南 昌 工 程 学 院

**毕 业 设 计 (论 文)**

信息工程学院 系（院） 计算机科学与技术 专业

毕业设计（论文）题目分段模型粒子群优化算法的多样性分析

学生姓名 苏远程

班 级 08计算机科学与技术

学 号 2008100855

指导教师 孙辉

完成日期 年 月 日

**分段模型粒子群优化算法的多样性分析**

**The diversity analysis of segmented model of particle swarm optimization algorithm**

总计 毕业设计（论文） 23 页

表 格 2 个

插 图 8 幅

# 摘要

基于标准微粒群优化算法容易出现早熟而陷入局部最优的陷阱之中，通过对微粒群优化算法的多样性分析对微粒群算法进行改进，提出了分段微粒群算法，在各阶段采用不同的进化模型。第一阶段，利用标准微粒群算法进行探索；第二阶段，利用局部搜索方法进行开发，更快地找到极值;第三阶段，提出一种改进方法增加种群多样性使粒子跳出局部极值点，进行全局最优搜索。经过实验测试该算法在搜索全局最优上效果有所提高。

**关键词：**标准微粒群算法 多样性分析 分段微粒群算法

# Abstract

Fall into the trap of local optimum based on the standard particle swarm optimization algorithm is prone to precocious .Through the diversity analysis of particle swarm optimization algorithm to improve particle swarm optimization, proposed a segmented model of particle swarm optimization, in various stages of different evolutionary model. The first stage, using a standard particle swarm algorithm to explore; the second stage, using of local search methods to develop, faster to find the extremum; the third stage, an improved method to increase the diversity of the population so that the particles escape from the local extreme points and search global optimum. Improved after the effect of the experimental test of the algorithm in the search for global optimum.

**Key Words:** standard particle swarm optimization ;diversity analysis ; segmented model of particle swarm optimization

目录

[摘要 I](#_Toc326784473)

[Abstract II](#_Toc326784474)

[第一章 引言 1](#_Toc326784475)

[1.1微粒群算法[11] 1](#_Toc326784476)

[1.2 基本微粒群算法的改进[11] 1](#_Toc326784477)

[第二章 基本微粒群算法与改进微粒群算法 3](#_Toc326784478)

[2.1 基本微粒群算法[11] 3](#_Toc326784479)

[2.2 基本微粒群算法的算法流程 4](#_Toc326784480)

[2.3基本微粒群算法的分析 5](#_Toc326784481)

[2.4改进的微粒群算法 5](#_Toc326784482)

[2.4.1 带惯性因子的改进微粒群算法 6](#_Toc326784483)

[2.4.2带有收缩因子的微粒群算法 6](#_Toc326784484)

[第三章 分段微粒群算法 8](#_Toc326784485)

[3.1分段微粒群算法原理 8](#_Toc326784486)

[3.2 分段微粒群算法分析 9](#_Toc326784487)

[3.3 分段微粒群算法流程 10](#_Toc326784488)

[第四章 仿真实验 12](#_Toc326784489)

[4.1测试函数 12](#_Toc326784490)

[4.2 实验参数的设置 13](#_Toc326784491)

[4.2.1 标准微粒群算法参数设置 13](#_Toc326784492)

[4.2.2 分段微粒群算法参数设置 13](#_Toc326784493)

[4.3 实验结果及分析 14](#_Toc326784494)

[第五章 分段微粒群算法的多样性分析 16](#_Toc326784495)

[5.1多样性 16](#_Toc326784496)

[5.2 分段微粒群算法的多样性分析 17](#_Toc326784497)

[第六章 结束语 21](#_Toc326784498)

[参考文献 22](#_Toc326784499)

[致谢 23](#_Toc326784500)

# 第一章 引言

## 1.1微粒群算法[11]

微粒群算法（pso）最早是在1995年由美国社会心理学家James Kennedy和电气工程师Russell Eberhart共同提出的，其基本思想是受他们早期对许多鸟类的群体行为进行建模与仿真研究的结果的启发。当一只鸟飞离鸟群飞向栖息地时，将导致它周围的其他的鸟也飞向栖息地。这些鸟一旦发现栖息地，将会降落在此，吸引更多的鸟落在栖息地，直到整个鸟群都停留在栖息地。鸟类寻找栖息地与对一个特定问题寻找解很类似，已经找到栖息地的鸟将引导它周围的其他鸟飞向栖息地，增加了鸟群找到栖息地可能性。在特定问题寻找解的过程中也可以采用这种思想，特别是十分复杂的问题，这样可能增加寻找最好解的可能性。

## 1.2 基本微粒群算法的改进[11]

为了提高算法的收敛性能，Shi和Eberhart于1998年又对pso算法的速度项引入了惯性权重ω并提出了在进化过程中动态调整惯性权重以平衡收敛全局和收敛速度，这种方程已经被相关学者称为标准pso算法。Clerc又于1999年在进化方程中引入收缩因子以保证算法的收敛性，同时使得速度的限制放松。

Angeline于1999年借鉴进化计算中的选择概念，将其引入pso算法中。通过比较各个微粒的适应值淘汰掉差的粒子，而将具有更好适应值的粒子进行复制以产生等数额的粒子来提高算法的收敛性。Lovbjerg等人进一步将进化计算应用于pso算法，如复制，交叉等，给算法交叉的具体形式，并通过典型测试函数的仿真实验说明了算法的有效性。

为了提高算法的收敛的全局性，保证粒子的多样性是其关键。为了保证进化过程中群体中粒子的多样性，Suganthan在标准pso算法中引入了空间邻域的概念，将处于同一个空间邻域的粒子构成一个粒子群分别进行进化，并随着进化邻域动态地改变选择阈值以保证群体的多样性;Kennedy引入了邻域拓扑的概念来调整邻域间的动态选择，同时引入社会信念将空间邻域与邻域拓扑中的环拓扑相结合增加邻域间的信息交流，提高群体的多样性。Lovbjerg等人于2001年将遗传算法中的子群体概念引入pso算法中，同时引入 了繁殖算子以进行群体信息交流。

在pso算法的行为分析和收敛性分析方面进行了大量的研究工作。首先是采用了代数方法对几种典型的pso算法的运行轨迹进行了分析，给出了保证收敛性的参数选择范围。在收敛性方面，Frans van den Bergh 引用 了Solis和Wets关于随机性算法的收敛准则，证明了标准pso算法不能收敛于全局最优解，甚至于局部最优解。证明了保证收敛的pso算法能够收敛 于局部最优，而不能保证收敛于全局最优解。

在pso算法的应用方面，pso算法最早应用于人工神经网络的训练的方法，Kenedy和Ebrhart 成功地将pso算法应用于XOR问题的神经网络训练。随后，pso算法在函数优化，约束优化。极大极小值问题，多目标优化等问题中均得到了成功的应用。

# 第二章 基本微粒群算法与改进微粒群算法

# 2.1 基本微粒群算法[11]

Kennedy 和Eberhart在1995年的IEEE国际神经网络学术会议上正式发表了题为“Particles Swarm　Optimization”的文章，标志微粒群算法正式诞生。

微粒群算法与其他进化类算法相类似，也采用“群体”与“进化”的概念，同样也是目标函数适应值的大小来进行一些操作的。不同之处在于微粒群算法不像其他进化算法那样对于个体使用进化算子，而是将每个个体看成是在n维搜索空间中的一个没有重量和体积的微粒子，并在搜索空间中以一定的速度飞行，此速度是根据自己历史经验位置与全局最好位置经验进行动态调节粒子位置，对粒子的进化起到至关重要的作用。我们假设：

Xi=(xi1,xi2,…,xin)为粒子i的当前位置向量;

Vi=(vi1,vi2 ,…,vin)为粒子i的当前速度向量;

Pi=(pi1,pi2,…,pin)为粒子所经历的最好位置向量，也就是粒子所经历过的具有最好适应值的位置，例如在最在化问题中，目标函数的适应值越大对应的适应值就越好。以f(X)为最小化目标函数为例，按基本微粒群算法的位置更新策略，则i的当前最好位置由下式确定

Pi(t+1)=min{f(pi(t)),f(Xi(t+1))} (2.1)

设群体中的粒子数为m,群体中所有粒子所经历过的最好位置为Pg（t）,称为全局最好位置，则

Pg(t)∈{P0（t）,P1(t),…,Pm(t)}

f(Pg(t))=min{f(P0（t）),f(P1(t)),…,f(Pm(t))} (2.2)

基本微粒群算法的速度与位置进化方程可描述为：

vij(t+1)=vij(t)+c1\*r2j(t)\*(pij(t)-xij(t))+c2\*r2j\*(pgj(t)-xij(t))

(2.3)

xij(t+1)= xij(t)+ vij(t+1) (2.4)

其中：下标“i”表示第“i”个粒子，“j”表示粒子的第“j”维，t表示第t次迭代，c1,c2为加速因子，通常在0～2之间，r1,r2～U(0,1)为两个相互独立的随机函数。从此式中我们可以看出c1是调节粒子向自身最好位置飞行的步长，而c2则是粒子向全局最优位置飞行的步长。通常为了提高种群的开发性避免飞过搜索可能最好解的空间，把速度vij 限定于一定的范围之内，即vij∈[-vmax,vmax]，我们可以设定vmax=k\*xmax,0.1<k<1.0。

## 2.2 基本微粒群算法的算法流程

1. 对微粒群算法进行初始化。分别对微粒群的位置与速度进行初始化设定，在一定区域里进行初始化利用随机产生方法vij∈[-vmax,vmax]，xi∈[-xmax,xmax]。设置种群规模N,每一粒子的维数Dim,最大迭代次数Tmax.
2. 按目标函数计算微粒的适应值。
3. 对于每一个微粒子，按（2）计算出来的适应值后，利用式（2.1）对单个粒子历史最好位置进行更新。
4. 经过了（3）对单个粒子历史最

（6） 微粒的速度与位置进行更新使得粒子进化。

（7）如果达到结束条件则结束算法（通常结束条件设为迭代次数Tmax）,否则进行入步骤（2）(5)进化。

对种群历史最好位置进行更新。

（5） 根据方程（2.3）和（2.4）对

好位置更新后，再按式（2.2）

基本微粒群算法流程图

开始

计算目标函数适应值

对微粒群进行初始化

结束

更新粒子历史与种群历史最好位置

按（2.3），（2.4）公式进化

是否结束

## 2.3基本微粒群算法的分析

在式（2.3）所描述的粒子速度进化方程中，其中第一部分为粒子的先前速度;其中第二部分称为“自身认知”部分，它是粒子仅根据自身经验来进化的，是粒子自身的文化;第三部分称为“社会”部分，意为种群粒子间的社会文化。

若仅包含第一部分，即

vij(t+1)=vij(t) （2.5）

则表示粒子一直维持先前速度不变一直“飞”下去直到到达边界，这样很难搜索到最好解。若仅包含前两部分，即Cognition-only Model

vij(t+1)=vij(t)+c1\*r2j(t)\*(pij(t)-xij(t)) （2.6）

则其性能会变差许多。主要原因是不同的粒子间缺乏信息的交流，也就是没有社会文化信息的共享，使得每一个粒子都只根据自身的经验进化，按各自的经验飞行，因而算法不易收敛同时想得到最好解的可能性也非常的小。若速度进化方程中仅包含社会部分，即Social\_only Model

vij(t+1)=vij(t) +c2\*r2j\*(pgj(t)-xij(t)) （2.7）

则粒子自身没认知能力，也就是所谓的“只有社会（Social-only）”的模型。在粒子间的相互作用下，有能力到达新的搜索空间，虽然收敛速度比基本微粒群算法的更快，但是对于复杂问题很容易陷入局部最优点。因为若社会最好粒子陷入局部最优点后种群就是受到吸引而全部陷入局部最优点而停止飞行。

在基本微粒群算法中社会部分与自身认知部分都是相当重要的。第一部分的先前速度使得粒子具有全局搜索能力，第二部分的“自身认知”与第三部分的“社会”部分使得粒子具有局部精细搜索能力。

## 2.4改进的微粒群算法

对于基本微粒群算法而言，速度进化公式的第一部分保证了算法具有一定的全局搜索能力。通过以上分析我们知道第二部分和第三部分使得微粒群算法具有局部搜索能力。基本微粒群算法的单峰，简单问题是性能表现很好，但在一些具有问题中特别是多极值复杂问题中全局搜索与局部搜索之间的平衡对于求解过程是相当重要的。因此，专家学者提出了一些改进算法。下面主要介绍两种改进算法。

### 2.4.1 带惯性因子的改进微粒群算法

Yuhui shi提出了带有惯性权重的改进微粒群算法，其速度与位置进化方程为：

vij(t+1)= ω \*vij(t)+c1\*r2j(t)\*(pij(t)-xij(t))

+c2\*r2j\*(pgj(t)-xij(t))

(2.8)

xij(t+1)= xij(t)+ vij(t+1) (2.9)

我们易知当惯性权重ω=1时，式（2.8）和（2.9）与基本微粒群算法的进化公式就是一样的了。专家学者建议ω的取值范围为[0，1.4]，但是实验结果表明ω取[0.8,1.2]时，算法收付速度更快，当ω>1.2时，算法则较多的陷入局部极值。

惯性权重ω表明微粒子原来速度能在多在程度上得到保留。惯性权重ω类似于模拟退火中的温度，较大的ω有较好的全局收敛能力，而较小的ω则具有较强的局部收敛能力。因此，随着迭代次数的增加，惯性权重ω应不断减少，从而使得微粒群算法在初期具有较强的全局收敛能力，而晚期具有较强的局部收敛能力。因此有学者提出ω应满足

ω（t）=0.9-(t/Tmax)\*0.5 (2.10)

其中t为迭代次数，Tmax为设置最大迭代次数。这样惯性权重ω看作一个线性迭次数的函数，可以多0.9到0.4线性递减。从实验结果来看效果很好,此后专家学者将带惯性因子的微粒群算法称为标准微粒群算法。

### 2.4.2带有收缩因子的微粒群算法

在Clercr 研究中，提出了收缩因子的概念。该方法描述了一种选择ω，c1和c2的值的方法，以确保算法收敛。通过正确地选择这些控制参数，就没有必要将vij  限制在[-vmax,vmax]之中。改进的速度进化方程如下：

vij(t+1)=λ（vij(t)+c1\*r2j(t)\*(pij(t)-xij(t))+c2\*r2j\*(pgj(t)-xij(t))）

（2.11）

其中

λ= （2.12）

且l=c1+c2,l>4。

Eberhgart 和Shi分别利用vmax和收缩因子来控制微粒速度的两种算法性能做了比较，结果表明，后者比前者具有更好的收敛率。但是在测试一些其它函数的求解过程中，使用收缩因子的pso在给定的迭代次数内无法达到全局极值点。按

Eberhgart 和Shi的观点，这是由于微粒偏离所期望的搜索空间太远而造成的。为了降低这种影响，他们建议在使用收缩因子时首先对算法进行限定，比如设参数vmax=xmax，或者预先设置搜索空间的大小。这样可以改进算法对所有测试函数的求解性能。

# 第三章 分段微粒群算法

## 3.1分段微粒群算法原理

上一章我们了解了基本微粒群算法以及一些改进算法包括带惯性因子的微粒群算法（标准微粒群算法）和带收敛因子的微粒群算法，他们对简单，单峰问题上表现很好，但是对于复杂问题是表现并不是很好。本章我们就来讨论一下本文提出的分段微粒群算法。为了便于讨论，我们将标准微粒群算法（pso）的速度与位置的进化公式列出来：

vij(t+1)= ω \*vij(t)+c1\*r2j(t)\*(pij(t)-xij(t))

+c2\*r2j\*(pgj(t)-xij(t))

(3.1)

xij(t+1)= xij(t)+ vij(t+1) (3.2)

其中，ω为惯性权重，也就是决定先前速度在本次进化过程中所占的比例；c1,c2为加速因子一般取c1=c2=2.0，也有专家学者认为c1=c2=1.8效果更好; r1,r2～U(0,1)为两个相互独立的随机函数；pij(t)是i单个粒子在第t代进化中历史最好位置j维的分量；pgj(t)是粒子群在第t代进化中全局历史最好位置j维的分量；惯性权重ω满足ω（t）=0.9-(t/Tmax)\*0.5线性减少 。

由（3.1）和（3.2）我们可以看出粒子飞行的速度就相当于搜索的步长一样，其大小直接影响到搜索的收敛性。由于惯性权重是表示的原来速度的保留程度，所以较大的ω使得速度值较大从而整个算法具有全局搜索能力，较小的ω使得速度值较小因此整个算法具有局部搜索能力。随着迭代次数的增加，惯性权重ω将不断的减小，因此标准微粒群算法在早期是具有较强的全局搜索能力（探测能力）而晚期则具有较强的局部搜索能力（开发能力）。虽然标准微粒群算法（pso）的性能较好，但是在整个迭代过程中会出现早期探测能力较强开发能力弱，而晚期开发能力较强探测能力弱。因此，平衡探测能力与开发能力对算法的性能起着关键作用。若是标准微粒群算法（pso）在早期探测期陷入到局部最优之中， pgj将不会再动飞行速度vij逐渐减小直至0，又随着迭代次数的增加整个算法的开发能力逐渐增加，其它粒子也会逐渐飞向pgj最终会停滞不行，这样整个种群就会趋同停滞在局部极值处不再飞行,最终就导致了“早熟”现象的出现。由于粒子在晚期趋同使得种群多样性减少使得进化方向单一，所以在多极值问题中当粒子一旦陷入局部最优值时会停下来就不能再次探测到新的搜索区域而跳不出局部极值。

基于以上分析，在迭代过程中，本文通过采取分段模型来进化种群增加粒子的多样性使得粒子的探测与开发能力平衡从而更能够寻找最优解。

第一阶段，本文采取标准微粒群算法（pso）进行进化，利用标准微粒群算法（pso）在前期探测能力强速度快的优点。如果若干次迭代后（本文取25次）全局最优解的差小于设置的阈值，本文就认为算法找到了一个局部极值搜索邻域，那么我们就进入下一阶段，否则继续探测。

第二阶段，本文采取精细搜索。基于上文的分析我们知道标准微粒群算法（pso）第二部分与第三部分使得粒子既相互作用又要根据自身认知，所以具有极强的局部搜索能力，所以本文利用公式

vij(t+1)=c1\*r2j(t)\*(pij(t)-xij(t))+c2\*r2j\*(pgj(t)-xij(t)) （3.3）

速度进化公式快速找到局部最优值。

第三阶段，再探测阶段。本文通过向人工蜂群算法的学习，利用其中引领蜂的思想将全局最优值重置，重置全局最优后并利用

pi=pi+rand()\*(pg-pi) (3.4)

将其中的一半粒子重置增加种群的多样性，从而引领其它粒子跳出局部极值邻域再次进行搜索。

## 3.2 分段微粒群算法分析

在本文提出的分段模型粒子群优化算法中。第一阶段为探测期，利用标准微粒群算法（pso）进化；第二阶段为开发期，进行精细搜索，尽快地找到局部极值点；第三阶段为再探测，通过第三阶段使得粒子群多样性增加让粒子能够进入到新的搜索域进行搜索。由标准微粒群算法（pso）速度进化公式（3.1），可知在进化早期具有较强的探测能力也就是具有较强的全局搜索能力能够较快地进化。但是标准微粒群算法的缺点就是随着迭代次数的增加种群多样性逐渐减小，使得粒子的全局搜索能力减小反而增加了局部搜索能力。因此，在多极值复杂问题中只要种群陷入到局部极值点就会自认为找到了全局最优点而停止飞行，这就出现了“早熟”现象，使得算法性能变差。因此本文提出了分段微粒群算法当粒子陷入局部极值点时增加粒子多样性平衡种群探测与开发之间的矛盾。

第一阶段，利用标准微粒群算法（pso）早期全局搜索能力强且收敛速度快的优点，来快速进化种群这样可以保证算法在开始时各粒子都能够以较大的速度步长在全局范围内探测到较好的种子。当进入到第二阶段后，使用（3.3）式来进化，能够保证微粒子能够在局部极值邻域周围快速精细搜索，（3.3）式具有极强的局部搜索能力，利用自身的认知与社会信息共享的相互作用使得种群的精细能力更强，相比于带惯性权重因子的微粒群算法更具精细搜索能力。从第一阶段到第二阶段进化的过程由于搜索域的缩小粒子位置也将更集中，因此是种群的多样性不断减小的过程，会出现与标准微粒群算法（pso）同样的陷入局部极值点很难跳出的缺陷。因此，本文通过向人工蜂群算法（引领蜂——全局搜索；跟随蜂——局部搜索；侦察蜂——跳出局部极值点）学习，在第三阶段将全局最好位置的粒子进行重置，引领其它粒子飞出局部极值点到达新的搜索域，在飞向全局最好位置时粒子群有可能搜索到更好的解同时也增加了搜索空间，这样增加了种群的多样性也就是增加了寻找到全局最优解的可能性。

## 3.3 分段微粒群算法流程

（1） 初始化。设置种群规模num，问题维数Dim，最大迭代次数Tmax，惯性权重最大值和最小值，加速因子c1,c1。初始化各粒子速度与位置，在一定区域里进行初始化，利用随机产生方法使vij∈[-vmax,vmax]，xi∈[-xmax,xmax]。

（2） 计算目标函数适应值，并将初始化位置作为粒子历史最好位置，初始化适应值保留为粒子历史最优值。

（3） 进入第一阶段探测期。按（3.1）和（3.2）式进行粒子速度与位置的更新，并且在更新过程中对速度与位置进行限幅处理，使不得超过搜索域。

（4） 计算目标函数适应值，根据适应值对粒子历史最好位置和最好适应值与全局历史最好位置和最好适应值进行更新。

（5） 比较若干次（本文取25次）迭代后全局最优值的绝对差是否小于设定的阈值，若小于进入步骤（6），否则转向步骤（3）。

（6） 进入第二阶段开发期。按公（3.3）

和（3.2）更新粒子的速度与位置，同时也对速度速度与位置进行限幅。

1. 计算目标函数适应值，并根据目标函数适应值对粒子历史最好位置与最好适应值与全局历史最好位置与最好适应值进行更新。
2. 判断是否到达算法终止条件（达到设置最大迭代次数），若没有则把全局最优解进行重置，即在搜索范围内随机产生一个值，并利用最优位置将其中一半粒子也重置见公式（3.4），进入步骤（3），否则结束程序。

分段微粒群算法流程图

开始

进入第一阶段：标准pso

对微粒群进行初始化

早熟

结束

进入第二阶段：精细搜索

进入第三阶段：重置

是否结束

# 第四章 仿真实验

## 4.1测试函数

F1：

F1是由K.A.DeJong提出的，是著名的Sphere函数，单峰函数，在xi =0时到极小值。

F2：

F2是由Griewank提出的，全局极小值在xi=0（i=1,2,…,n）时到达，局部极小值在xi≈±k\*,i=1,2,…,n；k=0,1,2,…,n。

F3：

F3被称为Rastrigin函数，多峰函数，在xi=0（i=1,…,n）时达到全局极小值。在S=｛xi∈(-5.12,5.12),i=1,2,…,n｝范围内大约存在10n个局部极小点。

F4:

F4被称为Rosenbrock函数，非凸，病态函数，在xi=1（i=1,…,n）处达到极小值，有专家学者认为当问题维数多于5时Rosenbrock是一个多峰函数。

F5：

F5称为Schaffer函数，多峰，全局最小值在xi=0（i=1,…,n）。

## 4.2 实验参数的设置

### 4.2.1 标准微粒群算法参数设置

（1）种群规模为20粒子，问题维数分别为10维和30维，程序运行500次；

（2）粒子最大速度控制为vmax=Xmax；

（3）ω惯性权重按ω（t）=0.9-(t/Tmax)\*0.5线性递减；

（4）加速因子c1=c2=2.0。

### 4.2.2 分段微粒群算法参数设置

（1）种群规模为20粒子，问题维数分别为10维和30维，程度运行500次；

（2）粒子最大速度控制为vmax=Xmax；

（3）ω惯性权重按ω（t）=0.9-(t/Tmax)\*0.5线性递减；

（4）加速因子c1=c2=2.0；

（5）利用（abs(tempGlbest-glbest)/glbest）来判断是否进入了局部极值邻域，其中,tempGlbest是进化前的全局最优值，glbest是进化若干代后的全局最优值。

## 4.3 实验结果及分析

表4.1（问题维数10，种群规模20）

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 函数名 | 搜索域 | 理论最小值 | 精确度 | 实验值（平均值） | | 最优值 | | 达优率（%） | |
| 标准pso | 分段pso | 标准pso | 分段pso | 标准pso | 分段pso |
| Sphere | [-100,100] | 0 |  | 0 | 0 | 0 | 0 | 100 | 100 |
| Griewank | [-600,600] | 0 |  | 0.085233 | 0.057300 | 0.000000 | 0.000000 | 0.2 | 4.2 |
| Rastrigrin | [-5.12,5.12] | 0 |  | 3.487759 | 1.5680592 | 0.000000 | 0.000000 | 1.6 | 16.6 |
| Rosenbrock | [-10,10] | 0 |  | 3.648086 | 2.860178 | 0.005501 | 0.000000 | 12 | 22.4 |
| Schaffer | [-100,100] | 0 |  | 0.303665 | 0.049236 | 0.000000 | 0.000000 | 19.4 | 57.2 |

由表4.1可知，在种群规模20，问题维数为10等相同环境条件下，对于简单单峰函数Sphere，分段pso与标准pso的结果相差无几都能够收敛到全局最优点且达优率高达100%，说明分段pso对于标准pso在优化单峰函数的优势保持了下来。对于特殊单峰函数Rosenbrock的实验中，分段pso在实验值（平均值），最优值和达优率方面都优于标准pso，特别是最优值方面分段pso能够达到0.00000而标准pso仅能到0.005501。在多峰函数Griewank,Rastrigin ，Schaffer的实验中，我们可知实验值（平均值）方面分段pso比标准pso都有一定的改善，特别是Schaffer函数的实验值分段pso比标准pso提高了近一个数量级，而且在三个函数达优率方面分段pso也都均优于标准pso，尤其是在Rastrigin和Schaffer两个函数的达优率分段pso比标准pso有明显的提高。

表4.2（问题维数30，种群规模20）

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 函数名 | 搜索域 | 理论最小值 | 精确度 | 实验值（平均值） | | 最优值 | | 达优率（%） | |
| 标准pso | 分段pso | 标准pso | 分段pso | 标准pso | 分段pso |
| Sphere | [-100,100] | 0 |  | 0 | 0 | 0 | 0 | 100 | 100 |
| Griewank | [-600,600] | 0 |  | 0.016612 | 0.012074 | 0.000000 | 0.000000 | 32.8 | 36.2 |
| Rastrigrin | [-5.12,5.12] | 0 |  | 45.606778 | 30.045949 | 19．89915 | 9.968840 | 0 | 0.2 |
| Rosenbrock | [-10,10] | 0 |  | 47.102955 | 44.710745 | 0.086226 | 0.000292 | 11.8 | 24 |
| Schaffer | [-100,100] | 0 |  | 5.166258 | 3.2061498 | 0.0770581 | 0.043484 | 13.4 | 25.8 |

由表4.2可知，在种群规模20，问题维数为30等相同环境条件下，对于简单单峰函数Sphere，分段pso与标准pso的结果相差无几还是都能够收敛到全局最优点且达优率高达100%，说明分段pso对于标准pso在优化单峰函数在提高问题维数的情况下还是能的保持优势。对于病态单峰函数Rosenbrock的实验中，分段pso在实验值（平均值），最优值和达优率方面都优于标准pso，特别是最优值方面分段pso能够达到0.000292而标准pso仅能到0.086226提高了两个数量级。在复杂、多峰函数Griewank,Rastrigin ，Schaffer的实验中，我们可知实验值（平均值）和最优值方面分段pso比标准pso都有一定的改善，特别是在Rastrigin的实验中分段pso的最优值比标准pso的最优值高了一个数量级，而且在三个函数达优率方面分段pso也都均优于标准pso。

由实验一和实验二我们可知，不管是在单峰函数还是复杂多峰函数，不管是低维搜索空间还是高维搜索空间，分段pso算法通过分阶段分模型进化增加了粒子群的多样性，从而种群当陷入到局部极值点时多样性越大也就说明了探测到新的搜索域的可能性越大，增加了跳出局部极值的可能性提高了算法的性能。在第五章我们将就分段pso的多样性进行分析。

# 第五章 分段微粒群算法的多样性分析

## 5.1多样性

群体多样性提供了群智能算法的运行信息，群体多样性给出了算法运行过程中粒子的“分散”程度与“收敛”程度的信息。种群的多样性就决定者种群的搜索能力，种群多样性越大粒子群的分散程序越大搜索的范围将会更大，这样找到全局最优解的可能性越大。关于位置多样性的定义，我们有两种定义：一种主要基于粒子个体（element-wise）的位置多样性定义，另一种是基于粒子维度（dimension-wise）的位置多样性定义。

基于粒子个体的多样性定义将粒子的所有的维度都视为一个整体，首先求出所有粒子的当前位置的平衡值也就是中心位置，然后再计算所有粒子当前位置到平衡中心的距离，再将结果求和后除以粒子个数和解空间维数，结果就是表示粒子的位置多样性。但是基于个体的多样性定义忽略了粒子在每一维的差异，特别是在极值点附近各维度数值不同的情况下，在这种情况下就难以合理表征粒子的多样性的作用。

本文采用参考文献[6]文中提出的基于L1范式的粒子群算法的基于维度的多样性定义，本文主要讨论粒子群的位置多样性因此我们给出位置多样性计算公式：

(5.1)

(5.2)

(5.3)

先利用（5.1）式求出各维度上的中心位置（1，2,…,Dim），然后再按（5.2）式计算粒子每一维到这一维度中心位置的L1范式的距离，将各维度上到中心位置的距离求和除以种群规模大小得到表征粒子在每一维度上的多样性，最后利用（5.3）将各维度上的多样性求平均值得到D表征整个粒子群的位置多样性。下一小节我们将按这种定义来对分段pso的位置多样性进行分析。

## 5.2 分段微粒群算法的多样性分析

在本实验中，标准pso和分段pso的参数设置:最大迭代次数Tmax=2000,问题维数Dim=10，种群规模num=20。本部分主要选取的是一些复杂、多峰函数作为实验目标函数。横轴表示迭代次数，坚轴表示位置多样性D值。

Rastrigin函数的位置多样性:

Griewank函数的位置多样性:

Rosenbrock函数的位置多样性：

Schaffer函数的位置多样性：

通过本实验，我们在这里把以上8幅曲线图结合第四章实验结果部分对种群的多样性进行分析。

首先，从以上8幅曲线图的比较我们可以得到最直观的结论。利用分段pso算法进行进化的粒子群的位置多样性均呈现出了一些波动，而标准pso算法进行优化的粒子群的位置多样性并没有出现过大的波动总体呈现出一个缓慢的递减的趋势。根据第一小节我们对多样性的意义的讨论我们知道了种群的多样性是表明了粒子群的“分散”与“收敛”程度，决定着种群的搜索能力。标准pso的位置多样性呈现出递减趋势证明了第二章对于标准pso算法的评介。在算法开始时粒子群多样性最大，这就证明了标准pso算法在早期探测能力极强，随着迭代次数的增加粒子群的多样性逐渐减小也就是说种群到晚期的开发能极强了，这样标准pso算法的缺点也就非常直观地摆在我们面前了。而在分段pso中算法在整个过程中若陷入了局部最优时，则按照算法进行重置的处理此时便增加了种群的多样性，也就是会出现波动的原因。再结合第四章的表4.1通过分析我们知道了分段pso相对于标准pso在性能方面有所改善，这又反过来证明了多样性的意义。

其次，我们仔细分析这8幅图我们可以看到标准pso算法在整个算法进化过程中种群的多样性是处于一个缓慢地递减过程，这是由于ω递减的结果。而分段pso算法进化过程中可以看成由若干个周期组成的。第一段，先是一个缓慢地递减过程，这是由于分段pso算法的第一阶段利用标准pso来进行进化的原因；第二段，此时出现了一个快速递减的过程，这是由于分段pso在此过程是采用的公式（3.3）进化的多样性快速下降说了明了此时具有极强的局部搜索能力，也同时证明了公式（3.3）具有极强的局部搜索能力；第三段，是一个位置多样性激增的过程，这是由于我们利用了人工蜂群中引领蜂的思想将全局最好位置进行了重置并利用公式（3.4）对其它一半粒子进行了处理，从而增加了种群的多样性。

第三，对于Griewank,Rastrigin这样的多峰，多极值函数，分段pso算法下其位置多样性的波动次数较多，说明分段pso跳出局部最优寻找全局最优的能力提高了。

最后，结合第四章的实验结果。我们可以得出：分段pso增加了种群的多样性，提高了算法的性能。

# 第六章 结束语

微粒群优化算法是一种智能优化算法，这种算法已经在人工神经网络方面成功应用了。这种算法正在显示着巨大的潜能，但是算法还处于研究阶段。通过对标准pso算法的理论分析与仿真实验，证明了标准微粒群算法（pso）在优化一些复杂问题上容易陷入到局部极值的陷阱之中出现“早熟”现象，使得算法性能变差。本文通过对标准pso缺点进行分体，从种群多样性分析与平衡算法探测与开发的矛盾着手，从增加种群多样性的角度出发提出了增加种群多样性的分段pso算法。通过仿真实验，本文主要通过选取了一些复杂、多峰为目标函数的实验，证明了分段pso算法在平均寻优值，达优率等各个性能方面相对于标准pso有所提高。最后本文根据文献[6]提出的位置多样性分析方法对分段pso算法的种群位置多样性进行了分析，验证了分段pso算法的理论。怎么样使得算法性能更好将依然值得我们去探讨。

# 参考文献

[1] Hackwood S, Bent G. Self-organization of sensors for Swarm Intelligence[C]. IN: IEEE International conference on Robotics and Automation. Piscataway, NJ:IEEE Press,1992. 819~829.

[2] Kennedy J, Eberhart R C. Swarm intelligence [M].San Francisco; Morgan Kaufmann,2003, 350~389.

[3] 吴烈阳. 基于多种群的改进微粒群优化算法研究,硕士学位论文[D]:南昌:南昌航空大学,2008.

[4] 高尚,杨静宇.群智能算法及应用[M].北京：中国水利水电出版社,2006,1~16.

[5] Kennedy J, Eberhart R. Particle swarm optimization[C].IEEE Int'Conf.on Neural Networks. Perth, Australia, IEEE Service CenterPiscatawayNJ,1995:1942-1948.

[6] 程适,史玉回.基于L1范式的粒子群算法群体多样性研究[J].计算机科学,2011, 38(7):190-193.

[7] 吴烈阳，孙辉等.基于不同进化模型的双群交换微粒群优化算法[J]. 南昌工程学院学报，2008，27(4)：1-4

[8] 赵嘉,孙辉.多阶段多模型的改进微粒群优化算法[J].计算机工程与应用.2010, 46(33):32-35.

[9] T. M. Blackwell.Particle swarms and population diversity[J].Soft Comput,2005, 9:793-802.

[10] Liang J J, Qin A K, Suganthan P N et al. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. IEEE Transactions on Evolutionary Computation, 2006, 10, (3): 281~295.

[11]曾建潮 等.微粒群算法[M].北京 ：科学出版社，2004.

# 致谢

在这里衷心地感谢孙辉老师在整个毕业设计过程中在理论与实验上的耐心细心的指导，感谢赵嘉老师的指导，李俊学长，史小露学姐在实验中的指导。