

混合多目标算法用于柔性作业车间调度问题

左 益 公茂果 曾久琳 焦李成

(西安电子科技大学智能感知与图像理解教育部重点实验室
智能感知与计算国际联合研究中心 西安 710071)

摘要 在生产调度领域,柔性作业车间调度问题是一个非常重要的优化问题。大多数研究通常优化的目标只是最大完工时间,而在实际中,往往要考虑多个目标。因此,提出了一种新的混合多目标算法用于解决柔性作业车间调度问题,其中考虑了3个目标,分别是,最大完工时间、机器总负载和瓶颈机器负荷。算法设计了有效的编码方式和遗传算子,并采用非支配近邻免疫算法求解非支配最优解。为了提高算法性能,提出了3种不同的局部搜索策略,并将其结合在多目标算法中。在多个数据集上的实验对比结果表明,所提算法优于其它代表性的算法。此外,实验结果还验证了局部搜索技术的有效性。

关键词 柔性作业车间调度问题,多目标,局部搜索,非支配近邻免疫算法

中图分类号 TP18 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2015.9.042

Hybrid Multi-objective Algorithm for Solving Flexible Job Shop Scheduling Problem

ZUO Yi GONG Mao-guo ZENG Jiu-lin JIAO Li-cheng

(Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education, International Research Center for Intelligent Perception and Computation, Xidian University, Xi'an 710071, China)

Abstract The flexible job shop scheduling problem is one of the most important optimization problems in the field of production scheduling, due to its complexity and practical applications in real life. Most studies focus on only one objective—the makespan that is the total time required to complete all jobs. However, a single objective may be insufficient for real applications. Therefore, a new hybrid multi-objective algorithm was proposed for solving the flexible job shop scheduling problem (FJSP) with three objectives, including the makespan, total workload, and critical workload. Effective chromosome representation and genetic operators are introduced. The nondominated neighbor immune algorithm is used to search for Pareto optimal solutions. To improve the search performance, three different local search strategies were designed and combined in the multi-objective algorithm. The computational results on several data sets show that the proposed algorithm outperforms other representative algorithms in general. In addition, the experiments validate the effectiveness of local search strategies.

Keywords Flexible job shop scheduling problem, Multi-objective, Local search, Nondominated neighbor immune algorithm

1 引言

在生产制造系统中,调度问题一直是一个值得关注和投入的关键问题。特别是在复杂的大规模制造系统中,合理而有效的调度才能获得较好的产出,因此,研究调度问题具有重要理论研究价值和实践意义。作为其中最基本的数学模型,作业车间调度问题(Job shop Scheduling Problem, JSP)近年来受到了广泛的关注^[1]。传统的作业车间调度问题中,工件的每个工序只允许在特定的某一个机器上进行加工,而在实际生产中,往往存在多个可以加工这个工序的机器,这就是本文研究的柔性作业车间调度问题(Flexible Job shop Scheduling

Problem, FJSP)。由于增加了一个将工序分配到一台可选机器的分配问题,相比于传统作业车间调度问题,柔性作业车间调度问题变得更为复杂。在文献[2]中,此问题已被证明是NP难问题。

在大多数文献中^[3-9],柔性作业车间调度问题一般研究的是在满足工作约束和机器约束的条件下,如何使得所有工件的完成时间最少,也就是说,问题被建模成一个单目标优化问题。然而,实际生产中常常需要考虑多个不同的目标,例如最大完工时间、机器总负载和瓶颈机器负荷等性能指标,这些指标有时候是相互矛盾的。因而,多目标柔性作业车间调度问题更符合实际需求。近年来,许多研究者提出了不同的算法

到稿日期:2014-05-14 返修日期:2014-07-28 本文受国家自然科学基金(61273317)资助。

左 益(1986—),男,博士生,主要研究方向为进化计算及其应用,E-mail: zuoyi55@gmail.com;公茂果(1979—),男,教授,主要研究方向为计算智能、图像理解、智能优化和数据挖掘;曾久琳(1990—),男,硕士,主要研究方向为机器学习;焦李成(1959—),男,教授,主要研究方向为模式识别、智能信息处理。

用于解决多目标柔性作业车间调度问题。在文献[10]中,这些算法被分为两大类:先验算法和后验算法。

在先验算法中,多个优化目标通过线性组合转化为一个单目标优化问题。这些线性组合的权重需要预先确定,因而称之为先验算法。在早期研究过程中,这种算法比较流行。其中,文献[11]结合了蚁群算法(Particle Swarm Optimization, PSO)和模拟退火算法(Simulated Annealing, SA)来求解多目标柔性作业车间调度问题。文献[12]结合了蚁群算法和禁忌搜索算法(Tabu Search, TS),其中禁忌搜索算法是一种局部搜索技术。文献[13]针对多目标柔性作业车间调度提出了一种有效的搜索策略。这种算法存在多个缺点:1)需要运行多次算法才能得到多个解;2)如何给定权重本身就是个很困难的问题,不同的权重可能得到同一个解。

不同于先验算法,后验算法大多基于非支配关系,能在一次运行过程中获得多个非支配解,因而越来越受研究者的青睐。文献[14]结合了基于非支配关系的进化算法和模糊逻辑技术。文献[15]提出了一种基于免疫和熵原则的多目标遗传算法。在文献[16]中,一种混合蛙跳算法被成功应用于求解多目标柔性作业车间调度问题。2012年,Chiang等人提出了一种混合算法,结合了遗传算法和局部搜索技术[17]。接着,他在文献[18]中提出了一种新的多目标进化算法,并设计了有效的遗传操作和多样性保持策略。文献[10]提出了两种混合多目标算法,分别采用不同的局部搜索策略,一种基于非支配关系,另一种基于最优原则。

本文提出了一种新的混合多目标算法用于求解多目标柔性作业车间调度问题。非支配近邻免疫算法(Nondominated Neighbor Immune Algorithm, NNIA)[19]具有收敛速度快、算法操作简单等特点,被选择作为求解柔性作业车间调度问题的多目标算法。由于问题本身的复杂性,设计了3种不同的局部搜索技术,并结合在NNIA中。在下文中,该算法简称为HNNIA。为了验证算法的有效性,算法在多个不同规模的实例上进行了测试,并与多种代表性的算法进行了比较。实验结果表明,相对于其他算法,HNNIA在求解多目标柔性作用问题上具有更优的性能。此外,本文通过实验证明了局部搜索技术的有效性。

2 相关知识

2.1 多目标柔性作业车间调度问题定义

多目标柔性作业车间调度问题可以表述如下:有 n 个工件 $J = J_1, J_2, \dots, J_n$ 要在 m 台机器 $M = M_1, M_2, \dots, M_m$ 上进行加工,每一个工件 J_i 包含 n_i 道工序 $O_{i,j} (j = 1, 2, \dots, n_i)$,这些工序必须按照一定的顺序进行加工。每道工序 $O_{i,j}$ 能在一个或多个可选的机器 $M_{i,j} \in M$ 上进行加工,在不同机器上的加工时间不一样。表1列出了一个简单的实例,其中符号“-”表示对应的机器不能加工该工序。调度的任务包含两个:确定每道工序的加工机器和每台机器上工序的加工顺序。本文优化的3个目标分别是最大完工时间、机器总负载和瓶颈机器负荷。假定 C_i 代表工件 J_i 的完成时间, W_k 代表在机器 k 上加工的工序时间的总和,这3个目标可以表示为:

$$C_{\max} = \max\{C_i | i = 1, 2, \dots, n\}$$

$$W_T = \sum_{k=1}^m W_k \quad (1)$$

$$W_{\max} = \max\{W_k | k = 1, 2, \dots, m\}$$

表1 一个简单的FJSP实例

		M ₁	M ₂	M ₃
J ₁	O _{1,1}	3	6	—
	O _{1,2}	5	—	2
	O _{1,3}	3	2	—
J ₂	O _{2,1}	—	3	4
	O _{2,2}	6	4	3
	O _{2,3}	2	—	5
J ₃	O _{3,1}	5	3	—
	O _{3,2}	3	—	2

2.2 多目标优化理论

不失一般性,多目标优化问题可表述为:

$$\min F(x) = (f_1(x), f_2(x), \dots, f_m(x))^T \quad (2)$$

其中, $x = [x_1, x_2, \dots, x_d] \in \Omega$ 是 d 维决策变量, Ω 是决策空间。假设 x_a, x_b 是多目标优化问题的两个解,如果满足: $f_i(x_a) \leq f_i(x_b) (i = 1, 2, \dots, m)$ 并且 $F(x_a) \neq F(x_b)$,则称 x_a 支配 x_b ,记为: $x_a > x_b$ 。

对于一个决策变量 $x \in \Omega$,如果不存在 $x^* \in \Omega$ 使得 $x^* > x$,则称解 x 为Pareto最优解。

所有Pareto最优解的集合称为Pareto最优解集PS,定义如下:

$$PS = \{x \in \Omega | \neg \exists x^* \in \Omega, x^* > x\} \quad (3)$$

Pareto最优解集中所有解对应的目标矢量构成的空间形状称为Pareto最优前沿端PF。

2.3 NNIA简介

NNIA模拟免疫响应,选择少数非支配作为活性抗体,根据活性抗体拥挤程度进行比例克隆,然后对克隆后抗体群进行重组和变异[19]。令 B_t, D_t, A_t, C_t 分别表示 t 时刻的克隆抗体种群、优势抗体总群、活性抗体种群和克隆抗体种群,NNIA的流程描述如下。

算法1 非支配近邻免疫算法

输入: G_{\max} (最大迭代次数), n_D (优势种群规模), n_A (活性种群规模), n_C (克隆种群规模)

输出: $D_{G_{\max}}$ (最终Pareto最优解)

Step 1 初始化:产生大小为 n_D 的抗体种群 B_0 ,并设置 $t=0$ 。

Step 2 更新优势种群:在 B_t 中识别出非支配抗体,构成临时优势抗体种群 DT_{t+1} ,若 DT_{t+1} 规模不大于 n_D ,则令 $D_{t+1} = DT_{t+1}$;否则,计算 DT_{t+1} 中所有个体拥挤距离,选择较大的 n_D 个体组成 D_t 。

Step 3 终止判断:如果 $t > G_{\max}$,输出结果,算法结束。否则, $t = t + 1$ 。

Step 4 非支配近邻选择:如果 D_t 规模不大于 n_A ,令 $D_t = A_t$;否则,计算 D_t 中所有个体拥挤距离,选择较大的 n_A 个体组成 A_t 。

Step 5 比例克隆:对 A_t 按比例克隆得到 C_t 。

Step 6 重组和变异:对 C_t 重组和变异,得到 C_t' 。

Step 7 合并 C_t' 和 D_t ,组成抗体种群 B_t ,转Step2。

3 HNNIA设计

针对多目标柔性作业车间调度问题,提出一种新的混合多目标优化算法,该算法结合了一种有效的多目标算法NNIA和多种局部搜索策略。下文详细介绍算法的具体操作,包括编码和解码、交叉变异和局部搜索策略等。

3.1 编码和解码

柔性作业车间调度问题实际上包含两个子问题:机器分配问题和确定加工顺序问题。针对这一特性,本文采用了一种扩展的基于工序的编码^[20],该编码由两部分组成:基于机器的编码和基于工序的编码。机器的编码根据工件工序的顺序排列,编码的每一位对应着相应工序可选机器的索引。图1展示了表1给出的实例的一个可行机器编码。在基于工序的编码中,一个工件的所有工序用同一符号表示,工件的加工顺序由工件在编码中出现的顺序决定。一个可行的工序编码 $[1\ 2\ 3\ 2\ 1\ 2\ 3\ 1]$ 对应的工序顺序为 $O_{1,1} \rightarrow O_{2,1} \rightarrow O_{3,1} \rightarrow O_{2,2} \rightarrow O_{1,2} \rightarrow O_{2,3} \rightarrow O_{3,2} \rightarrow O_{1,3}$ 。这种编码方式能保证所产生的调度都是可行调度,减少了搜索空间,也给后续的交叉和变异操作带来了便利。为了进一步提高搜索效率,本文采用文献[6]中提出的解码方法得到的解都是主动调度。该方法的步骤如下:首先按照机器编码获得每个工序所分配的机器;然后对每个机器,根据工序编码中工序的顺序,依次将工序放置在尽可能使得完成时间短的位置。

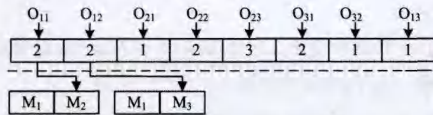


图1 机器编码的一个例子

3.2 交叉和变异

针对编码的两个部分,分别设计了不同的交叉和变异操作。对于机器编码,交叉操作随机选择多个工序,然后交换两个父代对应的这些工序。对于加工顺序编码,本文采用基于工件顺序的交叉^[20]。具体的步骤如下:随机选择一个工件并保持这个工件在两个父代个体的编码中的位置不变;对于其它的工序,按照工序在父代中出现的顺序,互换两个父代个体。

变异操作是对单个个体而言的。针对工序编码部分,变异操作随机选择某两个不属于同一工件的工序进行交换。针对机器编码,变异操作首先随机选择一个工序,然后分配给其它的某一个可选的机器。

3.3 局部搜索策略

柔性车间作业问题还可以用析取图(Disjunctive Graph)模型进行描述^[21]。一个析取图可以表示为 $G=(N,A,E)$ 。其中, N 是代表工序的节点的集合,节点 S 和 T 分别表示起始工序和终止工序; A 是连接同一工件的相邻工序的连接弧集,该连接弧称为有向弧; E 是同一机器上相邻工序的连接弧集,该连接弧称为析取弧。一个可行调度所表示的析取图一定是个无环图。在析取图中,从起始工序到终止工序的最长路径被称为关键路径,它决定了所有工件的完成时间。关键路径上任何一个工序的延迟将会导致所有工件完成时间的增加,这样的工序被称为关键工序。图2给出了对应表1问题的一个可行解的析取图及其相应的甘特图。显然,关键路径为 $O_{1,1} \rightarrow O_{1,2} \rightarrow O_{3,2} \rightarrow O_{2,2} \rightarrow O_{2,3}$,在图中加粗显示。

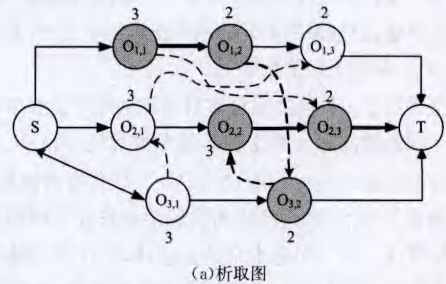
只有通过改变关键工序的位置才有可能减少最大完成时间。因此,本文提出的搜索策略都是在关键工序上进行操作的。针对多目标柔性作业车间调度问题,本文设计了3种不同的局部搜索策略:基于最大完成时间的局部搜索、基于机器

总负载的局部搜索和基于瓶颈机器负荷的局部搜索,它们拥有不同的搜索偏好。局部搜索的基本步骤是从当前解中选定某一个关键工序并对它进行重新分配,以得到更优的解。3种策略具体的搜索过程描述如下。

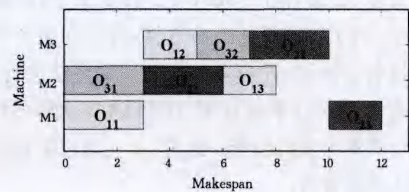
(1)基于最大完成时间的局部搜索:首先,找出当前解的一条关键路径。从当前解中删除关键路径中的某一个关键工序,并搜索其它可能分配的位置。该分配位置必须满足新产生的解在最大完成时间的指标上要优于当前解。为了减少搜索空间,本文采用了文献[4]提出的 $R1$ 邻域条件,满足该条件的解是可行的并且优于初始解。如果找不到合适的位置分配该工序,则换下一个关键工序。如果遍历完整个关键路径都找不到更优的解,说明当前解在这个目标上是局部最优解。

(2)基于机器总负载的局部搜索:选出某一个关键工序,然后搜索其可分配的机器,使得总负载变小。接下来是把该工序放置在所选机器的合理位置上,使得新产生的解可行且最大完成时间尽可能的小。这里,参考文献[5]中的方法,计算出所有可能放置的位置并估算其产生的新的调度的最大完成时间,选取其中最大完成时间最小的调度作为最后结果。此外,该估算方法在文献[5]中被证明是有效的。

(3)基于瓶颈机器负荷的局部搜索:搜索过程与基于机器总负载的局部搜索类似,不同的是根据瓶颈机器负荷搜索关键工序所分配的机器。



(a)析取图



(b)甘特图

图2 一个可行解的析取图和甘特图

3.4 算法框架

本文提出的混合算法在原始 NNIA 的基础上,结合了多种局部搜索技术以提高搜索性能。具体的算法流程如下:

输入:终止条件和参数

输出:最终 Pareto 解

Step1 初始化抗体种群 B_0 , 并设置 $t=0$ 。

Step2 终止条件判断,若满足条件,输出最终 Pareto 解,算法结束;否则, $t=t+1$ 。

Step3 更新优势种群 D_t 。

Step4 对 D_t 执行局部搜索生成 L_t , 将 L_t 合并入 D_t , 并更新 D_t 。

Step5 非支配近邻选择活性抗体种群 A_t 。

Step6 对 A_t 按比例克隆得到 C_t 。

Step7 对 C_t 重组和变异,得到 C_t' 。

Step8 合并 C_t' 和 D_t , 组成抗体种群 B_t , 转 Step2。

从算法流程看出,局部搜索技术作为重要的步骤结合在非支配近邻免疫多目标算法中。为了提高搜索效率和减少计算复杂度,每次只对非支配解集 D_i 的部分个体执行局部搜索。局部搜索的概率和迭代次数分别由参数 P_l 和 g_b 决定。对选定的非支配解,算法随机选出一种搜索技术执行局部搜索。

4 实验结果及分析

4.1 实验设置

为了检验算法性能,选取了 13 个常用的基准测试实例,其中,3 个测试实例来自 Kacem 数据集^[14],剩下的 10 个测试实例来自 BRdata 数据集^[3]。本文算法采用 VC++ 编程,在英特尔双核主频 1.86GHz 的处理器、内存为 2GB 的个人电脑上运行。

表 2 算法参数设置

参数	Kacem	BRdata
优势种群规模(n_D)	20	200
活性种群规模(n_A)	20	50
克隆种群规模(n_C)	50	300
交叉概率(P_c)	1.0	1.0
变异概率(P_m)	0.1	0.1
局部搜索迭代次数(g_b)	20	50
局部搜索概率(P_l)	0.1	0.2

算法参数是通过大量测试实验选定,以保证在算法效果和计算速度之间获得较好的平衡,具体的参数值如表 2 所列。

表 3 算法 AL+CGA、PSO+SA、PSO+TS、ESM、MOGA 和 HNNIA 在 Kacem 数据集上的实验结果比较

算法	8×8			10×10			15×10		
	C_{max}	W_T	W_{max}	C_{max}	W_T	W_{max}	C_{max}	W_T	W_{max}
AL+CGA	15	79	13	7	45	5	23	95	11
							24	91	11
PSO+SA	15	75	12	7	44	6	12	91	11
	16	73	13						
PSO+TS	14	77	12	7	43	6	11	93	11
	15	75	12						
ESM	15	76	12	7	42	6	11	91	11
	14	77	12	8	42	5	11	93	10
MOGA	15	81	11	7	45	5	12	95	10
	15	75	12	7	42	6	11	98	10
	16	73	13	8	42	5	11	91	11
				8	41	7			
HNNIA	14	77	12	7	43	5	11	91	11
	15	75	12	7	42	6	11	93	10
	16	77	11	8	42	5			
	16	73	13	8	41	7			

为了评价多目标算法所产生解的质量,本文采用了两种评价标准:超体积(hypervolume)^[22]和覆盖率(coverage metric)。超体积可以用来评价算法所产生的非支配解的收敛性和多样性,超体积的值越大说明非支配解集的结果越好。而覆盖率通过比较两个算法得到的解的支配关系,来评价两组解之间的优劣。如果 A、B 为两个算法得到的非支配解集,覆盖率 $C(A, B)$ 可用如下公式进行计算:

$$C(A, B) = \frac{|\{b \in B\}, \exists a \in A: a > b \text{ or } a = b\}}{|B|} \quad (4)$$

其中, $|B|$ 表示非支配解集 B 中解的个数。 $C(A, B)$ 的值在区间 $[0, 1]$ 内,值越大说明解集 A 的质量比解集 B 越高。

表 4 列出了算法在 BRdata 数据集上所得到的最终解集的超体积和覆盖率的结果,其中超体积是归一化后的结果。

算法的停止条件设为最大的函数评价次数。参考文献^[10],对 Kacem 和 BRdata 两个数据集的最大的函数评价次数分别设置为 10 万次和 50 万次。

4.2 实验结果

首先,算法在较为简单的 Kacem 数据集上进行测试,对比算法包括 AL+CGA^[14]、PSO+SA^[11]、PSO+TS^[12]、ESM^[13]和 MOGA^[15]。表 3 列出了这些算法所产生的最终非支配解集,其中最优解加粗表示。显然,在这 6 种算法中,本文提出的算法 HNNIA 处理这 3 个实例的效果最好。与最早应用于该问题的 AL+CGA 相比,HNNIA 在所有 3 个实例上都有明显改进。PSO+SA、PSO+TS 和 ESM 这 3 种算法都属于先验算法,实验结果表明,基于 Pareto 支配关系的多目标算法能够产生更多、更有效的非支配解,从而证明了该类算法的合理性和有效性。与另一个基于支配关系的算法 MOGA 相比,HNNIA 在两个实例上获得了较优的效果,在 ka08 实例上获得了相当的效果。

接着,算法在较为复杂的 BRdata 数据集上进行了测试。对比算法选取了 6 种最新提出来的用于求解多目标柔性作业车间调度问题的算法,它们是 MOGA^[15]、HSFLA^[16]、CMA^[17]、SEA^[18]、MA-1^[10]和 MA-2^[10]。其中 MA-1 和 MA-2 是文献^[10]提出的两种不同的混合算法。对比算法的结果是从相应的文献或者作者的主页中直接获得的。

相比于 MOGA 和 HSFLA,无论根据超体积还是覆盖率指标,本文提出的算法 HNNIA 在所有 10 个测试实例中都取得了较优的结果。相比于 CMA,HNNIA 在 6 个测试实例中得到较优的结果,其它的 4 个实例得到相似的结果。相比于 SEA,根据超体积指标,HNNIA 在 5 个测试问题上更优,在 4 个测试问题上略差。然而,根据覆盖率指标,在实例 MK06、MK09 和 MK10 这 3 个测试实例上,HNNIA 要明显优于 SEA;而在 MK03、MK07 和 MK08 这 3 个效果较差的测试实例上,HNNIA 与 SEA 只有微弱的差距。除了 MK07 实例,相对于 MA-1,HNNIA 获得了较优或者相似的结果。与 MA-2 相比,HNNIA 在 MK06 和 MK07 的效果较差,在其他的测试实例上获得了较优或者相似的结果。从整体来说,本文提出的算法获得了较优的结果,其有效性得到了验证。

表4 算法 MOGA、HSFLA、SEA、CMA、MA-1、MA-2 和 HNNIA 在 BRdata 数据集上所获得的最好结果比较

实例	MOGA	HSFLA	SEA	CMA	MA-1	MA-2	HNNIA
MK01	Hypervolume	0.8734	0.5796	0.9959	1.0000	1.0000	1.0000
	C(HNNIA, -)	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
	C(-, HNNIA)	0.1000	0.1000	0.8000	1.0000	1.0000	1.0000
MK02	Hypervolume	0.9293	0.7828	0.9877	0.9898	1.0000	1.0000
	C(HNNIA, -)	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
	C(-, HNNIA)	0.4286	0.0000	0.8571	0.8571	1.0000	1.0000
MK03	Hypervolume	0.7701	0.6667	1.0000	0.8255	0.8255	0.9456
	C(HNNIA, -)	0.5000	1.0000	0.8333	1.0000	1.0000	1.0000
	C(-, HNNIA)	0.1905	0.0476	1.0000	0.8095	0.8095	0.8095
MK04	Hypervolume	0.8632	0.7721	1.0000	0.8850	0.8881	0.9146
	C(HNNIA, -)	0.7000	1.0000	0.8214	0.9565	0.9615	0.9333
	C(-, HNNIA)	0.2069	0.0000	0.7931	0.7931	0.8276	0.7241
MK05	Hypervolume	0.9460	0.9959	1.0000	1.0000	1.0000	1.0000
	C(HNNIA, -)	1.0000	0.9286	1.0000	1.0000	1.0000	1.0000
	C(-, HNNIA)	0.4545	0.7273	1.0000	1.0000	1.0000	1.0000
MK06	Hypervolume	0.7849	0.7713	0.9452	0.9592	0.9641	0.9793
	C(HNNIA, -)	0.7778	1.0000	0.9455	0.8583	0.9065	0.4532
	C(-, HNNIA)	0.0095	0.0000	0.1810	0.2381	0.3333	0.7766
MK07	Hypervolume	0.9626	0.8889	1.0000	0.9770	0.9956	0.9949
	C(HNNIA, -)	0.8571	1.0000	0.8667	1.0000	0.9333	0.9333
	C(-, HNNIA)	0.4000	0.2667	0.9333	0.8667	1.0000	1.0000
MK08	Hypervolume	0.6644	0.8203	1.0000	0.9513	0.9513	0.9513
	C(HNNIA, -)	0.6000	1.0000	0.8000	1.0000	1.0000	1.0000
	C(-, HNNIA)	0.4444	0.3333	1.0000	1.0000	1.0000	1.0000
MK09	Hypervolume	0.9545	0.9677	0.9833	1.0000	1.0000	1.0000
	C(HNNIA, -)	0.6667	1.0000	0.9565	0.9649	0.9677	0.9538
	C(-, HNNIA)	0.0000	0.0000	0.2373	0.7966	0.7119	0.6949
MK10	Hypervolume	0.7004	0.8331	0.8630	0.9471	0.9691	0.9857
	C(HNNIA, -)	1.0000	0.8000	0.9085	0.8993	0.8295	0.6577
	C(-, HNNIA)	0.0000	0.0190	0.0474	0.0853	0.2038	0.4502

此外,为了验证局部搜索对算法性能的影响,在 BRdata 数据集上测试了 NNIA 和 HNNIA 的效果。算法对每个实例独立运行 30 次,所获得的平均结果如表 5 所列。显然,对所有的 10 个测试实例,局部搜索技术的使用都提高了解的质量,揭示了混合算法用于复杂的组合优化问题的合理性。

表5 算法 NNIA 和 HNNIA 的实验对比结果

实例	超体积		覆盖率	
	NNIA(B)	HNNIA(A)	C(A,B)	C(B,A)
MK01	0.6874	0.9835	1.0000	0.3000
MK02	0.7526	0.9830	1.0000	0.2857
MK03	0.8213	0.9956	1.0000	0.2381
MK04	0.8367	0.9882	1.0000	0.5172
MK05	0.9035	0.9912	0.9167	0.3636
MK06	0.8524	0.9819	0.9524	0.1429
MK07	0.9423	0.9973	1.0000	0.7333
MK08	0.7258	0.9905	1.0000	0.5556
MK09	0.9215	0.9864	0.9492	0.1864
MK10	0.7925	0.9769	0.9431	0.0474

结束语 在本文中,柔性车间调度问题被建模成一个多目标优化问题,考虑的 3 个目标分别是:最大完工时间、机器总负载和瓶颈机器负荷。为了求解该问题,提出了一种新的混合多目标算法。在非支配近邻免疫算法的基础上,提出了 3 种有效的局部搜索策略,并结合在算法中。对比实验表明,该算法能够获得较优的性能。下一步工作是设计自适应的局部搜索策略,以及使用本文提出的算法解决大规模作业车间调度问题。

参考文献

[1] 刘莹,谷文祥,李向涛. 置换流水线车间调度问题的研究[J]. 计算机科学,2013,40(11):1-7
Liu Y, Gu W X, Li X T. Research on permutation flow-shop

scheduling problem [J]. Computer Science, 2013, 40(11): 1-7
[2] Garey M R, Johnson D S, Sethi R. The complexity of flowshop and job-shop scheduling [J]. Mathematics of Operations Research, 1976, 93(1): 117-129
[3] Brandimarte P. Routing and scheduling in a flexible job shop by tabu search [J]. Annals of Operations Research, 1993, 41(3): 157-183
[4] Hurink E, Jurisch B, Thole M. Tabu search for the job shop scheduling problem with multi-purpose machine [J]. Operations Research Spectrum, 1994, 15(4): 205-215
[5] Mastroiilli M, Gambardella L M. Effective neighborhood functions for the flexible job shop problem [J]. Journal of Scheduling, 2000, 3(1): 3-20
[6] Gao J, Sun L, Gen M. A hybrid genetic and variable neighborhood descent algorithm for flexible job shop scheduling problems [J]. Computers & Operation Research, 2008, 35(9): 2892-2907
[7] Pezzella F, Morganti G, Ciaschetti G. A genetic algorithm for the flexible job-shop scheduling problem [J]. Computers & Operation Research, 2008, 35(10): 3202-3212
[8] Hmida B A, Haouari M, Huguet M J, et al. Discrepancy search for the flexible job shop scheduling problem [J]. Computers & Operation Research, 2010, 37(12): 2192-2201
[9] Yuan Y, Xu H, Yang J. A hybrid harmony search algorithm for the flexible job shop scheduling problem [J]. Applied Soft Computing, 2013, 13(7): 3259-3272
[10] Yuan Y, Xu H. Multiobjective Flexible Job Shop Scheduling Using Memetic Algorithms [J]. IEEE Transactions on Automation Science and Engineering, 2013, 12(1): 336-353
[11] Xia W, Wu Z. An effective hybrid optimization approach for multi-objective flexible job-shop scheduling problems [J]. Com-

puters & Industrial Engineering, 2005, 48(2):409-425

- [12] Zhang G, Shao X, Li P, et al. An effective hybrid particle swarm optimization algorithm for multi-objective flexible job-shop scheduling problem [J]. Computers & Industrial Engineering, 2009, 56(4):1309-1318
- [13] Xing L, Chen Y, Yang K. An efficient search method for multi-objective flexible job shop scheduling problems [J]. Journal of Intelligent Manufacturing, 2009, 20(3):283-293
- [14] Kacem I, Hammadi S, Borne P. Pareto-optimality approach for flexible job-shop scheduling problems [J]. Hybridization of evolutionary algorithms and fuzzy logic Mathematics and Computers in Simulation, 2002, 60(3):245-276
- [15] Wang X, Gao L, Zhang C, et al. A multi-objective genetic algorithm based on immune and entropy principle for flexible job-shop scheduling problem [J]. The International Journal of Advanced Manufacturing Technology, 2010, 51(5):757-767
- [16] Li J, Pan Q, Xie S. An effective shuffled frog-leaping algorithm for multi-objective flexible job shop scheduling problems [J]. Applied Mathematics and Computation, 2012, 218(18):9353-9371

- [17] Chiang T, Lin H. Flexible job shop scheduling using a multiobjective memetic algorithm [M]// Advanced Intelligent Computing Theories and Applications. With Aspects of Artificial Intelligence. Springer Berlin Heidelberg, 2012:49-56
- [18] Chiang T, Lin H. A simple and effective evolutionary algorithm for multiobjective flexible job shop scheduling [J]. International Journal of Production Economics, 2013, 141(1):87-98
- [19] Gong M G, Jiao L C, Du H, et al. Multi-objective immune algorithm with nondominated neighbor-based selection [J]. Evolutionary Computation, 2008, 16(2):225-255
- [20] Cheng R. A tutorial survey of job-shop scheduling problems using genetic algorithms, part I: Representation [J]. Computers & Industrial Engineering, 1996, 30(4):983-997
- [21] Balas E. Machine scheduling via disjunctive graphs: an implicit enumeration algorithm [J]. Operations Research, 1969, 17(6):941-957
- [22] Zitzler E, Thiele E, Laumanns M, et al. Performance assessment of multiobjective optimizers: An analysis and review [J]. IEEE Transactions on Evolutionary Computation, 2003, 7(2):117-132

(上接第 207 页)

- [3] 吴正洋, 汤庸, 洪少文. 社交网络中虚拟团队的语义描述及本体构建[J]. 计算机应用与软件, 2014(11):6-10
Wu Zheng-yang, Tang Yong, Hong Shao-wen. Semantic Description and Ontology Construction for Virtual Teams in Social Networking[J]. Computer Applications and Software, 2014(11):6-10
- [4] 于洪, 李俊华. 结合社交与标签信息的协同过滤推荐算法[J]. 小型微型计算机系统, 2013, 34(11):2467-2471
Yu Hong, Li Jun-hua. Collaborative Filtering Recommendation Algorithm Using Social and Tag Information[J]. Journal of Chinese Computer Systems, 2013, 34(11):2467-2471
- [5] 陈孝文. 基于社交网络的协同过滤推荐算法研究[D]. 广州: 华南理工大学, 2013
Chen Xiao-wen. Research of Collaborative Filtering Recommendation Algorithm Base On Social Network [D]. Guangzhou: South China University of Technology, 2013
- [6] 贺超波, 汤庸, 傅成州. 融合社交网络信息的协同过滤方法[J]. 暨南大学学报(自然科学版), 2013, 34(3):243-252
He Chao-bo, Tang Yong, Fu Cheng-zhou. Collaborative filtering method with social network information[J]. Journal of Jinan University(Natural Science), 2013, 34(3):243-252
- [7] Li Y, Bandar Z A, McLean D. An Approach for Measuring Semantic Similarity between Words Using Multiple Information Sources[J]. IEEE Transactions on Knowledge and Data Engineering, 2003, 15(4):871-882
- [8] Budanitsky A, Hirst G. Evaluating wordnet-based measures of lexical semantic relatedness [J]. Computational Linguistics, 2006, 32(1):13-47
- [9] Chen Jun-peng, Liu Juan. Combining ConceptNet and WordNet for Word Sense Disambiguation[C]//International Joint Conference on Natural Language Processing, 2011:686-694
- [10] Yu Weng, Qiu Li-rong. TECS: A Web Text Extraction Tool Based on Semantic Similarity Calculation[C]//2011 4th International Conference on Intelligent Networks and Intelligent Systems(ICINIS), 2011:25-228
- [11] Rada R, Mili H, Bicknell E, et al. Development and Application

- of a Metric on Semantic Nets[J]. IEEE Transactions on Systems, Man, and Cybernetics, 1989, 19(1):17-30
- [12] Harispe S, Ranwez S, Janaqi S, et al. The Semantic Measures Library: Assessing Semantic Similarity from Knowledge Representation Analysis[M]// Natural Language Processing and Information Systems. Springer, 2014:254-257
- [13] Mohammad S, Hirst G. Distributional Measures of Semantic Distance: A Survey[J]. Eprint Arxiv:12031858v1
- [14] wikipedia Distributional_semantics [OL]http://en.wikipedia.org/wiki/Distributional_semantics
- [15] Richardson R F, Smeaton A, Murphy J. Using WordNet as a Knowledge Base for Measuring Semantic Similarity Between Words[C]//Proceedings of AICS Conference, 1994
- [16] 刘宏哲, 须德. 基于本体的语义相似度和相关度计算研究综述[J]. 计算机科学, 2012, 39(2):8-13
Liu Hong-zhe, Xu De. Ontology Based Semantic Similarity and Relatedness Measures Review[J]. Computer Science, 2012, 39(2):8-13
- [17] Batet M, Sánchez D, Valls A, et al. Semantic similarity estimation from multiple ontologies[J]. Applied Intelligence, 2013, 38(1):29-44
- [18] 张忠平, 田淑霞, 刘洪强. 一种综合的本体相似度计算方法[J]. 计算机科学, 2008, 35(12):142-145
Zhang Zhong-ping, Tian Shu-xia, Liu Hong-qiang. Composite Approach for Ontology Similarity Computation [J]. Computer Science, 2008, 35(12):142-145
- [19] 方耀宁, 郭云飞, 扈红超, 等. 一种基于 Sigmoid 函数的改进协同过滤推荐算法[J]. 计算机应用研究, 2013, 30(6):1688-1691
Fang Yao-ning, Guo Yun-fei, Hu Hong-chao, et al. Improved collaborative filtering recommender algorithm based on Sigmoid function[J]. Application Research of Computers, 2013, 30(6):1688-1691
- [20] 文俊浩, 舒珊. 一种改进相似性度量的协同过滤推荐算法[J]. 计算机科学, 2014, 41(5):68-71
Wen Jun-hao, Shu Shan. Improved Collaborative Filtering Recommendation Algorithm of Similarity Measure [J]. Computer Science, 2014, 41(5):68-71