

一种基于差分策略的群搜索优化算法

熊聪聪 郝璐萌 王 丹 邓雪晨

(天津科技大学计算机科学与信息工程学院 天津 300222)

摘 要 针对群搜索优化(Group Search Optimizer, GSO)算法易陷入局部最优、收敛速度较慢、收敛精度较低等问题,提出一种基于差分策略的群搜索优化(Differential Ranking-based Group Search Optimizer, DRGSO)算法。主要进行两方面改进:1)按照适应度值的大小对种群进行排序,适当增加发现者的数目,使种群能够获得更好的启发式信息,加快了算法的收敛速度,有效地避免了算法陷入局部最优;2)在发现者搜索过程中,引入4种不同的差分变异策略,提高了算法的收敛精度,增强了算法的群体多样性。11组国际标准测试函数上的实验测试结果显示,与GA, GSO, PSO算法相比,DRGSO算法具有较强的全局搜索能力以及局部资源勘探能力,算法整体收敛性能明显提高。

关键词 群搜索优化算法, 差分变异, 收敛速度, 收敛精度

中图分类号 TP301 文献标识码 A DOI 10.11896/j.issn.1002-137X.2017.02.041

Group Search Optimizer Based on Differential Strategies

XIONG Cong-cong HAO Lu-meng WANG Dan DENG Xue-chen

(College of Computer Science and Information Engineering, Tianjin University of Science and Technology, Tianjin 300222, China)

Abstract The conventional group search optimizer (GSO) is not free from some drawbacks such as easily falling into local optimum, a relative long computing time and lower convergence accuracy. In this study, we proposed a differential ranking-based group search optimizer (DRGSO) algorithm to alleviate these limitations. There are mainly two improvements in the design of DRGSO. First, the population is initialized according to the ranking of fitness values. With this regard, the population obtains heuristic information and alleviates premature convergence to some extent. Second, four evolutionary operators based on differential strategies are constructed to improve the convergence of the algorithm and enhance the population diversity. To demonstrate the performance, eleven benchmark functions were included to evaluate the performance of DRGSO. Experimental results indicate that the proposed DRGSO exhibits better performance in comparison with the GA, PSO and GSO in terms of accuracy and speed of convergence.

Keywords Group search optimizer (GSO), Differential mutation, Convergence speed, Convergence accuracy

目前,模拟生物系统群体生活习性的群体智能优化算法(Swarm Intelligence, SI)等优化方法已经广泛应用于多个研究领域^[1-5]。其中,比较经典的SI算法有:Dorigo^[2]提出的模拟蚁群觅食行为的蚁群优化算法(Ant Colony Algorithm, ACO),Kenndy与Eberhart^[6]提出的基于鸟群以及鱼群觅食行为的粒子群优化算法(Particle Swarm Optimization, PSO),李晓磊等^[7]提出的人工鱼群算法以及S. He等^[8-9]提出的群搜索优化(GSO)算法。

GSO算法是基于动物“信息共享”和“相互合作”的社会特性而建立的一种优化算法^[10]。此算法首次提出了动物的视觉搜索机制,将种群分为3个角色进行资源启发式搜索,具有参数鲁棒性强、易于编程、收敛速度较快等优点;并且,作为

一种新兴的群体智能优化算法,群搜索优化算法受到国内外众多学者的关注。此外,GSO算法也面临着收敛后期搜索缓慢、不容易跳出局部最优等问题。

针对这些问题,研究者经过大量的研究,提出了很多改进的方法,其中包括快速群搜索改进^[11]、基于Metropolis准则的改进^[12]、对加入者和游荡者引入步长搜索进行改进^[13]等。因此,设计新的个体搜索方式来提高算法的性能非常必要。唯一的发现者在共享核心资源时,一定程度上限制了群体多样性,忽略其他较优的资源,搜索比较盲目,算法性能较单一,降低了算法的收敛精度^[14]。而差分变异策略作为差分进化算法的核心部分,通过进行向量差运算,与其他个体求和得到新的个体,以保持父代、子代之间的繁衍。该策略在不断迭

到稿日期:2015-12-22 返修日期:2016-04-12 本文受天津市高等学校科技发展基金计划项目(20140803),天津科技大学青年教师创新基金(2014CXIG30),国家自然科学基金面上项目(61272509),国家自然科学基金青年基金(61402332)资助。

熊聪聪(1961—),女,教授,主要研究方向为计算机网络与嵌入式系统、并行计算、智能信息处理等,E-mail: xiongcc@tust.edu.cn;郝璐萌(1990—),女,硕士,主要研究方向为智能计算;王 丹(1983—),女,博士,讲师,主要研究方向为智能计算;邓雪晨(1991—),男,硕士,主要研究方向为计算机网络与信息安全。

代过程中考虑到了全局随机点的较优可能性,从而能提高算法的全局搜索能力^[15]。为此,本文提出一种基于差分变异策略的群搜索优化算法(DRGSO)。在该算法的寻优过程中,对种群按照其适应值的优劣排序,增加发现者数目,使较优个体有更大机会参与发现者的核心搜索,从而加速算法收敛,以达到最优解。在进行发现者操作时,引入差分变异操作,使个体之间充分交流,多样化搜索路径,加强群体的多样性。文中选用11组标准测试函数将所提算法与GA, PSO, GSO算法进行了性能测试比较。仿真实验表明,DRGSO算法表现出了更高的收敛精度和较快的收敛速度,具有整体上较强的全局寻优能力和局部搜索能力。

1 群搜索优化算法

GSO优化算法是模拟动物群居行为的一种随机优化算法。该算法以发现-加入模型(Producer-Scrounger, PS)为原型^[8],在PS模型的基础上将种群划分为3种角色:1)搜索资源并且共享资源信息的发现者(Producer);2)从发现者获取信息并掠夺资源的加入者(Scrounger);3)兼有发现者和加入者角色并进行分散随机行为的游荡者(Ranger)^[10]。群搜索优化算法首次采用了生物学的视觉搜索机制,在每次迭代中,选择当前具有最优适应值的个体作为唯一的发现者,发现者按照设定的视觉搜索角度来寻找更好的资源信息,把握搜索的整体方向。其次,80%的个体被随机选择作为加入者,并以一定的步长跟随发现者的方向。最后,剩余的个体作为游荡者参与搜索。以下详细描述了GSO算法实现过程。

设搜索空间为 n 维空间,在第 k 次迭代中,第 i 个个体的位置用向量表示为 $X_i^k \in R^n$,其搜索角度为 $\varphi_i^k = (\varphi_{i1}^k, \dots, \varphi_{in}^k) \in R^{n-1}$,它的搜索方向是 $D_i^k(\varphi_i^k) = (d_{i1}^k, \dots, d_{in}^k) \in R^n$,搜索方向 $D_i^k(\varphi_i^k)$ 可以通过式(1)~式(3)按照搜索角度 φ_i^k 转换为笛卡尔坐标而得到。

$$d_{i1}^k = \prod_{p=1}^{n-1} \cos(\varphi_{ip}^k) \quad (1)$$

$$d_{ij}^k = \sin(\varphi_{ij-1}^k) \cdot \prod_{p=i}^{n-1} \cos(\varphi_{ip}^k) \quad (2)$$

$$d_{in}^k = \sin(\varphi_{in-1}^k) \quad (3)$$

首先,在第 k 次迭代时,GSO算法选择当前适应值最优的个体作为唯一的发现者,发现者 X_p^k 按以下3个步骤进行资源信息搜索。

步骤1 发现者在 0° 进行视觉扫描搜索,通过式(4)~式(6)分别在其前方、左方以及右方随机扫描3个位置点。

$$X_z = X_p^k + r_1 l_{\max} D_p^k(\varphi^k) \quad (4)$$

$$X_l = X_p^k + r_1 l_{\max} D_p^k(\varphi^k - r_2 \theta_{\max}/2) \quad (5)$$

$$X_r = X_p^k + r_1 l_{\max} D_p^k(\varphi^k + r_2 \theta_{\max}/2) \quad (6)$$

其中, $r_1 \in R^1$ 是一个均值为0、方差为1的正态分布随机数; $r_2 \in R^{n-1}$ 为 $[0,1]$ 间均匀分布的随机数; $l_{\max}$ 是最大搜索距离,并能够通过式(7)计算而得。

$$l_{\max} = \|U - L\| = \sqrt{\sum_{i=1}^n (U_i - L_i)^2} \quad (7)$$

其中, U_i 和 L_i 分别是设定变量取值范围的上界和下界。

步骤2 计算搜索的各点的适应值,发现者找出当前种群的最优适应值以及最优位置。如果新搜索的位置适应值优

于当前位置,发现者会跳到新搜索位置;否则会停留在当前的位置,并且通过式(8)随机转换一个新的搜索角度,准备下一次迭代。

$$\varphi^{k+1} = \varphi^k + r_2 \alpha_{\max} \quad (8)$$

其中, $\alpha_{\max} \in R^1$ 是最大的搜索转换角度。

步骤3 发现者如果在进行了 α 次迭代后并不能找到更好的资源信息,就会通过式(9)回到 0° 位置。

$$\varphi^{k+\alpha} = \varphi^k \quad (9)$$

其中, $\alpha \in R^1$ 是一个使用者自行确定的常量。

其次,在每一次种群个体的循环里,被选为加入者的个体会一直寻找机会去搜索、掠夺由发现者搜索到的资源信息。加入者会按照式(10)确定的随机步长跟随发现者进行搜索。

$$X_i^{k+1} = X_i^k + r_3 (X_p^k - X_i^k) \quad (10)$$

其中, $r_3 \in R^n$ 为 $[0,1]$ 间均匀分布的随机数。

最后,剩下的个体作为游荡者随机分布在群体搜索空间中,并以随机步长、有规律的搜索策略来有效定位资源信息。在第 k 次迭代中,游荡者按照式(11)产生一个搜索角度 φ_i ,按照式(12)得出的随机搜索距离 l_i 来确定新的搜索位置,并通过式(13)来确定移动的位置。

$$\varphi_i^{k+1} = \varphi_i^k + r_2 \alpha_{\max} \quad (11)$$

$$l_i = \alpha \cdot r_1 l_{\max} \quad (12)$$

$$X_i^{k+1} = X_i^k + l_i D_i^k(\varphi_i^{k+1}) \quad (13)$$

通过3种角色个体“共同合作”,反复迭代,更新种群个体的新解,收敛得到最优个体。值得注意的是,为了能够更容易地搜索到资源信息,会执行防止搜索跃界的策略:如果某个搜索个体跳出了搜索空间,会判断并返回到当前搜索空间界限中的位置。其中,算法1描述了GSO算法的伪代码。

算法1 GSO算法伪代码

1. 随机初始种群位置以及搜索角度;
2. 计算种群个体的适应度值 fvalue;
3. WHILE (迭代是否满足)
4. FOR $i=1$ to N_p
5. 根据式(4)~式(6)及式(8)、式(9)执行发现者基本操作;
6. 根据式(10)执行加入者操作;
7. 根据式(13)执行游荡者操作;
8. END FOR
9. 更新种群的位置,计算更新后种群的适应度值;
10. END WHILE

2 基于差分策略的群搜索优化算法

种群寻优能力的强弱取决于算法性能的好坏。为了提高个体的寻优能力,提出一种基于差分策略的群搜索优化算法(DRGSO)。该算法虽然保留了GSO算法的“发现者-加入者-游荡者”搜索框架,但通过对个体进行优劣排序,适当增加了发现者的数目,同时加入发现者的差分变异策略,提高了算法的搜索性能和收敛精度。算法2描述了DRGSO算法其中一种差分策略(DE/rand/1)的伪代码。从算法流程来看,主要在步骤3、步骤5、步骤7以及步骤14上不同于基本GSO算法(算法1)。

算法2 DRGSO算法伪代码(例:DE/rand/1)

1. 随机初始种群位置以及搜索角度;

2. 计算种群个体的适应度值 fvalue;
3. 根据算法 3 进行排序操作;
4. WHILE (迭代是否满足)
5. FOR i=1 to $N_{producer}$
6. 根据式(4)一式(6)及式(8)、式(9)执行发现者基本操作;
7. 根据算法 4 执行差分策略;
8. END FOR
9. FOR i = $N_{producer} + 1$ to N_p
10. 根据式(10)执行加入者操作;
11. 根据式(13)执行游荡者操作;
12. END FOR
13. 更新种群的位置, 计算更新后种群的适应度值;
14. 根据算法 3 进行排序操作;
15. END WHILE

2.1 排序策略

尽管 GSO 算法在解决众多问题上都表现优越, 但其仍面临着个体搜索能力较差、算法性能单一的问题。受自然界规律的启发: 携带较优基因的物种总会因“物竞天择, 适者生存”的法则更容易生存下来或被选择进行后代繁衍。因此, 本文将排序操作应用到基本 GSO 算法的个体随机选择中。算法 3 详细描述了该操作: 核心步骤 3 根据适应值的大小将种群进行排序(根据适应度值越小而种群越优的原则, 改进的算法采用升序排序), 使得较优个体更容易参与随机搜索, 平衡了算法对资源信息的勘探和利用能力, 加快了算法的收敛速度。

GSO 算法中, 发现者作为算法的控制中心, 能够决定算法的搜索方向, 引导搜索的过程, 影响算法的收敛性能, 但是发现者的唯一性在一定程度上限制了群体的多样性, 降低了算法的收敛性能。本文基于排序后的种群, 适当增加了发现者的数目, 使得更多的较优个体作为发现者角色参与到启发式共享信息搜索中。实验证明, 该改进在整体上加快了收敛速度, 避免了陷入局部最优。另外, 在仿真实验部分还讨论了发现者数目对算法性能的影响。

算法 3 基于适应值(fvalue)对种群进行升序排序

Begin

1. 输入: 当前的种群的位置
2. 输出: 排序后的种群位置
3. Rank(fvalue) $\rightarrow$ index;
4. 根据 index 更新种群个体的顺序;

End

2.2 差分变异策略

考虑到发现者种群在迭代过程中仅是在发现者的零度、左侧和右侧 3 个方向进行随机搜索, 寻优方向单一, 搜索过于盲目, 为了多元地利用启发式资源信息, 提高种群的资源勘探能力, 本文将 4 组不同的差分变异算子引入到发现者搜索过程中, 其中 4 个差分算子均单独在 DRGSO 算法中运行。仿真实验部分比较分析了 4 种差分算子^[16-17]对算法性能的影响。算法 4 详细描述了其中一个差分策略的伪代码。

算法 4 基于排序的差分策略(例: DE/rand/1)

Begin

1. 输入: 当前发现者个体 i
2. 输出: 差分变异后的新个体

3. 随机选择 $s_1 \in \{1, N_{producer}\}$;
4. WHILE rndreal[0,1) $> p_{s_1}$ or $s_1 == i$
5. 随机选择 $s_1 \in \{1, N_{producer}\}$;
6. END WHILE
7. 随机选择 $s_2 \in \{1, N_{producer}\}$;
8. WHILE rndreal[0,1) $> p_{s_2}$ or $s_2 == s_1$ or $s_2 == i$
9. 随机选择 $s_2 \in \{1, N_{producer}\}$;
10. END WHILE
11. 根据式(14)执行差分变异操作;

End

DRGSO 算法中 4 种差分变异策略的公式描述如下。

DE/rand/1:

$$X_i^{k+1} = X_i^k + r_4 \cdot |X_{s_1}^k - X_{s_2}^k| \quad (14)$$

DE/rand/2:

$$X_i^{k+1} = X_i^k + r_4 \cdot |X_{s_1}^k - X_{s_2}^k| + r_5 \cdot |X_{s_3}^k - X_{s_4}^k| \quad (15)$$

DE/current-to-best/1:

$$X_i^{k+1} = X_i^k + r_4 \cdot |X_{best}^k - X_i^k| + r_5 \cdot |X_{s_1}^k - X_{s_2}^k| \quad (16)$$

DE/current-to-best/2:

$$X_i^{k+1} = X_i^k + r_3 \cdot |X_{best}^k - X_i^k| + r_4 \cdot |X_{s_1}^k - X_{s_2}^k| + r_5 \cdot |X_{s_3}^k - X_{s_4}^k| \quad (17)$$

其中, DE/x/y 的含义是: x 表示差分变异个体, 它既可以是种群中的随机(rand)个体, 也可以是当前种群中的最优(current-to-best)个体; y 表示差分个体的对数。 $r_4 \in R^n$ 和 $r_5 \in R^n$ 均为 [0,1) 间均匀分布的随机数; X_{best}^k 表示在第 k 次迭代中最优的种群个体; $s_1, s_2, s_3, s_4 \in \{1, \dots, N_{producer}\}$ 且 $s_1 \neq s_2 \neq s_3 \neq s_4 \neq i$; $N_{producer}$ 代表发现者的数目。文中把 $X_{s_1}^k, X_{s_2}^k, X_{s_3}^k, X_{s_4}^k$ 称作差分个体, 差分个体是在第 k 次迭代过程中从发现者种群中随机选择的个体, 用作差分操作。同时, 在随机选择差分个体的过程中, 利用选择概率(p_i)^[15]筛选差分个体。本文采用的选择概率计算公式描述如下:

$$p_i = \frac{R_i}{N_p}, i=1, 2, \dots, N_{producer} \quad (18)$$

$$R_i = N_p - i, i=1, 2, \dots, N_{producer} \quad (19)$$

其中, N_p 代表种群的大小, R_i 是排序的种群个体, p_i 是种群的选择概率。

尽管 4 种差分策略的优势各有不同, 前两组(见式(14)、式(15))策略有利于保持群体的多样性, 后两组(见式(16)、式(17))则更强调算法的收敛速度, 但是这些差分策略都能在整体上增强发现者个体勘探资源的能力, 提高算法的收敛精度; 并且, 在迭代后期算法趋向于收敛, 种群间的个体差异性变小, 差分策略在一定程度上增强了算法的局部搜索能力, 使得算法具有较强的稳定性。

3 实验仿真与分析

3.1 国际标准测试函数

基于 4 种不同的差分策略, 本文得到 4 种不同的改进算法: DRGSO (DE/rand/1), DRGSO (DE/rand/2), DRGSO (DE/current-to-best/1), DRGSO (DE/current-to-best/2)。为了验证这些改进算法的性能, 选取 11 个国际标准函数对 DRGSO 算法进行测试, 如表 1 所列, 其中 f1-f6 为单峰函数, f7-f11 为多峰函数^[18]。

表 1 11 个测试函数

Test function	<i>n</i>	<i>S</i>	<i>f</i> _{min}
$f1(x)=\sum_{i=1}^n x_i^2$	30	$[-100,100]^n$	0
$f2(x)=\sum_{i=1}^n x_i +\prod_{i=1}^n x_i $	30	$[-10,10]^n$	0
$f3(x)=\sum_{i=1}^n (\sum_{j=1}^i x_j)^2$	30	$[-100,100]^n$	0
$f4(x)=\max_i\{ x_i ,1\leq i\leq n\}$	30	$[-100,100]^n$	0
$f5(x)=\sum_{i=1}^{n-1}(100(x_{i+1}-x_i^2)^2+(x_i-1))^2$	30	$[-30,30]^n$	0
$f6(x)=\sum_{i=1}^n (\lfloor x_i+0.5 \rfloor)^2$	30	$[-100,100]^n$	0
$f7(x)=\sum_{i=1}^n (x_i^2-10\cos(2\pi x_i)+10)^2$	30	$[-5.12,5.12]^n$	0
$f8(x)=-20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^n x_i^2})-\exp(\frac{1}{n}\sum_{i=1}^n \cos 2\pi x_i)+20+e$	30	$[-32,32]^n$	0
$f9(x)=\frac{1}{4000}\sum_{i=1}^{30}(x_i-100)^2-\prod_{i=1}^n \cos(\frac{x_i-100}{\sqrt{i}})+1$	30	$[-600,600]^n$	0
$f10(x)=\frac{\pi}{n}\{10\sin^2(\pi y_1)+\sum_{i=1}^{29}(y_i-1)^2[1+10\sin^2(\pi y_{i+1})]+(y_n-1)^2\}+\sum_{i=1}^{30}u(x_i,10,100,4)$ $y_i=1+\frac{1}{4}(x_i+1)$ $u(x_i,a,k,m)=\begin{cases} k(x_i-a)^m, & x_i>a \\ 0, & -a\leq x_i\leq a \\ k(-x_i-a)^m, & x_i<-a \end{cases}$	30	$[-50,50]^n$	0
$f11(x)=0.1\{\sin^2(\pi x_1)+\sum_{i=1}^{29}(x_i-1)^2[1+\sin^2(\pi x_{i+1})]+(x_n-1)^2[1+\sin^2(2\pi x_{30})]\}+\sum_{i=1}^{30}u(x_i,5,100,4)$	30	$[-50,50]^n$	0

3.2 参数设置

为了保证所有仿真对比实验的准确性,所有的参数设置与标准算法的均相同(特殊说明除外)。实验中基本参数设置如下:群体大小 *PopSize*=48,维数 *NDim*=30,最大迭代次数 *MaxIter*=5000,各算法运行次数 *runtimes*=50。

3.3 函数优化实验结果及算法比较分析

DRGSO 算法与 GA,PSO,GSO 算法的比较如表 2 所列。

表 2 函数 *f1*—*f11* 对各算法所得最小值及标准差的比较

函数	GA	PSO	GSO	DRGSO (DE/current-to-best/2)
<i>f1</i>	3.1711 (1.6621)	3.69E-37 (3.69E-37)	1.95E-08 (1.1629E-08)	1.02E-39 (3.82233E-39)
<i>f2</i>	0.5771 (0.1306)	2.92E-24 (1.14E-23)	3.70E-05 (8.62E-05)	4.11E-26 (2.15E-25)
<i>f3</i>	9749.9145 (2594.96)	1.20E-03 (2.11E-03)	5.78E+00 (3.68E+00)	0 0
<i>f4</i>	7.961 (1.5063)	0.4123 (0.25)	1.08E-01 (4.00E-02)	0 0
<i>f5</i>	338.5616 (361.497)	37.3582 (32.1436)	4.98E+01 (3.02E+01)	4.01E-33 (2.79E-32)
<i>f6</i>	3.70E+00 (1.95E+00)	0.146 (0.4182)	1.60E-02 (0.1333)	0 0
<i>f7</i>	0.6509 (0.3594)	20.7863 (5.94)	1.02E+00 (9.51E-01)	0 0
<i>f8</i>	0.8678 (0.2805)	1.34E-03 (4.24E-02)	2.65E-05 (3.08E-05)	8.20E-14 (1.76E-13)
<i>f9</i>	1.0038 (6.75E-02)	0.2323 (0.4434)	3.08E-02 (3.09E-02)	5.37E-03 (2.91E-02)
<i>f10</i>	4.36E-02 (5.06E-02)	3.95E-02 (9.14E-02)	2.76E-11 (9.17E-11)	3.02E-32 (4.56E-32)
<i>f11</i>	0.1681 (7.0681E-02)	5.05E-02 (0.5691)	4.69E-05 (7.00E-04)	6.53E-31 (2.92E-30)

表 2 列出了各算法分别对函数 *f1*—*f11* 求最小值及标准差的结果。其中 DRGSO 算法选取了效果明显的 DRGSO

(DE/current-to-best/2)算法,该算法数据为实验测试所得,GA,PSO,GSO 算法的结果均来自于文献[9]。为了避免算法随机性的影响,算法均独立运行 50 次,以平均值作为最终寻优结果。发现者数目设定为 10。

从表 2 的对比结果可知,无论是单峰测试函数 *f1*—*f6*,还是多峰测试函数 *f7*—*f11*,DRGSO(DE/current-to-best/2)算法在收敛精度上均明显高于 GA,PSO,GSO 算法。在单峰函数 *f1* 和 *f2* 上,DRGSO(DE/current-to-best/2)的表现最优,收敛精度远高于 GSO 和 GA 算法的,特别是高于 PSO 的精度。而 *f3*,*f4*,*f6*,*f7* 已经达到收敛的最优值 0。

图 1 给出了 GSO,DRGSO(DE/rand/1),DRGSO(DE/rand/2),DRGSO(DE/current-to-best/1)和 DRGSO(DE/current-to-best/2)5 种算法在 *f1*,*f2*,*f5*,*f8*,*f9*,*f10*,*f11* 共 7 个函数上关于迭代次数与平均适应值的关系对比。其中,所有算法的发现者数目均设定为 10。

由图 1 可知,本文提出的 DRGSO 的 4 种改进算法通过引入基于排序的不同差分策略以及扩大发现者数目,对加速算法收敛速度、增强算法跳出局部极小值的能力及提高收敛精度有着很大的帮助。由 *f1*,*f2*,*f8*,*f10*,*f11* 的曲线图可知,DRGSO(DE/rand/1)和 DRGSO(DE/rand/2)具有相当的表现。由 *f5* 的曲线图看到:对于易陷入局部最优的 *f5* 函数,DRGSO(DE/rand/2)算法表现出较强的资源勘探能力,容易跳出局部最优值,在收敛后期收敛速度加快;DRGSO(DE/current-to-best/2)从迭代初期就具有较强的收敛性能,收敛速度具有一定的优势。在 *f9* 中,4 种改进算法的收敛速度都快于 GSO 算法,特别是 DRGSO(DE/current-to-best/1)和 DRGSO(DE/current-to-best/2)算法的搜索能力较强,有效防止了算法的早熟现象。

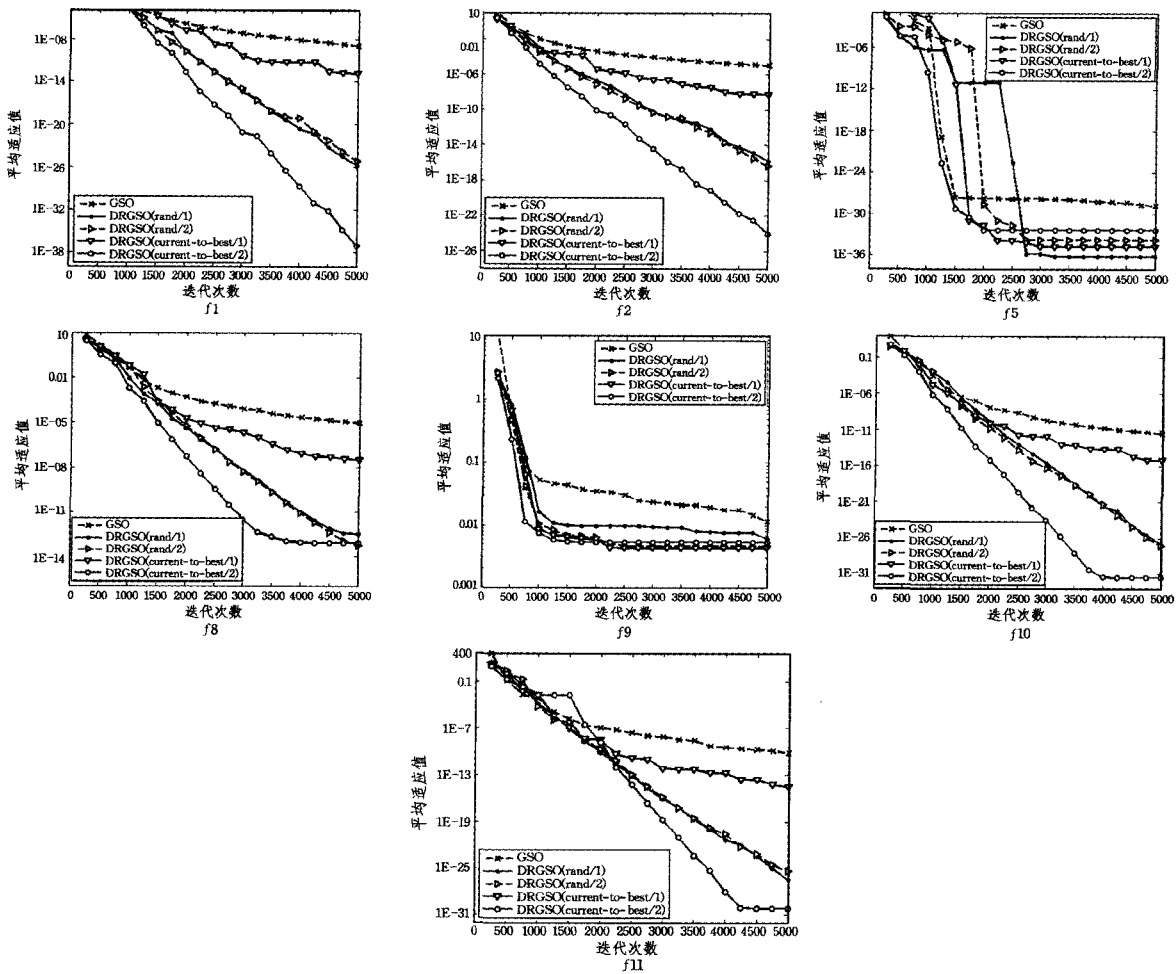


图 1 GSO 和 DRGSO 在函数上优化过程中的曲线对比

3.4 差分策略对算法性能的影响

表 3 给出了基于 4 种差分策略的 DRGSO 算法的最小值以及标准差的结果。

表 3 函数 $f1-f11$ 对 4 种 DRGSO 算法的最小值及标准差的对比

函数	DRGSO (DE/rand/1)	DRGSO (DE/rand/2)	DRGSO (DE/current- to-best/1)	DRGSO (DE/current- to-best/2)
$f1$	8.91E-28 (2.3654E-27)	2.81E-27 (1.62174E-26)	1.27E-13 (7.34444E-13)	1.02E-39 (3.82233E-39)
$f2$	3.15E-17 (1.14E-16)	1.05E-17 (2.40E-17)	5.33E-09 (1.73E-08)	4.11E-26 (2.15E-25)
$f3$	0 0	0 0	0 0	0 0
$f4$	0 0	0 0	0 0	0 0
$f5$	5.51E-37 (2.39E-36)	1.42E-34 (6.16E-34)	1.55E-35 (1.09E-34)	4.01E-33 (2.79E-32)
$f6$	0 0	0 0	0 0	0 0
$f7$	0 0	0 0	0 0	0 0
$f8$	3.08E-13 (1.10E-12)	3.81E-14 (2.29E-14)	2.96E-08 (8.93E-08)	8.20E-14 (1.76E-13)
$f9$	6.03E-03 (4.26E-02)	4.59E-03 (2.39E-02)	4.30E-03 (2.14E-02)	5.37E-03 (2.91E-02)
$f10$	1.08E-28 (6.74E-28)	3.79E-29 (1.87E-28)	6.07E-16 (4.13E-15)	3.02E-32 (4.56E-32)
$f11$	2.68E-28 (9.18E-28)	3.64E-27 (2.55E-26)	2.87E-15 (1.58E-14)	6.53E-31 (2.92E-30)

从表 3 的对比结果可知,DRGSO 的 4 种改进算法 DRGSO(DE/rand/1),DRGSO(DE/rand/2),DRGSO(DE/current-to-best/1),DRGSO(DE/current-to-best/2)在收敛精度以及算法的稳定性上均具有明显优势。DRGSO(DE/rand/1)在函数 $f5$ 上表现出非常好的收敛性能。同时,对于多峰函数 $f8$,DRGSO(DE/current-to-best/1)具有一定的优势,但是在 DRGSO(DE/rand/1),DRGSO(DE/rand/2),DRGSO(DE/current-to-best/2)上的表现具有更好的优势。对于 $f9$,即比较复杂的多峰函数,DRGSO 具有一定的优势,且 4 种改进算法表现相当。在 $f1,f2,f10,f11$ 上,DRGSO(DE/current-to-best/2)都表现出明显优势。在函数 $f3,f4,f6,f7$ 上计算均达到了最小值 0。

3.5 排序策略对算法性能的影响

为了分析排序策略对改进的 DRGSO 的影响,比较了 DRGSO 与 DEGSO 的最优适应值及标准差。如表 4 所列,DEGSO 算法在算法设计上没有加入排序操作。

由表 4 可知,基于 4 种差分排序策略的 DRGSO 算法整体上均好于 DEGSO 算法,只有在 $f5$ 上 DEGSO(DE/current-to-best/1)以及 DEGSO(DE/current-to-best/2)的收敛效果较好。在 $f3,f4,f6,f7$ 上,DRGSO 与 DEGSO 均达到了函数最优值。由此可知,排序策略对于加强算法的寻优能力以及提高算法的收敛精度有着重要的作用。

表 4 DRGSO 与 DEGSO 算法对于函数 $f1-f11$ 的对比

函数	DE/rand/1		DE/rand/2		DE/current-to-best/1		DE/current-to-best/2	
	DEGSO	DRGSO	DEGSO	DRGSO	DEGSO	DRGSO	DEGSO	DRGSO
$f1$	8.68E-10 (1.34E-09)	8.91E-28 (2.3654E-27)	1.17E-09 (9.66E-10)	2.81E-27 (1.62174E-26)	5.58E-10 (5.61E-10)	1.27E-13 (7.34444E-13)	6.93E-10 (6.59E-10)	1.02E-39 (3.82233E-39)
$f2$	7.84E-06 (4.07E-06)	3.15E-17 (1.14E-16)	8.29E-06 (4.50E-06)	1.05E-17 (2.40E-17)	9.01E-06 (6.00E-06)	5.33E-09 (1.73E-08)	9.76E-06 (6.53E-06)	4.11E-26 (2.15E-25)
$f3$	0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0
$f4$	0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0
$f5$	7.89E-32 (2.49E-31)	5.51E-37 (2.39E-36)	7.89E-32 (2.49E-31)	1.42E-34 (6.16E-34)	5.70E-38 (1.21E-37)	1.55E-35 (1.09E-34)	3.01E-37 (9.52E-37)	4.01E-33 (2.79E-32)
$f6$	0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0
$f7$	0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0
$f8$	4.82E-06 (2.27E-06)	3.08E-13 (1.10E-12)	5.87E-06 (1.78E-06)	3.81E-14 (2.29E-14)	1.16E-05 (1.45E-05)	2.96E-08 (8.93E-08)	6.73E-06 (2.98E-06)	8.20E-14 (1.76E-13)
$f9$	2.34E-02 (4.39E-02)	6.03E-03 (4.26E-02)	2.30E-02 (4.63E-02)	4.59E-03 (2.39E-02)	3.84E-02 (6.39E-02)	4.30E-03 (2.14E-02)	1.93E-02 (2.79E-02)	5.37E-03 (2.91E-02)
$f10$	1.03E-11 (2.70E-11)	1.08E-28 (6.74E-28)	2.43E-12 (3.13E-12)	3.79E-29 (1.87E-28)	1.94E-12 (1.59E-12)	6.07E-16 (4.13E-15)	1.24E-12 (1.16E-12)	3.02E-32 (4.56E-32)
$f11$	9.51E-12 (6.92E-12)	2.68E-28 (9.18E-28)	2.09E-11 (1.75E-11)	3.64E-27 (2.55E-26)	1.59E-11 (1.10E-11)	2.87E-15 (1.58E-14)	2.28E-11 (1.48E-11)	6.53E-31 (2.92E-30)

3.6 发现者数目对性能的影响

标准群搜索算法(GSO)将是唯一的发现者作为算法的控制角色,在一定程度上使算法搜索策略过于单一。本文扩大了发现者的数目,让更多携带较优资源信息的个体参与到寻优搜索中,在增强算法多样性的同时,也加快了算法收敛到最优值的速度。为了探讨发现者数目的适当性,列出了针对不同发现者数目 DRGSO(DE/current-to-best/2)算法在多峰函数 $f7-f11$ 上的对比结果,如表 5 所列。

表 5 基于不同发现者数目的 DRGSO 算法对比

函数	DRGSO(DE/current-to-best/2)			
	Producer=5	Producer=10	Producer=15	Producer=20
$f7$	0 0	0 0	0 0	0 0
$f8$	4.48E-09 (7.86E-09)	8.20E-14 (1.76E-13)	7.99E-15 0	8.70E-15 (1.50E-15)
$f9$	1.42E-02 (4.48E-02)	5.37E-03 (2.91E-02)	9.66E-03 (3.00E-02)	8.34E-14 (2.64E-13)
$f10$	2.05E-20 (2.68E-20)	3.02E-32 (4.56E-32)	1.57E-32 (2.88E-48)	4.23E-32 (3.61E-32)
$f11$	9.11E-17 (1.94E-16)	6.53E-31 (2.92E-30)	1.36E-32 (3.90E-34)	2.03E-31 (1.90E-31)

表 5 列出了发现者数目分别设置为 5,10,15,20 时算法的平均适应值及标准差。由此看出,当发现者数目从 5 逐渐增大到 15 时: $f8,f10$ 及 $f11$ 上的最优适应值在逐渐变小,收敛性能更强;对于 $f9$ 函数, $N_{producer}=15$ 比 $N_{producer}=10$ 的收敛精度稍差一些,但整体上具有越来越好的收敛趋势。当发现者数目再增大到 20 个时: $f8,f10$ 及 $f11$ 上的收敛精度均呈下降趋势, $f9$ 呈上升趋势。由此可知,对于 DRGSO 算法的改进,发现者数目在 10 和 15 时算法的收敛精度较好。同时,随着发现者数目的不断加大,算法的搜索效率以及收敛速度也会一定程度上降低,从而增加了算法寻优的复杂度。综合考虑以上因素,本文设定 DRGSO 算法的发现者参数 $N_{producer}=10$,以寻求较好的算法收敛性和稳定性。

4 应用实例

本实例采用 Fisher 的 Iris 植物样本数据。该数据集由 3 种类型的 Iris 植物的 150 个样本组成,每个样本均为 4 维向量,代表其 4 种不同的特征属性。表 6 列出了 4 种具体的属性特征。算法均进行了 50 次验证。

表 6 数据集 Iris 的 4 种特征属性(单位:cm)

特征名	含义
Sepal.Length	花萼长度
Sepal.Width	花萼宽度
Petal.Length	花瓣长度
Petal.Width	花瓣宽度

下面对 Iris 数据样本采用基于聚类中心的浮点数编码方式^[19],每个个体由 K 个聚类中心组成,它可表示成长度为 $K \times l$ 的浮点码串。个体的适应函数定义为: $f = \frac{1}{1+J_m}$, J_m 为各样本到各自聚类中心的欧氏距离的和。实例中 GA 和 PSO 算法的实验数据来自文献[19],将其与 DRGSO 算法的数据进行比较。表 7 的实验结果表明,DRGSO 算法的平均值均低于 GA 和 PSO 算法的平均值,收敛精度较高。

表 7 对 Iris 植物样本数据聚类的实验结果

算法	平均值
GA 算法	97.93448
PSO 算法	97.47503
DRGSO 算法	97.03246

结束语 本文介绍了一种融入差分进化策略的群搜索优化算法,此算法比标准的群搜索优化算法(GSO)高效,易于实现,且参数简单可控。将种群根据适应值优劣进行排序,避免了个体寻优过程的盲目性,加快了算法达到最优值的速度。同时,增加发现者数目,引入差分策略,丰富了发现者操作的资源搜索学习机制,增强了种群的多样性,有效避免了算法陷入局部最优。基于 11 组国际标准测试函数的对比实验结果

表明,无论是对于单峰函数还是多峰函数,本文提出的算法都具有较好的整体收敛性能和较强的搜索以及勘探能力,容易跳出局部最优,具有较强的全局搜索能力。

未来的工作可以在以下两方面展开:1)将更多优化策略引入到加入者和游荡者个体中;2)将改进的群搜索优化算法应用到聚类算法研究中。

参考文献

- [1] HUANG W, OH S K, PEDRYCZ W. A space search optimization algorithm with accelerated convergence strategies [J]. Applied Soft Computing, 2013, 13(12): 4659-4675.
- [2] COLORNI A, DORIGO M, MANIEZZO V, et al. Distributed Optimization by Ant Colonies[C]//Proceedings of the 1st European Conference on Artificial Life. Paris, France, Elsevier Publishing, 1991: 134-142.
- [3] HUANG W, DING L X. The shortest path problem on a fuzzy time-dependent network [J]. IEEE Transactions on Communications, 2012, 66(11): 3376-3385.
- [4] HUANG Wei, OH S K, WITOLD P. Design of hybrid radial basis function neural networks (HRBFNNs) realized with the aid of hybridization of fuzzy clustering method (FCM) and polynomial neural networks (PNNs) [J]. Neural Networks, 2014, 60: 166-181.
- [5] HUANG W, DING L X. Project-Scheduling Problem with Random Time-Dependent Activity Duration Times [J]. IEEE Transactions on Engineering Management, 2011, 58(2): 377-387.
- [6] KENNEDY J, EBERHART R. Particle swarm optimization [C]//IEEE International Conference on Neural Networks, 1995. 1995: 1942-1948.
- [7] LI X L, SHAO Z J, QIAN J X. An Optimizing Method Based on Autonomous Animats: Fish-swarm Algorithm [J]. System Engineering Theory and Practice, 2002, 22(11): 32-38. (in Chinese)
李晓磊, 邵之江, 钱积新. 一种基于动物自治体的寻优模式: 鱼群算法[J]. 系统工程理论与实践, 2002, 22(11): 32-38.
- [8] HE S, WU Q H, SAUNDERS J R. A Novel Group Search Optimizer Inspired by Animal Behavioural Ecology[C]//IEEE Congress on Evolutionary Computation, 2006 (CEC 2006). IEEE, 2006: 1272-1278.
- [9] HE S, WU Q H, SAUNDERS J R. Group Search Optimizer: An Optimization Algorithm Inspired by Animal Searching Behavior [J]. IEEE Transactions on Evolutionary Computation, 2009, 13(5): 973-990.
- [10] WANG S W, DING L X, XIE D T, et al. Group Search Optimizer Applying Opposition-based Learning [J]. Computer Science, 2012, 39(9): 183-187. (in Chinese)
汪慎文, 丁立新, 谢大同, 等. 应用反向学习策略的群搜索优化算法[J]. 计算机科学, 2012, 39(9): 183-187.
- [11] LIU F, QIN G, LI L J. A Quick Group Search Optimizer and Its Application Research [J]. Engineering Mechanics, 2010(7): 38-44. (in Chinese)
刘锋, 覃广, 李丽娟. 快速群搜索优化算法及其应用研究[J]. 工程力学, 2010(7): 38-44.
- [12] FANG J Y. Hybrid Group Search Optimizer and its Application [D]. Taiyuan: Taiyuan University of Science and Technology, 2010. (in Chinese)
房娟艳. 混合群搜索优化算法及其应用研究[D]. 太原: 太原科技大学, 2010.
- [13] ZHANG W W, TENG S H, LI L J. Improved Group Search Optimizer algorithm [J]. Computer Engineering and Applications, 2009, 45(4): 48-51. (in Chinese)
张雯雯, 滕少华, 李丽娟. 改进的群搜索优化算法[J]. 计算机工程与应用, 2009, 45(4): 48-51.
- [14] WANG S W, DING L X, XIE C W, et al. Study on Role Assignment Strategies of Group Search Optimizer [J]. Journal of Chinese Computer Systems, 2012(9): 1938-1943. (in Chinese)
汪慎文, 丁立新, 谢承旺, 等. 群搜索优化算法中角色分配策略的研究[J]. 小型微型计算机系统, 2012(9): 1938-1943.
- [15] LIU B, WANG L, JIN Y H. Advances in Differential Evolution [J]. Control and Decision, 2007, 22(7): 721-728. (in Chinese)
刘波, 王凌, 金以慧. 差分进化算法研究进展[J]. 控制与决策, 2007, 22(7): 721-728.
- [16] GONG W, CAI Z. Differential Evolution with Ranking-Based Mutation Operators [J]. IEEE Transactions on Cybernetics, 2013, 43(6): 2066-2081.
- [17] YANG Q W, CAI L, XUE Y C. A Survey of Differential Evolution Algorithms [J]. Pattern Recognition and Artificial Intelligence, 2009, 21(4): 506-513. (in Chinese)
杨启文, 蔡亮, 薛云灿. 差分进化算法综述[J]. 模式识别与人工智能, 2009, 21(4): 506-513.
- [18] YAO X, LIU Y, LIN G. Evolutionary programming made faster [J]. IEEE Transactions on Evolutionary Computation, 1999, 3(2): 82-102.
- [19] SUN Y, LUO K. Clustering method based on improved particle swarm optimization [J]. Computer Engineering and Applications, 2009, 45(33): 132-134. (in Chinese)
孙洋, 罗可. 基于改进的粒子群算法的聚类算法[J]. 计算机工程与应用, 2009, 45(33): 132-134.
- [15] OTTENSTEIN K J, OTTENSTEIN L M. The program dependence graph in a software development environment [J]. ACM Sigplan Notices, 1984, 19(5): 177-184.
- [16] HAN Z, CHEN S H. Analysis on follow region and algorithm for construction program dependence graph [C]//China National Computer Conference. 2009: 499-507. (in Chinese)
韩喆, 陈世鸿. 跳转语句跟随域分析与程序依赖图构造算法[C]//中国计算机大会. 2009: 499-507.
- [17] Zest[OL]. [2015-12-10]: <http://www.eclipse.org/gef/zest>.

(上接第 221 页)