

求解复杂优化问题的双层多种群粒子群优化算法^①

高 芳^② 崔 刚 吴智博 刘宏伟 杨孝宗

(哈尔滨工业大学计算机科学与技术学院 哈尔滨 150001)

摘 要 为解决粒子群优化算法存在的易早熟和精度低问题,提出了一种双层多种群粒子群优化算法。此算法采用上下两层,即下层 N 个基础种群和上层一个精英种群。各个基础种群相互独立进化,并从精英种群中得到优良信息指导自己的进化。上层精英种群首先通过接受各基础种群的当前最优粒子来更新自己的粒子集合,然后执行自适应变异操作,最后随机地向每一个基础种群输出本次进化后的一个最优粒子来改进其下一轮搜索。该算法的并行双进化机制增加了群体的随机性和多样性,提高了全局搜索能力和收敛精度。实例仿真表明该算法具有较好的性能,尤其对于复杂多峰函数优化,成功率显著提高。

关键词 粒子群优化(PSO), 双层多种群, 精英种群, 自适应变异

0 引 言

在解决复杂优化问题方面,例如多目标车间调度问题^[1]、路径优化^[2]、求解 TSP 问题及神经网络参数优化等^[3-6],粒子群优化(PSO)算法已经被证实非常有效^[7,8]。然而同其它进化算法相似,粒子群优化算法也采用随机搜索机制,同样存在易早熟和搜索精度低的缺点。造成粒子群优化算法早熟的直接原因是在进化过程中,粒子只接受其自身当前最好位置和种群当前最好位置的指导,各粒子之间没有信息交流,单个粒子所接受的信息非常局限,粒子急速奔向当前最优位置的意向很强烈,群体很快就失去多样性,从而造成算法很容易陷入局部极值,出现早熟收敛,这种现象在解决复杂多峰优化问题时更为突出。

近年来,多种群并行协同进化思想获得越来越多的关注和应用^[9-13],采用多种群思想逐渐成为解决进化算法不足的一个很好策略。本文提出了一种双层多种群粒子群优化(bi-level multi-population PSO, BMPSO)算法,该算法采用双层多种群的群体构架和并行运行机制,可以充分实现粒子群算法的群体间信息交流与共享,仿真实验结果表明其具有很好的性能。

1 多种群思想描述

克服粒子群优化算法易早熟及精度低的缺点正是我们改进算法的直接出发点。在粒子群算法中,指导粒子飞行的当前最优解只有一个,这势必无法实现对问题解空间的大范围覆盖,在这里本文将当前最优解扩大为一个集合,称为精英种群。该种群由各子群在每一次迭代过程中产生的最优解集与精英种群中全部粒子比较后得到的一组最优解集组成,这种策略的目的就是要保证精英种群的粒子永远是最优的,保证算法的收敛速度。同时,要实现粒子群算法具有很好的性能,必须做到在迭代初期具有很好的全局搜索能力及在迭代后期具有很强的局部搜索能力,这样才能在找到全局最优解的同时提高解的精度。

在精英种群中我们采用了自适应的变异策略,在算法迭代前期首先采用均匀变异策略,保持算法具有很好的全局搜索能力。随着搜索范围的逐渐减小,当满足给定的条件后自动改变为高斯变异,提高局部搜索能力,改善解的精度。

2 双层多种群粒子群优化算法

2.1 双层多种群粒子群优化算法结构

在双层多种群粒子群算法中,首先将群体分成

^① 863 计划(2006AA01A103)资助项目。

^② 女,1973 年生,博士;研究方向:人工智能,智能优化算法研究及应用;联系人, E-mail: gaofang@hit.edu.cn
(收稿日期:2008-03-19)

如图 1 所示的上下层两个层次:上层是由具有问题较好解的粒子组成的精英种群,下层由 N 个分别独立的粒子群构成的基础种群集合。上下两层组成一个相互影响的闭环循环。

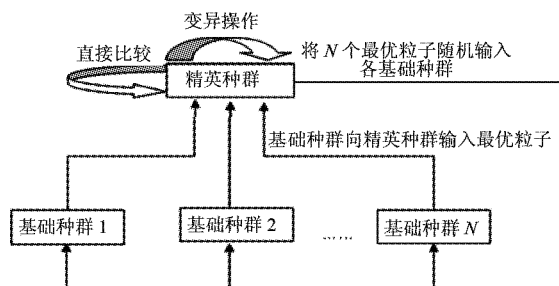


图 1 双层多种群粒子群算法结构图

这种思想与分群思想相类似,基础种群中各子群之间分别独立地进行初始化和迭代进化,相互之间没有信息交流。信息交流与共享在精英种群与基础种群中的每个子群之间独立地完成:精英种群的粒子由所接受的基础种群中各子群最好粒子与自身当前粒子直接比较竞争的优胜者组成。随后,精英种群中的粒子经过迭代后将自己最好的粒子输出到基础种群的每一个子群中,指导其粒子继续进化。基础种群的各子群所得到的当前最优解不尽相同,覆盖的解空间也不相同,因此 BMPSO 算法可以广泛地收集每个种群中的最优解,使最优解的特性范围更广。与传统的粒子群算法相比,BMPSO 算法的解空间具有更好的全局覆盖能力,从而降低了粒子陷入局部极值的概率,提高了算法的搜索效能。

BMPSO 算法的具体描述如下:精英种群从每个子群选择较好的粒子来进行初始化。下层的每个粒子群定期向上层精英种群输送好的粒子来取代其中较差的粒子,同时,精英种群向下层的每个粒子群输送自己的全局最好解的粒子,指导下层粒子群空间的粒子进化。为了提高所提算法的性能,不仅对下层粒子群中的每个粒子速度更新公式做了改进,还对精英种群中的粒子采用了两种可以自动转换的变异操作。

下层的每个子群分别进行初始化和搜索最优解。为了确保候选解的多样性以防止算法陷入局部最优解,每个子群分别被设置不同的参数。例如权重、最大和最小速度等参数。粗粒度参数的子群负责搜索全局最优解,而其它种群用于细化局部搜索和增强求解精度。

2.2 基础种群的改进策略

单个基础粒子群中的最优解只是群内信息的最

好值,包含问题解的信息较为片面,易于陷入局部最优。精英粒子群中每个粒子都是当前的基础粒子群算法产生的最优解及以前产生的最优解的后代集合,全面包含了问题解空间的信息,对基础粒子的下一步进化起到有效的指导作用。

为了提高基础粒子群的全局搜索能力,在其粒子的速度更新公式中加入一个附加项,用来调节粒子向精英种群中最好的粒子方向飞行。改进的粒子速度更新公式为

$$v_{ij}^{(k+1)} = \omega v_{ij}^{(k)} + c_1 r_1 (pbest_{ij}^{(k)} - x_{ij}^{(k)}) + c_2 r_2 (gbest_j^{(k)} - x_{ij}^{(k)}) + c_3 r_3 (Gbest_j^{(k)} - x_{ij}^{(k)}) \quad (1)$$

式中, c_3 是加速度常量。 r_3 是 $[0, 1]$ 区间的随机数。 $Gbest_j$ 是精英种群中的最好粒子位置。其它参数意义参见文献[1]和[2]。

而粒子位置更新公式与标准粒子群相同,具体为

$$x_{ij}^{(k+1)} = x_{ij}^{(k)} + v_{ij}^{(k+1)} \quad (2)$$

由公式(1)和(2)可见,下层粒子群中的粒子具有同时朝向精英种群中的最好粒子方向飞行、朝着自己先前最好位置方向飞行和朝着自己种群中先前最好粒子方向飞行的趋势,因此进一步提高了算法的全局收敛能力。

为了让算法能够更好地保持种群多样性,对不同子群的相关参数可以采用不同值,这样可以扩大对问题解空间的覆盖面。

2.3 精英种群粒子进化策略

精英粒子群中粒子的进化策略分为两步。首先接受基础种群中各子群的当前最优解作为精英种群粒子候选解,并由这些候选解与原来精英种群中的粒子直接比较竞争产生新的粒子种群;其次,新产生的粒子按照粒子群优化算法迭代公式进行粒子速度更新后,采用变异操作更新精英种群中的粒子当前位置。

对于变异操作,为了进一步改善算法的性能,BMPSO 可以在不同的迭代时期采用不同的变异算子。在算法初始阶段使用均匀分布变异,增加算法全局搜索能力,当粒子适应度增加到一定程度时,算法的变异类型自动更改为高斯变异方法,以增加算法的局部搜索能力,提高算法的精度。具体实现为:设 $f(x)$ 是粒子的目标函数, $\Delta f_i = |f(x_i^{(k)}) - f(x_i^{(k-1)})|$ 为两次相邻迭代的目标函数差值, ϵ_1 为根据具体应用预先指定的误差界值, δ 是按照上面

两种不同的变异方法随机产生的参数。当 $\Delta f_i \geq \varepsilon_1$ 时, $\delta \in U(a, b)$; 当 $\Delta f_i < \varepsilon_1$ 时, $\delta \in N(0, \sigma)$, σ 为高斯变异的标准差参数, 控制变异点的离散程度。精英种群中粒子变异操作具体如下:

$$x_{ij}^{(k+1)} = x_{ij}^{(k)} + \delta, \delta \in \begin{cases} U(a, b) & \Delta f_i > \varepsilon_1 \\ N(0, \sigma) & \Delta f_i < \varepsilon_1 \end{cases} \quad (3)$$

3 BMPSO 算法描述

BMPSO 算法实际是一种双进化机制, 它的具体执行步骤如下:

步骤 1: $k = 0$;

步骤 2: 分别独立初始化基础种群的各粒子群中粒子及相关参数;

步骤 3: 评价个体适应度值;

步骤 4: 初始化精英种群及相关参数;

步骤 5: 按照公式(1)更新基础种群的各粒子群中粒子的速度;

步骤 6: 按照公式(2)更新基础种群的各粒子群中粒子的位置;

步骤 7: 评价基础种群各粒子群中个体适应度值;

步骤 8: 基础种群各子群输出当前最好粒子给精英种群;

步骤 9: 直接比较产生新的精英粒子;

步骤 10: 按照精英种群更新策略更新精英种群粒子的速度和位置;

步骤 11: 按照公式(3)更新精英种群中粒子的位置;

步骤 12: 评价精英种群个体适应度值;

步骤 13: 选择前 N 个最好的粒子随机输入到各基础种群;

步骤 14: 重复步骤 5-步骤 13, 直到满足算法结束条件。

4 仿真试验

为了测试双层多种群粒子群优化算法的性能, 本文采用其它 3 种典型的粒子群优化算法与之进行对比:(1)标准粒子群(PSO), (2)惯性权重线性变化的粒子群改进算法(简称 PSO-CIV)^[14], (3)文献[13]中的多粒子群协同优化算法(PSCO)。

在仿真试验中, 使用 Rastrigin 函数和 Schwefel 函数两个著名的测试函数来进行实验, 这两个函数都是具有多个局部极值的复杂多峰函数, 很多优化算法求解时都很容易陷入局部极值。具体介绍如下:

(1) Rastrigin 函数。函数式为

$$f(x) = \sum_{j=1}^n (x_j^2 - 10\cos(2\pi x_j) + 10) \quad x_j \in [-5.12, 5.12] \quad (4)$$

它是多峰函数, 空间形状较复杂, 在 $S = \{x_i \in (-2, 2), i = 1, 2, \dots, n\}$ 范围内大约存在 9 个局部极小点, 当 $x_i = 0$ 时达到全局极小值点, 最小值为 0。用常规算法一般很难求解。是测试 PSO 算法是否陷入早熟是一个比较好的函数。此函数的三维图如图 2 所示。

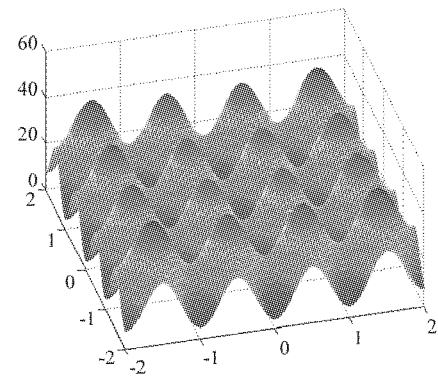


图 2 Rastrigin 函数三维图

(2) Schwefel 函数。函数式为

$$f(x) = - \sum_{j=1}^n x_j \sin(\sqrt{|x_j|}) \quad x_j \in [-500, 500] \quad (5)$$

它是一个典型的欺骗问题, 有 1 个全局极小值点, 四维时函数最小值近似为 -1675.9316 , 在 $(420.9687, 420.9687, 420.9687, 420.9687)$ 处, 距离另一个局部最优点很远, 因此如果陷入局部最优就很难跳出。此函数的三维图如图 3 所示。

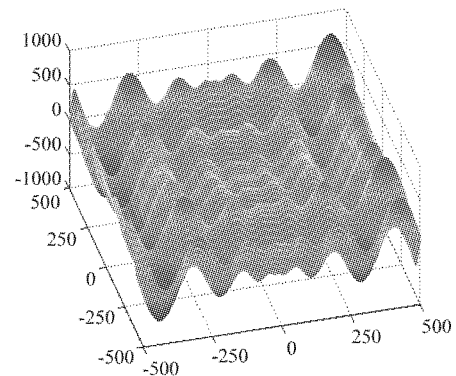


图 3 Schwefel 函数三维图形

Rastrigin 函数仿真实验 4 种算法的仿真参数如下:

(1) 标准粒子群算法仿真参数。

测试函数的维数取 5 维,种群大小为 400,惯性因子 $w = 0.8$, 迭代次数 $iterMax = 100$, 粒子速度范围: $[-2, 2]$, 加速度参数 $c_1 = c_2 = 2$ 。

(2) 多种群粒子群算法仿真参数。

测试函数的维数取 5 维,每个子群大小为 40,子群数量 $nS = 10$;惯性因子 $w = 0.8$, 内层迭代次数为 10,外层迭代次数为 10,粒子速度范围为 $[-2, 2]$, 加速度参数 $c_1 = c_2 = 2$ 。

(3) PSO-CIV 算法仿真参数。

测试函数的维数取 5 维,种群大小为 400,惯性因子 $w = 0.8$, 迭代次数为 100,粒子速度范围为 $[-2, 2]$, 加速度参数 $c_1 = 2.8, c_2 = 1.2$ 。

(4) PSCO 算法的仿真参数。

测试函数的维数取 5 维,每个子群大小为 40,子群数量 $nS = 10$;惯性因子 $w = 0.8$, 内层迭代次数为 10,外层迭代次数为 10,粒子速度范围为

$[-2, 2]$, 加速度参数 $c_1 = c_2 = 2$ 。

Schwefel 函数仿真实验的 4 种算法仿真除测试函数维数取 4 维、粒子速度范围取 $[-500, 500]$ 之外,其它均与 Rastrigin 函数试验的仿真参数相同。

由于粒子群优化算法性能对参数有很大的依赖性,以上各种算法参数设置的目的是为了尽量保证测试的公平性。以 Rastrigin 函数为例说明,各种算法的粒子总数是相同的,均为 400 个粒子,在 BMPSO 中,将 400 个粒子平均分成 10 组,每组为 40 个粒子。其次,各种算法迭代总次数也是相同的,均为 100 次,只是在 PSO 和 PSO-CIV 算法中,100 次迭代是顺序完成的,而在 BMPSO 算法中分为内层和外层迭代次数均为 10 的嵌套方式,所以该算法的总迭代次数也为 100 次,与 PSO 算法相同。最后各算法的其它测试参数均相同。

各算法测试程序运行的环境均相同,都是在酷睿™2(2GHz)双 CPU、内存为 1G 的笔记本电脑上使用 Matlab 实现的,各自分别连续运行 30 次,算法性能对照参见表 1 和表 2。

表 1 Rastrigin 函数连续运行 30 次下 4 种算法结果对照表

项目 算法	平均值	最优值	标准差	成功率	平均运行 时间(s)
PSO	0.49963	0	1.22224	76.7%	1.4343
PSO-CIV	0	0	0	100%	1.5323
PSCO	0.35207	0	0.64717	81.3%	1.6515
BMPSO	0.00026	0	0.00144	96.7%	1.7757

表 2 Schwefel 函数连续运行 30 次下 4 种算法结果对照表

项目 算法	平均值	最优值	标准差	成功率	平均运行 时间(s)
PSO	-1573.13864	-1675.93155	92.01259	36.7%	1.4254
PSO-CIV	-1581.18076	-1675.93151	90.93354	43.3%	1.5102
PSCO	-1634.42451	-1675.92969	57.92980	67.8%	1.6632
BMPSO	-1675.50315	-1675.93157	2.29281	96.7%	1.7317

表 1 数据表明, BMPSO 算法在求解 Rastrigin 函数的平均值、标准差和成功率三个方面明显优于 PSO 算法及 PSCO 算法,而成功率稍差于 PSO-CIV 算法。同时在耗时上, BMPSO 比其它算法所用时间长,主要由于该算法为双层结构,内有两层嵌套,比其它算法要复杂,从而导致耗时长。

表 2 中数据表明, BMPSO 算法在求得 Schwefel 问题解的平均值、最优值、标准差和成功率 4 个方面都优于其它算法,尤其是在平均值、标准差和成功率

方面更是具有明显的优势。再一次说明 BMPSO 算法性能更稳定。

对比表 1 和表 2 可发现,两表中成功率差别较大。对于同一 PSO 算法, Rastrigin 的成功率为 76.7%, 而 Schwefel 的成功率仅为 36.7%。这是因为 Schwefel 函数比 Rastrigin 函数更具欺骗性,一旦陷入局部极值很难逃出, 36.7% 的成功率再次印证了 PSO 算法易早熟的缺点。而在两组测试中, BMPSO 算法的成功率却是相同的,均为 96.7%, 即连续运

行 30 次,只有一次未成功,而且尽管未成功但找到的最小值极为接近函数的最优值。这充分说明所提的双层多种群粒子群算法性能非常稳定,全局搜索能力很强,求解精度高。

图 4 为 Rastrigin 函数在这 4 种算法下分别连续 6 次运行。由图 4(a)可知 PSO 的 6 次运行的结果差别较大,有 4 次完全成功,另有两次结果明显地偏离最优值,算法成功率不高。同时可看出,有些迭代曲

线形状不规则,跳跃性很大,这表明 PSO 算法具有很大的随机性。从图 4(b)中可以看出 PSO-CIV 算法在求解 Rastrigin 函数时性能很好,成功率为 100%。从图 4(c)中可以看出 PSCO 算法性能还是存在一定的波动性。而从图 4(d)中可看出连续 6 次运行均获成功,迭代曲线很平滑,这表明 BMPSO 算法是受到某种规律控制,算法的性能和稳定性都很好。

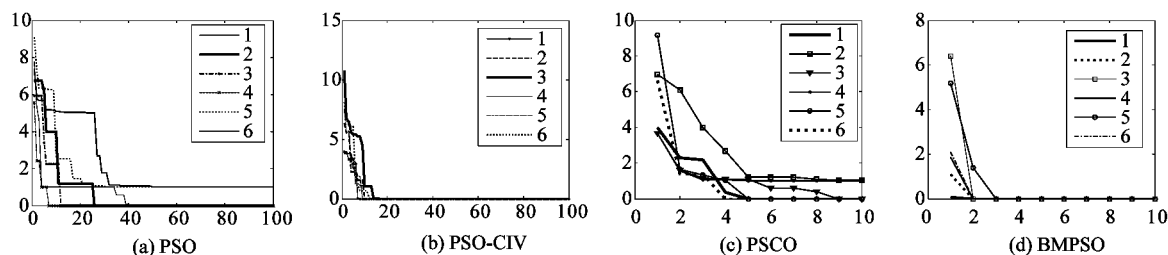


图 4 Rastrigin 函数连续 6 次运行结果

图 5 为 Schwefel 函数在四种算法下分别连续 6 次运行。类似地,从成功率、迭代曲线平滑程度都可

看出, BMPSO 的性能和稳定性都很好。

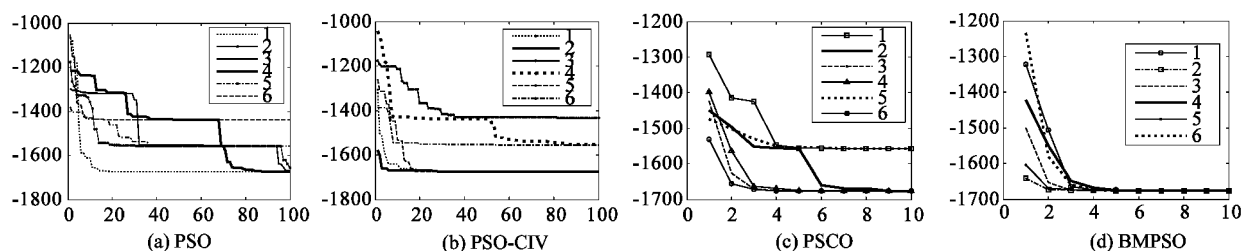


图 5 Schwefel 函数连续 6 次运行结果

在上面的两个仿真试验中,4 种算法在连续 30 次运行中,都能求得问题的最优解。但是比较 4 种算法在两个测试函数的求解实验结果发现,PSO 算法对于求解复杂的多峰问题的成功率不高,很容易陷入早熟。目前比较广泛被认同的改进算法 PSO-CIV 的性能比 PSO 算法有很大提高,对于 Rastrigin 函数的求解成功率非常高,而对 Schwefel 函数,其性能改进不大,同时它对参数的依赖程度很高,尤其是对参数 c_1 、 c_2 的取值要求很严格。就 Rastrigin 函数而言,当 $c_1 + c_2 = 4$ 时性能很好,达到 100%,而如果 $c_1 + c_2 \neq 4$,则成功率很低,只有 15% 左右。PSCO 算法在两个试验中的性能表现都不错,比较稳定。本文提出的 BMPSO 算法在两个测试实验中表现最好,性能非常稳定,成功率均为 96.7%,充分显示了双并行机制对 PSO 算法性能改进的效果。

5 结 论

双层多种群粒子群算法拓展了标准粒子群算法的群体空间,实现了算法的双并行运行机制,极大地提高了算法的效能。仿真测试结果表明,本文算法具有很好的性能,尤其是在解决复杂的多峰函数时,体现了很好的稳定性。该算法具有以下 3 个特点:

(1) 基础种群中的每个粒子群都是相互独立并行进化的,并且采用不同的参数,这样保持了种群的多样性。

(2) 精英种群采用不同的变异策略,首先采用均匀分布策略,保持算法具有很好的全局搜索能力,随着搜索范围的逐渐减小,当满足给定的条件后自动改变为高斯变异,增强局部搜索能力,提高解的精度。

(3) 基础种群中单个粒子的速度更新公式的修改,实现了种群之间的经验信息交流,避免了单一种群由于信息的片面性而造成的算法早熟现象。

参考文献

- [1] 雷德明,吴智铭. 基于粒子群的多目标作业车间调度. 上海交通大学学报,2007,41(11):1796-1800
- [2] 徐杰,黄德先. 基于混合粒子群算法的多目标车辆路径研究. 计算机集成制造系统,2007,13(3):573-584
- [3] Zhong W H, Zhang J, Chen W N. A novel discrete particle swarm optimization to solve traveling salesman problem. In: Proceedings of the 2007 IEEE Congress on Evolutionary Computation. Singapore: IEEE, 2007. 3283-3287
- [4] 黄艳新,周春光,邹淑雪等. 一种基于类覆盖和粒子群优化的模糊神经网络系统. 计算机研究与发展,2004,41(7):1053-1061
- [5] Lei D M. A Pareto archive particle swarm optimization for multi-objective job shop scheduling. *Computers & Industrial Engineering*, 2008, 54(4): 960-971
- [6] Yang Z S, Shao C, Li G Z. Multi-objective optimization for EGCS using improved PSO algorithm. In: Proceedings of the American Control Conference, New York, USA, 2007. 5059-5063
- [7] Eberhart R C, Kennedy J. A new optimizer using particle swarm theory. In: Proceedings of the 6th International Symposium on Micro Machine and Human Science, Nagoya, Japan, 1995. 39-43
- [8] Kennedy J, Eberhart R C. Particle swarm optimization. In: Proceedings of the International Conference on Neural Networks, Piscataway, NJ, USA, 1995. 4. 1942-1948
- [9] Jeffery K C, Horng S M, John W. A multi-population genetic algorithm to solve multi-objective scheduling problems for parallel machines. *Computers and Operations Research*, 2003, 30(7):1087-1102
- [10] Yao J, Kharma N, Peter G. A multi-population genetic algorithm for robust and fast ellipse detection. *Pattern Analysis & Applications*, 2005, 8(1):149-162
- [11] Patrick S, Alain P, Mourad B. A multipopulation genetic algorithm aimed at multimodal optimization. *Advances in Engineering Software*, 2002, 33(4):207-213
- [12] 蔡良伟,张基宏,李霞. 作业车间调度问题的多种群遗传算法. 电子学报,2005,33(6):991-994
- [13] 李爱国. 多粒子群协同优化算法. 复旦学报,2004,43(5):923-925
- [14] Shi Y, Eberhart R C. Parameter selection in particle swarm optimization. In: Proceedings of the 7th International Conference on Evolutionary Programming VII, San Diego, California, USA, 1998. 591-600

A Bi-level multi-population particle swarm optimization algorithm for solving complicated optimization problems

Gao Fang, Cui Gang, Wu Zhibo, Liu Hongwei, Yang Xiaozhong

(School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001)

Abstract

To overcome the PSO algorithm's drawbacks of easily premature converging and low convergence precision, the paper proposes a new improved bi-level multi population particle swarm optimization (PSO) algorithm. This algorithm includes two levels: the lower level of N basic swarms and the upper level of elite swarm. These basic swarms independently evolve and obtain the advanced information to supervise their evolution. The elite swarm first accepts the current optimal particles from each basic swarm to update its particle set, then executes an adaptive mutation, and randomly outputs one of the current optimal particles to each swarm to improve its next search. The parallel dual evolving mechanism in this algorithm enhances the swarm randomness and diversity, and improves the global search ability and converging precision. The simulations show that this algorithm has better performance, and particularly its success rate is significantly increased for the multi-peak function.

Key words: particle swarm optimization (PSO), bi-level multi-population, elite swarm, adaptive mutation