

多目标优化算法求解多柔性作业车间调度问题



董 海¹ 徐晓鹏² 谢 谢³

1 沈阳大学应用技术学院 沈阳 110044

2 沈阳大学机械工程学院 沈阳 110044

3 沈阳大学信息工程学院 沈阳 110044

(13898802977@163.com)

摘要 针对车间调度中存在的机器柔性、工人柔性和并行工序柔性,文中用优先级间的加工顺序替代单独工件间的顺序约束来表示并行工序柔性,建立了以最小化最大完成时间、总耗能和平均完成时间为目地的多柔性作业车间调度模型,设计了一种四染色体编码方法及对应的交叉和变异算子,并用两条染色体来编码加工顺序。结合入侵肿瘤生长优化算法的算法结构和NSGAIII 算法中对解的筛选机制,提出一种多目标优化算法求解模型。该算法使用快速非支配排序方法和基于特征点的选择方法对细胞进行分类和转化,设计替代重复细胞的机制,并基于交叉和变异算子,重新设计了细胞的生长和入侵机制。最后,求解数值实例,用超体积、延展度和分布度对比所提算法和其他多种智能算法得到的解集,结果证明所提算法收敛更快且所得解集分布更均匀。

关键词:作业车间调度问题;工人柔性;并行工序柔性;入侵肿瘤生长优化算法;NSGAIII

中图法分类号 TH165;TP301.6

Solving Multi-flexible Job-shop Scheduling by Multi-objective Algorithm

DONG Hai¹, XU Xiao-peng² and XIE Xie³

1 School of Applied Technology, Shenyang University, Shenyang 110044, China

2 School of Mechanical Engineering, Shenyang University, Shenyang 110044, China

3 School of Information Engineering, Shenyang University, Shenyang 110044, China

Abstract In view of machine flexibility, worker flexibility and parallel operation flexibility in the job-shop scheduling, this paper denotes the parallel operation flexibility by replacing sequence constraints between individual operations with sequence constraints between priorities, and proposes a multi-flexible job-shop scheduling model with objectives of minimizing the maximum completion time, total energy consumption and average completion time. A four-chromosome coding method and corresponding crossover and mutation operators are designed, in which two chromosomes are used to encode the processing sequence. A multi-objective optimization algorithm is proposed, based on the combination of the structure of the invasive tumor growth optimization and the screening mechanism of NSGAIII. The algorithm uses a fast non-dominant sorting method and a feature-based selection method to classify and transform cells. A mechanism is designed to replace duplicate cells. Finally, the proposed algorithm is compared with several intelligent algorithms in calculation examples by hypervolume, distribution and extensibility, which prove its effectiveness and feasibility.

Keywords Job-shop scheduling problem, Worker flexibility, Parallel operation flexibility, Invasive tumor growth optimization algorithm, NSGAIII

1 引言

自 Johnson 首次提出流水车间调度问题(Flow-sShop Scheduling Problem, FSP)以来,传统车间调度问题(Job-sShop Scheduling Problem, JSP)被大量研究^[1]。Brucker 等

在经典 JSP 的基础上考虑了机器柔性,首次提出柔性车间调度问题(Flexible Job-sShop Scheduling Problem, FJSP)^[2]。

FJSP 问题包含多个相互冲突的优化目标。针对多目标规划问题,一些学者采用加权函数的方法将多个目标转化为一个目标,其他学者采用基于 Pareto 优化的多目标算法^[3-5],

该类方法可以得到多种优化方案,且兼顾各目标间的均衡。

在 FJSP 的基础上,Gong 等首次同时考虑机器和工人的柔性,建立了双柔性作业车间调度问题(Double Flexible JobsShop Scheduling Problem,DFJSP)的模型^[6]。Cao 等建立了基于员工学习行为的多目标柔性车间调度模型,并使用杂草算法优化了机器选择和员工指派^[7]。

Huang 团队对实际车间生产中的工序顺序柔性做出了全面的研究。该团队将工序柔性概括为 3 类^[8],提出了 5 种描述工序柔性的方法^[9],建立柔性工序块(flexible block)来表示一定的工序对象以及这些工序对象之间的顺序关系,并给出了对应的编码方案^[10]。

本文在 DFJSP 的基础上考虑并行工序柔性和机器间的运输时间,建立多柔性作业车间调度问题(Multi-Flexible Jobshop Scheduling Problem,MFJSP)模型,并在入侵肿瘤生长优化算法的基础上提出多目标入侵肿瘤优化算法(Multi-objective Invasive Tumor Growth Optimization,MOITGO),获得 MFJSP 的处于 Pareto 非支配解集的多种调度方案。

2 多柔性作业车间调度问题模型

2.1 问题描述

在实际柔性车间生产中,同时存在并行机器柔性、工人柔性和工序柔性。本文提出的多柔性作业车间调度问题可概括为:由分属各技能等级的多名工人在多台机器上加工多个工件,每个工件的工序分属不同优先级,需按照优先级约束进行加工,同一优先级的工序无顺序约束。图 1 给出了从 4 台机器和分属 3 个等级的 5 名工人中选择机器和工人,加工一个可由分属 3 个优先级的 5 道工序完成的工件的情况。调度方案需为每一道工序指定一台机器和一名工人,并给出开始加工该道工序的时间。

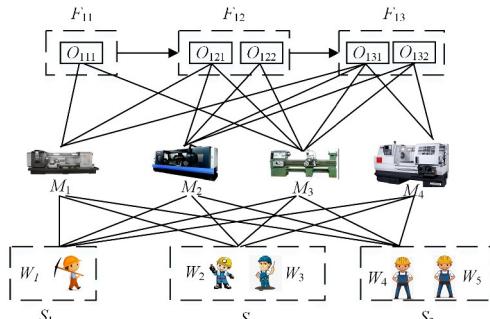


图 1 多柔性作业车间调度问题示例

Fig. 1 Example of MFJSP

2.2 符号定义

模型中所用符号的定义如下:

n 表示工件总数;

m 表示机器总数;

u 表示工人技能等级总数;

i, e 为工件下标, J_i, J_e 分别为需加工的第 i 个和第 e 个工件;

L_i, L_e 分别为工件 J_i, J_e 所包含优先级的数量;

j, f 为优先级下标, $F_{ij}, F_{ij-1}, F_{il_i}$ 为工件 J_i 第 j 层、第 $j-1$ 层(即第 j 层的前一层)和最后一层的优先级, F_{ef} 为工件 J_e 第 f 层的优先级;

$K_{ij}, K_{ij-1}, K_{il_i}, K_{ef}$ 为优先级 $F_{ij}, F_{ij-1}, F_{il_i}, F_{ef}$ 中的工序数量;

k, q, g 为工序下标, O_{ijk}, O_{ijg} 是优先级 F_{ij} 中第 k 道和第 g 道工序, O_{il_ik} 是优先级 F_{il_i} 中第 k 道工序, O_{ij-1q} 是优先级 F_{ij-1} 中的第 q 道工序, O_{efg} 是优先级 F_{ef} 中的第 g 道工序;

p, r 为机器下标, M_p, M_r 为第 p 台和第 r 台机器;

s 为工人技能水平的下标, S_s 为第 s 级技能水平;

N_s 为处于技能水平 S_s 的工人总数;

v 为工人下标, W_v 为技能水平 S_v 的第 v 名工人;

$t_{ijkpsv}, t_{ijgpsv}, t_{efgpsv}$ 为工人 W_v 在机器 M_p 上加工工序 O_{ijk} , O_{ijg} , O_{efg} 的用时;

P_p 为机器 M_p 加工工序时单位时间的耗能;

$transT_{pr}$ 为将工件从机器 M_p 转移到机器 M_r 的耗时;

$sT_{ijk}, sT_{il_i k}, sT_{ijg}$ 和 $eT_{ijk}, eT_{ik}, eT_{ijg}, eT_{efg}$ 为工序 $O_{ijk}, O_{il_i k}, O_{ijg}, O_{efg}$ 的开始加工时间和完成时间;

C_i 为工件 J_i 的完成时间;

C_{\max} 为最大完成时间;

C_{mean} 为平均完成时间;

E 为总耗能。

模型中有两类决策变量,定义如下:

$$x_{ijkp} = \begin{cases} 1, & \text{工序 } O_{ijk} \text{ 在机器 } M_p \text{ 上加工} \\ 0, & \text{否则} \end{cases}$$

$x_{ijgp}, x_{ijgp}, x_{ij-1q}$ 同上文的定义;

$$x_{ijksv} = \begin{cases} 1, & \text{工序 } O_{ijk} \text{ 被工人 } W_v \text{ 加工} \\ 0, & \text{否则} \end{cases}$$

x_{efgs}, x_{ijgs} 同上文的定义。

2.3 模型建立

多柔性作业车间调度问题数学的模型如下:

$$\min C_{\max} = \max_{1 \leq i \leq n} (C_i) = \max_{1 \leq i \leq n} (\max_{1 \leq k \leq K_{il_i}} (C_{il_i k})) \quad (1)$$

$$\min E = \sum_{p=1}^m \sum_{i=1}^n \sum_{j=1}^{L_i} \sum_{k=1}^{K_{ij}} \sum_{s=1}^{N_s} \sum_{v=1}^u x_{ijkp} x_{ijksv} P_p t_{ijkpsv} \quad (2)$$

$$\min C_{\text{mean}} = \frac{\sum_{i=1}^n C_i}{n} = \frac{\sum_{i=1}^n \max_{1 \leq k \leq K_{il_i}} (C_{il_i k})}{n} \quad (3)$$

式(1)~式(3)为优化目标:式(1)表示最小化最大完工时间;式(2)表示最小化总耗能;式(3)表示最小化平均完工时间。该模型有如下约束。

(1) 加工资源约束

$$\sum_{p=1}^m x_{ijkp} = 1, \forall i, j : i=1, 2, \dots, n; j=1, 2, \dots, L_i \quad (4)$$

$$\sum_{s=1}^u \sum_{v=1}^{N_s} x_{ijksv} = 1, \forall i, j : i=1, 2, \dots, n; j=1, 2, \dots, L_i \quad (5)$$

式(4)、式(5)分别确保一道工序只选择一名工人和一台机器。

$$\sum_{p=1}^m (sT_{efg} - eT_{ijk}) x_{efgs} x_{ijksv} \geq 0 \vee \sum_{p=1}^m (sT_{ijk} - eT_{efg}) x_{efgs} \\ x_{ijksv} \geq 0, \forall i, j, k, e, f, g, s, v : i, e=1, 2, \dots, n; s=1, 2, \dots, u; v=1, 2, \dots, N_s; j=1, 2, \dots, L_i; f=1, 2, \dots, L_e; k=1, 2, \dots, K_{ij}; g=1, 2, \dots, K_{ef} \quad (6)$$

$$\sum_{s=1}^u \sum_{v=1}^{N_s} (sT_{efg} - eT_{ijk}) x_{efgp} x_{ijkp} \geq 0 \vee \sum_{s=1}^u \sum_{v=1}^{N_s} (sT_{ijk} - eT_{efg}) x_{efgp} x_{ijkp} \geq 0, \forall i, j, k, e, f, g, p : i, e=1, 2, \dots, n; p=1, 2, \dots, m; j=1, 2, \dots, L_i; f=1, 2, \dots, L_e; k=1, 2, \dots, K_{ij}; p=1, 2, \dots, m; g=1, 2, \dots, K_{ef} \quad (7)$$

$$2, \dots, L_e; k=1, 2, \dots, K_{ij}; g=1, 2, \dots, K_{cf} \quad (7)$$

式(6)、式(7)分别确保一名工人或一台机器同一时间只加工一道工序。

(2) 加工时间约束

$$eT_{ijk} - sT_{ijk} = \sum_{p=0}^m \sum_{s=1}^{N_s} \sum_{v=1}^{N_v} t_{ijkpsv} x_{ijkp} x_{ijksv}, \forall i, j, k, i=1, 2, \dots, n; j=1, 2, \dots, L_i; k=1, 2, \dots, K_{ij} \quad (8)$$

式(8)表示工序开始时间和完成时间的关系。

(3) 工件完整约束与运输时间约束

$$sT_{ief} - eT_{ijk} - \sum_{p=0}^m \sum_{r=0}^m x_{ijkp} x_{iefr} trans T_{pr} \geq 0, \forall i, j, e, k, f; i=1, 2, \dots, n; j=1, 2, \dots, L_i; e=1, 2, \dots, L_i; k=1, 2, \dots, K_{ij}; f=1, 2, \dots, K_{ie} \quad (9)$$

式(9)表示工件在同一时刻只被加工一道工序,同时确保其从一台机器转移到另一台机器所需的运输时间。

(4) 优先级约束

$$sT_{ijk} \geq \max_{1 \leq q \leq K_{ij-1}} (eT_{ij-1q}), \forall i, j; i=1, 2, \dots, n; j=1, 2, \dots, L_i \quad (10)$$

式(10)确保工件中优先级在前的工序先被加工完成。

2.4 编码

对MFJSP可采用四染色体整数编码。一条染色体编码机器,一条染色体编码工人指派,如图2所示,其中工序 O_{122} 被选择和指派了三号机器和三号工人(工人序号是对所有工人的统一编号,例如分属两个技能等级的工人 W_{11}, W_{12} 和 W_{21}, W_{22} 将被重新编号为 W_1, W_2, W_3, W_4)。

	O_{111}	O_{121}	O_{122}	O_{131}	O_{132}	O_{211}	O_{212}	O_{221}	O_{222}
机械选择	1	2	3	1	4	2	1	2	4
工人指派	1	3	3	2	4	3	4	1	2

图2 机器和工人编码

Fig. 2 Coding of machines and workers

两条染色体编码工序排序:一条染色体的基因为工件下标,另一条染色体的基因为打乱的从1到工序总数的整数,如图3所示。图3中,将工序排序1的染色体中相同工件按顺序约束分成各优先级,如图4所示。图4中的各个优先级中都包含多道工序,每个优先级中的工序都按图3中工序排序2的染色体上对应位置的整数大小进行排序,如图5所示。从四染色体编码解码得到的调度方案均符合模型的约束要求。

工序排序1	1	2	2	1	2	1	1	2	1
	O_{111}	O_{121}	O_{122}	O_{131}	O_{132}	O_{211}	O_{212}	O_{221}	O_{222}
工序排序2	7	5	3	2	6	8	1	9	4

图3 工序编码

Fig. 3 Coding of operations

优先级排序	F_{11}	F_{21}	F_{21}	F_{12}	F_{22}	F_{12}	F_{13}	F_{22}	F_{13}
-------	----------	----------	----------	----------	----------	----------	----------	----------	----------

图4 优先级编码

Fig. 4 Coding of priority levels

加工顺序	O_{111}	O_{121}	O_{122}	O_{131}	O_{132}	O_{211}	O_{212}	O_{221}	O_{222}
------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------

图5 工序编码所对应的加工顺序

Fig. 5 Processing sequence decoded

针对问题和编码的特点,本文为启发式算法准备了变异和交叉算子。

(1) 变异算子。对机器选择和工人指派染色体进行一到多个位置的单点变异;对工序排序1的染色体进行位置交换;对工序排序2的染色体等概率采取交换、插入、反转逆序、打乱互换4种变邻域搜索方式^[11]。

(2) 交叉算子。将父代的个体两两配对;遴选基因位置,交换父代个体中机器选择和工人指派染色体上的基因;遴选基因位置,对比父代个体中工序排序1的染色体在这些位置上的工件编号,选取其他位置的基因进行补充,使基因交换后染色体上各工件编号的总数不变;对于工序排序2的染色体,分别等概率采用基于位置的交叉算子(Position-based Crossover, PBX)和线性次序交叉算子(Linear Order Crossover, LOX)^[11]。

3 优化算法

3.1 入侵肿瘤生长优化算法

入侵肿瘤优化算法(Invade Tumor Growth Optimization, ITGO)是Tang等^[12]于2015年提出的新型群体智能优化算法,其在连续优化问题^[12]、数据聚类问题^[13]和云计算任务调度问题^[14]方面较其他智能算法有明显的优势。

ITGO是对肿瘤细胞的模拟,该算法将问题的解类比为生长细胞(Pcell)、入侵细胞(Icell)、休眠细胞(Qcell)、死亡细胞(Dcell)等,将适应度值类比为营养液浓度,细胞将朝着营养液浓度高的方向移动。其中,Icell最为活跃,承担部分搜索任务及全部跳出局部最优解的任务;Pcell承担绝大部分的搜索工作;Qcell缓慢执行搜索策略;Dcell由休眠细胞衰退形成,释放占据的计算资源。各种细胞之间可以相互转化。

3.2 NSGAIII 算法

NSGAII采用快速非支配排序(Fast Non-dominated Sorting, FNDS)将种群中的个体分归到不同非支配层(Non-dominated Rank, NDR),并计算个体间的拥挤距离,然后根据NDR和拥挤距离筛选个体。在面对3个及以上目标的多目标优化问题时,基于拥挤距离的排序方式作用不明显,因此Deb等^[15-16]将其替换为基于参考点的方法(Reference-point-based Method, RPBM),提出NSGAIII。但当处理离散度高的问题时,NSGAIII仍会遇到NSGA系列算法在算法后期最优NDR中解的数目过多而影响进一步筛选的问题。

3.3 多目标入侵肿瘤生长优化算法

本文提出一种多目标入侵肿瘤生长优化算法,其结构如图6所示。MOITGO在运行中会产生重复的细胞,替换重复细胞的方法为:将各细胞的解码结果逐个放入列表中,若某个细胞的解码结果已在列表中,则对其进行变异,直到其解码结果不在列表中。MOITGO对细胞的分类和选取均采用NSGAIII中基于FNDS和RPBM的选择方法,其中RPBM的参考点采用Das and Dennis's method^[17]生成。

MOITGO中“分裂”指复制一份原细胞为其他种类的细胞,“转化”指改变细胞的种类。在生长周期内保持不变的Pcell可能陷入局部最优解,让其分裂出Icell可避免算法陷入该解。细胞的生长方式和入侵方式根据MFJSP的离散特点重新设计,分别如图7和图8所示。

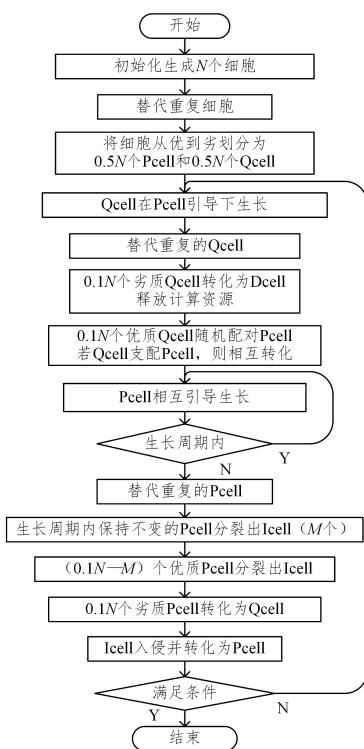


图 6 多目标入侵肿瘤生长优化算法的结果

Fig. 6 Structure of MOITGO

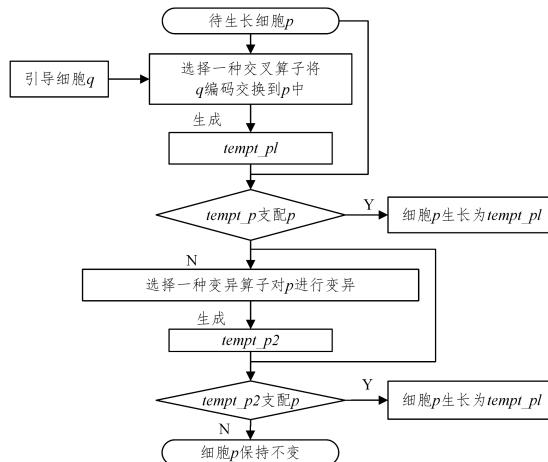


图 7 在另一细胞引导下的细胞生长

Fig. 7 Growth of one cell guided by another cell

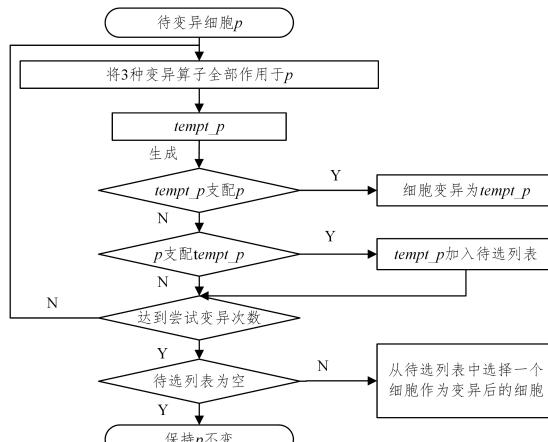


图 8 细胞的入侵

Fig. 8 Invasion of one cell

3.4 算法复杂度分析

设种群规模为 N , 目标函数数量为 M , 则快速非支配排序的算法复杂度为 $O(MN^2)$ ^[15]。设染色体基因数量为 K , 则染色体变异、交叉和解码的算法复杂度均为 $O(K)$ 。设细胞变异过程中变异尝试次数为 n , 则由细胞生长和入侵的流程易得生长的算法复杂度为 $O(K+M)$, 入侵的算法复杂度为 $O(n(K+M))$ 。设 k 为生长细胞的生长周期, 则在该过程中细胞相互之间引导生长的算法复杂度为 $O(kN(K+M))$ 。由上述分析可得, 算法每次迭代的算法复杂度为 $O(MN^2) + O((kN+n)(K+M))$ 。

4 算法验证

本文算法采用 python 3.5.2 编程语言来实现, 运行环境如下: 处理器主频 3.3 GHz, 内存 4 GB, Windows 7 操作系统。

MFJSP 问题首次被提出, 需创建该问题的数值实例, 各数值实例的参数如表 1 所列。

表 1 多柔性作业车间调度问题实例参数表

Table 1 Parameter table of numerical examples of MFJSP

序号	工件数	优先级数	工序数	机器数	工人数
1	4	11	14	4	5
2	4	14	20	4	5
3	4	14	24	4	5
4	5	16	30	4	5
5	5	20	35	4	5
6	7	25	41	6	8
7	7	28	47	6	8
8	9	32	52	6	8
9	9	32	57	6	8
10	9	34	60	6	8

图 9 给出了 MOTIGO 求解数值实例 1 时所得到的 Pareto 优化解集, 可见解集中的解均匀分布在解空间。图 10 给出了图 9 中一个解对应的调度方案。

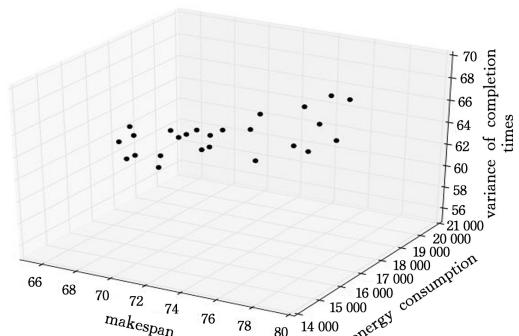


图 9 MOTIGO 所得 Pareto 非支配前沿

Fig. 9 Pareto non-dominated frontier of MOTIGO

图 11 给出了 MOTIGO 和 NSGA 分别求解数值实例 1 时, 最优非支配层中解的个数占总体个数的比例。可以看到 MOTIGO 避免了算法后期最优非支配层中解的数目过多的问题。

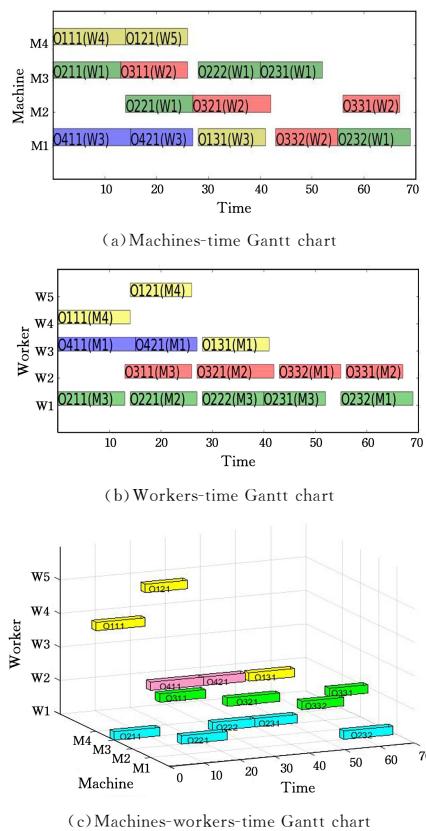


图 10 调度方案甘特图

Fig. 10 Gantt chart of scheduling scheme

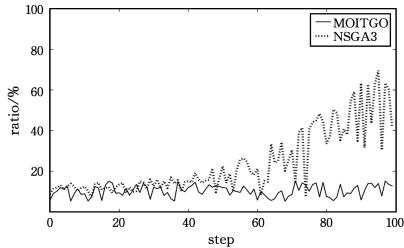


图 11 最优非支配层中解的个数占总体个数的比例

Fig. 11 Proportion of solutions in optimal NDR

本文采用超体积(Hypervolume)^[18]、分布度(Distribution)^[19]以及延展度(Extensibility)^[20]评价算法求得的 Pareto 解集,3 个指标的定义如式(11)~式(13)所示。超体积越大,说明越逼近原优化问题的真实 Pareto 前沿;分布度越小,说明解的分布越均匀;延展度越大,说明解的分布越广。

$$H = \lambda \left(\bigcup_{X \in A} \bigcup_{0 \leq i \leq p} \{h \mid f_i(X) < h < ref_i\} \right) \quad (11)$$

$$D = -\sqrt{\sum_{j=1}^N (d^* - d_j(X))^2 / (N-1)} \quad (12)$$

$$E = \sqrt{\sum_{i=1}^m \left(\max_{j=1}^N f_i(X_j) - \min_{j=1}^N f_i(X_j) \right)^2} \quad (13)$$

式(11)中, λ 表示 Lebesgue 测度, p 为目标函数个数, X 为决策变量的集合, A 为可行解集合, $f_i(X)$ 为决策变量取 X 时的目标函数值, ref_i 为参考点在目标函数 f_i 上的值。式(12)中, $d(X_j)$ 为可行解 X_j 的拥挤距离, 拥挤距离为该解在解空间中相邻两个解之间的欧氏距离; d^* 为拥挤距离的平均值。式(13)中, $\max_{j=1}^N f_i(X_j)$ 和 $\min_{j=1}^N f_i(X_j)$ 分别为解集中所有

解在第 i 个目标函数中所取的最大值和最小值。

为方便对比,在计算 3 项指标前需按式(14)来归一化对比的多个 Pareto 解集中的解对应的目标值,归一化后计算超体积时的参考点设为(1,1,1)。

$$f'_i(x) = \frac{f_i(x) - \min_f_i}{\max_f_i - \min_f_i} \quad (14)$$

其中, $f_i(x), f'_i(x)$ 分别是目标函数 f_i 在任一解 x 处归一化前和归一化后的值; \max_f_i, \min_f_i 分别为所有 Pareto 解集中目标函数 f_i 的最大值和最小值。

分别使用 MOTIGO 与 NSGAIII、NSGAII、基于分解的多目标进化算法^[21](Multi-Objective Evolutionary Algorithm Based on Decomposition, MOEA/D)、离散粒子群算法^[22](Discrete Particle Swarm Optimization, DPSO)、改进强度 Pareto 进化算法^[23](Improved Strength Pareto Evolutionary Algorithm, SPEA2)等对数值实例进行求解,各算法均使用本文所提编码方案。图 12 为不同算法运行 1 000 s 内求解各数值实例所得的 Pareto 解集在 3 个评价指标下的对比结果,其中横坐标为数值实例序号。

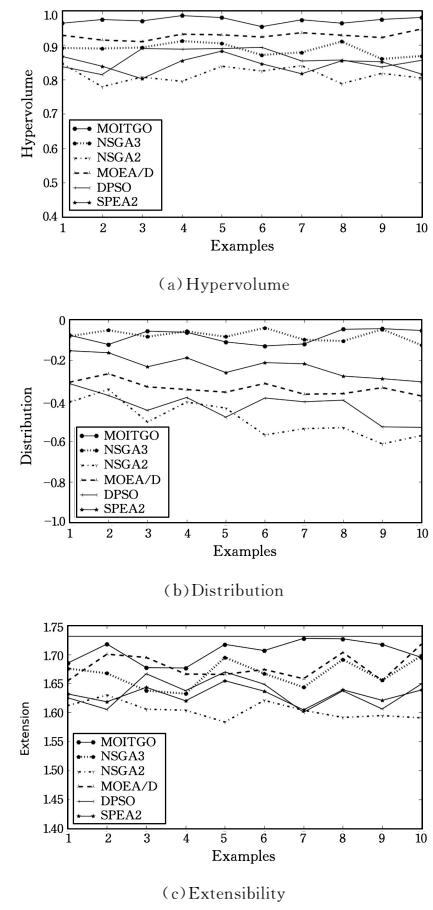


Fig. 12 Comparison of sets of solutions

由图 12 可见, MOTIGO 的超体积大于其他算法,说明 MOTIGO 算法的收敛速度较其他优化算法更快; MOTIGO 与 NSGAIII 的分布度相差不大,均大于其他算法,说明使用 FNDS 和 RPBM 相结合的筛选方法会使所得解集中的解分布得更均匀; 各算法的延展度相差不大,且均接近 $\sqrt{3}$,说明各

算法的解集都分布得比较广。

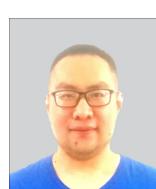
结束语 本文针对同时具有机器、工人和并行工序柔性的作业车间调度问题,建立了多目标优化模型,并给出了对应的编码方法。结合 NSGAIII 中对解集的筛选方法,将入侵肿瘤生长优化算法升级为可解决离散问题的多目标优化的算法。对数值实例的求解,验证了 MOITGO 可以有效地给出包含一组可行调度方案的最优解集供决策者选择,与其他算法相比,MOITGO 具有良好的收敛速度,解集均匀分布且覆盖面广。

参 考 文 献

- [1] JOHNSON S. Optimal Two-and-Three Stage Production Schedules With Setup Times Included [J]. Naval Research Logistics Quarterly, 1954, 1(1): 61-68.
- [2] BRUCKER P, SCHLIE R. Job-Shop Scheduling with Multi-Purpose Machines [J]. Computing, 1991, 45(4): 369-375.
- [3] ZHONGY J, YANG H C, MO R, et al. Optimization Method of Flexible Job-Shop Scheduling Problem Based on Niching and Particle Swarm Optimization Algorithms [J]. Computer Integrated Manufacturing Systems, 2015, 21(12): 3231-3238.
- [4] ZHAO B X, GAO J M, CHEN K. Two-Stage Hybrid Pareto Ant Colony Algorithm for Multi-Objective Flexible Job Shop Scheduling [J]. Journal of Xi'an Jiaotong University, 2016, 50(7): 145-151.
- [5] ZHANG Z Q, WANG K P, ZHU L X, et al. Pareto Hybrid Ant Colony and Genetic Algorithm for Multi-Objective U-Shaped Disassembly Line Balancing Problem [J]. Journal of Xi'an Jiaotong University, 2018, 53(3): 628-637.
- [6] GONG G L, DENG Q W, GONG X R, et al. A New Double Flexible Job-Shop Scheduling Problem Integrating Processing Time, Green Production, and Human Factor Indicators [J]. Journal of Cleaner Production, 2018, 174(10): 560-576.
- [7] CAO L, YE C M, HUANG X. Multi-Objective Flexible Job-Shop Scheduling Based on Learning Effect [J]. Computer Integrated Manufacturing Systems, 2018, 24(8): 2023-2034.
- [8] HUANG X W, MA X L, CAO D M. Improved Generic Algorithm for Job-Shop Scheduling Problem with Process Sequence Flexibility [J]. Operations Research and Management Science, 2013, 22(1): 65-70.
- [9] HUANG X W, ZHAO X W, YU W Q, et al. Description and Measurement of Key Flexibility in Jobshop Manufacturing System [J]. CIMS, 2015, 21(12): 3153-3165.
- [10] HUANG X W, SUN N, SUN R. Operation Sequence Flexibility Description Model [J]. CIMS, 2016, 22(10): 2275-2283.
- [11] CUI Q, WU X L, YU J J. Improved Genetic Algorithm Variable Neighborhood Search for Solving Hybrid Flow Shop Scheduling Problem [J]. CIMS, 2017, 23(9): 1917-1927.
- [12] TANG D, DONG S, JIANG Y, et al. ITGO: Invasive Tumor Growth Optimization Algorithm [J]. Applied Soft Computing, 2015, 36 (Supplement C): 670-698.
- [13] TANG D, DONG S, HE I, et al. Intrusive Tumor Optimization Algorithm for Data Clustering Growth [J]. Neural Computing and Applications, 2016, 27(2): 349-374.
- [14] ZHOU J, DONG S B, TANG D Y. Task Scheduling Algorithm in Cloud Computing Based on Invasive Tumor Growth Optimization [J]. Chinese Journal of Computers, 2018, 41(6): 1140-1155.
- [15] DEB K, PRATAP A, AGARWAL S, et al. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II [J]. IEEE Transactions on Evolutionary Computation, 2002, 6(2): 182-197.
- [16] DEB K, JAIN H. An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems with Box Constraints [J]. IEEE Transactions on Evolutionary Computation, 2013, 18(4): 577-601.
- [17] JAIN H, DEB K. An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point Based Nondominated Sorting Approach, Part II: Handling Constraints And Extending to an Adaptive Approach [J]. IEEE Transactions on Evolutionary Computation, 2013, 18(4): 602-622.
- [18] ZITZLER E. Evolutionary Algorithms Multiobjective Optimization: Methods and Applications [D]. Zurich: Swiss Federal Institute of Technology, 1999.
- [19] SCHOOT J R. Fault Tolerant Design Using Single and Multicriteria Genetic Algorithms Optimization [D]. Cambridge: Massachusetts Institute of Technology, 1995.
- [20] ZITZLER E, DEB K, THIELE L. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results [J]. Evolutionary Computation, 2008, 8(2): 173-195.
- [21] ZHANG Q, LI H. MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition [J]. IEEE Transactions on Evolutionary Computation, 2007, 11(6): 712-731.
- [22] SHAO X, LIU W, LIU Q, et al. Hybrid Discrete Particle Swarm Optimization for Multi-Objective Flexible Job-Shop Scheduling Problem [J]. The International Journal of Advanced Manufacturing Technology, 2013, 67(9-12): 2885-2901.
- [23] ZITHER E, LAUMANNS M, THIELE L. SPEA2: Improving the Strength Pareto Evolutionary Algorithm [C]// Proc. of Int. Conf. on Evolutionary Method for Design, Optimization and Control with Applications to Industrial Problems. Berlin: Springer, 2002: 95-100.



DONG Hai, born in 1971, Ph.D, professor. His main research interests include the modeling, optimization and control of advanced production system, the process optimization of manufacturing enterprise logistics and supply chain management.



XU Xiao-peng, born in 1991, postgraduate. His main research interests include the modeling, optimization and control of advanced production system, application and improvement of intelligent algorithms and the application of machine learning.