二次微粒群算法及其参数自适应策略

杨亚平，谭　瑛，曾建潮
（太原科技大学　系统仿真与计算机应用研究所，太原　030024）
E-mail:scar285@sohu.com

摘　要：在对标准微粒群算法进行分析的基础上，提出了一种二次微粒群算法，并在对二次微粒群算法和标准微粒群算法进行比较分析的基础上给出了二次微粒群算法的参数自适应方案。通过对典型测试函数进行仿真，结果表明二次微粒群算法比标准微粒群算法的性能有很大提高，说明了二次微粒群算法是可行的。另外，将参数自适应时的结果同参数固定时的结果相比较，结果表明算法性能有很大提高，说明了该方案的正确性和有效性。

关键词：二次微粒群算法；标准微粒群算法；参数

文章编号：1002-8331(2006)31-0064-04　**文献标识码**：A　**中图分类号**：TP301.6

Quadratic Particle Swarm Optimization and its Self-Adaptive Parameters

YANG Ya-ping, TAN Ying, ZENG Jian-chao
(Taiyuan University of Science & Technology, Taiyuan 030024, China)

Abstract: This paper presents a Quadratic Particle Swarm Optimization and gives a strategy of self-adapting quadratic Particle Swarm Optimization parameters on the basis of comparing quadratic Particle Swarm Optimization with Particle Swarm Optimization.The simulation illustrates the new method improves the performance of the Particle Swarm Optimization.Further, for most of the benchmarks, the self-adapting parameters strategy outperforms the fixed parameters, the experimental results show that the strategy is correct and efficient.

Key words: Quadratic PSO; standard PSO; parameter

1　引言

微粒群算法是由 Kennedy 和 Eberhart 于 1995 年提出的一种新的进化算法，自提出以来，由于它的计算快速性和算法本身的易实现性，引起了国际上相关领域众多学者的关注和研究[1]。由于微粒群算法提出时间较短，在很多方面的研究还不够完善，为此，国内外的众多学者进行了多方面的研究来改进和完善微粒群算法，如参数的选择[3,4,7]，以及与其他算法相结合的混合算法[8]。

此文在分析了标准微粒群算法模型和其机理的基础上对标准微粒群算法的进化方程进行了改进，提出了一种二次微粒群算法；并在对二次微粒群算法和标准微粒群算法的进化方程进行比较分析的基础上，给出了一种二次微粒群算法参数的自适应策略。通过对典型测试函数进行仿真，结果表明了该参数自适应方案的正确性和有效性。

2　标准微粒群算法及其分析

由 Kennedy 和 Eberhart 受鸟群觅食行为的启发而提出的微粒群算法是近年来发展起来的一种新的进化算法，主要用于解决优化问题。算法采用速度-位置搜索模型，每个微粒代表界空间中的一个解，解的优劣程度由适应函数决定。其中，适应函数由优化目标来决定。每个微粒在搜索空间中以一定的速度飞行，并根据自己的飞行经验和群体的飞行经验来调整自己的飞行。

PSO 随机初始化为一群粒子，每个粒子根据自己的历史最好位置 P_i 和群体历史最好位置 P_g 来更新自己的速度和位置。设 $X_i=(X_{i1},X_{i2},\cdots,X_{in})$ 为微粒 i 的当前位置，$V_i=(V_{i1},V_{i2},\cdots,V_{in})$ 为微粒 i 的当前飞行速度，$P_i=(P_{i1},P_{i2},\cdots,P_{in})$ 为微粒 i 所经历过的最好位置，则粒子根据以下公式来更新自己的速度和位置：

$$V_i(t+1)=w*V_i(t)+c_1*r_1*(P_i-X_i(t))+c_2*r_2*(P_g-X_i(t)) \quad (1)$$

$$X_i(t+1)=X_i(t)+V_i(t+1) \quad (2)$$

其中，c_1 和 c_2 是加速常数，通常在 0~2 间取值；r_1 和 r_2 是[0,1]上的随机数；w 为惯性权重，w 使微粒保持运动惯性，使其有扩展搜索空间的趋势，有能力探索新的区域。式(1)中第一部分是粒子先前的速度，第二部分为“认知”部分，考虑微粒自身的经验，表示微粒自身的思考，第三部分为“社会”部分，表示微粒间的信息共享。这三部分共同决定粒子的空间搜索能力。

考虑标准微粒群算法的微粒的进化方程，从式中可以看出，$c_1*r_1*(P_i-X_i(t))+c_2*r_2*(P_g-X_i(t))$ 是保证 PSO 算法收敛性和全局最优性的关键，下面从 PSO 算法的生物学机理方面来进行分析。

微粒群算法[2]是模拟鸟群、鱼群等群体智能行为并利用生物学家 Frank Heppner 的鸟类模型发展起来的。鸟群等的群体

基金项目：教育部重点科研项目(204018)。

作者简介：杨亚平(1981-)，女，硕士研究生；曾建潮(1963-)，男，教授，博士，主要研究领域为系统建模与仿真、智能计算与 Petri 网理论、系统工程理论与实践、现代集成制造系统与分步计算技术与环境等。

智能行为主要表现在：

(1)鸟类在飞行时尽量飞离最近的个体，以避免相互碰撞；

(2)鸟类根据自身的飞行经验及群体的飞行经验来确定自己的飞行速度和飞行方向；

(3)当一只鸟飞到栖息地，将吸引其他鸟也飞向栖息地。

James Kennedy 和 Russel Eberhart 根据上述原则构造的微粒进化方程中，将栖息地作为群体经历的最好位置 P_g，并引入微粒所经历的最好位置 P_i，微粒向 $(P_g-X_i(t))$ 和 $(P_i-X_i(t))$ 的随机加权处依照惯性方式飞行。

标准微粒群算法从如下几个方面模拟了鸟群等的群体智能行为：

(1)$c_1*r_1*(P_i-X_i(t))+c_2*r_2*(P_g-X_i(t))$ 反映了微粒自身认知能力和社会信息共享能力，反映了鸟类在飞向栖息地的同时避免相互碰撞。

(2)$c_1*r_1*(P_i-X_i(t))+c_2*r_2*(P_g-X_i(t))$ 是一个随机函数，满足了随机性。

(3)微粒根据自身位置与 P_i、P_g 来确定飞行方向和飞行速度。飞行方向由 $(P_g-X_i(t))$ 和 $(P_i-X_i(t))$ 的随机加权值来确定。

(4)固定 P_g 不变，则当 $t\to\infty$ 时，$c_1*r_1*(P_i-X_i(t))+c_2*r_2*(P_g-X_i(t))\to 0$，这就保证了当一只鸟飞到栖息地时，将吸引其他鸟也飞向栖息地。

3 二次微粒群算法

只要能够满足上述原则，就能够模拟鸟群的群体智能行为。可以设想在遵循上述原则的情况下构造不同形式飞行方式，考虑将式(1)中的第二部分和第三部分引入平方项，即以

$$V_i(t+1)=w*V_i(t)+c_1*r_1*\mathrm{sign}(P_i-X_i(t))*(P_i-X_i(t))^2+c_2*\mathrm{sign}(P_g-X_i(t))*(P_g-X_i(t))^2 \quad (3)$$

$$X_i(t+1)=X_i(t)+V_i(t+1) \quad (4)$$

为进化方程。引入平方项后，当 $|P_i-X_i(t)|$ 或 $|P_g-X_i(t)|<1$ 时，可以使收敛速度加快；当 $|P_i-X_i(t)|$ 或 $|P_g-X_i(t)|>1$ 时，可以提高种群的多样性，增强全局寻优能力。定义

$$R(X_i(t),P_i,P_g)=c_1*r_1*\mathrm{sign}(P_i-X_i(t))*(P_i-X_i(t))^2+c_2*r_2*\mathrm{sign}(P_g-X_i(t))*(P_g-X_i(t))^2$$

现在考虑以式(3)、(4)为进化方程的可行性。

(1)固定 P_g 不变，则当 $t\to\infty$ 时，$R(X_i(t),P_i,P_g)\to 0$，这就保证了当 $P_g=X^*$（即最优解）时，所有微粒将最终飞向 X^*，也就是说最终收敛于全局最优点；

(2)$R(X_i(t),P_i,P_g)$ 为一随机函数，这就保证了随机性；

(3)$R(X_i(t),P_i,P_g)$ 由两部分组成，第一部分反映了微粒自身的认知能力，第二部分反映了社会信息共享，这就反映了鸟类飞向栖息地的同时避免相互碰撞。$R(X_i(t),P_i,P_g)$ 平衡了全局勘探和局部开采，从而在收敛速度与最优性等方面达到有效的平衡；

(4)微粒根据自身位置与 P_i、P_g 确定飞行方向和飞行速度。

由此可见，以式(3)、(4)为进化方程的微粒群算法同样能够模拟鸟类的群体智能行为，根据进化方程的形式特点，本文称之为二次微粒群算法。

4 二次微粒群算法的参数自适应

标准微粒群算法的性能受其主要几个参数的影响很大，研究表明标准微粒群算法的参数总取固定值并不总是最好[5]。二次微粒群算法的性能也受其几个主要参数影响，从微粒进化方程中可以看出 c_1 是调节微粒飞向自身最好位置方向的步长，c_2 是调节微粒向全局最好位置飞行的步长，由于每代每个微粒的状态不同，所以参数 c_1、c_2 取为固定值并不合适。在此考虑根据每代每个微粒的状态来动态调整参数的取值。

为了提高 PSO 算法收敛的全局性，保证微粒的多样性是其关键。群体缺乏多样性往往使微粒陷入局部搜索，导致出现收敛到局部最优解的“早熟”现象。近来，有很多有关提高种群多样性的报道，如文献[6,9]等。下面研究参数取值对多样性及收敛速度的影响。

在基于群体的优化算法中，控制全局勘探和局部开采的比例关系是有效寻找全局最优点的关键，通常期望在搜索的早期阶段微粒尽可能地分布在整个搜索空间中，而不是聚集在局部最优解的附近；而在搜索的后期阶段，应尽可能地加快收敛速度来有效地找到全局最优点。也就是说，在搜索的早期阶段，应提高微粒的全局搜索能力；而在搜索的后期阶段，应提高微粒向全局最优点收敛的能力。而当微粒的自身认知能力相对较强时，将会导致微粒过多地徘徊在搜索空间中，相反，社会能力相对较强时将会导致早熟。因此，在搜索的早期阶段，应加强微粒的认知能力；在搜索的后期阶段，应加强微粒的社会能力。考虑到这些，本文采用随 t 的增大来改变参数的自适应策略，目的是在搜索的早期阶段提高全局搜索能力，在搜索的后期阶段加快向全局最好点收敛的速度。在本文中，随 t 的增大，c_1 减小；随 t 的增大，c_2 增大。即在搜索的早期阶段，具有较大的认知能力和较小的社会能力使得微粒尽可能地分布于整个搜索空间而不是移向全局最好点；相反，在搜索的晚期阶段，具有较小的认知能力和较大的社会能力，使得微粒收敛于全局最好点。

另外，考虑进化方程(3)与(1)，不同之处就在于引入了平方项，现在考虑引入平方项后的影响。很显然，引入平方项后，当 $|P_i-X_i(t)|$（或 $|P_g-X_i(t)|$）<1 时，使得收敛速度加快，容易陷入局部最优点；当 $|P_i-X_i(t)|$（或 $|P_g-X_i(t)|$）>1 时，收敛速度变慢，提高了种群多样性，增强了全局寻优能力。在搜索的早期阶段，$|P_i-X_i(t)|$（或 $|P_g-X_i(t)|$）>1 时，提高了多样性；在搜索的后期阶段，$|P_i-X_i(t)|$（或 $|P_g-X_i(t)|$）<1 时，收敛速度加快，这是引入平方项后的优点。在搜索的早期阶段，$|P_i-X_i(t)|$（或 $|P_g-X_i(t)|$）<1 时，降低了多样性；在搜索的后期阶段，$|P_i-X_i(t)|$（或 $|P_g-X_i(t)|$）>1 时，收敛速度变慢，这是引入平方项后的负面影响。我们应该尽可能地利用引入平方项的优点而尽可能地降低其负面影响，本文通过适当地调节 c_1、c_2 来降低引入平方项的负面影响。在搜索的早期阶段，$|P_i-X_i(t)|$（或 $|P_g-X_i(t)|$）<1 时，引入平方项后更小，降低了多样性，所以应该适当地增大 c_1（或 c_2）来降低引入平方项后的影响，增大的幅度由 $|P_i-X_i(t)|$（或 $|P_g-X_i(t)|$）和 t 来决定：$|P_i-X_i(t)|$（或 $|P_g-X_i(t)|$）越小，增大的幅度越大，在 $|P_i-X_i(t)|$（或 $|P_g-X_i(t)|$）一定时，t 越小，c_1（或 c_2）增大的幅度越大；在搜索的后期阶段，$|P_i-X_i(t)|$（或 $|P_g-X_i(t)|$）>1 时，引入平方项后变得更大，相比未引入平方项时减慢了收敛速度，故应该适当地减小 c_1（或 c_2）来减小引入平方项后的负面影响，减小的

幅度由$|P_i-X_i(t)|$(或$|P_g-X_i(t)|$)和t的大小来决定:$|P_i-X_i(t)|$(或$|P_g-X_i(t)|$)越大,减小的幅度越大,在$|P_i-X_i(t)|$(或$|P_g-X_i(t)|$)一定时,t越大,c_1(或c_2)减小的幅度越大。本文中,在搜索的早期阶段,$|P_i-X_i(t)|$(或$|P_g-X_i(t)|$)<1时,c_1由公式

$$c_1=c_{1initial}-\frac{t}{t_{\max}}*(c_{1initial}-c_{1end})+2*(1-|P_i-X_i(t)|)*(1-\frac{t}{t_{\max}}) \quad (5)$$

来确定大小,c_2由公式

$$c_2=c_{2initial}+\frac{t}{t_{\max}}*(c_{2end}-c_{2initial})+2*(1-|P_g-X_i(t)|)*(1-\frac{t}{t_{\max}}) \quad (6)$$

来确定大小,上两式中前两部分表示c_1(或c_2)随t线性递变,第三部分表示在递变的基础上根据$|P_i-X_i(t)|$和t的大小来调节c_1(或c_2)的调节量,由于在早期阶段应该提高多样性,所以应给c_1(或c_2)加上适当的调节量,调节量的大小随$|P_i-X_i(t)|$的减小而增大,随t的增大而减小,从而降低了引入平方项后的负面影响,这与上面的分析是一致的;当$|P_i-X_i(t)|$(或$|P_g-X_i(t)|$)>1时,c_1由公式

$$c_1=c_{1initial}-\frac{t}{t_{\max}}*(c_{1initial}-c_{1end}) \quad (7)$$

来确定大小,c_2由公式

$$c_2=c_{2initial}+\frac{t}{t_{\max}}*(c_{2end}-c_{2initial}) \quad (8)$$

来确定大小,上两式表示c_1(或c_2)只随t线性递变,由于$|P_i-X_i(t)|$(或$|P_g-X_i(t)|$)>1,所以引入平方项后提高了多样性,所以无需再进一步调节c_1(或c_2);而在搜索的后期阶段,当$|P_i-X_i(t)|$(或$|P_g-X_i(t)|$)>1时,c_1由公式

$$c_1=c_{1initial}-\frac{t}{t_{\max}}*(c_{1initial}-c_{1end})-2*\frac{P_i-X_i(t)}{X_{MAX}-X_{MIN}}*\frac{t}{t_{\max}} \quad (9)$$

来确定大小,c_2由公式

$$c_2=c_{2initial}+\frac{t}{t_{\max}}*(c_{2end}-c_{2initial})-2*\frac{P_g-X_i(t)}{X_{MAX}-X_{MIN}}*\frac{t}{t_{\max}} \quad (10)$$

来确定大小,上两式的前两部分表示c_1(或c_2)随t线性递变,第三部分表示根据$|P_i-X_i(t)|$(或$|P_g-X_i(t)|$)和t的大小来调节c_1(或c_2)的调节量,由于在此搜索阶段应该加快收敛速度,而当$|P_i-X_i(t)|$(或$|P_g-X_i(t)|$)>1时,平方项的引入减慢了收敛速度,故应该适当减小c_1(或c_2)来降低引入平方项后的负面影响,减小的这部分调节量的大小由$|P_i-X_i(t)|$(或$|P_g-X_i(t)|$)和t的大小来确定,调节量的大小随$|P_i-X_i(t)|$(或$|P_g-X_i(t)|$)增大而减小,随t增大而增大,从而降低了引入平方项后的负面影响,这也满足上面的分析;当$|P_i-X_i(t)|$(或$|P_g-X_i(t)|$)<1时,c_1由公式(7)来确定大小,c_2由公式(8)来确定大小,这是因为在搜索的后期阶段,$|P_i-X_i(t)|$(或$|P_g-X_i(t)|$)<1时,引入平方项后更小,从而相比较未引入平方项时加速了收敛速度,是引入平方项后的优点,故无需再调节c_1(或c_2)的大小。其中,$t_{\max}$为最大代数,$(c_{1initial},c_{1end})$为c_1的递减范围,$(c_{2initial},c_{2end})$为c_2的递增范围,(X_{MIN},X_{MAX})为搜索空间。

综上,本文所构造的参数自适应的公式与理论上分析的参数变化趋势是一致的,故认为本文中构造的参数自适应公式是可行的。

5 仿真与结果

在评价一个进化优化算法的性能时,经常采用一些著名的BenchMark测试函数,下面采用4个常用测试函数进行仿真。以下仿真中,群体规模均为80,最大代数为1 000,误差为0.01。

5.1 测试函数

(1)Schwefel's Problem 2.22

$$f(x)=\sum_{i=1}^{n}|x_i|+\prod_{i=1}^{n}|x_i| \quad \text{其中 } x_i\in[-10,10]$$

(2)Griewank 函数

$$f(x)=\frac{1}{4\,000}*\sum_{i=1}^{n}x_i^2-\prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}})+1 \quad \text{其中 } x_i\in[-10,10]$$

(3)Ackley 函数

$$f(x)=-20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2})-\exp(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i))+20+e \quad \text{其中 } x_i\in[-30,30]$$

(4)Rastrigin 函数

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

5.2 仿真结果

以下仿真中,群体规模均为80,最大代数为1 000,误差为0.01。

对于标准微粒群算法,参数选为:$c_1=c_2=1.8$,w从0.9到0.4线性递减。结果如表1所示。

表1 标准微粒群算法的结果

函数	维数	收敛率	平均收敛代数	平均最好值
Schwefel's Problem	10	50/50	394.620	5.891 03e-015
	20	50/50	420.760	7.583 27e-006
	30	7/50	799.571	0.162 805
Griewank	10	0/50	-	0.081 282
	20	15/50	288.600	0.027 779
	30	26/50	572.962	0.018 513
Ackley	10	50/50	420.320	3.171 54e-006
	20	30/50	419.067	0.602 618
	30	2/50	782.000	1.659 090
Rastrigin	10	0/50	-	5.770 760
	20	0/50	-	33.649 500
	30	0/50	-	78.589 600

二次微粒群算法几组固定参数的仿真结果如表2所示,第一组参数:$c_1=7.5$,$c_2=2.5$,$w=1.0$;第二组参数:$c_1=5.0$,$c_2=5.0$,$w=1.0$;第三组参数:$c_1=2.0$,$c_2=2.0$,$w=1.0$。

比较表2的结果,总体来讲第一组参数的结果较理想,收敛率高且平均收敛代数小,故二次微粒群算法的参数选为$c_1=7.5$,$c_2=2.5$,$w=1.0$。用这一组参数的结果与标准微粒群算法的结果相比较,显然,二次微粒群算法的性能优于标准微粒群算法的性能,二次微粒群算法很快收敛到全局最优点,收敛速度明显优于标准微粒群算法。尤其对于标准微粒群算法做的不理想的Rastrigin函数也取得了理想的结果。实验结果表明二次微粒群算法能更快更好地找到全局最优点,说明了这种算法的正确性和有效性。

参数自适应,第一组参数:$c_{1initial}=7.5$,$c_{1end}=2.0$,$c_{2initial}=0.5$,$c_{2end}=2.5$,$w=1.0$;第二组参数:$c_{1initial}=9.5$,$c_{1end}=4.5$,$c_{2initial}=1.5$,$c_{2end}=4.0$,$w=1.0$;第三组参数:$c_{1initial}=4.5$,$c_{1end}=2.0$,$c_{2initial}=0.5$,

c_{2end}=3.0,w=1.0。结果如表 3 所示。

表 2　固定参数的仿真结果

函数	维数	第一组参数		第二组参数		第三组参数	
		收敛率	平均收敛代数	收敛率	平均收敛代数	收敛率	平均收敛代数
Schwefel's Problem	10	50/50	5.680	49/50	19.612	50/50	13.460
	20	50/50	9.280	48/50	27.417	49/50	12.347
	30	50/50	35.260	34/50	104.971	26/50	76.577
Griewank	10	50/50	20.980	50/50	47.860	50/50	43.220
	20	39/50	78.051	29/50	104.386	16/50	47.875
	30	35/50	161.429	13/50	255.231	10/50	51.000
Ackley	10	49/50	33.510	49/50	34.082	48/50	34.438
	20	50/50	47.260	49/50	127.653	49/50	62.184
	30	45/50	358.311	33/50	296.939	39/50	214.795
Rastrigin	10	50/50	6.560	50/50	11.140	50/50	11.100
	20	50/50	9.980	49/50	23.796	45/50	37.800
	30	50/50	31.240	36/50	87.444	37/50	79.108

表 3　参数自适应的仿真结果

函数	维数	第一组参数		第二组参数		第三组参数	
		收敛率	平均收敛代数	收敛率	平均收敛代数	收敛率	平均收敛代数
Schwefel's Problem	10	50/50	6.42	50/50	6.620	50/50	6.800
	20	50/50	7.02	50/50	7.820	50/50	8.140
	30	50/50	22.54	50/50	25.100	50/50	23.540
Griewank	10	50/50	12.02	50/50	13.680	50/50	18.480
	20	50/50	21.20	42/50	56.381	43/50	37.326
	30	48/50	109.75	31/50	123.097	27/50	96.778
Ackley	10	50/50	18.14	49/50	18.959	48/50	15.750
	20	50/50	16.06	50/50	41.620	50/50	29.740
	30	50/50	95.42	50/50	218.100	50/50	92.020
Rastrigin	10	50/50	7.34	50/50	8.040	50/50	7.760
	20	50/50	9.26	50/50	9.900	50/50	11.000
	30	50/50	17.64	50/50	25.500	50/50	21.320

比较表 3 的结果，可以看出第一组参数的结果较好，用这组参数值与参数固定时的第一组参数的结果相比较，只有对于参数固定时做的比较理想的 Schwefel's Problem 和 Rastrigin 函数，在 10 维的时候参数自适应的平均收敛代数稍微大于参数固定时的结果，其他时候参数自适应时结果明显好于参数固定时的结果，即总体来说参数自适应时比参数固定时性能有很大提高。这说明根据每代每个微粒所处的状态(离自身历史最好位置及全局历史最好位置的远近)及搜索阶段(搜索的早期阶段或后期阶段)来动态调整参数的方法是可行的。

c_1、c_2采用自适应方案，w 取不同固定值时的结果如图 1-图 4 所示，其中纵坐标 t 表示平均收敛代数，横坐标表示 w 的取值，收敛率均为 100%。

从图 1 中可看出，Schwefel's Problem 函数在 $w\in(1.0,5.2)$ 内均能取得理想的结果，尤其在 $w\in(1.4,4.8)$ 内时很快就能收敛到全局最优点；从图 2 中可看出，Griewank 函数在 $w\in(1.2,3.5)$ 时能取得理想的结果，尤其在 $w\in(1.5,3.5)$ 时能快速收敛到全局最优点；从图 3 中可看出 Ackley 函数的 w 选择范围较大，在 $w\in(1,13)$ 内均能取得很好的结果，尤其在(1.3,13.0)内在 10 代以内就收敛到全局最优点；从图 3、4 中可看出，Griewank 函数和 Rastrigin 函数的 w 范围较小，w 在(1.0,2.4)内能取得理想的结果，在 $w\in(1.5,2.0)$ 内能快速收敛于全局最优点。总体来说，w 在(1.5,2.0)内取值时效果比较稳定，影响不大。

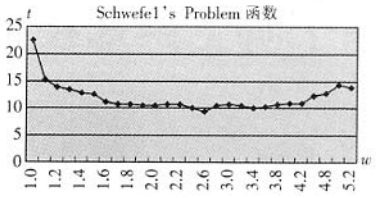

图 1　Schwefel's Problem 函数的仿真结果

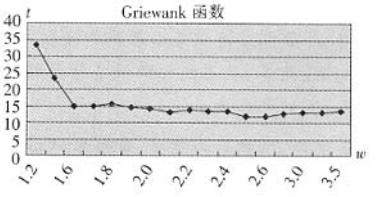

图 2　Griewank 函数的仿真结果

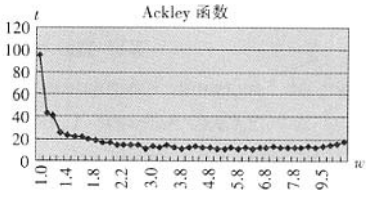

图 3　Ackley 函数的仿真结果

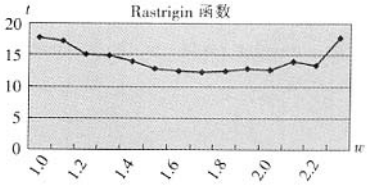

图 4　Rastrigin 函数的仿真结果

6　结语

此文在对标准微粒群算法进行分析的基础上提出了一种二次微粒群算法，并在对二次微粒群和标准微粒群算法进行比较分析的基础上给出了一种二次微粒群算法参数的自适应策略，其次，分别用微粒群算法、二次微粒群算法、采用参数自适应的二次微粒群算法来对典型测试函数进行仿真，并对这几种方法的结果进行比较，结果说明二次微粒群算法比标准微粒群算法的性能有很大提高，表明了二次微粒群算法是可行的，而采用参数自适应比参数固定时性能有很大提高，说明了这种参数自适应策略的正确性和有效性。

此文从实验的角度给出了参数的变化趋势及选择范围，从理论的角度来给出参数的选择范围是作者下一步要进行的研究。(收稿日期：2006 年 1 月)

参考文献：

[1] 曾建潮，介婧，崔志华.微粒群算法[M].北京：科学出版社，2004.

[2] KENNEDY J，EBERHART R.Particle swarm optimization[C]//Proc IEEE Int Conf Neural Networks，1995：1942-1948.

[3] 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.

(下转 79 页)

出变为有效之间的最大时间。第五个重要参数是 t_{WLD}，是获得端写门限由高电平变为低电平和数据在收到忙信号的端口输出变为有效之间的最大时间。

3.2 存储器高速刷新功能

对于图 2，从表面上分析，t_{ABH} 是无用的，其对 t_{WDV} 和 t_{WLD} 的长度有影响。存储器采用了以下的策略：以计算机 1 对 U_{11} 某一单元写一个数据开始，到其他计算机从 U_{i1} 的相应单元读取数据作为一个刷新周期，图 3 表示了 U_{11} 左端口执行写操作的时序，右端口异步执行读操作时序和 U_{i1} 右端口为执行写操作异步导向同一地址时序。

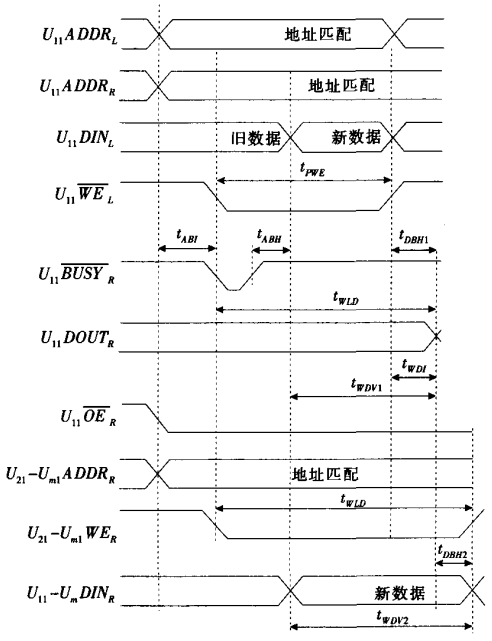

图 3 双端口 RAM 时序

当失去端得知其失去竞争时，它的忙信号立刻变为高电平。地址总线 $U_{11}ADDR_R$ 和需要刷新的 U_{21}~U_{m1} 的右端口的 U_{21}-$U_{2n}ADDR_R$ 在 T_1 时刻分别是开启和锁闭的。U_{11} 的控制总线 $U_{11}OE_R$ 和 U_{21}~U_{m1} 的控制总线 WE_R 在 T_2 时刻是开启的，并且等待更新数据传送。此时，数据稳定性控制单元也被激活。在图 3 中，t_{WDV1} 是从 U_{11} 左端口写入数据到此数据在接收端口有效的最大时间。t_{WDV1} 可由以下公式解释：

$$t_{WDV1}=t_{WDV}-t_{ABH} \tag{1}$$

t_{WDV2} 是从 U_{11} 左端口写入稳定性数据到此数据在接收更新数据的 U_{i1} 有效的最大时间，可由以下公式解释：

$$t_{WDV2}=t_{WDV1}+t_{DBH2}\approx t_{WDV} \tag{2}$$

因为存储器采用硬件刷新电路，数据稳定性控制单元采用旁路刷新，而且实际刷新时间少于 t_{WDV1} 与 t_{WDV2}。所以存储器具有高刷新速度。

4 实验证明

本文采用 5 个普通存储器和本文设计的存储器，证明存储器共享性能。本文设计的存储器可以执行具有复杂数据融合方法（包括 BP 网络、模糊逻辑、D-S 证据理论和其他信息）的传动变压器多故障诊断功能。表 1 列出了运行时间测试结果。从表中可以看到，相比普通的共享存储器而言，此存储器可以极大增强计算机和处理器之间的通信能力。

表 1 运行时间测试结果

方法	计算机	
	普通共享存储器	本文设计的存储器
模糊逻辑	21.56 min	14.21 min
D-S 理论	18.44 min	12.65 min
BP 网络	34.87 min	25.43 min

5 结论

本文设计的多端口存储器以多处理器和计算机实时控制系统的分析与双端口 RAM 为基础。在处理器与计算机之间具有强共享能力、高刷新速度和弱冲突访问等特点。理论和实验结果证明，此存储器可以极大增强计算机和处理器之间的通信能力。（收稿日期：2006 年 1 月）

参考文献：

[1] TOCCI R J，AMBROSIO F J 著.微处理器与微型计算机：硬件和软件[M].北京：清华大学出版社，2004.

[2] TINDELL K，CLARK J.Holistic schedulability analysis for distributed hard real-time systems[J].Micro-processing and Micro-programming，1994，50(2)：117-134.

[3] WU M Y.On runtime parallel scheduling for processor load balancing[J].IEEE Transactions on Parallel and Distributed Systems，1997，8(2)：173-185.

[4] CAREY M，LIVNY M.Distributed concurrency control performance：a study of algorithms，distribution and replication[C]//Proc of 4th Intl Conf on Very Large Databases，1998.

（上接 67 页）

[4] KENNEDY J，EBERHART R.Parameter selection in particle swarm optimization[C]//Lecture Notes in Computer Science—Evolutionary Programming VII，Proc 7th Int Conf Evolutionary Programming，1998，1447：591-600.

[5] SUGANTHAN P N.Particle swarm optimizer with neighborhood operator[C]//Proc IEEE Int Congr Evolutionary Computation，1999，3：1958-1962.

[6] XIE X F，ZHANG W J，YANG Z L.A dissipative particle swarm optimization[C]//Proc IEEE Congr Evolutionary Computation 2002. Honolulu，HI：[s.n.]，2002，2：1456-1461.

[7] CLERC M.The swarm and the queen：toward a deterministic and adaptive particle swarm optimization[C]//Proc of the Congress on Evolutionary Computation.Washington，DC：[s.n.]，1999：1951-1957

[8] LOVBHERG M，RASMUSSEN T K.Hybrid particle swarm optimizer with breeding and subpopulation[C]//Proc of the Genetic and Evolutionary Computation Conference.San Fracdisco，USA：[s.n.]，2001.

[9] LOVBJERG M，KRINK T.Extending particle swarm optimizers with self-organized critically[C]//Proc IEEE Int Congr Evolutionary Computation.Honolulu，HI：[s.n.]，2002，2：1588-1593.

二次微粒群算法及其参数自适应策略

作者： 杨亚平， 谭瑛， 曾建潮， YANG Ya-ping， TAN Ying， ZENG Jian-chao
作者单位： 太原科技大学,系统仿真与计算机应用研究所,太原,030024
刊名： 计算机工程与应用 ISTIC PKU
英文刊名： COMPUTER ENGINEERING AND APPLICATIONS
年，卷(期)： 2006,42(31)
被引用次数： 2次

参考文献(9条)

1. RATNAWEERA A;HALGAMUGE S K;WATSON H C Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients 2004(03)

2. KENNEDY J;EBERHART R Particle swarm optimization[外文会议] 1995

3. 曾建潮;介倩;崔志华 微粒群算法 2004

4. LOVBJERG M;KRINK T Extending particle swarm optimizers with self-organized critically 2002

5. LOVBHERG M;RASMUSSEN T K Hybrid particle swarm optimizer with breeding and subpopulation 2001

6. CLERC M The swarm and the queen:toward a deterministic and adaptive particle swarm optimization[外文会议] 1999

7. XIE X F;ZHANG W J;YANG Z L A dissipative particle swarm optimization[外文会议] 2002

8. SUGANTHAN P N Particle swarm optimizer with neighborhood operator 1999

9. KENNEDY J;EBERHART R Parameter selection in particle swarm optimization 1998

引证文献(3条)

1. 姜慧霖 基于粒群优化模糊PID的温控系统[期刊论文]-商丘师范学院学报 2010(12)

2. 曾渊.宋涛.王少波.许家栋 多阶段参数动态控制微粒群优化算法[期刊论文]-计算机工程与应用 2007(30)

3. 姜慧霖 基于粒群优化模糊PID的温控系统[期刊论文]-商丘师范学院学报 2010(12)