一种新型的多种群微粒群算法

王 辉
WANG Hui

上海应用技术学院 计算机科学与信息工程学院,上海 200235
School of Computer Science and Information Engineering,Shanghai Institute of Technology,Shanghai 200235,China
E-mail:zgwangh@163.com

WANG Hui.Novel multi-population particle swarm optimizer.Computer Engineering and Applications,2010,46(35):45-48.

Abstract: To prevent the problem of premature convergence frequently appeared in particle swarm optimizer(PSO),a shuffled population and subpopulations dynamic of PSO(SPSDPSO) is proposed.In the approach,the whole population is divided into different subpopulations when particles search stagnated for certain iterations.Moreover,some individuals of subpopulations are re-initiated randomly and some individuals are substituted to improve the search ability further.Particles of different subpopulations are shuffled together to search for the destination after certain iterations.The processes of population and subpopulations optimization alternate are repeated until the terminal conditions satisfied.The strategy of shuffled population and subpopulations dynamic enhances the diversity of the swarm and subpopulations can exchange useful optimization information among themselves.The SPSDPSO is guaranteed to converge to the global solution with probability one.The functional test shows that SPSDPSO algorithm has advantages of convergence property.

Key words: particle swarm optimizer;subpopulation;shuffled dynamic;re-initiated randomly;substituted

摘 要: 针对微粒群算法容易出现早熟问题,提出一种动态种群与子群混合的微粒群算法(SPSDPSO)。该算法在微粒群搜索停滞时对微粒进行分群,在子群内部通过微粒随机初始化以及个体替代策略提高优化性能,在子群进化一定代数后重新混合为一个种群继续优化,种群进化与子群进化交替进行直至满足算法终止条件。SPSDPSO的种群与子群混合进化策略增强了群体多样性,并且使得子群体之间能够进行充分的信息交流。收敛性分析表明,SPSDPSO以概率1收敛到全局最优解。函数测试结果表明,新算法的全局收敛性能有了显著提高。

关键词: 微粒群算法;子群;动态混合;随机重新初始化;替代
DOI:10.3778/j.issn.1002-8331.2010.35.013 **文章编号:** 1002-8331(2010)35-0045-04 **文献标识码:** A **中图分类号:** TP18

1 引言

微粒群算法(Particle Swarm Optimizer,PSO)是一种模拟鸟类群体飞行行为的群体智能算法[1],具有参数设置简单、收敛速度快的特点,已广泛应用于神经网络训练、数据挖掘、控制参数优化等方面[2-3]。作为一种随机优化算法,微粒群算法在快速收敛的同时也具有容易出现算法早熟的缺点,针对这一问题国内外学者采用多种改进方法,如加速因子动态变化的微粒群算法[4],惯性权重非线性动态变化的微粒群算法[5]以及各种混合微粒群算法[6-8]等[9-11]。

在微粒群进化过程中,微粒在向群体最优位置聚集的同时其多样性也在迅速减少,当搜索到的群体最优位置为局部极小点时,微粒群体很难再跳出局部最小位置从而出现算法早熟。因此,保持微粒群体的多样性是避免微粒群算法出现早熟现象的关键。Løbjerg等提出将微粒分为多个种群进行目标优化,可以使微粒在搜索空间能够更广泛地分布,增强微粒种群的多样性提高算法全局收敛性能[12]。然而该算法在子群微粒的分配时具有随机性、个体之间的差异性考虑不足,子群之间的信息交流相对薄弱,因此在函数优化时并未取得理想的效果。

本文将微粒群算法的单种群进化快速收敛与多子群进化群体多样性保持相结合,提出一种动态混合种群与子群的微粒群算法(Shuffled Population and Subpopulations Dynamic of PSO,SPSDPSO)。在SPSDPSO中,微粒群在搜索停滞时将个体均匀分配到不同子群增强群体多样性,在子群进化到一定代数时混合为一个种群继续进化使得各个子群的微粒进行充分的信息交流,并且采用微粒重新初始化及个体替代措施提高子群的搜索性能。

2 基本微粒群算法

微粒群体在 D 维空间中飞行,第 i 个微粒个体的所处位

基金项目: 上海应用技术学院科研基金(the Science Foundation of Shanghai Institute of Technology under Grant No.YJ2009-06)。
作者简介: 王辉(1972-),男,博士,讲师,CCF会员,主要从事计算智能、计算机网络的研究。
收稿日期: 2010-08-10 **修回日期:** 2010-10-13

置为 $x_i=(x_{i1},x_{i2},\ldots,x_{iD})$，其飞行速度为 $V_i=(v_{i1},v_{i2},\ldots,v_{iD})$，经历过的最优位置为 $P_i=(p_{i1},p_{i2},\ldots,p_{iD})$，其中 $i=1,2,\ldots,k$。群体微粒经过的最优位置为 $P_g=(p_{g1},p_{g2},\ldots,p_{gD})$。在群体飞向最优位置的过程中，微粒通过以下公式对其速度和位置进行更新：

$$v_i(t+1)=w\cdot v_i(t)+c_1r_1(p_i-x_i(t)+c_2r_2(p_g-x_i(t)) \quad (1)$$

$$x_i(t+1)=x_i(t)+v_i(t+1) \quad (2)$$

其中，w 为惯性权重，表明微粒速度的历史信息对当前速度的影响。c_1、c_2 为正的加速系数，r_1、r_2 为[0,1]之间的随机数。速度公式(1)的第二项表明微粒个体的认知能力；第三项表明社会认知能力。基于公式(1)、(2)的微粒群算法称作基本微粒群算法。

3 动态种群与子群混合的微粒群算法

3.1 SPSDPSO的基本原理

SPSDPSO中的微粒首先在一个群体中进行目标搜索，当一定代数内适应值的变化小于阈值时微粒群出现搜索停滞，微粒群体被分为两个子群。微粒分群时，按照个体适应值进行子群分配：适应值最高的微粒分配到第一个子群，适应值次之的微粒分配到第二个子群，按上述分配原则将其余微粒分配到各个子群。这种分群方法更好地实现了个体在不同子群内的均匀分配，微粒分配完毕后子群分别对目标进行搜索。

在子群搜索出现停滞时，针对不同子群分别采取不同策略增强子群的寻优能力。在分群时，第一个子群微粒适应值高于第二个子群微粒适应值、微粒聚集在较小的搜索空间，随着搜索的进行其群体多样性迅速减少。因此，为了使第一个子群出现搜索停滞时对部分个体进行重新初始化，这样子群微粒就能在搜索空间内进行更为广泛的搜索。第二个子群微粒适应值比第一个子群微粒适应值差，在搜索空间分布较为广泛，因此在其出现搜索停滞时用部分适应值好的个体代替适应值差的个体使群体进一步向最优区域搜索。

在一定代数内两个子群未满足终止条件时，将其混合为一个种群进行目标搜索，使得不同子群内的微粒就可以实现充分的信息交流。

SPSDPSO通过不断地种群与子群动态混合优化进行目标寻优，直至终止条件满足。SPSDPSO的子群部分微粒重新初始化与替代措施相当于对子群微粒进行了变异操作，而子群混合进化措施相当于对两个子群进行了交叉操作。

SPSDPSO子群微粒速度与位置更新公式为：

$$v_{i,j}(t+1)=w(t)\cdot v_{i,j}(t)+c_3r_3(p_{i,j}(t)-x_{i,j}(t)+c_4r_4(p_{g,j}(t)-x_{i,j}(t)) \quad (3)$$

$$x_{i,j}(t+1)=x_{i,j}(t)+v_{i,j}(t+1) \quad (4)$$

其中，$i=1,2,\ldots,m$ 为子群中微粒数，$j=1,2$ 为子群数目。$v_{i,j}(t)$ 是第 j 个子群中的第 i 个微粒的飞行速度。$p_{g,j}(t)$ 为第 j 个子群进化到 t 代的全局最优位置，$p_{i,j}(t)$ 为第 j 个子群进化到 t 代所求个体最优位置。c_3、c_4 为正的加速系数，r_3、r_4 为[0,1]之间的随机数。

3.2 SPSDPSO求解过程

SPSDPSO的求解过程为：

步骤1 初始化：设置微粒群规模；子群混合前进化代数；微粒群总的最大进化代数；随机设置每个微粒的位置和速度；计算微粒的初始适应值。

步骤2 微粒群进行目标搜索，根据公式(1)、(2)更新微粒群体最优位置。

步骤3 当出现搜索停滞时按照微粒适应值大小依次分配到不同子群：即适应值最大的微粒分到子群1，适应值次之的微粒分到子群2，然后按上述分配原则将微粒分配完毕。

步骤4 在子群进化代数内，子群进行目标搜索，通过公式(3)、(4)更新子群位置，若子群最优位置优于群体最优位置则对群体最优位置进行更新。

步骤5 在子群进化代数内，若子群1出现搜索停滞，则对其部分微粒进行随机初始化，然后继续搜索；若子群2出现搜索停滞则对部分适应值差的微粒用适应值好的微粒进行替代，然后继续目标搜索。若子群搜索达到最小误差值则算法停止，否则进行子群混合。

步骤6 终止条件是否满足。满足则算法结束，否则转步骤2。

SPSDPSO流程图见图1。

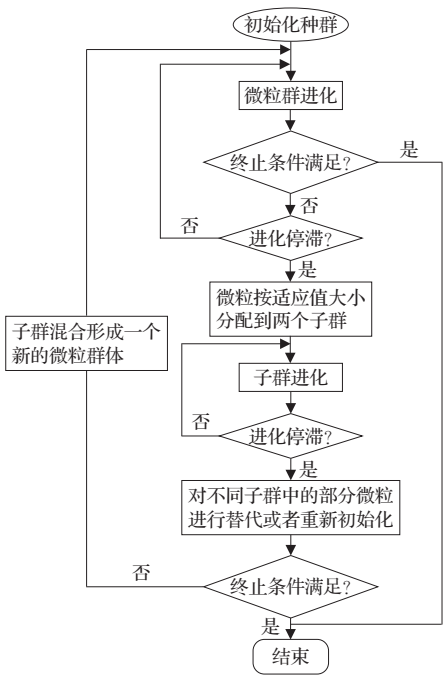

图1 SPSDPSO流程图

3.3 SPSDPSO收敛性分析

作为一种随机优化算法，基本微粒群算法的收敛性可以通过随机优化算法的收敛准则来进行分析。Solis和Wets对随机优化算法提出了其全局收敛须满足的条件，首先定义以下两个假设：

假设1 若 $f(D(z,\xi))\leq f(z)$，$\xi\in S$，则 $f(D(z,\xi))\leq f(\xi)$，其中 D 为产生问题解的函数，ξ 为从概率空间 (R^n,B,u_k) 产生的随机向量，f 为目标函数，S 为 R^n 的子集，表示问题的约束空间。u_k 为 B 上的概率度量，B 为 R^n 子集的 σ 域。

假设2 若对 S 的Borel子集 A，有 $v(A)>0$，则 $\prod_{k=0}^{\infty}(1-u_k(A))=0$，其中 $v(A)$ 为子集 A 的 n 维闭包。$u_k(A)$ 为测度 u_k 产生的概率。

然后根据假设1与假设2，有以下定理：

定理1 设 f 为一可测函数，S 为 R^n 的一可测集，$\{z_k\}_{k=1}^{+\infty}$

为随机算法产生的解序列,则当满足假设1和假设2时,有 $\lim_{k\to+\infty} P[z_k \in R_\varepsilon]=1$, R_ε 为全局最优点集合, $P[z_k \in R_\varepsilon]$ 是第 k 算法所生成的解 $z_k \in R_\varepsilon$ 的概率。

因此,当随机优化算法满足假设1与假设2时,该算法保证以概率1收敛于全局最优解。

根据分析,基本微粒群算法满足假设1[13]。

为了满足假设2,规模为 S 的微粒群的样本空间的并必须包含 S ,即 $S\subseteq\bigcup_{i=1}^{S} M_{i,k}$,其中 $M_{i,k}$ 为 k 代时微粒 i 的样本空间的支撑集。

根据基本微粒群算法的位置更新公式,有

$$X(t+1)=X(t)+wV(t)-X(t)(\psi_1+\psi_2)+\psi_1 P+\psi_2 P_g$$

其中 $\psi_1=c_1r_1$, $\psi_2=c_2r_2$,整理得:

$$X(t+1)=(1+w-\psi_1-\psi_2)X(t)-wX(t-1)+\psi_1 P+\psi_2 P_G$$

代入 $M_{i,k}$ 可得:

$$M_{i,k}=(1+w-\psi_1-\psi_2)x_{i,j,k-1}-wx_{i,j,k-2}+\psi_1 P+\psi_2 P_g$$

即

$$M_{i,k}=x_{i,j,k-1}+w(x_{i,j,k-1}-x_{i,j,k-2})+\psi_1(P-x_{i,j,k-1})+\psi_2(P_g-x_{i,j,k-1})$$

其中 $0\le\psi_1\le c_1$, $0\le\psi_2\le c_2$, $x_{i,j,k}$ 为微粒在第 k 代时第 j 维分量的长度。 $M_{i,k}$ 为一具有顶点 $\psi_1=0$, $\psi_2=0$ 和 $\psi_1=c_1$, $\psi_2=c_2$ 的超矩形体,当 $\max(c_1|P_i-x_{i,j,k}|, c_2|P_g-x_{i,j,k}|)<0.5\times diam_j(S)$ 时 $v[M_{i,k}\cap S]<v(S)$,其中 $diam_j(S)$ 表示 S 的长度。根据定理1知,基本微粒群算法随着 k 的增长,搜索空间在不断减小,每一 $M_{i,k}$ 的闭包 $v[M_{i,k}]$ 在逐渐变小,其并集 $\bigcup_{i\ne j} M_{i,k}$ 的闭包 $v[\bigcup_{i\ne j} M_{i,k}]$ 也在变小。因此存在 N ,当 $k>N$ 时, $v[\bigcup_{i\ne j} M_{i,k}\cap S]<v[S]$ 。

因此,基本微粒群算法不满足假设2,即不能保证全局收敛。SPDPSO种群与子群的微粒更新公式与基本微粒群算法相同,因此其满足假设1。SPDPSO算法在搜索停滞时通过分群与部分微粒重新初始化机制,使得微粒样本空间的支撑集 $M_k=S$,则 $v(M_k)=v(S)$,即SPSDPSO满足假设2,根据随机优化算法定理1,SPSDPSO将以概率1收敛于全局最优解。

4　SPSDPSO性能测试

4.1　测试函数

SPSDPSO性能通过Sphere、Rosenbrock、Rastrigrin和Griewank四个非线性测试函数进行性能测试:

(1)Sphere函数,其表达式为:

$$f(x)=\sum_{i=1}^{n} x_i^2\ ,\ x_i\in[-100, 100]$$

在 $x_i=0$ 时达到极小值0,各个变量之间没有相互作用。

(2)Rosenbrock函数为单峰值函数,非凸、病态函数,在一些变量之间有明显的相互作用,在 $x_i=1$ 时达到极小值0。其表达式为:

$$f_1(x)=\sum_{i=1}^{n}[100(x_{i+1}-x_i^2)^2+(x_i-1)^2]\ ,\ x_i\in[-30, 30]$$

(3)Rastrigrin函数为多峰值函数,在 $x_i=0(i=1, 2, \ldots, n)$ 时达到全局极小点0,其表达式为:

$$f_2(x)=\sum_{i=1}^{n}(x_i^2-10\cos(2\pi x_i)+10)\ ,\ x_i\in[-5.12, 5.12]$$

(4)Griewank函数为多峰值函数,全局极小0在 $x_i=100$ $(i=1, 2, \ldots, n)$ 时达到,变量之间有显著的相互作用。其表达式为:

$$f_3(x)=\frac{1}{400}\sum_{i=1}^{n}(x_i-100)^2-\prod_{i=1}^{n}\cos(\frac{x_i-100)}{\sqrt{i}})+1$$

$$x_i\in[-600, 600]$$

4.2　SPSDPSO与PSO比较

在实验中SPSDPSO、PSO微粒数分别为20、40、80与160,进化代数为2 000,加速系数 $c_1=2.0$, $c_2=2.0$, $c_3=2.0$, $c_4=2.0$,惯性权重 w 从0.9下降到0.4。SPSDPSO与PSO分别进行100次独立运算。SPSDPSO算法中,当种群进化5代而适应值变化在5%以内时进行分群寻优。当子群进化5代而适应值变化在5%以内时,子群1中10%适应值最差的微粒重新初始化,子群2中40%适应值最差的微粒被相同数目适应值高的微粒替代,子群进化200代未达到算法停止条件时混合为一个种群。

SPSDPSO、PSO优化四个测试函数的结果见表1,函数优化过程见图2~5(图2~5中SPSD20、SPSD40 SPSD80、SPSD160分别表示SPSDPSO20、SPSDPSO40、SPSDPSO80与SPSDPSO160)。图2~5中的横坐标表示进化代数、纵坐标表示适应值。各个测试函数的维数均设为30,函数优化目标值设置为0。

从仿真实验结果可以看出,在高维函数优化时,在不同种群规模情况下SPSDPSO与PSO相比,算法的鲁棒性和收敛性都有很大提高。对于单峰值函数Sphere与Rosenbrock,随着

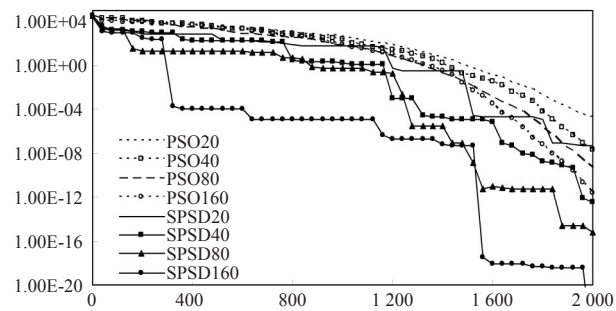

图2　不同种群规模时Sphere函数优化过程

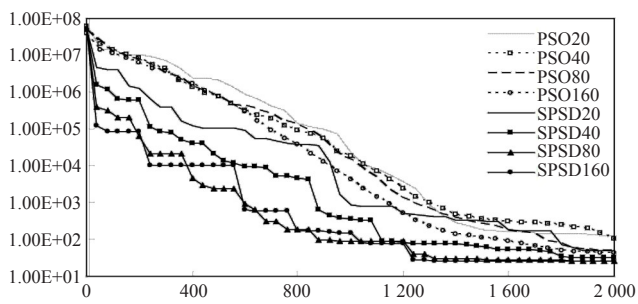

图3　不同种群规模时Rosenbrock函数优化过程

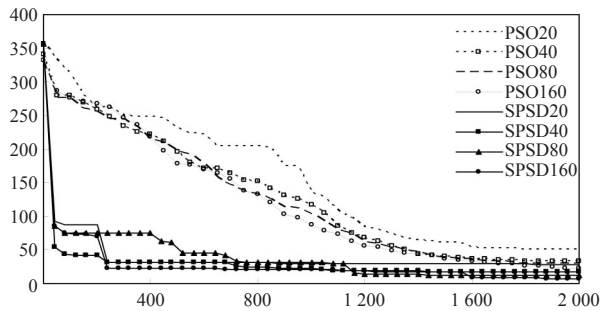

图4　不同种群规模时Rastrigrin函数优化过程

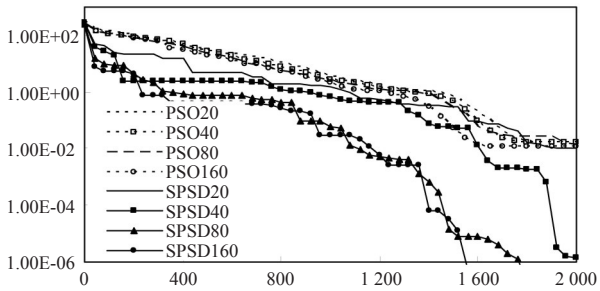

图5 不同种群规模时Griewank函数优化过程

表1 SPSDPSO与PSO函数优化平均值(标准差)比较

函数	微粒数	PSO算法	SPSDPSO算法
Sphere	20	2.14E-5(1.1751E-5)	1.44E-7(1.36E-7)
	40	8.22E-9(3.226E-9)	3.83E-13(1.95E-13)
	80	4.48E-10(2.563E-10)	5.83E-16(3.69E-16)
	160	2.61E-12(7.353E-13)	1.18E-26(7.34E-27)
Rosenbrock	20	125.202 0(137.59)	48.926 82(43.123 40)
	40	106.003 0(108.21)	31.377 25(22.604 78)
	80	50.136 5(30.820)	26.865 76(20.752 81)
	160	43.030 6(27.967)	22.867 95(18.412 71)
Rastrigrin	20	50.181 0(13.344 6)	27.792 31(13.314 91)
	40	33.951 1(7.755 4)	17.478 19(10.879 05)
	80	29.307 8(8.427 4)	14.494 45(4.898 80)
	160	23.069 2(6.356 9)	8.768 96(4.637 89)
Griewank	20	0.016 9(0.019 7)	0.010 709(0.028 638)
	40	0.016 4(0.021 1)	2.203 41E-6(7.033 26E-6)
	80	0.012 6(0.012 0)	5.225 32E-8(2.319 60E-7)
	160	0.011 6(0.011 0)	2.928 60E-13(1.267 58E-12)

种群规模的增大PSO优化结果取得显著的提高。然而对于多峰值函数Rastrigrin与Griewank,其局部最优点数目随着问题维数的增加成指数增长。对于PSO,在优化过程中很容易陷入局部最优,难以获得很好的优化效果,随着种群规模的增大其改善效果也不显著。对于SPSDPSO,在不同种群规模情况下都获得了比PSO更好的优化效果,并且随着种群规模的增大优化效果更好。这是由于PSO算法目标寻优时所有微粒都被群体中的最优个体所吸引,一旦最优微粒陷入局部最优则其余微粒很难从局部最优跳出。因此,对于难以优化的高维函数尤其是有多个局部极小点的多峰值函数时,PSO更难于取得理想的效果。SPSDPSO通过均匀分群措施以及子群微粒重新初始化与个体替代措施增强了群体多样性,使得其更容易跳出局部最优在更广泛的搜索范围内进一步寻找更好的解,通过子群混合进化增强了子群间的信息交流,提高了算法性能,因此新算法法可以获得比PSO更好的优化效果。

4.3 SPSDPSO与其他多种群微粒群算法及遗传算法比较

当种群规模为20时,SPSDPSO与文献[12]中的多种群微粒群算法(即混合繁殖操作的多种群微粒群算法,称作HPSO)、遗传算法(GA)进行了性能比较。采用的优化函数为Sphere、Rosenbrock、Rastrigrin与Griewank,优化结果见表2与表3。其中,GA、HPSO(Hybrid(1)、Hybrid(2)、Hybrid(4)、Hybrid(6)为不同子群模型的多种群微粒群算法)数据来源于文献[12],SPSDPSO参数设置同4.2节。各种算法在测试函数为10维时进化代数为1 000、为30维时进化代数为2 000。

根据表2与表3可知,对于Sphere函数只有在其10维时HPSO的Hybrid(1)与Hybrid(2)性能优于GA,其他情况下

表2 各种算法在函数为10维时的优化平均值比较

算法	Sphere	Rosenbrock	Rastrigrin	Griewank
GA	2.43E-4	283.251 0	3.166 7	109.810 00
Hybrid(1)	2.42E-4	43.521 0	3.059 9	0.090 78
Hybrid(2)	3.76E-5	51.701 0	3.561 5	0.464 23
Hybrid(4)	2.23E-3	63.369 0	3.684 0	0.692 00
Hybrid(6)	2.124E-2	81.283 0	6.803 6	0.746 94
SPSDPSO	2.837 3E-8	2.910 4	0.862 0	0.005 58

表3 各种算法在函数为30维时的优化平均值比较

算法	Sphere	Rosenbrock	Rastrigrin	Griewank
GA	4.42E-3	199.730 0	49.321 20	889.537 000
Hybrid(1)	1.120 3E-2	187.033 0	27.811 90	0.099 110
Hybrid(2)	0.173 96	196.554 0	38.589 70	0.063 160
Hybrid(4)	0.020 23	279.390 0	29.582 70	0.163 890
Hybrid(6)	0.056 69	247.724 0	29.174 70	0.375 010
SPSDPSO	1.44E-7	43.030 6	27.792 31	0.010 709

HPSO并未表现出良好的性能。对于Rosenbrock函数,HPSO算法中Hybrid(1)与Hybrid(2)性能较好。对于多峰值函数Rastrigrin与Griewank,HPSO具有较好的优化性能,特别是在维数增大时可以取得更好的优化效果。因此,HPSO通过分群策略在一定程度上提高了算法性能。然而HPSO的子群之间仅通过个别微粒的繁殖操作实现信息交流使得其信息交流不足,因此并未取得更好的优化性能,并且该算法的分群措施减缓了优化进程使得其在单峰值函数优化时性能较差。从表2与表3可以看出,在不同维数的单峰值函数与多峰值函数优化时,SPSDPSO都可以获得比其他算法更好的结果。这是由于SPSDPSO在分群优化之前,微粒首先在一个种群内进行目标搜索实现快速收敛,只有在搜索停滞时才进行分群优化,增强群体多样性,提高全局收敛性能,子群微粒的重新初始化与替代措施也仅对部分个体进行,这样就能保证微粒在保留已发现局部最优区域的同时又能探索新的最优区域进一步寻找全局最优点,因此SPSDPSO在单峰值函数与多峰值函数优化时都具有良好的性能。

5 结论

将种群进化与子群进化相结合,提出一种动态种群与子群混合的微粒群算法(SPSDPSO)。SPSDPSO在种群进化时个体可以实现充分的信息交流迅速向最优区域聚集,子群优化时个体均匀分配到不同群体,增强群体的多样性,种群进化与子群进化根据算法优化进程交替进行。在种群与子群动态混合框架下,SPSDPSO对子群采用部分微粒重新初始化以及微粒替代策略进一步提高算法性能。收敛性分析表明,SPSDPSO以概率1收敛到全局最优解。函数仿真表明,SPSDPSO优于微粒群算法、混合繁殖操作的多种群微粒群算法以及遗传算法。

参考文献:

[1] Kennedy J,Eberhart R C.Particle swarm optimization[C]//Proceedings of the IEEE International Conference on Neural Networks(Perth,Australia).Piscataway,NJ,IV,IEEE Service Center,1995:1942-1948.

[2] Poli R,Kennedy J,Blackwell T.Particle Swarm Optimization:An overview[J].Swarm Intelligence,2007,1:33-57.

(下转51页)

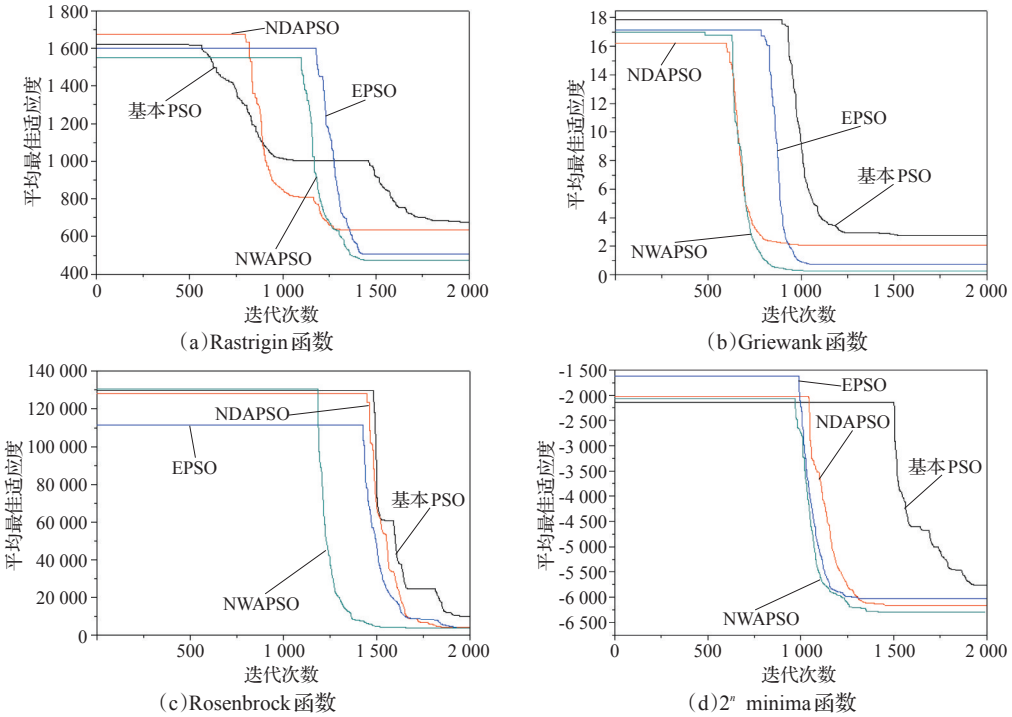

图1　4种基准函数的平均最佳适应度进化曲线

最优解的能力，摆脱了后期易陷入局部最优点的束缚。实验结果表明，NWAPSO算法增强了全局搜索能力，提高了收敛速度和收敛精度以及搜索全局最优解的能力。特别当优化问题是高维、多峰等复杂非线性时，NWAPSO算法的优越性更明显。

参考文献：

[1] Kennedy J，Eberhart R C.Particle swarm optimization[C]//Proceedings of IEEE International Conference on Neural Networks.Perth：IEEE Press，1995，4：1942-1948.

[2] Boeringer D W，Werner D H.Particle swarm optimization versus genetic algorithms for phased array synthesis[J].IEEE Trans on Antennas and Propagation，2004，52(3)：771-779.

[3] Shi X H，Liang Y C.Particle swarm optimization-based algorithms for TSP and generalized TSP[J].Information Processing Letters，2007，103(5)：169-176.

[4] 谢强，张磊，周良.基于改进粒子群优化算法的Ontology划分方法[J].华南理工大学学报：自然科学版，2007，35(9)：118-122.

[5] Chau K W.Application of a PSO-based neural network in analysis of outcomes of construction claims[J].Automation in Construction，2007，16(5)：642-646.

[6] Shi Y H，Eberhart R C.A modified particle swarm optimizer[C]//Proceedings of the IEEE Congress on Evolutionary Computation.Piscataway，USA：IEEE Service Center，1998：69-73.

[7] 陈国初，俞金寿.增强型微粒群优化算法及其在软测量中的应用[J].控制与决策，2005，20(4)：377-381.

[8] Chatterjeea A，Siarry P.Nonlinear inertia weight variation for dynamic adaptation in particle swarm optimization[J].Computers & Operations Research，2006，33：859-871.

[9] 张选平，杜玉萍.一种动态改变惯性权重的自适应粒子群算法[J].西安交通大学学报，2005，39(10)：1039-1042.

[10] 徐刚，瞿金平.一种用于多目标优化的混合粒子群优化算法[J].计算机工程与应用，2008，44(33)：18-21.

(上接48页)

[3] 王辉，钱锋.群体智能优化算法[J].化工自动化及仪表，2007，34(5)：7-13.

[4] Chatterjee A，Siarry P.Nonlinear inertia weight variation for dynamic adaptation in particle swarm optimization[J].Computers &Operations Research，2006，33：859-871.

[5] Ratnaweera A，Halgamuge S K，Watson H C.Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients[C]//IEEE Transactions on Evolutionary Computation，2004，8(3)：240-255.

[6] Zhang W J，Xie X F.DEPSO：hybrid particle swarm with differential evolution operator[C]//IEEE Interenational Conference on Systems，Man and Cybernetics(SMCC)，Washington DC，USA，2003：3816-3821.

[7] Grosan C，Abraham A，Han S，et al.Hybrid Particle Swarm-evolutionary algorithm for search and optimization[J].Lecture Notes in Computer Science，2005，3789：623-632.

[8] Shelokar P S，Siarry P，Jayaraman V K，et al.Particle swarm and ant colony algorithms hybridized for improved continuous optimization[J].Applied Mathematics and Computation，2007，188：129-142.

[9] 王光辉，曾建潮.一种具有动态群体规模的微粒群算法[J].计算机工程与应用，2008，44(11)：52-56.

[10] 袁代林，陈虬.马氏模型PSO及其随机过程分析[J].计算机工程与应用，2009，45(31)：49-52.

[11] 罗辞勇，陈民轴.克服恋食行为的PSO算法改进研究[J].控制与决策，2008，23(7)：776-780.

[12] Løbjerg M，Rasmussen T K，Krink T.Hybrid Particle Swarm Optimiser with breeding and subpopulations[C]//Proceeding of the 3rd Genetic Evolutionary Computation Conference，San Francisco，CA，2001：469-476.

[13] Van den Bergh F.An Analysis of particle swarm optimisers[D].University of Pretoria，South Aferica，2002.