

文章编号:1001-9081(2015)06-1617-06

doi:10.11772/j.issn.1001-9081.2015.06.1617

教与同伴学习粒子群算法求解多目标柔性作业车间调度问题

吴定会^{1,2},孔飞^{1*},田娜³,纪志成¹

(1. 轻工过程先进控制教育部重点实验室(江南大学),江苏 无锡 214122;

2. 江苏省食品先进制造装备技术重点实验室(江南大学),江苏 无锡 214122; 3. 江南大学 教育技术系,江苏 无锡 214122)

(* 通信作者电子邮箱 kongfei0608@126.com)

摘要:针对多目标柔性作业车间调度问题,提出了带 Pareto 非支配解集的教与同伴学习粒子群算法。首先,以工件的最大完工时间、最大机器负荷和所有机器总负荷为优化目标建立了多目标柔性作业车间调度模型。然后,该算法结合多目标 Pareto 方法和教与同伴学习粒子群算法,采用快速非支配排序算法产生初始 Pareto 非支配解集,用提取 Pareto 支配层程序更新 Pareto 非支配解集,同时采用混合分派规则产生初始种群,采用开口向上抛物线递减的惯性权重选择策略提高算法的收敛速度。最后,对 3 个 Benchmark 算例进行仿真实验。理论分析和仿真表明,与带向导性局部搜索的多目标进化算法(MOEA-GLS)和带局部搜索的控制遗传算法(AL-CGA)相比,对于相同的测试实例,该算法能产生更多更好的 Pareto 非支配解;在计算时间方面,该算法要小于带向导性局部搜索的多目标进化算法。实验结果表明该算法可以有效解决多目标柔性作业车间调度问题。

关键词:多目标;柔性作业车间调度;Pareto 非支配解集;教与同伴学习粒子群;停滞阻止策略

中图分类号: TP301.6;TP18 **文献标志码:**A

Teaching and peer-learning particle swarm optimization for multi-objective flexible job-shop scheduling problem

WU Dinghui^{1,2}, KONG Fei^{1*}, TIAN Na³, JI Zhicheng¹

(1. Key Laboratory of Advanced Process Control for Light Industry, Ministry of Education (Jiangnan University), Wuxi Jiangsu 214122, China;

2. Jiangsu Key Laboratory of Advanced Food Manufacturing Equipment and Technology (Jiangnan University), Wuxi Jiangsu 214122, China;

3. Department of Educational Technology, Jiangnan University, Wuxi Jiangsu 214122, China)

Abstract: To solve multi-objective Flexible Job-shop Scheduling Problems (FJSP), a Teaching and Peer-Learning Particle Swarm Optimization with Pareto Non-Dominated Solution Set (PNDSS-TPLPSO) algorithm was proposed. First, the minimum completion time of jobs, the maximum work load of machines and the total work load of all machines were taken as the optimization goals to establish a multi-objective flexible job-shop scheduling model. Then, the proposed algorithm combined multi-objective Pareto method with Teaching and Peer-Learning Particle Swarm Optimization (TPLPSO). A fast Pareto non-dominated sorting operator was applied to generate initial Pareto non-dominated solution set, and extracting Pareto dominance layer program was adopted to update Pareto non-dominated solution set. Furthermore, composite dispatching rule was adopted to generate the initial population, and opening up parabola decreasing inertia weigh strategy was taken to improve the convergence speed. Finally, the proposed algorithm was adopted to solve three Benchmark instances. In the comparison experiments with Multi-Objective Evolutionary Algorithm with Guided Local Search (MOEA-GLS) and Controlled Genetic Algorithm with Approach by Localization (AL-CGA), the proposed algorithm can obtain more and better Pareto non-dominated solutions for the same Benchmark instance. In terms of computing time, the proposed algorithm is less than MOEA-GLS. The simulation results demonstrate that the proposed algorithm can solve multi-objective FJSP effectively.

Key words: multi-objective; flexible job-shop scheduling; Pareto non-dominated solution set; Teaching and Peer-Learning Particle Swarm Optimization (TPLPSO); Stagnation Prevention Strategy (SPS)

0 引言

车间作业调度问题(Job-shop Scheduling Problem, JSP)研究多个工件在多台机器上的加工分配问题,其目的是为每台机器找到合适的加工工序使性能指标最优^[1]。Bruker 和 Schlie 在前人研究的基础上提出了柔性作业车间调度问题

(Flexible Job-shop Scheduling Problem, FJSP),并给出了求解两个工件 FJSP 的多项式方法^[2]。FJSP 是典型的 JSP 的一个扩展,它允许工序在一组机器集上加工^[3]。由于需要考虑机器选择和工序排序问题使得 FJSP 更难处理,因此 FJSP 是一个 NP-hard 问题。

FJSP 包括单目标 FJSP 和多目标 FJSP,目前,单目标 FJSP

收稿日期:2015-01-09;修回日期:2015-03-27。

基金项目:国家 863 计划项目(2013AA040405);江苏省食品先进制造装备技术重点实验室开放课题资助项目(FM-201408)。

作者简介:吴定会(1970-),男,安徽合肥人,副教授,博士,主要研究方向:智能调度;孔飞(1986-),男,安徽合肥人,硕士研究生,主要研究方向:车间优化调度;田娜(1983-),女,河北石家庄人,副教授,博士,主要研究方向:智能计算、模式识别;纪志成(1959-),男,浙江杭州人,教授,博士生导师,博士,主要研究方向:智能调度。

得到了广泛关注,众多学者提出了各种智能算法求解该类问题^[4-5]。但在实际生产中调度问题往往包含着多个相互冲突的目标,相比单目标 FJSP,多目标 FJSP 更佳贴近生产实际^[6]。目前求解多目标 FJSP 的方法主要包括进化算法和群智能算法。进化算法在优化和搜索领域中的应用获得了成功,而且已经应用于求解多目标 FJSP:Rahmati 等^[7]提出了两种多目标进化算法求解多目标 FJSP;王伟玲等^[8]提出一种将正逆序调度方法与生成调度活动的遗传算法相结合的双种群遗传算法。近些年许多学者采用群智能算法求解多目标 FJSP,并获得了较好的求解结果:Xia 等^[9]采用粒子群算法和模拟退火算法相结合的混合算法求解多目标 FJSP;Li 等^[10]提出基于 Pareto 的离散蜂群算法,并将此算法与局部搜索方法相结合,基于 Pareto 概念对种群进行快速非支配排序。

求解多目标柔性作业车间调度问题主要有两种方法,即加权和方法和多目标 Pareto 方法。加权和方法是将多目标问题通过加权和的方式转化为单目标问题求解,这种方法虽然简单,但是每次只能获得一个解,经过多次计算才能获得最优解;多目标 Pareto 方法是将几个目标同时考虑,在求解的过程中可以产生一组 Pareto 非支配解,然后再根据决策者的喜好获得最优解,这样可以避免了多次计算,从而减少了计算时间。此外,大部分改进的粒子群算法,当粒子在搜索的过程中不能改进自身的适应度值时都不能提供学习策略。为此,本文结合教与同伴学习粒子群优化(Teaching and Peer-Learning Particle Swarm Optimization, TPLPSO)算法和多目标 Pareto 非支配解集更新算法^[11-12],提出了带 Pareto 非支配解集的教与同伴学习粒子群优化(Teaching and Peer-Learning Particle Swarm Optimization with Pareto Non-Dominated Solution Set, PNDSS-TPLPSO)算法。算法对每次迭代产生的解都进行 Pareto 非支配关系判断,用 Pareto 非支配解集存储搜索的非支配解,保证解的多样性。此外,算法采用混合分派规则产生初始种群,通过迭代最终能产生较好的非支配解;采用开口向上的抛物线递减的惯性权重更新策略,以改善 FJSP 求解的收敛速度及最终结果。通过对 FJSP 实例仿真,验证了所提出算法的可行性和有效性。

1 问题描述及数学模型

1.1 问题描述

多目标 FJSP 是一类满足任务配置和顺序约束的资源分配问题,可以描述为: n 个独立的工件安排在 m 台机器上加工,每个工件有若干道工序,能加工某一道工序的机器有多台,工序的加工时间根据机器的性能不同而变化。本调度需要解决的问题是:为每道工序分配合适的加工机器,并确定每台机器上工序的加工顺序,在满足一定的约束条件下使某些性能指标最优。

1.2 数学模型

在描述数学模型之前首先作出如下假设:

- 1) 所有机器在零时刻都是可用的;
- 2) 所有工件在零时刻都是可加工的;
- 3) 每台机器某一时刻只能处理一道工序;
- 4) 工序一旦开始加工就不能中断;
- 5) 每个工件的工序都是预先设定好的;
- 6) 不指定开始时间和完成时间;
- 7) 不考虑工件在机器间的转移时间。

首先定义如下符号和标注: i 为工件号, j 为工序号, k 为

机器号, c_{ijk} 为工件 i 的 j 道工序在机器 k 上的完成时间, t_{ijk} 为工件 i 的 j 道工序在机器 k 上的处理时间, T_k 为机器 k 的总运行时间。

1.2.1 目标函数

三个目标函数的描述如下,式(1)给出了第一个目标 Makespan,即最小化所有工件的最大完工时间;式(2)给出了第二个目标,即最小化最大机器负荷;式(3)给出了第三个目标,即最小化所有机器的总负荷。

$$f_1: \min \max \{c_{ijk}\}; i \in J, j \in n_i, k \in M \quad (1)$$

$$f_2: \min \max \left\{ T_k = \sum_{i=1}^n \sum_{j=1}^{n_i} t_{ijk} \right\} \quad (2)$$

$$f_3: \min \max \left\{ \sum_{k=1}^m T_k \right\} \quad (3)$$

1.2.2 约束条件

$$c_{ijk} - t_{ijk} \geq c_{i(j-1)p}; \\ i = 1, 2, \dots, n; j = 2, 3, \dots, n_i; k, p = 1, 2, \dots, m \quad (4)$$

$$c_{ijk} - t_{ijk} \geq c_{ghk}; i, g = 1, 2, \dots, n; j = 1, 2, \dots, n_i; \\ h = 1, 2, \dots, n_h; k = 1, 2, \dots, m \quad (5)$$

$$c_{ijk} \geq 0; i = 1, 2, \dots, n; j = 1, 2, \dots, n_i; \\ k = 1, 2, \dots, m \quad (6)$$

式(4)保证了同一工件的不同工序之间的优先约束;式(5)保证了当前机器加工完一个工序以后对于其他所有的工序都可用。

2 Pareto 教与同伴学习 PSO 求解多目标 FJSP

2.1 相关算法

2.1.1 基本粒子群算法

粒子群优化(Particle Swarm Optimization, PSO)算法是由 Kennedy 和 Eberhart 最先提出的一种基于种群的优化技术^[13],该算法的提出是基于对鸟群觅食行为的研究。PSO 的算法描述如下:由若干个粒子组成的群体在维度为 D 的搜索空间内以一定速度飞行。在求解 FJSP 时,每一粒子对应一个候选调度方案。为寻得优化解,粒子依据个体最优位置与群体最优位置调整自身飞行速度及位置。粒子位置与速度的更新公式如下:

$$v_{id}(t+1) = wv_{id}(t) + c_1 r_1(p_{id}(t) - x_{id}(t)) + \\ c_2 r_2(p_{gd}(t) - x_{id}(t)) \quad (7)$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) \quad (8)$$

其中: c_1, c_2 为学习因子; r_1, r_2 为 $[0, 1]$ 均匀分布的随机数; w 为惯性权重; $1 \leq d \leq D$; $p_{id}(t)$ 为粒子 i 在 t 时刻个体当前最好位置; $p_{gd}(t)$ 为群体当前最好位置。

2.1.2 教与学优化算法

教与学优化(Teaching-Learning-Based Optimization, TLBO)算法是 Rao 等^[14]在 2010 年提出的一种基于种群的智能优化算法,它模拟了老师给学生的教学过程和学生之间的相互学习过程,目的是通过老师的“教”和学生之间的“相互学习”来提高学生的成绩。在 TLBO 算法中,搜索空间中点的集合称为一个班级,班级中每个个体 $x_i = (x_{i1}, x_{i2}, \dots, x_{iD})$ 称为一个学生, x_i 中的每一维对应为学生所学的科目,班级中成绩最好的学生称之为老师,用 $x_{teacher}$ 来表示。TLBO 算法通过两个阶段的学习来提高每个学生的成绩:教阶段和相互学习阶段。在教阶段,位置更新式如下:

$$x_{new,i} = x_i + r(x_{teacher} - (T_F x_{mean})) \quad (9)$$

其中: r 为 $[0, 1]$ 的随机数; T_F 为教学因子,它强调了学生成

绩平均水平的重要性, T_F 的取值为 1 或 2; x_{mean} 为所有学生的平均水平。

在相互学习阶段位置更新式如下:

$$x_{\text{new},i} = \begin{cases} x_i + r(x_j - x_i), & f(x_j) \leq f(x_i) \\ x_i + r(x_i - x_j), & f(x_i) < f(x_j) \end{cases} \quad (10)$$

2.2 带 Pareto 非支配解集的教与同伴学习粒子群算法

2.2.1 教与同伴学习粒子群算法

对大部分改进的粒子群算法,当粒子在搜索的过程中不能改进自身的适应度值时都不能提供学习策略,为此,本文在教与学算法的基础上提出了教与同伴学习粒子群(TPLPSO)算法。与 TLBO 算法相类似, TPLPSO 算法也由两个阶段组成,即教阶段和同伴学习阶段。在 TPLPSO 中,每个粒子 x_i 称为一名学生,全局最好粒子 p_g 称为班级里的老师。在教阶段,每个学生通过自我学习和老师教学来提高自身的知识。在教阶段粒子的速度和位置更新如式(7)和(8)所示。

但是并不是所有学生在教阶段都能提高自身的知识,对于那些在教阶段没有提高自身知识的学生而言,提供了向同伴学习的机会。为了增加获得知识的机会,这些学生选择向比他们成绩好的学生学习,而远离比他们成绩差的学生。在同伴学习阶段,粒子的速度更新如式(11)和(12)所示,

$$v_i = \omega v_i + cr_3(p_{ei} - x_i) \quad (11)$$

$$v_i = \omega v_i - cr_4(p_{ei} - x_i) \quad (12)$$

其中: p_{ei} 为模范粒子, ω 为惯性权重, c 为学习因子, r_3 和 r_4 为 $[0,1]$ 的随机数。具体地说,当 $f(p_{ei}) \leq f(p_i)$ 时,使用式(11)更新粒子的速度;当 $f(p_{ei}) > f(p_i)$ 时,使用式(12)更新粒子的速度。此外,模范粒子 p_{ei} 采用轮盘赌选择策略产生。

在迭代的过程中使用一个停滞阻止策略(Stagnation Prevention Strategy, SPS)来防止粒子早熟收敛。具体地说,当全局最优粒子 p_g 的适应度值连续 l 次没有改进,使用一个跳出机制,使全局最优粒子 p_g 跳出局部最优位置。停滞阻止策略中使用的公式如式(13)所示:

$$p_{gd}^{\text{per}} = p_{gd} + \text{sgn}(r_5)r_6(x_{\text{max}} - x_{\text{min}}) \quad (13)$$

其中: p_{gd}^{per} 是 p_{gd} 的变异; $\text{sgn}()$ 是符号函数; r_5 是 $[-1,1]$ 均匀分布的随机数; r_6 是由正态分布 $N \sim (0, R^2)$ 产生的随机数, R 随着迭代次数的增加线性递减。 R 的更新如式(14)所示:

$$R = R_{\text{max}} - (R_{\text{max}} - R_{\text{min}})gen/Ngen \quad (14)$$

其中: R_{max} 和 R_{min} 分别是变异范围的最大和最小值, gen 和 $Ngen$ 分别是当前迭代次数和最大迭代次数。

2.2.2 Pareto 非支配解集更新算法

本文采用文献[12]中快速非支配排序算法对搜索空间中的个体进行快速非支配排序,选择第一前沿作为初始 Pareto 非支配解集,在迭代的过程中采用提取 Pareto 支配层程序更新 Pareto 非支配解集。Pareto 非支配解集更新算法流程如下:

步骤 1 参数初始化。对于种群内每个个体 i 设置两个参数 n_i (n_i 为种群中支配个体 i 的个体数量) 和 s_i (s_i 为被个体 i 所支配个体集合)。

步骤 2 找到种群中 $n_i = 0$ 的个体,将它们存入 Pareto 非支配解集,作为初始 Pareto 非支配解集。

步骤 3 算法迭代。对每个粒子的位置和速度进行更新,将更新后的粒子与 Pareto 非支配解集中每个粒子进行支配关系判断,选择与 Pareto 非支配解集中所有粒子互不支配

的个体直接加入 Pareto 非支配解集,清除 Pareto 非支配解集中被当前粒子支配的个体,然后将当前粒子加入 Pareto 非支配解集。

2.3 PNDSS-TPLPSO 算法求解多目标 FJSP

本文结合 Pareto 非支配解集更新算法和教与同伴学习粒子群算法,提出了求解多目标 FJSP 的带 Pareto 非支配解集的教与同伴学习粒子群算法。

2.3.1 粒子编码

FJSP 包括两个子问题:机器选择和工序排序。为了处理这两个子问题,本文采用分段编码方式,具体地说,采用两个 L 维(L 为所有工件的总工序数) 向量表示粒子位置,这两个 L 维向量分别为工序向量 $X_{\text{process}}[L]$ 与机器向量 $X_{\text{machine}}[L]$,其中 $X_{\text{process}}[L]$ 的每一分量所组成的序列代表相应工件的加工顺序,该分量为不大于工件总数的自然数。 $X_{\text{machine}}[L]$ 的每一分量代表加工工序 $X_{\text{process}}[L]$ 的每一工序所对应的机器号,该分量为不大于机器总数的自然数。如图 1 所示是粒子的编码图,其中第一行的 1 表示工件 1,2 表示工件 2;第一行第一次出现的 1 表示工件 1 的第一道工序,第二次出现的 1 表示工件 1 的第二道工序。第二行第一个元素 3 表示工件 1 的第一道工序在机器 3 上加工,第二个元素 2 表示工件 2 的第一道工序在机器 2 上加工。

$X_{\text{process}}[L]$	1	2	1	3	3	2	1	3	2
$X_{\text{machine}}[L]$	3	2	1	1	2	3	4	1	2

图 1 粒子编码示意图

2.3.2 解码

解码就是将粒子转化为一个有序的加工操作表,然后基于此操作表和工艺约束条件对各加工操作以最早允许的加工时间逐一进行加工,从而产生相应的调度方案,生成对应的调度甘特图。

2.3.3 粒子更新

种群内每个粒子的速度和位置都由两个 L 维的向量组成,为此采用如下步骤更新粒子的速度和位置:

步骤 1 按式(7)更新粒子的速度;

步骤 2 记录当前粒子位置中的工序向量 $X_{\text{process}}[L]$,通过式(8)更新粒子的位置;

步骤 3 对更新后的工序向量 $X_{\text{process}}[L]$ 按照从小到大的顺序排序,并将原工序向量也按照对应的顺序进行排序,用排序后的原工序向量替代 $X_{\text{process}}[L]$ 从而得到新的工序向量 $X_{\text{process}}[L]$,再根据顺序约束和机器约束逐一找到每个工序对应的加工机器,这样机器向量 $X_{\text{machine}}[L]$ 也得到了更新。

2.3.4 变异

本文提出的算法中,粒子在教阶段和同伴学习阶段都用到了变异操作。具体地,当全局最优粒子连续数次没有获得更新,就会对全局最优粒子进行变异操作,以产生更好的全局最优粒子。具体的操作步骤如下:

步骤 1 存储当前全局最好粒子位置;

步骤 2 根据式(13)和式(14)产生变异粒子;

步骤 3 对变异粒子的第一行按从小到大的顺序进行排列,并将原全局最好粒子的第一行按相同的顺序进行排列;

步骤 4 再根据顺序约束和机器约束逐一找到每个工序对应的加工机器,从而产生新的变异粒子。

2.3.5 适应度函数计算

本文含有 3 个目标函数, 将对应的粒子 x_i 从第一列到最后一列逐一解码得到机器时间向量 t 和等待时间向量 $waittime$; 然后根据机器时间向量 t 和等待时间向量 $waittime$ 求解当前粒子对应的 3 个目标函数值。第一个目标函数最大完工时间 $f_1(x_i) = \max(t)$, 机器运行时间向量 $runtime = t - waittime$; 第二个目标函数单台机器最大负荷 $f_2(x_i) = \max(runtime)$; 第三个目标函数所有机器的总负荷 $f_3(x_i) = \text{sum}(runtime)$ 。粒子 x_i 所对应的适应度函数 $f(x_i) = [f_1(x_i) \ f_2(x_i) \ f_3(x_i)]$ 。

2.3.6 算法的详细流程

带 Pareto 非支配解集的教与同伴学习粒子群算法的流程如下:

步骤 1 初始参数, 采用混合分派规则初始化种群, 采用快速非支配排序算法初始化 Pareto 非支配解集;

步骤 2 计算每个粒子的适应度值;

步骤 3 随机选取 Pareto 非支配集中某个粒子作为全局最好粒子;

步骤 4 如果当前评价次数小于最大评价次数, 使用式(16)新惯性权重, 使用式(7)和(8)更新种群中每个粒子的速度和位置;

步骤 5 计算每个粒子的适应度值, 采用提取 Pareto 支配层程序更新粒子个体历史最好位置和种群全局最好位置;

步骤 6 如果满足停滞阻止条件, 执行停滞阻止策略, 使用式(13)和(14)对全局最优粒子进行变异产生变异粒子, 并采用提取 Pareto 支配层程序判断变异粒子是否支配当前全局最优粒子, 如果满足条件, 则用变异粒子代替当前全局最优粒子;

步骤 7 如果粒子满足同伴学习条件, 则粒子执行同伴学习阶段, 采用轮盘赌策略选择模范粒子, 采用式(11)、(12)和(8)更新粒子的速度和位置;

步骤 8 计算粒子的适应度值, 采用提取 Pareto 支配层程序更新粒子的个体历史最优位置和种群的全局最优位置;

步骤 9 如果满足停滞阻止条件, 执行与步骤 6 相同的操作;

步骤 10 采用提取 Pareto 支配层程序更新 Pareto 非支配解集;

步骤 11 如果满足迭代停止条件, 则输出 Pareto 非支配解集及对应粒子的适应度值, 否则转步骤 4。

算法中判断粒子间支配关系的伪代码实现步骤如下:

步骤 1 选取两个目标粒子 $particle1$ 和 $particle2$, 分别计算这两个粒子的目标函数 $fitness1$ 和 $fitness2$, 初始化 $flag = zeros(1,3)$ 。

步骤 2 执行循环程序比较 $fitness1$ 和 $fitness2$ 对应位置元素的大小, 求出 $flag$ 。

步骤 3 如果 $flag(1) \geq 1$ 且 $flag(3) = 0$, $particle1$ 支配 $particle2$; 如果 $flag(1) = 0$ 且 $flag(3) \geq 1$, $particle2$ 支配 $particle1$; 否则 $particle1$ 和 $particle2$ 互不支配。

3 算例实验与分析

3.1 运行环境及参数设置

在酷睿 i3, CPU 主频 3.4 GHz, 4 GB 内存, Windows 7 32

位操作系统下, 用 Matlab R2011b 仿真工具实现上述算法。

本文算法的参数设置为: 种群规模 $N_{pop} = 200$, 最大评价次数 $N_{gen} = 20\,000$, 学习因子 $c1 = c2 = c = 2.0, l = 5, R_{\max} = 1, R_{\min} = 0.1, w_{\max} = 0.9, w_{\min} = 0.4$ 。

3.2 种群初始化

提高种群初始解的质量可以明显改善算法求解 FJSP 的收敛速度及最终求解结果。为此, 本文使用文献[15]提出的混合分派规则产生初始种群, 然后采用本文算法对相关实例进行求解。为了说明混合分派规则的优越性, 仿真中对比了随机初始化方法。具体操作流程以表 1 所示 4×5 实例的加工时间表来说明, 其中 4×5 实例指 4 个工件共 12 道工序在 5 台机器上加工。表 1 中数字表示加工时间, 例如第一行的第一个数字 2 表示工件 1 的第一道工序在机器 1 上的加工时间是 2, 其他数字表示的意思相同。

表 1 4×5 实例加工时间表

工件	工序	机器				
		M1	M2	M3	M4	M5
J_1	o_{11}	2	5	4	1	2
	o_{12}	5	4	5	7	5
	o_{13}	4	5	5	4	5
J_2	o_{21}	2	5	4	7	8
	o_{22}	5	6	9	8	5
	o_{23}	4	5	4	54	5
J_3	o_{31}	9	8	6	7	9
	o_{32}	6	1	2	5	4
	o_{33}	2	5	4	2	4
J_4	o_{34}	4	5	2	1	5
	o_{41}	1	5	2	4	12
	o_{42}	5	1	2	1	2

此实例共有 12 道工序, 下面详细介绍该初始化方法的实现步骤:

步骤 1 随机产生一组工件序列 [3 1 4 2], 同时产生一个初始化粒子:

$$particle = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

步骤 2 对 $particle$ 的第一行按照随机产生的工件序列顺序, 安排每一个工件的第一道工序在对应的机器上加工, 得:

$$particle = \begin{bmatrix} 3 & 1 & 4 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 3 & 4 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

步骤 3 再按照相同的工件加工顺序分别对每个工件的后续工序安排合适的机器进行加工, 如果当前工件所有工序都已经加工完毕, 则跳过当前工件, 加工下一个工件, 最后得:

$$particle = \begin{bmatrix} 3 & 1 & 4 & 2 & 3 & 1 & 4 & 2 & 3 & 1 & 2 & 3 \\ 3 & 4 & 1 & 1 & 2 & 5 & 4 & 1 & 4 & 5 & 3 & 4 \end{bmatrix}$$

每随机产生一组工件序列就会有一个与之对应的粒子产生, 这样产生的初始种群可以明显改善 FJSP 的最终求解结果。为了验证本文初始化方法的优越性, 在相同的实验环境下与随机初始化方法进行了对比。对 4×5 实例分别采用两种初始化方法初始化种群, 然后采用本文算法对该实例进行求解。求解结果如表 2 所示。

从表 2 可以看出, 与随机初始化方法相比, 采用混合分派方法产生初始种群, 然后采用本文算法求解 4×5 实例可以获得更多的非支配解, 而且得到的 3 个 Pareto 非支配解均支

配采用随机初始化方法产生的解。

表2 采用两种初始化方法的求解结果

测试 问题	混合分派方法			随机初始化方法		
	f_1	f_2	f_3	f_1	f_2	f_3
4×5	11	10	35	14	13	37
	12	9	36			
	13	8	36			

3.3 惯性权重选取

本文算法在教阶段和同伴学习阶段粒子速度更新都要用到惯性权重,因此,不同的惯性权重策略会对算法的运行时间和最终的求解结果产生很大影响。陈贵敏提出了4种惯性权重更新策略^[16],分别是线性递减策略、开口向上抛物线递减策略、开口向下抛物线递减策略和指数递减策略。这4种惯性权重更新策略如式(15)~式(18)所示。

策略1:

$$w = w_{\max} - (w_{\max} - w_{\min}) \text{gen}/Ngen \quad (15)$$

策略2:

$$w = (w_{\max} - w_{\min})(\text{gen}/Ngen)^2 + (w_{\min} - w_{\max})(2\text{gen}/Ngen) + w_{\max} \quad (16)$$

策略3:

$$w = -(w_{\max} - w_{\min})(\text{gen}/Ngen)^2 + w_{\max} \quad (17)$$

策略4:

$$w = w_{\min}(w_{\max}/w_{\min})^{(1/(1+10 \cdot \text{gen}/Ngen))} \quad (18)$$

其中: w_{\max} 为惯性权重初始值; w_{\min} 为惯性权重终止值; gen 为当前迭代次数; $Ngen$ 为最大迭代次数。将式(15)~(18)分别运用于本文算法中惯性权重的更新,在相同实验环境及算法下,对 15×10 标准测试问题进行求解,实验结果如表3所示。

表3 本文算法采用四种惯性权重策略的求解结果

惯性权重更新策略	f_1	f_2	f_3	运行时间/s
策略1	18	16	105	
	19	16	98	43.6
	21	15	99	
策略2	18	13	91	
	18	13	92	31.3
	19	13	91	
策略3	19	13	95	
	20	13	94	27.5
策略4	18	13	94	
	19	13	93	50.1

从表3中可以看出,采用策略2,本文算法产生的Pareto非支配解均支配采用其他三种算法产生的解;在算法的运行时间方面,采用策略2,算法的运行时间只比采用策略3的算法运行时间略长,比采用策略1和策略4的算法运行时间要短得多。为此,本文采用策略2更新算法的惯性权重。

3.4 实验结果与分析

本节列出了带Pareto非支配解集的教与同伴学习粒子群算法求解 10×10 、 15×10 和 8×8 算例所得的Pareto非支配解集;同时为了验证本文算法的优越性,与Kacem等^[17-18]提出的带局部搜索的控制遗传算法(Controlled Genetic Algorithm with Approach by Localization, AL-CGA)算法和Ho等^[19]提出的带向导性局部搜索的多目标进化算法(Multi-Objective Evolutionary Algorithm with Guided Local Search, MOEA-GLS)

算法的求解结果作了对比。

3.4.1 10×10 算例

本算例是一个小规模全柔性的作业车间调度问题,由10个工件共30道工序在10台机器上加工。算法独立运行30次得到的最优Pareto非支配解集如表4所示。作为比较,其他两种算法的求解结果也列于表4中。

表4 算法对 10×10 算例求解结果

算法	最大完工时间	最大机器负荷	所有机器总负荷	平均运行时间/s
MOEA-GLS	8	7	41	
	7	5	43	16.65
	7	6	42	
AL-CGA	8	7	41	
	8	5	42	—
	7	5	43	
PNDSS-TPLPSO	7	7	41	
	7	6	42	15.21
	8	7	41	
—	8	5	42	

注:“—”表示文献并未给出具体值,以下表示意思相同。

3.4.2 15×10 算例

本算例是一个中规模全柔性的作业车间调度问题,由15个工件共56道工序在10台机器上加工。算法独立运行30次得到的最优Pareto非支配解集如表5所示。作为比较,其他两种算法的求解结果也列于表5中。

表5 算法对 15×10 算例求解结果

算法	最大完工时间	最大机器负荷	所有机器总负荷	平均运行时间/s
MOEA-GLS	23	11	91	
	23	10	93	24.15
AL-CGA	24	11	91	
	23	11	95	—
PNDSS-TPLPSO	12	12	92	
	13	11	93	22.96
	12	10	93	

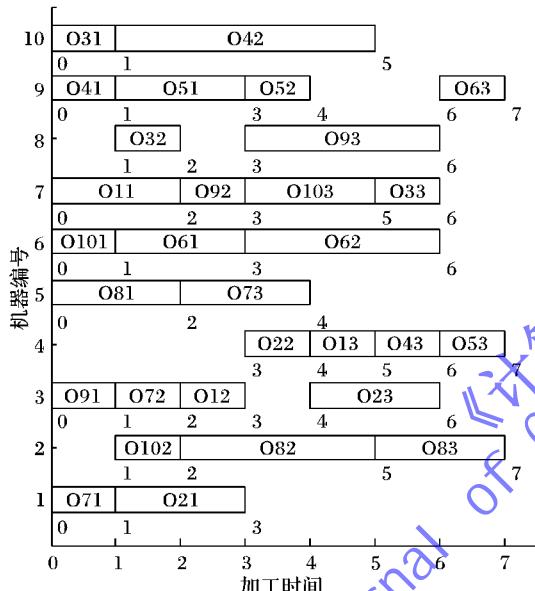
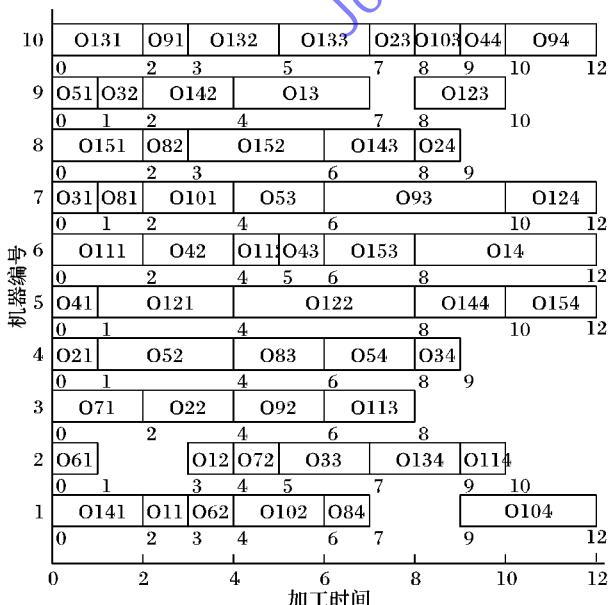
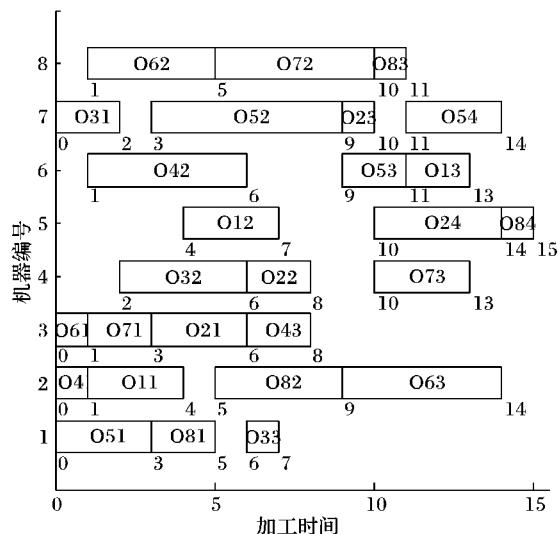
3.4.3 8×8 算例

本算例是一个小规模部分柔性的作业车间调度问题,由8个工件共27道工序在8台机器上加工。算法独立运行30次得到的最优Pareto非支配解集如表6所示。作为比较,其他两种算法的求解结果也列于表6中。

表6 算法对 8×8 算例求解结果

算法	最大完工时间	最大机器负荷	所有机器总负荷	平均运行时间/s
MOEA-GLS	16	13	73	
	15	12	75	9.097
	16	11	77	
AL-CGA	15	13	79	
	16	13	75	—
PNDSS-TPLPSO	15	12	74	
	15	13	73	9.591
	16	12	73	
	16	11	74	

从表 4 可以看出本文算法求解 10×10 算例可以获得更多的 Pareto 非支配解, 其他两种算法所获得的解均是本文算法所求解的一个子集; 在运行时间方面, 本文算法要小于 MOEA-GLS。从表 5 可以看出, 本文算法对 15×10 算例求解所得的解均支配其他两种算法所获得的解, 且本文算法产生的解对应的最大完工时间值要明显小于其他两种算法求解结果; 在运行时间方面, 本文算法要小于 MOEA-GLS。从表 6 可以看出, 本文算法对于部分柔性的 8×8 算例求解得到的所有解均支配其他两种算法求解所得的解, 相比较其他两种算法本文算法能获得更多的 Pareto 非支配解; 在运行时间方面, 本文算法与 MOEA-GLS 相当。综上所述, 本文算法的优化能力要明显好于对比的两种算法, 且本文算法的运行时间较低。表 4~6 中本文算法所求 Pareto 非支配解集中符合决策者要求的最优解对应的甘特图分别如图 2~4 所示, 其中横坐标表示加工时间, 纵坐标表示加工机器号。

图 2 10×10 所求解 2 对应甘特图图 3 15×10 所求解 1 对应甘特图图 4 8×8 所求解 1 对应甘特图

4 结语

柔性作业车间调度是制造业、计算机科学、物流等领域内一个很重要的问题。本文结合教与同伴学习粒子群算法和多目标 Pareto 方法的优点设计了求解多目标 FJSP 的带 Pareto 非支配解集的教与同伴学习粒子群算法。首先建立了柔性车间多目标优化调度模型, 然后采用带 Pareto 非支配解集的教与同伴学习粒子群算法, 以最大完工时间、最大机器负荷和所有机器的总负荷为准则, 对 FJSP 进行多目标优化。实验中采用混合分派规则产生初始种群, 采用快速非支配排序算法初始化 Pareto 非支配解集, 然后对种群内每个粒子执行教阶段, 并对满足同伴学习条件的粒子执行同伴学习阶段, 最后采用提取 Pareto 支配层程序更新 Pareto 非支配解集, 经过多次迭代, 产生一组 Pareto 非支配解, 然后采用决策的思想得到工件调度的最优解。通过不同规模测试实例证明, 本文算法能够有效求解柔性作业车间调度问题。

参考文献:

- [1] NASIRI M M, KIANFAR F. A GES/TS algorithm for the job shop scheduling [J]. Computers and Industrial Engineering, 2012, 62(4): 946–952.
- [2] BRUKER P, SCHLIE R. Job - shop scheduling with multi - purpose machines [J]. Computing, 1990, 45(4): 369–375.
- [3] GU F. Optimization algorithm research for flexible job shop scheduling [D]. Hefei: University of Science and Technology of China, 2006. (谷峰. 柔性作业车间调度中的优化算法研究[D]. 合肥: 中国科学技术大学, 2006.)
- [4] LI X, LU J, CHAI G, et al. Hybrid bee colony algorithm for flexible job shop scheduling problem [J]. Computer Integrated Manufacturing Systems, 2011, 17(7): 1495–1500. (李修琳, 鲁建夏, 柴国钟, 等. 混合蜂群算法求解柔性作业车间调度问题[J]. 计算机集成制造系统, 2011, 17(7): 1495–1500.)
- [5] ZHAO S, FANG S, GU X. Machine selection and FJSP solution based on limit scheduling completion time minimization [J]. Computer Integrated Manufacturing Systems, 2014, 20(4): 854–865. (赵诗奎, 方水良, 顾新建. 基于极限调度完工时间最小化的机器选择及 FJSP 求解[J]. 计算机集成制造系统, 2014, 20(4): 854–865.)

(下转第 1627 页)

- tation Science, 1999, 33(4): 429 – 430.
- [2] KUNKEL A G, ITALLIE E S V, WU D. Optimal distribution of medical backpacks and health surveillance assistants in Malawi [J]. Health Care Management Science, 2014, 17(3): 230 – 244.
- [3] FLESZAR K, HINDI K S. An effective VNS for the capacitated p-median problem [J]. European Journal of Operational Research, 2008, 191(3): 612 – 622.
- [4] XU X, LI X, LI X, et al. Notice of retraction an improved scatter search algorithm for capacitated p-median problem [C]// Proceedings of the 2010 2nd International Conference on Computer Engineering and Technology. Piscataway: IEEE, 2010, 2: V2-316 – V2-320.
- [5] MAZINAN H G, AHMADI G R, KHAJI E. An efficient hybrid CS and k-means algorithm for the capacitated p-median problem [EB/OL]. [2014 – 12 – 08]. <http://arxiv.org/abs/1406.7473>.
- [6] STEFANELLO F, de ARAUJO O C B, MÜLLER F M. Matheuristics for the capacitated p-median problem [J]. International Transactions in Operational Research, 2014, 22(1): 149 – 167.
- [7] MURAT A, VERTER V, LAPORTE G. A continuous analysis framework for the solution of location-allocation problems with dense demand [J]. Computers and Operations Research, 2010, 37(1): 123 – 136.
- [8] MURAT A, VERTER V, LAPORTE G. A multi-dimensional shooting algorithm for the two-facility location-allocation problem with dense demand [J]. Computers and Operations Research, 2011, 38 (2): 450 – 463.
- [9] ZHENG L, LIU Y, JIANG T, et al. A layout approach of city emergency centers with dense demand [J]. Journal of Computer-Aided Design & Computer Graphics, 2014, 26(6): 948 – 955. (郑利平, 刘玉飞, 江婷, 等. 稠密需求下城市应急中心布局方法 [J]. 计算机辅助设计与图形学学报, 2014, 26(6): 948 – 955.)
- [10] AURENHAMMER F, HOFFMANN F, ARONOV B. Minkowski-type theorems and least-squares clustering [J]. Algorithmica, 1998, 20(1): 61 – 76.
- [11] BALZER M and HECK D. Capacity-constrained Voronoi diagrams in finite spaces [C]// Proceedings of the 5th Annual International Symposium on Voronoi Diagrams in Science and Engineering. Piscataway: IEEE, 2008: 44 – 56.
- [12] BALZER M. Capacity-constrained Voronoi diagrams in continuous spaces [C]// ISVD'09: Proceedings of the 2009 Sixth International Symposium on Voronoi Diagrams. Piscataway: IEEE, 2009: 79 – 88.
- [13] BALZER M, DEUSSEN O. Voronoi treemaps [C]// INFOVIS'05: Proceedings of the 2005 IEEE Symposium on Information Visualization. Washington, DC: IEEE Computer Society, 2005: 49 – 56.
- [14] REITSMA R, TRUBIN S, MORTENSEN E. Weight-proportional space partitioning using adaptive Voronoi diagrams [J]. GeoInformatica, 2007, 11(3): 383 – 405.
- [15] BALZER M, SCHLÖMER T, DEUSSEN O. Capacity-constrained point distributions: a variant of Lloyd's method [EB/OL]. [2014 – 12 – 03]. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.177.6047>.
- [16] DU Q, FABER V, GUNZBURGER M. Centroidal Voronoi tessellations: applications and algorithms [J]. SIAM Review, 1999, 41 (4): 637 – 676.
- [17] MACQUEEN J. Some methods for classification and analysis of multivariate observations [EB/OL]. [2014 – 12 – 03]. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.308.8619>.
- [18] CORTES J, MARTINEZ S, KARATAS T, et al. Coverage control for mobile sensing networks [C]// ICRA'02: Proceedings of the 2002 IEEE International Conference on Robotics and Automation. Piscataway: IEEE, 2002: 1327 – 1332.

(上接第 1622 页)

- [6] WANG X. Research on methods for multi-objective flexible job-shop scheduling problem [D]. Wuhan: Huazhong University of Science and Technology, 2011. (王晓娟. 多目标柔性作业车间调度方法研究 [D]. 武汉: 华中科技大学, 2011.)
- [7] RAHMATI M, ZANDIEH M, YAZDANI M, et al. Developing two multi-objective evolutionary algorithms for the multi-objective flexible job shop scheduling problem [J]. The International Journal of Advanced Manufacturing Technology, 2013, 64 (5/6/7/8): 915 – 932.
- [8] WANG W, LI J, WANG J. Double-population genetic algorithm for multi-objective job shop scheduling problem [J]. Computer Integrated Manufacturing Systems, 2011, 17(4): 808 – 815. (王伟玲, 李俊芳, 王晶. 求解多目标作业车间调度问题的双种群遗传算法 [J]. 计算机集成制造系统, 2011, 17(4): 808 – 815.)
- [9] XIA W, WU Z. An effective hybrid optimization approach for multi-objective flexible job-shop scheduling problem [J]. Computers and Industrial Engineering, 2005, 48(2): 409 – 425.
- [10] LI J-Q, PAN Q-K, GAO K-Z. Pareto-based discrete artificial bee colony algorithm for multi-objective flexible job shop scheduling problems [J]. International Journal of Advanced Manufacturing Technology, 2011, 55(9/10/11/12): 1159 – 1169.
- [11] LIM W H, MAT ISA N A. Teaching and peer-learning particle swarm optimization [J]. Applied Soft Computing, 2014, 18: 39 – 58.
- [12] DEB K, PRATAP A, AGARWAL S, et al. A fast and elitist multi-objective genetic algorithm: NSGA-II [J]. IEEE Transactions on Evolutionary Computation, 2002, 6(2): 182 – 197.
- [13] KENNEDY J, EBERHART R. Particle swarm optimization [C]// Proceedings of the 1995 IEEE International Conference on Neural Networks. Piscataway, IEEE, 1995, 4: 1942 – 1948.
- [14] RAO R V, SAVSANI V J, VAKHARIA D P. Teaching-learning based optimization: a novel method for constrained mechanical design optimization problems [J]. Computer-Aided Design, 2011, 43(3): 303 – 315.
- [15] HO N B, TAY J C, LAI E M-K. An effective architecture for learning and evolving flexible job-shop schedules [J]. European Journal of Operational Research, 2007, 179(2): 316 – 333.
- [16] CHEN G, JIA J, HAN Q. Study on the strategy of decreasing inertia weight in particle swarm optimization algorithm [J]. Journal of Xi'an Jiaotong University, 2006, 40(1): 53 – 56, 61. (陈贵敏, 贾建援, 韩琪. 粒子群优化算法的惯性权值递减策略研究 [J]. 西安交通大学学报, 2006, 40(1): 53 – 56, 61.)
- [17] KACEM I, HAMMADI S, BORNE P. Approach by localization and multi-objective evolutionary optimization for flexible job-shop scheduling problems [J]. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, 2002, 32 (1): 1 – 13.
- [18] KACEM I, HAMMADI S, BORNE P. Pareto-optimality approach for flexible job-shop scheduling problems: hybridization of evolutionary algorithms and fuzzy logic [J]. Mathematics and Computers in Simulation, 2002, 60(3/4/5): 245 – 276.
- [19] HO N B, TAY J C. Solving multiple-objective flexible job shop problems by evolution and local search [J]. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, 2008, 38(5): 674 – 685.