



文章编号:1001-9081(2017)02-0517-06

DOI:10.11772/j.issn.1001-9081.2017.02.0517

## 求解作业车间调度问题的混合帝国主义竞争算法

杨小东<sup>1</sup>, 康 雁<sup>1,2\*</sup>, 柳 青<sup>1,2</sup>, 孙金文<sup>1</sup>

(1. 云南大学 软件学院, 昆明 650091; 2. 云南省软件工程重点实验室(云南大学), 昆明 650091)

(\*通信作者电子邮箱 [kangyan@ynu.edu.cn](mailto:kangyan@ynu.edu.cn))

**摘要:**针对最小化最大完工时间的作业车间调度问题(JSP),提出一种结合帝国主义竞争算法(ICA)和禁忌搜索(TS)算法的混合算法。混合算法以帝国主义竞争算法为基础,在同化操作中融入遗传算法中的杂交算子和变异算子,使算法全局搜索能力更强。为了克服帝国主义竞争算法局部搜索能力弱的缺点,引入禁忌搜索算法进一步优化同化操作后的后代。禁忌搜索算法采用混合邻域结构和新型选择策略,使得算法能够更有效地搜索邻域解。混合算法兼具全局搜索能力和局部搜索能力,通过对13个经典的Benchmark调度问题进行仿真测试,并与近年4种新型混合算法进行对比分析,实验结果表明了所提算法求解Job Shop调度问题的有效性和稳定性。

**关键词:**Job Shop 调度问题; 帝国主义竞争算法; 遗传算法; 禁忌搜索; 混合优化算法

**中图分类号:** TP301.6; TP18    **文献标志码:**A

### Hybrid imperialist competitive algorithm for solving job-shop scheduling problem

YANG Xiaodong<sup>1</sup>, KANG Yan<sup>1,2\*</sup>, LIU Qing<sup>1,2</sup>, SUN Jinwen<sup>1</sup>

(1. College of Software, Yunnan University, Kunming Yunnan 650091, China;

2. Key Laboratory in Software Engineering of Yunnan Province (Yunnan University), Kunming Yunnan 650091, China)

**Abstract:** For the Job-shop Scheduling Problem (JSP) with the objective of minimizing the makespan, a hybrid algorithm combining with Imperialist Competitive Algorithm (ICA) and Tabu Search (TS) was proposed. Based on imperialist competitive algorithm, crossover operator and mutation operator of Genetic Algorithm (GA) were applied in the hybrid algorithm as assimilation to strengthen its global search ability. To overcome the weakness of imperialist competitive algorithm in local search, TS algorithm was used to improve the offspring of assimilation. The hybrid neighborhood structure and a novel selection strategy were used by TS to make the search more efficient. By combining with the ability of global search and local search, testing on the 13 classic benchmark scheduling problems and comparing with other four hybrid algorithms in recent years, the experimental results show that the proposed hybrid algorithm is effective and stable.

**Key words:** Job-shop Scheduling Problem (JSP); Imperialist Competitive Algorithm (ICA); Genetic Algorithm (GA); Tabu Search (TS); hybrid optimization algorithm

### 0 引言

作业车间调度问题(Job-shop Scheduling Problem, JSP)是最困难的组合优化问题之一,也是经典的NP-hard问题<sup>[1]</sup>。作业车间调度问题可以简单描述如下:给定n个作业,每个作业包含m个有预定加工顺序、加工机器和加工时间的工序。调度的目标就是合理安排所有工序在m台机器上的加工顺序,最小化最大完工时间,亦即最小化最后一个工序的完工时间。作业车间调度问题是许多实际生产调度问题的简化模型,具有广泛应用背景,如工业生产、交通规划、物流和网络通信等诸多方面的问题,都可借助于作业车间调度问题的求解结果。所以,高效、快速地求解作业车间调度问题,既有重要的理论意义,也有重大的应用价值。

求解作业车间调度问题的方法目前可以分为两大类型:精确算法和近似算法。精确算法主要通过运筹学的求解方法寻求问题的理论最优解,主要有分支定界方法(Branch and

Bound, B&B)<sup>[2]</sup>、整数规划方法(Integer Linear Programming, ILP)<sup>[3]</sup>和拉格朗日松弛法(Lagrangian Relaxation, LR)<sup>[4]</sup>等。精确算法虽然能得到理论最优解,但由于其计算复杂,运算时间长,因此只适用于小型实例。而对于大规模的调度问题,近似算法是更好的选择,近似算法主要有优先分配规则法(Priority Dispatch Rule, PDR)、瓶颈移动法(Shifting Bottleneck Procedure, SBP)、人工智能和元启发式算法等。优先分配规则法是最早提出的近似算法,Giffler等<sup>[5]</sup>提出的构造主动高度和无延迟调度的算法被认为是优先分配规则法的基础。瓶颈移动法由Adams等<sup>[6]</sup>提出,并且是首次求出经典的FT10问题最优解的近似算法。随后人们尝试将人工智能的研究成果,如约束满足技术、专家系统和人工神经网络等应用到解决调度问题中,其中比较经典的有文献[7–9]中应用Hopfield模型以及文献[10]中应用反向传播(Back Propagation, BP)神经网络模型求解调度问题。元启发式算法<sup>[11]</sup>是近些年求解调度问题最热门、最高效的算法,主要包括基于生物启发的

收稿日期:2016-08-20;修回日期:2016-09-25。

基金项目:国家自然科学基金资助项目(61462095);云南省软件工程重点实验室开放基金资助项目(2015SE103)。

作者简介:杨小东(1991—),男,福建连城人,硕士研究生,CCF会员,主要研究方向:智能车间调度;康雁(1972—),女,四川绵阳人,副教授,博士,主要研究方向:智能车间调度、数据挖掘;柳青(1963—),男,云南祥云人,教授,硕士,主要研究方向:软件体系结构、海量数据处理和分析;孙金文(1989—),男,山西朔州人,硕士研究生,主要研究方向:模糊决策。



群体算法和局部搜索算法。其中基于生物启发的群体算法有遗传算法 (Genetic Algorithm, GA)、蚁群优化 (Ant Colony Optimization, ACO) 算法和粒子群优化 (Particle Swarm Optimization, PSO) 算法等; 局部搜索算法有禁忌搜索 (Tabu Search, TS) 算法和模拟退火 (Simulated Annealing, SA) 算法等。

帝国主义竞争算法 (Imperialist Competitive Algorithm, ICA) 是 Atashpaz-Gargari 等<sup>[12]</sup> 在 2007 年受历史上帝国之间的竞争启发而提出的一种新型进化算法。算法中的初始种群被分成两类: 帝国和殖民地。殖民地通过不断向帝国学习进化自身, 而帝国之间的竞争使得所有殖民地最终归附到一个最强的帝国之下。禁忌搜索 (TS) 算法是 Glover<sup>[13]</sup> 在 1986 年提出的一种经典的局部搜索算法。禁忌搜索算法基于局部邻域搜索方法, 通过设置禁忌表来避免迂回搜索, 利用藐视准则来特赦被禁忌的优解, 能够高效地求解优化问题。

近些年的研究显示, 混合算法比单一算法能提供更高质量的解和更好的搜索效率。因此, 将单一算法有机组合以弥补各自的不足成为了研究者关注的热点。文献 [14] 将树搜索算法与人工蜂群算法进行结合, 强化对 JSP 的搜索能力; 文献 [15] 在文化算法中融入了局部搜索, 对求解 JSP 进行了研究; 文献 [16] 将人工免疫系统与禁忌算法混合求解 JSP, 提出了带禁忌搜索的人工免疫系统 (Artificial Immune System with Tabu Search, AIS-TS) 算法; 文献 [17] 将局部搜索和遗传算法结合求解 JSP; 文献 [18] 将空闲时间的邻域搜索融入到遗传算法中求解 JSP。

本文将基于帝国主义竞争算法和禁忌搜索算法提出一种新型混合算法 HICATS (Hybrid Imperialist Competitive Algorithm with Tabu Search)。帝国主义竞争算法具有很强的全局优化能力, 但缺乏局部搜索能力, 而禁忌搜索算法虽然缺乏全局搜索能力, 但具有很强的局部搜索能力, 因此两者的结合能够达到优势互补。针对问题特性, 混合算法还融入遗传算法的交叉和变异算子, 使算法能够更高效地求解 JSP。

## 1 问题模型

经典的作业车间调度问题可以描述如下: 给定  $m$  台机器,  $n$  个作业, 每个作业按指定顺序, 依次在  $m$  台机器上加工, 即每个作业都包含  $m$  个工序。对作业和机器的约束如下:

- 1) 每个工序有预先设置的加工时间与加工机器;
- 2) 每个工序有且只能在一台机器上加工一次;
- 3) 不同作业之间无先后约束关系;
- 4) 每个工序一旦开始就不能中断;
- 5) 每台机器同一时刻只能加工一个作业;
- 6) 不计工序开始前布置时间和工序之间交接时间;

本文约定符号如下:  $m$  台机器表示为  $\{M_1, M_2, \dots, M_m\}$ ,  $n$  个作业表示为  $\{J_1, J_2, \dots, J_n\}$ , 第  $i$  个作业的第  $j$  个工序表示为  $o_{ij}$ , 工序  $o_{ij}$  对应的加工机器表示为  $p_{ij}$ , 对应的加工时间为  $t_{ij}$ 。每个作业最后一道工序的完工时间即是该作业的完成时间, 表示为  $C_i$ 。求解作业车间调度问题即是最小化所有作业的最大完工时间  $C_{\max}$ , 目标函数如式(1) 所示:

$$C_{\max} = \min \left\{ \max_{i=1}^n (C_i) \right\} \quad (1)$$

一个  $3 \times 3$  的 JSP 实例如表 1 所示, 其中“—”代表工序在

机器上无加工数据。

表 1 一个  $3 \times 3$  实例  
Tab. 1 An instance with  $3 \times 3$  jobs

作业	工序	编号	加工机器及时间		
			$M_1$	$M_2$	$M_3$
$J_1$	$o_{11}$	1	—	—	2
	$o_{12}$	2	—	1	—
	$o_{13}$	3	1	—	—
$J_2$	$o_{21}$	4	—	—	1
	$o_{22}$	5	—	2	—
	$o_{23}$	6	2	—	—
$J_3$	$o_{31}$	7	4	—	—
	$o_{32}$	8	—	1	—
	$o_{33}$	9	—	—	2

## 2 HICATS 算法

帝国主义竞争算法是新型的进化算法, 其中所有的国家将被分为两类: 一是帝国; 二是殖民地。具有较优适应度值的国家将成为帝国, 拥有较差适应度值的国家将成为帝国的殖民地。各帝国之间不断展开竞争, 强帝国以大几率从弱帝国中夺得殖民地。整个竞争过程到只剩一个帝国、达到算法迭代次数或找到最优解时停止。

本文提出的混合帝国主义禁忌搜索算法 HICATS 是一种新型的元启发式算法, 算法集成了帝国主义竞争算法与禁忌搜索算法的优点, 且融合了遗传算法的杂交算子和变异算子, 使得该算法能够高效求解 JSP。

### 2.1 初始化帝国与殖民地

本文使用基于工序的编码方式, 并采用 Zhang 等<sup>[19]</sup> 提出的全主动调度 (Full-Active Schedule, FAS) 来提高初始解的质量。传统主动调度 (Active Scheduling, AS) 能够在不破坏优先顺序以及不延迟其他工序的条件下, 使得没有工序可以提前; 而全主动调度则是在主动调度的基础上, 增加了反转调度, 使得不仅可以将允许左移的工序左移, 也可以将允许右移的工序右移, 因此全主动调度的最大完工时间小于或者等于其对应的主动调度的最大完工时间。

以表 1 中  $3 \times 3$  实例为例, 随机生成一个初始解  $\{1, 2, 1, 2, 1, 3, 2, 3, 2\}$ , 解对应的工序编号为  $\{1, 4, 2, 5, 3, 7, 6, 8, 9\}$ , 其 AS 结果如图 1 所示, 调度结果为 11。

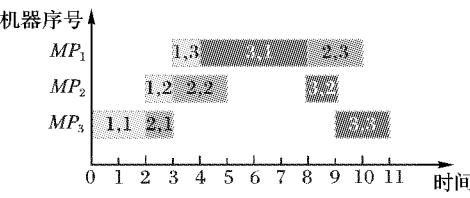


图 1 AS 调度结果

Fig. 1 Result of scheduling obtained by AS

而对于同样的初始解, 采用 FAS 调度结果如图 2 所示。图 2(a) 是将允许右移的工序右移的反转调度, 即以初始解反向开始调度, 调度顺序为  $\{9, 8, 6, 7, 3, 5, 2, 4, 1\}$ , 同时也是基于主动调度方式, 最终调度的结果是 8; 图 2(b) 为基于反转调度结果的主动调度, 使能够左移的工序左移, 同时也得到各个工序正确的开工和结束时间, 并形成最终的调度结果。从图 2 中可知, 对于同样的初始解, AS 的调度结果是 11, 而 FAS 的调度结果是 8。



示例演示了FAS的有效性,而在FT10实例上的实验进一步验证了FAS生成初始解的高质量。实验分别用FAS和AS随机生成100个FT10的初始解,从图3的实验结果中可以看出,不论最优适应值还是平均适应值,FAS生成的初始国家都优于AS生成的初始国家。

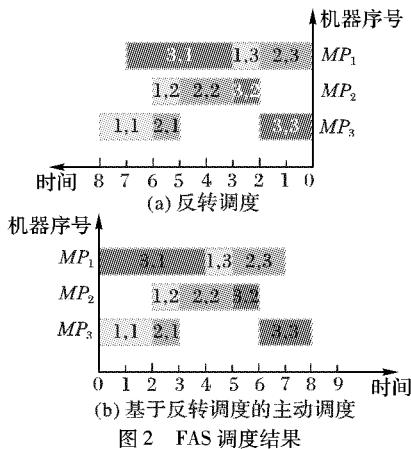


Fig. 2 Result of scheduling obtained by FAS

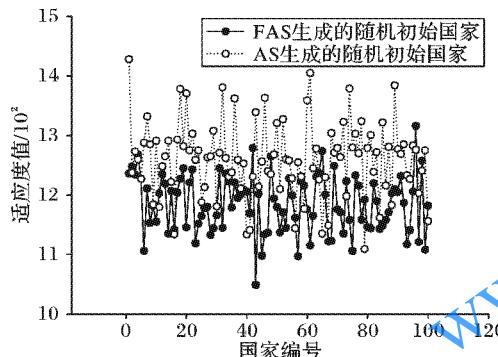


Fig. 3 Initial makespan comparison between FAS & AS

本文中帝国主义竞争算法的符号与Atashpaz-Gargari等<sup>[12]</sup>定义帝国主义竞争算法时保持一致。FAS调度生成的初始国家定义为 $N_{\text{pop}}$ ,从中选择适应值最优的 $N_{\text{imp}}$ 个国家为帝国,剩下的 $N_{\text{col}}$ 个国家将作为殖民地并根据各帝国力量随机分配到各个帝国旗下。按照式(2)标准化各帝国力量:

$$C_n = c_n - \max_i \{c_i\} \quad (2)$$

其中: $c_n$ 是第n个帝国的适应度值; $C_n$ 是其标准化后的适应度值。基于标准化后的适应度值,用式(3)计算出各个帝国的力量:

$$p_n = \left| C_n / \left( \sum_{i=1}^{N_{\text{imp}}} C_i \right) \right| \quad (3)$$

其中 $p_n$ 是一个比值,越大代表这个帝国拥有的力量越强,因此应该拥有的殖民地也就越多。基于 $p_n$ ,按式(4)给每个帝国划分应该拥有的殖民地数:

$$N.C._n = \text{round} \{ p_n \times N_{\text{col}} \} \quad (4)$$

其中 $N.C._n$ 就是每个帝国初始拥有的殖民地数,按照这个数量将殖民地随机分配到各个帝国旗下。这样就完成对所有帝国及殖民地的初始化。

## 2.2 同化操作

在Atashpaz-Gargari等<sup>[12]</sup>提出的帝国主义竞争算法中,采用距离公式实现殖民地向帝国的学习移动过程,而本文则

根据问题特性借鉴遗传算法的杂交思想作为学习过程。在一个帝国集团中,每一组殖民地将按照概率 $P_c$ 从以下两种杂交方式随机选择一种:

- 1) 任一殖民地与所在帝国按照优先工序交叉(Precedence Operation crossover, POX)算法交叉T次;
- 2) 两个殖民地按照POX算法交叉T次。

其中优先工序交叉(POX)算法是Zhang等<sup>[19]</sup>在2008年提出的一种高效交叉算子。假设有父代Parent1和Parent2,POX产生两个子代的流程如下:

- 1) 随机划分工件集 $\{1, 2, \dots, n\}$ 为两个非空子集 $S_1$ 和 $S_2$ ;
- 2) 复制Parent1包含在 $S_1$ 中的工件到Child1,Parent2包含在 $S_1$ 中的工件到Child2,分别保留它们的位置;
- 3) 复制Parent2包含在 $S_2$ 中的工件到Child1,Parent1包含在 $S_2$ 中的工件到Child2,分别保留它们的顺序。

如图4所示,Parent1 = {1, 2, 1, 2, 1, 3, 2, 3, 3}, Parent2 = {2, 1, 3, 1, 2, 2, 3, 1, 3},  $S_1 = \{2\}$ ,  $S_2 = \{1, 3\}$ 。最后生成的Child1 = {1, 2, 3, 2, 1, 3, 2, 1, 3}, Child2 = {2, 1, 1, 1, 2, 2, 3, 3, 3}。通过POX方法既能很好地保留父代的优良特征,而且产生的子代又能保证是可行解,因此是一种高效的交叉算子。

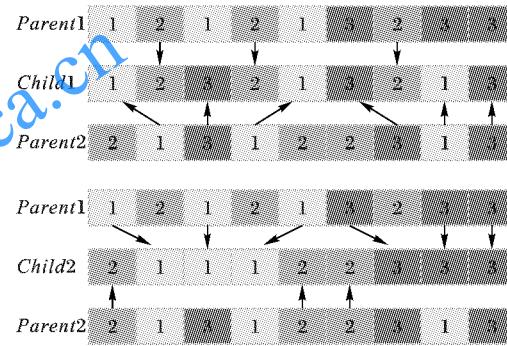


Fig. 4 POX crossover operator

交叉结束后对最优的后代使用禁忌搜索算法,进一步探索邻域中更优的个体。

禁忌搜索算法中最重要的部分就是邻域结构,其中比较有效的结构有N4、N5、N6邻域结构<sup>[20-22]</sup>。N4邻域结构要求把关键块内工序尽可能地移至块首或块尾位置;N5邻域结构仅交换块首和块尾两相邻工序以避免交换关键块内的工序;N6邻域结构是把内部工序移至块首之前和块尾之后。本文在N4、N5、N6邻域结构的基础上,采用混合的邻域结构,具体如下:

- 1) 关键块中含有两个工序,则直接交换。
- 2) 关键块中含有三个工序:
  - a) 非首工序移至块首;
  - b) 非尾工序移至块尾。
- 3) 关键块中含有四个及以上工序:
  - a) 随机选择部分非首工序移至块首;
  - b) 随机选择部分非尾工序移至块尾;
  - c) 首工序移至随机一个中间位置;
  - d) 尾工序移至随机一个中间位置;
  - e) 块首和块尾同时与其相邻工序交换位置。

通过以上的混合邻域结构,既能获得高质量的邻域解,又



能使算法高效运行。

选择策略是禁忌搜索算法中下次迭代的开始。通常禁忌搜索算法会从非禁忌的最优解中选择最优的一个解作为下次算法的初始解,但在算法的后期容易陷入局部最优。本文禁忌搜索算法受模拟退火算法中以一定概率选择差解启发,提出一种新型选择策略,即下次初始解的产生是从  $L$  个非禁忌的最优解中随机选择,  $L$  是一个随迭代次数增加而增加的值,按式(5)计算:

$$L = \text{iteration}/\text{cycle} + 1 \quad (5)$$

其中:*iteration* 是算法运行代数,*cycle* 是自定义的一个递增周期。通过此方法可使算法在初期加速收敛,而在后期扩大解的搜索范围,避免过早陷入局部最优。

禁忌搜索算法的流程如图 5 所示。

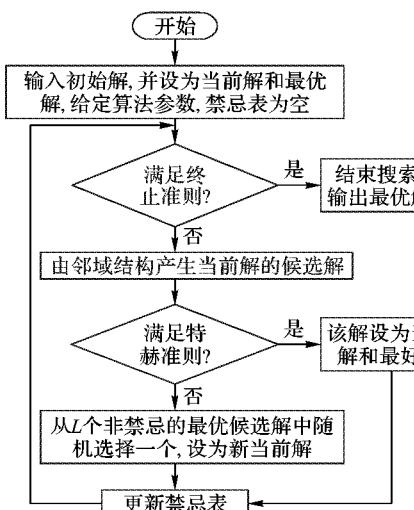


Fig. 5 Flowchart of tabu search algorithm

当两个适应度值相同的个体被选择时,则对任一个体进行变异。变异算子的流程如图 6 所示。首先随机选择  $K$  个位置,并且各个位置上的基因各不相同;再基于选择的  $K$  个基因作随机变换,从而得到变异个体。

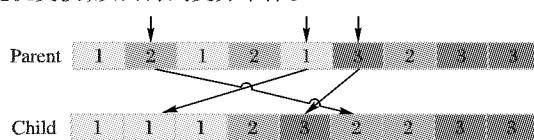


Fig. 6 Mutation operator

同化操作是帝国主义竞争算法的核心。针对 JSP 特性,采用遗传算法的交叉思想实现进化;同时为了避免陷入局部最优,引入变异算子扩大搜索空间;最后再使用局部搜索能力强的禁忌搜索算法进一步优化后代,从而实现更全、更快的个体进化。当新的个体适应度值优于当前帝国时,则将更优个体取代旧帝国。此后该帝国下的同化操作将以新的帝国为基础,直到该帝国内的同化操作结束或有更优的后代取代该帝国。同化操作的整个流程如图 7 所示。

### 2.3 帝国竞争

在帝国互相竞争之前,需要衡量各个帝国集团的力量,不仅需要考虑帝国的适应度值,还需要考虑其所拥有的殖民地对帝国集团力量的影响。而这个影响系数本文定义为  $\xi$ ,第  $n$

个帝国集团的总力量定义为  $T.C._n$ ,计算公式如式(6)所示:

$$T.C._n = \text{Cost}(\text{imperialist}_n) + \xi \cdot \text{mean}\{\text{Cost}(\text{colonies of empire}_n)\} \quad (6)$$

帝国竞争是实现帝国之间优胜劣汰的重要步骤。各帝国将基于各自的力量来争夺最弱帝国中的最弱殖民地。首先按式(7)标准化各帝国集团的总力量:

$$N.T.C._n = T.C._n - \max_i\{T.C._i\} \quad (7)$$

其中  $N.T.C._n$  代表第  $n$  个帝国集团标准化后的力量。基于标准化力量,各帝国集团能够获得殖民地的概率如式(8)所示:

$$p_{p_n} = \left| N.T.C._n / \left( \sum_{i=1}^{N_{\text{imp}}} N.T.C._i \right) \right| \quad (8)$$

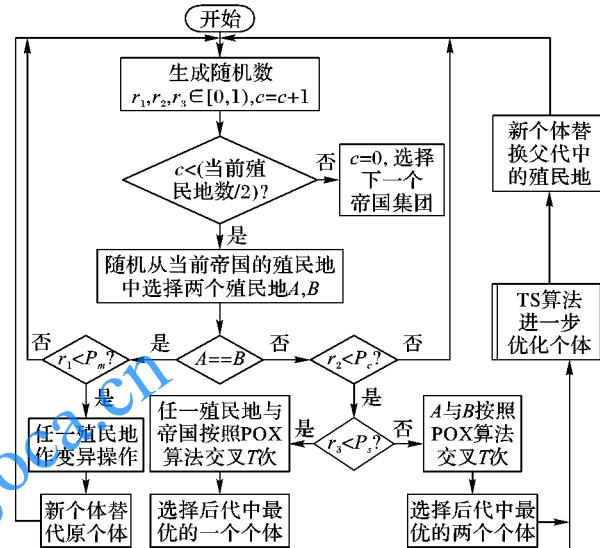


图 7 同化操作流程

Fig. 7 Assimilating flowchart

为了将最弱帝国下最弱殖民地按各帝国集团所应该拥有的概率来分配,首先基于之前计算的概率初始化一个向量  $P$ :

$$P = [p_{p_1}, p_{p_2}, \dots, p_{p_{N_{\text{imp}}}}]$$

然后随机初始化一个与向量  $P$  同样大小的向量  $R$ :

$$R = [r_1, r_2, \dots, r_{N_{\text{imp}}}] ; r_1, r_2, \dots, r_{N_{\text{imp}}} \sim U(0,1) \text{ 最后通过将向量 } P \text{ 与向量 } R \text{ 相减来获得向量 } D :$$

$$D = P - R = [D_1, D_2, \dots, D_{N_{\text{imp}}}] =$$

$$[p_{p_1} - r_1, p_{p_2} - r_2, \dots, p_{p_{N_{\text{imp}}}} - r_{N_{\text{imp}}}]$$

向量  $D$  中最大值对应的帝国将获得一个新的殖民地。

在每一次迭代中,将检查当前最弱帝国拥有的殖民地数,当其流失了所有的殖民地后,该帝国集团将被认为已崩塌,将删除该帝国。

算法终止准则是满足以下三个条件之一:一是找到最优解;二是算法达到最大迭代数;三是只剩一个帝国。帝国主义竞争算法的流程如图 8 所示。

### 3 实验分析

为了验证本文算法的实际效果,在 3.3 GHz CPU 和 4.0 GB 内存的计算机上,采用 Java 编程进行实验;算法参数设置为:国家总数为 100,其中初始帝国数为 10,初始殖民地为 90,帝国竞争次数为 200,交叉概率  $P_c$  为 0.8,变异率  $P_m$  为 0.1,每代 POX 交叉次数  $T$  为 200,每个实例求解 20 次。



本文算法选用 13 个经典实例与其他算法进行对比分析,这些实例在文献[22]中被认定为“难”级别,因此更具有代表性。对比结果如表 2 所示,其中:Instance 是实例名;Size 是由作业数和机器数表示的实例规模;BKS\* 代表实例已知的最优解; $C_{\max}$  代表各算法求得实例的最优解; $Re$  为  $C_{\max}$  与  $BKS^*$  的相对偏差,即  $Re = (C_{\max} - BKS^*)/BKS^* \times 100\%$ ;Av(Re) 为各算法对 13 个实例相对偏差的平均值;“—”表示无对应数据。对比算法分别为文献[14]的混合人工蜂群(Hybrid Artificial Bee Colony, HABC)算法、文献[15]的文化基因算法(Memetic Algorithm, MA)、文献[17]的混合遗传算法(Hybrid Genetic Algorithm, HGA)和文献[18]的空闲时间邻域搜索遗传算法(Idle Time Neighborhood Search Genetic Algorithm, ITNSGA)。

表 2 典型实例求解结果  
Tab. 2 Computational results of classic instances

Instance	Size	BKS*	HICATS		ITNSGA <sup>[18]</sup>		HGA <sup>[17]</sup>		MA <sup>[15]</sup>		HABC <sup>[14]</sup>	
			$C_{\max}$	Re/%	$C_{\max}$	Re/%	$C_{\max}$	Re/%	$C_{\max}$	Re/%	$C_{\max}$	Re/%
FT10	$10 \times 10$	930	930	0.00	930	0.00	930	0.00	930	0.00	935	0.54
LA02	$10 \times 5$	655	655	0.00	655	0.00	655	0.00	655	0.00	655	0.00
LA19	$10 \times 10$	842	842	0.00	842	0.00	844	0.24	842	0.00	842	0.00
LA21	$15 \times 10$	1046	1052	0.57	1052	0.57	1046	0.00	1055	0.86	1052	0.57
LA24	$15 \times 10$	935	941	0.64	941	0.64	953	1.93	940	0.53	940	0.53
LA25	$15 \times 10$	977	984	0.72	977	0.00	981	0.41	984	0.72	982	0.51
LA27	$20 \times 10$	1235	1255	1.62	1246	0.89	1236	0.08	1261	2.11	1243	0.65
LA29	$20 \times 10$	1152	1176	2.08	1165	1.13	1160	0.69	1190	3.30	1180	2.43
LA36	$15 \times 15$	1268	1281	1.03	1279	0.87	1287	1.50	1281	1.03	1274	0.47
LA37	$15 \times 15$	1397	1425	2.00	1407	0.72	1407	0.72	1431	2.43	1408	0.79
LA38	$15 \times 15$	1196	1214	1.51	1213	1.42	1196	0.00	1216	1.67	1196	0.00
LA39	$15 \times 15$	1233	1249	1.30	1244	0.89	1233	0.00	1241	0.65	1238	0.41
LA40	$15 \times 15$	1222	1242	1.64	1233	0.90	1229	0.57	1233	0.90	1233	0.90
Av(Re)	—	—	—	1.01	—	0.62	—	0.47	—	1.09	—	0.60

从表 2 中可以看出本文算法能够求解出 13 个实例中的 3 个最优解。与 ITNSGA<sup>[18]</sup>相比,除了结果相等的实例,并没有发现更优的实例,但与其他算法的对比可以发现各有优劣。进一步统计分析,本文算法与对比算法的求解优劣数统计对比如表 3 所示。其中:AS 表示本文算法求解的 13 个实例中优于对比算法的数量,LS 表示本文算法差于对比算法的实例数,ES 表示本文算法与对比算法相同的求解实例数。从表中可知,本文算法求解的实例质量是具有一定竞争力的,但是从 LS 列中也可以看出,本文算法除了优于 MA 算法外,与其他三个混合算法都有些差距,还有待完善。

表 3 优于解和差于解数目对比分析

Tab. 3 Comparative analysis between prior solutions and inferior solutions

算法	AS	LS	ES	算法	AS	LS	ES
ITNSGA <sup>[18]</sup>	0	9	4	MA <sup>[15]</sup>	5	3	5
HGA <sup>[17]</sup>	3	8	2	HABC <sup>[14]</sup>	2	9	2

根据表 2 的 Av(Re) 结果也可以看出,本文算法的整体性能还有待进一步提高。

图 9 给出了本文算法在解决 FT10( $10 \times 10$ )基准问题时的收敛曲线,从图中可以看出算法在第 20 代左右收敛到近似最优解,随后在第 110 代左右求得最优解,说明算法具有较快的收敛性。

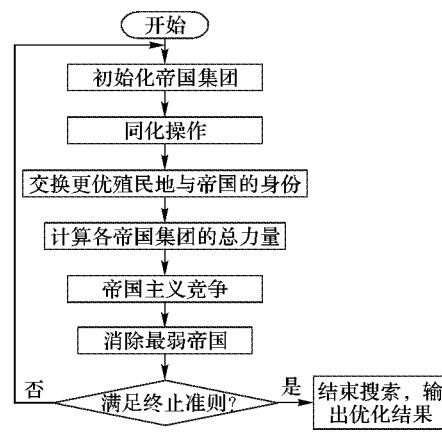
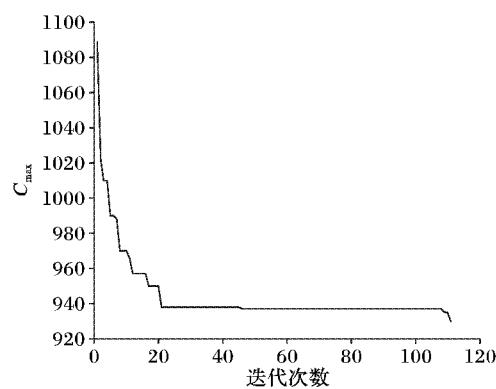


图 8 帝国主义竞争算法流程

Fig. 8 Flowchart of imperialistic competitive algorithm

图 9 FT10( $10 \times 10$ )基准问题的收敛曲线Fig. 9 Convergence curve of FT10 ( $10 \times 10$ ) benchmark

在求解速度和平均结果方面,将本文算法与 ITNSGA<sup>[18]</sup>进行对比,结果如表 4 所示,其中 ITNSGA 的求解环境是 CPU 主频为 2.6 GHz、内存为 2.0 GB 的个人计算机。表 4 中: $pSize$  表示初始种群大小; $Av(C_{\max})$  表示 20 次独立实验的平均结果; $Av(t)$  表示 20 次独立实验的平均运行时间; $\gamma$  是本文算法与对比算法在实例平均结果( $Av(C_{\max})$ )上的比值; $\lambda$  是本文算法与对比算法在实例平均运行时间( $Av(t)$ )上的比值; $Av(\gamma/\lambda)$  表示 13 个实例  $\gamma$  值的平均值和  $\lambda$  值的平均值;“—”



表示无对应数据。从表 4 中可见,尽管本文算法使用的 CPU 主频和内存较大,但在整体硬件环境相差不大的情况下,求得  $Av(\lambda) = 0.34$ ,说明本文算法的平均求解速度大约是对比算

法的 3 倍,验证了本文算法的高效性。同时本文算法的初始种群在远小于对比算法初始种群的情况下,由  $Av(\gamma) = 1$  可知两者算法的平均求解能力几乎相等,验证了算法的稳定性。

表 4 本文算法与 ITNSGA<sup>[18]</sup>的平均求解时间和结果对比  
Tab. 4 Average runtime and result comparison with ITNSGA<sup>[18]</sup>

Instance	Size	BKS*	HICATS				ITNSGA <sup>[18]</sup>				$\gamma$	$\lambda$
			pSize	$C_{\max}$	$Av(C_{\max})$	$Av(t)/s$	pSize	$C_{\max}$	$Av(C_{\max})$	$Av(t)/s$		
FT10	$10 \times 10$	930	100	930	934.55	28.9	600	930	937.25	127.9	0.997	0.23
LA02	$10 \times 5$	655	100	655	655.00	2.0	200	655	655.45	13.1	0.999	0.15
LA19	$10 \times 10$	842	100	842	845.15	18.9	200	842	846.95	23.6	0.998	0.80
LA21	$15 \times 10$	1046	100	1052	1057.00	67.4	400	1052	1055.65	198.9	1.001	0.34
LA24	$15 \times 10$	935	100	941	947.50	64.3	400	941	945.90	125.6	1.002	0.51
LA25	$15 \times 10$	977	100	984	985.05	65.7	500	977	987.25	332.4	0.998	0.20
LA27	$20 \times 10$	1235	100	1255	1263.20	104.8	400	1246	1263.80	460.0	1.000	0.23
LA29	$20 \times 10$	1152	100	1176	1192.05	107.7	400	1165	1190.25	610.9	1.002	0.18
LA36	$15 \times 15$	1268	100	1281	1297.10	110.1	500	1279	1290.65	233.1	1.005	0.47
LA37	$15 \times 15$	1397	100	1425	1441.20	109.5	500	1407	1432.25	478.2	1.006	0.23
LA38	$15 \times 15$	1196	100	1214	1225.80	95.8	400	1213	1235.45	270.1	0.992	0.35
LA39	$15 \times 15$	1233	100	1249	1250.00	101.9	500	1244	1249.45	317.6	1.000	0.32
LA40	$15 \times 15$	1222	100	1242	1248.05	101.3	400	1233	1250.20	287.6	0.998	0.35
$Av(\gamma/\lambda)$	—	—	—	—	—	—	—	—	—	—	1.000	0.34

## 4 结语

本文针对作业车间调度问题展开研究,提出一种新型算法 HICATS,混合了帝国主义竞争算法全局搜索能力和禁忌搜索算法的局部搜索能力。HICATS 算法首先在帝国主义竞争算法的同化操作中融入了遗传算法的交叉算子和变异算子,有效地提高殖民地的学习能力;然后应用禁忌搜索算法进一步地优化后代,利用混合邻域结构使禁忌搜索更为有效;并借鉴模拟退火算法以一定概率接受差解的思想,提出一种新型选择策略克服禁忌搜索算法易陷入局部最优的缺陷。该混合算法在最后的实例验证中表现出了高效的求解能力,验证了算法的有效性和稳定性。今后将进一步研究该算法在大型实例中的设计与实现,进一步优化算法性能,并探索算法在相类似问题领域中的应用。

## 参考文献 (References)

- [1] GAREY M R, JOHNSON D S, SETHI R. The complexity of flow-shop and jobshop scheduling [J]. Mathematics of Operations Research, 1976, 1(2): 117–129.
- [2] BALAS E. Machine sequencing via disjunctive graphs: an implicit enumeration algorithm [J]. Operations Research, 1968, 17(6): 941–957.
- [3] VAN HULLE M M. A goal programming network for mixed integer linear programming: a case study for the job-shop scheduling problem [J]. International Journal of Neural Systems, 1991, 2(3): 201–209.
- [4] LUH P B, HOITOMT D J. Scheduling of manufacturing systems using the Lagrangian relaxation technique [J]. IEEE Transactions on Automatic Control, 1993, 38(7): 1066–1079.
- [5] GIFFLER B, THOMPSON G L. Algorithms for solving production scheduling problems [J]. Operations Research, 1960, 8(4): 487–503.
- [6] ADAMS J, BALAS E, ZAWACK D. The shifting bottleneck procedure for job shop scheduling [J]. Management Science, 1988, 34(3): 391–401.
- [7] SIMON F Y-P, TAKEFUJI Y. Stochastic neural networks for solving job-shop scheduling. I. problem representation [C]// Proceedings of the 1988 IEEE International Conference on Neural Networks. Piscataway, NJ: IEEE, 1988: 275–282.
- [8] SIMON F Y-P, TAKEFUJI Y. Stochastic neural networks for solving job-shop scheduling. II. architecture and simulations [C]// Proceedings of the 1988 IEEE International Conference on Neural Networks. Piscataway, NJ: IEEE, 1988: 283–290.
- [9] SIMON F Y-P, TAKEFUJI Y. Integer linear programming neural networks for job-shop scheduling [C]// Proceedings of the 1988 IEEE International Conference on Neural Networks. Piscataway, NJ: IEEE, 1988: 341–348.
- [10] REMUS W. Neural network models of managerial judgment [C]// Proceedings of the 23rd Annual Hawaii International Conference on System Sciences. Piscataway, NJ: IEEE, 1990: 340–344.
- [11] GLOVER F. New approaches for heuristic search: a bilateral linkage with artificial intelligence [J]. European Journal of Operational Research, 1989, 39(2): 119–130.
- [12] ATASHPAZ-GARGARI E, LUCAS C. Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition [C]// CEC 2007: Proceedings of the 2007 IEEE Congress on Evolutionary Computation. Piscataway, NJ: IEEE, 2007: 4661–4667.
- [13] GLOVER F. Future paths for integer programming and links to artificial intelligence [J]. Computers & Operations Research, 1986, 13(5): 533–549.
- [14] ZHANG R, SONG S, WU C. A hybrid artificial bee colony algorithm for the job shop scheduling problem [J]. International Journal of Production Economics, 2013, 141(1): 167–178.
- [15] GAO L, ZHANG G, ZHANG L, et al. An efficient memetic algorithm for solving the job shop scheduling problem [J]. Computers & Industrial Engineering, 2011, 60(4): 699–705.

(下转第 552 页)



- (2): 023006.
- [50] LIU Y, FANG Z, LI H, et al. A mixed finite element method for a time-fractional fourth-order partial differential equation [J]. Applied Mathematics and Computation, 2014, 243: 703–717.
- [51] ATANGANA A, BALEANU D. Numerical solution of a kind of fractional parabolic equations via two difference schemes [J]. Abstract & Applied Analysis, 2013, 2013: Article ID 828764.
- [52] MEERSCHAERT M M, TADJERAN C. Finite difference approximations for fractional advection dispersion equations [J]. Journal of Computational and Applied Mathematics, 2004, 172(1): 65–77.
- [53] ZHANG Y. A finite difference method for fractional partial differential equation [J]. Applied Mathematics and Computation, 2009, 215(2): 524–529.
- [54] TADJERAN C, MEERSCHAERT M M, SCHEFFLER H-P. A second order accurate numerical approximation for the fractional diffusion equation [J]. Journal of Computational Physics, 2016, 213(1): 205–213.
- [55] CHEN C-M, LIU F, TURNER I, et al. A Fourier method for the fractional diffusion equation describing sub-diffusion [J]. Journal of Computational Physics, 2007, 227(2): 886–897.
- [56] YUSTE S B, ACEDO L. An explicit finite difference method and a new von Neumann-type stability analysis for fractional diffusion equations [J]. SIAM Journal on Numerical Analysis, 2005, 42(5): 1862–1874.
- [57] AL-KHALED K, MOMANI S. An approximate solution for a fractional diffusion-wave equation using the decomposition method [J]. Applied Mathematics and Computation, 2005, 165(2): 473–483.
- [58] SAKTIVEL R, REVATHI P, REN Y. Existence of solutions for nonlinear fractional stochastic differential equations [J]. Nonlinear Analysis: Theory, Methods & Applications, 2013, 81: 70–86.
- [59] BAYOURA B, TORRESB D F M. Existence of solution to a local fractional nonlinear differential equation [J]. Journal of Computational and Applied Mathematics, 2017, 312: 127–133.
- [60] ZHU B, LIU L, WU Y. Existence and uniqueness of global mild solutions for a class of nonlinear fractional reaction-diffusion equa-
- tions with delay [J/OL]. Computers and Mathematics with Applications, 2016 [2016-07-16]. <http://www.sciencedirect.com/science/article/pii/S0898122116300244>.
- [61] ATANGANA A. On the stability and convergence of the time-fractional variable order telegraph equation [J]. Journal of Computational Physics, 2015, 293: 104–114.
- [62] LI F, LI Z, PI L. Variable exponent functionals in image restoration [J]. Applied Mathematics and Computation, 2010, 216(3): 870–882.
- [63] WEI G W. Generalized Perona-Malik equation for image restoration [J]. IEEE Signal Processing Letters, 1999, 6(7): 165–167.
- [64] TSENG C-C, PEI S-C, HSIA S-C. Computation of fractional derivatives using Fourier transform and digital FIR differentiator [J]. Signal Processing, 2000, 80(1): 151–159.
- [65] TAI Y-W, LIU S, BROWN M S, et al. Super resolution using edge prior and single image detail synthesis [C]// CVPR 2010: Proceedings of the 2010 IEEE Computer Vision and Pattern Recognition. Washington, DC: IEEE Computer Society, 2010: 2400–2407.
- [66] MARQUINA A, OSHER S J. Image super-resolution by TV regularization and Bregman iteration [J]. Journal of Scientific Computing, 2008, 37(3): 367–382.

This work is partially supported by the Major Project of Fundamental Science and Frontier Technology Research of Chongqing CSTC (cstc2015jcyjBX0124).

**ZHOU Shangbo**, born in 1963, Ph. D., professor. His research interests include video and image processing, artificial neural network, nonlinear dynamics.

**WANG Liping**, born in 1981, Ph. D. candidate. His research interests include partial differential equation, nonlinear dynamics, pattern recognition, image restoration.

**YIN Xuehui**, born in 1986, Ph. D., lecturer. His research interests include partial differential equations for image processing, nonlinear dynamics.

(上接第 522 页)

- [16] ZUO X, WANG C, TAN W. Two heads are better than one: an AIS- and TS-based hybrid strategy for job shop scheduling problems [J]. International Journal of Advanced Manufacturing Technology, 2012, 63(1): 155–168.
- [17] REN Q-D-E-J, WANG Y. A new hybrid genetic algorithm for job shop scheduling problem [J]. Computers & Operations Research, 2012, 39(10): 2291–2299.
- [18] 赵诗奎, 方水良, 顾新建. 作业车间调度的空闲时间邻域搜索遗传算法[J]. 计算机集成制造系统, 2014, 20(8): 1930–1940。(ZHAO S K, FANG S L, GU X J. Idle time neighborhood search genetic algorithm for job shop scheduling [J]. Computer Integrated Manufacturing Systems, 2014, 20(8): 1930–1940.)
- [19] ZHANG C, RAO Y, LI P. An effective hybrid genetic algorithm for the job shop scheduling problem [J]. International Journal of Advanced Manufacturing Technology, 2008, 39(9): 965–974.
- [20] DELL'AMICO M, TRUBIAN M. Applying tabu-search to the job shop scheduling problem [J]. Annals of Operations Research, 1993, 41(3): 231–252.
- [21] NOWICKI E, SMUTNICKI C. A fast taboo search algorithm for the

job shop problem [J]. Management Science, 1996, 42(6): 797–813.

[22] BALAS E, VAZACOPOULOS A. Guided local search with shifting bottleneck for job shop scheduling [J]. Management Science, 1998, 44(2): 262–275.

This work is partially supported by the National Natural Science Foundation of China (61462095), the Open Foundation of Key Laboratory in Software Engineering of Yunnan Province (2015SE103).

**YANG Xiaodong**, born in 1991, M. S. candidate. His research interests include intelligent shop scheduling.

**KANG Yan**, born in 1972, Ph. D., associate professor. Her research interests include intelligent shop scheduling, data mining.

**LIU Qing**, born in 1963, M. S., professor. His research interests include software hierarchical architecture, massive data processing and analysis.

**SUN Jinwen**, born in 1989, M. S. candidate. His research interests include fuzzy decision-making.