doi: 10.3969/j.issn.0490-6756.2017.04.010

基于动态多子族群自适应群居蜘蛛优化算法

刘洲洲[1], 李 彬[2]
(1. 西安航空学院, 西安 7100772; 2. 西北工业大学电子信息学院, 西安 710072)

摘 要: 为了提高群居蜘蛛优化算法(SSO)样本多样性和算法收敛性能,提出了一种基于动态多子族群自适应群居蜘蛛优化算法(DMASSO).根据算法样本多样性和算法进化程度,动态的将蜘蛛种群分成若干个主导子族群和辅助子族群,在不同子族群中分别引入自适应学习因子和高斯扰动因子改进算法个体更新方式,实现提高算法全局寻优能力和保持群体样本多样性.针对具有典型特点的测试函数仿真结果表明,较 SSO 算法、MSFLA 算法等优化算法相比,新算法在收敛速度和收敛精度上均有明显改善.

关键词: 群居蜘蛛优化算法; 多子族群; 自适应; 函数优化
中图分类号: TP273 **文献标识码**: A **文章编号**: 0490-6756(2017)04-0721-07

An adaptation social spider optimization algorithm based on dynamic multi-swarm strategy

LIU Zhou-Zhou[1], *LI Bin*[2]
(1. Xi'an Aeronautical University, Xi'an 710077, China;
2. School of Electronics and Information, Northwestern Polytechnical University, Xi'an 710072, China)

Abstract: In order to improve the samples diversity and convergence properties of social spiders optimization algorithm (SSO), an adaptation social spider optimization algorithm based on dynamic multi-swarm strategy (DMASSO) is proposed. According to the algorithm samples diversity and evolutionary level, the spider population is dynamically divided into different sizes leading groups and supporting groups, and the adaptive learning factor and Gaussian disturbance factor are introduced to improve the algorithm update ways, which helps to improve the algorithm global optimization ability and maintain the diversity of the sample population. For the test results of typical characteristics functions show that compared to SSO algorithm, SFLA algorithm and other optimization algorithms, the new algorithm has better convergence speed and convergence accuracy.

Keywords: Social spider optimization algorithm; Multi-swarm; Adaptation; Function optimization

1 引 言

群居蜘蛛优化算法(Social Spider Optimization Algorithm,SSO)是 Cuevas 等学者[1]通过模拟群居蜘蛛协作行为,提出的一种新型随机全局优化技术.在 SSO 算法中,雌性和雄性两类不同搜索个体按照不同的搜索准则协同进化,能够实现全局寻优[2].经典函数测试表明,与 PSO 算法、ABC 算法相比,SSO 具有良好的收敛性能和鲁棒性[3].由于 SSO 算法提出时间不长,与其相关的理论研究和实践应用还相对较少,特别是对于复杂多极值函数优化问题,提高算法收敛效果仍是 SSO 需要解

收稿日期: 2016-06-13
基金项目: 国家自然科学基金(61401499); 陕西省教育厅科研计划项目 (16JK1395)
作者简介: 刘洲洲(1981-), 男, 博士, 副教授, 研究方向为智能优化算法和无线传感器网络研究. E-mail: nazi2005@126.com

决的难题.

目前,已有部分学者开始对 SSO 算法改进及应用进行研究,王艳娇等[4]提出了一种基于动态学习策略的群集蜘蛛优化算法,采用云模型和随机交叉策略蜘蛛个体更新方式,提高了算法收敛精度;王文川[3]、吴光琼[5]分别将 SSO 算法应用于自适应数值积分皮尔逊-III 型曲线参数估计和水文频率曲线参数优化中,取得了不错效果. 同其它群智能优化算法一样,SSO 算法也存在收敛速度慢、容易陷入局部最优的缺陷[6],为此,本文提出了一种基于动态多子族群自适应群居蜘蛛优化算法(Dynamic Multi-swarm strategy Adaptation Social Spider Optimization Algorithm, DMASSO),根据种群样本多样性和算法进化程度,将蜘蛛种群分成若干个主导子族群和辅助子族群,并动态调整子族群规模,以平衡算法收敛速度和保持样本多样性. 为进一步提高算法全局搜索能力,在不同子族群中分别引入自适应学习因子和高斯扰动因子改进算法个体更新方式. 仿真结果表明,较 SSO 算法、SFLA 算法[7]等优化算法相比,DMASSO 算法在收敛速度和收敛精度上均有明显改善.

2 基本 SSO 算法描述

群居蜘蛛由雄雌性两类个体和蜘蛛网组成,个体之间相互协作并通过蜘蛛网传递信息. SSO 算法在模拟群居蜘蛛群体协作行为的基础上,将整个搜索空间等效为蜘蛛网络,每个蜘蛛位置即为目标优化问题的一个解,算法根据雄雌性个体不同的任务性质,赋予个体不同的迭代进化策略,并通过婚配等行为实现群体间信息交换,最终实现问题寻优目的. SSO 算法工作过程可以描述为:

Step1 参数初始化. 设定种群数目 N、雌性蜘蛛数量 N_f、雄性蜘蛛数量 N_m,设定算法有关参数:空间搜索维度 n、概率因子 PF、婚配半径 r. N_f 计算公式为:

$$\begin{cases} N_f = \lfloor (0.9 - \mathrm{rand} \times 0.25) \times N \rfloor \\ N_m = N - N_f \end{cases} \tag{1}$$

其中,rand 为[0,1]随机数;$\lfloor \cdot \rfloor$ 为向下取整数.

Step 2 种群初始化. 设蜘蛛种群 S 由雌性 $F = \{F_1, F_2, \cdots, F_{N_f}\}$ 和雄性 $M = \{M_1, M_2, \cdots, M_{N_m}\}$ 两个子群组成. 随机产生初始 F 和 M 子群.

$$\begin{cases} F_{ij} = F_{j\min} + \mathrm{rand}(F_{j\max} - F_{j\min}) \\ M_{kj} = M_{j\min} + \mathrm{rand}(M_{j\max} - M_{j\min}) \end{cases} \tag{2}$$

其中,F_{ij} 为雌性个体 $F_i(F_{i1}, F_{i2}, \cdots, F_{in})$ 第 j 为变量($j \in \{1,2,\cdots,n\}$);$F_{j\max}$、$F_{j\min}$ 分别为第 j 为变量的上下限. 婚配半径 r 由式(3)确定.

$$r = \frac{\sum_{j=1}^{n} [\max(F_{j\max}, \mathrm{M}_{j\max}) - \min(\mathrm{F}_{j\min}, \mathrm{M}_{j\min})]}{2n} \tag{3}$$

Step 3 雌性子群协作进化. SSO 算法模拟雌性个体通过震动吸引或排斥其它个体行为,设计了两种震动更新模式,即

$$F_i^{T+1} = \begin{cases} F_i^T + \alpha Vib_{c_i}(S_c - F_i^T) + \\ \beta Vib_{b_i}(S_b - F_i^T) + \delta(\mathrm{rand} - 0.5); \mathrm{rand} \leqslant PF \\ F_i^T - \alpha Vib_{c_i}(S_c - F_i^T) - \\ \beta Vib_{b_i}(S_b - F_i^T) + \delta(\mathrm{rand} - 0.5); \mathrm{else} \end{cases} \tag{4}$$

其中,α、β、δ 为[0,1]之间随机数;S_c、S_b 分别为距离 F_i 最近且高于其权重的个体 F_c 和整个雌性子群中权重最高的个体 F_b. Vib_{v_l} 表示个体 F_l 对个体 F_v 的震动感知能力. 个体权重值 ω_i 和震动感知能力 Vib_{v_l} 计算公式如下(以最小值优化问题为例).

$$\omega_i = \frac{J(S_i) - worst_i}{best_i - worst_i} \tag{5}$$

$$Vib_{v_l} = \omega_l e^{-d_{vl}^2}, (b_{vl} = \| S_v - S_l \|) \tag{6}$$

$$best_i = \min_{i \in \{1,2,\cdots,N\}} \{J(S_i)\}$$

$$worst_i = \max_{i \in \{1,2,\cdots,N\}} \{J(S_i)\} \tag{7}$$

其中,$J(S_i)$ 为个体 F_l 目标函数值.

Step 4 雄性子群协作进化. SSO 算法将雄性个体按权重值降序排列,并取中间权重 $\omega_{N_{f+m}}$ 为参考值,分别赋予大于或小于该权重个体不同的震动更新模式.

Step 5 婚配行为. SSO 算法规定权重值大于 $\omega_{N_{f+m}}$ 的雄性个体 M_i 能够与其婚配半径内的雌性个体发生婚配行为.

3 基于动态多子族群自适应群居蜘蛛优化算法

3.1 动态多子族群策略

同其它群智能优化算法一样,随着进化迭代次数的不断增加,种群样本多样性逐渐降低,算法一旦陷入局部极值,就很难跳出,出现“早熟”[8]现象. 为此,本文提出一种动态多子族群策略(如图 1 所示),将雌性子群和雄性子群分成若干个规模相同

的子族群，并根据算法样本多样性和算法进化程度动态调整子族群规模.

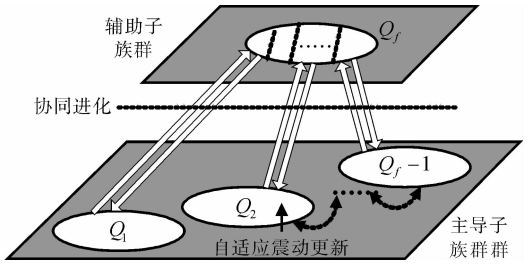

图 1　动态多子族群策略

Fig. 1　Dynamic multi population strategy

以雌性子群为例，子族群划分规则为

Step 1　降序排列. 将雌性子群 $F=\{F_1,F_2,\cdots,F_{N_f}\}$ 内个体按权重值降序排列，并划分成 Q_f 个规模相同的子族群，每个子族群内蜘蛛数量为 $q_f(N_f=Q_f\times q_f)$.

Step 2　“辅助子族群”和“主导子族群”划分. 取前 q_f 个权重值较优个体组成为一个辅助子族群 Au_f，其他个体随机分配到 Q_f-1 个主导子族群 $Le_{f,k}(k=1,2,\cdots,Q_f-1)$ 中.

Step 3　“次辅助子族群”划分. 为了保证 $Le_{f,k}$ 和 Au_f 间信息交流，再次将辅助子族群划分成 Q_f-1 个“次辅助子族群” $Au'_{f,k}$ $(k=1,2,\cdots,Q_f-1)$. 每个次辅助子族群 $Au'_{f,k}$ 与 $Le_{f,k}(k=1,2,\cdots,Q_f-1)$ 建立一一对应关系.

Step 4　信息协作交流. $Au'_{f,k}$ 和 $Le_{f,k}$ 采用不同的个体更新方式，当其完成内部个体迭代进化后，取 $Au'_{f,k}$ 内权重最优个体 $S_{best}^{Au,k}$ 与其对应的 $Le_{f,k}$ 内权重最优个体 $S_{best}^{Le,k}$ 进行比较，若 $S_{best}^{Au,k}$ 权重值优于 $S_{best}^{Le,k}$，则用 $S_{best}^{Au,k}$ 替代 $S_{best}^{Le,k}$. 当所有 $Le_{f,k}$ 完成更新后，所有雌性个体重新组合，并转到 Step1.

对于子族群规模数 Q_f，其计算公式为(以最大值优化问题为例).

$$Q'_f=Q_{\min}+\kappa\cdot(Q_{\max}-Q_{\min})\left[e^{\left(\min\limits_{i\in\{1,2,\cdots,N_f\}}\{J(F_i)\}\right)/\left(\max\limits_{i\in\{1,2,\cdots,N_f\}}\{J(F_i)\}\right)\cdot\ln 2}-1\right] \tag{8}$$

$$Q_f=\min\left(N_f/\left|\frac{N_f}{Q'_f}\right|,Q_{\max}\right) \tag{9}$$

其中，$Q_{\max}$、$Q_{\min}$ 分别为最大和最小子族群数，为子族群规模控制系数. 从式(8)可以看出，在算法初期，由于样本差异性较大，如果子族群数量较多，将会导致蜘蛛婚配半径范围内个体数量极少，反而不利于算法迭代进化；随着迭代次数不断增加，个体之间差异逐渐变小，此时算法将雌性子群划分为较多数量的子族群，有利于增加样本多样性，从而提高了算法全局寻优能力.

3.2　自适应个体震动更新模式

平衡算法收敛速度和深度挖掘能力是智能优化算法需要研究解决的问题之一[9]，研究表明合理设置算法个体迭代进化方式，能够有效提高算法收敛性能. 对于 SSO 算法，蜘蛛个体学习对象设定为种群最优个体和距离较近个体，但是在算法初期，个体之间距离较大，使得蜘蛛震动感知能力极小，导致无法获取其他优秀个体信息；算法进化后期，种群样本差异性很低，蜘蛛个体从其他优秀获取得来的信息与自身差距不大，降低了算法深度搜索能力. 为此，本文提出了一种自适应个体震动更新模式，在不同主导子族群和辅助子族群中分别引入自适应学习因子和高斯扰动因子，其具体工作过程可以描述为(以雌性子群和最小化问题为例).

(1) 主导子族群. 在主导子族群中，蜘蛛个体进化更新方式为

$$F_i^{T+1}=\begin{cases}\eta_i F_i^T+\alpha\cdot\mu_c\cdot(S_c-F_i^T)+\\ \beta\cdot\mu_b\cdot(S_b-F_i^T)+\delta(\text{rand}-0.5); \\ \quad \text{rand}\leqslant PF\\ \eta_i F_i^T-\alpha\cdot\mu_c\cdot(S_c-F_i^T)-\\ \beta\cdot\mu_b\cdot(S_b-F_i^T)+\delta(\text{rand}-0.5);\text{else}\end{cases} \tag{10}$$

$$\eta_i=\eta_{\min}+(\eta_{\max}-\eta_{\min})\left[(e^{\ln 2\cdot T/T_{\max}}-1)\omega_0+\frac{J(F_i)-J_{\min}(\cdot)}{J_{\max}(\cdot)-J_{\min}(\cdot)}(1-\omega_0)\right] \tag{11}$$

$$\mu_j=\mu_{\min}+(\mu_{\max}-\mu_{\min})\left[(2-e^{\ln 2\cdot T/T_{\max}})\omega_0+\frac{J(F_j)-J_{\min}(\cdot)}{J_{\max}(\cdot)-J_{\min}(\cdot)}(1-\omega_0)\right] \tag{12}$$

其中，η_i、μ_j 为自适应学习因子，$T_{\max}$ 为最大迭代次数. 从式(11)和(12)可以看出，随着迭代次数增加，个体向自身学习的趋势不断增强，向优秀个体学习的趋势不断减弱，有效平衡了算法收敛速度和

种群样本多样性.

(2) 辅助子族群. 在辅助子族群中,蜘蛛个体进化更新方式为

$$F_i^{T+1}=F_i^T+F_i^T\cdot\frac{1}{\sigma\sqrt{2\pi}}\exp\left[-\frac{T^2}{2\sigma^2}\right] \tag{13}$$

$$\sigma=\sigma_{\max}-(\sigma_{\max}-\sigma_{\min})\frac{T}{T_{\max}} \tag{14}$$

其中,σ 为高斯扰动因子. 从式(14)可以看出,随着进化迭代次数不断增加,σ 逐渐减小,使得蜘蛛个体能够在小范围学习空间内进行深度搜索,有利于提高算法收敛精度.

3.3 算法收敛性分析

DMASSO 算法本质上属于随机收索算法范畴,文献[9]给出了一般随机优化算法收敛性判断准则,本文在此基础上讨论 DMASSO 算法的全局收敛性.

对于优化问题 $<S,F>$(S 为可行解空间,F 为目标函数),有随机算法 G,其第 k 次迭代结果为 X_k,第 $k+1$ 次迭代结果为 $X_{k+1}=G(X_k,\delta)$(δ 为算法 G 搜索过的解).

条件 1 $F(G(X,\delta))\leqslant F(X)$,并且如果 $\delta\in S$,则有 $F(G(X,\delta))\leqslant F(\delta)$.

条件 2 若 $\forall D\in S$ 有 $\nu(D)>0$,且有 $\prod_{k=0}^{\infty}(1-\mu_k(D))=0$. 其中,$\nu(D)$ 为 D 的勒贝格测度,$\mu_k(D)$ 为算法 G 第 k 次迭代解的概率测度.

定理 1 设 F 可测,可测空间 S 是 R^n 的可测度子集,若算法满足条件 1 和条件 2,则有 $\lim_{k\to\infty}p(X_k\in S^*)=1$,即算法以概率 1 收敛于全局最优. 其中,$S^*$ 为全局最优点集合.

定理 2 DMASSO 算法以概率 1 收敛于全局最优解.

证明 根据 3.3 节的描述,有 $F[X_i(t+1)]\leqslant F[X_i(t)]$,因此满足判定定理 1 中的条件 1. 设 M_i^t 为蜘蛛个体 i 在第 t 次迭代过程中搜索解支撑集,随着迭代次数增加,$X_i(t)\underset{t\to\infty}{\to}X_g(t)$,因此 $\nu(M_i^t)$ 逐渐变小,使得 $\nu(\bigcup_{i=1}^{N}M_i^t\cap P)<\nu(P)$,即不满足判定定理 1 中的条件 2,但是不能产生更优解时,算法产生随机解替代,随机替代的方式使得 $M_i^t=P$,因此有 $\bigcup_{i=1}^{N}M_i^t\supseteq P$,即 $\forall D\in S$,有 $0<\sum_{i=1}^{N}\mu_k(D)\leqslant 1$,并且有 $\prod_{k=0}^{\infty}(1-\mu_k(D))\to\lim_{k\to\infty}(1-\sum_{i=1}^{N}\mu_k(D))^k=0$,所以此时满足条件 2,即 IDMASSO 以概率 1 收敛于全局最优解,证毕.

3.4 DMASSO 算法实现流程

基于动态多子族群自适应群居蜘蛛优化算法(DMASSO)实现流程如图 2 所示.

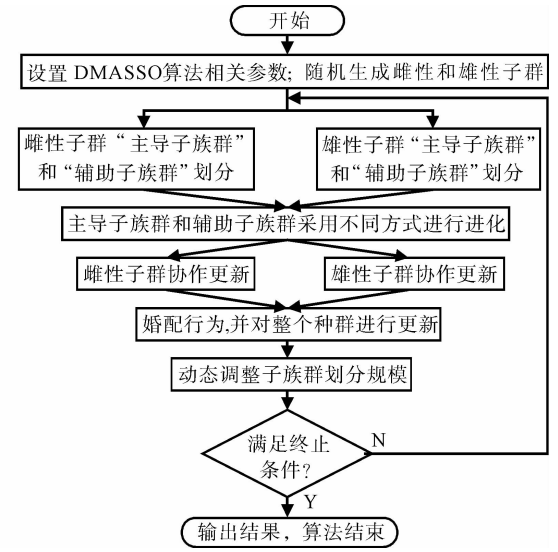

图 2 DMASSO 算法流程图

Fig. 2 DMASSO algorithm flow chart

4 实验仿真

为了验证 DMASSO 算法优化性能,采用 6 个具有典型特点的测试函数[10,11]进行实验仿真. 表 1 给出了具体测试函数. DMASSO 参数设置如下:$N_f=200$,$Q_f=20$,$q=10$,$K_{\max}=10$,$T_{\max}=500$,$r_{\max}=0.8$,$r_{\min}=0.2$,$\varepsilon=10^{-4}$,$\tau=0.6$. 分别采用 DSSO、PDACO[13]、文献[3]算法和 DMASSO 对 TSPLIB 中的 bayg29、eil51 和 ch150 对进行测试. 仿真测试环境设置及算法参数设置参考第三章相关设置. 每种算法独立重复运行 50 次,取平均距离 $\overline{S}$、平均运算时间 $\bar{t}$ 和相对误差 Err 等指标进行对比,其中,Err 计算公式为

$$Err=(\overline{S}-Opt)/Opt \tag{15}$$

Opt 为已知全局最优环路长度(bayg29、eil51 和 ch150 理论最优路线长度为 9074.1,428.87 和 6528),表 1 给出了具体实验结果数据.

在算法求解精度方面,DSSO、文献[3]算法和 DMASSO 都能够给出较好的行进路线,且 DMASSO 的求解精度明显优于其他两种算法,而 PDACO 无法得到最优解. 在算法运行速度方面,对于 bayg29、eil51 和 ch150 测试问题,DMASSO 运算速度都要快于其他三种算法.

4.1 二维连续多峰函数仿真结果及分析

函数 f_1、f_2 是经典的二维连续多峰函数,其中

f_1 最优解为 $f_1(5.9776,5.9776)=-150.2514$，$f_2$ 最优值为 $f_2(0,0)=1$（实验取最小值即-1）. 图 3 和图 4 给出了两个函数的收敛曲线.

表 1　TSPLIB 不同算法测试结果

Tab. 1　TSPLIB test results of different algorithms

		bayg29	eil51	ch150
DSSO	$\overline{S}$	16513.7	1227.6	46210.3
	$\bar{t}$	6.11	12.71	27.16
	Err (%)	81.9	186.2	607.9
PDACO	$\overline{S}$	10231.1	511.6	7001.2
	$\bar{t}$	7.13	14.18	30.22
	Err (%)	12.8	19.4	7.2
文献[3]	$\overline{S}$	9978.4	477.8	6892.1
	$\bar{t}$	5.26	10.17	25.39
	Err (%)	9.9	11.4	5.6
DMASSO	$\overline{S}$	9978.4	477.8	6892.1
	$\bar{t}$	4.21	7.32	12.17
	Err (%)	0.4	0.7	0.8

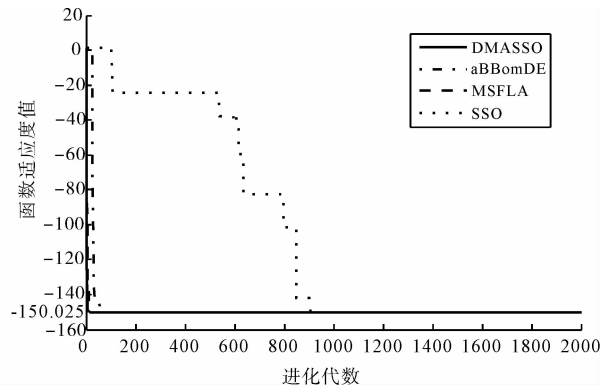

图 3　函数 f_1 收敛曲线

Fig. 3　Function 1 convergence curve

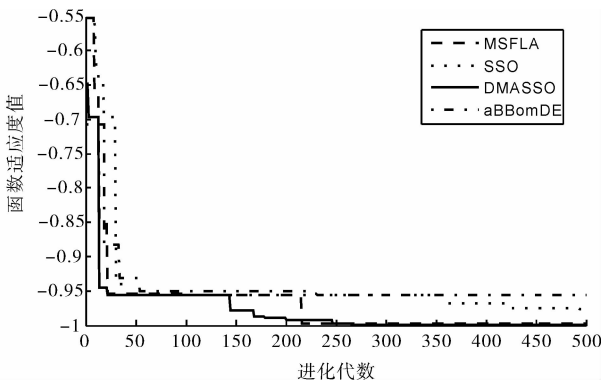

图 4　函数 f_2 收敛曲线

Fig. 4　Function 2 convergence curve

从图 3 可以看出，对于函数 f_1，4 种算法都能够最终找到全局最优解，但是 SSO 需要迭代接近 100 次才能找到最优解. 从图 4 可以看出，对于函数 f_2，DMASSO、MSFLA 和 aBBomDE 能够找到全局最优解，而 SSO 算法无法收敛于全局最优，而且 DMASSO 收敛速度明显优于其它 3 种算法.

4.2　高维多峰大量局部极值函数仿真结果及分析

函数 f_3、f_4、f_5 为高维多峰大量局部极值函数，且都在 $(0,0,\cdots,0)$ 取得全局最优值 0. 图 5～图 7 给出了 3 个函数收敛曲线，表 2 给出了相关评价指标. （设 f_3、f_4 和 f_5 收敛精度依次为 10^{-4}、10^{-2} 和 10^{-3}）.

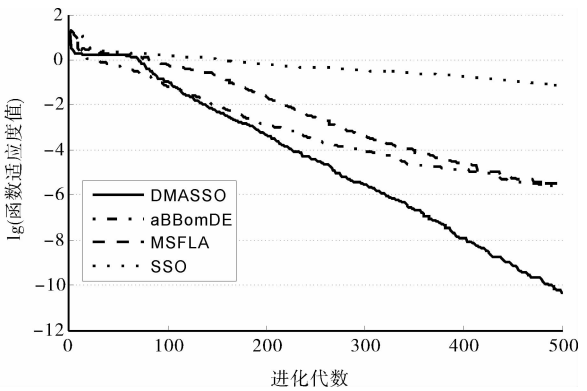

图 5　函数 f_3 收敛曲线

Fig. 5　Function 3 convergence curve

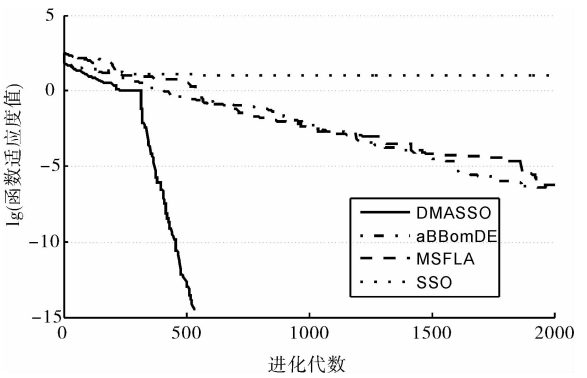

图 6　函数 f_4 收敛曲线

Fig. 6　Function 4 convergence curve

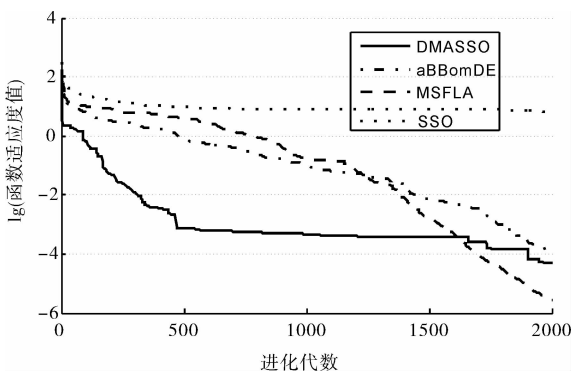

图 7　函数 f_5 收敛曲线

Fig. 7　Function 5 convergence curve

表 2 高维多峰大量局部极值函数实验数据

Tab. 2 High dimensional multi peak large number of local extreme value function experimental data

函数	算法	成功数	最大值	最小值	平均值	时间(s)
f_3	SSO	0	0.14922	0.0993	0.2247	1.441
	DMASSO	30	7.14e-15	5.66e-017	6.21e-16	2.668
	MSFLA	30	2.38e-5	3.66e-7	2.96e-6	3.361
	aBBomDE	30	1.02e-5	4.21e-7	2.77e-6	3.427
f_4	SSO	0	19.2136	2.0015	8.9937	4.123
	DMASSO	30	2.32e-14	0.0000	0.99e-16	5.779
	MSFLA	30	1.32e-5	4.07e-6	5.88e-5	8.262
	aBBomDE	30	0.78e-4	8.35e-7	5.31e-6	7.113
f_5	SSO	0	20.541	1.883	10.116	18.15
	DMASSO	30	1.97e-4	2.17e-5	8.45e-5	17.88
	MSFLA	30	0.554e-3	3.321e-4	9.887e-4	25.41
	aBBomDE	30	2.12e-4	0.33e-5	6.94e-5	28.32

从图 5～图 7 及表 2 可以看出,对于 f_3、f_4、f_5,SSO 没有找到全局最优解;其它三种算法都能收敛于全局最优(在设定的收敛精度范围内),并且 DMASSO 算法无论在收敛精度还是收敛速度上都要优于 MSFLA 算法和 aBBomDE 算法.

4.3 高维病态函数仿真结果及分析

函数 f_6 是典型高维病态函数,在 $(0,0,\cdots,0)$ 取得全局最优值 0.图 8 给出了 f_6 函数收敛曲线,表 3 给出了相关评价指标.(设定 f_6 收敛精度 1).

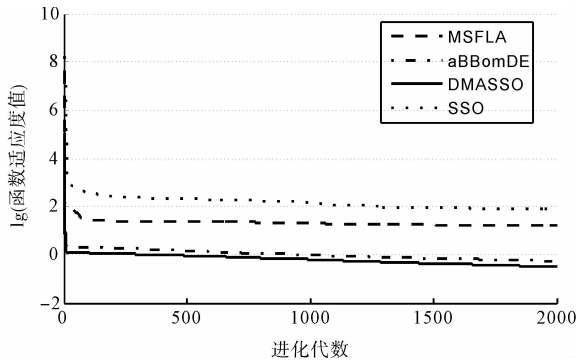

图 8 函数 f_6 收敛曲线

Fig. 8 Function 6 convergence curve

表 3 函数 f_6 实验数据

Tab. 3 Function 6 experimental data

函数	算法	成功数	最大值	最小值	平均值	时间(s)
f_3	SSO	0	66.5423	19.4326	42.1185	9.714
	MSFLA	0	14.1128	8.3312	10.9831	28.40
	aBBomDE	27	1.6521	0.0975	0.6628	19.33
	DMASSO	30	0.4328	0.0845	0.2254	8.52

从图 8 和表 3 可以看出,对于函数 f_6,在 30 次试验中 DMASSO 算法能够全部寻找到最优解(在设定的收敛精度范围内),aBBomDE 算法也有 27 次寻找到最优解,并且 DMASSO 收敛速度要快于 aBBomDE 算法,而 SSO 和 MASFLA 算法则无法得到最优解.

从上述 3 类测试函数仿真结果可以看出,DMASSO 算法无论在收敛精度和收敛速度上都具有一定优势,特别是对于复杂优化函数,DMASSO 算法表现出了良好性能,这也说明基于动态多子族群自适应群居蜘蛛优化算法能够有效平衡算法收敛速度和保持样本多样性,具有较为出色的全局寻优能力.

5 结 论

本文提出了一种基于动态多子族群自适应群居蜘蛛优化算法(DMASSO),根据种群样本多样性和算法进化程度,将蜘蛛种群分成若干个主导子族群和辅助子族群,并动态调整子族群规模,有效平衡了算法收敛速度和保持样本多样性;在不同子族群中分别引入了自适应学习因子和高斯扰动因子以改进算法个体更新方式,提高了算法全局收敛性能.对 6 个经典测试函数进行仿真,结果表明 DMASSO 算法在收敛速度和收敛精度具有较好表现.

参考文献:

[1] Cuevas E, Cienfuegos M, Zaldiva D, *et al*. A swarm optimization algorithm inspired in the behavior of the social spider [J]. Exp Sys Appl, 2013, 40: 6374.

[2] Erik C, Miguel Cienfuegos. A new algorithm inspired in the behavior of social-spider for constrained optimization [J]. Exp Sys Appl, 2014, 41: 412.

[3] 王文川, 雷冠军, 刘惠敏, 等. 基于群居蜘蛛优化算法的自适应数值积分皮尔逊-III 型曲线参数估计[J]. 应用基础与工程科学学报, 2015, 23(suppl): 122.

[4] 王艳娇, 李晓杰, 肖婧. 基于动态学习策略的群集蜘蛛优化算法[J]. 控制与决策, 2015, 30: 1575.

[5] 江马群, 陈渝, 王小玲, 等. 基于马尔可夫链的主机负载预测能耗优化算法[J]. 四川大学学报: 自然科学版, 2014, 51: 701.

[6] 吴建辉, 章兢, 李仁发, 等. 多子种群微粒群免疫算法及其在函数优化中应用[J]. 计算机研究与发展, 2012, 49: 1883.

[7] EUSUFF M M, LANSEY K E. Optimization of water distribution network design using the shuffled frog leaping algorithm [J]. Water Res Pl-asce, 2003, 129: 210.

[8] 骆剑平，李霞，陈泯融. 混合蛙跳算法的 Markov 模型及其收敛性分析[J]. 电子学报，2010，38：2875.

[9] 谢安世，于永达，黄思明. 一种基于标杆管理的优化算法[J]. 软件学报，2014，25：953.

[10] 张军丽，周永权. 一种用 Powell 方法局部优化的人工萤火虫算法[J]. 模式识别与人工智能，2011，24：680.

[11] 张强，李盼池. 量子混合蛙跳算法求解连续空间优化问题[J]. 吉林大学学报：理学版，2013，51：471.

[12] Elbeltagi E, Hegazy T, Grierson D. A modified shuffled frog-leaping optimization algorithm applications to project management [J]. Struct Infrastruct, 2007, 3: 53.

[13] Lohokare M R. Pattnaik S S, Pranigrahi B K, *et al*. Accelerated biogeography-based optimization with neighborhood search for optimization [J]. Appl Soft Comput, 2013, 13: 2318.