

文章编号:1671-6833(2020)04-0059-09

基于反向学习的微种群教与学优化算法及其应用

范勤勤^{1,2}, 柳缔西子¹, 王筱薇¹, 韩新³, 王维莉¹

(1.上海海事大学 物流研究中心,上海 201306; 2.上海交通大学 电子信息与电气工程学院,上海 200237; 3.同济大学 上海防灾救灾研究所,上海 200092)

摘要:为提高教与学优化算法的收敛速率且保证其可靠性,提出了一种基于反向学习的微种群教与学优化算法(opposition-based learning teaching-learning-based optimization algorithm with a micro population, OBL-μTLBO)。在所提算法中,利用小的种群规模(微种群)来提高教与学优化算法的收敛速率和降低对计算机内存的要求。同时,OBL-μTLBO 算法使用反向学习所得“教师”来指引种群进化,以此提高算法的全局探索能力和避免其陷入局部最优。仿真结果表明,OBL-μTLBO 算法不仅具有较好的寻优性能,而且还具有较快的收敛速度。最后,将 OBL-μTLBO 算法应用于求解非合作博弈纳什均衡问题,取得令人满意的结果。

关键词:教与学优化;微种群;反向学习;非合作博弈

中图分类号: TP311.1 **文献标志码:** A **doi:**10.13705/j.issn.1671-6833.2020.01.020

0 引言

教与学优化算法(teaching-learning-based optimization, TLBO)是 Rao 等^[1-2]于 2011 年提出的一种新的智能优化算法,它主要模拟老师的“教”过程和学生的“学”过程。由于教与学算法具有参数少、算法易于实现、寻优性能好等优点,受到学者的关注和研究^[3]。目前,教与学优化算法已被成功地应用于多个领域。

一般来说,小的种群规模(微种群)会加速优化算法收敛,但会使其全局探索能力下降^[4],即易陷入局部最优。因此,为提高微种群优化算法的全局探索能力,众多学者对其进行研究。例如, Ren 等^[5]提出一种小种群的差分进化算法(differential evolution using smaller population, DESP)。在该算法中,变异策略使用额外的扰动来提高其全局搜索能力,并使用一种自适应调整策略来控制扰动的强度。Brown 等^[6]提出一种微种群的 JADE(adaptive differential evolution with optional external archive)算法。该算法在 Rcr-JADE 算法^[7]的基础上进行改进,并加入了新的差分变异

策略来提高算法的全局搜索能力。Marquez-Grajales 等^[8]提出了 μJADEε 算法。该算法将 ε 约束方法用于 μJADE 算法,以此来提高 μJADE 算法求解约束优化问题的能力。Salehinejad 等^[9]提出一种向量化随机变异因子的微种群差分进化算法(micro-differential evolution with vectorized random mutation factor, MDEV)。在该算法中,每个个体均有各自的差分变异因子 F ,以此来提高种群多样性。上述研究表明,提高微种群启发式算法的全局搜索能力是避免算法“早熟收敛”的有效方式之一。

博弈论又称对策论,是研究具有冲突和对抗性质现象的数学理论和方法,在很多科学领域中得到了广泛的应用。在博弈问题中,非合作博弈纳什均衡是其最核心的研究内容。1951 年, Nash^[10]在提出的“纳什定理”中揭示并证明了博弈纳什均衡解的存在,但是 Nash 并没有给出求解纳什均衡的一般性方法。传统方法在求解纳什均衡问题时均存在一定的局限性,特别是对于高维的博弈模型(如 3 维及以上的矩阵策略),传统算法计算复杂且求解成本较高。最近几十年来,随

收稿日期:2019-09-13;修订日期:2019-11-12
基金项目:国家重点研发计划资助项目(2016YFC0800200);国家自然科学基金资助项目(61603244);中国博士后科学基金资助项目(2018M642017)
作者简介:范勤勤(1986—),男,浙江宁波人,上海海事大学副教授,博士,研究方向为进化计算、机器学习、超启发式算法等, E-mail: qqfan@shmtu.edu.cn。

着元启发式算法的迅速发展,诸多学者开始使用遗传算法^[11]、免疫算法^[12]、粒子群算法^[13]、蚁群算法^[14]等来求解博弈纳什均衡问题,并取得较好结果,这为求解非合作博弈问题提供了一种新的有效途径和方法。

为进一步提高微种群 TLBO 算法的全局搜索能力并有效地求解非合作博弈问题,笔者提出一种 OBL-μTLBO 算法。在该算法中,微种群有利于加快 TLBO 算法的收敛速度,而反向学习有助于提高 TLBO 算法的全局探索能力。实验结果表明,OBL-μTLBO 算法的整体性能明显优于其他所比较微种群和非微种群算法。最后,将改进的算法应用于非合作博弈问题的求解,所得结果明显好于其他比较算法。

1 反向学习

反向学习 (opposition-based learning, OBL) 是 Tizhoosh^[15] 于 2005 年提出的一种应用于计算智能领域的机器学习方法。因为反向学习策略具有较强的探索能力^[16],所以被广泛地用于提高元启发式算法的搜索性能。一些术语介绍如下。

反向数 (opposite number): 设存在实数 $x \in [a, b]$, 则与其对应的反向数 x^o 被定义为:

$$x^o = a + b - x, x \in [a, b]。 \quad (1)$$

反向点 (opposite point): 对于 D 维的搜索空间, 设 $X_i = (x_{i,1}, x_{i,2}, \dots, x_{i,D})$ 为其解向量空间内的一个点, 且 $x_{ij} \in [a_j, b_j] (j = 1, 2, \dots, D)$, 则与其对应的反向点表示为:

$$x_{ij}^o = a_j + b_j - x_{ij}。 \quad (2)$$

另外, Wang 等^[17] 于 2011 年提出一种广义的反向学习策略 (generalized opposition-based learning, GOBL), 其反向个体生成如下:

$$x_{ij}^{GO} = r \cdot (a_j + b_j) - x_{ij}, \quad (3)$$

式中: r 是被用来控制反向解的搜索范围。研究表明, GOBL 策略可以有效地使算法跳出局部最优^[18]。

2 基于反向学习的微种群教与学优化算法

2.1 基于反向学习的“教”阶段

对于微种群而言, 由于种群规模较小, 因此种群多样性更容易迅速丢失。故笔者将结合 GOBL 策略, 在教学阶段使用反向学习个体 X^{GO} 来指导种群进化, 从而提高微种群教与学优化算法的探索能力。其反向个体更新方式如下:

$$x_{ij}^{GO} = r \cdot (a_j + b_j) - x_{ij}, \quad (4)$$

式中: x_{ij}^{GO} 表示反向个体, $x_{ij} \in [x_{j,\min}, x_{j,\max}]$ 。

$$X_{i,\text{new}} = X_{i,\text{old}} + N(0.5, 0.2) \cdot (X_T - T_F \cdot X_M) + N(0.5, 0.2) \cdot (X^{GO} - X_{i,\text{old}}), \quad (5)$$

式中: N 表示正态分布函数。若 $X_{i,\text{new}}$ 的适应度值优于 $X_{i,\text{old}}$, 则更新个体; 否则, 不更新。

2.2 OBL-μTLBO 算法的实现步骤

Step 1 初始化。设定微种群规模 NP , 最大评价次数 FES ; 初始化种群。

Step 2 教学阶段。根据式 (5) 对个体进行更新, 并对个体进行选择。

Step 3 学习阶段。根据文献 [19] 利用正态分布随机数代替式 (2) 的均匀随机数, 其更新公式如下:

$$\begin{cases} X_{p,\text{new}} = X_{p,\text{old}} + N(0.5, 0.2) \cdot (X_p - X_l), \\ f(X_p) < f(X_l); \\ X_{p,\text{new}} = X_{p,\text{old}} + N(0.5, 0.2) \cdot (X_l - X_p), \\ f(X_l) < f(X_p)。 \end{cases} \quad (6)$$

Step 4 判定算法是否达到最大适应度评价次数, 若没有达到, 则跳转至 Step 2 继续执行; 如达到, 则执行 Step 5。

Step 5 输出结果。

3 仿真测试

为验证 OBL-μTLBO 算法的有效性, 笔者选取文献 [6] 中 13 个 30 维的测试函数。其中 $f_1 \sim f_7$ 为单峰函数, $f_8 \sim f_{13}$ 为多峰函数。OBL-μTLBO 算法性能分别与 TLBO^[1-2]、DESP^[5]、μJADE^[6]、Rcr-JADE-s4^[7]、MDEV^[9]、TLBO-CSWL^[19]、FSADE^[20]、ODE^[21] 以及 MDEV^[9]-Fajfar^[22] 等微种群或非微种群算法进行对比。对于每个测试函数, 所有比较算法均独立运行 50 次。同时, 为保证所得结论的可靠性, 仿真结果采用 Friedman^[23]、Bonferroni-Dunn^[24]、Iman-Davenport^[25]、Holm 和 Hochberg^[25] 检验进行统计分析。显著性水平设定为 5%。此外, 所得最好结果在表中用黑体表示。

3.1 与微种群算法的比较

在该实验中, 本文算法 OBL-μTLBO 与其他 5 种微种群算法进行比较, 并通过设定的求解精度来比较哪种算法所使用的适应度函数评价次数最小。对于所有比较算法, 实验参数设置均与文献 [19] 一致。即种群规模设定为 8, 对于 f_7 , 函数的结果精度设定为 $1.0E-02$, 其余测试函数的寻优精度设定为 $1.0E-08$, 最大适应度函数评价次数为 3 000 000。若算法在达到终止条件时仍然没

有求解到设定的求解精度,则认为算法在该函数上寻优失败,用“—”表示。仿真结果如表 1 所示。由表 1 可知,OBL- μ TLBO 算法在计算 f_1 f_2 f_3 f_4 f_5 f_6 f_7 f_8 f_9 f_{10} f_{11} f_{12} 上使用的函数评价次数最少,寻优效率明显优于其他所比较算法。这说明 OBL- μ TLBO 算法不仅可以加速收敛,还可以提高寻优性能。对于函数 f_8 和 f_{13} , μ JADE 算法的仿真结果略优于 OBL- μ TLBO 算法。其主要原因可能是:虽然反向学习能够有效地帮助 TLBO 提高全局搜索能力,但是差分

进化算法的全局搜索能力要好于 TLBO 算法。除以上比较外,还使用非参数的统计分析方法来对实验结果进行进一步分析,所得统计分析结果见表 2 和表 3。从表 2 中可知,OBL- μ TLBO 算法在所有算法中的寻优效率是最高的。从表 3 可以看出,OBL- μ TLBO 算法的寻优效率要显著优于 DESP、MDEVM、MDEVM-Fajfar 算法,OBL- μ TLBO 算法的整体性能要比 μ JADE 好。综上所述,OBL- μ TLBO 算法的收敛速度在所有比较算法中是最快的,且可得到高质量的解。

表 1 微种群算法的实验结果

Table 1 Experimental results of micro-population algorithms

函数	指标	MDEVM-Fajfar	MDEVM	DESP	μ JADE	OBL- μ TLBO
f_1	平均值	8. 6E+04	2. 8E+04	1. 9E+04	2. 2E+04	2. 6E+02
	标准方差	4. 1E+05	3. 3E+03	8. 0E+02	7. 8E+02	8. 9E+00
f_2	平均值	6. 0E+04	3. 1E+04	—	3. 7E+04	3. 9E+02
	标准方差	1. 5E+05	4. 7E+03	—	1. 2E+03	1. 1E+01
f_3	平均值	3. 4E+05	1. 4E+05	—	1. 6E+05	4. 3E+02
	标准方差	4. 7E+05	1. 5E+04	—	7. 7E+03	5. 3E+01
f_4	平均值	—	—	—	2. 2E+05	4. 3E+02
	标准方差	—	—	—	1. 6E+04	1. 2E+01
f_5	平均值	—	2. 3E+06	—	2. 1E+05	1. 8E+05
	标准方差	—	4. 0E+05	—	5. 5E+04	4. 3E+04
f_6	平均值	4. 1E+05	1. 8E+04	—	1. 2E+04	1. 2E+02
	标准方差	3. 1E+05	6. 3E+03	—	3. 8E+03	1. 0E+01
f_7	平均值	3. 4E+05	8. 9E+05	—	2. 3E+05	2. 1E+03
	标准方差	2. 2E+05	7. 1E+05	—	1. 6E+05	8. 1E+02
f_8	平均值	6. 9E+04	—	—	1. 0E+05	1. 9E+05
	标准方差	1. 6E+05	—	—	5. 1E+03	6. 7E+03
f_9	平均值	1. 6E+05	—	—	1. 2E+05	2. 7E+02
	标准方差	4. 0E+05	—	—	6. 8E+03	1. 3E+01
f_{10}	平均值	9. 5E+04	—	—	3. 8E+04	3. 9E+02
	标准方差	1. 0E+05	—	—	5. 0E+03	1. 0E+01
f_{11}	平均值	4. 0E+04	2. 6E+04	3. 1E+04	4. 6E+04	2. 8E+02
	标准方差	4. 7E+04	3. 1E+03	1. 7E+03	3. 7E+04	1. 3E+01
f_{12}	平均值	9. 5E+04	6. 6E+04	1. 8E+06	3. 8E+04	4. 1E+03
	标准方差	2. 3E+05	6. 5E+04	0. 0E+00	7. 5E+03	7. 8E+02
f_{13}	平均值	4. 2E+05	7. 0E+04	—	3. 2E+04	2. 5E+05
	标准方差	6. 8E+05	6. 4E+04	—	1. 8E+04	4. 8E+05

表 2 Friedman 测试在微种群算法上所得排序结果

Table 2 Ranking obtained by Friedman's test for micro-population algorithms

算法	排序
OBL- μ TLBO	1. 346 2
μ JADE	2. 346 2
MDEVM	3. 307 7
MDEVM-Fajfar	3. 653 8
DESP	4. 346 2

表 3 Bonferroni-Dunn、Holm 和 Hochberg 在 OBL- μ TLBO 和微种群算法上所得 p 值

Table 3 p -values obtained by Bonferroni-Dunn's, Holm's, and Hochberg's procedures for OBL- μ TLBO and micro-population algorithms

算法	Bonferroni-Dunn	Holm	Hochberg
DESP	5. 26E-06	5. 26E-06	5. 26E-06
MDEVM-Fajfar	7. 94E-04	5. 95E-04	5. 95E-04
MDEVM	6. 25E-03	3. 12E-03	3. 12E-03
μ JADE	4. 27E-01	1. 06E-01	1. 06E-01

3.2 与非微种群算法的比较

在本实验中,OBL-μTLBO 算法与 4 种非微种群算法的性能进行比较。期望求解精度同 3.1 节。标准 TLBO 算法的种群规模设定为 30; TLBO-CSWL 算法的种群规模设定为 40; FSADE 算法和 Rcr-JADE-s4 的种群规模分别设定为 50 和 100,算法参数设置均与文献[6]一致,结果见表 4。从表 4 可以看出,OBL-μTLBO 算法在 f_1 、 f_2 f_3 f_4 f_6 f_7 f_9 f_{10} f_{11} 上的平均评价次数均明显

少于所有比较算法,这说明所提方法具有较快的收敛速度。而在函数 f_5 f_8 f_{13} 上,Rcr-JADE-s4 算法的实验结果要优于本文的 OBL-μTLBO 算法。这说明 Rcr-JADE-s4 算法在这 3 个函数上具有较快的收敛速度。统计分析结果见表 5 和表 6。由表 5 可知,笔者所提算法与非微种群算法相比,OBL-μTLBO 同样具有较高的搜索效率。从表 6 可以看出,OBL-μTLBO 算法的搜索效率要明显优于其他所有比较的非微种群算法。

表 4 OBL-μTLBO 算法与非微种群算法的结果

Table 4 Experimental results of the OBL-μTLBO algorithm and non-micro-population algorithms

函数	指标	FSADE (NP = 50)	Rcr-JADE-s4 (NP = 100)	TLBO (NP = 30)	TLBO-CSWL (NP = 40)	OBL-μTLBO (NP = 8)
f_1	平均值	3. 5E+04	2. 1E+04	4. 3E+03	1. 9E+03	2. 6E+02
	标准方差	5. 3E+02	4. 9E+02	9. 1E+02	4. 2E+01	8. 9E+00
f_2	平均值	4. 5E+04	3. 3E+04	7. 6E+03	2. 8E+03	3. 9E+02
	标准方差	5. 7E+02	7. 6E+02	1. 3E+02	4. 6E+01	1. 1E+01
f_3	平均值	2. 6E+06	6. 7E+04	1. 7E+04	2. 9E+03	4. 3E+02
	标准方差	4. 8E+05	3. 6E+03	8. 6E+02	1. 6E+02	5. 3E+01
f_4	平均值	2. 2E+05	1. 0E+05	7. 2E+03	3. 2E+03	4. 3E+02
	标准方差	4. 2E+03	3. 5E+04	1. 5E+02	5. 1E+01	1. 2E+01
f_5	平均值	2. 8E+06	7. 8E+04	—	4. 1E+05	1. 8E+05
	标准方差	6. 1E+04	2. 7E+03	—	5. 8E+04	4. 3E+04
f_6	平均值	1. 3E+04	8. 8E+03	1. 7E+03	1. 1E+03	1. 2E+02
	标准方差	4. 7E+02	3. 6E+02	2. 2E+02	2. 1E+02	1. 0E+01
f_7	平均值	1. 1E+05	2. 2E+04	—	2. 2E+04	2. 1E+03
	标准方差	2. 6E+04	9. 8E+03	—	6. 4E+03	8. 1E+02
f_8	平均值	4. 4E+04	7. 0E+04	—	4. 5E+05	1. 9E+05
	标准方差	1. 8E+03	2. 8E+03	—	8. 2E+03	6. 7E+03
f_9	平均值	8. 9E+04	9. 1E+04	—	7. 9E+03	2. 7E+02
	标准方差	4. 8E+03	1. 3E+03	—	8. 6E+03	1. 3E+01
f_{10}	平均值	4. 8E+04	3. 1E+04	2. 3E+04	2. 8E+03	3. 9E+02
	标准方差	5. 0E+02	6. 4E+02	1. 3E+02	4. 4E+01	1. 0E+01
f_{11}	平均值	3. 9E+04	2. 2E+04	4. 1E+03	2. 1E+03	2. 8E+02
	标准方差	3. 7E+03	6. 4E+02	3. 9E+02	5. 8E+02	1. 3E+01
f_{12}	平均值	4. 0E+04	1. 9E+04	—	5. 0E+04	4. 1E+03
	标准方差	1. 0E+03	5. 5E+02	—	2. 5E+03	7. 8E+02
f_{13}	平均值	4. 1E+04	2. 1E+04	—	3. 8E+05	2. 5E+05
	标准方差	1. 1E+03	4. 9E+02	—	6. 1E+04	4. 8E+03

3.3 OBL-μTLBO 与 μJADE、TLBO-CSWL、ODE 的比较

为进一步验证 OBL-μTLBO 算法的寻优性能,本部分将对 OBL-μTLBO 算法与表现较好的微种群算法 μJADE 和非微种群算法 TLBO-CSWL、ODE 进行性能比较。与 3.1 和 3.2 节不同,在本实验中,所有算法终止条件设定为最大的适应度函数评

价次数,即 30 000。同时,μJADE、ODE、TLBO-CSWL 的种群规模分别设定为 8、100、40。所得结果见表 7。从表 7 可以看出,OBL-μTLBO 算法在 f_1 f_2 f_3 f_4 f_5 f_6 f_7 f_9 f_{10} f_{11} f_{12} 上得到的平均结果明显优于 μJADE 算法。而 μJADE 只在 f_8 和 f_{13} 这两个函数上要比所提算法表现得好。这足以说明 OBL-μTLBO 算法的寻优效率要好于 μJADE 算

表 5 Friedman 测试在 OBL-μTLBO 和非微种群
算法上所得排序结果

Table 5 Ranking obtained by Friedman's test
for OBL-μTLBO and non-micro-population algorithms

算法	排序
OBL-μTLBO	1. 384 6
TLBO-CSWL	2. 756 9
R cr-JADE-s4	3. 115 4
TLBO	3. 923 1
FSADE	4. 000 0

表 6 Bonferroni-Dunn、Holm 和 Hochberg 在
OBL-μTLBO 和非微种群算法上所得 p 值

Table 6 p -values obtained by Bonferroni-Dunn's,
Holm's, and Hochberg's procedures for OBL-
μTLBOand non-micro-population algorithms

算法	Bonferroni- Dunn	Holm	Hochberg
FSADE	9. 90E-05	9. 90E-05	9. 90E-05
TLBO	1. 70E-04	1. 28E-04	1. 28E-04
R _{cr} -JADE-s4	2. 10E-02	1. 05E-02	1. 05E-02

表 7 所有比较算法 30 维仿真测试结果

Table 7 Experimental results on all Comparison algorithms with 30 D

函数	指标	μJADE ($NP=8$)	TLBO-CSWL ($NP=40$)	ODE ($NP=100$)	OBL-μTLBO ($NP=8$)
f_1	平均值	9. 0E-13	4. 0E-223	1. 1E+02	0. 0E+00
	标准方差	2. 5E-12	0. 0E+00	3. 2E+01	0. 0E+00
f_2	平均值	1. 3E-06	9. 8E-141	2. 3E+01	0. 0E+00
	标准方差	1. 6E-06	6. 5E-140	8. 6E+00	0. 0E+00
f_3	平均值	9. 7E+02	2. 8E-221	9. 6E+03	0. 0E+00
	标准方差	4. 2E+02	0. 0E+00	2. 2E+03	0. 0E+00
f_4	平均值	4. 9E+00	5. 5E-224	2. 8E+01	0. 0E+00
	标准方差	2. 5E+00	0. 0E+00	4. 3E+00	0. 0E+00
f_5	平均值	4. 3E+01	2. 8E+01	7. 6E+03	2. 8E+01
	标准方差	2. 9E+01	4. 1E-01	4. 2E+03	2. 78E-01
f_6	平均值	2. 0E-02	9. 2E-16	1. 2E+02	0. 0E+00
	标准方差	1. 4E-01	1. 5E-15	3. 5E+01	0. 0E+00
f_7	平均值	4. 6E-02	0. 0E+00	8. 3E-02	4. 4E-05
	标准方差	1. 4E-02	0. 0E+00	2. 7E-02	3. 2E-05
f_8	平均值	4. 0E+03	0. 0E+00	7. 7E+03	6. 4E+03
	标准方差	4. 5E+02	0. 0E+00	3. 4E+02	1. 3E+03
f_9	平均值	1. 2E+02	0. 0E+00	2. 3E+02	0. 0E+00
	标准方差	1. 4E+01	0. 0E+00	1. 3E+01	0. 0E+00
f_{10}	平均值	2. 3E-02	4. 9E+03	4. 5E+00	2. 1E-16
	标准方差	1. 6E-01	1. 1E+03	4. 2E-01	8. 5E-16
f_{11}	平均值	2. 1E-03	0. 0E+00	1. 9E+00	0. 0E+00
	标准方差	4. 8E-03	0. 0E+00	3. 0E-01	0. 0E+00
f_{12}	平均值	3. 2E-01	3. 2E-219	7. 8E+00	8. 3E-02
	标准方差	6. 1E-01	0. 0E+00	2. 7E+00	9. 7E-02
f_{13}	平均值	2. 6E-03	2. 9E+01	2. 3E+01	2. 6E+00
	标准方差	1. 4E-02	6. 30E-02	2. 6E+01	5. 3E-01

法。相比于 TLBO-CSWL,除测试函数 f_7 、 f_8 、 f_{12} 外, OBL-μTLBO 算法在所有函数上都要比它表现得好。此外, OBL-μTLBO 算法在所有测试函数上的优化结果都要好于 ODE。综上, OBL-μTLBO 算法不仅具有较快的收敛速度和能够降低计算内存的要求,而且还有较强的全局搜索能力。

为得到可靠结论,3 种非参数统计分析方法

被用来对所有比较算法的平均值进行统计分析,其结果见表 8。由表 8 可知, OBL-μTLBO 算法的性能要明显优于 μJADE 和 ODE。另外,虽然 OBL-μTLBO 算法和 TLBO-CSWL 算法的性能在统计学意义上不存在显著性的差异,但是从表 7 中的结果可知, OBL-μTLBO 算法的整体性能要优于 TLBO-CSWL 算法。

表 8 Bonferroni-Dunn、Holm 在 OBL-μTLBO 和其他 3 种算法上所得 *p* 值

Table 8 <i>p</i> -values obtained by Bonferroni-Dunn's, Holm's, and Hochberg's procedures for OBL-μTLBO and three algorithms			
比较算法	Bonferroni-Dunn (<i>p</i> 值)	Holm (<i>p</i> 值)	Hochberg (<i>p</i> 值)
μJADE	3.66E-02	2.44E-02	2.44E-03
ODE	5.12E-06	5.12E-06	5.12E-06
TLBO-CSWL	6.73E-01	2.24E-01	2.24E-01

3.4 种群规模的敏感性分析

本部分对 OBL-μTLBO 算法的种群规模进行敏感性分析。其种群规模分别设定为 5、8、10、15、20、30、40、50、100。对于每个测试函数,算法终止条件设定为最大适应度函数评价次数 30 000,并使用 13 个 30 维的测试函数来对不同的种群规模进行分析。同时,Friedman 统计方法用来对实验结果进行统计分析,结果见表 9。从表 9 可知,种群规模对 OBL-μTLBO 算法的性能没有显著影响。因此,算法使用者可以对 OBL-μTLBO 算法的种群规模进行方便设置。此外,从 Friedman 的排序结果(见图 1)来看,在有限的计算资源且要保证算法良好寻优能力的情况下,较小的种群规模更有利于 OBL-μTLBO 算法的搜索。

表 9 OBL-μTLBO 算法的种群规模敏感性分析
Table 9 Sensitivity analysis to population size of OBL-μTLBO

Friedman 值	χ^2 值	<i>p</i> 值
1.11	9.49	8.93E-01

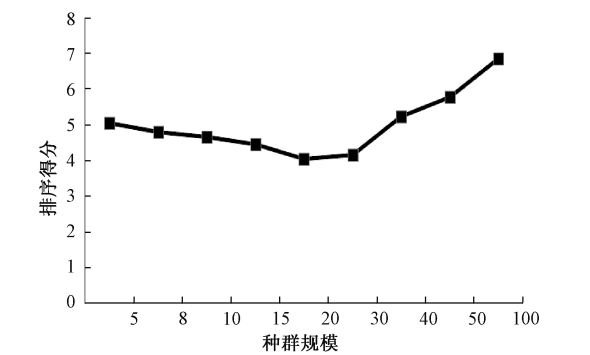


图 1 Friedman 测试排序结果

Figure 1 Ranking obtained by Friedman's test
4 OBL-μTLBO 算法在非合作博弈问题中的应用

4.1 博弈问题描述

假设有 *n* 个局中人参与博弈,所有局中人策

略构成一个策略组合(strategy profile),如果某情况下无一参与者可以独自行动而增加收益(即为了自身利益的最大化,没有任何单独的一方愿意改变其策略),此时博弈模型处于稳定的状态,此策略组合被称为纳什均衡。参考文献[13]中的问题描述,对于两人有限非合作博弈问题:设局中人 1 的混合策略 $\mathbf{x}=(x_1, x_2, \cdots, x_m)$,局中人 2 的混合策略 $\mathbf{y}=(y_1, y_2, \cdots, y_n)$ 。 $\mathbf{A}_{m \times n}$ 、 $\mathbf{B}_{m \times n}$ 分别为局中人 1 和局中人 2 的支付矩阵,则局中人 1 和局中人 2 的期望支付分别为 $\mathbf{x}\mathbf{A}\mathbf{y}^T$ 和 $\mathbf{x}\mathbf{B}\mathbf{y}^T$ 。其中, $(\mathbf{x}^*, \mathbf{y}^*)$ 为双矩阵博弈问题的一个纳什均衡解的充分必要条件为:

$$\begin{cases} \mathbf{x}^* \mathbf{A} \mathbf{y}^{*T} \geq \mathbf{x} \mathbf{A} \mathbf{y}^{*T}, & \forall \mathbf{x}; \\ \mathbf{x}^* \mathbf{B} \mathbf{y}^{*T} \geq \mathbf{x} \mathbf{B} \mathbf{y}^{*T}, & \forall \mathbf{y}. \end{cases} \quad (7)$$

算法中的每个个体的取值表示所有局中人的混合策略,则双矩阵博弈问题 $\mathbf{z}=(\mathbf{x}, \mathbf{y})$ 的适应度函数可以表示为:

$$f(\mathbf{z}) = \max \left\{ \max_{0 \leq i \leq n} (\mathbf{A}_i \mathbf{y}^T - \mathbf{x} \mathbf{A} \mathbf{y}^T), 0 \right\} + \max \left\{ \max_{0 \leq j \leq m} (\mathbf{x} \mathbf{B}_j - \mathbf{x} \mathbf{B} \mathbf{y}^T), 0 \right\}, \quad (8)$$

式中: \mathbf{A}_i 表示 $\mathbf{A}_{m \times n}$ 的第 *i* 行; \mathbf{B}_j 表示 $\mathbf{B}_{m \times n}$ 的第 *j* 列。

根据纳什均衡的定义和性质^[10]可知,混合局势 $\mathbf{z}^*=(\mathbf{x}^*, \mathbf{y}^*)$ 为双矩阵博弈问题的一个纳什均衡解的充分必要条件为:存在 $\mathbf{z}^*=(\mathbf{x}^*, \mathbf{y}^*)$,使得 $f(\mathbf{z}^*)=0$,且对于任意的 $\mathbf{z} \neq \mathbf{z}^*$,都有 $f(\mathbf{z})>0$ 。

4.2 案例研究

选取两个双矩阵博弈问题^[26-27]。

例 1:博弈模型 1, $\Gamma_1 \equiv (\mathbf{x}_1, \mathbf{y}_1; \mathbf{A}_1, \mathbf{B}_1)$, 支付矩阵如下:

$$\mathbf{A}_1 = \begin{pmatrix} 1 & 2 & 0 \\ 0 & 1 & 2 \\ 2 & 0 & 1 \end{pmatrix}, \quad \mathbf{B}_1 = \begin{pmatrix} 1 & 0 & 2 \\ 2 & 1 & 0 \\ 0 & 2 & 1 \end{pmatrix},$$

该问题的唯一纳什均衡解为 (1/3, 1/3, 1/3; 1/3, 1/3, 1/3)。

例 2:博弈模型 2, $\Gamma_2 \equiv (\mathbf{x}_2, \mathbf{y}_2; \mathbf{A}_2, \mathbf{B}_2)$, 支付矩阵如下:

$$\mathbf{A}_2 = \begin{pmatrix} 0 & 4 & 5 \\ 4 & 0 & 5 \\ 3 & 3 & 6 \end{pmatrix}, \quad \mathbf{B}_2 = \begin{pmatrix} 4 & 0 & 3 \\ 0 & 4 & 3 \\ 5 & 5 & 6 \end{pmatrix},$$

该问题的唯一纳什均衡解为 (0, 0, 1; 0, 0, 1)。

为了验证微种群算法 OBL-μTLBO 求解非合作博弈纳什均衡问题的有效性,将 OBL-μTLBO

算法和测试中性能较好的 TLBO-CSWL^[19] 算法应用于求解以上两个 3 维双矩阵博弈问题,并与其他几种知名算法 jDE^[28]、SaDE^[29]、CLPSO^[30] 进行性能比较。其中 OBL- μ TLBO 的种群规模设定为 8,而其余所有对比算法的参数设定均与对比文献[6]一致。求解结果使用 Wilcoxon 秩和检验^[31]

所有算法的求解结果(平均值和标准方差,括号内是标准方差)以及统计分析结果见表 10。从表 10 中可以看出,OBL- μ TLBO 算法在两个

博弈实例上所得平均值均要好于所比较算法。表 11 和表 12 为所有算法在算例 1 和算例 2 上得到的最好结果。从表 11 和 12 可以看出,OBL- μ TLBO 算法在两个算法上均可以取得最优的纳什均衡解,且求解的适应度函数精度明显优于 jDE、SaDE、CLPSO 以及 TLBO-CSWL 算法。

以上分析结果表明,所提出的 OBL- μ TLBO 算法能够有效地求解非合作博弈纳什均衡问题。

表 10 所有算法在博弈问题上得到的实验结果
Table 10 Experimental results of all algorithms on game problems

博弈模型	jDE	SaDE	CLPSO	TLBO-CSWL	OBL- μ TLBO
博弈模型 1	3.47E-04(1.81E-03)	4.31E-03(1.52E-03)	5.93E-02(2.96E-02)	1.56E-03(3.70E-03)	4.54E-06(2.16E-05)
博弈模型 2	6.65E-04(4.11E-04)	1.00E-03(4.31E-04)	1.24E-01(6.72E-02)	3.86E-10(3.28E-10)	4.56E-13(6.71E-13)

表 11 所有算法在博弈算例 1 上得到的最好结果
Table 11 The best experimental results of all algorithms on game problem 1

算法	局中人 1 混合策略	局中人 2 混合策略	适应度函数
jDE	(0.332 8,0.333 8,0.333 4)	(0.333 4,0.333 2,0.333 4)	8.524 5E-04
SaDE	(0.332 7,0.333 7,0.333 6)	(0.333 4,0.332 6,0.334 0)	1.816 8E-03
CLPSO	(0.327 0,0.337 4,0.335 6)	(0.335 0,0.320 0,0.345 1)	2.365 3E-02
TLBO-CSWL	(0.333 3,0.333 3,0.333 3)	(0.333 3,0.333 3,0.333 3)	1.900 2E-06
OBL- μ TLBO	(0.333 3,0.333 3,0.333 3)	(0.333 3,0.333 3,0.333 3)	8.542 1E-08

表 12 所有算法在博弈算例 2 上得到的最好结果
Table 12 The best experimental results of all algorithms on game problem 2

算法	局中人 1 混合策略	局中人 2 混合策略	适应度函数
jDE	(0.000 0,0.000 1,0.999 9)	(0.000 0,0.000 0,1.000 0)	1.026 2E-04
SaDE	(0.000 0,0.000 1,0.999 9)	(0.000 0,0.000 1,0.999 9)	3.482 5E-04
CLPSO	(0.012 9,0.000 1,0.987 0)	(0.003 9,0.013 4,0.982 7)	3.034 9E-02
TLBO-CSWL	(0.000 0,0.000 0,1.000 0)	(0.000 0,0.000 0,1.000 0)	3.537 4E-11
OBL- μ TLBO	(0.000 0,0.000 0,1.000 0)	(0.000 0,0.000 0,1.000 0)	3.558 2E-15

5 结论

为平衡微种群教与学优化算法的全局和局部搜索能力,笔者提出了 OBL- μ TLBO 算法。在该算法中,利用微种群来提高教与学优化算法的收敛速率,且在教学阶段加入了反向学习来指导种群进化,使算法具备反向学习的能力,提高了算法的全局探索能力。OBL- μ TLBO 算法跟 μ JADE、DESP、MDEVIM、MDEVIM-Fajfar 等微种群算法和 FSADE、Rcr-JADE-s4、TLBO、TLBO-CSWL、ODE 等非微种群算法进行性能比较,结果表明,OBL- μ TLBO 算法的整体性能在所有比较算法中是最好的。因此,反向学习策略对于提升微种群算法的优化性能是有效的。另外,对 OBL- μ TLBO

算法的种群规模进行敏感性分析。统计分析结果表明,OBL- μ TLBO 算法的性能不受种群大小影响,但在有限计算资源下,小的种群规模更有利于其搜索。最后,将 OBL- μ TLBO 算法应用到非合作博弈问题,仿真结果表明,所提 OBL- μ TLBO 算法能够取得较好的优化结果。

参考文献:

[1] 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.
[2] 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.
- [3] 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.
- [4] MALLIPEDDI R, SUGANTHAN P N. Empirical study on the effect of population size on differential evolution algorithm[C]//IEEE Congress on Evolutionary Computation. Hong Kong:IEEE,2008: 3663-3670.
- [5] REN X, CHEN Z Z, MA Z. Differential evolution using smaller population [C]//Second International Conference on Machine Learning and Computing. Bangalore:IEEE,2010: 76-80.
- [6] BROWN C, JIN Y C, LEACH M, et al. μ JADE: adaptive differential evolution with a small population [J]. Soft computing, 2016, 20(10): 4111-4120.
- [7] GONG W Y, CAI Z H, WANG Y. Repairing the crossover rate in adaptive differential evolution [J]. Applied soft computing, 2014, 15: 149-168.
- [8] MARQUEZ-GRAJALES A, MEZURA-MONTES E. μ JADE ϵ : micro adaptive differential evolution to solve constrained optimization problems [C]// IEEE Congress on Evolutionary Computation. Vancouver: IEEE, 2016: 4183-4190.
- [9] SALEHINEJAD H, RAHNAMAYAN S, TIZHOOSH H R, et al. Micro-differential evolution with vectorized random mutation factor[C]// IEEE Congress on Evolutionary Computation. Beijing: IEEE, 2014: 2055-2062.
- [10] NASH J. Non-cooperative games [J]. The annals of mathematics, 1951, 54(2):286.
- [11] 陈士俊,孙永广,吴宗鑫.一种求解 NASH 均衡解的遗传算法[J].系统工程,2001, 19(5):67-70.
- [12] 邱中华,高洁,朱跃星.应用免疫算法求解博弈问题[J].系统工程学报,2006, 21(4):398-404.
- [13] 贾文生,向淑文,杨剑锋,等.基于免疫粒子群算法的非合作博弈 Nash 均衡问题求解[J]. 计算机应用研究, 2012, 29(1):28-31.
- [14] 王志勇,韩旭,许维胜,等.基于改进蚁群算法的纳什均衡求解[J].计算机工程, 2010, 36(14):166-168,171.
- [15] TIZHOOSH H R. Opposition-based learning: a new scheme for machine intelligence [C]// International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce.Vienna:IEEE,2005.
- [16] WANG W J, WANG H, SUN H, et al. Using opposition-based learning to enhance differential evolution: a comparative study [C]// IEEE Congress on Evolutionary Computation. Vancouver: IEEE, 2016: 71-77.
- [17] WANG H, WU Z J, RAHNAMAYAN S, et al. Enhancing particle swarm optimization using generalized opposition-based learning [J]. Information sciences, 2011, 181(20): 4699-4714.
- [18] CHEN X, YU K J, DU W L, et al. Parameters identification of solar cell models using generalized oppositional teaching learning based optimization [J]. Energy, 2016, 99:170-180.
- [19] 柳娅西子,范勤勤,胡志华.基于混沌搜索和权重学习的教与学优化算法及其应用[J]. 智能系统学报, 2018, 13(5):818-828.
- [20] SHARMA H, SHRIVASTAVA P, BANSAL J C, et al. Fitness based selfadaptive differential evolution[M]// Nature Inspired Cooperative Strategies for Optimization (NISCO 2013). Cham: Springer International Publishing, 2014:71-84.
- [21] RAHNAMAYAN S, TIZHOOSH H R, SALAMA M M A. Opposition-based differential evolution [J]. IEEE transactions on evolutionary computation, 2008, 12(1): 64-79.
- [22] FAJFAR I, TUMA T, PUHAN J, et al. Towards smaller populations in differential evolution[J]. Journal of microelectronics, electronic components and materials, 2012, 42(3):152-163.
- [23] FRIEDMAN M. The use of ranks to avoid the assumption of normality implicit in the analysis of variance [J]. Journal of the american statistical association, 1939, 34(205):109.
- [24] DUNN O J. Multiple comparisons among means[J]. Journal of the american statistical association, 1961, 56(293):52-64.
- [25] GARCÍA S, MOLINA D, LOZANO M, et al. A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the CEC' 2005 special session on real parameter optimization[J]. Journal of heuristics, 2009, 15(6): 617-644.
- [26] WEIBULL J W. Evolutionary game theory[M]. Cambridge: MIT Press, 1997.
- [27] WOLPERT D H, MACREADY W G. No free lunch theorems for optimization [J]. IEEE transactions on evolutionary computation, 1997, 1(1): 67-82.
- [28] BREST J, GREINER S, BOSKOVIC B, et al. Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems [J]. IEEE transactions on evolutionary computation, 2006, 10(6):646-657.

[29] QIN A K, HUANG V L, SUGANTHAN P N. Differential evolution algorithm with strategy adaptation for global numerical optimization [J]. IEEE transactions on evolutionary computation, 2009, 13(2):398-417.

[30] LIANG J J, QIN A K, SUGANTHAN P N, et al. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions [J]. IEEE transactions on evolutionary computation, 2006, 10(3):281-295.

[31] WILCOXON F. Individual comparisons by ranking methods [J]. Biometrics bulletin, 1945, 1(6):80.

Opposition-based Learning Teaching-learning-based Optimization Algorithm
with a Micro Population and Its Application

FAN Qinqin^{1, 2}, LIU Dixizi¹, WANG Xiaowei¹, HAN Xin³, WANG Weili¹

(1.Logistics Research Center, Shanghai Maritime University, Shanghai 201306, China; 2.School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai 200237, China; 3.Shanghai Institute of Disaster Prevention and Relief, Tongji University, Shanghai 200092, China)

Abstract: To improve the convergence speed and reliability of Teaching-learning-based optimization (TLBO) algorithm, an opposition-based learning TLBO with a micro population (OBL-μTLBO) was proposed in the current study. In the proposed algorithm, a small population size (i.e., micro population) was used to speed up the convergence of TLBO and was able to reduce the computer memory requirements. Moreover, a “teacher” achieved by an opposition-based learning in the proposed algorithm was utilized to improve the global exploration capability and avoid trapping into local optimum region. Simulation results indicated that OBL-μTLBO not only had better overall performance, but also had more quick convergence speed when compared with other competitors. Finally, OBL-μTLBO was used to solve two Nash equilibrium problems of non-cooperative game, and satisfactory results were achieved.

Key words: teaching-learning-based optimization; micro population; opposition-based learning; non-cooperative game