

文章编号: 2095-2163(2020)11-0073-06

中图分类号: TP391

文献标志码: A

基于竞争侵略策略的粒子群算法

安宁, 张军, 季伟东

(哈尔滨师范大学 计算机科学与信息工程学院, 哈尔滨 150025)

摘要: 针对粒子群优化算法(Particle Swarm Optimization, PSO)容易陷入局部最优及算法收敛速度慢的问题, 提出基于竞争侵略策略的粒子群算法(Particle Swarm Optimization based on competitive aggression Strategy, CAPSO)。凭借高自由度的侵略性, 自由粒子群中粒子与择优进化群中粒子进行比较, 得出竞争最优, 进而通过竞争池指导择优进化群更新, 使CAPSO算法快速跳出局部最优, 且提高算法收敛速度。使用8个标准测试函数分别对4个算法以及CAPSO算法进行仿真, 对寻优结果进行分析。结果表明, CAPSO算法无论是在单峰函数问题还是多峰函数问题上, 总体拥有优秀的寻优结果。

关键词: 群体智能; 竞争侵略; 优化算法; 粒子群

Particle Swarm Optimization based on competitive aggression Strategy

AN Ning, ZHANG Jun, JI Weidong

(Department of Computer Science and Information Engineering, Harbin Normal University, Harbin 150025, China)

[Abstract] In order to solve the problem that particle swarm optimization (PSO) algorithm is easy to fall into local optimization and slow convergence speed, a particle swarm optimization algorithm based on competitive aggression strategy is proposed. By comparing aggressive free particles with high degrees of freedom with optimized particle swarm optimization particles, competitive optimization is obtained, and then particle swarm optimization update is guided by competitive pool, which makes CAPSO algorithm jump out of local optimization quickly. And improve the convergence speed of the algorithm. Eight standard function tests are used to simulate the four algorithms and the CAPSO algorithm, and the optimization results are analyzed. The results show that the CAPSO algorithm has excellent optimization results in both unimodal function problems and multi-peak function problems.

[Key words] swarm intelligence; competitive aggression; optimization algorithm; particle swarm

0 引言

群体智能是模拟自然界生物群体行为而构造的随机优化算法, 它为解约束性、非线性和求极值等问题提供了新的思路。遗传算法^[1]、蚁群算法^[2]与粒子群优化(Particle Swarm Optimization, PSO)算法^[3]是群体智能中重要的3种方法。其中PSO算法概念简明、易于实现、参数设置少, 是一种高效的搜索算法。但在算法迭代后期, 粒子群算法容易陷入局部最优且收敛速度较慢。为了提高算法性能, 涌现出不同思想的优化策略。在基本粒子群算法进化方程的改进方面, Shi Y^[4]在惯性权重的基础上, 提出模糊系统动态调整惯性权重策略; Taherkhani M^[5]等根据每个粒子与最优位置的距离和粒子性能, 赋予每个粒子在不同维度上的惯性权重, 实现了惯性权重的动态调整。惯性权重的调整可以平衡算法全局搜索与局部搜索能力, 但对于提升算法后期, 收敛速度仍有欠缺。胡旺^[6]等将简化粒子速度更

新策略应用于一种更简化而高效的粒子群优化算法。该算法删除粒子速度项, 将二阶微分方程简化为一阶微分方程, 大幅度的减少了算法的运行时间。同样, 在改变速度更新策略方面, El-Sherbiny^[7]提出无速度方程的粒子群优化算法。该算法中每个粒子的新位置直接由个体与群体最佳位置的不同线性组合来确定, 用较少的迭代次数寻到较优的解。在借鉴其它优化策略方面, Clerc^[8]等为保证算法的收敛效果, 在粒子群算法中引入收缩因子, 但收缩因子能力有限, 不能较好的平衡全局搜索和局部搜索。为了解决该问题, Wang F^[9]等将自适应学习策略应用到粒子群算法中, 提出自主学习的候选生成策略与竞争学习的预测策略, 同时制定了一种自适应的寻优方向调整机制, 更好的平衡全局搜索能力和局部搜索能力。该算法中竞争学习的预测策略提供了一种粒子间竞争学习的思路。随着多种群策略的提出, 衍生出大量多种群协同寻优的改进算法。2018

基金项目: 国家自然科学基金(31971015); 哈尔滨市科技局科技创新人才研究专项资助项目(2017RAQXJ050); 哈尔滨师范大学硕士研究生创新科研资助项目(HSDSSCX2019-08)。

作者简介: 安宁(1995-), 男, 硕士研究生, 主要研究方向: 大数据、粒子群算法; 张军(1964-), 男, 硕士, 教授, 主要研究方向: 网络安全; 季伟东(1978-), 男, 博士, 教授, 主要研究方向: 大数据、群体智能。

收稿日期: 2020-09-30

年,邓先礼^[10]等在多种群同步进化的基础上,赋予子种群不同的搜索特性,利用各子种群性能的周期性评价指导个体进行迁移,实现种群间的合作;Niu B^[11]在种群合作的基础上,提出种群内部和种群之间两种社会活动,在进化过程中粒子不仅与自身种群的其他粒子交换社会经验,而且还受到其他种群粒子经验的影响,该策略增强了粒子的学习能力。文献^[12]将反向学习策略与局部学习策略结合,当算法陷入局部最优时,利用较差粒子的位置信息指导部分粒子以较快的飞行速度进行反向学习,跳出局部最优;罗强^[13]等将反向学习策略与最优粒子策略结合,利用全局最优粒子进行反向学习,通过全局最优粒子的更新引领粒子向更好的位置飞行,加强算法多样性的同时拥有较高的收敛速度。为提高算法的收敛速度与摆脱局部最优的能力,各种算法提出了不同的改进策略,文献中的算法在提高算法收敛速度或摆脱局部最优状态过程中,都存在一方面的效果比较显著,但在另一方面表现稍差的现象。

本文提出一种基于竞争侵略策略的粒子群算法(CAPSO)。受最优粒子策略与竞争学习策略启发,将种群分类、最优粒子与竞争学习思想应用于算法中,定义了侵略性自由粒子和竞争池两个概念。侵略性自由粒子与竞争池挑选竞争最优粒子,利用竞争最优粒子简化传统速度公式并指导择优进化群更新。为验证CAPSO算法在提高算法收敛速度与摆脱局部最优能力方面的性能,将其与4种经典算法(RLPSO^[12]算法、TDDALFPSO^[14]算法、ScDCPS算法^[15]、IPSO算法^[16],)进行对比。其中,RLPSO算法在最优粒子与竞争学习方面进行改进;TDDALFPSO算法在自适应学习方面进行改进;ScDCPS算法在种群分类与种群控制方面进行改进;IPSO算法在粒子学习机制上采用了竞争策略,与CAPSO算法形成多方面的对比。在8个标准测试函数下进行仿真实验,结果表明CAPSO算法具有较快的收敛速度和较高的求解精度。

1 基于竞争侵略策略的粒子群算法(CAPSO)

CAPSO算法将种群划分为规模相同的二部分:一部分为侵略性自由粒子群,另一部分为择优进化群。二部分粒子遵循不同的更新规则,侵略性自由粒子群遵循高自由度参数更新原则,择优进化群遵循改进速度公式进行更新。在CAPSO算法的每次迭代过程中,侵略性自由粒子群中随机选择一个粒子与择优进化群进行一次适应度值竞争,获胜的粒子作为竞争最优(Cbest)存入竞争池中,并利用该粒

子指导择优进化群进行下一次迭代。此策略增强了算法摆脱局部最优的能力,且算法的收敛速度有较大提高。近年来也涌现出很多结合竞争策略的粒子群优化算法。如2015年,Cheng R^[17]提出的大规模优化的竞争群优化算法,在所有粒子中随机挑选两个粒子成对进行竞争学习,失败的粒子直接向获胜的粒子进行学习,在粒子更新的过程中个体最优与全局最优不参与更新过程;2019年,刘明^[18]等将竞争策略与周期性结合,即每隔一定迭代次数运用一次竞争学习机制,竞争机制的原理与Cheng R提出的竞争原理相似,同样是竞争失败的粒子向获胜的粒子学习。上述二种算法中均省去了历史位置的记录,算法中个体最优与全局最优历史位置仍存在,但不参与算法迭代。本文提出的CAPSO算法中,已经完全摒弃了个体最优与全局最优的概念,利用竞争最优粒子直接指导择优进化群更新。在竞争策略方面,CAPSO算法将所有粒子分为不同更新规则的二个种群,与上述二种算法竞争策略的不同之处为:

- (1) CAPSO算法为二个不同种群间的粒子竞争,而不是同种群内相同更新规则的粒子间竞争。
- (2) 获胜的粒子定义为竞争最优,指导择优进化群的更新,而不是指导失败粒子进行更新。

1.1 竞争池

竞争池存储竞争最优(Cbest)粒子的位置信息,空间大小为 $N/2$ 。竞争最优指每一代更新过程中,侵略性自由粒子与择优进化群中的粒子进行适应度值的竞争,获胜的粒子定义为竞争最优,并将位置信息保存在竞争池中。竞争最优存入竞争池后,择优进化群的下一代更新需在竞争池中取出当代竞争最优的位置信息,来指导择优进化群进行下一次更新。位置信息取出后,竞争池中仍保存每代竞争最优的位置信息。为保证择优进化群第一次更新,初始所有粒子后,选择适应度值最优的粒子作为第一个竞争最优并存入竞争池。由于竞争最优的选择机制,保证了竞争最优的粒子信息为每一代全部粒子的最优信息。竞争池的存在为改进速度公式打下基础,将传统的粒子群算法中的个体最优和全局最优用竞争最优替代,以达到简化传统速度公式的目的。图1及图2为竞争最优的选择与指导择优进化群的二维示意图。

图中:U是解空间;三角形和圆形分别代表侵略性自由粒子和择优进化群中的粒子;Cbest是竞争最优;五角星是需要寻找的理论解。

图1中,侵略性自由粒子距离理论解较近,代表

适应度值更优,故侵略性自由粒子作为竞争最优并存入竞争池中。相反,若择优进化群的粒子距离理论解较近,则择优进化群的粒子作为竞争最优并存入竞争池中;图 2 中,①为在竞争池中取出竞争最优的位置信息,②为竞争最优指导择优进化群更新。

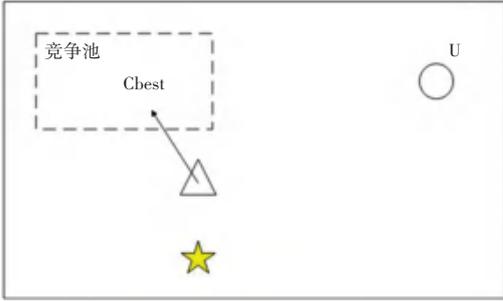


图 1 竞争最优入池示意图

Fig. 1 Schematic diagram of competition for optimal entry into the pool

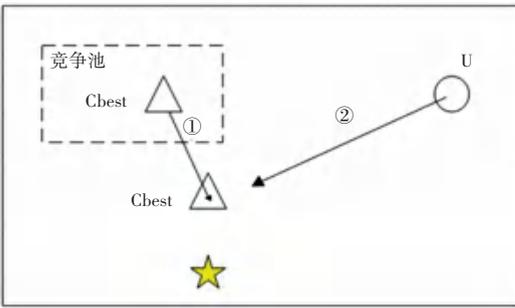


图 2 竞争最优出池示意图

Fig. 2 Schematic diagram of competitive optimal pool

1.2 侵略性自由粒子

初始设定 N 个粒子,当 N 为偶数,即选取 $N/2$ 个粒子作为侵略性自由粒子;当 N 为奇数时,随机选取一个粒子,定义该粒子即是侵略性自由粒子,也是择优进化群中的粒子。此策略可保证侵略性自由粒子与择优进化群的规模保持一致。侵略性自由粒子具备高自由度和侵略性。

1.2.1 高自由度

本文设置为 D 维,粒子初始赋予位置信息。大部分随机优化算法(包括粒子群算法和遗传算法等)对于粒子的更新多数是 D 维,采用相同的参数更新。本文侵略性自由粒子,在粒子所具有 D 维的基础上,各维逐一与相对应的高自由度参数进行更新。以此更新策略为前提,为保证侵略性自由粒子的每次更新不超出预设边界,高自由度参数设置为 $[-1, 1]$ 之间的随机数,由此改变粒子位置,该策略增强了算法的多样性。更新过程如图 3 所示:(其中: X_i 为各维的位置信息, R_i 是随着 X_i 逐一生成的高自由度参数, $1 \leq i \leq D$ 。)

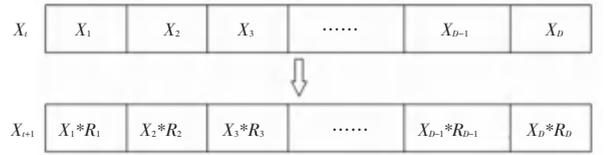


图 3 侵略性自由粒子更新示意图

Fig. 3 Update diagram of aggressive free particles

1.2.2 侵略性

自由粒子的侵略性体现在粒子群的优化更新过程中。当择优进化群的粒子在适应度值竞争中获胜作为竞争最优时,侵略性未被激活,择优进化群按原有轨迹进行更新;当侵略性自由粒子在适应度值竞争中获胜作为竞争最优时,激活侵略性。此时侵略性自由粒子以改进速度公式作为载体,参与择优进化群的更新过程,指导择优进化群向侵略性自由粒子的位置信息进行学习,使择优进化群摆脱原有更新轨迹。该策略增强了算法摆脱局部最优的能力。

1.3 改进速度公式

传统粒子群算法 (PSO) 的速度更新公式中,存在个体最优与全局最优两个概念。本文提出的算法凭借竞争最优 (Cbest) 的选择机制,摒弃个体最优与全局最优,将原有的两个外部存档,优化为仅对竞争最优进行外部存档,降低了算法的空间复杂度。算法在粒子速度更新时,仅向竞争最优进行学习,提高粒子的学习精度。对比传统速度公式,改进后的速度公式提高了算法的运算速度,将算法效率进一步提升。速度公式如式 (1) 所示。

$$v(t + 1) = \omega v(t) + cr(Cbest - x(t)). \quad (1)$$

2 实验结果分析

通过对 8 个典型标准函数的测试,对比 CAPSO 算法和其它改进算法的性能。其中 $f_1 \sim f_4$ 为单峰函数; $f_5 \sim f_8$ 为多峰函数,具体函数见表 1。

表 1 标准测试函数

Tab. 1 Standard test functions

测试函数	函数名	定义域范围	最优值
f_1	Sphere	-100, 100	0
f_2	Schwefel	-10, 10	0
f_3	Rosenbrock	-30, 30	0
f_4	Step	-100, 100	0
f_5	Ackley	-32, 32	0
f_6	Rastrigin	-5.12, 5.12	0
f_7	Griewank	-600, 600	0
f_8	Quartic	-1.28, 1.28	0

参数设置: 维数 $D = 30$ 、种群规模 $N = 50$ 、最大

迭代次数为 1 000, $c = 2$, r 为 $[0, 1]$ 之间的随机数, ω 最大值为 0.9、最小值为 0.4; 每个算法计算 10 次, 最优解取平均值。

通过 Matlab 仿真, 得到不同算法在 8 个标准测试函数的收敛曲线如图 4、图 5, 各算法测试结果见表 2。

表 2 不同算法在 8 个标准测试函数中的结果

Tab. 2 Results of different algorithms in 8 standard test functions

算法		CAPSO	PSO	RLPSO	TDDALFPSO	ScDCPS	IPSO
平均值	f_1	0.0000E+00	2.8584E+01	8.2500E-02	1.2000E-02	1.9076E+01	1.2916E+02
方差		0.0000E+00	1.2548E+01	2.4760E-01	1.1000E-03	2.3357E+01	5.9742E+01
平均值	f_2	0.0000E+00	7.1345E+00	2.0000E+00	2.2000E-02	5.3378E+00	8.6550E+00
方差		0.0000E+00	4.3074E+00	4.0000E+00	1.4400E-02	5.2050E+00	3.6594E+00
平均值	f_3	2.8729E+01	2.3179E+07	4.6436E+01	2.8997E+01	9.7146E+03	6.2332E+03
方差		8.0200E-02	5.0339E+03	5.3118E+01	1.4500E-02	2.6866E+04	4.6032E+03
平均值	f_4	2.3809E+00	2.0097E+01	6.8890E-01	5.2226E+00	3.6088E+01	1.2738E+02
方差		7.3420E-01	1.2321E+01	3.3430E-01	8.2370E-01	1.0178E+01	5.5742E+01
平均值	f_5	8.8818E-16	4.7990E+00	3.7560E-01	5.7000E-03	2.5003E+00	6.3597E+00
方差		0.0000E+00	8.4450E-01	7.1160E-01	4.2000E-03	1.7719E+00	9.9810E-01
平均值	f_6	0.0000E+00	7.0674E+01	3.4546E+01	2.5000E-03	4.9198E+01	7.4068E+01
方差		0.0000E+00	1.6676E+01	2.0548E+01	4.5000E-03	3.1900E+01	1.0529E+01
平均值	f_7	0.0000E+00	1.3042E+00	1.3400E-02	1.2000E-03	1.2187E+00	2.3108E+00
方差		0.0000E+00	1.1970E-01	4.0200E-02	1.1000E-03	4.0920E-01	8.3470E-01
平均值	f_8	2.1412E-05	8.1000E-02	1.8100E-02	1.4000E-03	6.2700E-02	9.7600E-02
方差		2.2010E-05	2.9100E-02	1.2100E-02	1.3000E-03	2.7600E-02	7.6700E-02

注: 表中添加阴影数据为对照算法中精度最高的解。

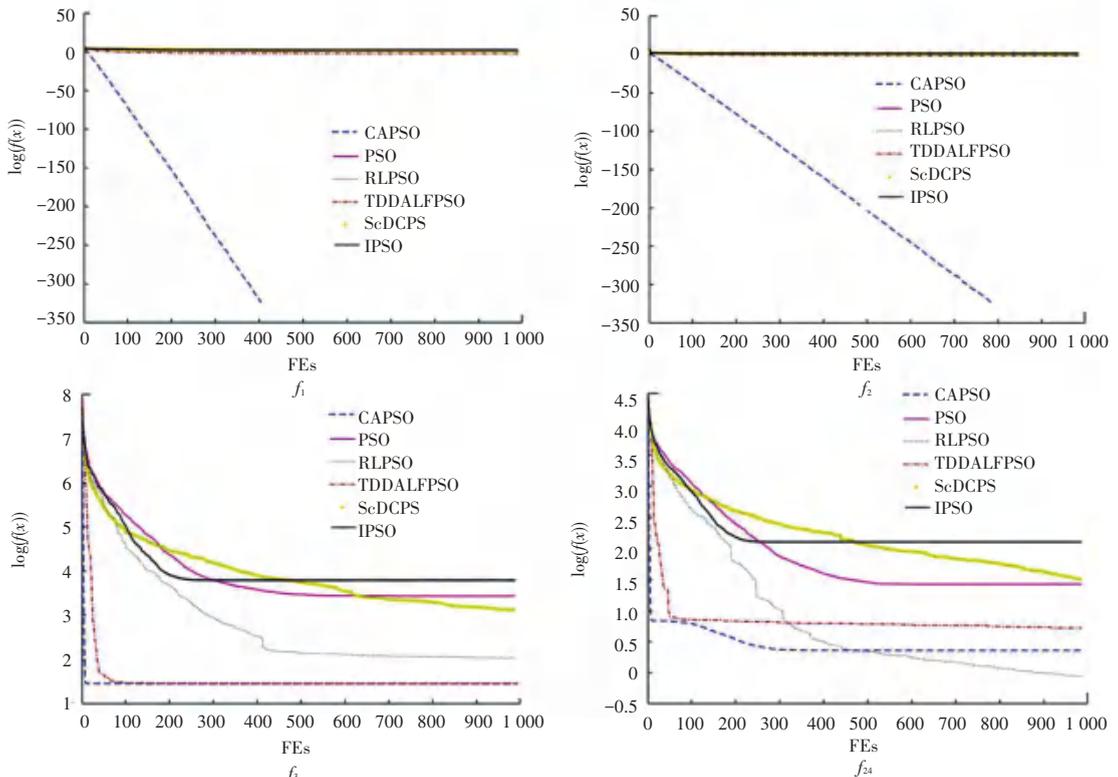


图 4 不同算法在 4 个单峰标准测试函数的收敛曲线

Fig. 4 Convergence curves of different algorithms on four unimodal standard test functions

函数 f_1, f_2 是简单的单峰测试函数, CAPSO 算法可以直接寻优。函数 f_1 进化次数为 420 次寻优, 函数 f_2 进化次数为 800 次寻优。在图像 f_1 与图像 f_2 中, CAPSO 对比其余五种算法, 在收敛速度与收敛精度方面, 性能大幅提升, 造成图像重叠; 而在表 2 中可以看出, RLPSO 算法与 TDDALFPSO 算法在精度上优于 PSO 与 ScDCPS 算法。结合 f_1 与 f_2 图像可知, 竞争池机制发挥了重要作用, 保证算法的每次更新都向最优的粒子学习, 算法寻优精度提升明显; 函数 f_3 和 f_4 稍复杂, CAPSO 算法未寻到最优值, 相对比其它算法收敛精度较高。在 f_3 图像中可以看出

CAPSO 算法与 TDDALFPSO 算法在收敛精度上几乎一致, 但 CAPSO 算法的收敛速度更快; 在 f_4 图像中, 收敛精度最高的是 RLPSO 算法, 但 CAPSO 算法的收敛速度更快, 弥补了收敛精度稍差的劣势。结合 f_3 图像和 f_4 图像, 通过 ScDCPS 算法与 PSO 算法对比, 可以得出 ScDCPS 算法牺牲收敛速度来提高收敛精度, 在此基础上收敛精度仍不如 RLPSO 算法。通过 4 个单峰测试函数可得到, CAPSO 算法对比其他改进算法收敛速度更快, 收敛精度较高。CAPSO 算法在单峰函数求解问题上性能较优。

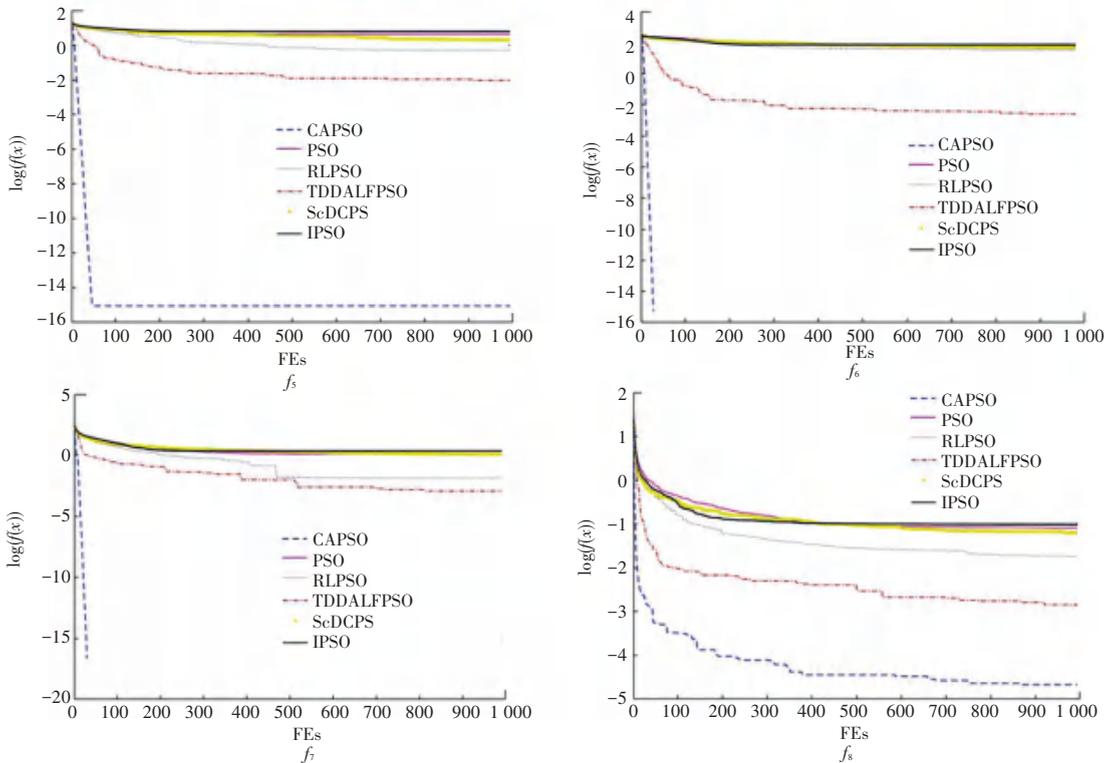


图 5 不同算法在 4 个多峰标准测试函数的收敛曲线

Fig. 5 Convergence curves of different algorithms on four multi-peak standard test functions

函数 $f_5 \sim$ 函数 f_8 为较复杂的多峰函数, 在 $f_5 \sim f_8$ 的 4 张图像中, 可直观地观察到 CAPSO 算法无论在收敛速度上, 还是收敛精度上都保持最优, 而在函数 f_6 和函数 f_7 , CAPSO 算法可以直接寻优。函数 f_6 进化次数为 32 次寻优, 函数 f_7 进化次数为 32 次寻优。在函数 f_5 的测试结果中, CAPSO 算法最优, PSO 算法、RLPSO 算法与 ScDCPS 算法在收敛速度和精度上相差较小, TDDALFPSO 算法略优, 但对比 CAPSO 算法存在较大差距。在函数 f_8 的测试结果中, CAPSO 算法同样优于其它算法, 虽然有短暂陷入局部最优的情况, 但侵略性自由粒子凭借自身高自由度的特性, 迅速令算法摆脱局部最优的情况, 并且在 1 000 次进化

后, 仍有下降的趋势。TDDALFPSO 算法优于 RLPSO 算法, ScDCPS 算法略优于 PSO 算法。CAPSO 算法在多峰函数求解问题上, 性能同样较优。

由上述对比可知, 经过改进的 CAPSO 算法, 无论在单峰函数问题还是多峰函数问题, 总体性能相比较 PSO 算法以及其他思想改进方式的 RLPSO 算法、TDDALFPSO 算法、ScDCPS 算法和 IPSO 算法, 有较大的提升。与 IPSO 算法的对比可以得出, 经本文优化后的竞争侵略策略对比 IPSO 算法的竞争策略性能更优, 种群间的粒子竞争机制发挥着重要作用。从表 2 的数据方差中也可看出, 在多数测试函数中 CAPSO 算法的方差值较小, 算法稳定性较高。

3 结束语

本文结合多种改进策略,提出基于竞争侵略策略的粒子群算法(CAPSO),该算法利用侵略性自由粒子的高自由度特性结合竞争池概念,大幅度的提升了算法的性能。CAPSO 算法在避免陷入局部最优和提高收敛速度的问题上表现出较强的能力,通过竞争最优(Cbest)的选择机制,摒弃个体最优与全局最优,达到简化速度公式的目的,令算法进化过程中更新速度更快,学习效率更高,在单峰函数和多峰函数中都具有优秀的收敛速度和收敛精度,且算法稳定,证明 CAPSO 算法的机制可以解决多种优化问题。

参考文献

- [1] CONTALDI C, VAFAEE F, NELSON P C. Bayesian network hybrid learning using an elite-guided genetic algorithm [J]. *Artificial Intelligence Review*, 2018(293):1-28.
- [2] LI P F, WANG H B, HE D Q. Ship weather routing based on improved ant colony optimization algorithm [C]// 2018 IEEE Industrial Cyber-Physical Systems (ICPS). IEEE, 2018.310-315.
- [3] KENNEDY J, EBERHART R. Particle swarm optimization [C]// Proceedings of 1995 IEEE International Conference on Neural Networks. Piscataway, NJ: IEEE Press, 2002:1942-1948.
- [4] SHI Y, EBERHART R C. Fuzzy adaptive particle swarm optimization [C]// Proceedings of the IEEE Congress on Evolutionary Computation. Seoul, Korea, 2001: 1011-106.
- [5] TAHERKHANI M, SAFABAKHSH R. A novel stability-based adaptive inertia weight for particle swarm optimization [J]. *Applied Soft Computing*, 2016, 38:281-295.

- [6] EL-SHERBINY M M. Particle swarm inspired optimization algorithm without velocity equation [J]. *Egyptian Informatics Journal*, 2011, 12(1):1-8.
- [7] CLERC M. The swarm and the queen: Toward a deterministic and adaptive particle swarm optimization [C]// Proceedings of the Congress on Evolutionary Computation, 1999: 1951-1957.
- [8] WANG F, ZHANG H, LI K, et al. A hybrid particle swarm optimization algorithm using adaptive learning strategy [J]. *Information Sciences*, 2018, 436: 162-177.
- [9] NIU B, HUANG H, TAN L. Symbiosis-based alternative learning multi-swarm particle swarm optimization [J]. *IEEE/ACM Transactions on Computational Biology & Bioinformatics*, 2017, 14(1):4-14.
- [10] 胡旺, 李志蜀. 一种更简化而高效的粒子群优化算法 [J]. *软件学报*, 2007(4):93-100.
- [11] 邓先礼, 魏波, 曾辉. 基于多种群的自适应迁移 PSO 算法 [J]. *电子学报*, 2018, 46(8):1858-1865.
- [12] 夏学文, 刘经南, 高柯夫. 具备反向学习和局部学习能力的粒子群算法 [J]. *计算机学报*, 2015(7):1397-1407.
- [13] 罗强, 季伟东, 徐浩天, 等. 一种最优粒子逐维变异的粒子群优化算法 [J]. *小型微型计算机系统*, 2020, 41(2):259-263.
- [14] 王磊, 王行甫, 苗付友. 一种带有二维扰动和自适应学习因子的粒子群算法 [J]. *小型微型计算机系统*, 2018, 39(11):3-7.
- [15] 王蓉芳, 焦李成, 刘芳. 自适应动态控制种群规模的自然计算方法 [J]. *软件学报*, 2012, 23(7):1760-1772.
- [16] 李旭渊, 许化龙. 一种基于竞争策略的粒子群优化算法 [J]. *计算机仿真*, 2008, 25(8):166-168.
- [17] CHENG R, JIN Y. A Competitive Swarm Optimizer for Large Scale Optimization [J]. *Cybernetics, IEEE Transactions on*, 2015, 45(2):191-204.
- [18] 刘明, 董明刚, 敬超. 基于定期竞争学习的多目标粒子群优化算法 [J]. *计算机应用*, 2019, 39(2):26-31.

(上接第 72 页)

3 结束语

本文提出了一种新的近似最近邻搜索算法,该算法可以在基于图的算法上缩短搜索时间并降低计算成本。该方法是基于 kNN 图构造的,为了从与查询点的角度引导查询路径,预先根据角度对所有点的邻居进行聚类,并且在搜索阶段仅比较一些靠近查询点的点的聚类。通过大量实验,该方法在 Audio 和 Trevi 等数据集上,与现存的先进算法 NSG 和 HNSW 对比可以达到 50% 和 40% 的成本缩减。

参考文献

- [1] FU C, XIANG C, WANG C, et al. Fast approximate nearest neighbor search with the navigating spreading-out graph [J]. *Very large data bases*, 2019, 12(5): 461-474.

- [2] MALKOV Y A, YASHUNIN D A. Efficient and robust approximate nearest neighbor search using Hierarchical Navigable Small World graphs [J]. *arXiv: Data Structures and Algorithms*, 2016.
- [3] FU C, CAI D. EFANNA: An Extremely Fast Approximate Nearest Neighbor Search Algorithm Based on kNN Graph. [J]. *arXiv: Computer Vision and Pattern Recognition*, 2016.
- [4] HARWOOD B, DRUMMOND T. FANNG: Fast Approximate Nearest Neighbour Graphs [C]// Computer vision and pattern recognition, 2016: 5713-5722.
- [5] TOUSSAINT G T. The relative neighbourhood graph of a finite planar set. *Pattern recognition*, 1980, 12(4):261-268.
- [6] MALKOV Y, PONOMARENKO A, LOGVINOV A, et al. Approximate nearest neighbor algorithm based on navigable small world graphs [J]. *Information Systems*, 2014, 45: 61-68.