

文章编号:1005-0523(2013)03-0044-06

一种基于邻域空间的混合粒子群优化算法

曾毅,朱旭生,廖国勇

(华东交通大学基础科学学院,江西 南昌 330013)

摘要:针对标准粒子群优化算法在信息共享机制的不足,提出基于邻域空间的混合粒子群优化算法。该算法修改了粒子速度更新方程,提出了一种将模式搜索算法嵌入粒子群优化算法新方法。通过4个典型的测试函数的实验研究,表明了所提出的算法充分发挥了模式搜索算法强大的局部搜索能力和基于邻域空间的粒子群优化算法的全局寻优能力,很好地平衡了算法的全局“探索”与局部“开发”。新算法具有优化精度高、鲁棒性强的特点,特别适合对高维多峰函数进行优化。

关键词:邻域空间;模式搜索算法;粒子群优化算法

中图分类号:TP18

文献标志码:A

粒子群优化算法(particle swarm optimization, PSO)是Eberhart和Kennedy1995年开发的一种新的群体智能计算技术^[1-2],源于对鸟群捕食的行为研究。由于PSO算法概念简单,容易实现,只有少数参数需要调整,所以自其被提出以来受到学术界的重视和研究,并广泛地应用于函数优化、神经网络训练、模糊系统控制等领域。

PSO算法是属于有导向的随机性启发式算法。尽管在求解大多数优化问题时表现出色,但在求解高维复杂函数优化问题时,极易陷入局部最优解,进化后期收敛速度及解的精度也不够理想。为此许多学者提出了不少改进算法,Shi和Eberhart提出了惯性权重方法^[3-4],用惯性权重 ω 来控制前面的速度对当前速度的影响,较大的 ω 可以加强PSO算法的全局搜索能力,而较小的 ω 能加强PSO算法的局部搜索能力。该方法加快了收敛速度,提高了PSO算法的性能。但当待解问题很复杂时,在迭代后期全局搜索能力不足,导致不能找到要求的最优解。Clerc提出收缩因子概念^[5],通过引入收缩因子 χ ,可以有效地控制惯性权重 ω 和学习因子 c_1, c_2 ,以确保算法收敛。Kennedy等^[6-7]提出几种基本的邻域结构及邻域的动态选择策略,结果表明邻域结构影响算法的性能。Ratnaweera等^[8]为改善算法的性能,提出了学习因子 c_1, c_2 随时间动态线性调整的策略。文献[9-10]将求解无约束最优化问题的直接法嵌入粒子群优化算法中,加快了算法的收敛速度,提高了解的质量。本文引入邻域空间概念,提出一种基于邻域空间的混合粒子群优化算法(hybrid particle swarm optimization),该算法提出新的粒子速度更新方程,给出学习因子 c_1, c_2 非线性调整策略及局部搜索算法嵌入粒子群优化算法的新方法。数值模拟表明新的算法很好地平衡了算法的全局“探索”与局部“开发”,在避免早熟现象的发生、改善迭代后期的收敛速度和解的精度效果明显。

1 标准粒子群优化算法

式(1)(2)为标准粒子群优化算法的在第 $d(d=1, 2, \dots, n)$ 维上的速度和位置更新方程。

$$v_{id}^{k+1} = \omega v_{id}^k + c_1 \text{rand}() (pbest_{id}^k - x_{id}^k) + c_2 \text{rand}() (gbest_d^k - x_{id}^k) \quad (1)$$

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

其中: k 为进化代数, v_{id}^k 表示粒子 x_i 在第 k 代 d 维上的速度; x_{id}^k 表示粒子 x_i 在第 k 代 d 维上的位置;

收稿日期:2013-03-26

基金项目:国家自然科学基金项目(11161021);华东交通大学科学基金项目(09111114)

作者简介:曾毅(1965—),男,副教授,硕士,研究方向为智能计算。

$pbest_i$ 表示粒子 x_i 所经历的最好位置; $gbest$ 表示群体中所有粒子所经历的最好位置; c_1, c_2 为学习因子,且取非负常数; $rand()$ 是均匀分布于 $[0, 1]$ 之间的随机数; ω 为惯性权重,它决定了粒子先前速度对当前速度的影响程度,从而起到平衡算法全局和局部搜索能力的作用。

粒子的速度更新方程(1)由三部分构成,其第一部分为 ωv_{id}^k 表示粒子先前速度所起的惯性作用,它使粒子有在搜索空间中扩张的趋势,从而使算法具有全局搜索能力;其第二部分为 $c_1 rand()(p_{id}^k - x_{id}^k)$,也称为个体“认知(cognition)”部分,表示粒子本身的思考。如果标准粒子群的速度更新方程仅包含认知部分(称为 cognition-only-model),这时不同的粒子间缺乏信息交流,并没有社会信息共享。一个规模为 n 的粒子群等同于运行了 n 次单个粒子,这样明显地降低了找到最优解的概率;最后部分为 $c_2 rand()(g_d^k - x_{id}^k)$,也称为“社会(social)”部分,表示粒子间的社会信息共享。如果标准粒子群的速度更新方程仅包含社会部分(称为 social-only-model),则粒子缺乏个体的认识能力。虽然粒子在相互作用下,有能力到达新的搜索空间,但对于复杂优化问题,容易陷入局部最优解^[11]。

2 基于邻域空间的混合粒子群优化算法

2.1 算法思想

粒子间的信息共享是 PSO 算法的基础,而利用哪些信息,如何利用信息则是信息共享机制的核心问题。在标准粒子群优化算法中,所有粒子进化过程中仅共享当前种群中最优粒子的信息,而忽略了其它次运行获得的最优信息。这种信息共享策略使算法在进化后期局部开发能力较强,而全局探索的能力却较弱。一旦粒子赶上种群最优,粒子会聚集到相同位置并停止移动,种群的多样性会逐渐丧失,从而导致“早熟”现象的发生。粒子在寻求共同认识的过程中,粒子会保留自己的最佳信息,也应考虑同伴的信息,当粒子获得优于自己的同伴信息时,就会参考同伴信息,更新自己的最佳信息。这表明粒子的同伴信息的共享可提供了进化的优势。基于以上分析,我们先给出邻域空间概念,然后提出新的更能反映信息共享机制的粒子速度更新方程以及局部优化算法嵌入粒子群优化算法的新方法。

定义 1 在第 k 代粒子群中,与粒子 x_i 欧氏距离最近的 num 个粒子的集合称为粒子 x_i 的邻域空间^[12]。

定义 2 在第 k 代粒子群中,随机选择的 num 个粒子的集合称为粒子 x_i 的邻域空间^[12]。

定义 3 若 $lbest_i^k$ 为粒子 x_i 第 k 代的邻域空间中的最佳粒子的位置,称之为粒子 x_i 的邻域极值。每个粒子都有邻域极值,且不同的粒子的邻域极值可能相同。

本文将粒子 x_i 速度更新方程修改:

$$v_{id}^{k+1} = \omega v_{id}^k + c_1 rand()(plbest_{id}^k - x_{id}^k) + c_2 rand()(gbest_d^k - x_{id}^k) \quad (3)$$

其中: $plbest_i$ 为粒子 x_i 迄今为止所搜索到的最好的邻域极值; c_1, c_2 仍称为学习因子,且学习因子 c_1, c_2 随着进化代数 k 变化。学习因子 c_1, c_2 的取值如下

$$\begin{aligned} c_1 &= \alpha(k) = 2.5 - 2.0 \exp(-10(k/run_max)^s) \\ c_2 &= 3 - c_1 \end{aligned} \quad (4)$$

其中: run_max 为算法迭代的总次数, s 可取大于 1 的整数,本文取 $s=5$ 。在整个迭代过程中,学习因子 c_1 从 2.5 非线性逐渐递减至 0.5,而学习因子 c_2 从 0.5 非线性逐渐递增至 2.5。迭代初期,学习因子 c_1 取值较大,学习因子 c_2 取值较小,算法强调粒子共享邻域空间的局部信息,体现局部优化的特性;而在进化的后期,学习因子 c_1 取值较小,学习因子 c_2 取值较大,算法强调粒子共享种群中最优粒子的信息,体现全局优化的特性。

2.2 模式搜索算法

在智能优化算法研究中,人们发现单靠一种算法往往无法保持探测(exploration)和开发(exploitation)的平衡,从而影响了算法的性能,所以人们想到了将两种算法或者多种算法混合在一个模型当中,使混合

的算法能充分发挥各种算法的优点,起到提升算法的性能的目的。

模式搜索算法由 Hooke 和 Jeeves 于 1961 年提出的,一种不需要计算导数的无约束最优化的直接算法^[13]。算法主要由探测移动和模式移动过程组成,探测移动是沿坐标轴方向的移动,模式移动则是沿相邻两个探索点连线方向上移动,两种移动交替进行,顺着函数值下降的最佳方向搜索到函数的最小值。模式搜索算法的步骤^[14]:

Step1: 给定初始点 $x^{(0)} \in R^n$, n 个坐标方向 e_1, e_2, \dots, e_n , 初始步长 δ , 加速因子 $\alpha \geq 1$, 缩减率 $\beta \in (0, 1)$, 允许误差 $\varepsilon > 0$, 置 $y^{(0)} = x^{(0)}$, $k=1, j=1$;

Step2: 如果 $f(y^{(j)} + \delta e_j) < f(y^{(j)})$, 则令 $y^{(j+1)} = y^{(j)} + \delta e_j$, 进行 Step4; 否则, 进行 Step3;

Step3: 如果 $f(y^{(j)} - \delta e_j) < f(y^{(j)})$, 则令 $y^{(j+1)} = y^{(j)} - \delta e_j$, 进行 Step4; 否则, 令 $y^{(j+1)} = y^{(j)} + \delta e_j$, Step4;

Step4: 如果 $j < n$, 则置 $j = j + 1$, 转 Step2; 否则, 进行 Step5;

Step5: 如果 $f(y^{(n+1)}) < f(x^{(k)})$, 则进行 Step6; 否则, 进行 Step7;

Step6: 置 $x^{(k+1)} = y^{(n+1)}$, 令 $y^{(1)} = x^{(k+1)} + \alpha(x^{(k+1)} - x^{(k)})$, 置 $k = k + 1, j = 1$, 转 Step2;

Step7: 如果 $\delta \leq \varepsilon$, 则停止迭代, 得点 $x^{(k)}$; 否则, 置 $\delta = \beta\delta, y^{(1)} = x^{(k)}$;

$x^{(k+1)} = x^{(k)}$, 置 $k = k + 1, j = 1$, 转 Step2。

如何使混合优化算法能充分发挥模式搜索算法强大的局部“开发”能力,又有效地克服模式搜索算法对初始值敏感、易陷入局部最优的缺点。目前,通用的一个做法是以最优粒子为初值进行模式搜索,并将搜索到的新位置替代粒子原来位置,但这种算法对高维多峰函数优化问题的全局寻优能力并不理想。考虑到粒子群优化算法粒子的“聚集”特性,若粒子 x_j^k 的位置是第 k 代粒子群中 m 个粒子的邻域极值,则 m 越大,表明 x_j^k 所在的位置附近存在最优解可能性越大。以这样的粒子为初值进行局部搜索就应该比较容易找到问题的最优解。为此,在粒子群优化算法中嵌入模式搜索算法的条件是,若 K 为某一给定正整数,粒子 x_j^k 的位置是 m 个粒子的邻域极值,且 $m \geq K$,则以粒子 x_j^k 为模式搜索算法的初值进行深度开发。

2.3 混合粒子群优化算法步骤

算法具体步骤:

Step1: 设置模式搜索算法和粒子群优化算法的参数,初始化种群中各粒子的位置和速度;

Step2: 评价初始种群中粒子,确定每个粒子邻域极值 $lbest$, 并将每个粒子邻域极值 $lbest$ 作为每个粒子的 $plbest$, 初始种群中的最佳位置设置为 $gbest$;

Step3: 判定粒子是否满足嵌入模式搜索算法的条件,若条件满足,则以该粒子为初值,采用模式搜索算法进行深度开发,并将搜索到的位置和相应的适应值替代粒子开发前的位置和适应值;

Step4: 按式(2)(3)更新粒子的速度和位置,评价种群的所有粒子,并更新 $plbest$ 和 $gbest$;

Step5: 若满足算法的终止条件,则输出 $gbest$ 及其适应值;否则,转 step3。

3 实验结果及其分析

为了测试本文提出的混合粒子群优化算法的性能,选取如表 1 所示的 4 个全局最优值为 0 的最小化函数进行仿真实验。混合粒子群优化算法的参数设置见表 2,最大速度限制 V_{\max} 取为各函数初始范围的上限,惯性权重设置为从 0.9 到 0.4 的线性变化。为方便表述,本文的算法 1 为标准粒子群优化算法,粒子按式(1)(2)更新速度和位置,学习因子 c_1, c_2 均取值为 2; 算法 2 为基于邻域空间的粒子群算法,粒子按式(2)(3)更新速度和位置,粒子的邻域空间按定义 1 产生,学习因子 c_1, c_2 按式(4)取值; 算法 3 为算法 1+模式搜索算法,每代仅对最优粒子进行模式搜索; 算法 4 为算法 2+模式搜索算法,每代仅对满足局部搜索条件的粒子进行模式搜索。本文采用 Matlab7.1 实验平台进行仿真实验。实验结果如表 3 所示,其中 $MEAN$,

BEST、*WORST*、*SD* 分别表示算法独立运行 20 次的平均值、最佳适应值、最差适应值和适应值方差, *SR* 表示达到优化目标的寻优次数占实验次数的比例。所谓达到优化目标,是指算法搜索到的测试问题解与问题的最优解之差的绝对值小于 1.0×10^{-4} 。为了直观地反映算法的寻优效果,图 1 给出了不同算法对相关测试函数的收敛曲线。

表 1 测试函数
Tab.1 Test functions

函数	搜索空间	函数	搜索空间
$f_1(x) = \sum_{i=1}^n (\sum_{j=1}^i x_j)^2$	$[-100, 100]^{30}$	$f_3(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos(\frac{x_i}{\sqrt{i}}) + 1$	$[-600, 600]^{30}$
$f_2(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$[-30, 30]^{30}$	$f_4(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + e$	$[-32, 32]^{30}$

表 2 参数设置
Tab.2 Parameter setting

参量	取值	参量	取值	参量	取值
种群规模	40	HJ算法初始步长 δ	0.5	学习因子 c_1	2.5~0.5
最大进化代数	1 000	HJ算法加速因子 α	1.2	学习因子 c_2	0.5~2.5
初始惯性权重	0.9	HJ算法缩减率 β	0.2	<i>num</i>	7
最终惯性权重	0.4	HJ算法允许误差 ϵ	1.0×10^6	<i>K</i>	7

表 3 4种算法仿真实验结果
Tab.3 Four kinds of algorithm simulation results

函数	算法	<i>MEAN</i>	<i>BEST</i>	<i>WORST</i>	<i>SD</i>	<i>SR</i> /%
f_1	算法 1	1.8309×10^{03}	8.9281×10^2	3.3966×10^3	6.9774×10^2	0
	算法 2	1.2977	6.5751×10^{-2}	9.5086×10	2.0993×10	0
	算法 3	4.5768×10^{-12}	2.0464×10^{-14}	3.8460×10^{-11}	1.0887×10^{-11}	100
	算法 4	7.1519×10^{-19}	6.1453×10^{-20}	4.1798×10^{-18}	1.0664×10^{-18}	100
f_2	算法 1	2.7577×10^3	3.6654×10^2	5.9280×10^3	1.7197×10^3	0
	算法 2	3.6247×10^2	3.6343×10	1.4468×10^3	3.2859×10^2	0
	算法 3	8.0754×10^{-13}	3.0356×10^{-13}	1.6391×10^{-12}	4.325×10^{-13}	100
	算法 4	5.0582×10^{-17}	8.0966×10^{-18}	1.2368×10^{-16}	3.0365×10^{-17}	100
f_3	算法 1	2.1057×10^{-2}	1.5476×10^{-4}	1.1466×10^{-1}	2.5391×10^{-2}	0
	算法 2	1.0943×10^{-2}	7.4938×10^{-12}	9.8205×10^{-2}	2.2171×10^{-2}	55
	算法 3	0.3423	0	0.7484	0.2524	5
	算法 4	0	0	0	0	100
f_4	算法 1	2.8118	1.2249×10^{-3}	4.7783×10	2.6500×10	0
	算法 2	1.5987	3.439×10^{-6}	3.5735×10	1.1916×10	25
	算法 3	1.1451	2.0499×10^{-11}	3.8857	1.3722	55
	算法 4	3.1548×10^{-13}	3.9080×10^{-14}	2.6148×10^{-12}	5.7932×10^{-13}	100

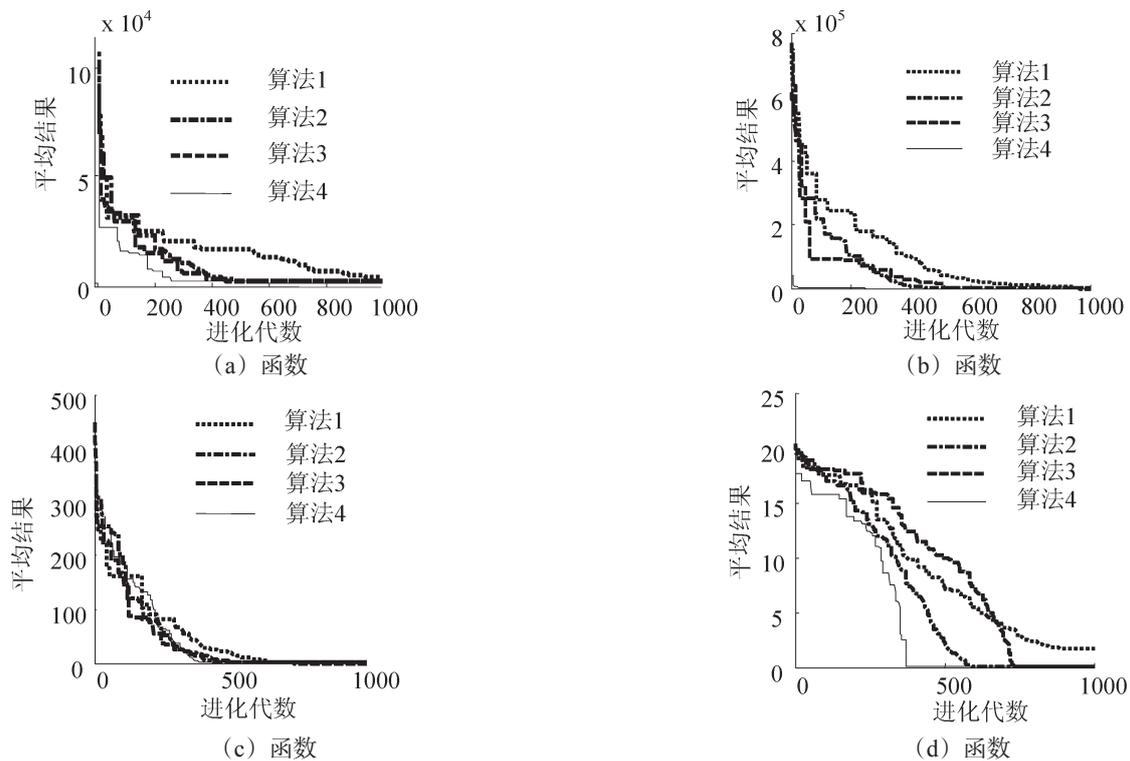


图1 测试函数收敛曲线

Fig.1 Test function convergence curve

比较表3的数据和图1测试函数收敛曲线,我们得到以下结论:

1) 对单峰优化函数而言,算法1和算法2搜索到的最优解的精度都不高,算法2的精度略好于算法1;对多峰优化函数而言,算法2表现出良好全局寻优能力,优化率明显地高于算法1。这主要是算法2在算法初期,学习因子 c_1 取值较大,递减的速度较慢,而学习因子 c_2 取值较小,递增速度较慢,算法更强调粒子共享邻域空间的局部信息,使算法在初期能够在局部范围内进行比较细致的搜索,进而保证算法2搜索到多峰优化函数的全局最优解。算法2在算法后期,学习因子 c_1 取值较小,递减的速度较慢,而学习因子 c_2 取值较大,递增速度较慢,算法强调种群中最优粒子信息的共享,加快算法收敛的速度,提高最优解的精度。但从实际搜索到的最优解来看要真正提高解的精度有必要通过局部优化算法来提升解的精度。

2) 对单峰优化函数而言,算法3和算法4搜索到的最优解的精度得到明显的提高,优化达标率均为100%,这表明模式搜索算法提高解的精度作用是明显的;但对多峰优化函数而言,算法4的全局寻优能力要比算法3强很多。这主要是因为标准粒子群优化算法易于收敛到局部最优解,且跳出局部最优解的能力较弱,对每一代的最优粒子进行模式搜索往往起到提高局部最优解的精度作用,这必然导致算法3的寻优能力不理想;算法4搜索到的最优解的平均值、最佳适应值、适应值方差和达标率都明显地好于算法2搜索到的。这表明每代对满足局部搜索条件的粒子进行模式搜索,使得算法4既能发挥算法2全局寻优能力,又能充分发挥模式搜索算法强大的局部开发能力,并较好地保持这两种能力的平衡。

4 结语

本文提出的基于邻域空间的混合粒子群优化算法较好地解决了粒子群优化算法在求解高维复杂函数优化问题时易陷入局部最优解的问题。新的算法既能充分发挥基于邻域空间的粒子群算法的全局“探测”能力,又能充分发挥模式搜索算法的局部“开发”能力。通过4个典型测试函数的实验研究表明提出算法具有优化精度高、收敛速度快、鲁棒性好,特别适合高维多峰函数的优化问题的求解。在仿真测试的过程中,

发现参数 num 、参数 K 及两种邻域空间的定义方式都会对算法的性能产生一定的影响,我们继续对这些问题作进一步的研究,并将提出的算法应用于实际问题进一步检测算法的性能。

参考文献:

- [1] EBERHART R C, KENNEDY J. A new optimizer using particle swarm theory[C]//The 6th Int Symp on Micro Machine and Human Science, Nagoya, 1995:39-43.
- [2] KENNEDY J, EBERHART R C. Particle swarm optimization[C]//Proc of the IEEE Int Conf on Neural Networks. Piscataway, 1995:1942-1948.
- [3] SHI Y, EBERHART R C. A modified particle swarm optimizer[R]. IEEE International Conference of Evolutionary Computation, Anchorage, Alaska, 1998.
- [4] SHI Y, EBERHART R C. Empirical study of particle swarm optimization[C]//Proceeding of Congress on Evolutionary Computation. Piscataway, NJ: IEEE Service Center, 1999: 1945-1949.
- [5] CLER C M. The swarm and the queen: towards a deterministic and adaptive particle swarm optimization[C]//Proceedings of the Congress on Evolutionary Computation. Piscataway, NJ: IEEE Service Center, 1999: 1951-1957.
- [6] KENNEDY J. Stereotyping: Improving particle swarm performance with cluster analysis[C]//Proceedings of the Congress on Evolutionary Computing. Piscataway, NJ: IEEE Service Center, 2000: 1507-1512.
- [7] KENNEDY J, MENDES R. Population structure and particle swarm performance[C]//Proceedings of the 2002 Congress on Evolutionary Computation, Piscataway, NJ, USA, 2002: 1671-1676.
- [8] RATNAWEERA A, HALGAMUGE S K. WATSON H C. Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients[J]. IEEE Transactions on Evolutionary Computation, 2004, 8(3): 240-255.
- [9] 李炳宇, 萧蕴诗, 汪镭. 一种求解高维复杂函数优化问题的混合粒子群优化算法. 信息与控制, 2004, 33(1): 27-30.
- [10] 贾树晋, 杜斌. Rosenbrock 搜索与动态惯性权重粒子群混合优化算法[J]. 控制与决策, 2011, 26(7): 1060-1064.
- [11] 杨维, 李歧强. 粒子群优化算法综述[J]. 中国工程科学, 2004, 6(5): 87-93.
- [12] 巩敦卫, 张勇, 张建化, 等. 新型粒子群优化算法[J]. 控制理论与应用, 2008, 25(1): 112-119.
- [13] HOOKE R, JEEVES T A. Direct Search solution of numerical and statistical problems[J]. Journal of the Association for Computing Machinery (ACM), 1961, 8(2): 212-229.
- [14] 陈宝林. 最优化理论与算法[M]. 2版. 北京: 清华大学出版社, 2005: 333-334.

Hybrid Particle Swarm Optimization Based on Neighborhood Space

Zeng Yi, Zhu Xusheng, Liao Guoyong

(School of Basic Science, East China Jiaotong University, Nanchang 330013, China)

Abstract: Considering the information sharing deficiency of standard particle swarm optimization, this paper proposes a hybrid particle swarm optimization based on neighborhood space which modifies the updating equation for particle velocity by embedding pattern search algorithm into the particle swarm. The experimental study of four typical test functions demonstrates the suggested algorithm has accomplished the balance between global “exploration” and local “exploitation” by taking advantage of the local search power of pattern search and the global optimum capacity of particle swarm algorithm based on neighborhood space. The study also shows that the suggested algorithm is especially applicable to optimizing high-dimensional multimodal functions with the characteristics of high precision and strong robustness.

Key words: neighborhood space; pattern search algorithm; PSO