坏求解情况下(去除高阶小量后)的计算复杂度为 $o(4N^2)$ ，属多项式时间算法。这一计算复杂度与代表性算法 NSGA-，MOGLS 等的计算复杂度相当(这些算法的计算复杂性为 $o(MN^2)$ ^[15]。其中 M 为优化目标个数，本文 $M = 2$ 。因此，本文算法的求解时间约为这两种算法的 2 倍)。

4 算例分析

4.1 算例描述

为检验本文算法求解双目标 job shop 排序问题的有效性，选取 OR_library 发布的 82 个 job shop 算例(包括 5 个 ABZ 系列算例，3 个 FT 系列算例，40 个 LA 系列算例，10 个 ORB 系列算例，20 个 SWV 系

表 2 采用 3 种算法优化双目标 job shop 算例集的进化参数

算例	Index	POP	GEN	P_C	P_M	E_{\max}	L_{\max}	算例	Index	POP	GEN	P_C	P_M	E_{\max}	L_{\max}
1	ABZ5	200	800	0.9	0.1	4	15	...							
2	ABZ6	200	800	0.9	0.1	4	15	40	LA35	200	2 400	0.8	0.2	8	35
3	ABZ7	200	1 500	0.9	0.1	5	20	41	LA36	200	1 200	0.8	0.2	6	20
4	ABZ8	200	1 500	0.9	0.1	5	20	...							
5	ABZ9	200	1 500	0.9	0.1	5	20	45	LA40	200	1 200	0.8	0.2	6	20
6	LA01	200	500	0.6	0.3	5	10	46	ORB01	200	800	0.65	0.2	4	15
...								...							
10	LA05	200	500	0.6	0.3	5	10	55	ORB10	200	800	0.65	0.2	4	15
11	LA06	200	600	0.6	0.3	6	15	56	SWV01	200	1 200	0.8	0.15	6	20
...								...							
15	LA10	200	600	0.6	0.3	6	15	60	SWV05	200	1 200	0.8	0.15	6	20
16	LA11	200	1 000	0.6	0.3	5	20	61	SWV06	200	1 500	0.8	0.15	5	20
...								...							
20	LA15	200	1 000	0.6	0.3	5	20	65	SWV10	200	1 500	0.8	0.15	5	20
21	LA16	200	800	0.6	0.3	4	15	66	SWV11	200	4 800	0.9	0.1	8	50
...								...							
25	LA20	200	800	0.6	0.3	4	15	75	SWV20	200	4 800	0.9	0.1	8	50
26	LA21	200	1 000	0.8	0.2	5	20	76	YN1	200	1 500	0.75	0.15	5	20
...								...							
30	LA25	200	1 000	0.8	0.2	5	20	79	YN4	200	1 500	0.75	0.15	5	20
31	LA26	200	1 200	0.8	0.2	6	20	80	FT06	200	400	0.9	0.1	4	10
...								81	FT10	200	800	0.9	0.1	4	15
35	LA30	200	1 200	0.8	0.2	6	20	82	FT20	200	1 000	0.9	0.1	5	20
36	LA31	200	2 400	0.8	0.2	8	35								

注：POP 为进化的群体规模，GEN 为进化代数， P_C 为交叉概率， P_M 为变异概率， E_{\max} 为递进层数， L_{\max} 为 MOGLS 和 HEMEA 算法中的邻域搜索最大步长。

列算例和 4 个 YN 系列算例),对本文算法进行验证.

由于 OR_library 公布的算例未包含交货期信息,本文参照文献[16]中 TWK 的思想,利用下式计算各算例中的工件交货期:

$$d_j = k \prod_{i=1}^m p_{ji}, j = 1, 2, \dots, n. \quad (12)$$

其中 k 为 TWK 方法中确定误工比例的常数,此处取 $k = 1.5$ (即工件平均延误比例约为 50%).

Deb 等^[17]提出的 NSGA- 算法,是迄今为止公认的最为有效的多目标进化算法之一,而 MOGLS^[18]是最早用来求解多目标排序问题的进化算法.因此,本文采用 NSGA- 和 MOGLS 两种算法作为参照算法,验证本文混合进化算法的性能.

4.2 进化参数分析及设定

为使算法 NSGA- 和 MOGLS 与本文 HEMEA 算法的计算结果的比较具有公平性,参照文献[8]的进化参数实验设计方法,通过参数实验分析,对于每种类型的算例,采用不同算法在每组参数下各自优化 10 次,然后对结果进行分析,最终确定出各算法优化不同算例的进化参数(如表 2 所示).

4.3 结果分析

分别采用 HEMEA, NSGA- 和 MOGLS 算法对上述 82 个算例各自独立优化 20 次,将各次运行所得的非劣解集合并后,剔除其中的劣解,得到 3 种算法求解各算例的非劣解集.图 2~5 给出了几个典型算例的非劣解集.

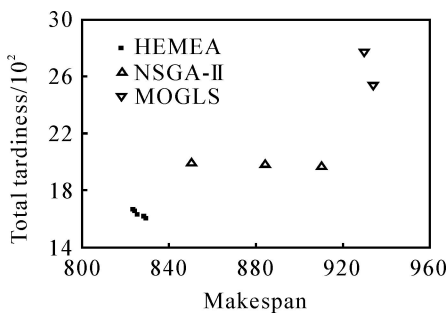


图 2 3 种算法优化 ABZ9 获得的非劣解集

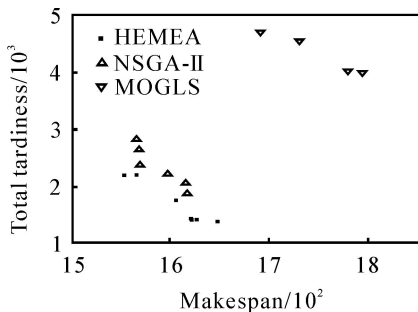


图 3 3 种算法优化 LA37 获得的非劣解集

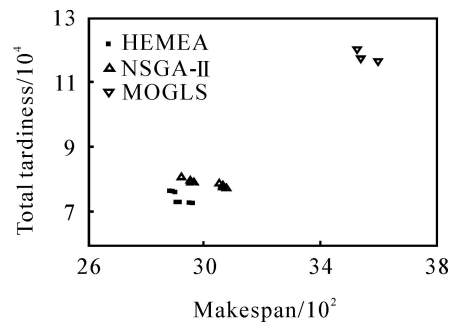


图 4 3 种算法优化 SWV19 获得的非劣解集

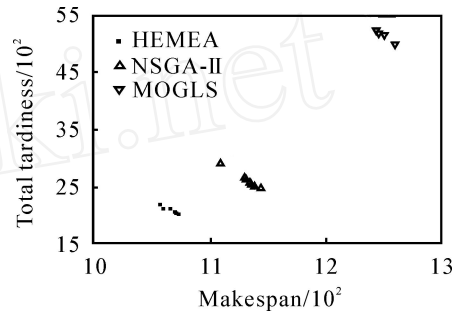


图 5 3 种算法优化 YN1 获得的非劣解集

采用 3 种算法对所有 82 个 job shop 算例进行双目标优化的最终结果见表 3,该表给出了 3 种算法对上述算例优化所得非劣解集之间的相对优劣关系(以 C 指标统计结果描述).

表 3 中,评价 3 种算法两两之间非劣解集优劣的 C 指标由 Zitzler^[19]提出,其定义为

$$C(A, B) = \frac{| \{ b \in B \mid \exists a \in A : a \succ b \} |}{| B |} \quad (13)$$

其中 $A, B \subseteq X$ 是两种算法优化获得的非劣解集. C 指标是一种值域定义在 $(0, 1)$ 上、用来刻画 (A, B) 之间偏序性能的指标: $C(A, B) = 1$,说明对于 B 集任一非劣解, A 集总存在“优于”它的解个体; $C(A, B) = 0$,说明对于 B 集中任一解个体, A 集中都不存在“优于”它的解个体.

从图 2~5 中 3 种算法优化各算例获得的非劣解集分布图可以看出,本文 HEMEA 算法所得的非劣解集明显优于 NSGA- 和 MOGLS 所得的非劣解集;从表 3 的 C 统计结果可以看出,HEMEA 在所有 82 个算例优化中,有 68 个算例的非劣解集优于 NSGA- ,13 个算例结果非劣于 NSGA- ,仅有 1 个算例的非劣解集劣于 NSGA- ;而在所有 82 个算例优化结果中,HEMEA 仅有 2 个算例结果与 MOGLS 互为非劣,其余算例结果均优于 MOGLS.从而表明,本文算法在双目标 job shop 算例求解中,比现有公认的性能优越的 NSGA- 和 MOGLS 算法取得了更好的优化结果,从而验证了本文算法的有效性.

表 3 采用 3 种算法优化双目标 job shop 算例结果

算例	Index	$C(A, B)$	$C(A, C)$	$C(B, A)$	$C(C, A)$	算例	Index	$C(A, B)$	$C(A, C)$	$C(B, A)$	$C(C, A)$
1	ABZ5	1	1	0	0	37	LA32	0	1	0	0
2	ABZ6	0.5	1	0.18	0	38	LA33	1	1	0	0
3	ABZ7	0.67	1	0	0	39	LA34	1	1	0	0
4	ABZ8	1	1	0	0	40	LA35	0.5	1	0	0
5	ABZ9	1	1	0	0	41	LA36	1	1	0	0
6	LA01	0.8	1	0.33	0	...					
7	LA02	1	1	0	0	44	LA39	1	1	0	0
8	LA03	1	1	0	0	45	LA40	0.67	1	0	0
9	LA04	0.33	1	0.6	0	46	ORB01	1	1	0	0
10	LA05	0.8	1	0	0	47	ORB02	0.5	1	0.33	0
11	LA06	1	1	0	0	48	ORB03	0.5	1	0	0
12	LA07	0.8	1	0	0	49	ORB04	0.67	1	0	0
13	LA08	1	1	0	0	50	ORB05	1	1	0	0
14	LA09	1	1	0	0	51	ORB06	1	1	0	0
15	LA10	0	1	0	0	52	ORB07	0	0	0	0
16	LA11	0.25	1	0	0	53	ORB08	0.67	1	0	0
17	LA12	1	1	0	0	54	ORB09	0	1	0	0
18	LA13	1	1	0	0	55	ORB10	0.5	1	0	0
19	LA14	1	1	0	0	56	SWV01	0.56	1	0	0
20	LA15	0.5	1	0	0	57	SWV02	1	1	0	0
21	LA16	0.78	0.5	0.14	0.29	58	SWV03	1	1	0	0
22	LA17	1	1	0	0	59	SWV04	0	1	0.38	0
23	LA18	0.33	1	0.5	0	60	SWV05	0.75	1	0	0
24	LA19	0.5	0.67	0.25	0	61	SWV06	1	1	0	0
25	LA20	1	1	0	0	...					
26	LA21	1	1	0	0	69	SWV14	1	1	0	0
27	LA22	0.6	1	0.29	0	70	SWV15	0.33	1	0	0
28	LA23	1	1	0	0	71	SWV16	1	1	0	0
29	LA24	1	1	0	0	...					
30	LA25	1	1	0	0	76	YN1	1	1	0	0
31	LA26	0.78	1	0	0	77	YN2	0	1	0	0
32	LA27	1	1	0	0	78	YN3	1	1	0	0
33	LA28	0.55	1	0	0	79	YN4	1	1	0	0
34	LA29	1	1	0	0	80	FT06	0	1	0	0
35	LA30	0.75	1	0	0	81	FT10	1	1	0	0
36	LA31	0.25	1	0	0	82	FT20	0	1	0	0

注：表中 C 指标描述中，符号 A 代表 HEMEA 优化获得的非劣解集， B 代表 NSGA- 优化获得的非劣解集， C 代表 MOGLS 算法优化获得的非劣解集。

5 结 语

本文提出一种求解多目标 job shop 排序的混合进化算法。算法采用改进的精英复制策略和递进进化策略，获得了良好的收敛结果。采用本文算法 (HEMEA) 及现有代表性算法 NSGA- 和 MOGLS

对 82 个双目标 job shop 算例进行优化，研究结果表明，本文算法获得了明显优于其他两种算法的非劣解集，从而验证了本文算法的有效性。如何对本文方法加以改进，将其应用于实际生产排序问题的多目标优化求解，是进一步研究的课题。

参考文献(References)

- [1] Pinedo M. Scheduling — Theory, algorithms and systems[M]. New Jersey: Prentice Hall, 1995.
- [2] 唐恒永, 赵传立. 排序引论[M]. 北京: 科学出版社, 2002.
(Tang Heng-yong, Zhao Chuan-li. Introduction to scheduling[M]. Beijing: Science Press, 2002.)
- [3] Nagar A, Haddock J, Heragu S. Multiple and bicriteria scheduling: A literature survey [J]. European J of Operational Research, 1995, 81(1): 88-104.
- [4] Tkindt V, Billaut J C. Multicriteria scheduling: Theory, models and algorithms[M]. Berlin: Springer, 2002.
- [5] 谢涛, 陈火旺, 康立山. 多目标优化的演化算法[J]. 计算机学报, 2003, 26(8): 997-1003.
(Xie Tao, Chen Huo-wang, Kang Li-shan. Evolutionary algorithms of multi-objective optimization problems[J]. Chinese J of Computers, 2003, 26(8): 997-1003.)
- [6] 王凌. 车间调度及其遗传算法[M]. 北京: 清华大学出版社, 2003.
(Wang Ling. Shop scheduling and genetic algorithms [M]. Beijing: Tsinghua University Press, 2003.)
- [7] Brizuela C, Sannomiya N, Zhao Y. Multi-objective flow-shop: Preliminary results[C]. The Proc of 1st Int Conf. Zurich, 2001:443-457.
- [8] Murata T. Genetic algorithms for multi-objective optimization[D]. Osaka: Osaka Prefecture University, 1997.
- [9] Bagchi T P. Multiobjective scheduling by genetic algorithms[M]. Boston: Kluwer Academic Publisher, 1999.
- [10] Gen M, Cheng R W. Genetic algorithms and engineering design [M]. New York: John Wiley & Sons, 1996.
- [11] Cheng R W, Gen M, Tsujimura Y. A tutorial survey of job-shop scheduling problems using genetic algorithms —Part 1: Hybrid genetic search strategies [J]. Computers and Industrial Engineering, 1999, 36(1): 343-364.
- [12] Ceollo C A C, Pulido G T. A micro-genetic algorithm for multiobjective optimization[C]. The Proc of 1st Int Conf on Evolutionary Multi-criteria Optimization. Zurich, 2001: 126-140.
- [13] 师瑞峰, 周泓, 谭小卫. 递进多目标遗传算法[J]. 系统工程理论与实践, 2005, 25(12): 48-56.
(Shi Rui-feng, Zhou Hong, Tan Xiao-wei. A multi-objective genetic algorithm based on escalating strategy [J]. Systems Engineering — Theory & Practice, 2005, 25(12): 48-56.)
- [14] Hansen P, Nenalid M. Variable neighborhood search: Principles and applications [J]. European J of Operational Research, 2001, 130(3): 449-467.
- [15] Deb K. Multi-objective optimization using evolutionary algorithms [M]. Chichester: John Wiley & Sons, 2001.
- [16] Sabuncuoglu I, Bayiz M. Job shop scheduling with beam search[J]. European J of Operational Research, 1999, 118(2): 390-412.
- [17] Deb K, Pratap A, Agarwal S, et al. A fast and elitist multiobjective genetic algorithm: NSGA- [J]. IEEE Trans on Evolutionary Computation, 2002, 6(2): 182-197.
- [18] Ishibuchi H, Yoshida T, Murata T. Balance between genetic search and local search in memetic algorithms for multiobjective permutation flowshop scheduling[J]. IEEE Trans on Evolutionary Computation, 2003, 7(2): 204-223.
- [19] Zitzler E. Evolutionary algorithms for multiobjective optimization: Methods and applications [D]. Switzerland: Swiss Federal Institute of Technology, 1999.

下期要目

- 故障诊断的信息融合方法 朱大奇, 刘永安
- 一种直觉模糊熵的构造方法 王毅, 雷英杰
- 基于广义扩展优势关系的粗糙决策分析方法 胡明礼, 刘思峰
- 双层进化交互式遗传算法的知识提取与利用 郭一楠, 巩敦卫
- 知识化制造系统中动态调度的自适应策略研究 杨宏兵, 严洪森
- 奖励收集斯坦利最小树的混合拉格朗日与分散搜索算法 潘常春, 杨根科
- 基于反步法的一类非线性切换系统控制器设计 向嵘嵘, 向伟铭