

基于改进教学算法的车间作业调度问题

张梅[†], 吴凯华, 胡跃明

(华南理工大学 a. 自动化科学与工程学院, b. 精密电子制造装备教育部工程研究中心, 广州 510641)

摘要: 为求解车间作业调度问题, 提出一种基于个体差异化自学习的改进教学算法. 针对教学算法局部搜索能力不高的缺陷, 提出学生不仅应向能力好的学习者学习, 亦应进行有差异的自我学习. 通过学习者的完工时间评估学生的学习能力, 提出学习次数概念, 并设计自学习算子, 完善学生阶段的更新, 提高算法的局部搜索能力. 最后, 对OR-Library中的标准仿真实例进行实验, 结果表明改进教学算法是有效的, 其在收敛精度和鲁棒性能上均有较好的提高.

关键词: 教学优化算法; 差异化学习; 车间作业调度

中图分类号: TP18

文献标志码: A

Improved teaching-learning-based optimization algorithm for solving job shop scheduling problem

ZHANG Mei[†], WU Kai-hua, HU Yue-ming

(a. College of Automatic Science and Engineering, b. Engineering Research Centre for Precision Electronic Manufacturing Equipments of Ministry, South China University of Technology, Guangzhou 510641, China)

Abstract: To solve the job shop scheduling problem, an improved teaching-learning-based optimization algorithm(TLBO) is proposed in this paper. Aiming at the weak local search ability in the existing TLBO, it is proposed that the learner should learn knowledge not only from the better learners, but also from itself, therefore, a differential self-learning operator is designed. The learning ability of learner is evaluated by its optimal, and its learning times are adaptive calculated according to its learning ability. The learners with higher learning ability have more chance to self-learn. Finally, the proposed method is applied to solve the benchmark instances in OR-Library. The experimental results show that the proposed algorithm is effective while solving the job shop scheduling problem, and its accuracy and robustness can be improved further.

Keywords: teaching-learning-based optimization algorithm; differential self-learning operator; job shop scheduling problem

0 引言

车间作业调度问题(JSSP)^[1]是制造业中十分重要的实际应用问题,是一种NP难题.当问题规模增大时,使用传统的确定性算法如分支定界法、整数规划法等,在有限时间内很难得到最优解.因此,在过去的30年内,很多专家学者对基于进化计算的方法,如遗传算法(GA)^[2]、模拟退火算法(SA)^[2]、人工免疫算法(AIS)^[3]、多智能体进化算法(MA)^[4]等进行了大量研究,将这些方法应用到调度问题中,并验证了这些方法的有效性.近几年,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,并被应用到各类优化问题上,如多目标热交换器参数设置^[7]的优化、连续非线性的大规模优化问题^[8]、复杂约束条件的优化问题^[9]、多源电力系统的负荷频率控制^[10]和各类参数优化问题^[11-12]上,并取得了良好的效果.另外,也有不少学者对该方法进行改进,并应用到调度问题的优化上.如王凌等^[13]在教学算法的每个搜索阶段加入了前向-反向改进策略,增强了算法的局部搜索

收稿日期: 2016-01-07; 修回日期: 2016-04-08.

基金项目: 中央高校基本科研业务费专项资金项目(2015zz100); 广州市科技重大专项计划-产学研专项项目(2012Y5-00004).

作者简介: 张梅(1977-),女,副教授,博士,从事调度优化、控制理论与控制工程研究; 胡跃明(1960-),男,教授,博士生导师,从事控制理论与控制工程、精密电子装备等研究.

[†]通讯作者. E-mail: zhangmei@scut.edu.cn

能力,并将其成功应用于多目标资源受限项目的调度问题中. Tansel^[14]将TLBO和禁忌搜索算法相结合,设计禁忌表,并将其应用到二次分配问题的求解过程中. Gonca等^[15]证明了TLBO算法在双边装配线平衡上具有高效的求解能力. Adil等^[16]利用教学算法,采用连续编码方式,并利用G&T算法进行解码,有效地解决了JSSP的调度优化问题.

在现有的各类改进教学算法中,学习过程一般包含两个阶段^[5]:一是学生向教师学习,二是学生间相互学习,通过不断的进化过程使学习个体朝最优方向发展.但在实际学习过程中,应具有学生向外界的学习过程,每位学生自我学习潜力的不同和自我学习过程也会影响最终的学习效果.所以在学习过程中,不仅需要考虑学生向老师学习和学生之间相互学习等外部学习,还应该考虑内部的自我学习.在此基础上,本文根据学生学习能力的强弱进行有差异的自我学习,提出了基于个体差异化自学习方式的改进教学算法(TLBO-DL),实现算法搜索能力的提高,并将该方法应用到JSSP调度优化上.最后,对OR-Library中的标准仿真实例进行实验,结果表明改进教学算法是有效的,其在收敛精度和鲁棒性能上均有较好的提高.

1 JSSP问题描述

车间作业调度问题^[1]包含 n 个工件,每个工件含有 m 个工序,目标函数是最小化最大完成时间(工期).每个机器在同一时刻最多只能加工一个工件,每道工序必须等到其所有前继工序加工完毕后才能开始加工,每个工序的加工过程不能中断.假设 $J = 1, 2, \dots, n$ 为工件集合, $M = 1, 2, \dots, m$ 为机器集合,其目标函数和约束条件^[1]为

$$f = \min\{\max\{c_{ik}\}\}(\forall i, k); \quad (1)$$

$$\text{s.t. } c_{ik} - t_{ik} + R(1 - a_{ihk}) \geq c_{ih}, \quad (2)$$

$$c_{jk} - c_{ik} + R(1 - x_{ijk}) \geq t_{jk}, \quad (3)$$

$$c_{ik} \geq 0. \quad (4)$$

其中: $i, j = 1, 2, \dots, n; h, k = 1, 2, \dots, m; R$ 为一个足够大的正数; c_{ik} 为第 i 个工件在第 k 台机器上加工的完成时间; t_{ik} 为第 i 个工件在第 k 台机器上加工的加工时间.且有

$$a_{ihk} = \begin{cases} 1, & \text{工件 } i \text{ 在机器 } h \text{ 上加工先于在 } k \text{ 上加工;} \\ 0, & \text{otherwise;} \end{cases} \quad (5)$$

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

式(1)为目标函数,式(2)为链式约束条件,式(3)为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 TLBO算法

教学算法是在教师对学生的课堂教学过程中,以达到最佳教学效果和提高班级学习质量为目的,而提取出的一种新型算法.在TLBO算法中,教师和学生是两个重要的部分,学生们不仅从老师的传授中获得知识,而且学生们内部相互学习,这是TLBO算法提升求解质量的主要方式. TLBO算法也是一种群体智能进化算法,在整个群体中,每一代的最优个体为Teacher,以此将整个群体学习分成教师阶段学习和学生阶段学习^[5].

2.1 教师阶段

教师是每一代种群中的最优个体,负责整个教学团队进化的方向.教师期望学生们能够达到教师的知识水平,引导整个群体的平均位置向最优解靠近.首先,由下式描述整个种群与教师之间的差距:

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

其中: $T_F = \text{round}[1 + \text{rand}(0, 1)]$ 为教学因子; r_i 为 $[0, 1]$ 之间的随机数; M_i 为第 i 代整个班级的平均水平; M_{new} 为下一代期望的种群平均水平,一般取当前种群中最优个体.

在教师阶段,每个学生根据Difference_Mean_{*i*}按照下式进行学习:

$$X_{\text{new},i} = X_{\text{old},i} + \text{Difference_Mean}_i, \quad (8)$$

其中 $X_{\text{old},i}$ 和 $X_{\text{new},i}$ 为更新前后的第 i 个个体.当学生的水平有所提高,即 $F(X_{\text{new},i}) \leq F(X_{\text{old},i})$ 时,当前的学习过程才被接受.

2.2 学生阶段

学生是教学过程中知识的受体,在教学过程中,学生学习新知识的途径有两种:从教师处获得新知识,从其他学生处通过相互交流获得新知识.在学生阶段,学生们通过向其他更优的个体学习,获得新的知识,学生阶段的学习过程按下式进行:

$$X_{\text{new},i} = \begin{cases} X_{\text{old},i} + r_i(X_i - X_j), & F(X_i) \leq F(X_j); \\ X_{\text{old},i} + r_i(X_j - X_i), & F(X_i) > F(X_j). \end{cases} \quad (9)$$

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

从TLBO算法的学习过程上看,学生仅通过向教师和其他个体学习获得新知识,学生自我的学习能力并未得到体现.实际上,学习能力强的学生具备更快

地提高自身知识水平的能力,而学习能力弱的将会较弱,即学生应进行差异化自主学习. 自主学习的过程中,学生可以对自身所学知识进行扩展,提高其局部搜索能力. 因此,为了提高种群的搜索效率和搜索精度,本文提出基于个体差异化自主学习的改进TLBO算法.

3 基于个体差异化自主学习改进TLBO算法

3.1 改进的TLBO算法

基于个体差异化自主学习的改进TLBO算法流程如图1所示. 首先根据问题的特点设计编码方式,同时对种群进行初始化,教师阶段按照式(7)和(8)进行更新;在学生阶段,为体现学生自主学习的能力,增加了差异化学习算子. 该算子强调学生可以根据自身能力大小朝不同的方向搜索学习,而且能力越强的学生,学习的机会越多,同时根据学习得到的最优结果对种群进行更新,提高算法的局部搜索能力.

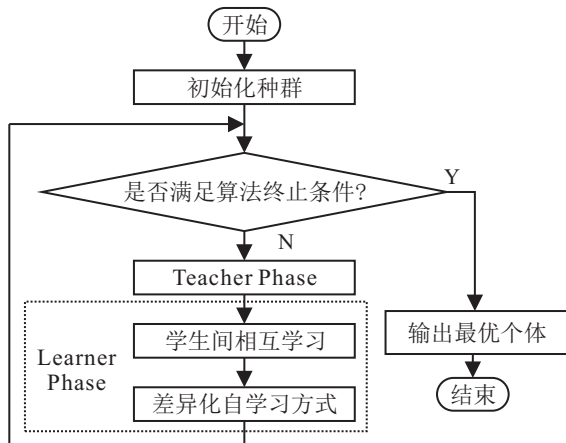


图1 改进TLBO算法基本框架

3.2 编码方式

编码方式即为调度方案解的构造. 在种群进化算法中,每一个个体都对应于该问题的一个解,进化的过程即为由原始解逐步迭代产生新的优秀个体的过程. 针对要解决的JSSP离散调度优化问题,采用基于操作的编码方式^[2]对问题进行编码.

每个个体由 $\sum_{j=1}^m n_j$ 个基因组成(其中: j 为工件号, m 为工件总数, n_j 为第 j 个工件的工序数),给同一工件的工序指定相同的符号,染色体基因顺序决定工件调度顺序. 解码时,根据染色体中基因出现的顺序进行解码,例如:某染色体(2,1,1,1,3,2,3,3,3,1,3,2,2),染色体中第1次出现的“1”表示工件1的第1个工序,第2次出现“1”表示工件1的第2个工序,以此类推,这样编码可使染色体自动满足链式约束.

3.3 教师阶段更新

从每代中选出最优个体作为 M_{new} (即 Teacher),负责引导种群进化. 由式(7),需求取 M_{new} 和 M_i 的差值,本文涉及离散问题求差,因此考虑使用基于位

置的交叉算子(PBX)^[17]或顺序交叉算子(OX)求取 M_{new} 和 M_i 的差距 $Difference_Mean_i$. 另外,在离散问题中, M_i 一般取为种群中适应值靠近中间的个体作为种群的平均位置^[6],在本文方法的研究中发现,若按此方式对 M_i 进行取值,会使经过交叉后得到的后代是近亲的概率增大,种群多样性减小,寻优精度降低. 因此,本文采用在种群中随机选取个体作为 M_i ,代替固定取种群中适应值靠近中间个体作为 M_i 的取值方式,这也意味着每个个体以不同的步长向最优个体(Teacher)学习. 相比 M_i 取适应值靠近中间个体的取值方式,该方式更好地保证了教师阶段个体更新的种群多样性,避免过早收敛,并设计相关实验证明其有效性. 同时,通过尝试发现,两种交叉算子的随机选取更有利于扩大解的搜索空间,因此在求 $Difference_Mean_i$ 时,随机选取PBX和OX任意一个算子进行求差操作. 另外,由式(8)随机选取PBX算子或OX算子对 $X_{old,i}$ 进行更新,得到新个体 $X_{new,i}$. 若得到的新个体比原来个体更优秀,则用新个体替换原个体,否则保留原个体.

3.4 学生阶段更新

学生阶段更新方式包括学生间相互学习和学生自主学习.

3.4.1 相互学习阶段

算法中,应考虑学习者之间相互影响共同进步的因素,从而进行相互学习. 相互学习过程按照式(9)进行,其操作流程如下.

Step 1: 初始化 $i = 1$.

Step 2: 选取个体 X_i , 随机选取另一个体 X_j .

Step 3: X_i, X_j 根据PBX算子或OX算子更新得到新个体 X_{new} .

Step 4: 若 $F(X_{new}) \leq F(X_i)$, 则接受 X_{new} 替换 X_i , 否则保留 X_i , 同时,令 $i = i + 1$; 若 $i \leq N$, 则转至 Step 2, 否则结束相互学习过程.

3.4.2 学生差异化自主学习阶段

在自主学习过程中,考虑到优秀个体自主学习的能力更强,搜索得到最优解的概率更大,应给优秀个体更多的学习机会和次数,鼓励其朝不同方向学习,扩大搜索领域,提高其找到最优解的能力. 自主学习完成后,用本次自主学习的最优位置进行学生个体的更新. 基于差异化自主学习算子的学生阶段更新步骤如下.

Step 1: 假设种群中共有 N 个个体,逐一评估个体学习能力.

Step 2: 根据个体的学习能力确定个体学习次数 $Num(i), i = 1, 2, \dots, N$.

Step 3: 对第 i 个体进行 $Num(i)$ 次自主学习算子操作,得到该学生的邻域种群. 每一次自主学习操作得到

一个新的个体, Num(*i*)次自学习之后形成该学生的邻域种群.

Step 4: 对每个学生邻域种群利用当前最优位置进行种群更新操作, 选取 *N* 个个体进入下一代.

1) 学习能力计算.

个体学习能力指的是个体对该问题的求解效果, 个体学习能力主要与个体的工期有关, 工期越小个体表现越优秀, 学习能力越强, 有

$$LA(i) = C/F(i). \quad (10)$$

其中: *LA*(*i*) 为第 *i* 个个体学习能力, *F*(*i*) 为第 *i* 个个体的工期, *i* = 1, 2, ..., *N*; 每次迭代过程中, *C* 为常数, 取 *C* = min(*F*(*i*)).

2) 学习次数计算.

自学习的过程中, 越优秀的学生自我学习的能力越强, 因此学习的机会更多, 即会获得更多的学习次数, 同时也应该给学习能力差的学生一定的学习机会提高自己. 因此, 将学习次数的取值限定在固定的范围 [*S*_{min}, *S*_{max}] 内, 学习次数为

$$\text{Num}(i) = \text{round}\left(\alpha \frac{LA(i) - LA_{\text{mean}}}{LA_{\text{max}} - LA_{\text{min}}}(S_{\text{max}} - S_{\text{min}}) + S_{\text{mean}}\right). \quad (11)$$

考虑到算法的运算效率, 取 *S*_{min} = 1, *S*_{max} = 15, *S*_{mean} = round((*S*_{max} - *S*_{min})/2), 其中 $\alpha = 0$ 或 1 , 称为差异化参数. 当 $\alpha = 1$ 时, 表示考虑个体间差异, 优秀个体的学习次数比其他个体多; 当 $\alpha = 0$ 时, 表示不考虑个体差异性, 每个个体以相同的学习次数 *S*_{mean} 进行学习.

3) 多样性自学习算子.

学习的过程由自学习算子体现, 为了增加搜索方向, 采用基于字符变异^[18]的多种邻域搜索方式. 在对个体进行邻域搜索时, 随机选取两个变异点 (Position1, Position2), 从如下 3 种变异搜索方式中随机选择一种进行邻域搜索:

i) 字符交换. 字符交换方式对个体改变较小, 适用于搜索个体距离较近的个体. 具体作法是: 确定两个不同的变异点, 若两个变异点的基因值不同, 则交换两点的基因值得到新个体. 图2为一次字符交换变异搜索的实例.

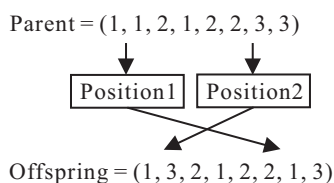


图2 字符交换算子

ii) 字符串反转. 对于字符串反转算子, 期望扩大

搜索范围, 若两个变异点选取过于接近, 则对基因改变较小, 因此期望两个变异点之间的距离不能太小, 即当 round(Position1 - Position2) > *l* * GenNum 时, 对其进行字符串反转操作, 其中 GenNum 为基因长度. 字符串反转方式如图3所示, 将所选取的两个变异点之间的基因片段翻转, 得到新个体.

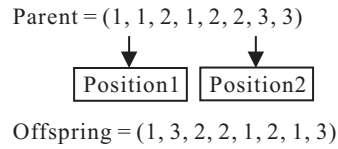


图3 字符串反转算子

iii) 字符串移位. 将 Position1 - Position2 之间的基因 (Position1 除外) 向左移动一位, 将原来 Position1 的基因值插到 Position2 处, 字符串移位算子如图4所示. 为了提高全局搜索能力, 与字符串反转算子相同, 限制两变异点之间的距离.

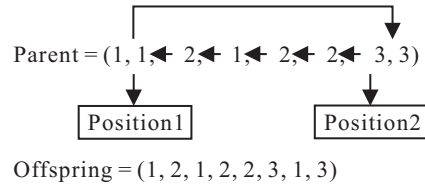


图4 字符串移位算子

4) 群体更新操作.

产生各学生的邻域种群后, 需要对其进行选择更新, 以保持种群大小不变. 群体更新过程中需要保持种群的多样性, 同时又要保证种群朝着最优解的方向收敛. 在种群经过自我学习过程后, 第 *i* 个体进行 Num(*i*) 次变异搜索操作后产生与之相近的邻域种群, 表示该学生向各个方向学习后的结果. 整个群体经过学习阶段更新后可以获得 *N* 个邻域种群, 为了保持种群大小固定, 需要对这些种群进行选择.

若采取经典的轮盘赌算子, 将 *N* 个种群混合之后再从中进行挑选, 则容易保留过多的近亲个体, 降低算法的全局搜索能力. 因此, 本文采取保留每个学生自学习的最优方向个体进行种群更新, 即在每个邻域种群中选取最优个体保留至下一代. 这种选择方式可以避免保留过多与上一代相同的个体, 使算法陷入局部最优, 同时保证了整个群体是朝着最优解的方向前进, 提高解的收敛精度.

3.4.3 算法时间复杂度分析

算法的关键步骤包括: 初始化种群、教师阶段更新操作 (包括 PBX, OX 操作)、学生阶段更新操作 (包括向其他个体学习过程和差异化自学习过程). 定义问题规模为 $n = n_p m$ (n_p 为工件总数, m 为机器总数). PBX 和 OX 操作时间复杂度为 $O(n)$, 学生阶段的自学习算子中字符交换算子复杂度为 $O(1)$, 字符串

反转和字符串移位算子的复杂度为 $O(n)$.

考虑种群规模为 N ,初始化种群的时间复杂度为 $O(n \times N)$,教师阶段更新操作的时间复杂度为 $O(n \times N)$,学生阶段更新操作的时间复杂度为 $O(n \times N \times S)$,其中 S 为学习次数.由于文中学习次数在常数范围内变化,学生阶段更新操作的时间复杂度可简化为 $O(n \times N)$.由此可以得出,算法的整体时间复杂度为 $O(n \times N)$,算法的时间复杂度不高.

4 仿真结果和分析

为验证所提出算法的正确性和可行性,选取OR-Library中的40个标准问题进行仿真,其规模大小包括 $6 \times 6, 10 \times 5, 10 \times 10, 15 \times 5, 15 \times 10, 15 \times 15, 20 \times 5, 20 \times 10, 20 \times 20, 30 \times 10$,具有代表性.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为算法重复运行所能求得最好解的平均值,反映了算法解的精度;Std为每次运行得到最好解的标准差,反映了算法的鲁棒性;ARPD(相对平均误差)^[16]评价算法寻优精度的稳定性,有

$$ARPD = \frac{\sum_{i=1}^n (x_i - WKS)}{n \times WKS} \quad (12)$$

其中: n 为算法重复运行次数, x_i 为第 i 次运行所能找到的最好解.

为了评价算法对所有仿真实例的整体求解能力,定义所考虑实例的统计性评价指标:最好解相对于最优解的偏差之和(SRPEB)为 $\sum(\text{Best} - \text{WKS})/\text{WKS}$;平均值相对于最优解的偏差和(SRPEA)为 $\sum(\text{Avg} - \text{WKS})/\text{WKS}$;所有实例的标准差均值(MS)和所有实例的ARPD均值(MARPD),用于评价算法对所有实例的求解精度.本文算法采用Matlab编写,运行在2.4 GHz CPU,4 GB内存的PC机上.

4.1 算法参数测试

算法设计的参数包括种群大小 N 、迭代次数Gen和差异化参数 α .为确定种群大小和迭代次数,选用测试用例FT10(10×10)对种群大小、迭代次数进行测试,算法独立重复运行20次,并采用文献[16]中的测试方法对 N 和Gen进行测试.分别选取 $N = 50, 100, 150, 200$; Gen = 1000, 1500, 2000, 2500进行组合,对算法进行测试,测试组合如表1所示,取种群大小相同迭代次数不同时,算法运行结果的均值作为该迭代次数下的算法结果.类似地,当迭代次数相同而种群大小不同时,算法运行结果的均值即为此迭代次数下的算法结果.测试结果如图5所示.

表1 种群大小和迭代次数实验组合

实验编号	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

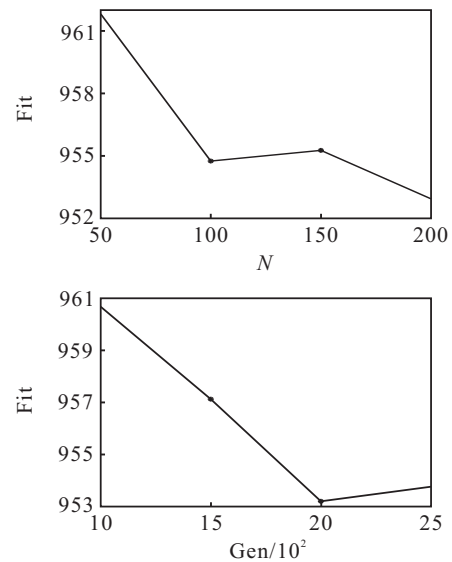


图5 种群大小 N 和迭代次数Gen测试

由图5可见,在保证算法收敛精度和考虑算法时间成本的情况下,本文取 $N = 100, \text{Gen} = 2000$.

对差异化参数 α 进行测试,分别取 $\alpha = 0$ 和 $\alpha = 1$,并选取不同规模的标准实例(FT10(10×10), FT20(20×5), la02(10×5), la03(10×5), la16(10×10), la17(10×10), la21(15×10), la22(15×10), la26(20×10), la27(20×10), la32(30×10), la33(30×10), la36(15×15), la40(15×15))进行测试,仿真结果如表2所示.由表2可见,当 $\alpha = 1$ 时,最好解、平均值和平均相对偏差均明显优于 $\alpha = 0$ 时的测试结果.对于la02, la32, la33实例,两种参数选择均能保证找到最优解.对于la03, la17实例,当 $\alpha = 1$ 时可以找到最优解.当 $\alpha = 0$ 或 $\alpha = 1$ 时,各实例SRPEB、SRPEA、MS和MARPD对比如表3所示.两种情况下算法的标准差均值相差不大,而对于SRPEB、SRPEA和MARPD而言, $\alpha = 1$ 的测试结果优于 $\alpha = 0$.由此可见,对个体进行差异化学习有助于提高算法对所有实例的整体

求解精度.

4.2 M_i 不同取值比较

M_i 随机取值和取每次迭代中适应值靠中间个体两种情况下,对不同规模的标准实例进行仿真,测试结果如表4所示.

由表4可见,除实例1a21的均值外, M_i 随机取值

的求解效果明显优于 M_i 固定取适应值靠近中间的个体. M_i 两种不同取值时,各实例的SRPEB、SRPEA、MS和MARPD对比如表5所示. M_i 取随机个体时,算法的各项统计指标都较优.可见,在TLBO算法的离散编码过程中,对 M_i 随机取值有利于提高种群收敛精度.

表2 差异化参数 α 的影响

Problem	WKS	$\alpha = 0$					$\alpha = 1$				
		Optimal	Worst	Avg.	Std	ARPD	Optimal	Max	Avg.	Std	ARPD
FT10	930	934	968	957.25	7.80	2.93	930	965	949.90	9.47	2.14
FT20	1165	1174	1202	1186.10	8.45	1.81	1174	1198	1182.30	5.52	1.49
la02	655	655	655	655.00	0.00	0.00	655	655	655.00	0.00	0.00
la03	597	597	603	597.30	1.34	0.05	597	597	597.00	0.00	0.00
la16	945	945	979	956.50	8.95	1.22	945	966	950.00	5.62	0.53
la17	784	784	789	784.50	1.54	0.06	784	784	784.00	0.00	0.00
la21	1046	1077	1105	1090.00	8.51	4.21	1066	1095	1082.10	8.19	3.45
la22	927	939	962	948.55	6.56	2.32	930	954	943.45	6.68	1.77
la26	1218	1238	1264	1251.30	8.19	2.73	1221	1253	1238.80	8.88	1.71
la27	1235	1296	1323	1313.00	6.99	6.32	1277	1313	1299.30	8.74	5.20
la32	1850	1850	1850	1850.00	0.00	0.00	1850	1850	1850.00	0.00	0.00
la33	1719	1719	1719	1719.00	0.00	0.00	1719	1719	1719.00	0.00	0.00
la36	1268	1315	1334	1325.10	7.98	4.50	1297	1334	1320.30	9.02	4.12
la40	1222	1264	1305	1287.70	9.60	5.37	1243	1299	1268.80	15.15	3.83

表3 α 取不同值时算法统计性指标对比

	SRPEB	SRPEA	MS	MARPD
$\alpha = 0$	0.1919	0.3153	5.4221	2.2514
$\alpha = 1$	0.1066	0.2425	5.5193	1.7314

表4 M_i 不同取值测试结果

Problem	WKS	M_i 固定取适应值靠近中间的个体					M_i 随机取值				
		Optimal	Worst	Avg.	Std	ARPD	Optimal	Max	Avg.	Std	ARPD
FT10	930	940	965	952.15	9.69	2.38	930	965	949.90	9.47	2.14
FT20	1165	1178	1196	1184.50	5.42	1.67	1174	1198	1182.30	5.52	1.49
la02	655	655	655	655.00	0.00	0.00	655	655	655.00	0.00	0.00
la03	597	597	597	597.00	0.00	0.00	597	597	597.00	0.00	0.00
la16	945	946	979	955.40	9.02	1.10	945	966	950.00	5.62	0.53
la17	784	784	789	784.30	1.13	0.04	784	784	784.00	0.00	0.00
la21	1046	1070	1098	1081.00	6.64	3.34	1066	1095	1082.10	8.19	3.45
la22	927	936	962	949.00	8.45	2.37	930	954	943.45	6.68	1.77
la26	1218	1227	1256	1239.80	7.67	1.79	1221	1253	1238.80	8.88	1.71
la27	1235	1284	1329	1309.50	8.46	6.03	1277	1313	1299.30	8.74	5.20
la32	1850	1850	1850	1850.00	0.00	0.00	1850	1850	1850.00	0.00	0.00
la33	1719	1719	1719	1719.00	0.00	0.00	1719	1719	1719.00	0.00	0.00
la36	1268	1306	1334	1321.80	10.27	4.25	1297	1334	1320.30	9.02	4.12
la40	1222	1252	1289	1273.70	11.07	4.23	1243	1299	1268.80	15.15	3.83

表5 M_i 取不同值时算法统计性指标对比

	SRPEB	SRPEA	MS	MARPD
M_i 固定取适应值靠近中间的个体	0.1572	0.4083	5.5586	1.9429
M_i 取随机个体	0.1066	0.2425	5.5193	1.7314

4.3 与其他算法性能比较

取 $\alpha = 1$,将TLBO-DL算法与其他算法进行比较,具体比较过程分为两部分:1)为验证本文TLBO

算法的改进效果,将其与TLBO算法^[16]数据进行比较;2)为进一步表明本文算法的寻优精度,将其与PaGA算法^[4]、NIMGA算法^[20]和MA(GR-RS)算法^[21]

的最好解和平均值进行比较. 文献[16]种群大小为100, 迭代次数为1000($N \times \text{Gen} = 100 \times 1000$), 本文算法也选取相同的参数, 运算结果如表6和表7所示. 文献[4]种群大小和迭代次数均为1000($N \times \text{Gen} = 1000 \times 1000$); 文献[20]孤岛个数为3, 每个孤岛上的

种群大小为100, 迭代次数为1000次 ($\text{Islands} \times N \times \text{Gen} = 3 \times 100 \times 1000$); 文献[21] $N \times \text{Gen} = 2500 \times 1000$; 本文算法参数 $N \times \text{Gen} = 100 \times 2000$. 运行结果对比如表8所示.

表6 与参考文献[16]中数据比较

Problem	Size	WKS	TLBO-DL					TLBO ^[16] (2014)				
			Optimal	Worst	Avg.	Std	ARPD	Optimal	Max	Avg.	Std	ARPD
FT06	6×6	55	55	55	55.00	0.00	0.00	55	58	56.50	1.08	2.73
FT10	10×10	930	934	975	958.70	9.69	3.09	938	1092	967.30	21.86	13.24
FT20	20×5	1165	1178	1203	1190.80	7.06	2.21	1165	1252	1199.20	12.38	8.91
la01	10×5	666	666	666	666.00	0.00	0.00	666	689	676.80	7.89	1.62
la02	10×5	655	655	655	655.00	0.00	0.00	655	713	687.80	12.51	5.01
la03	10×5	597	597	603	597.30	1.34	0.05	597	658	638.50	12.63	6.95
la04	10×5	590	590	593	590.60	1.23	0.10	607	633	619.60	10.22	5.02
la05	10×5	593	593	593	593.00	0.00	0.00	593	600	592.10	3.68	0.78
la06	15×5	926	926	926	926.00	0.00	0.00	926	943	931.10	6.61	0.55
la07	15×5	890	890	890	890.00	0.00	0.00	890	962	903.70	5.83	0.78
la08	15×5	863	863	863	863.00	0.00	0.00	864	923	896.40	21.53	3.87
la09	15×5	951	951	951	951.00	0.00	0.00	951	966	955.50	2.73	0.94
la10	15×5	958	958	958	958.00	0.00	0.00	958	965	960.20	2.09	0.75
la11	20×5	1222	1222	1222	1222.00	0.00	0.00	1222	1277	1248.00	21.62	2.13
la16	10×10	945	945	982	958.65	9.89	1.44	946	981	956.10	14.73	6.95
la21	15×10	1046	1075	1105	1090.90	10.18	4.29	1091	1245	1126.10	23.77	18.54
la27	20×10	1235	1289	1349	1322.20	13.66	7.06	1256	1627	1488.80	27.90	31.62
la31	30×10	1784	1784	1784	1784.00	0.00	0.00	1784	2134	1991.30	26.49	29.75
la36	15×15	1268	1309	1344	1331.00	8.52	4.97	1332	1504	1405.70	29.94	26.44
la40	15×15	1222	1268	1333	1295.60	17.84	6.02	1241	1423	1385.90	31.80	21.60

由表6可以看出, 除FT20、la27、la40最优值外, TLBO-DL算法得到的收敛效果明显优于文献[16]所提出的TLBO算法. 针对文献[16]所给实例, TLBO-DL和TLBO统计性指标对比如表7所示, TLBO-DL算法的整体收敛效果优于TLBO算法. 特别地, 对于FT06、FT10、la01~la20、la23和la31~la35, TLBO-DL都能寻找到最优解, 由此可见TLBO-DL具有更好的求解精度. 且TLBO-DL算法结果的Avg、Std和ARPD明显优于TLBO, 这表明TLBO-DL算法具有较好的鲁棒性和精度.

表7 与文献[16]数据的统计性指标对比

	SRPEB	SRPEA	MS	MARPD
文献[16]	0.1657	1.0219	14.8645	9.4090
TLBO-DL	0.1569	0.2924	3.9705	1.4615

由表8可见, TLBO-DL算法的收敛最好解和平均值都明显优于文献[4]的方法. 另外, 为了评价算法对所有仿真实例的整体求解能力, 根据前面所提到的统计指标(SRPEB和SRPEA)分析可知, TLBO-DL的SRPEB(0.1875)和SRPEA(0.3807)均优于文献[4]方法的0.4189和1.6237; 与文献[20]相比, 两种算法的

寻优精度均较好, 但TLBO-DL的SRPEB(0.2276)和SRPEA(0.4654)均优于文献[20]的0.2294和0.5021; 与文献[21]相比较, 除实例la26、la29、la30的仿真结果和la34的均值外, 本文算法最好解和平均值均优于文献[21]算法. 另外, TLBO-DL的SRPEB(0.1910)和SRPEA(0.3897)均优于文献[21]的0.2692和0.5697, 表明本文算法对所有实例的整体寻优能力优于其他算法.

为进一步比较算法的差异性, 运用双侧T-检测法将TLBO-DL算法的测试结果与其他文献算法所得结果分别根据式(14)和(15)进行显著性检验. 其中: n_1 、 n_2 为样本数, s_1 、 s_2 为标准差, \overline{D}_1 、 \overline{D}_2 为两个样本的样本均值. 以每个实例的测试结果为一个样本, 与参考文献中的数据进行显著性检验, 有

$$s_{12}^2 = \frac{(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2}{n_1 + n_2 - 2}, \quad (13)$$

$$t = \frac{\overline{D}_1 - \overline{D}_2}{s_{12} \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}}. \quad (14)$$

与文献[16]相比较, $n_1 = n_2 = 19$, 检验自由度 $d_f = n_1 - 1 + n_2 - 1 = 36$, 计算得到 $|t| = 2.219$, 取显

表8 TLBO-DL与其他算法比较

Problem	Size	WKS	TLBO-DL		PaGA(2010) ^[4]		NIMGA(2016) ^[20]		MA(GR-RS) (2009) ^[21]	
			Optimal	Avg.	Optimal	Avg.	Optimal	Avg.	Optimal	Avg.
FT 06	6×6	55	55	55	55	55	55	55	—	—
FT 10	10×10	930	930	949.9	997	1028	930	952.33	—	—
FT 20	20×5	1165	1174	1182.3	1196	1224	1173	1187.2	—	—
la 01	10×5	666	666	666	666	666	666	666	666	667.60
la 02	10×5	655	655	655	655	659	655	656.53	655	656.27
la 03	10×5	597	597	597	617	617	597	597.8	597	613.93
la 04	10×5	590	590	590.3	607	609	590	590	590	593.33
la 05	10×5	593	593	593	593	593	593	593	593	593.00
la 06	15×5	926	926	926	926	926	926	926	926	926.00
la 07	15×5	890	890	890	890	890	890	890	890	890.00
la 08	15×5	863	863	863	863	863	863	863	863	863.00
la 09	15×5	951	951	951	951	951	951	951	951	951.00
la 10	15×5	958	958	958	958	958	958	958	958	958.00
la 11	20×5	1222	1222	1222	1222	1223	1222	1222	1222	1222.00
la 12	20×5	1039	1039	1039	1039	1039	1039	1039	1039	1039.00
la 13	20×5	1150	1150	1150	1150	1150	1150	1150	1150	1150.00
la 14	20×5	1292	1292	1292	1292	1292	1292	1292	1292	1292.00
la 15	20×5	1207	1207	1207	1207	1273	1207	1207	1207	1207.13
la 16	10×10	945	945	950	994	1005	946	956.07	945	968.27
la 17	10×10	784	784	784	793	799	784	784.4	784	788.93
la 18	10×10	848	848	853.45	860	876	848	854.07	848	859.27
la 19	10×10	842	842	847.85	873	876	842	847.67	842	855.47
la 20	10×10	902	902	905	912	917	907	909.47	907	910.00
la 21	15×10	1046	1066	1082.1	1146	1169	1058	1087.5	1079	1097.60
la 22	15×10	927	930	943.45	1007	1023	937	952.67	960	981.00
la 23	15×10	1032	1032	1032	1033	1051	1032	1032.3	1032	1032.00
la 24	15×10	935	945	958.65	1012	1026	947	969.33	959	996.40
la 25	15×10	977	989	1011.9	1067	1078	992	1012.3	991	1016.67
la 26	20×10	1218	1221	1238.8	1323	1333	1218	1239.5	1218	1234.27
la 27	20×10	1235	1277	1299.3	1359	1370	1269	1294.7	1286	1306.33
la 28	20×10	1216	1251	1261.2	1369	1382	1247	1263	1286	1306.33
la 29	20×10	1152	1230	1247.8	1322	1381	1241	1262.3	1221	1240.47
la 30	20×10	1355	1357	1378	1437	1505	1355	1357.8	1355	1362.33
la 31	30×10	1784	1784	1784	1844	1857	1784	1784	1784	1784.00
la 32	30×10	1850	1850	1850	1907	1918	1850	1850	1850	1850.00
la 33	30×10	1719	1719	1719	—	—	1719	1719	1719	1719.00
la 34	30×10	1721	1721	1729.8	—	—	1721	1723.5	1721	1721.00
la 35	30×10	1888	1888	1888	—	—	1888	1888.7	1888	1888.00
la 36	15×15	1268	1297	1320.3	—	—	1293	1317.7	1307	1328.67
la 40	15×15	1222	1243	1268.8	—	—	1246	1266.3	1252	1279.60

显著性水平 $\alpha = 0.05$, 查表可得 $t_{0.05/2} = 2.028$, $|t| > t_{0.05/2}$, 可见两种算法的运算结果存在显著性差异. 同理, 与文献[4]的算法进行显著性检验得到 $|t| = 3.6533$, 取显著性水平 $\alpha = 0.05$, 查表可得 $t_{0.05/2} = 1.995$, $|t| > t_{0.05/2}$, 可见两种算法的运算结果存在显著性差异. 与文献[20]相比较, 经过检验, 两种算法显著性差异并不明显, 但根据分析, TLBO-DL 的统计指标 SRPEA 和 SRPEB 优于文献[20]方法, 表明本文算法寻优精度更稳定, 对所有实例的整体寻优能力更好. 与文献[21]的算法进行显著性检验得到 $|t| = 4.467$, 查表可知 $t_{0.05/2} = 2.042$, 两种算法对于 JSSP

问题的求解效果具有显著性差异. 由此可见, 本文所提出的 TLBO-DL 对于 JSSP 具有较好的寻优精度和稳定性.

5 结 论

本文对车间作业调度问题进行了基于教学算法的调度方法研究. 针对教学算法局部搜索能力的不足, 提出一种基于差异化学习算子的教学算法, 使学习过程不仅考虑个体间的相互学习, 还应考虑个体的自我学习能力. 采取交叉算子 PBX 和 OX, 改进教师阶段和学生阶段的更新, 对学生的学习能力进行评估, 并设计自学习算子, 根据学生的能力大小进行差异化

学习. 采用保留最优个体的种群更新策略, 以保证种群的规模. 另外, 对算法的复杂度进行分析, 并通过在 OR-Library 中的实例进行验证, 表明了差异化自学习算子的有效性. 与其他文献中的方法进行比较, 实例验证了所提出算法在求解 JSSP 上是有效的.

参考文献(References)

- [1] 王书锋, 邹益仁. 车间作业调度 (JSSP) 问题简明综述 [J]. 系统工程理论与实践, 2003, 1(1): 49-55.
(Wang S F, Zou Y R. Techniques for the job shop scheduling problem: A survey[J]. Systems Engineering—Theory & Practice, 2003, 1(1): 49-55.)
- [2] 夏柱昌, 刘芳, 公茂果, 等. 基于记忆库拉马克进化算法的作业车间调度 [J]. 软件学报, 2010, 21(12): 3082-3093.
(Xia Z C, Liu F, Gong M G, et al. Memory based Lamarckian evolutionary algorithm for job shop scheduling problem[J]. J of Software, 2010, 21(12): 3082-3093.)
- [3] Xu Xiao-dong, Li Cong-xin. Research on immune genetic algorithm for solving the job-shop scheduling problem[J]. Int J of Advanced Manufacturing Technology, 2007, 34(7/8): 783-789.
- [4] Leila Asadzadeh, Kamran Zamanifar. An agent-based parallel approach for the job shop scheduling problem with genetic algorithms[J]. Mathematical and Computer Modelling, 2010, 52(11/12): 1957-1965.
- [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] Venkata Rao R, Vivek Patel. Multi-objective optimization of heat exchangers using a modified teaching-learning-based optimization algorithm[J]. Applied Mathematical Modelling, 2012, 37(3): 1147-1162.
- [8] Rao R V, Savsani V J, Vakharia D P. Teaching-learning-based optimization: A optimization method for continuous non-linear large scale problems[J]. Information Sciences, 2012, 183(1): 1-15.
- [9] Venkata Rao R, Vivek Patel. An elitist teaching-learning-based optimization algorithm for solving complex constrained optimization problems[J]. Int J of Industrial Engineering Computations, 2012, 3(4): 535-560.
- [10] Binod Kumar Saha, Tridipta Kumar Patia, Jyoti Ranjan Nayaka, et al. A novel hybrid LUS-TLBO optimized fuzzy-PID controller for load frequency control of multi-source power system[J]. Electrical Power & Energy Systems, 2016, 74(1): 58-69.
- [11] 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.)
- [12] 李瑞国, 张宏立, 王雅. 基于反馈教学优化算法的混沌系统参数辨识 [J]. 计算机应用, 2015, 35(5): 1367-1372.
(Li R G, Zhang H L, Wang Y. Parameter identification in chaotic system based on feedback teaching-learning-based optimization algorithm[J]. J of Computer Applications, 2015, 35(5): 1367-1372.)
- [13] 王凌, 郑环宇. 一种多目标资源受限项目调度问题的教学算法 [J]. 控制与决策, 2015, 30(10): 1868-1872.
(Wang L, Zheng H Y. A teaching-learning-based optimization algorithm for multi-objective resource constrained project scheduling problem[J]. Control and Decision, 2015, 30(10): 1868-1872.)
- [14] Tansel Dokeroglu. Hybrid teaching-learning-based optimization algorithms for the quadratic assignment problem[J]. Computers & Industrial Engineering, 2015, 85: 86-101.
- [15] Gonca Tuncel, Dilek Aydin. Two-sided assembly line balancing using teaching-learning based optimization algorithm[J]. Computers & Industrial Engineering, 2014, 74(1): 291-299.
- [16] Adil Baykasoglu, Alper Hamzadayi, Simge Yelkenci Kose. 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.
- [17] Tailip Kellegoz, Bilal Toklu, John Wilson. Comparing efficiencies of crossover operators for one machine total weighted tardiness problem[J]. Applied Mathematics and Computation, 2008, 199(2): 590-598.
- [18] 李茂军, 罗安, 童调生. 人工免疫算法及其应用研究 [J]. 控制理论与应用, 2004, 21(2): 153-157.
(Li M J, Luo A, Tong T S. Artificial immune algorithm and its applications[J]. Control Theory & Applications, 2004, 21(2): 153-157.)
- [19] Marnix Kammer, Marjan van den Akker, Han Hoogeveen. Identifying and exploiting commonalities for the job-shop Scheduling problem[J]. Computers & Operations Research, 2011, 38(11): 1556-1561.
- [20] Mohamed Kurdi. An effective new island model genetic algorithm for job shop scheduling problem[J]. Computers & Operations Research, 2016, 67(5): 132-142.
- [21] Kamrul Hasan S M, Ruhul Sarker Daryl Essam. Memetic algorithms for solving job-shop scheduling problems[J]. Memetic Computing, 2009, 1(1): 69-83.

(责任编辑: 郑晓蕾)