

文章编号: 1001-0920(2013)08-1235-07

群体区域搜索算法

刘昌军, 卫军胡, 王虹, 高一星, 孙国基

(西安交通大学 机械制造系统工程国家重点实验室, 西安 710049)

摘要: 借鉴自然界群居生物的搜索行为模式, 提出一种群体区域搜索算法. 该算法在优化过程中逐步收缩个体搜索半径并进行适度随机调整, 引入巡游追随机制, 以一种简单而自然的方式有效地实现了算法广域探索能力与局部开发能力之间的平衡. 算法结构简单、易实现, 易与其他算法相结合. 通过6个典型测试函数的实验结果表明, 该算法全局优化能力强、收敛精度高、稳定性好、总体性能优, 适用于复杂函数优化问题的处理.

关键词: 群体智能; 进化计算; 巡游追随机制; 群体区域搜索算法; 全局优化

中图分类号: TP13; TP18

文献标志码: A

Group area search for optima

LIU Chang-jun, WEI Jun-hu, WANG Hong, GAO Yi-xing, SUN Guo-ji

(State Key Laboratory for Manufacturing Systems Engineering, Xi'an Jiaotong University, Xi'an 710049, China.

Correspondent: LIU Chang-jun, E-mail: chjliu@sei.xjtu.edu.cn)

Abstract: A novel swarm intelligence optimization algorithm, group area search(GAS), is proposed, which mimics the searching behavior patterns of gregarious creatures. In the algorithm, the search radius of each member is gradually shrunk and moderately adjusted in the optimization process. Coupled with a cruising-following mechanism, GAS can achieve a good balance between global exploration and local exploitation in a natural way. With the characteristics of robustness and parallelism in nature, GAS is simple to be implemented and can easily be combined with other optimization techniques. The test results on six benchmark functions show that the proposed algorithm has excellent global optimization capability, high convergence accuracy and stability, which outperforms the other eight nature-inspired algorithms in general and can cope with heterogeneous complicated function optimization problems.

Key words: swarm intelligence; evolutionary computing; cruising-following mechanism; group area search; global optimization

0 引言

智能优化算法是近年来国内外优化界和工程界研究的一个重点和热点. 此类算法要么借鉴自然界中某个系统的行为模式, 要么模拟物理或生命体的某种现象. 例如, 遗传算法(GA)^[1]借鉴生物界的适者生存、优胜劣汰的进化思想和遗传机制来进行优化; 免疫算法(IA)^[2]是在遗传算法基础上引入了免疫概念; 模拟退火法(SA)^[3]模拟加热熔化的金属退火过程来寻找全局最优解; 蚁群算法(ACO)^[4]仿效蚂蚁利用信息素寻找最佳觅食路径的方法进行搜索; 粒子群优化算法(PSO)^[5]模拟鸟群觅食过程中的迁徙和群聚行为来进行优化; 人工蜂群算法(ABC)^[6-7]模仿蜜蜂觅食过程中的分工协作行为实现全局优化.

众多智能优化算法的成功在很大程度上得益于

其对自然生物系统中所包含的优化行为与策略的发现和借鉴. 受此启发, 本文对自然界高级生物的一些觅食行为模式进行深入分析, 通过必要的简化和抽象, 提出一种群体区域搜索算法(GAS). 该算法不仅具有一般群体智能优化算法的鲁棒性、并行性等特点, 而且简单有效, 易与其他方法相结合, 更适合复杂多模态优化问题和现实工程优化问题的处理.

1 生物搜索行为模式与群体区域搜索算法

在人类的许多搜索活动中, 经常采用区域限制搜索策略(ARS)^[8], 即首先划定较大的搜索区域进行大范围粗略搜索, 然后根据所得信息一步步缩小搜索圈, 直至找到目标或者确定目标不在该区域而大幅转移. 大量的生物学观测发现, 在大量动物物种的捕食活动中存在类似的现象. 捕食者首先将搜索努力集中在某

收稿日期: 2012-03-08; 修回日期: 2012-05-22.

作者简介: 刘昌军(1977—), 男, 博士生, 从事智能优化算法、复杂系统等研究; 孙国基(1936—), 男, 教授, 博士生导师, 从事复杂系统建模、仿真与优化方法等研究.

一特定区域,采用广泛搜索模式(长距离小转向角)搜索猎物;在发现猎物的某些迹象之后则放慢移动速度和步长,开始在小范围内来来回回细致搜寻;当密集搜索不成功时将再次快速移动进行广泛搜索^[8].捕食者可能的运动路线如图 1 所示.这种少量远程高速粗略搜寻与大量短距离低速精细搜索相间的搜索模式,可以使动物在自然环境中获得更高的觅食效率.尤其对于自然界随机稀疏分布的目标而言,这是最有效的搜索策略^[9].

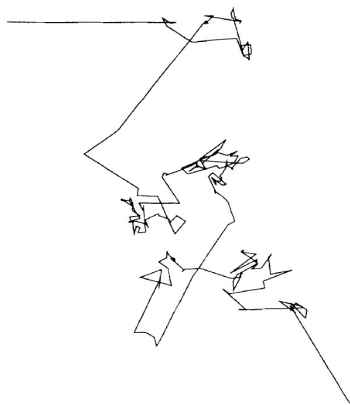


图 1 鸟类长时间飞行可能的飞行路线^[10]

为了提高觅食成功率和生存概率,生物常常群居而生,群体协作现象在自然界中普遍存在.群体成员之间通过信息共享、分工协作形成独特的优化搜索策略.根据生物学的研究,生物群体可分成两种不同的基本结构模型:信息共享模型(IS)^[11]和发现者-加入者模型(PS)^[12].加入或追随行为常见于鱼类、鸟类、狮子、猩猩等动物群体^[13].在PS模型中,发现者四处巡游寻找资源,加入者追随发现者向已发现的资源移动、分享资源,发现者和加入者共同构成搜索的主体,巡游和追随是生物的两种基本搜索行为.

基于以上分析,群体区域搜索算法模拟高级群居动物的觅食搜索行为,以PS模型为基础,采用区域限制搜索策略,实现对复杂问题的优化.在GAS算法中,个体最初都以发现者身份在较大区域内巡游(粗略搜索),然后逐步收缩各自的搜索范围,由前期的大范围粗略搜索逐渐演变为局部细致搜索.当迭代进行到一定阶段之后,较差的个体将被迫从其当前搜索区域跳离,转移到最有希望的区域附近进行搜索,成为追随者;而较优的个体则仍在其原来区域继续巡游.此即巡游-追随机制.在搜索过程中个体的角色是动态变化的.

以 2 维 Sphere 函数为例,4 个个体 30 代搜索时 GAS 个体的搜索过程如图 2 所示.由图 2 可见,GAS 个体的搜索轨迹与生物搜索路径有着高度的相似性,相当于生物个体搜索路径的某个局部.当然,群体区

域搜索算法并非生物系统的逼真描述,只是一定程度上的简化、抽象和模仿而已.

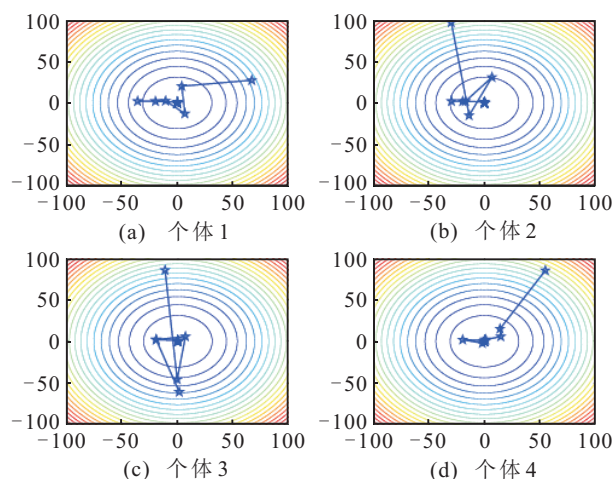


图 2 个体搜索轨迹 ($m = 4, G = 40$)

2 群体区域搜索算法

2.1 算法结构

GAS 的基本结构如图 3 所示.该算法结构简单且易于实现;在邻域随机搜索阶段对搜索方式无要求,易与其他优化算法相结合,具有内在的开放性.

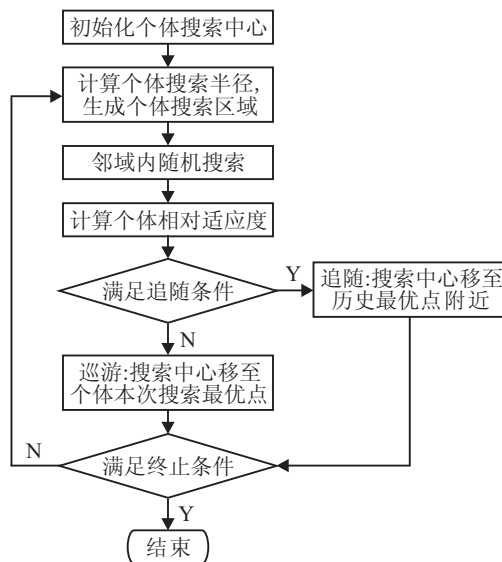


图 3 群体区域搜索算法的基本结构

对于最大化问题

$$\begin{aligned} \max & f(x_1, x_2, \dots, x_n); \\ \text{s.t.} & x_{\min j} \leq x_j \leq x_{\max j}, j = 1, 2, \dots, n. \end{aligned}$$

GAS 算法具体描述如下.

2.1.1 初始化

设定个体的数量 m , 每代搜索中个体在其邻域内随机搜索的次数 T , 搜索的终止代数 G .一般采用随机方法产生种群中个体的初始搜索中心,也可以利用确定值法或其他方法产生.具体如下:

$$x_{ij} = x_{\min j} + (x_{\max j} - x_{\min j})\text{random}(0, 1). \quad (1)$$

其中: $\text{random}(0, 1)$ 为 $[0, 1]$ 内均匀分布的随机数; i 表示个体 $i, i = 1, 2, \dots, m$.

2.1.2 区域随机搜索

1) 个体邻域搜索半径的计算.

个体邻域搜索半径调整公式为

$$R_i^k = \min \left\{ \alpha_i^k \left(1 - \frac{k}{G} \right)^\beta, 1 \right\}. \quad (2)$$

其中: α_i^k 为第 k 次迭代个体 i 的搜索半径收缩系数, β 为迭代收缩指数.

当 $k = 0$ 时, α_i^0 由以下经验公式计算:

$$\alpha_i^0 = \frac{10}{\sqrt{m \cdot G}}; \quad (3)$$

也可直接指定, 通常取值在 0.1 附近. 当 $k > 0$ 时, 有

$$\alpha_i^k = \begin{cases} \alpha_i^{k-1} \text{random}(1-d, 1+d), & \text{若个体 } i \text{ 巡游;} \\ \alpha_{\text{best}}, & \text{若个体 } i \text{ 追随.} \end{cases} \quad (4)$$

其中: $\text{random}(1-d, 1+d)$ 为 $[1-d, 1+d]$ 内均匀分布的随机数; d 为收缩系数的随机调整幅度; α_{best} 为历史最优收缩系数值, 即历史最优个体最优时刻的收缩系数值.

2) 个体搜索区域的确定.

搜索域上界为

$$u_{ij}^k = \min \{ x_{ij} + R_i^k (x_{\max j} - x_{\min j}), x_{\max j} \}; \quad (5)$$

搜索域下界为

$$l_{ij}^k = \max \{ x_{ij} - R_i^k (x_{\max j} - x_{\min j}), x_{\min j} \}. \quad (6)$$

生成搜索位置. 个体在其搜索邻域空间内进行 T 次随机搜索, 新位置坐标如下:

$$x_{ij}^t = l_{ij}^k + (u_{ij}^k - l_{ij}^k) \text{random}(0, 1), \quad (7)$$

将这 T 次搜索中的最优位置记为 X_i' .

3) 搜索中心的更新.

① 定义个体相对适应度. 在完成一代搜索后, 需要对每个个体的搜索中心进行更新. 为了描述方便, 引入如下符号:

$$f_i = \max_t f(X_i^t) = f(X_i'). \quad (8)$$

针对最大化问题, 个体相对适应度定义为

$$F_i = \begin{cases} 1, & \max_i f_i = \min_i f_i; \\ \frac{f_i - \min_i f_i}{\max_i f_i - \min_i f_i}, & \text{其他.} \end{cases} \quad (9)$$

② 更新个体搜索中心位置. 若 $F_i \geq c$ 或 $k/G \leq k_c$ (c 称为追随度, k_c 称为追随起始点), 则个体 i 作为发现者采用巡游方式, 新的搜索中心位置更新为

$$x_{ij} = x_{ij}', \quad (10)$$

同时对区域搜索半径收缩系数 α 按式 (4) 进行一定幅度的随机调整.

若 $F_i < c$ 且 $k/G > k_c$, 则个体 i 作为追随者采用追随方式, 需要首先移动到群体历史最优位置, 即

$$x_{ij} = x_{\text{best } j}, \quad (11)$$

再随机选取其中一维 q 进行适度迁移. 迁移半径计算如下:

$$R_{iq} = \min \left\{ 0.05(s+1)R_i^k G \left(1 - \frac{k}{G} \right)^2 \text{random}(0, 1), 1 \right\}, \quad (12)$$

其中 s 为算法陷入停滞的代数, 即历史最优值累计不再刷新的代数.

然后按照 2.1.2 节中 2) 同样的步骤在生成的迁移区域内进行 3 次随机搜索得到最优位置 x_{iq} , 必要时也可进行多次迁移搜索. GAS 算法对区域内搜索方式不作要求, 可以使用其他搜索方式.

最后调整区域搜索半径收缩系数 $\alpha_i^k = \alpha_{\text{best}}$.

2.1.3 终止判断

一般地, 当算法达到最大允许迭代次数或者已获得了满足要求的满意解时, 搜索结束, 即

$$f_{\max} \geq f_{\text{opt}} \parallel k \geq G.$$

2.2 算法具体流程

GAS 算法流程可描述如下.

Step 1: 初始化. 设定算法参数, 对个体搜索中心进行随机初始化.

Step 2: 各区域随机搜索. 按式 (2) 计算个体的邻域搜索半径, 按式 (5) 和 (6) 生成个体搜索邻域的边界, 在区域内进行 T 次随机搜索.

Step 3: 计算适应度. 标记个体当次搜索的最优位置, 按式 (9) 计算个体相对适应度.

Step 4: 如果个体 i 的适应度 $F_i < c$ 且 $k/G > k_c$, 则追随到历史最优位置附近, 随机选取一维进行迁移, 并由式 (12) 计算迁移半径, 按式 (5) 和 (6) 得到迁移区间, 在区间内进行 3 次随机搜索, 取其最优位置作为新的搜索中心, 并令 $\alpha_i^k = \alpha_{\text{best}}$, 转 Step 5; 否则, 个体 i 采用巡游方式, 按式 (10) 更新搜索中心, 并按式 (4) 调整 α , 然后转 Step 5.

Step 5: 终止判断. 若满足终止条件, 则结束搜索过程, 输出优化结果; 否则转 Step 2.

3 算法测试与比较分析

3.1 测试问题

为了测试 GAS 的性能, 用 6 个经典的 Benchmark 问题对算法进行测试. 问题如下:

Sphere 函数

$$f_1(x) = \sum_{i=1}^n x_i^2;$$

Rosenbrock 函数

$$f_2(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2];$$

Rastrigin 函数

$$f_3(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10];$$

Griewank 函数

$$f_4(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1;$$

Ackley 函数

$$f_5(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + e;$$

Schwefel 函数

$$f_6(x) = -\sum_{i=1}^n x_i \sin(\sqrt{|x_i|}).$$

3.2 β 取值对算法性能的影响

在 GAS 算法中, 搜索半径调整策略和追随策略是影响算法性能的两大关键因素, 而半径收缩指数 β 是决定搜索半径调整过程的关键参数. 下面对 β 的影响进行详细分析.

实验中 GAS 的参数设置为: 种群数 $m = 10$, 邻域随机搜索数 $T = 10$, 终止代数 $G = 2000$, 其余参数见表 1. 表 2 给出了算法在不同 β 取值时 200 次独立运行的实验结果.

表 1 测试函数与参数设置

函数	名称	维数	搜索范围	最优值	容许误差	参数设置 (β, k_c, c, d)
f_1	Sphere	30	$[-100, 100]^N$	0	0.01	15, 0.0, 0.75, 0.5
f_2	Rosenbrock	30	$[-30, 30]^N$	0	100	5, 1.0, 0.0, 0.02
f_3	Rastrigin	30	$[-5.12, 5.12]^N$	0	100	15, 0.0, 0.99, 0.02
f_4	Griewank	30	$[-600, 600]^N$	0	0.01	15, 0.65, 0.75, 0.02
f_5	Ackley	30	$[-32, 32]^N$	0	0.01	15, 0.0, 0.9, 0.02
f_6	Schwefel	30	$[-500, 500]^N$	-12569.5	-10000	5, 0.0, 0.99, 0.02

表 2 不同 β 时的实验结果

函数	性能指标	$\beta = 1$	$\beta = 3$	$\beta = 5$	$\beta = 10$	$\beta = 15$	$\beta = 20$
f_1	Mean	4.062 69e-009	5.999 84e-047	3.131 91e-079	4.414 08e-103	1.018 46e-107	1.256 74e-105
	Std Dev	5.729 36e-008	8.463 81e-046	3.118 96e-078	6.047 25e-102	1.372 88e-106	1.665 1e-104
	Suc Rate	100%	100%	100%	100%	100%	100%
	Best	7.402 55e-115	2.554 56e-116	2.548 36e-119	1.686 16e-123	5.803 9e-125	1.535 37e-127
f_2	Mean	27.853	23.753 5	22.814 3	23.635 6	23.982 9	23.984 4
	Std Dev	2.736 97	1.917 33	2.185 19	1.930 27	2.342 43	2.477 34
	Suc Rate	100%	100%	100%	100%	100%	100%
	Best	21.423 8	16.238 2	12.751 2	15.613 2	11.102 6	9.077 88
f_3	Mean	0.003 956 31	1.374e-010	1.163 51e-015	0	0	0.034 823 6
	Std Dev	0.002 476 89	1.520 73e-010	1.604 61e-015	0	0	0.182 853
	Suc Rate	100%	100%	100%	100%	100%	100%
	Best	0.000 510 308	7.176 48e-013	0	0	0	0
f_4	Mean	0.042 948 1	0.014 272 1	0.005 147 71	1.110 22e-018	0	2.775 56e-018
	Std Dev	0.048 309 6	0.016 753 3	0.007 036 87	1.104 66e-017	0	3.224 91e-017
	Suc Rate	23.5%	58.5%	80%	100%	100%	100%
	Best	5.840 01e-005	2.435 07e-011	2.220 45e-016	0	0	0
f_5	Mean	0.026 624 3	3.643 84e-006	5.711 34e-009	2.735 84e-014	1.245 48e-014	1.233 04e-014
	Std Dev	0.006 632 08	1.936 66e-006	2.502 11e-009	6.887 85e-015	2.202 68e-015	2.107 74e-015
	Suc Rate	0%	100%	100%	100%	100%	100%
	Best	0.011 169 7	6.970 11e-007	7.467 74e-010	1.479 96e-014	7.694 15e-015	7.694 15e-015
f_6	Mean	-12 569.5	-12 569.5	-12 569.5	-12 569.5	-12 569.5	-12 569.5
	Std Dev	0.012 005 3	1.104 29e-009	1.291 74e-011	1.356 34e-011	1.349 92e-011	1.352 67e-011
	Suc Rate	100%	100%	100%	100%	100%	100%
	Best	-12 569.5	-12 569.5	-12 569.5	-12 569.5	-12 569.5	-12 569.5

由表 2 可见, 在 6 组不同的 β 取值条件下 GAS 在所有 6 个测试问题上大都达到良好的优化效果. 在一定范围内 GAS 的优化质量随着 β 的增大而显著提高, 但当 β 取值大于某一阈值后平均性能将不再提高, 如 f_2 、 f_6 , 甚至可能有所恶化, 如 f_1 、 f_3 、 f_4 .

产生上述现象的原因在于 β 的取值与个体邻域搜索半径的收缩速度直接相关. β 取值越大, 搜索半径的收缩速度越快, 这直接造成算法进行大范围探索的时间缩短, 而进行小范围聚焦开发的时间增加. 因此 β 取值增大可以加快算法的收敛速度, 提高算

法的局部搜索精度. 这对于简单的单峰二次凸函数 Sphere (f_1) 非常有益; 但是对于复杂的优化问题则未必完全适宜. 如对于单模态问题 Rosenbrock (f_2), 由于 GAS 搜索本质上的随机性, 函数广阔平坦的谷底致使算法后期搜索缓慢, 提高 β 取值对算法性能影响不大. 对于较复杂的多模态问题, β 取值过大反而易使算法过早收敛而影响全局优化性能, 典型的如 Rastrigin (f_3), 其唯一的全局最优点被众多局部次优点包围, 过大的 β 值会使算法陷入局部最优点的概率增大, 可能对算法的全局性能造成不利影响.

由上述分析可知, 对于不同的优化问题, 存在不同的最优 β 取值. 针对简单函数, 可以选择较大的 β 值以获得更理想的收敛速度和搜索精度; 针对复杂问题, β 的取值应适中以免影响算法的全局性能.

3.3 算法比较与分析

为了评估 GAS 的优化效果, 首先选用 GA 和两种 PSO 算法进行对比. GA 使用 Matlab 7.5 优化与直接搜索工具箱中的 GA 算法, PSO 分别采用 Standard PSO 2007 (SPSO)^[14]和 CLPSO^[15]. GA 和 SPSO 的群体大小为 50, 最大进化代数为 4 000, 其余参数保持默认设置. CLPSO 最大函数评估次数为 200 000, 其余参数参照文献[15]设置. GAS 参数设置同前. 所有实验均在 2.80 GHz Intel 处理器、2.0 GB 内存的个人 PC 上进行, 操作系统为 Windows XP3. SPSO 和 GAS 采用 C/C++ 实现, GA 和 CLPSO 为 Matlab 编码. 独立运行 20 次实验的结果见表 3. 为了增强结论的可靠性, 选择近年一些主要智能优化算法的相关文献[16-20]的结果与 GAS 进行进一步比较, 具体如表 4 所示.

表 3 算法性能比较

算法	性能指标	f_1	f_2	f_3	f_4	f_5	f_6
GA	Mean	8.642 160e-009	11.084 32	9.004 381	3.952 870e-010	6.884 902e-005	-118.359 0
	Std Dev	3.179 584e-009	21.386 45	3.853 121	2.175 309e-010	1.480 582e-005	8.437 189e-010
	Suc Rate	100%	100%	100%	100%	100%	0%
	Best	3.454 765e-009	1.955 531e-004	3.979 838	1.304 596e-010	4.525 440e-005	-118.359 0
SPSO	Mean	3.575 169e-066	25.195 47	30.696 93	2.587 507e-003	6.838 974e-015	-9 491.464
	Std Dev	7.553 908e-066	20.663 06	11.291 83	4.832 496e-003	1.421 085e-015	744.542 0
	Suc Rate	100%	100%	100%	95%	100%	30%
	Best	7.635 863e-069	5.743 324	3.996 803e-015	0	3.996 803e-015	-10 970.56
CLPSO	Mean	6.000 900e-019	8.053 552	6.748 212e-009	3.568 636e-009	3.335 956e-010	-12 569.5
	Std Dev	2.712 887e-019	3.748 186	8.006 426e-009	4.283 911e-009	1.163 455e-010	2.860 897e-012
	Suc Rate	100%	100%	100%	100%	100%	100%
	Best	2.397 911e-019	2.542 682	3.271 357e-009	1.595 361e-009	1.468 621e-010	-12 569.5
GAS	Mean	3.112 74e-115	22.564 7	0	0	1.177 98e-014	-12 569.5
	Std Dev	6.342 24e-115	2.047 06	0	0	2.033 13e-015	1.772 93e-012
	Suc Rate	100%	100%	100%	100%	100%	100%
	Best	3.173 15e-122	18.115 1	0	0	7.694 15e-015	-12 569.5

表 4 与相关文献结果比较

算法	性能指标	f_1	f_2	f_3	f_4	f_5	f_6
ALEP ^[16]	Mean	6.32e-004	43.40	5.85	2.4 e-002	1.9 e-002	-11 469.2
	Std Dev	7.6 e-005	31.52	2.07	2.8 e-002	1.0 e-003	58.2
FEP ^[17]	Mean	5.7 e-004	5.06	4.6 e-002	1.6 e-002	1.8 e-002	-12 554.5
	Std Dev	1.3 e-004	5.87	1.2 e-002	2.2 e-002	2.1 e-002	52.6
GAAP ^[18]	Mean	6.67e-009	NA	1.02e-006	2.2e-005	1.63e-004	-12 569.5
	Std Dev	1.89e-004	NA	4.58e-009	2.05e-008	9.1e-007	5.77e-007
AEPSO ^[19]	Mean	1.70e-023	4.318 630	0	0	2.70e-011	-10 050.368
	Std Dev	1.63e-023	5.0148 65	0	0	6.12e-012	352
JADE ^[20]	Mean	1.3e-054	0.32	0	2.0e-004	4.4e-015	-12 562.387
	Std Dev	9.2e-054	1.1	0	1.4e-003	0	28
GAS	Mean	3.112 74e-115	22.564 7	0	0	1.177 98e-014	-12 569.5
	Std Dev	6.342 24e-115	2.047 06	0	0	2.033 13e-015	1.772 93e-012

表 3 和表 4 的结果表明, 对于所有测试问题, GAS 都获得了较好的优化效果, 表现出良好的稳定性. 具体而言, GAS 在其中 4 个问题 f_1 、 f_3 、 f_4 与 f_6 上效果理想, 具有显著优势; 对于多模态问题, 如 f_3 、 f_4 、 f_6 , GAS 均达到最优解, 对 f_5 的优化结果也与最优结果相近; 对于单模态问题, 在所有 9 个算法中, GAS 在

f_1 上的优化精度是最高的, 在 f_2 上的结果居中. 总体而言, GAS 无论是对于单模态还是多模态 Benchmark 问题, 均具有优良的优化能力, 显示出良好的适应性和鲁棒性, 总体性能优于其他 8 种算法 (见表 5). 究其原因, GAS 优秀的优化性能主要源于合理的邻域半径调整策略和巡游-追随机制. 具体分析如下:

表 5 算法性能排序(均值)

算法	f_1	f_2	f_3	f_4	f_5	f_6	平均排名	总体排名
ALEP	9	8	7	9	9	6	8	9
FEP	8	3	6	8	8	5	6.33	7
GA	7	5	8	3	6	9	6.33	7
SPSO	2	7	9	7	2	8	5.83	6
GA-API	6	NA	5	5	7	3	5.2	5
CLPSO	5	4	4	4	5	2	4	4
AEPSO	4	2	1	1	4	7	3.17	3
JADE	3	1	1	6	1	4	2.67	2
GAS	1	6	1	1	3	1	2.17	1

1) 邻域半径调整公式使算法随着搜索过程的进行不断调整个体的搜索范围,使其快速收敛到局部或全局极值点附近;然后减小半径的调整速度,使搜索焦点聚于极值区域进行高密度细致寻优.这样既可以提高算法收敛速度,也可以提高算法的局部搜索精度,较好地平衡了全局探索能力与局部开发能力之间的矛盾.另外,搜索半径的适度随机调整(通过 α 的调节实现)在一定程度上起到了加速收敛、提高算法适应性的作用,这在函数 f_1 的优化过程中表现明显.

2) GAS的巡游-追随机制可使相对适应度较差的搜索个体有机会从当前局部跳出,转移到最有希望区域附近.这些采用追随行为方式的个体将在最有希望区域适度迁徙进行搜索.如果搜索失败,追随个体的迁移半径将不断加大,以使其能够在保留有一定历史最优信息的新区域内开始搜索.这有点类似于进化类算法的变异操作,既有利于强化个体的全局搜索能力,同时又可以产生群聚效应,因此能够有效提高算法的搜索效率和收敛精度.当个体采用巡游行为方式时,它在邻域随机搜索中发现的最佳位置将被接受作为下次迭代的搜索中心,并不考虑其与上次搜索中心的优劣,这有利于个体发现较大范围内的潜在目标,从而使算法获得较好的全局优化性能.在搜索过程中,所有个体根据自身相对适应度的大小和搜索阶段的需要在巡游和追随之间不断切换.巡游-追随机制反映了高级群居动物群体和人类社会组织中大量存在的竞争与协作行为,具有天然的优化功能.

总之,GAS对以上6个问题表现出良好的适用性,通过合理的选择算法参数,可以获得比较理想的优化效果.

4 结 论

本文模拟人类和群居生物的搜索行为,提出一种新的群体智能优化算法——群体区域搜索算法.该算法根据动物的区域限制搜索策略和PS结构模型,在不断调整个体邻域搜索半径的基础上设立了巡游-追随机制.巡游方式有利于个体进行大范围搜索,发现潜在目标,使算法获得良好的全局优化能力;追随则使算法能够在搜索进行到一定阶段之后将适应度较

差的个体从其当前搜索区域转移投放到最有希望区域附近进行搜索,既提高了算法的求解精度,又加快了算法的收敛速度.巡游-追随机制巧妙而自然地实现了广域探索与局部开发之间的动态平衡.对6个标准测试函数的仿真实验以及与其他8种不同的群体智能优化算法的比较结果显示,本文提出的GAS算法在各测试问题上均达到了优良的优化质量,在其中4个问题上效果理想,总体性能优于其他8种算法,尤其在复杂多模态优化问题的处理上表现出潜在优势.GAS算法结构简单、容易实现,具有内在的鲁棒性和并行性,对区域内搜索方式无要求,易与其他算法结合,适用于复杂多模态黑箱问题的求解,可应用于复杂工程优化问题.

参考文献(References)

- [1] Holland J H. Adaptation in natural and artificial systems[M]. Cambridge: MIT Press, 1992: 1-228.
- [2] Dasgupta D. Artificial immune systems and their applications[M]. Heidelberg: Springer-Verlag, 1998: 1-300.
- [3] Kirkpatrick S, Gelatt C D, Vecchi M P, et al. Optimization by simulated annealing[J]. Science, 1983, 220(4598): 671-680.
- [4] Dorigo M, Maniezzo V, Colomi A. Ant system: Optimization by a colony of cooperating agents[J]. IEEE Trans on Systems, Man, and Cybernetics, Part B: Cybernetics, 1996, 26(1): 29-41.
- [5] Kennedy J, Eberhart R. Particle swarm optimization[C]. Proc of the 1995 IEEE Int Conf on Neural Networks. 1995: 1942-1948.
- [6] Karaboga D. An idea based on honey bee swarm for numerical optimization[R]. Turkey: Computer Engineering Department, Erciyes University, 2005.
- [7] Karaboga D, Basturk B. On the performance of artificial bee colony(ABC) algorithm[J]. Applied Soft Computing, 2008, 8(1): 687-697.
- [8] Krakauer D C, Rodríguez-Gironés M A. Searching and learning in a random environment[J]. J of Theoretical Biology, 1995, 177(4): 417-419.
- [9] Michael F Shlesinger. Search research[J]. Nature, 2006, 443(21): 281-282.
- [10] Viswanathan G M, Afanasyev V, Buldyrev S V, et al. Lévy flight search patterns of wandering albatrosses[J]. Nature, 1996, 381: 413-415.
- [11] Clark C W, Mangel M. Foraging and flocking strategies: Information in uncertain environment[J]. American Naturalist, 1984, 12(3): 626-641.

- [12] Bamard C J, Sibly R M. Producers and scroungers: A general model and its application to captive flocks of house sparrows[J]. *Animal Behaviour*, 1981, 29(2): 543-550.
- [13] Giraldeau L A, Beauchamp G. Food exploitation: Searching for the optimal joining policy[J]. *Trends in Ecology and Evolution*, 1999, 14(3): 102-106.
- [14] Particle swarm central[EB/OL]. [2012-03-12]. <http://www.particleswarm.info/Programs.html#Standard-PSO-2007>.
- [15] Liang J J, Qin A K, Suganthan P N, et al. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions[J]. *IEEE Trans on Evolutionary Computation*, 2006, 10(3): 281-295.
- [16] Lee C Y, Yao X. Evolutionary programming using mutations based on the levy probability distribution[J]. *IEEE Trans on Evolutionary Computation*, 2004, 8(1): 1-13.
- [17] Yao Xin, Liu Yong, Lin Guangming. Evolutionary programming made faster[J]. *IEEE Trans on Evolutionary Computation*, 1999, 3(2): 82-102.
- [18] Ciornei I, Kyriakides E. Hybrid ant colony-genetic algorithm(GA-API) for global continuous optimization[J]. *IEEE Trans on Systems, Man, and Cybernetics, Part B: Cybernetics*, 2012, 42(1): 234-245.
- [19] 高卫峰, 刘三阳. 一种高效粒子群优化算法[J]. *控制与决策*, 2011, 26(8): 1158-1162.
(Gao W F, Liu S Y. An efficient particle swarm optimization[J]. *Control and Decision*, 2011, 26(8): 1158-1162.)
- [20] Zhang J Q, Sanderson A C. JADE: Adaptive differential evolution with optional external archive[J]. *IEEE Trans on Evolutionary Computation*, 2009, 13(5): 945-958.

(上接第1234页)

- [2] 章仁为. 卫星轨道姿态动力学与控制[M]. 北京: 北京航空航天大学出版社, 1998: 293-296.
(Zhang R W. *Satellite orbit and attitude dynamics and control*[M]. Beijing: BeiHang University Press, 1998: 293-296.)
- [3] 李铁军, 黄海. 基于误差四元数分解的刚体姿态跟踪滑模控制[J]. *中国空间科学技术*, 2009, 29(4): 28-36.
(Li T J, Huang H. Decomposed error quaternion-based rigid body attitude tracking system with sliding mode control[J]. *Chinese Space Science and Technology*, 2009, 29(4): 28-36.)
- [4] 宋斌, 卜劭华, 颜根廷. 航天器姿态跟踪系统的非线性鲁棒自适应控制[J]. *上海航天*, 2009, 26(5): 1-6.
(Song B, Bu S H, Yan G T. Nonlinear robust adaptive control for spacecraft attitude tracking system[J]. *Aerospace Shanghai*, 2009, 26(5): 1-6.)
- [5] 梅生伟, 申铁龙, 刘志康. 现代鲁棒控制理论与应用[M]. 北京: 清华大学出版社, 2009: 64-67.
(Mei S W, Shen T L, Liu Z K. *Theory and application of modern robust control*[M]. Beijing: Tsinghua University Press, 2009: 64-67.)
- [6] 魏静波, 刘昆. 三轴稳定小卫星LQG姿态稳定控制与仿真[J]. *系统仿真学报*, 2011, 8(23): 1739-1742.
(Wei J B, Liu K. LQG controller and simulation for three-axis stabilized small satellite[J]. *J of System Simulation*, 2011, 8(23): 1739-1742.)
- [7] Zhou Di, Shen Tielong, Tamura Katsutoshi. Nonlinear controllers for attitude stabilization and tracking of a spacecraft[J]. *Trans on The Japan Society for Aeronautical and Space Sciences*, 2005, 48(159): 7-12.
- [8] Pierluigi Pisu, Andrea Serrani. Attitude tracking with adaptive rejection of rate gyro disturbances[J]. *IEEE Trans on Automatic Control*, 2007, 52(12): 2374-2379.
- [9] Luo Wencheng, Chu Yunchung. Inverse optimal adaptive control for attitude tracking of spacecraft[J]. *IEEE Trans on Automatic Control*, 2005, 50(11): 1639-1654.
- [10] Hu Qinglei. Robust adaptive attitude tracking control with L_2 -gain performance and vibration reduction of an orbiting flexible spacecraft[J]. *J of Dynamic Systems, Measurement, and Control*, 2011, 133(1): 1-11.
- [11] Xia Yuanqing, Zhu Zheng, Fu Mengyin, et al. Attitude tracking of rigid spacecraft with bounded disturbances[J]. *IEEE Trans on Industrial Electronics*, 2011, 58(2): 647-659.
- [12] 卜劭华, 宋斌. 航天器姿态跟踪控制系统的自适应滑模控制[J]. *控制工程*, 2009: 15(S1): 38-41.
(Bu S H, Song B. Adaptive sliding mode control for spacecraft attitude tracking system[J]. *Control Engineering of China*, 2009: 15(S1): 38-41.)
- [13] Jing Wu-xing, Xu Shi-jie. Nonlinear attitude tracking control of a spacecraft with thrusters based on error quaternion[J]. *Chinese J of Aeronautics*, 2002, (15)3: 130-138.
- [14] Anton H J de Ruiter. Adaptive spacecraft attitude tracking control with actuator saturation[J]. *J of Guidance, Control, and Dynamics*, 2010: 33(5): 1692-1695.