

具有混合群智能行为的萤火虫群优化算法研究

吴斌 崔志勇 倪卫红

(南京工业大学工业工程系 南京 210009)

摘要 萤火虫群优化算法是一种新型的群智能优化算法,基本的萤火虫群优化算法存在收敛精度低等问题。为了提高算法的性能,借鉴蜂群和鸟群的群体智能行为,改进萤火虫群优化算法的移动策略。运用均匀设计调整改进算法的参数取值。若干经典测试问题的实验仿真结果表明,引入混合智能行为大幅提升了算法的优化性能。

关键词 萤火虫群优化算法,人工蜂群算法,粒子群算法,全局优化

中图分类号 TP18 **文献标识码** A

Research on Glowworm Swarm Optimization with Hybrid Swarm Intelligence Behavior

WU Bin CUI Zhi-yong NI Wei-hong

(Department of Industrial Engineering, Nanjing University of Technology, Nanjing 210009, China)

Abstract Glowworm swarm optimization(GSO) algorithm is the one of the newest nature inspired heuristics for optimization problems. In order to enhance accuracy and convergence rate of the GSO, two behaviors which are inspired by artificial bee colony algorithm(ABC) and particle swarm optimization(PSO) of the movement phase of GSO were proposed. The effects of the parameters about the improvement algorithms were discussed by uniform design experiment. A number of experiments were carried out on a set of well-known benchmark global optimization problems. Numerical results reveal that the proposed algorithms can find better solutions compared with classical GSO and other heuristic algorithms and are powerful search algorithms for various global optimization problems.

Keywords Glowworm swarm optimization, Artificial bee colony algorithm, Particle swarm optimization, Global optimization

1 引言

自然界中,萤火虫有许多种类,不同种类的萤火虫发光的目的不同,有的用来吸引伴侣,有的用来捕食猎物。通常,光强度大的萤火虫更能吸引伴侣或更容易捕食猎物,形成多数萤火虫聚集的现象。受此行为启发,基于群智能理论,Krishnanand 和 Ghose^[1]在2005年提出萤火虫群优化算法(Glowworm swarm optimization, GSO)。与其他基于群智能理论的优化算法,如蚁群算法(Ant Colony Optimizaiton, ACO)和粒子群优化算法(Particle Swarm Optimzation, PSO)有类似的信息共享机制,但它也有自身的特点:在GSO算法中,萤火虫发光的强度由其自身携带的荧光素(Luciferin)决定,而荧光素的量与萤火虫所在位置的目标函数有关,即发光越亮的萤火虫所在位置的目标值越优。萤火虫拥有自己的视线范围,不同于其他算法的固定区域。在GSO算法中,萤火虫依靠动态决策来选择自己的邻居,并控制自身下一时刻的移动。萤火虫总是选择荧光素值比自己高的个体作邻居,并用概率机制进行移动选择。这种仅依赖局部信息进行的搜索,使萤火虫不会像其他算法那样所有的个体聚集在一起,而是分成若干群体,形成若干个聚集区域。这样算法整体上不容易

陷入局部极值点,但会影响算法收敛的速度和精度。Krishnan 和 Ghose^[2]从理论上分析了GSO算法的收敛性,并在多峰函数中进行了大量的仿真实验,以验证算法的性能^[3,4]。他们还将GSO算法用于群体移动机器人的协同运作优化中^[5]。Bharat^[6]将GSO用于工程参数特征值的估计中。He 和 Zhu^[7]研究了多种群GSO算法。李咏梅、周永权等^[8]研究了混合人工鱼群算法的GSO算法。综上所述,GSO算法的研究刚刚起步,在算法改进和应用方面有大量研究工作要做。

本文针对算法中的移动过程,借鉴蜂群算法和粒子群算法中个体产生机制,引入新的个体位置移动策略。实验仿真证明改进算法大幅提升了算法的性能。

2 基本萤火虫群优化算法

在基本萤火虫群优化算法中,初始有 NP 个萤火虫随机地散布在 D 维空间,萤火虫 i 当前所在位置用 x_i 表示,每个萤火虫携带的荧光素量用 l_i 表示。在算法伊始,所有萤火虫的荧光素都是 l_0 ,荧光素越多,发光强度越大。每个萤火虫都有一个感知范围 r_d ($0 < r_d < r_s$),在这个范围内,萤火虫选择光强度比自己高的邻居,并向其移动,初始阶段所有的感知范围都是 r_s 。在完成算法的初始设置以后,GSO算法进入循环

到稿日期:2011-06-24 返修日期:2011-09-15 本文受江苏省自然科学基金(BK2010555),教育部人文社科青年项目(11YJCZH184)资助。

吴斌(1979-),男,博士,讲师,主要研究方向为计算智能、系统建模与仿真, E-mail: seasimen@yahoo. com. cn; 崔志勇(1983-),男,硕士,实验师,主要研究方向为计算智能;倪卫红(1968-),女,硕士,副教授,主要研究方向为物流供应链管理。

迭代过程,主要包括以下3个过程:荧光素更新、位置更新、感知范围更新。

2.1 荧光素更新阶段

萤火虫的发光亮度与其所在的位置密切相关。所在位置的目标函数值越好,则荧光素增长越多,光强度也越高,这是一种正反馈机制。同时,荧光素除了受所在位置影响,还和萤火虫上一时刻的携带量和挥发速度有关,荧光素根据式(1)更新:

$$l_i(t+1) = (1-\rho)l_i(t) + \gamma J_i(t+1) \quad (1)$$

式中, $l_i(t)$ 表示萤火虫 i 在 t 时刻荧光素的量, ρ ($0 < \rho < 1$) 表示荧光素的挥发速度, γ 表示萤火虫加强系数, $J_i(t+1)$ 表示萤火虫 i 在 $t+1$ 位置的目标函数值。

2.2 位置更新阶段

萤火虫在解空间的移动,采用基于贪婪策略的随机选择机制。萤火虫 i 向自己的同伴 j 移动,要满足两个条件:

- (1) j 首先要在 i 的感知范围内;
- (2) j 的发光强度要高于 i 。

在所有满足条件的同伴中,萤火 i 依据式(2)计算每个同伴的选择概率:

$$p_{ij} = \frac{l_j(t) - l_i(t)}{\sum_{k \in N_i(t)} l_k(t) - l_i(t)} \quad (2)$$

式中, $j \in N_i(t)$, $N_i(t) = \{j; d_{i,j}(t) < r_d^i(t); l_i(t) < l_j(t)\}$ 是萤火虫 i 在 t 时刻的同伴集合, $d_{i,j}(t)$ 表示萤火虫 i 和 j 在 t 时刻的距离, $r_d^i(t)$ 表示萤火虫 i 在 t 时刻的感知范围。萤火虫 i 依概率选择出同伴 j 后,根据式(3)移动:

$$x_i(t+1) = x_i(t) + s \left(\frac{x_j(t) - x_i(t)}{\|x_j(t) - x_i(t)\|} \right) \quad (3)$$

式中, s 表示步长, $\|\cdot\|$ 表示欧式距离。

2.3 感知范围更新

在移动后,萤火虫会根据感知范围内同伴的数量动态调整感知范围的大小。如果数量多,就缩小感知范围,反之扩大感知范围,具体根据式(4)进行计算:

$$r_d^i(t+1) = \min\{rs, \max\{0, r_d^i(t) + \beta(nt - |N_i(t)|)\}\} \quad (4)$$

式中, β 是控制感知范围大小的参数, nt 是控制感知范围内同伴数量的参数。

3 改进萤火虫群优化算法

已有的研究结果显示, GSO 算法在多维、多峰函数优化中,存在优化精度低、收敛速度慢等问题。出现这种情况,很大程度上是 GSO 算法中移动更新策略造成的。在基本 GSO 算法中,萤火虫个体在感知范围内,随机选择发光强度大于自己的同伴并向其移动。这种移动是在 D 维空间内的整体移动,移动后的新位置的目标函数可能劣于原来位置,这在高维多峰函数中会经常发生。针对这种情况,本文提出两个改进策略:

(1) 从整体移动变为部分移动,即逐维地移动。当出现目标函数改进的情况,停止移动。将基本算法中的 **整体移动** 无条件接受 **新位置**, 改为 **逐维移动** 更优接受 **新位置**。

(2) 改进移动方式。借鉴人工蜂群算法和粒子群算法的个体产生机制,改进式(3),提出如下两种移动策略:

$$x_{ik}(t+1) = x_{ik}(t) + r \times (x_{jk}(t) - x_{ik}(t)) \quad (5)$$

$$x_{ik}(t+1) = \omega \times x_{ik}(t) + r \times c_1 \times (x_k^{\text{best}}(t) - x_{ik}(t)) + r \times c_2 \times (x_{jk}(t) - x_{ik}(t)) \quad (6)$$

上述公式都是针对萤火虫所在位置的第 k ($0 < k < D$) 维进行的移动,其中式(5)是借鉴人工蜂群算法中觅食蜂寻找新蜜源的方式,总体上与基本 GSO 算法的方式类似,但 x_j 的选择不再基于贪婪策略,而是从萤火虫的感知范围内随机选择一个而不考虑其发光强度。这种改进的移动方式基于局部移动,随机地选择加上逐维搜索和更优接受策略,既可以保持群体的多样性,又能保证收敛的精度和速度。式(6)中除了有邻居 x_j 的影响外,还有个体自身以及种群中最优个体 x^{best} 的影响,新的位置综合考虑了自身、邻居和最优个体,这是粒子群算法基于社会认知心理学的角度建立的状态更新公式。本文将其改造成逐维搜索的形式,用于 GSO 的搜索策略。式(6)中的 ω, c_1, c_2 分别表示各部分在新位置中所占的权重。改进后 GSO 算法的流程如下:

Step1 初始化所有参数,如 $l_0, MCN, \rho, \gamma, \beta, N_t$ 等。

Step2 设置 $t=0$, 随机初始化每个萤火虫的位置 $x_i = (x_{i1}, x_{i2}, \dots, x_{iD})$, 设置 $l_i(t) = l_0, r_d^i(t) = r_s$ 。

Step3 如果 $t \leq MCN$, 算法进行循环, 否则转 Step5。

Step3.1 荧光素更新阶段。根据式(1)计算每个萤火虫在 $t+1$ 时刻的荧光素量。

Step3.2 位置更新。对于每个萤火虫 $x_i(t)$, 首先计算出其邻居集合 $N_i(t)$, 然后根据式(2)计算集合中每个邻居的选择概率 p_{ij} , 依概率选择出邻居 j , 进行如下部分移动: 当 $k \leq D$ 且 $J_i(t) > J_i(t+1)$ 时, 用式(5)或式(6)计算新的位置, 直到出现更优解; 如果所有 $k=D$ 后仍没有改进, 则保持原位置不变。

Step3.3 感知范围更新。每个萤火虫根据式(4)计算其下一时刻的 $r_d^i(t+1)$ 。

Step4 $t=t+1$, 转 Step3。

Step5 算法结束。

改进的 GSO 算法依据位置更新中移动策略的不同分为两类: 执行式(5)蜂群移动策略的, 用 AGSO 表示; 执行式(6)粒子群移动策略的, 用 PGSO 表示。下面通过实验仿真来研究改进算法的性能。

4 实验仿真

为了验证分析改进 GSO 算法的性能, 本文从标准 Benchmark 中选择 12 个函数进行实验仿真。各函数的性质、取值范围、理论最优解如表 1 所列, 函数具体形式见文献[9]。算法用 C++ 实现, 运行在 Pentium E2200 CPU、2G 内存的计算机平台上。实验中函数的变量维数分别取 $D=10$ 和 $D=30$ 。在实验中, 每个函数进行 30 次蒙特卡洛模拟, 统计其最小值(Min)和平均值(Mean)。

参数对智能优化算法的性能有重要的影响, 而 AGSO 算法中涉及 $l_0, \rho, \gamma, \beta, rs, nt$ 共 6 个参数, Krishnanand 等人给出了参数的大致取值范围, 其中 $l_0 \in [0, 5], \rho, \gamma, \beta \in [0, 1], nt \in [0, NP], rs \in (0, 1) \times \|ub - lb\|$, 这里 ub 和 lb 分别是函数变量的取值范围。改进的 PGSO 算法中, 另有 3 个参数 $\omega \in [0, 1], c_1, c_2 \in [0, 2]$ 。为了更好地调整这些参数的取值, 本文根据均匀设计进行参数调整的实验。AGSO 算法有 6 个参数, 查均匀设计表可知, 选择 $U_{20}^{(20)}$ 表进行实验, 每个参数

根据取值范围,均匀选取 20 个值进行 20 组实验,实验方案可参考文献[10]中的均匀设计表。实验中,种群规模设置为 $NP=100$,迭代次数 $MCN=5000$,用 Sphere 函数进行测试,统计结果如图 1 所示。

表 1 测试函数

函数名	特点	取值范围
Sphere	单峰	$[-5.12, 5.12]$
Step	单峰	$[-100, 100]$
Quartic	单峰	$[-1.28, 1.28]$
Dixon-Price	单峰	$[-10, 10]$
Schwefel 1.2	单峰	$[-100, 100]$
Schwefel 2.22	单峰	$[-10, 10]$
Rosenbrock	单峰	$[-30, 30]$
Ackley	多峰	$[-32, 32]$
Griewank	多峰	$[-600, 600]$
Rastrigin	多峰	$[-5.12, 5.12]$
Schwefel2.26	多峰	$[-500, 500]$
Penalized	多峰	$[-50, 50]$

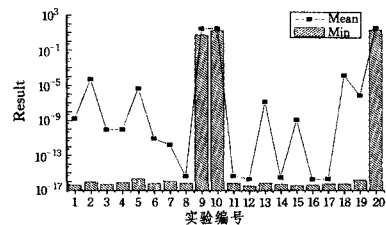


图 1 AGSO 均匀设计实验结果

从图 1 中可以看出,第 12 组和第 16 组实验的优化结果比其他方案要好。综合两组实验的参数设置,最终 AGSO 算法的参数设置如下: $\rho=0.8, \gamma=0.05, l_0=0.5, \beta=0.65, nt=95, r_s=0.5 \times \|ub-lb\|$ 。对于 PGSO,在上述参数设置的基础上,对其余的 3 个参数 ω, c_1, c_2 进行讨论,计算结果如图 2 所示。从图 2 中的数据可以看出,第 6 组方案的优化结果要优于其他方案,因此 3 个参数的取值如下: $\omega=0.45, c_1=0.3, c_2=0.15$ 。

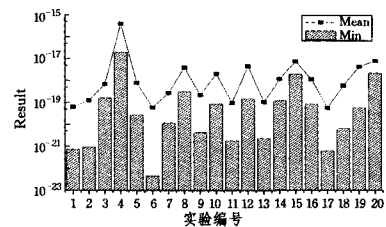


图 2 PGSO 均匀设计实验结果

依据上述参数设置,对 GSO 算法及其改进算法进行比较。计算结果如表 2 和表 3 所列,加黑字体表示两种算法中优秀的结果。表 2 是基本 GSO 和基于蜂群行为改进的 AGSO 的比较,从表中数据可以看出,对于所有函数,AGSO 算法的优化结果要远高于基本 GSO 算法,其最少提升了一个数量级(对于 Schwefel2.26,提升了 10^{-1}),最多提升了 24 个量级(对于 Penalized,精度从 10^7 降低到 10^{-17})。AGSO 在 Step、Rastrigin 和 Schwefel2.22 几个函数上都搜索到了全局最优值。AGSO 算法的性能大幅提升,一方面依赖于算法参数的设置,另一方面主要是蜂群搜索行为的引入、逐维位置改变和更优接受策略,可以使算法进行更精细的搜索。表 3 是两种改进算法的比较,从表中数据可以看出,除 Step、Rosenbrock 和 Schwefel2.22 3 个函数外,PGSO 的优化结果要优于 AG-

SO。这主要是由于 PGSO 中模拟的粒子群行为引入了最优个体,使萤火虫在搜索过程中,不仅受到局部优秀个体的引导,而且可以共享全局最优个体的信息。图 3 和图 4 是两种算法在函数中进化曲线的比较,从图中可以看出,由于最优个体的引入,使 PGSO 算法具有更快的收敛速度和更高的收敛精度。

表 2 GSO 与 AGSO 的比较($D=10$)

	GSO		AGSO	
	Mean	Min	Mean	Min
Sphere	3.37E+01	2.16E+01	2.55E-16	9.17E-17
Rosenbrock	2.55E+07	5.67E+06	3.80E+00	1.99E-05
Griewank	1.14E+02	4.65E+01	3.89E-16	1.11E-16
Rastrigin	1.05E+02	8.35E+01	1.36E+00	0.00E+00
Schwefel2.26	2.66E+03	2.00E+03	3.43E+02	1.27E-04
Ackley	1.94E+01	1.84E+01	6.46E-14	2.96E-15
Step	1.22E+04	5.21E+03	0.00E+00	0.00E+00
Quartic	4.23E+00	2.44E+00	3.38E-03	1.08E-03
Schwefel2.22	2.35E+01	1.70E+01	0.00E+00	0.00E+00
Schwefel1.2	4.03E+03	1.42E+03	2.54E-05	5.57E-16
Dixon-Price	4.47E+04	6.18E+03	9.36E-08	3.00E-16
Penalized	1.17E+08	1.51E+07	2.34E-16	6.51E-17

表 3 AGSO 与 PGSO 的比较($D=10$)

	AGSO		PGSO	
	Mean	Min	Mean	Min
Sphere	2.55E-16	9.17E-17	4.84E-20	8.19E-23
Rosenbrock	3.80E+00	1.99E-05	5.35E+00	2.02E-04
Griewank	3.89E-16	1.11E-16	0.00E+00	0.00E+00
Rastrigin	1.36E+00	0.00E+00	0.00E+00	0.00E+00
Schwefel2.26	3.43E+02	1.27E-04	1.79E+02	1.27E-04
Ackley	6.46E-14	2.96E-15	-5.89E-16	-5.89E-16
Step	0.00E+00	0.00E+00	0.00E+00	0.00E+00
Quartic	3.38E-03	1.08E-03	3.82E-05	7.52E-07
Schwefel2.22	0.00E+00	0.00E+00	0.00E+00	0.00E+00
Schwefel1.2	2.54E-05	5.57E-16	6.32E-20	3.84E-22
Dixon-Price	9.36E-08	3.00E-16	5.82E-22	1.58E-24
Penalized2	2.34E-16	6.51E-17	1.91E-16	5.93E-17

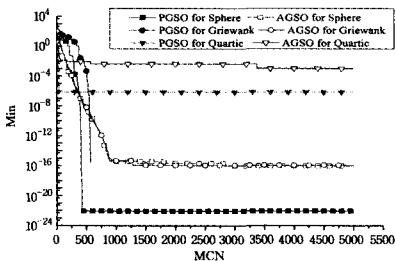


图 3 算法进化曲线比较

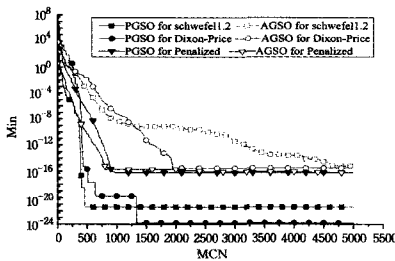


图 4 算法进化曲线比较

表 4 给出了 PGSO 算法与其他算法优化结果平均值的比较。遗传算法(GA)、粒子群算法(PSO)和差分进化算法(DE)的优化结果由文献[9]给出。在文献中, $D=30$ 。当优化

(下转第 228 页)

2)实事求是地讲,Dijkstra 算法的时间复杂度为 $O(n^2m)$,并不是太高。改进的 Dijkstra 算法,在时间复杂度方面,并没有明显改进。

所以,下一步的工作,将着重从如下方面入手:

1)对改进的 Dijkstra 算法进行严格证明,确保对阶数较大的图有效。

2)对 Dijkstra 算法继续改进,改进 Dijkstra 算法的时间复杂度,提高 Dijkstra 的运行效率。这将是具有理论意义的。

3)Dijkstra 算法应用非常广泛。下一步的工作,要对 Dijkstra 标号法的应用进行深入研究。这将是具有使用意义的。

参 考 文 献

[1] Dijkstra E. A note two problems in connection with graphs[J]. Numerical Mathemat,1959,1:269-271
[2] 刘代波,侯孟书,武泽旭,等. 一种高效的最短路径树动态更新算法[J]. 计算机科学,2011,38(7):96-99
[3] 廖巍,吴晓平,严承华,等. 一种新的道路网络连续查询处理方法

[J]. 计算机科学,2009,36(9):151-153
[4] 廖巍,熊伟,王钧,等. 可伸缩的增量连续 k 近邻查询处理[J]. 软件学报,2007,18(2):268-278
[5] 庄毅,庄越挺,吴飞. 基于数据网格的书法字 k 近邻查询[J]. 软件学报,2006,17(12):2289-2301
[6] 孙亚. 空间网络数据库中最近邻查询的设计与实现[J]. 计算机科学,2008,35(3):73-75
[7] 陈继东,孟小峰,赖彩凤. 基于道路网络的对象聚类[J]. 软件学报,2007,18(2):332-344
[8] 林澜,闫春钢,蒋昌俊,等. 动态网络最短路问题的复杂性与近似算法[J]. 计算机学报,2007,30(4):608-614
[9] 张涛,柳重堪,张军. 卫星时变拓扑网络最短路径算法研究[J]. 计算机学报,2006,29(3):371-377
[10] 刘健,杨文宇,余健明,等. 一种基于改进最小生成树算法的配电网架优化规划[J]. 中国电机工程学报,2004,24(10):103-108
[11] 于海彦,郝忠孝. 时空数据库中基于 TPR-树的反向最近邻查询[J]. 哈尔滨理工大学学报,2007,12(3):87-90
[12] 冯惠妍,郭俊凤. 道路网络中的连续最近邻查询[J]. 计算机工程,2010,36(8):79-82

(上接第 200 页)

结果精度低于 10^{-12} ,则认为结果为 0。从表 4 中数据可以看出,PGSO 在 7 个函数中的结果优于其它 3 种算法,在 9 个函数中搜索到了全局最优值。综上所述,经过大量的实验对比分析,结果表明本文提出的改进 GSO 算法是求解高维全局优化问题的有效方法。

表 4 PGSO 与其他算法的比较(D=30)

	GA ^[3]	PSO ^[3]	DE ^[3]	PGSO
Sphere	1.11E+03	0.00E+00	0.00E+00	0.00E+00
Rosenbrock	1.96E+05	1.51E+01	1.82E+01	2.12E-04
Griewank	1.06E+01	1.74E-02	1.48E-03	0.00E+00
Rastrigin	5.29E+01	4.40E+01	1.17E+01	0.00E+00
Schwefel2.26	9.76E+02	5.56E+03	2.30E+03	3.82E-04
Ackley	1.47E+01	1.65E-01	0.00E+00	0.00E+00
Step	1.17E+03	0.00E+00	0.00E+00	0.00E+00
Quartic	1.81E-01	1.16E-03	1.36E-03	9.06E-05
Schwefel2.22	1.10E+01	0.00E+00	0.00E+00	0.00E+00
Schwefel1.2	7.40E+03	0.00E+00	0.00E+00	0.00E+00
Dixon-Price	1.22E+03	6.67E-01	6.67E-01	0.00E+00
Penalized	1.25E+02	7.68E-03	2.20E-03	0.00E+00

结束语 基于群智能理论的优化方法研究方兴未艾。萤火虫群优化算法作为新型的优化方法,在算法和应用方面的研究刚刚起步。本文针对 GSO 算法中存在的收敛精度低的问题,基于群智能理论,借鉴人工蜂群算法和粒子群算法中个体生成机制,对 GSO 算法的移动策略和移动方式进行了改进,提出了更优接受策略和逐维更新机制。通过对标准测试函数进行实验仿真,并与基本 GSO 算法和其他算法比较,结果表明本文提出的改进算法能大幅提高 GSO 算法的收敛精度,是求解高维全局优化问题的有效方法。

参 考 文 献

[1] Krishnanand K N,Ghose D. Detection of multiple source locations using a glowworm metaphor with applications to collective

robotics [C]//IEEE Swarm Intelligence Symposium. Pasadena, California:IEEE Press,2005:84-91
[2] Krishnanand K N,Ghose D. Theoretical foundations for rendezvous of glowworm-inspired agent swarms at multiple locations [J]. Robotics and Autonomous Systems,2008,56(7):549-569
[3] Krishnanand K N,Ghose D. Glowworm swarm optimization for simultaneous capture of multiple local optima of multimodal functions[J]. Swarm Intelligence,2009,3(2):87-124
[4] Krishnanand K N,Ghose D. Glowworm swarm based optimization algorithm for multimodal functions with collective robotics applications[J]. Multi-agent and Grid Systems,2006,2(3):209-222
[5] Kaipa K N,Puttappa A,et al. Rendezvous of glowworm-inspired robot swarms at multiple source locations:A sound source based real-robot implementation[C]//Lecture Notes in Computer Science. 2006,4150:259-269
[6] Bharat T V. Agents based algorithms for design parameter estimation in contaminant transport inverse problems [C]// 2008 IEEE Swarm Intelligence Symposium (SIS 2008). St. Louis, MO, United States,2008:1-7
[7] He D, Zhu H. Glowworm swarm optimization algorithm based on multi-population [C]// Proceedings-2010 6th International Conference on Natural Computation(ICNC 2010). Yantai, Shandong, China, IEEE Computer Society,2010:2624-2627
[8] 李咏梅,周永权,姚祥光. 基于追尾行为的改进型人工萤火虫群算法[J]. 计算机科学,2011,38(3):248-251
[9] Karaboga D, Akay B. A comparative study of artificial bee colony algorithm [J]. Applied Mathematics and Computation, 2009, 214:108-132
[10] 方开泰. 均匀设计与均匀设计表[M]. 北京:科学出版社,1994:74-97