

文章编号:1001-9081(2010)10-2578-04

基于自适应动态邻居和广义学习的改进粒子群算法

刘衍民^{1,2}, 赵庆祯², 牛奔³

(1. 遵义师范学院 数学系, 贵州 遵义 56300; 2. 山东师范大学 管理与经济学院, 济南 250014;

3. 深圳大学 管理学院, 广东 深圳 518060)

(yanmin7813@163.com)

摘要:为了克服粒子群算法在求解多峰函数时极易陷入局部最优解的缺陷,提出一种基于自适应动态邻居广义学习的改进粒子群算法(ADPSO)。在ADPSO算法中,根据每个粒子邻居中最好运行粒子的状态动态地调整邻居拓扑结构;每个粒子的学习样本包括全局最优粒子、自身最优粒子和粒子邻居中最优运行粒子;并且在新产生的粒子位置上,加上一个随机位置以增加粒子向全局最优解移动的概率。在基准函数的测试中,结果显示ADPSO算法比其他PSO算法有更好的运行效果,是求解多峰问题的一种有效算法。

关键词:自适应;粒子群算法;动态邻居;广义学习;多峰函数

中图分类号: TP18 **文献标志码:** A

Improved particle swarm optimization based on adaptive dynamic neighborhood and generalized learning

LIU Yan-min^{1,2}, ZHAO Qing-zhen², NIU Ben³

(1. Department of Mathematics, Zunyi Normal College, Zunyi Guizhou 563002, China

2. School of Management and Economics, Shandong Normal University, Jinan Shandong 250014, China;

3. College of Management, Shenzhen University, Shenzhen Guangdong 518060, China)

Abstract: As Particle Swarm Optimization (PSO) may easily get trapped in a local optimum, an improved PSO based on adaptive dynamic neighborhood and comprehensive learning named ADPSO was proposed. In ADPSO, the neighbors of each particle were dynamically constructed in terms of the best performing particle among the current particle neighborhood. Then the learning mechanism of each particle was separated into three parts: its own historical best position, the best neighbor and the global best one. At the position of the new particle, a random position around itself was added to increase the probability for the particle to move to that promising region. The test results on benchmark functions show that ADPSO achieves better solutions than other improved PSO, and it is an effective algorithm to solve multi-objective problems.

Key words: adaptive; Particle Swarm Optimization (PSO); dynamic neighbor; comprehensive learning; multimodal function

0 引言

粒子群优化算法(Particle Swarm Optimization, PSO)^[1]起源于对简单社会系统(鸟群觅食的过程)的模拟,自提出以来出现了大量的研究成果,在函数优化、神经网络训练、模糊系统控制等领域得到了广泛的应用。但是,与其他进化算法一样,PSO算法极易陷入局部最优解,在进化后期收敛速度慢,优化精度较差。针对这些问题,许多学者从不同方面对其进行了改进:文献[2-3]提出了一种动态邻居拓扑结构粒子群算法,通过调整邻居的拓扑结构来提升粒子的搜索范围;Liang等人^[4]提出了一种广义学习粒子群算法,介绍了一种新的粒子学习机制以提升粒子跳出局部最优解的能力,其实质是以一种动态邻居拓扑结构来进行粒子的学习样本的选择。本文通过对粒子群优化算法的收敛特性及种群多样性的分析,在文献[5]提出的改进PSO算法基础上,提出一种自适应动态邻居拓扑结构PSO算法(Adaptive Dynamic PSO, ADPSO),该算法根据粒子的运行自适应地调整邻居拓扑结构,并且每个粒子的飞行方向由全局最优粒子、自身最优粒子

和粒子邻居中最优运行粒子来共同调整;在新产生的粒子位置上,加上一个随机位置以增加粒子向全局最优位置移动的概率。这些策略提升了粒子跳出局部最优解的能力。

1 粒子群算法

在文献[6]中,作者提出了粒子群算法常用的三种邻居拓扑结构:局部版本(Lbest)、全局版本(Gbest)和冯诺依曼版本(Von Neumann, Vbest)。Lbest拓扑结构中,每个粒子的邻居由与它直接相连的那些粒子构成;Gbest拓扑结构中,每个粒子的邻居由群体中除自己外的所有粒子构成;Vbest拓扑结构中,每个粒子有上、下、左、右四个邻居。这三种拓扑结构衍生出三种不同的PSO算法,如文献[7]根据粒子邻居拓扑结构的不同把粒子群算法分为局部版本粒子群算法(LPSO)和全局版本粒子群算法(GPSO)。三种结构下粒子速度与位置更新的公式描述如式(1)~(3):

$$\begin{cases} v_i(t+1) = v_i(t) + \varphi_1 \cdot r_1(p_i(t) - x_i(t)) + \\ \quad \varphi_2 \cdot r_2(p_g(t) - x_i(t)) \\ x_i(t+1) = x_i(t) + v_i(t+1) \end{cases} \quad (1)$$

收稿日期:2010-04-13;修回日期:2010-06-08。

基金项目:广东自然科学基金资助项目(9451806001002294);深港创新圈项目(200810220137A);贵州教育厅社科项目(0705204)。

作者简介:刘衍民(1978-),男,黑龙江牡丹江人,讲师,博士,主要研究方向:运筹学理论、进化计算;赵庆祯(1943-),男,山东利津人,教授,博士生导师,主要研究方向:运筹学理论、进化计算、物流工程;牛奔(1980-),男,安徽全椒人,博士,主要研究方向:进化计算。

$$\begin{cases} \mathbf{v}_i(t+1) = \mathbf{v}_i(t) + \varphi_1 \cdot r_1 (\mathbf{p}_i(t) - \mathbf{x}_i(t)) + \\ \quad \varphi_2 \cdot r_2 (\mathbf{p}_{neighbor_i}(t) - \mathbf{x}_i(t)) \\ \mathbf{x}_i(t+1) = \mathbf{x}_i(t) + \mathbf{v}_i(t+1) \end{cases} \quad (2)$$

$$\begin{cases} \mathbf{v}_i(t+1) = \mathbf{v}_i(t) + \varphi_1 \cdot r_1 (\mathbf{p}_i(t) - \mathbf{x}_i(t)) + \\ \quad \varphi_2 \cdot r_2 (\mathbf{p}_{Von\ neumann_i}(t) - \mathbf{x}_i(t)) \\ \mathbf{x}_i(t+1) = \mathbf{x}_i(t) + \mathbf{v}_i(t+1) \end{cases} \quad (3)$$

其中: $\mathbf{p}_i(t)$ 表示在迭代时刻 t , 粒子 i 所经历的最好位置, 称做粒子的“认知部分”; $\mathbf{p}_g(t)$, $\mathbf{p}_{neighbor_i}(t)$ 和 $\mathbf{p}_{Von\ neumann_i}(t)$ 表示在迭代时刻 t , 粒子 i 的邻居中所有粒子经历的最好位置, 也称为“社会部分”。这三种算法从本质上是相同的, 只是在粒子速度更新时, “社会部分”学习样本的选择范围不同。

2 不同拓扑结构的种群多样性

根据 PSO 算法的性质, 粒子间信息交流的拓扑结构将对算法的运行有直接影响。为了测试这些影响, 采用文献[8]提出的种群多样性度量方法(如式(4))作为种群多样性的度量标准。实验设置如下: 三种拓扑结构对应的算法(CF-LPSO, CF-GPSO, V-PSO)在 30 维变量函数 Sphere, Rosenbrock, Ackley, Griewanks 上独立运行 30 次, 每次运行的函数评价次数为 3×10^4 , 每种算法包含 30 个粒子。

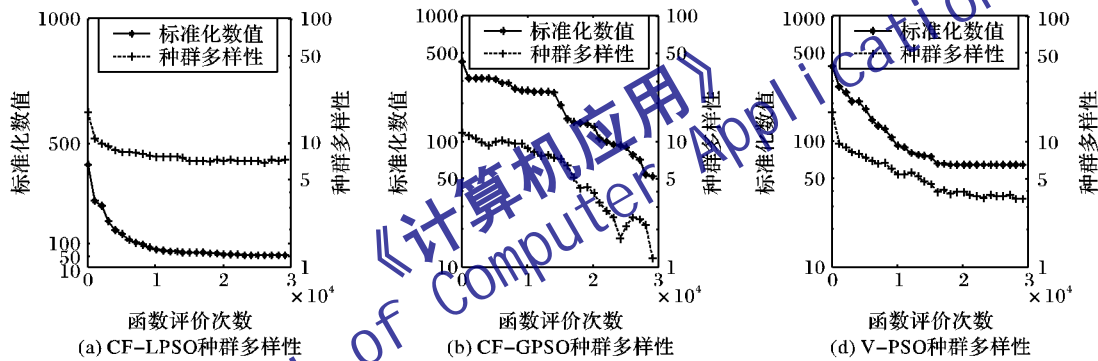


图 1 种群多样性与和标准化数值关系

图 1 显示出当种群多样性降低到一定程度后, 适应函数值无法进一步得到提升, 也就是种群中的粒子会出现“聚集”现象, 种群多样性降低。因此, 寻求一种能够增加种群多样性的粒子邻居拓扑结构对于提升算法运行效率非常重要。

3 自适应动态邻居广义学习 PSO 算法

3.1 自适应动态邻居拓扑结构

种群的拓扑结构将影响种群的多样性, 进而影响种群的搜索效率。因此, 为了增加种群的多样性, 提升粒子跳出局部最优解的能力, 本文在文献[5]算法的基础上, 提出一种自适应动态邻居拓扑结构。在该结构中, 每个粒子邻居的选择规则是粒子间的欧氏距离。根据这一准则, 当前粒子 i 的邻居由式(6)确定:

$$\begin{cases} L_i(t) = \{d_{ij}(t) \mid d_{ij}(t) = \|\mathbf{x}_i(t) - \mathbf{x}_j(t)\|, j \neq i, j \in ps\} \\ neighbor_i(t) = \arg[\min[\text{sort}(L_i(t)), n(t)]] \\ n(t) = 2 + (ps - 2) \times (e^{\frac{5(t-1)}{T-1}} - 1) \times \frac{1}{e^5 - 1} \end{cases} \quad (6)$$

其中: $L_i(t)$ 表示在迭代时刻 t 当前粒子 i 与种群中其他粒子的欧氏距离集合; ps 表示种群规模; $neighbor_i(t)$ 表示在迭代时刻 t 当前粒子 i 的所有邻居的集合; T 表示最大迭代次数; $n(t)$ 表示在迭代时刻 t , 粒子 i 拥有的邻居个数; $\arg(C)$ 表示识别

$$D(t) = \frac{\sum_{i=1}^N \sqrt{\sum_{d=1}^n (x_{id}^t - p_d)^2}}{S \cdot L} \quad (4)$$

其中: L 为搜索空间对角最大长度, N 为种群规模大小, n 为解空间维数, x_{id}^t 表示在迭代时刻 t 第 i 个粒子的第 d 维坐标值, p_d 表示所有粒子的第 d 维坐标值的平均值。平均粒子间距描述了种群中各粒子相互之间分布的离散程度, $D(t)$ 越小表示种群越集中, 种群的多样性越低。

由于不同检测函数最优值的量纲不同, 函数值用式(5)进行标准化处理, 以消除量纲的影响; 同时将标准化后的不同检测函数对应的函数值按照文献[9]处理数据的方式合并成一个数值。图 1 给出了三种拓扑结构的种群多样性与标准化函数值之间的关系, 其中图(a)右侧纵坐标采用以 10 为底的对数坐标; 图(b)、(c)中左、右侧纵坐标均采用以 10 为底的对数坐标。

$$X = \frac{x_{ij} - \mu_{ij}}{\sigma_{ij}} \quad (5)$$

其中: X 表示标准化数值, x_{ij} 表示第 j 种算法在第 i 个检测函数上的函数值, μ_{ij} 和 σ_{ij} 分别表示 x_{ij} 的均值和标准差。

元素 C 的位置; $\text{sort}(A)$ 表示对集合 A 中元素从小到大进行排序。文献[6]中已经指出, PSO 算法在运行初期阶段, 需要在可行区域内探索尽可能多的解, 这样 LPSO 算法运行效果最好; 在搜索后期, 需要在探索到的可行解中, 寻找全局最优解, 则 GPSO 算法运行效率最高。LPSO 算法中每个粒子拥有较少的邻居, 而 GPSO 算法中每个粒子的邻居是除了自身外的所有其他粒子。因此, 在本文提出的算法中, 每个粒子的邻居随着迭代次数而增加, 增加规则依照式(6)进行。

为了充分利用每个粒子的邻居信息, 提出一种自适应组建邻居策略, 即在算法运行中, 如果一个粒子的邻居中最好运行的粒子 ($\mathbf{p}_{neighbor_i}$) 连续 T 代没有得到提升, 那么对粒子 i 重新选择邻居。

利用如下实验来检测参数 T 对算法的影响: Sphere、Rosenbrock、Griewanks、Ackley、Rastrigin 和 Rastrigin_noncont 函数运行自适应动态邻居算法, 其中, 间隔代数 $T = 2, 3, 4, 5, 6, 7, 8, 9, 10$, 算法最大迭代次数是 1000; 数据处理方法与图 1 数据处理方法一致。图 2 显示了不同参数 T 对于算法的影响, 由于所有的检测函数都是最小化问题, 那么标准化值越小, 意味着运行效果越好。从图中看出当参数 $T = 7$ 时, 算法取得了最好的结果, 因此, 本文算法取参数 $T = 7$ 。

为了证实自适应动态邻居拓扑结构的优势所在, 采用式(7)的进化方程来测试各种拓扑结构对应的种群多样性, 其

中 $p_x(t)$ 分别表示 Lbest、Gbest 和 Vbest 三种拓扑结构对应的算法 CF-LPSO、CF-GPSO、V-PSO 和本文提出的 ADPSO 算法中,每个粒子邻居中运行最好的粒子。实验设置及参数处理方法与图1采用的方法一致。图3给出四种拓扑结构的种群多样性比较,可以看出自适应动态邻居拓扑结构增加了种群多样性。

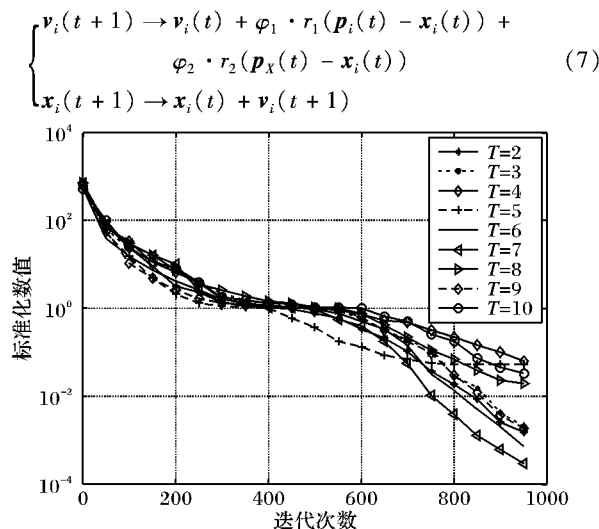


图2 不同参数 T 对于算法的影响

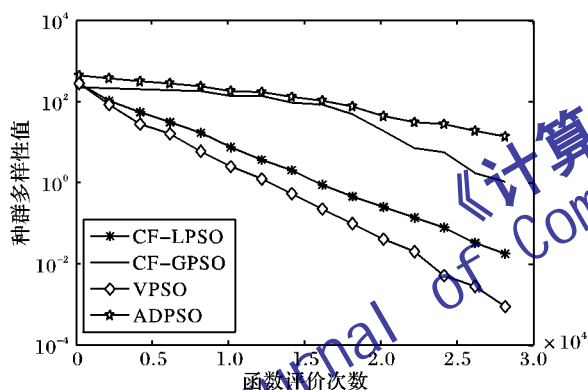


图3 不同拓扑结构的种群多样性比较

3.2 基于自适应动态邻居结构的 PSO 算法

文献中出现的各种改进粒子群算法都是基于 CF-GPSO 算法和 CF-LPSO 算法的,如何均衡这两种算法对于求解复杂的多峰问题非常重要。在 CF-GPSO 算法中,所有粒子的“社会学习”部分严格地被全局最优运行的粒子控制,因此所有的粒子都被它吸引,即使这个粒子陷入了局部最优解;而在 CF-LPSO 算法中,每个粒子受全局最优运行粒子的影响不是很大,它仅仅受到它的邻居中最好运行粒子的影响,但是它的收敛速度会很慢。鉴于此,综合 CF-LPSO 和 CF-GPSO 提出一种广义的学习策略,使每个粒子的“社会学习”部分由 Gbest 和 Lbest 组成,以融合上述两种算法的优点。向粒子邻居中最优的一个粒子学习能够使得粒子探索环绕它周围的局部最优区域,这就避免了仅仅向全局最优位置学习导致陷入局部最优解的可能;同时,在新产生的每个粒子上,加上一个随机位置以增加粒子向全局最优位置移动的概率。基于上述分析,每个粒子的速度和位置更新公式如下:

$$\begin{cases} v_i(t+1) = v_i(t) + \varphi_1 \cdot r_1(p_i(t) - x_i(t)) + \\ \quad \varphi_2 \cdot r_2(p_g(t) - x_i(t)) + \varphi_3 \cdot r_3(p_{neighbor}(t) - x_i(t)) \\ x_i(t+1) = x_i(t) + v_i(t+1) + v_{ave} \cdot N(0,1) \end{cases} \quad (7)$$

其中: $p_{i-neighbor}(t)$ 表示粒子 i 在迭代时刻 t 时,它的邻居中运行最好的粒子; $\varphi_1 = \varphi_2 = \varphi_3 = 4/3$ 。这里粒子的邻居不是固定的,而是间隔若干代动态地进行构建。这种策略极大地增加了种群的多样性,提升了种群跳出局部最优解的能力。算法1给出了 ADPSO 算法的基本流程。

算法1

```
Initialize positions and associated velocities of all particles.
Stopnum(1,ps) = 0 //No improvement iteration
For each particle(i=1:ps)
    Construct the neighborhood of the particle i using Eq. (6)
End
While (fitcount < Max_FES) && (k < iteration)
    For i=1:ps
        If stopnum(i) == 7
            Reconstruct neighborhood and stopnum = 0.
        End
    End
    For each particle (i=1:ps)
        Updating velocity and position using Eq. (8) and (9)
        Update pbest stopnum. Evaluate the fitness values.
    End for
End While
```

4 仿真实验及分析

4.1 收敛特性仿真实验

为了检测各种算法的收敛特性,选择两个单峰(Sphere (f_1) 和 Rosenbrock (f_2)) 和两个多峰(Ackley (f_3) 和 Griewank (f_4)) 作为检测函数(函数的表达式具体如文献[4]所示)。Lbest、Gbest 和 Vbest 三种拓扑结构形成的 PSO 算法分别用 CF-LPSO、CF-GPSO 和 V-PSO 来表示。为了使不同的算法具有可比性,实验的参数设置如下:每种算法都包含 30 个粒子,每个检测函数独立运行 30 次,每次运行 3×10^4 次函数评价。在 ADPSO 算法中 $T = 7$, $\varphi_1 = \varphi_2 = \varphi_3 = 4/3$,其他三种算法 $\varphi_1 = \varphi_2 = 2$ 。图4给出了算法的收敛特征曲线图。表1给出了不同算法在 30 次独立运行中的结果,其中,B 表示运行最好的结果,W 表示运行最差的结果,M 表示平均值;I 表示 CF-GPSO 算法,II 表示 CF-GPSO 算法,III 表示 V-PSO 算法,IV 表示 ADPSO 算法。

表1 不同算法的运行结果

算法	f_1	f_2	f_3	f_4
I	B 4.55E-17	2.42E+01	1.89E+00	4.66E-02
	W 6.38E-15	9.79E+01	2.11E+01	9.36E-01
	M 5.43E-16	6.58E+01	4.32E+00	8.93E-02
II	B 4.75E-14	2.41E+01	1.38E+01	3.28E-01
	W 6.92E-11	8.53E+01	6.43E+01	8.54E-01
	M 2.18E-13	5.34E+01	3.12E+01	7.51E-01
III	B 5.81E-01	2.56E+01	6.60E-01	3.23E-01
	W 9.84E-01	4.39E+01	9.94E-01	9.21E-01
	M 7.26E-01	3.14E+01	7.63E-01	7.34E-01
IV	B 5.28E-12	8.61E+00	5.78E-01	2.45E-06
	W 7.13E-09	9.64E+01	4.65E+00	4.38E-04
	M 8.39E-13	9.11E+00	6.78E-01	1.16E-05

从图4可以看出除 f_1 函数外,ADPSO 算法都取得了最好的运行效果,对于多峰问题 f_3 和 f_4 函数表现出明显的优势。从表1可以看出,ADPSO 算法在函数 f_2 , f_3 和 f_4 函数上,30 次

最好运行的结果要优于其他三种算法。

另外,由于 ADPSO 算法引入了自适应动态邻居和广义学习策略,利用 Matlab 软件中的 (tic, toc) 命令在型号为 ThinkPad-SL400 电脑上来度量达到停止标准(3×10^4 次函数

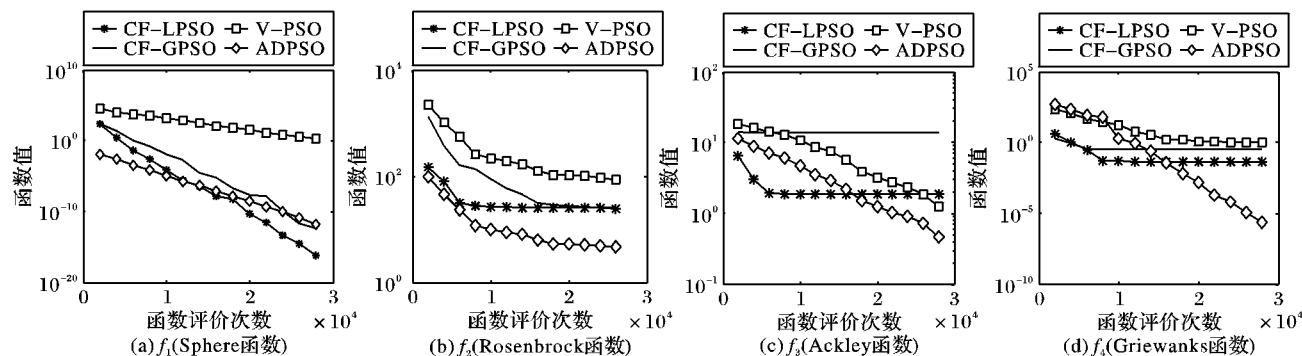


图 4 函数收敛特征曲线

表 2 达到停止标准时各种算法运行时间 s

算法	f_1	f_2	f_3	f_4
I	11	15	23	35
II	10	12	19	31
III	14	9	26	45
IV	19	30	41	66

4.2 鲁棒性分析实验

算法的稳定性将直接影响到算法的优化效率,因此,选择两个 Benchmark 函数(Ackley 和 Griewanks)来测试各种算法对同一检测函数在不同环境(函数旋转和不旋转)下的稳定性。表 3 给出了鲁棒性分析结果,这里的鲁棒性是指在 60 次独立运行过程中,算法成功达到给定的阈值的比例,各函数的阈值为分别为 0.05, 0.02 和 50。

对于 Ackley 函数,CF-LPSO, V-PSO 和 ADPSO 算法在非旋转的情况下 100% 达到了给定阈值;而旋转情况下,只有 ADPSO 算法成功地达到了阈值。对于 Griewanks 函数,不论函数是否旋转仅 ADPSO 算法成功地达到了阈值,而 V-PSO 算法在非旋转合旋转的情况下相对 CF-LPSO 和 CF-GPSO 算法显示了较稳定的运行。

表 3 各种算法的鲁棒性分析

算法	f_1		f_2		f_1^*		f_2^*	
	FEs	S	FEs	S	FEs	S	FEs	S
I	20943	100	13987	42	4152	18	13981	19
II	21207	69	9157	37	3418	21	5983	24
III	18945	100	7972	68	14739	48	22491	47
IV	1312	100	16971	100	2105	100	19437	100

注: * 表示函数进行旋转; S 表示成功率(%); FEs 表示达到阈值所需的函数评价次数

综上所述,ADPSO 算法表现出了较好的收敛特性和稳定性,引入的机制没有增加算法的计算复杂度。

5 结语

本文提出了一种基于自适应动态邻居拓扑结构的广义学习 PSO 算法(ADPSO)。在 ADPSO 算法中,根据粒子的邻居中最好运行粒子的状态动态地组建邻居;每个粒子的学习样本包括全局最优粒子、自身最优粒子和粒子邻居中最优粒子;

评价)时,每种算法所需时间是否与计算复杂度的理论分析相一致。从表 2 可以看出,ADPSO 算法在所有的检测函数上的运行时间与其他算法在同一数量级,这与计算复杂性分析基本一致。

并且在新产生的每个粒子上,增加一个随机位置提升粒子跳出局部最优解的能力。所有这些策略能增加种群的多样性,避免种群陷入局部最优解,提升了求解多峰问题的能力。从检测函数模拟结果来看,ADPSO 算法相对其他拓扑结构构成的算法获得了更好的运行结果。

将来的工作主要集中在以下两点:1)应用更复杂的检测函数(如复合函数、旋转函数)来测试提出算法的性能;2)将 ADPSO 算法应用于实际问题(金融投资组合等)以检测算法的性能。

参考文献:

- [1] KENNEDY J, EBERHART R C. Particle swarm optimization [C]// Proceedings of IEEE International Conference on Neural Networks. Washington, DC: IEEE, 1995: 1942 - 1948.
- [2] SUGANTHAN P N. Particle swarm optimizer with neighborhood operator [C]// Proceedings of IEEE Congress on Evolution Computation. Washington, DC: IEEE, 1999: 1958 - 1962.
- [3] HU X, EBERHART R C. Multiobjective optimization using dynamic neighborhood particle swarm optimization [C]// Proceedings of IEEE Congress on Evolution Computation. Washington, DC: IEEE, 2002: 1677 - 1681.
- [4] LIANG J J, QIN A K, SUGANTHAN P N, et al. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions [J]. IEEE Transactions on Evolutionary Computation, 2006, 10(3): 281 - 295.
- [5] 刘衍民, 赵庆祯. 一种基于动态邻居和变异因子的粒子群算法 [J]. 控制与决策, 2010, 25(7): 968 - 974.
- [6] KENNEDY J, MENDES R. Neighborhood topologies in fully-informed and best-of-neighborhood particle swarms [J]. IEEE Transactions on Systems, Man and Cybernetics: Part C: Applications and Reviews, 2006, 36(4): 156: 175.
- [7] CLERC M, KENNEDY J. The particle swarm-explosion, stability, and convergence in a multidimensional complex space [J]. IEEE Transactions on Evolutionary Computation, 2002, 2(6): 58 - 73.
- [8] RIGET J, VESTERSTROEM J S. A diversity guided particle swarm optimizer—the ARPSO [R]. University of Aarhus, Department of Computer Science, 2002.
- [9] MENDES R, KENNEDY J, NEVES J. The fully informed particle swarm: Simpler, maybe better [J]. IEEE Transactions on Evolutionary Computation, 2004, 7(8): 204 - 210.