

基于多小组协同学习教学算法的车间作业调度问题

张梅^{1†}, 杨晟轩¹, 朱金辉²

(1. 华南理工大学 自动化科学与工程学院, 广州 510641; 2. 华南理工大学 软件学院, 广州 510641)

摘要: 为求解车间作业调度问题(JSSP), 提出一种新颖的多小组协同学习的教学算法, 实现小组间学习的协同及基于学习能力的深度和广度搜索策略. 针对JSSP问题因其复杂度较高容易导致算法陷入局部最优的不足, 引入学习小组协同学习, 通过组内学习和组内交流, 使学习过程跳出当前的局限. 为了兼顾局部和全局搜索能力, 引入基于学习能力的深度和广度搜索策略, 小组内学生按照学习能力强弱进行学习, 较优的学生进行深度的学习, 较差的学生进行广度的学习. 最后, 对OR-Library中的标准仿真实例进行实验, 结果表明, 所提出的教学算法在JSSP问题上的收敛精度和搜索能力较其他算法均得到了有效的提高.

关键词: 小组学习; 教学优化算法; 协同进化; 车间作业调度

中图分类号: TP18

文献标志码: A

Teaching-learning-based optimization algorithm with group collaboration for job shop scheduling problem

ZHANG Mei^{1†}, YANG Sheng-xuan¹, ZHU Jin-hui²

(1. College of Automatic Science and Engineering, South China University of Technology, Guangzhou 510641, China;

2. College of Software, South China University of Technology, Guangzhou 510641, China)

Abstract: This paper presents a novel teaching-learning-based optimization(TLBO) algorithm with group collaboration for the job shop scheduling problem(JSSP). Firstly, a collaborative learning strategy between groups is introduced to avoid being trapped into local optimum when solving the JSSP with high complexity. After learning within the group, individuals would communicate and collaborate with individuals of other groups so that they can jump out of the limitation of the current learning process. Then, in order to balance the local and global searching ability, a depth or width searching strategy is introduced, which enables individuals to learn in different ways according to their learning ability. For individuals with stronger learning ability, the depth learning will be adopted, and for individuals with weaker learning ability, the width learning will be adopted, so that different individuals can play different roles in the learning process. Experimental results on benchmark instances of OR-Library show that the search ability and convergence accuracy can be effectively improved in solving the JSSP.

Keywords: group learning; TLBO; coevolution; JSSP

0 引言

车间作业调度问题(JSSP)是制造业中十分重要的实际应用问题, 已被证明是一种NP难题^[1]. 早期的调度算法致力于寻求问题的精确解, 如整数规划法和分支定界法等, 但当问题规模增大或更复杂时, 使用传统方法在有限时间内很难得到最优解. 因此, 很多研究人员利用近似算法在可以接受的时间内寻求问题的满意解. 其中遗传算法^[2](GA)、禁忌搜索

算法^[3](TS)、模拟退火方法^[4](SA)等已被成功应用于求解车间调度问题. 近年来, Rao等^[5-6]提出了一种基于课堂教学效果的新型优化方法TLBO(Teaching-learning based optimization algorithm), 该方法模型简单, 计算效率较高, 得到了研究者的青睐, 被应用于各类优化问题上, 如连续大规模非线性优化问题^[7]、平面钢框架的设计^[8]、多目标热交换器参数设置^[9]、现代加工过程中的参数^[10]、最优功率流^[11]以及JSSP的

收稿日期: 2017-04-21; 修回日期: 2017-08-01.

基金项目: 广州市科技计划项目(201707010437); 中央高校基本科研业务费专项资金项目(2015zz100); 广东省科技计划项目(2014A010104004).

责任编辑: 刘德荣.

作者简介: 张梅(1977—), 女, 副教授, 博士, 从事智能优化算法等研究; 杨晟轩(1993—), 男, 硕士生, 从事智能优化算法的研究.

[†]通讯作者. E-mail: zhangmei@scut.edu.cn.

调度问题^[12]优化等,并取得了良好的效果。

针对TLBO算法的不足,很多研究者提出了改进的方法,主要有以下几方面. 1) 增加新的学习阶段: Yu等^[13]提出了一种精英教学优化算法,保留每代的最优解,用精英解替换较差个体,并在下一次迭代开始前随机对精英个体进行变异操作;张凌波等^[14]针对基础教学算法全局搜索能力差的问题提出了一种基于改进的教学算法,在教师阶段前后增加了预习和复习过程,并在学生阶段采用量子方式进行更新;于坤杰等^[15]在精英教学算法的基础上,在学生阶段后加入反馈阶段,使较差的学生能快速向教师靠拢. 2) 对教师阶段进行改进: 文献[9,16-17]在教师阶段采用了自适应的教学因子. 3) 对学生学习阶段进行改进: 王培崇^[18]为了克服教学优化算法容易早熟的问题,提出了具有自学机制和退火选择的教学优化算法;张梅等^[19]针对教学算法局部搜索能力不高的缺陷,提出了一种基于个体差异化自学习的改进教学(DL-TLBO)算法。

在现有的各类改进教学算法中,大都是基于单一的教师和学生群体的教与学过程的改进. 但在实际教学过程中,为了获得更好的学习效果,教师可以将学生分组,教师带领小组进行学习,提高小组成员的总体水平. 当各学习小组学习成果持续得不到改进时,小组间可以交流经验,从而获得更好的思想和学习成果. 另外,在小组学习的过程中,优等生一般具有更强的深度挖掘能力,而能力差的学生更适合进行学习的广度搜索以寻找更好的学习方向. 基于此,本文提出一种多小组协同学习的教学算法,实现小组间学习的协同及基于学习能力的深度和广度搜索策略,并将该方法应用到JSSP调度优化上以验证其有效性。

1 车间作业调度问题描述

作业车间调度问题^[1]可以描述如下: 给定一组工件 $J = 1, 2, \dots, n$, 要求在一组机器 $M = 1, 2, \dots, m$ 上加工完成, 要满足以下约束条件: 1) 每个工件在机器上的加工次序给定; 2) 每台机器在任何时刻最多只能加工一个工件, 工序加工的时间是固定的, 且工序一旦开始便不能被中断; 3) 每道工序必须等到其所有的前继工序加工完毕之后才能开始加工. JSSP问题的常用数学描述^[1]如下:

$$f = \min\{\max(c_{ik})\}. \quad (1)$$

$$\begin{aligned} \text{s.t. } & c_{ik} - p_{ik} + M(1 - a_{ihk}) \geq c_{ih}, \\ & i = 1, 2, \dots, n, h, k = 1, 2, \dots, m; \\ & c_{jk} - c_{ik} + M(1 - x_{ijk}) \geq p_{jk}, \end{aligned} \quad (2)$$

$$i, j = 1, 2, \dots, n, k = 1, 2, \dots, m; \quad (3)$$

$$c_{ik} \geq 0, i = 1, 2, \dots, n, k = 1, 2, \dots, m. \quad (4)$$

其中: c_{ik} 和 p_{ik} 分别为工件 i 在机器 k 上的完成时间和加工时间; M 为一个足够大的正数; a_{ihk} 和 x_{ijk} 分别为指示系数和指示变量, 其意义如下:

$$a_{ihk} = \begin{cases} 1, & \text{机器 } h \text{ 先于机器 } k \text{ 加工工件 } i; \\ 0, & \text{非上述情况.} \end{cases} \quad (5)$$

$$x_{ijk} = \begin{cases} 1, & \text{工件 } i \text{ 先于工件 } j \text{ 在机器 } k \text{ 上加工;} \\ 0, & \text{非上述情况.} \end{cases} \quad (6)$$

式(1)表示目标函数,即最大完工时间;式(2)表示链式约束条件;式(3)表示加工工件过程不可中断约束;式(4)表示工件在每个机器上都要加工。

2 基本教学算法

教学算法是模拟教师对学生的课堂教学所提出的一种新型算法,教学算法也是基于群体智能进化的算法,教师和学生群体相当于进化算法中的种群. 每代中最优的个体作为教师,其余个体作为学生. 教学算法包含两个学习阶段,分别为教师阶段和学生阶段,教师阶段为学生向老师学习的阶段,学生阶段为学生间相互学习的阶段^[5]。

2.1 教师阶段

每一代中最优的个体作为教师,负责引领种群进行进化. 教师会传授自己的知识给学生,并且能引导每个学生达到自己的目标,使整个群体向最优解靠近. 用下式描述种群与教师之间的差距:

$$\text{Difference_Means}_i = r_i(M_{\text{new}} - T_F M_i). \quad (7)$$

其中: $T_F = [1 + \text{round}(0, 1)]$ 表示教学因子; r_i 表示 $[0, 1]$ 间的随机数; M_i 表示第 i 代的平均水平, M_{new} 表示期望的下一代的平均水平,一般取当前的最优个体. 在教师阶段,每个学生根据 $\text{Difference_Means}_i$ 按照下式进行学习:

$$X_{\text{new},j} = X_{\text{old},j} + \text{Difference_Means}_i. \quad (8)$$

其中: $X_{\text{old},j}$, $X_{\text{new},j}$ 表示更新前后的第 i 代的第 j 个个体,只有当学生的水平有所提高时,即适应值更优时,当前的学习结果才会被接受。

2.2 学生阶段

学生除了会向老师学习以外,还会相互学习交流获得新知识,取长补短,从而相互影响共同进步. 在学生阶段,学生个体 X_i 向 X_j 的学习过程按下式^[5]进行:

$$X_{new,i} = X_{old,i} + r_i(X_i - X_j), F(X_i) \leq F(X_j); \tag{9}$$

$$X_{new,i} = X_{old,i} + r_i(X_j - X_i), F(X_i) > F(X_j). \tag{10}$$

其中: $F(X_i)$ 表示个体 X_i 的适应值, 为 $[0,1]$ 间的随机数. 当 $F(X_{new,i}) < F(X_{old,i})$ 时, 接受当前的学习过程, 保留个体 $X_{new,i}$, 否则保留 $X_{old,i}$.

在教学算法中, 单一的教与学可能会使求解过程陷入局部最优. 在实际的学习过程中, 可以将学生进行分组学习. 当各学习小组学习得不到改进时, 将进行小组间交流, 从而补充各小组缺乏的新知识, 帮助跳出思维的局限性. 另外, 在小组学习的过程中, 考虑到优等生具有更强的对问题的深度挖掘能力, 而能力差的学生可进行学习的广度搜索以寻找比较擅长的领域. 因此, 为了提高搜索精度, 本文提出一种新颖的多小组协同学习的教学算法.

3 基于多小组协同学习的教学算法

3.1 改进的TLBO算法

基于多小组协同学习的改进教学算法的基本流程如图1所示. 首先根据问题的特点设计编码方式, 将学生分为 M 个容量为 N 的学习小组(子种群), 同时对这 M 个小组进行随机初始化, 每个小组独立进化迭代. 教师阶段仍按照式(7)、(8)进行更新. 学生阶段将改进为两个阶段, 首先是学生相互学习阶段, 按式(9)、(10)进行更新; 然后是按照新提出的基于深度和广度的方式进行学生学习, 按照适应值高低将学生分为优等生和差生, 从而采取不同的学习方式; 最后, 当某小组学习能力得不到改善时, 进行基于最优小组的协同学习, 实现各小组间的交流. 算法的终止条件为当前迭代代数达到设定值 T . 具体的过程在后面的小节详细介绍.

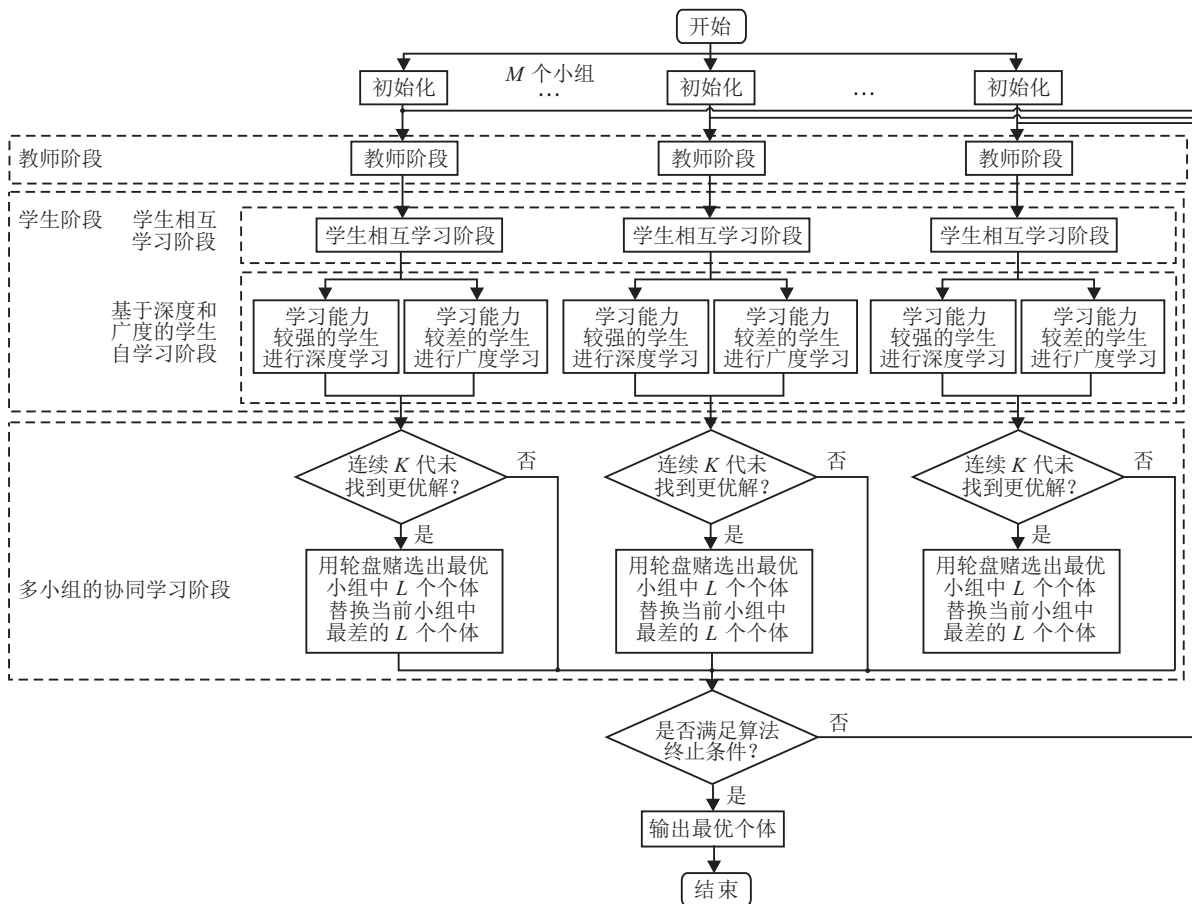


图1 算法流程

3.2 编码方式

在进化算法中, 每个个体即为问题的一个解, 对于本文要解决的JSSP问题, 为避免算法出现不可行解, 采用基于工序的编码方式进行编码. 染色体的基因数即为工序的总数, 即每个个体的染色体由 $n \times m$ 个基因组成(其中 n 表示工件总数, m 表示每个工件

的工序数), 基因表示工件号, 每个工件的工件号出现的次数等于它的工序数, 染色体基因的顺序决定了工件的调度顺序. 解码时从左向右进行解码, 第 k 次出现的同一工件序号, 表示该工件的第 k 个工序, 如编码为 $(1, 1, 3, 3, 2, 1, 3, 2, 2, 3, 1, 2)$ 的染色体, 出现的第1个“1”表示工件1的第1个工序, 出现的第2个“1”

表示工件1的第2个工序,以此类推.

3.3 教师阶段

每个子种群中每一代最优的个体作为组长,负责引领种群进行进化.由式(7)可知,需求取 M_{new} 与 M_i 的差值,而本文中对于离散问题的求差,采用基于位置的交叉算子^[20](PBX)和基于顺序的交叉算子^[20](OX)实现个体间的求差,求差时随机选取两种交叉算子以扩大解的搜索空间^[19].式(7)中的 M_i 一般取种群中适应值大小排序为中间位置的个体^[6],而文献[19]的研究发现,相比于选取种群中适应值靠近中间个体作为 M_i 的取值方式,随机选取个体作为 M_i 能更好地保证教师阶段个体更新的种群多样性,避免过早收敛,本文采用文献[19]的方法确定 M_i 的值.根据式(7)求出 $Difference_Means_i$ 之后,再根据式(8)对 $X_{old,i}$ 进行更新,只有当得到的 $X_{new,i}$ 比 $X_{old,i}$ 更优时,新个体才会被保留,否则保留老的个体.

3.4 改进的学生阶段

在学生阶段,除了考虑所有学生之间的相互学习和影响以外,还应考虑具有不同学习能力的学生对知识的拓展和挖掘能力的不同.因此,在本算法的学生阶段,学生将分为两个阶段进行学习,一是学生相互学习,二是根据学生能力进行基于深度和广度的学习.

3.4.1 学生相互学习

在学习过程中,每个学生都具有相互学习的机会,学生相互学习过程按照式(9)、(10)进行.对于种群中每个个体 X_i ,随机选取种群中另一个个体 X_j ,采用PBX或OX算子更新得到新个体 $X_{new,i}$,只有当 $X_{new,i}$ 比 X_i 更优时,才接受 $X_{new,i}$ 替换 X_i .

3.4.2 基于深度和广度的学生自学习

在连续问题中,每个个体会依照自身的适应值按照不同的步长进行更新:越优的个体步长越短,以提升算法的局部搜索能力;越差的个体步长越长,以扩大搜索范围并提升算法的全局搜索能力^[21].基于此思想,在本文涉及的离散问题中,将小组中学生按照适应值(完工时间)由小到大进行排名,将排名在前半部分的学生定义为学习能力较强的学生,将排名在后半部分的学生定义为学习能力较差的学生.对于学习能力较强的学生,进行加强局部搜索能力的深度学习操作,而对于学习能力较弱的学生,则进行加强随机搜索能力的广度学习操作.

1) 学生深度学习阶段.对于学习能力较好的学生,其更善于进行知识的深度挖掘.因此对学习能力较强的个体 X_i 进行 $Num(i)$ 次自学习算子操作,根据

文献[19]中给予学习能力强的个体更多的学习机会和次数的思想, $Num(i)$ 按照下式计算获得:

$$Num(i) = \text{round}\left(\frac{F_{\max} - F(i)}{F_{\max} - F_{\min}}(S_{\max} - S_{\min}) + S_{\min}\right). \quad (11)$$

其中: $F(i)$ 为第 i 个个体的适应值, $i = 1, 2, \dots, N$; $F_{\min} = \min(F(i))$; $F_{\max} = \max(F(i))$; S_{\min}, S_{\max} 分别为设定的学习次数下限和上限,考虑到运行的效率和参考文献[19],取 $S_{\min} = 1, S_{\max} = 15$;自学习算子为文献[22]提到的3种经典变异算子:单点交叉算子、倒位算子和移位算子,如图2所示.

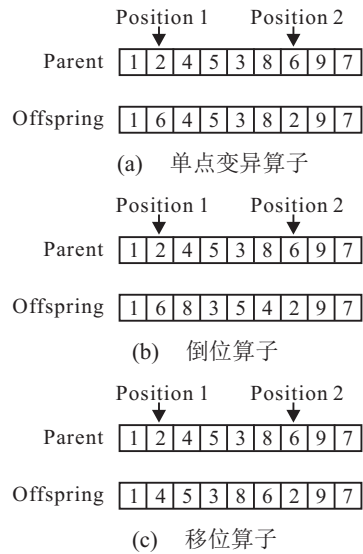


图2 3种自学习算子

每次自学习随机选择3种自学习算子中的一种,若本次学习得到的新个体比当前个体 X_i 更优,则对当前个体 X_i 进行更新,否则本次的学习成果不被接受,个体 X_i 不更新,以此类推,个体共完成 $Num(i)$ 次学习,学生的深度学习过程如图3所示.

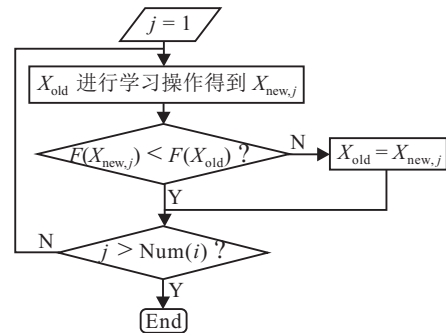


图3 学生的深度学习过程

与文献[19]中以当前个体为起点一次性产生多个邻域个体然后取最好个体的差异性自学习不同,本文的学生深度学习更注重学习的循序渐进,每次学习都是在上次学习的基础上一步一步向更好的结果迈进,可以使个体到达更大范围的邻域之外,因此能

得到更深层次的挖掘效果,提高算法的局部搜索能力.此处改进的效果将在实验部分得到验证.

2) 学生广度学习阶段.对于学习能力较差的学生,对知识的深度挖掘能力有限,应进行更多的广度搜索,尽量能够探寻到新的知识领域.当学习能力比较差的学生找到自己比较擅长的领域时,就有机会成为学习能力好的学生,然后进行深度学习,从而提高整个群体的学习能力.广度学习的具体过程为对学习能力较差的个体 X_i ,随机选择一个比自己更优(排名比自己靠前)的个体 X_j 进行学习.学习过程如下:

$$X_i = X_i + r_i(X_j - X_i). \quad (12)$$

与式(10)不同之处在于,采用式(12)得到的新个体将被无条件接受,而采用式(10)的择优保持策略可能会拒绝掉很多有潜力的个体.通过该操作,可以扩大算法的搜索空间,增强算法的全局搜索能力.

3.5 基于最优小组的协同学习阶段

为了得到更好的学习效果,应该在小组自身学习能力不能得到进一步改善时进行小组间协同学习.为提高交流的效率,本文的多小组的协同学习方式设定为当某个小组在连续 K 代内都没找到更优解时,与其他小组中最优的小组(平均适应值最优)进行交流,从其他小组最优的小组中采用轮盘赌的方式^[23]选取 L 名学生,替换掉当前小组中成绩最差的 L 名学生.在迭代的初期,为了充分发挥个体不同的创造力,提高算法的搜索能力,应该减少参与协同学习交流的学生人数 L .随着学习进程的不断推进,小组内个体变得更好,各个小组都得到了自己的学习心得,应该增加小组间协同学习的强度,即增加参与协同学习交流的学生人数 L ,进一步提高种群的整体水平.小组中参与协同人数 L 利用下式确定:

$$L = G/P. \quad (13)$$

其中: G 为当前迭代代数, P 为一个正常数.

本文采用轮盘赌方式进行参加协同学习的学生选取,即

$$P(X_i) = \frac{F(X_i)}{\sum_{j=1}^N F(X_j)}. \quad (14)$$

每个学生被选取的概率与其学习能力有关.协同学习的策略保证了子种群的多样性,可以提高算法的全局搜索能力.

3.6 算法复杂度分析

算法的步骤包括种群初始化、教师阶段、学生阶段(包括学生相互学习阶段、学生深度学习和广度学习阶段)、协同学习阶段.对于 M 个种群大小为 N 的

学习小组,定义问题规模为 $n = n_p \times m$ (n_p 为工件总数, m 为机器总数).初始化阶段的时间复杂度为 $O(N \times M \times n)$;PBX和OX算子的时间复杂度为 $O(n)$,故教师阶段的时间复杂度为 $O(N \times M \times n)$;单点变异算子的时间复杂度为 $O(1)$,倒位算子和移位算子的时间复杂度为 $O(n)$,故学生阶段的时间复杂度为 $O(S \times N \times M \times n)$,其中 S 为学习次数,而文中 S 限定在一定常数范围内变化,故可以简化为 $O(N \times M \times n)$;协同学习阶段的时间复杂度为 $O(N \times M \times n)$.因此,可以得出算法迭代一次的时间复杂度为 $O(N \times M \times n)$,而在本文中, M 取3,故算法迭代一次的时间复杂度可以简化为 $O(N \times n)$,可以看出算法的复杂度不高.

4 仿真结果及分析

为验证所提算法的正确性及可行性,本文选取OR-Library中43个具有代表性的标准JSSP问题进行仿真.其规模大小包括 $6 \times 6, 10 \times 5, 10 \times 10, 15 \times 5, 15 \times 10, 15 \times 15, 20 \times 5, 20 \times 10, 20 \times 20, 30 \times 10$.仿真结果中:WKS(Well known solution)表示已知的最优解;Best表示当前找到的最好解,反映算法的全局搜索能力;Avg表示算法重复运行所能求得最好解的平均值,反映算法解的精度;Std表示算法重复运行所能求得解的标准差,反映算法的鲁棒性.本文算法采用C++编写,运行在2.2 GHz CPU, 4 Gb内存的PC机上.

4.1 算法参数设定

为确定种群大小 N 和迭代次数Gen,采用一个规模适中的测试实例ft20(5×20)对 N 、Gen进行测试,在此将学习小组个数设定为3组,通过实验,取 $K = 20, P = 50$.采用文献[12]中的测试方法进行测试,分别取不同的种群大小和迭代次数进行组合,组合列表如表1所示.

表1 种群大小和迭代次数的测试组合

编号	种群大小 N	迭代次数 Gen
1	50	1000
2	50	1500
3	50	2000
4	50	2500
5	100	1000
6	100	1500
7	100	2000
8	100	2500
9	150	1000
10	150	1500
11	150	2000
12	150	2500
13	200	1000
14	200	1500
15	200	2000
16	200	2500

算法在每种组合下独立运行20次,取种群大小相同而迭代次数不同时的算法运行结果的均值作为该种群大小下的算法结果,取迭代次数相同而种群大小不同时的算法运行结果的均值作为该迭代次数下的算法结果,如图4所示。

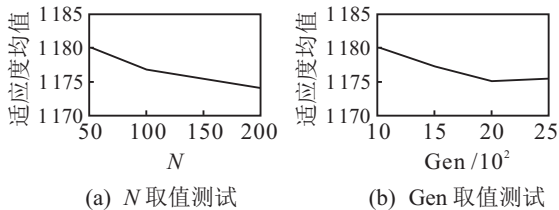


图4 种群大小N和迭代次数Gen测试

如图4(a)所示,当N增大时,算法的精度会变得更优,但通过实验可发现,N = 100时的结果与N取

更大值时所获得的结果差距不大,且因考虑到算法的时间成本问题,本文取N = 100;如图4(b)所示,当Gen增大时,算法的精度也更优,但当Gen增大到2000以后趋于稳定,因此本文取Gen= 2000。

4.2 学习策略的实验对比

为验证本文所提出的基于多小组协同学习的改进教学算法的有效性,尤其是本文所提出的深度和广度的学生学习策略的有效性,分别采用如下3种方式对OR-Library中的不同规模的部分标准实例进行优化测试:优差生分别采用深度和广度学习策略;不考虑优差生区别,所有学生均采用本文所提的深度学习策略;所有学生按照学习能力的强弱利用文献[19]的差异性自学习策略。算法在每种策略下独立运行20次,测试结果如表2所示。

表2 采用不同的学习策略对比

Problem	Size	WKS	深度和广度学习策略			仅深度学习策略			文献[19]的学习策略		
			Optimal	Avg.	Std.	Optimal	Avg.	Std.	Optimal	Avg.	Std.
ft06	6×6	55	55	55	0	55	55	0	55	55	0
ft10	10×10	930	930	946.1	6.66	930	945.35	8.94	930	945.35	8.13
ft20	20×5	1165	1165	1175.65	6.03	1165	1177.65	3.62	1165	1177.9	5.16
la01	10×5	666	666	666	0	666	666	0	666	666	0
la02	10×5	655	655	655	0	655	655	0	655	655	0
la06	15×5	926	926	926	0	926	926	0	926	926	0
la07	15×5	890	890	890	0	890	890	0	890	890	0
la11	20×5	1222	1222	1222	0	1222	1222	0	1222	1222	0
la12	20×5	1039	1039	1039	0	1039	1039	0	1039	1039	0
la16	10×10	945	945	947.45	3.71	945	947.4	3.73	945	948.95	4.76
la17	10×10	784	784	784	0	784	784	0	784	784	0
la21	15×10	1046	1050	1065.95	4.2	1058	1066.15	4.48	1063	1069.75	5.3
la22	15×10	927	930	936.55	4.97	930	942.7	4.85	930	941.6	5.38
la26	20×10	1218	1218	1218	0	1218	1218.35	1.57	1218	1219.5	3.36
la27	20×10	1235	1255	1267.2	4.03	1256	1268.9	5.2	1264	1279.15	9.98
la31	30×10	1784	1784	1784	0	1784	1784	0	1784	1784	0
la32	30×10	1850	1850	1850	0	1850	1850	0	1850	1850	0
la36	15×15	1268	1278	1295.95	8.83	1278	1301.4	10.12	1296	1316.4	14.73
la37	15×15	1397	1410	1431.4	11.78	1415	1434.9	14.8	1428	1449.5	14.14

由学生学习阶段利用基于深度和广度的学习策略与仅利用深度学习、仅利用文献[19]的基于差异性自学习的实验对比可知,对于不同规模的问题,采用仅深度学习策略得到的最优解及平均解均优于或等于采用文献[19]的自学习策略得到的解,说明了深度学习的有效性。另外,通过深度和广度学习与仅进行深度学习的结果可以看出,在问题规模较小时,两种策略得到的平均解和最优解均相差不大,而当问题规模变大时,深度和广度学习策略得到的平均解和最优解均优于仅采用深度学习策略得到的解,说明采用本文所提到的基于深度和广度学习的学生阶段学习策略可更好地解决大规模的优化问题。且除问题ft10和la22以外,采用深度和广度学习策略得到的解的标准差也均小于或等于另外两种学习策略的标准差,可

见本文提出的深度和广度学习策略具有更强的鲁棒性,提升了算法的稳定性。

为进一步说明学习策略的有效性,选用规模不同的实例ft10、ft20、la21、la36,对3种策略中的最优解的迭代曲线进行对比,如图5~图8所示。

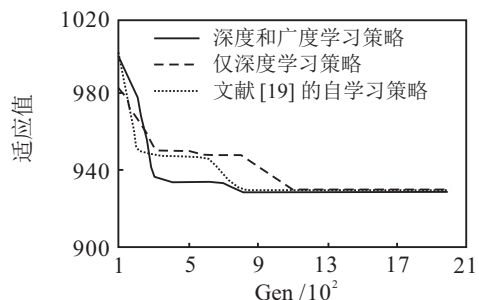


图5 实例ft10下的对比

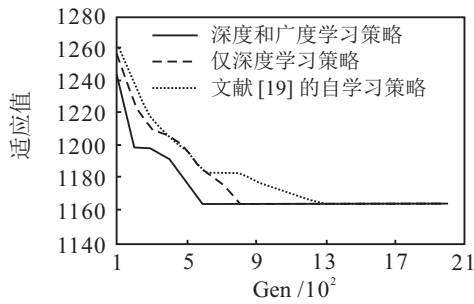


图6 实例ft20下的对比

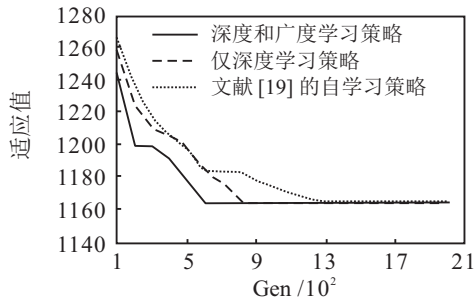


图7 实例la21下的对比

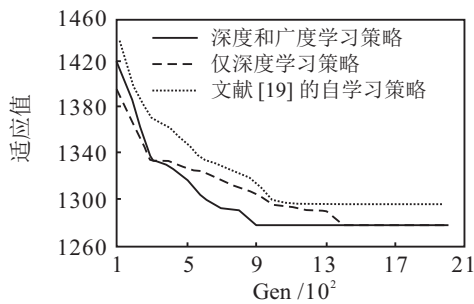


图8 实例la36下的对比

由图5~图8可见:除实例ft10外,仅深度学习策略的收敛速度快于文献[19]的自学习策略;且在这4个实例上,深度和广度学习策略的收敛速度均快于另外两种方法。

4.3 与其他算法的比较

为验证本文所提算法的有效性,将其与文献[12]中基本TLBO算法、文献[19]的改进教学DL-TLBO算法、文献[22]的多种群协同进化的岛群模型遗传算法(NIMGA)的最好解和平均解进行比较.进行对比的各算法参数设定如下:基本TLBO算法中种群大小为100,迭代次数为1000;改进教学DL-TLBO算法中种群大小为100,迭代次数为2000;NIMGA算法中子种群个数为3,子种群大小为100,子种群内循环次数为1000,外循环次数为1000;本文所提算法中学习小组个数为3,组内个体为100,迭代次数为2000.为进一步评价算法的整体求解能力,引入文献[22]中提到的以下性能指标:RPD-B= $\sum(\text{Best-WKS})/\text{WKS}$ 表示最好解相对于最优解的偏差之和;RPD-A= $\sum(\text{Avg-WKS})/\text{WKS}$

表示平均值相对于最优解的偏差和.本文算法与其他算法的对比如表3和表4所示,其中NIS表示各算法在文献中被测试的实例个数,OA表示其他算法,Improvement表示在当前NIS个实例下本文算法相比于OA的RPD-B和RPD-A的提升.

由表3可见,本文算法能找到其中31个实例的已知最优解,占总测试实例的72%,且所有实例中找到的平均解和最优解都优于或等于其他3种算法.表4的数据表明:利用本文所提方法和文献[12]中的TLBO方法对20个共同的实例进行求解,其最好解相对于最优解的偏差之和的值RPD-B被改善了76.43%,而平均值相对于最优解的偏差之和的RPD-A被改善了88.36%;本文所提方法和文献[19]中的DL-TLBO方法对40个共同的实例进行求解,其最好解相对于最优解的偏差之和的值RPD-B被改善了59.89%,而平均值相对于最优解的偏差之和的RPD-A被改善了49.55%;本文所提方法和文献[22]中的NIMGA方法对43个共同的实例进行求解,其最好解相对于最优解的偏差之和的值RPD-B被改善了59.37%,而平均值相对于最优解的偏差之和的RPD-A被改善了52.93%.因此,由表3和表4可知,本文所提方法对所有实例的整体求解的精度相比其他方法均有较大提升和改善.

为进一步比较算法的差异性,用双侧 t 检验法将本文算法的结果与其他文献中算法所得结果进行显著性检验.其中: S_1^2 和 S_2^2 为两样本方差, n_1 和 n_2 为两样本容量,以每个实例的平均解与WKS的差作为一个样本,与参考文献中相同实例的数据进行显著性检验.与文献[12]进行比较, $n_1 = n_2 = 20$,检验自由度 $d_f = n_1 - 1 + n_2 - 1 = 38$,经计算可得 $|t| = 3.146$,取显著性水平 $\alpha = 0.01$,查表可得 $t_{0.01/2,38} = 2.712$, $|t| > t_{0.01/2,38}$,因此两种算法的运算结果存在非常显著的差异;与文献[19]进行比较, $n_1 = n_2 = 40$,检验自由度 $d_f = n_1 - 1 + n_2 - 1 = 78$,经计算可得 $|t| = 1.784$,取显著性水平 $\alpha = 0.1$,查表可得 $t_{0.1/2,78} = 1.665$, $|t| > t_{0.1/2,78}$,因此两种算法的运算结果存在一般的显著的差异;与文献[22]进行比较, $n_1 = n_2 = 43$,检验自由度 $d_f = n_1 - 1 + n_2 - 1 = 84$,经计算可得 $|t| = 2.140$,取显著性水平 $\alpha = 0.05$,查表可得 $t_{0.05/2,84} = 1.989$, $|t| > t_{0.05/2,84}$,因此两种算法的运算结果存在显著的差异.综合以上测试结果可知,本文提出的算法对于JSSP具有较好的寻优精度和稳定性.

表3 与其他算法的对比

Problem	Size	WKS	本文算法		TLBO ^[12] (2014)		DL-TLBO ^[19] (2016)		NIMGA ^[22] (2016)	
			Optimal	Avg.	Optimal	Avg.	Optimal	Avg.	Optimal	Avg.
ft06	6×6	55	55	55	55	56.5	55	55	55	55
ft10	10×10	930	930	946.1	938	967.3	930	949.9	930	952.33
ft20	20×5	1165	1165	1175.65	1165	1199.2	1174	1182.3	1173	1187.2
la01	10×5	666	666	666	666	676.8	666	666	666	666
la02	10×5	655	655	655	655	687.8	655	655	655	656.53
la03	10×5	597	597	597	597	638.5	597	597	597	597.8
la04	10×5	590	590	590	607	619.6	590	590.3	590	590
la05	10×5	593	593	593	593	592.1	593	593	593	593
la06	15×5	926	926	926	926	931.1	926	926	926	926
la07	15×5	890	890	890	890	903.7	890	890	890	890
la08	15×5	863	863	863	864	896.4	863	863	863	863
la09	15×5	951	951	951	951	955.5	951	951	951	951
la10	15×5	958	958	958	958	960.2	958	958	958	958
la11	20×5	1222	1222	1222	1222	1248	1222	1222	1222	1222
la12	20×5	1039	1039	1039	—	—	1039	1039	1039	1039
la13	20×5	1150	1150	1150	—	—	1150	1150	1150	1150
la14	20×5	1292	1292	1292	—	—	1292	1292	1292	1292
la15	20×5	1207	1207	1207	—	—	1207	1207	1207	1207
la16	10×10	945	945	947.45	946	956.1	950	950	946	956.07
la17	10×10	784	784	784.2	—	—	784	784	784	784.4
ka18	10×10	848	848	849.3	—	—	848	853.45	848	854.07
la19	10×10	842	842	848.8	—	—	842	847.75	842	847.67
la20	10×10	902	902	904.5	—	—	902	905	907	909.47
la21	15×10	1046	1050	1065.95	1091	1126.1	1066	1082.1	1058	1087.5
la22	15×10	927	930	936.55	—	—	930	943.45	937	952.67
la23	15×10	1032	1032	1032	—	—	1032	1032	1032	1032.3
la24	15×10	935	941	945.2	—	—	945	958.65	947	969.33
la25	15×10	977	984	994.9	—	—	989	1011.9	992	1012.3
la26	20×10	1218	1218	1218	—	—	1221	1238.8	1218	1239.5
la27	20×10	1235	1255	1267.2	1256	1488.8	1277	1299.3	1269	1294.7
la28	20×10	1216	1218	1232.45	—	—	1251	1261.2	1247	1263
la29	20×10	1152	1189	1206.5	—	—	1230	1247.8	1241	1262.3
la30	20×10	1255	1355	1355	—	—	1357	1378	1355	1357.8
la31	30×10	1784	1784	1784	1784	1991.3	1784	1784	1784	1784
la32	30×10	1850	1850	1850	—	—	1850	1850	1850	1850
la33	30×10	1719	1719	1719	—	—	1719	1719	1719	1719
la34	30×10	1721	1721	1721	—	—	1721	1729.8	1721	1723.5
la35	30×10	1888	1888	1888	—	—	1888	1888	1888	1888.7
la36	15×15	1268	1278	1295.95	1322	1405.7	1297	1320.3	1293	1317.7
la37	15×15	1397	1410	1431.4	—	—	—	—	1439	1464
la38	15×15	1196	1211	1223.9	—	—	—	—	1222	1261.7
la39	15×15	1233	1247	1254.7	—	—	—	—	1259	1280.7
la40	15×15	1222	1224	1240.9	1241	1385.9	1243	1268.8	1246	1266.3

表4 与其他算法的整体求解能力对比

Algorithm	NIS	RPD-B		Improvement / %	RPD-A		Improvement / %
		OA	本文算法		OA	本文算法	
TLBO	20	0.1578	0.0372	76.43	0.9597	0.1117	88.36
DL-TLBO	40	0.2276	0.0913	59.89	0.4630	0.2336	49.55
NIMGA	43	0.3064	0.1245	59.37	0.6423	0.3023	52.93

5 结论

本文提出了一种新颖的多小组协同学习的教学算法,并对车间作业调度问题进行了研究.针对教学算法容易陷入局部最优的缺点,将学生分成小组进行学习.同时在算法的学生学习阶段加入学生的深度和广度学习策略,对优等生采用提高算法局部搜索能

力的深度学习策略,对差生采用提升随机搜索能力的广度搜索策略以提高算法的全局搜索能力.并引入基于最优小组的协同学习策略,当各学习小组学习效果持续得不到改进时,将进行小组间交流,从而帮助算法跳出局部最优.最后通过对OR-Library中的实例进行测试验证了本文策略的有效性,并与其他文献

中的方法进行比较,实验结果验证了本文算法在求解JSSP上的有效性。

参考文献(References)

- [1] Amirghasemi M, Zamani R. An effective asexual genetic algorithm for solving the job shop scheduling problem[C]. *Computers & Industrial Engineering*. Amsterdam: Pergamon Press, 2015: 123-138.
- [2] Deng Y, Liu Y, Zhou D. An improved genetic algorithm with initial population strategy for symmetric TSP[J]. *Mathematical Problems in Engineering*, 2015, 2015(3): 1-6.
- [3] Grabowski J, Wodecki M. A very fast tabu search algorithm for job shop problem[C]. *Metaheuristic Optimization via Memory and Evolution*. New York: Springer, 2005: 117-144.
- [4] Bandyopadhyay S, Saha S, Maulik U, et al. A simulated annealing-based multiobjective optimization Algorithm: AMOSA[J]. *IEEE Trans on Evolutionary Computation*, 2008, 12(3): 269-283.
- [5] 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.
- [6] Keesari H S, Rao R V. Optimization of job shop scheduling problems using teaching-learning-based optimization algorithm[J]. *Opsearch*, 2014, 51(4): 545-561.
- [7] Rao R V, Savsani V J, Vakharia D P. Teaching-learning-based optimization: An optimization method for continuous non-linear large scale problems[J]. *Information Sciences*, 2012, 183(1): 1-15.
- [8] Toan V. Design of planar steel frames using teaching-learning based optimization[J]. *Engineering Structures*, 2012, 34(1): 225-232.
- [9] Rao R V, Patel V. Multi-objective optimization of heat exchangers using a modified teaching-learning-based optimization algorithm[J]. *Applied Mathematical Modeling*, 2013, 37(3): 1147-1162.
- [10] Rao R V, Kalyankar V D. Parameter optimization of modern machining processes using teaching-learning-based optimization algorithm[J]. *Engineering Applications of Artificial Intelligence*, 2013, 26(1): 524-531.
- [11] Boucekara H R E H, Abido M A, Boucherma M. Optimal power flow using teaching-learning-based optimization technique[J]. *Electric Power System Sreseapch*, 2014, 114(3): 49-59.
- [12] Baykasolu A, Hamzadayi A, Köse S Y. Testing the performance of teaching-learning based optimization (TLBO) algorithm on combinatorial problems: Flow shop and job shop scheduling cases[J]. *Information Sciences*, 2014, 276(C): 204-218.
- [13] Yu Kunjie, Wang Xin, Wang Zhenlei. Study and application of improved teaching-learning-based optimization algorithm[J]. *Chemical Industry and Engineering Progree*, 2014, 33(4): 850-854.
- [14] 张凌波, 宰娜, 顾幸生. 基于改进教学算法优化BP神经网络的催化剂碳含量预测模型[J]. *控制与决策*, 2016, 31(9): 1723-1728.
(Zhang L B, Zai N, Gu X S. Predictive model for catalyst carbon content based on MTLBO-BP[J]. *Control and Decision*, 2016, 31(9): 1723-1728.)
- [15] 于坤杰, 王昕, 王振雷. 基于反馈的精英教学优化算法[J]. *自动化学报*, 2014, 40(9): 1976-1983.
(Yu K J, Wang X, Wang Z L. Elitist teaching-learning-based optimization algorithm based on feedback[J]. *Acta Automatica Sinica*, 2014, 40(9): 1976-1983.)
- [16] Patel V K, Savsani V J. A multi-objective improved teaching-learning-based optimization algorithm(MO-ITLBO)[J]. *Information Sciences*, 2016, 357: 182-200.
- [17] Rao R V, Patel V. An improved teaching-learning-based optimization algorithm for solving unconstrained optimization problems[J]. *Scientia Iranica*, 2013, 20(3): 710-720.
- [18] 王培崇. 具有自学机制和退火选择的优化算法[J]. *计算机工程与科学*, 2016, 38(4): 706-712.
(Wang P C. An improved teaching learning based optimization with self-study and simulated anneal[J]. *Computer Engineering & Science*, 2016, 38(4): 706-712.)
- [19] 张梅, 吴凯华, 胡跃明. 基于改进教学算法的车间作业调度问题[J]. *控制与决策*, 2017, 32(2): 349-357.
(Zhang M, Wu K H, Hu Y M. An improved teaching-learning-based optimization algorithm for solving job shop scheduling problem[J]. *Control and Decision*, 2017, 32(2): 349-357.)
- [20] Kellegoz T, Toklu B, Wilson J. Comparing efficiencies of genetic crossover operators for one machine total weighted tardiness problem[J]. *Applied Mathematics & Computation*, 2008, 199(2): 590-598.
- [21] Tan Y, Zhu Y. Fireworks algorithm for optimization[C]. *Int Conf in Swarm Intelligence*. Berlin: Springer Berlin Heidelberg, 2010: 355-364.
- [22] Kurdi M. An effective new island model genetic algorithm for job shop scheduling problem[J]. *Computers & Operations Research*, 2016, 67: 132-142.
- [23] David W Coit. Genetic Algorithms and engineering design[J]. *The Engineering Economist*, 1997, 43(4): 379-381.

(责任编辑: 闫妍)