求解约束优化问题的融合粒子群的教与学算法*

赵乃刚[1]，李　勇[1]，王振荣[2]

（1. 山西大同大学 数学与计算机科学学院，山西 大同 037009；2. 山西省大同市人民政府信息化中心，山西 大同 037009）

摘　要：针对约束优化问题，提出了一种融合粒子群的教与学算法。算法采用了一种自适应的教学因子，使得算法的搜索性能可以自适应地调整。引入了自我学习和相互学习的学习模式，使得信息交流更加多样化，增强了算法的全局搜索能力；最后根据适应度值将整个种群分为两个子种群，对适应度值差的子种群采用粒子群算法以提升收敛性能，对适应度值优的子种群采用教与学优化算法以增强种群的多样性，通过两种算法的优势互补，提升了算法的整体优化性能。通过在 22 个标准测试函数的实验和与其他三种算法的比较表明，融合粒子群的教与学算法求解精度高，收敛速度快，它是一种可行、高效的优化算法。

关键词：教与学算法；粒子群算法；约束优化问题；自适应；约束处理

中图分类号：TP301.6　　**文献标志码：**A　　**文章编号：**1001-3695(2018)05-1307-03

doi:10.3969/j.issn.1001-3695.2018.05.006

Hybrid teaching-learning-based fused with particle swarm optimization of constrained optimization problem

Zhao Naigang[1], Li Yong[1], Wang Zhenrong[2]

(1. *College of Mathematics & Computer Science, Shanxi Datong University, Datong Shanxi* 037009, *China*; 2. *Information Center of Datong People's Government of Shanxi Province, Datong Shanxi* 037009, *China*)

Abstract: Aiming at the constrained optimization problem, this paper developed a hybrid teaching-learning-based fuse with particle swarm optimization. It adopted an adaptively teaching factors, and it could adjust the search performance of algorithm adaptively. The introduction of the learning mode of self-study and learn from each other made information communication was more diverse and it enhanced the global search ability of the algorithm. Finally, the algorithm divided the whole population into two subpopulations according to the fitness value. The poor particles adopted particle swarm algorithm to improved the convergence of the algorithm, and the better particles adopted teaching-learning-based algorithm to increase the diversity of population. The complementary advantages of both algorithms improved the performance of algorithm. This paper compared the new algorithm with the other three kinds of algorithms on twenty-two test functions, the simulation experiments show that the new algorithm has better performance. It is a feasible and efficient optimization algorithm.

Key words: teaching-learning-based; particle swarm optimization; constrained optimization problem; self-adaptive; constraint processing method

0　引言

单目标约束优化问题是指一类只有一个目标函数和多个约束的数学规划问题，它是一类 NP 难问题。它存在于人们日常生活的方方面面，包括自然科学、工程管理等领域。因此，寻找一种求解此类问题的高效算法意义重大。单目标约束优化问题的一般模型为

$$\begin{aligned} &\min\ F=f(x,y)\\ &\text{s.t.}\ \ h_i(x)=0\quad i=1,2,\cdots,m\\ &\qquad g_j(x)\leqslant 0\quad j=1,2,\cdots,l\\ &\qquad x\in X,y\in Y \end{aligned}$$

其中：$h_i(x,y)$是等式约束；$g_j(x,y)$是不等式约束；x 表示实数变量；m 和 l 分别是等式和不等式约束的个数。

目前，求解此类问题主要有两类方法，一类是传统的优化算法，如割平面法、分支定界法、外逼近法等，然而在求解单目标约束优化问题时存在很大的局限性；另外一类是智能优化算法，主要包括粒子群算法[1]、遗传算法、差分进化算法等。实际应用表明，它们是求解约束优化问题的一类可行、高效的优化算法。目前的群智能算法均是模拟了自然界中的生物，为了优化它们的生存环境而进行的无意识寻优行为。不同的智能算法有很多的共同点，如算法均有群体初始状态，本代群体的更新依赖上代群体；群体中的个体之间存在直接或间接的学习方式，个体的选择会受到其他个体的影响；算法对数学性质的要求较弱，不要求连续性、可微性、可导性，只注重结果性；在寻优的过程中，会受随机性的影响，即使初始状态相同，也不一定会有相同的寻优过程，随机性是算法寻找新位置的重要条件，它们是一类不确定的算法；算法中没有确定的中央控制器来指导个体下一步的行为，个体均有着自身的自我组织性和自我进化性来决定自身的下一步行为，个体会受到贪婪策略的诱导以此保证正确的寻优方向。文献[2]为了求解带凸不等式和线

收稿日期：2017-01-02；**修回日期：**2017-02-27　**基金项目：**国家自然科学基金资助项目(61672331)；山西省高等学校教学改革创新项目(J2017093)；山西大同大学科学研究项目(2016K1)

作者简介：赵乃刚(1975-)，男，山西应县人，讲师，硕士研究生，主要研究方向为数据挖掘(2893966820@qq.com)；李勇(1974-)，男，辽宁大连人，博士，主要研究方向为智能算法及无线网络；王振荣(1988-)，男，山西应县人，硕士研究生，主要研究方向为计算机科学与技术.

性等式的非光滑凸优化问题，提出了一种递归神经网络，并将其应用于基因调控网络识别。文献[3]结合神经网络集成和多种群智能构造了一种改进的神经网络集成算法，首先每个组件的前向神经网络通过混沌粒子群算法和梯度下降法进行优化；其次采用多个不同的群体来构建群体智能；最后实验结果表明了提出的算法在功能预测方面的优越性。文献[4]将一种全新的混沌寻优策略引入粒子群算法中，提高了粒子群算法求解约束优化问题的全局搜索能力，增强了算法的收敛性和鲁棒性。文献[5]将随机权重引入遗传算法中，提高了算法求解约束优化问题的寻优精度。文献[6]为了求解约束优化问题，对差分进化算法进行了改进，实验结果表明它是求解复杂约束优化问题的一种有效算法。在教与学算法和粒子群算法求解约束优化问题时，单个算法求解都存在一些缺陷。为了更好地求解约束优化问题，本文将教学式算法与粒子群算法相结合，充分利用两种算法的优点，提出了一种求解约束优化问题的融合粒子群的教学式算法（TLBO-PSO）。

1 教与学优化算法

教与学优化算法[7~9]是一种基于对课堂上以班级为单位教学过程的仿真模拟的群智能优化算法。在算法中，将整个种群充当班级，种群中的最优个体充当老师，种群个体充当学生。算法分为“教师阶段”和“学生阶段”，通过这两个阶段的协同进化从而提升种群的整体水平。在教师阶段，老师根据班级的整体水平采用以下方式对学生进行教学：

$$x_{\text{new},i}=x_{\text{old},i}+r_i(x_t-T_F\times\text{mean}(x)) \tag{1}$$

其中：$x_{\text{new},i}$和$x_{\text{old},i}$分别为第i个学生粒子在更新后和更新前的位置；r_i是介于闭区间[0,1]的随机数；T_F为教学因子，一般取1或2；mean(x)为群体粒子的平均位置。

在学生阶段，学生为了进一步提高自身的学习水平，与班级中的其他个体进行进一步交流，采用如下方法进行学习：

$$x_{\text{new},i}=\begin{cases}x_{\text{old},i}+r_i(x_{\text{old},i}-x_{\text{old},j}) & f(x_{\text{old},j})<f(x_{\text{old},i})\\ x_{\text{old},i}+r_i(x_{\text{old},j}-x_{\text{old},i}) & f(x_{\text{old},i})<f(x_{\text{old},j})\end{cases} \tag{2}$$

其中：$x_{\text{old},j}$表示当前群体中第j个学生个体粒子的位置；$f(x_{\text{old},i})$和$f(x_{\text{old},j})$分别为第i个学生个体粒子和第j个学生个体粒子的适应度值。

2 标准粒子群算法

在粒子群算法，所有粒子在搜索空间中飞行，并根据对自身的自我认知和与其他粒子的信息交流来搜索最优位置。在标准粒子群算法中，粒子采用下面的迭代方式来更新它们的速度和位置。

$$v_i(t+1)=\omega v_i(t)+c_1r_1(P_i(t)-x_i(t))+c_2r_2(g_b-x_i(t)) \tag{3}$$

$$x_i(t+1)=x_i(t)+v_i(t+1) \tag{4}$$

其中：$v_i(t)$、$x_i(t)$分别表示第i个粒子在t时刻的速度和位置；$P_i(t)$为第i个粒子在t时刻个体最优位置；g_b为t时刻的群体最优位置；ω为粒子更新的惯性权重值；c_1、c_2为粒子的学习因子；r_1、r_2是介于闭区间[0,1]的随机数。

3 基于混合粒子群算法求解约束优化问题

3.1 自适应教学因子

在基本的教与学优化算法中，教学因子T_F的值只能随机地取1或2，教学的方式非常单一。如果采用这样的教学方式，教师向学生传授学习内容时，学生要么全部接受，要么全部不接受。但在实际的教学过程中，学习能力强的学生学习很快，而学习能力弱的学生则学习较慢，当所有学员的整体水平接近老师的水平时，学习进步的速度也会逐步变慢。因此，本文提出了一种自适应教学因子T_F为

$$T_F(t)=\frac{\text{mean}(x(t))}{x_t}+1 \tag{5}$$

其中：mean($x(t)$)表示学员学习的平均水平值；x_t表示教师本身的教学水平。改进后的教学因子可以在整个搜索过程中根据不同学生的学习能力进行自适应的更新，从而使得TLBO算法的寻优性能可以被动态地调整。

3.2 自我学习和相互学习的结合

在学生的整个学习过程中，老师的教学是一方面，但最主要的还是学生的自我学习和与其他学生的相互学习。但基本的教与学算法中没有引入学生的自学和相互学习模式。因此，本文采用了如下的一种最简单的自学方式：

$$x_{\text{new},i}=x_{\text{old},i}+\text{rand}(0,1)\times(x_i^U-x_i^L)+\text{rand}(0,1)\times(x_{\text{old},i}-x_{\text{old},j}) \tag{6}$$

引入自我学习和相互学习之后，所有学生的学习方式更加多样化，增强了TLBO算法的全局寻优能力。

3.3 新的约束处理方法

在求解约束优化问题时，它的最优解往往在约束边界的附近，但是一般的约束处理办法只考虑可行域中的粒子。譬如，当目标函数连续时，在距离边界非常近的不可行域中某些粒子的适应度值可能比可行域中的粒子的适应度值要好很多，因此，此时不可行域中的这些粒子更能对算法的优化进程起到引导作用。在本文中，粒子在比较优劣时，会让非常接近边界的不可行域中的某些粒子与可行粒子进行适应度值的比较，以此来保留不可行域中的某些粒子，使它们能够引导别的粒子朝着有优良信息的区域移动。具体的约束处理方法如下：a）如果粒子i和粒子j都是可行粒子，选择适应度值小的粒子；b）如果粒子i和粒子j都是不可行粒子，选择违反约束条件小的粒子；c）如果粒子i是可行粒子而粒子j是不可行粒子，若此时粒子j与可行域的边界的距离小于等于一个给定的正常数ε，则比较它们的适应度值，选择适应度值小的粒子；若此时粒子j与可行域的边界的距离大于ε，选择可行粒子i。

3.4 TLBO-PSO求解约束优化问题的流程

a）初始化种群并设置参数，包括初始化粒子位置，设置最大迭代次数、种群规模、可行域等；

b）计算粒子的适应度值，并将种群平均分为两个子种群（适应度值优的为种群1，适应度值差的为种群2），并确定教师个体位置；

c）对种群1中的粒子，采用改进的TLBO算法更新粒子的位置，并根据新的约束处理方法更新每个个体，对种群2中的粒子，采用标准PSO算法更新粒子的位置，也根据新的约束处理方法更新每个个体；

d）将更新后的种群1和种群2合并成大种群；

e）若满足终止条件，输出最优解；否则返回b）继续执行。

4 实验结果与分析

为了体现本文改进的算法（TLBO-PSO）的有效性，以及在求解问题时的优越性，本文选取了25个混合整数规划问题的测试函数来评测本文算法。其中测试问题$P1\sim P12$来自文献[10]，$P13\sim P22$来自文献[11]，每个测试问题均单独运行了100次。TLBO-PSO算法的实验结果如表1所示。其中mean表示平均值，best表示算法求出的最好值，worst表示算法求出的最差值，std表示方差，t表示算法运行时间，ave表示算法的

平均迭代次数，*Ps* 表示算法的成功率。

表 1　TLBO-PSO 算法的运行结果

测试问题（*P*）	mean	best	worst	std	*t*	ave	*Ps*
*P*1	87.500893	87.499591	87.502231	0.000182	0.289400	2548	1
*P*2	7.564421	7.485991	7.657340	0.048826	0.028000	477	0.99
*P*3	4.579594	4.579590	4.579600	1.100371e−5	0.587100	28619	1
*P*4	−0.805078	−0.807844	−0.786119	0.004687	2.033370	59930	0.70
*P*5	−0.974565	−0.974565	−0.974565	0	0.005510	221	1
*P*6	−0.999771	−0.999900	−0.999585	3.887321e−6	0.076100	2988	1
*P*7	2.000400	2.000001	2.000877	2.871121e−5	0.193200	1788	1
*P*8	2.124611	2.124469	2.124787	1.998210e−6	0.188000	3072	1
*P*9	1.0765434	1.0765433	1.0765435	4.098721e−9	0.390350	19986	1
*P*10	99.241311	99.239411	99.243118	0.001699	0.041300	1611	1
*P*11	3.581077	3.557462	3.603398	0.245387	5.301100	43800	0.90
*P*12	−32217.41	−32217.42	−32217.39	0.011087	0.867210	8671	1
*P*13	−6961.813830	−6961.813877	−6961.81384	0.0000031	2.110900	198722	1
*P*14	−68	−68	−68	0	0.007900	268	1
*P*15	−6	−6	−6	0	0.000190	10	1
*P*16	−0.946195	0.950197	0.9434705	0.000221	0.015700	109	1
*P*17	8	8	8	0	0.000690	1	1
*P*18	14	14	14	0	0.790580	5001	1
*P*19	−42.632121	−42.632121	−42.632121	0	0.001300	4	1
*P*20	0.004221	3.344275e−7	0.006100	0.003061	0.000014	3	1
*P*21	807	807	807	0	1.670050	11890	1
*P*22	−1038911	−1095387	−1030421	1.138762e+4	2.557090	9699	1

为了进一步说明 TLBO-PSO 算法的性能，针对问题 *P*1 ~ *P*12，把 TLBO-PSO 算法的实验结果与文献[10]中的 AXNUM 算法、文献[11]中的 MA-MADE 算法、文献[6]中的 IDE 算法的实验结果进行了比较。算法的性能是从算法获得的平均值、标准差、成功率三个方面进行比较，比较结果如表 2 所示。

表 2　四种算法的性能比较

测试问题（*P*）	已知最优值	求解最优值	评价指标	AXNUME’	MA-MADEE	IDE	TLBO-PSO
			成功率	0.47	0.86	1	1
1	87.5	87.499591	平均值	88.198729	88.100621	87.501908	87.500893
			标准差	2.114390	1.901762	0.0031812	0.000182
			成功率	0.03	0.06	0.2	0.99
2	7.667	7.485991	平均值	8.000878	7.903341	7.836696	7.564421
			标准差	0.039771	0.098771	0.119771	0.048826
			成功率	0.93	1	0.8	1
3	4.5796	4.579590	平均值	4.590021	4.579594	4.579599	4.579594
			标准差	0.291036	0.000002	0.000007	1.100371e−5
			成功率	0.93	0.9	1	0.70
4	−0.808844	−0.807844	平均值	−0.807701	−0.807907	−0.808744	−0.805078
			标准差	0.000414	0.004155	0.000000	0.004687
			成功率	0.96	0.93	1	1
5	−0.974565	−0.974565	平均值	−0.974511	−0.974340	−0.974565	−0.974565
			标准差	0.000870	0.000910	0.000000	0
			成功率	1	1	1	1
6	−0.999486	−0.999900	平均值	−0.999631	−0.999636	−0.999635	−0.999771
			标准差	0.000113	0.000111	0.000113	3.887321e−6
			成功率	0.93	1	1	1
7	2	2.000001	平均值	2.008805	2.000000	2.000001	2.000400
			标准差	0.039771	0.000000	0.000000	2.871121e−5
			成功率	0.9	1	1	1
8	2.1247	2.124469	平均值	2.155991	2.124561	2.124598	2.124611
			标准差	0.146909	0.000067	0.000089	1.998210e−6
			成功率	0.4	0.53	0.83	1
9	1.076543	1.0765433	平均值	1.124591	1.077718	1.094771	1.0765434
			标准差	0.068898	0.051928	0.052113	4.098721e−9
			成功率	1	1	0.96	1
10	99.245209	99.239411	平均值	99.239510	99.512140	99.243391	99.241311
			标准差	0.003411	1.367291	0.003842	0.001699
			成功率	0.3	0.86	0.83	0.90
11	3.557463	3.557462	平均值	3.598822	3.563313	3.561157	3.581077
			标准差	0.080213	0.030001	0.383811	0.245387
			成功率	1	1	1	1
12	−32217.4	−32217.42	平均值	−32217.43	−32217.43	−32217.43	−32217.41
			标准差	0.002738	0.001690	0.000000	0.011087

从表 1 和 2 中可以看出，TLBO-PSO 算法对 22 个测试函数均求出了已知最优解，并且对 *P*2、*P*6、*P*10 和 *P*11 等问题，TLBO-PSO 求出的最优值比目前已知的最优值更好。本文算法除了对问题 *P*2、*P*4、*P*11 的求解没有达到 100% 的成功率，其他 19 个测试问题的求解均达到 100% 的成功率。在标准差方面，TLBO-PSO 算法也是相对比较稳定没有出现相对较大的值，说明本文算法的寻优性能较稳定。综上，本文算法不管是从算法成功率上，还是从函数评价次数和运行时间上看，本文算法均比其他三种算法要好。

5　结束语

为求解约束优化问题，提出了一种融合粒子群的教与学算法。算法采用了一种自适应教学因子，并引入了自我学习和相互学习的学习模式，最后根据每次迭代后计算出的适应度值将整个种群分为两个子种群，对适应度值变差的子种群采用粒子群算法以提升收敛性能，对于适应度值变优的子种群采用教与学优化算法以增强种群的多样性，通过两种算法的优势互补，进而提升算法的整体优化性能。最后通过实验测试结果表明了 TLBO-PSO 算法在求解约束优化问题的可行性和有效性。

参考文献：

[1] Pan Feng, Li Xiaoting, Zhou Qian, *et al*. Analysis of standard particle swarm optimization algorithm based on Markov chain[J]. Acta Automatic Sinica, 2013, 39(4): 381-389.

[2] Zhao Zengshun, Feng Xiang, Lin Yanyan, *et al*. Evolved neural network ensemble by multiple heterogeneous swarm intelligence[J]. Neurocomputing, 2015, 149: 29-38.

[3] Cheng Long, Hou Zengguang, Lin Yingzi, *et al*. Recurrent neural network for non-smooth convex optimization problems with application to the identification of genetic regulatory networks[J]. IEEE Trans on Neural Networks, 2011, 21(5): 714-726.

[4] Tan Yue, Tan Guanzheng, Deng Shuguang. Hybrid particle swarm optimization with chaotic search for solving integer and mixed integer programming problems[J]. Journal of Central South University, 2014, 21(7): 2731-2742.

[5] Zou Guocheng, Jia Liping, Zou Jin. Random-weight based genetic algorithm for multiobjective bilevel mixed linear integer programming[C]//Proc of the 8th International Conference on Natural Computation. Piscataway, NJ: IEEE Press, 2012: 693-697.

[6] Wu Jun, Gao Yuelin, Yan Lina. An improved differential evolution algorithm for mixed integer programming problems[C]//Proc of the 9th International Conference on Computational Intelligence and Security. Piscataway, NJ: IEEE Press, 2013: 31-35.

[7] Rao R V, Savsani V J. Teaching-learning-based optimization: an optimization method for uncontinuous non-linear large scale problems[J]. Enigineering Optimization, 2012, 44(2): 1447-1462.

[8] 李会荣，乔希民. 融合差分变异的教与学优化算法[J]. 计算机工程与应用, 2016, 52(5): 36-40.

[9] Rao R V, Patel V. An elitist teaching-learning-based optimization algorithm for solving complex constrained optimization problems[J]. International Journal of Industrial Engineering Computations, 2012, 3(4): 535-560.

[10] Deep K, Singh K P, Kansal M L, *et al*. A real coded genetic algorithm for solving integer and mixed integer optimization problems[J]. Applied Mathematics and Computation, 2009, 212(2): 505-518.

[11] Liao T W. Two hybrid differential evolution algorithms for engineering design optimization[J]. Applied Soft Computing, 2010, 10(4): 1188-1199.