文章编号:1001-506X(2010)11-2453-06

多邻域改进粒子群算法

杨雪榕[1], 梁加红[1], 陈 凌[1], 尹大伟[2]
(1. 国防科学技术大学机电工程与自动化学院, 湖南 长沙 410073;
2. 国防科学技术大学航天与材料工程学院, 湖南 长沙 410073)

摘 要: 为了改进标准粒子群算法的性能,提出了多邻域改进粒子群算法。算法提出了一种较为简单的多邻域拓扑方案,对速度惯性权重的更新策略进行了改进,引入了速度和搜索区间限制算法。经过对经典测试函数的计算测试,算法表现出良好的复杂问题求解能力。最后,针对多目标优化问题,给出了多目标应用在粒子群算法中的处理方法,并对经典的 5 维优化和 Golinski 减速器设计问题进行了求解,通过数据比对,证明了算法性能远优于现有的一些算法。

关键词: 粒子群; 多目标优化问题; 多邻域拓扑; 性能测试; Golinski 减速器问题
中图分类号: TP 301.6 **文献标志码**: A **DOI**:10.3969/j.issn.1001-506X.2010.11.42

Multi-neighborhood improved particle swarm optimization algorithm

YANG Xue-rong[1], LIANG Jia-hong[1], CHEN Ling[1], YIN Da-wei[2]
(*1. Coll. of Mechanotronics Engineering and Automation, National Univ. of Defense Technology, Changsha 410073, China;*
2. School of Aerospace and Material Engineering, National Univ. of Defense Technology, Changsha 410073, China)

Abstract: The multi-neighborhood improved particle swarm optimization algorithm (MNI-PSO) is proposed for the purpose of improving the capability of the standard particle swarm optimization (PSO). The MNI-PSO contains a simple neighborhood topology and sets an improved update scheme of the velocity inertial weight. The velocity and searching area restriction algorithms are also proposed for the MNI-PSO. The optimization results of the classical testing problems show that the MNI-PSO has performed a great capability for the complex optimization problems. Finally, the solution to multi-objective optimization problems (MOOP) using MNI-PSO is proposed. The classical 5-D optimization problem and Golinski's speed reducer problem are optimized. The results show that the MNI-PSO's performance is better than some other popular algorithms.

Keywords: particle swarm optimization (PSO); multi-objective optimization problem (MOOP); multi-neighborhood topology; capability test; Golinski's speed reducer problem

0 引 言

随着 Kennedy 和 Eberhart 两位学者提出了粒子群(particle swarm optimization, PSO)算法[1],PSO 已经被用于多个领域。PSO 的优势在于算法简单,易于实现,不需要求解问题的梯度信息,参数少等。为此众多学者针对 PSO 算法提出了改进方案,用于提高算法的性能,例如谭皓提出的混合 PSO 算法[2],Nakagawa 提出的速度控制 PSO[3],Shi 提出的模糊自适应 PSO[4],Li 提出的多群多最优 PSO[5]以及高鹰提出的基于种群密度的粒子群优化算法[6]等。以上算法都在性能上对标准 PSO 有了一定的提升。倪庆剑提出一种基于可变多簇结构的动态概率粒子群优化算法[7],对粒子邻域给出了动态可变拓扑策略,较为明显地提高了算法优化性能,但动态计算拓扑必定增加了原有的计算量。

PSO 在复杂非线性优化问题上表现出的前景更让工程界为之振奋,由于其不需要求解问题的梯度信息,在许多大型多约束问题的求解上表现出良好的效果。张雷对基于粒子群优化算法的无人战斗机路径规划方法进行了探讨和研究[8],王东等利用粒子群算法优化 SVM 分类器的超参数[9],Coello 针对粒子群算法对怎样求解多目标优化问题进行了详细的分析,并对多种测试函数进行了测试试验[10]。贾兆红提出的面向多目标的自适应动态概率粒子群优化算法也较好地解决了多目标优化的问题[11]。

本文针对标准 PSO 算法,首先给出了多邻域改进粒子

收稿日期:2009-10-12; **修回日期**:2010-05-01。
基金项目:国家部委级基金(9140A20010409KG0153)资助课题
作者简介:杨雪榕(1981-),男,博士研究生,主要研究方向为武器系统控制、建模与仿真。E-mail: yangxuerong@126.com

群算法(multi-neighborhood improved particle swarm optimization algorithm ,MNI-PSO)的相关基本定义。然后对算法初始化和搜索区间处理问题进行了分析,给出了具体解决方案。其后给出了算法的邻域划分方案、速度惯性权重改进方案和算法流程。本文针对经典粒子群方法,给出了综合的改进策略。为了测试算法的性能,对经典测试函数 Griewank 函数和 Rosenbroek 函数进行了性能测试。最后针对多目标优化问题,给出了解决方案和测试结果。

1 MNI-PSO 算法原理

一个由 m 个粒子组成的群体在 D 维搜索空间中以一定的速度飞行,每个粒子在搜索时,考虑到了自己搜索到的历史最好点和群体内(或邻域内)其他粒子的历史最好点,在此基础上进行位置(状态,也就是解)的变化。下面给出基本定义:

第 i 个粒子的位置表示为

$$\boldsymbol{x}_i=(x_{i1},x_{i2},\cdots,x_{id},\cdots,x_{iD}),\ \boldsymbol{x}\in[\boldsymbol{X}_{\min},\boldsymbol{X}_{\max}]$$

第 i 个粒子的速度表示为

$$\boldsymbol{v}_i=(v_{i1},v_{i2},\cdots,v_{id},\cdots,v_{iD}),\ \boldsymbol{v}\in[-\boldsymbol{V}_{\max},\boldsymbol{V}_{\max}]$$

式中,$1\leqslant i\leqslant m$,$1\leqslant d\leqslant D$,表示第 d 个参数。

第 i 个粒子经历过的历史最好点表示为

$$\boldsymbol{p}_i=(p_{i1},p_{i2},\cdots,p_{id},\cdots,p_{iD})$$

群体内所有粒子经历过的历史最好点为

$$\boldsymbol{p}_g=(p_{g1},p_{g2},\cdots,p_{gd},\cdots,p_{gD}),\ \boldsymbol{p}_g=best(\boldsymbol{p}_i)$$

N_j 表示群体中的第 j 个邻域,N_j 内所有粒子经历过的历史最好点表示为

$$\boldsymbol{pl}_j=(pl_{j1},pl_{j2},\cdots,pl_{jd},\cdots,pl_{jD})$$

式中,$1\leqslant j\leqslant L$,L 表示群体中邻域的个数。

则

$$\boldsymbol{pl}_j=best(\boldsymbol{p}_i),\ i\in N_j$$

粒子的位置和速度都是在连续的空间内进行取值,其变化依据以下方程进行

$$v_{id}^{k+1}=w^k v_{id}^k+c_1\xi(p_{id}^k-x_{id}^k)+c_2\mu(p_{gd}^k-x_{id}^k)(1-T_i)+c_3\eta(pl_{jd}^k-x_{id}^k)T_i \tag{1}$$

$$x_{id}^{k+1}=x_{id}^k+v_{id}^{k+1} \tag{2}$$

式中,k 表示第 k 代粒子;w 表示速度的惯性权重;c_1、c_2、c_3 分别为自学因子、全局学习因子和邻域学习因子,一般取值范围在[0,4];$\xi,\mu,\eta\in U[0,1]$,是在[0,1]区间内均匀分布的三个伪随机数。这里强调式(1)中是对每个粒子的每一维参数分别求取速度,这样,每一维数据将会得到不同的随机变化。

T_i 为邻域学习能力函数,用于区别粒子获取全局或邻域知识的能力,T_i 可以取为[0,1]区间内的常数或者随机数。为了隔离各个邻域,使得邻域内搜索相对独立,一种较为简单的学习判别函数为

$$T_i=\begin{cases}0, & i\in H\\ 1, & i\notin H\end{cases} \tag{3}$$

式中,H 表示邻域内有权限获取全局知识的粒子集合。

1.1 初始化与搜索区间处理

搜索区间指的是粒子能够飞行的区域,一般问题对粒子的状态参数都有约束区间,但标准粒子群算法一般不约束搜索空间,利用算法的收敛特性使得粒子自动回归到最优区间。但是,对于某些优化问题,状态空间外的状态是没有意义的,所以必须对粒子速度和位置更新进行处理。

初始化时一般取 $\boldsymbol{V}_{\max_d}=\alpha(\boldsymbol{X}_{\max_d}-\boldsymbol{X}_{\min_d})$,其中取最大速度系数 $\alpha\in(0,1]$为常数,即根据问题可以设置最大速度的大小。粒子初始化时,采用以下公式

$$v_{id}^0=-\boldsymbol{V}_{\max_d}+\varepsilon_{id}\times 2\boldsymbol{V}_{\max_d} \tag{4}$$

$$x_{id}^0=\boldsymbol{X}_{\min_d}+\delta_{id}\times(\boldsymbol{X}_{\max_d}-\boldsymbol{X}_{\min_d}) \tag{5}$$

式中,$\varepsilon_{id},\delta_{id}\in U[0,1]$,$\varepsilon_i$、$\delta_{id}$ 是第 i 个粒子第 d 维参数初始化时所选取的均匀分布的伪随机数。α 取值越大,粒子具有的初始速度区间越大,粒子可能具有的初始速度越大,搜索范围越广。对于粒子更新时超越最大速度和飞出区间的处理,本文采用以下算法进行搜索范围控制

$$x_{id}^{k+1}=\begin{cases}x_{id}^k+0.5v_{id}^{k+1}, & v_{id}^{k+1}\geqslant\boldsymbol{V}_{\max_d}\\ x_{id}^k+v_{id}^{k+1}, & v_{id}^{k+1}<\boldsymbol{V}_{\max_d}\end{cases} \tag{6}$$

$$\tilde{x}_{id}^{k+1}=\begin{cases}x_{id}^{k+1}-0.25\mid v_{id}^{k+1}\mid, & x_{id}^{k+1}>\boldsymbol{X}_{\max_d}\\ x_{id}^{k+1}+0.25\mid v_{id}^{k+1}\mid, & x_{id}^{k+1}<\boldsymbol{X}_{\min_d}\end{cases} \tag{7}$$

式(6)在一次迭代中只执行一次,这样在不减小速度的情况下起到了限速的作用,这样可以使得粒子不会丧失搜索能量。式(7)需要反复执行直到满足初始区间约束。

1.2 邻域划分

全局粒子群搜索算法由于共享了全局最优信息,可以以较快的速度收敛,但是有时会陷入局部最优;在粒子群中加入邻域拓扑结构,即将粒子群中的粒子按照某种拓扑结构划分若干个邻域,在邻域内的粒子可以共享邻域内的知识,这样虽然收敛速度变慢了,但是很难陷入局部最优。

为了能够有效地传递全局信息,又较好地保持邻域内的独立搜索性能,本文设计了一种简单的邻域方案,如图 1 所示,m 个粒子按照粒子索引号进行邻域划分,假设分为 L 个邻域,令前 $L-1$ 个邻域的粒子数为 $h=(m\ \mathrm{div}\ L)+1$,即总粒子数整除邻域个数再加 1,则最后一个邻域粒子个数为 $m\bmod h$,第 j 个邻域 N_j 的表达式为

$$N_j=\begin{cases}\{i\mid(j-1)h+1\leqslant i\leqslant jh\}, & 1\leqslant j<L\\ \{i\mid(L-1)h+1\leqslant i\leqslant m\}, & j=L\end{cases} \tag{8}$$

例如粒子数 $m=20$,分为 3 个邻域,则每个邻域粒子数为 $h=(20\ \mathrm{div}\ 3)+1=7$。这样,前 2 个邻域拥有相同的粒子个数为 7,最后一个邻域粒子个数为 6。

邻域信息传递方案是,每个邻域的第一个粒子接受全局知识,其他粒子只接受邻域内知识,这样全局搜索的知识通过每个邻域的第一个粒子传递到各个邻域,但不直接传递给全部粒子,增加了邻域内粒子搜索的独立性。于是邻域学习能力函数 T_i 变为

$$T_i=\begin{cases}0,\ i=(j-1)h+1,\ j=1,2,\cdots,L\\ 1,\ i\neq(j-1)h+1\end{cases}\tag{9}$$

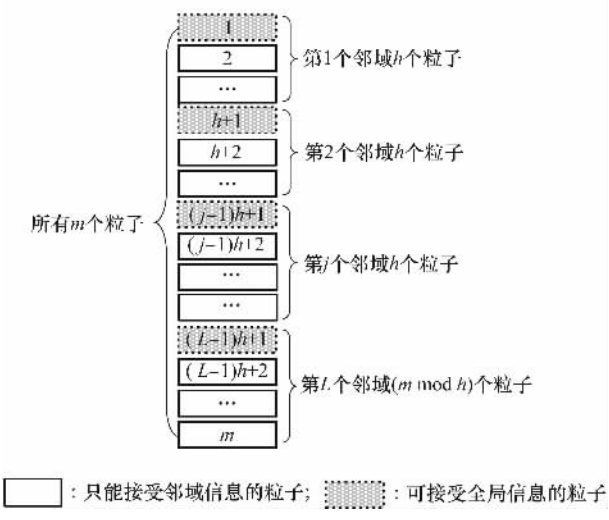

图 1　邻域拓扑方案

1.3　惯性权重

惯性权重 w 是对粒子群算法的探索能力和开发能力进行平衡的关键性参数。较大的惯性权重使粒子在自己原来的方向上具有更大的速度,从而在原方向上飞行更远,具有较好的探索能力;较小的惯性权重使粒子继承了较少的原方向的速度,具有更好的开发能力。通过调解惯性权重的大小便能够调节粒子的搜索能力。

一般情况下,我们希望在优化初期,粒子具有较大的探索能力,在优化末期有较大的开发能力,所以可以使用时变权重,本文采用线性化时变权重

$$w^k=w_{\max}-\frac{w_{\max}-w_{\min}}{I_{\max}}\times k\tag{10}$$

式中,$I_{\max}$表示设定的最大迭代次数;k 是当前迭代次数。在时变惯性权重下,权重随着迭代的进行逐步变小。但是每一个粒子进化的程度不同,在最优点附近的点如果具有较大的速度,将会飞过最优点。所以本文提出了对惯性权重的改进,在线性时变权重上加入当前优化信息。如果被优化的函数为 $f(\boldsymbol{x})$,且求取函数最小值,则当前的惯性权重在式(10)的基础上,进行进一步判断

$$\tilde{w}^k=\begin{cases}w^k,\ f(\boldsymbol{x}^k)\leqslant f(\boldsymbol{x}^{k-1})\\ \upsilon w^k,\ f(\boldsymbol{x}^k)>f(\boldsymbol{x}^{k-1})\end{cases}\tag{11}$$

式中,υ 是在[0,1]区间内均匀分布的伪随机数。这个公式的含义是,如果当前状态比前一状态更优,则粒子速度按照原有的惯性大小进行速度更新;若当前状态比前一状态性能差,说明粒子飞离了局部优点,则将原有的惯性进行随机减小,粒子飞离局部优点的速度有所下降,增加开发能力。这样做的结果是,个别粒子能够迅速在局部最优点收敛,当其他粒子搜索到更优的点时,这个粒子便能够跳出局部优化点,特别适合本文提出的具有独立性较强的邻域划分方案。

但是,当某个粒子处于邻域最优点且邻域内其他粒子无法找到更优点时,由于式(11)其速度会很快降至 0,导致其自身的搜索能力丧失。这时,系统还需要消耗计算量用于重复计算该粒子的状态值。针对这一问题,在算法中设置当前状态变异率 Mut,当速度 v_i^{k+1} 下降到某一程度时,对更新后状态进行位置随机跃迁,便可使得该粒子重新具有搜索能量,进行继续搜索。

本文设计的速度判别函数基于速度的欧几里德范数(euclidean norm)的速度判别函数,用于使得失去能量的粒子重新获得搜索能量。

$$\|\boldsymbol{v}_i\|_2=\sqrt{\sum_{d=1}^{D}v_{id}^2}\tag{12}$$

$$\tilde{x}_{id}^{k+1}=\begin{cases}x_{id}^{k+1}+v_{\max_d}\times Mut\times(\upsilon-0.5),\ \|\boldsymbol{V}_i\|_2<\tau\\ x_{id}^{k+1},\ \|\boldsymbol{V}_i\|_2\geqslant\tau\end{cases}\tag{13}$$

式中,$\upsilon\in U[0,1]$,τ 是最小速度限制,为常数,其大小与问题求解精度有关。

1.4　算法流程

基于以上论述,算法按照下面的步骤进行:

步骤 1　按照式(4)和式(5)进行初始化,并将初始最优值赋为精度范围最大值;

步骤 2　按照状态量计算优化函数值 $f_i(x_i^k)$,并更新 $\boldsymbol{p}_i$、$\boldsymbol{p}_g$ 和 $\boldsymbol{pl}_j$;

步骤 3　判断 $k<I_{\max}$是否成立,成立则终止算法;否则进入步骤 4;

步骤 4　按照式(10)计算惯性权重,依据式(11)进行惯性权重更新;

步骤 5　按照式(9)和式(1)进行速度更新;

步骤 6　按照式(6)进行位置更新,重复执行式(7)直到满足区间约束;

步骤 7　计算式(12),并依据结果按照式(13)进行位置更新,最后返回步骤 2。

2　经典测试函数测试实验

为了验证 MNI-PSO 算法,选择以下两个函数进行计算验证,其中 Rosenbroek 函数为单峰,非凸的病态函数,也是难度较大的复杂优化问题,可用来评价算法的执行效率;Griewank 函数是多峰函数,有很多的局部极小值点,一般算法都较难找到全局最优值,因此可以用来检验算法跳出局部最优的能力。

(1) Rosenbroek 函数

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

$\boldsymbol{x}\in[-15,30]^n$,最小速度限制 $\tau=10^{-4}$。

(2) Griewank 函数

$$f_2(x)=\frac{1}{4\ 000}\sum_{i=1}^{n}x_i^2-\prod_{i=1}^{n}\cos\left(\frac{x_i}{i}\right)+1$$

$\boldsymbol{x}\in[-300,600]^n$,最小速度限制 $\tau=10^{-6}$。

这些算法被多位学者进行引用并测试了自己的算法[4,12-16],表 1 中列出了算法对比 Shi[4] 提出的模糊自适应 PSO 测试结果的对比。测试函数 $n=30$,算法均运行 100 次; MNI-PSO 算法参数为 $c_1=c_2=c_3=2$,群体规模 $m=20$,群体中邻域的个数 $L=3$,最大最小惯性权重 $w_{\max}=0.7$,$w_{\min}=0.1$,最大速度系数 $\alpha=1$,状态变异率 $Mut=0.01$,最大迭代次数为 $I_{\max}=2\ 000$。

由测试结果可以看出,MNI-PSO 算法在复杂函数上表现出良好的性能,并且具有良好的跳出局部最优的能力,非常有利于复杂工程应用的最优化求解问题。MNI-PSO 算法对 Rosenbroek 函数求解结果比 Shi 提出的模糊自适应 PSO 具有更小的统计均值,说明 MNI-PSO 具有更高的计算效率。MNI-PSO 算法对 Griewank 函数求解与 Shi 提出的模糊自适应 PSO 具有相同量级的统计均值,且最优点达到10^{-11}方量级,说明 MNI-PSO 具有很好的局部最小跳出能力。

表 1 经典测试函数测试结果

	Shi 实验平均值	MNI-PSO		
		最优值	最差值	平均值
f_1	316.446 8	0.558 0	628.332 7	96.292 5
f_2	0.018 2	1.787 9e−11	0.141 4	0.031 6

3 多目标优化问题解决方案及实验

对于多目标优化(multi-objective optimization problems, MOOP)问题的定义可以参考文献[17],以最小化问题为例,可将 MOOP 问题描述为以下数学模型

$$\begin{aligned}&\min\ \boldsymbol{f}(\boldsymbol{x})=(f_1(\boldsymbol{x}),f_2(\boldsymbol{x}),\cdots,f_m(\boldsymbol{x}))^{\mathrm{T}}\\&\text{s.t. }\boldsymbol{g}(\boldsymbol{x})=(g_1(\boldsymbol{x}),g_2(\boldsymbol{x}),\cdots,g_p(\boldsymbol{x}))^{\mathrm{T}}\leqslant 0\\&\boldsymbol{x}\in\mathbf{R}^n,\ \boldsymbol{f}(\boldsymbol{x})\in\mathbf{R}^m,\ \boldsymbol{g}(\boldsymbol{x})\in\mathbf{R}^p\end{aligned}\tag{14}$$

由于粒子群算法为单目标寻优算法,对于多目标优化问题,在使用 MNI-PSO 算法时,需要将其转化为单目标问题,如何进行最优解的描述,成为多目标优化问题的难点之一。由于 MNI-PSO 算法是针对整个群体进行更新操作,每个粒子所经历的最优位置,也是一个最优解集。MOOP 的 Pareto 最优解一般也是一个集合,这种相似性使得 MNI-PSO 较为适合于求解 MOOP 的 Pareto 最优集。Pareto 最优相关定义可见文献[17],文献[10]给出了粒子群算法的多目标优化问题处理办法。本文给出一种简单的罚函数法,可有效地处理多目标问题,将式(14)转化为标量形式,第 k 代第 j 个粒子 $\boldsymbol{x}_j^k$ 的目标函数

$$\min\ J_j^k=\phi_j^k+\sum_{i}^{p}\varphi_i$$

$$\phi_j^k=\begin{cases}\phi_j^{k-1},\ \boldsymbol{x}_j^k\sim\boldsymbol{p}_j\\ \phi_j^{k-1}-1,\ \boldsymbol{x}_j^k\preceq\boldsymbol{p}_j\end{cases},\quad\varphi_i=\begin{cases}\kappa,\ g_i(\boldsymbol{x}_j^k)>0\\ 0,\ g_i(\boldsymbol{x}_j^k)\leqslant 0\end{cases}\tag{15}$$

式中,κ 是惩罚系数,取一个非常大的值,用于识别不符合约束的惩罚值。ϕ_j^k 代表第 k 代粒子性能评价系数,取 $\phi_0=I_{\max}$最大迭代次数。目标函数的含义是,如果当前粒子 $\boldsymbol{x}_j^k$ 弱优超当前粒子最优点 $\boldsymbol{p}_j$,则将当前粒子登记评价系数 ϕ_i^{k-1} 减少 1;若 $\boldsymbol{x}_j^k$ 与 $\boldsymbol{p}_j$ 无差异,则不改变当前粒子登记评价系数。

由式(15)可以看出,满足约束,且性能较优的粒子具有较小的目标函数。进行粒子更新时,如果 $J_j^k<J_{j\min}$,即当前目标函数小于该粒子最小目标函数,则 $\boldsymbol{p}_j=\boldsymbol{x}_j^k$,否则不进行最好点更新。如果 $\boldsymbol{p}_j\preceq\boldsymbol{p}_g$,即当前粒子的最优点弱优超全局最优点,$\boldsymbol{p}_g=\boldsymbol{p}_j$。由此,便可得到 Pareto 最优解集$\{\boldsymbol{p}_j, j=1,2,\cdots,m\}$和优化最优解 $\boldsymbol{p}_g$。

为了对比 MNI-PSO 与其他多目标优化算法的性能,以下对两组非线性约束优化问题进行了 MNI-PSO 算法测试。测试的参数设置与上一节经典测试函数试验的设置相同,不同的参数为:最小速度限制 $\tau=0.000\ 01$,取约束不满足的惩罚系数 $\kappa=10^6$。

3.1 五维非线性约束优化问题

$$\begin{aligned}&\min\ f(\boldsymbol{x})=5.357\ 854\ 7x_3^2+0.835\ 689\ 1x_1x_5+37.293\ 239x_1-40\ 792.141\\&\text{s.t. }0\leqslant c_1(\boldsymbol{x})\leqslant 92,\ 90\leqslant c_2(\boldsymbol{x})\leqslant 110,\ 20\leqslant c_3(\boldsymbol{x})\leqslant 25\\&78\leqslant x_1\leqslant 102,\ 33\leqslant x_2\leqslant 45,\ 27\leqslant x_3,x_4,x_5\leqslant 45\end{aligned}$$

其中

$$\begin{cases}c_1(x)=85.334\ 407+0.005\ 685\ 8x_2x_5+0.000\ 25x_1x_4-0.002\ 205\ 3x_3x_5\\ c_2(x)=80.512\ 49+0.007\ 131\ 7x_2x_5+0.002\ 995\ 5x_1x_2+0.002\ 181\ 3x_3^2\\ c_3(x)=9.300\ 961+0.004\ 702\ 6x_3x_5+0.001\ 254\ 7x_1x_3+0.001\ 908\ 5x_3x_4\end{cases}$$

以上五维非线性约束优化问题罗亚中在其博士论文中[17]用混沌优化算法(chaos optimization algorithm, COA)[18]、遗传算法(genetic algorithm, GA)[19]、进化规划(evolutionary programming, EP)[20]、模拟退火算法(simulated annealing algorithm, SAA),以及论文中提出的并行模拟退火单纯形算法(parallel simulated annealing using simplex method, PSASM)[21],对该问题 10 次求解的最好求解结果进行了对比,本文不加改变的直接引用文献[17]计算的数据,并加入本文方法计算结果,结果数据如表 2 所示。

表 2　五维优化结果对比

	COA	GA	EP	SAA	PSASM	MNI-PSO	
						最优值	最差值
x_1	78.63	80.61	78.00	78.00	78.00	78.000 124	78.000 001
x_2	33.26	34.21	33.00	33.10	33.00	33.000 012	33.023 603
x_3	27.79	31.34	27.08	27.23	27.07	27.071 141	27.085 267
x_4	44.26	42.05	45	44.92	44.99	44.999 795	44.999 967
x_5	43.49	34.85	44.97	44.53	44.96	44.968 979	44.925 333
c_1	91.80	90.58	92.00	91.95	92.00	92.000	91.999
c_2	100.34	99.41	100.40	100.38	100.40	100.405	100.409
c_3	20.07	20.12	20.01	20.01	20.00	20.000	20.000
f_x	−30 862.53	−30 175.80	−31 022.65	−31 004.98	−31 025.02	−31 025.526 4	−31 024.281 9
计算次数	10 198	40 000	4 000	10 000	3 315	3 000	3 000

本文随机计算 5 次,设置最大迭代次数为 $I_{max}=150$,在表 2 中列出了最好和最坏结果,图 2 中给出了算法收敛曲线。由图中曲线可以看出,MNI-PSO 算法在非常少的迭代中便收敛,对于求解这类带约束的优化问题非常适合。最好的一次优化,算法在进行 10 代更新就已经收敛。从表 2 中的结果可以看出,相对于其他算法,MNI-PSO 具有更小的计算量,而且得到了更高的优化精度。

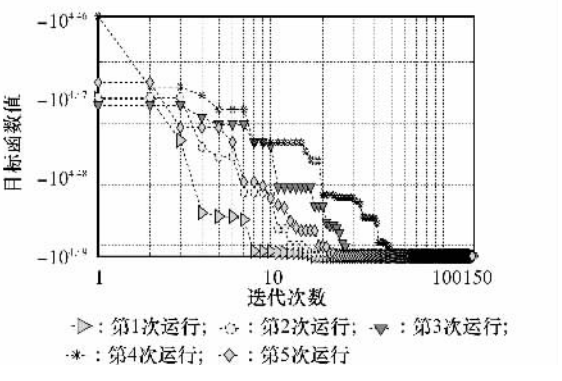

图 2　5 维优化问题的 MNI-PSO 迭代曲线

3.2　Golinski 减速器设计问题

$$\min f(\boldsymbol{x}) =$$

$$0.785\,4x_1x_2^2(3.333\,3x_3^2+14.933\,4x_3-43.093\,4)-1.507\,9x_1(x_6^2+x_7^2)+7.477(x_6^3+x_7^3)+0.785\,4(x_4x_6^2+x_5x_7^2)$$

$$\text{s.t. } g_i(\boldsymbol{x})\leqslant 1(i=1,2,\cdots,9),\ 2.6\leqslant x_1\leqslant 3.6$$

$$0.7\leqslant x_2\leqslant 0.8,\ 17\leqslant x_3\leqslant 28(\text{Integer Value})$$

$$7.3\leqslant x_4,x_5\leqslant 8.3,\ 2.9\leqslant x_6\leqslant 3.9,\ 5.0\leqslant x_7\leqslant 5.5$$

式中

$$g_1(\boldsymbol{x})=27x_1^{-1}x_2^{-2}x_3^{-1},\ g_2(\boldsymbol{x})=397.5x_1^{-1}x_2^{-2}x_3^{-2}$$

$$g_3(\boldsymbol{x})=1.93x_2^{-1}x_3^{-1}x_4^3x_6^{-4},\ g_4(\boldsymbol{x})=1.93x_2^{-1}x_3^{-1}x_5^3x_7^{-4}$$

$$g_5(\boldsymbol{x})=\left[(745x_4x_2^{-1}x_3^{-1})^2+1.69\times10^6\right]^{0.5}(110.0x_6^3)^{-1}$$

$$g_6(\boldsymbol{x})=\left[(745x_5x_2^{-1}x_3^{-1})^2+157.5\times10^6\right]^{0.5}(85.0x_7^3)^{-1}$$

$$g_7(\boldsymbol{x})=x_2x_3/40,\ g_8(\boldsymbol{x})=5x_2/x_1$$

$$g_9(\boldsymbol{x})=x_1/12x_2,\ g_{10}(\boldsymbol{x})=(1.5x_6+1.9)x_4^{-1}$$

$$g_{11}(\boldsymbol{x})=(1.1x_7+1.9)x_5^{-1}$$

Golinski 减速器设计问题是 NASA 提出的多学科设计优化测试问题,许多文献报告了对它优化结果[21,22-25]。罗亚中[17](Luo)在其博士论文中进行了归纳和对比,并指出 Li[26]、Azarm[23]、Rao[24] 和 Kuang[25] 所提供的解都是非可行的。Ray[22] 采用粒子群算法获得了可行解。Luo 在文献[17,21]中给出的并行模拟退火单纯形算法的解优于以上文献中得到的解。

本文给出的 MNI-PSO 算法得到了比 Luo 更优的结果,其计算效率也相对较高。表 3 是进行了 5 次随机试验得到的最优解和最差解。算法设置最大迭代次数为 $I_{max}=600$,图 3 中给出了 4 次实验的收敛情况。图中跳变表示状态量由不满足约束到满足约束的过程。由于粒子总数为 20,所以从图中可以看出 MNI-PSO 算法在计算次数小于 2 000次情况下就已经收敛。

表 3　Golinski 减速器问题优化结果

	Ray 的结果	Luo 的结果	MNI-PSO	
			最优值	最差值
x_1	3.500 000 02	3.500 001 71	3.500 000 000 0	3.500 000 119 3
x_2	0.700 000 0	0.700 000 33	0.700 000 000 0	0.700 000 012 0
x_3	17	17	17	17
x_4	7.300 000 09	7.300 006 44	7.300 000 000 9	7.300 013 553 0
x_5	7.800 000 00	7.715 327 69	7.715 319 913 8	7.715 321 536 0
x_6	3.350 214 68	3.350 215 90	3.350 214 666 2	3.350 214 753 9
x_7	5.286 683 25	5.286 654 60	5.286 654 465 0	5.286 654 568 2
g_1	0.926 084 7	0.926 083 4	0.926 084 719 6	0.926 084 656 2
g_2	0.802 001 5	0.802 000 3	0.802 001 472 9	0.802 001 418 0
g_3	0.500 827 9	0.500 828 1	0.500 827 752 0	0.500 830 480 3
g_4	0.098 528 3	0.095 356 3	0.095 356 095 5	0.095 356 146 6
g_5	1.000 000 0	0.999 998 9	0.999 999 999 9	0.999 999 943 9
g_6	1.000 000 0	0.999 999 9	1.000 000 000 0	0.999 999 941 7
g_7	0.297 500 0	0.297 500 1	0.297 500 000 0	0.297 500 005 1
g_8	1.000 000 0	0.999 999 98	1.000 000 000 0	0.999 999 983 1
g_9	0.416 666 7	0.416 666 7	0.416 666 666 7	0.416 666 673 7
g_{10}	0.948 674 13	0.948 673 66	0.948 674 246 4	0.948 672 503 2
g_{11}	0.989 147 62	0.999 999 01	0.999 999 999 7	0.999 999 804 2
f_x	2 996.232 16	2 994.357 83	2 994.355 026 2	2 994.355 370 4

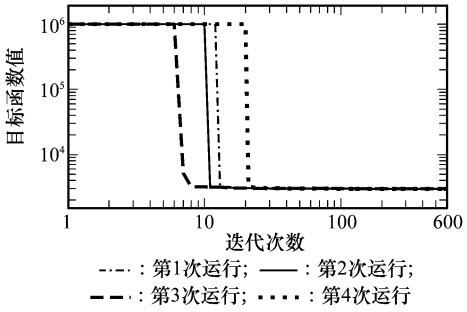

图 3 Golinski 减速器问题的 MNI-PSO 迭代曲线

4 结 论

本文给出的 MNI-PSO 优化算法，在多个方面对标准 PSO 算法进行了改进，得到了较好的效果。从复杂经典问题优化到非线性多约束优化问题的测试结果，可以看出，该算法具有较高的计算性能，可以应用于更加复杂的工程问题。下一阶段的工作是针对 MNI-PSO 算法中的各种参数，进行进一步实验，并确定最优的参数组合，以求达到更好的计算性能。

本文直接引用了罗亚中博士[17,21]所整理的实验结果，在此对他所做的工作表示感谢！

参考文献：

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

[2] 谭皓,沈春林,李锦. 混合粒子群算法在高维复杂函数寻优中的应用[J]. 系统工程与电子技术, 2005, 27(8):1471 - 1474. (Tan H, Shen C L, Li J. Hybrid particle swarm optimization algorithm for high-dimension complex functions[J]. *Systems Engineering and Electronics*, 2005, 27(8):1471 - 1474.)

[3] Nakagawa N, Ishigame A, Yasuda K. Particle swarm optimization with velocity control[J]. *IEEE Trans. on Electrical and Electronic Engineering*, 2009, 4(1):130 - 132.

[4] Shi Y, Eberhart R C. Fuzzy Adaptive particle swarm optimization[C]// *Proc. of IEEE International Congress on Evolutionary Computation*, 2001:101 - 106.

[5] Li J, Xiao X P. Multi-swarm and multi-best particle swarm optimization algorithm[C]// *Proc. of 7th World Congress on Intelligent Control and Automation*, 2008:6281 - 6286.

[6] 高鹰,姚振坚,谢胜利. 基于种群密度的粒子群优化算法[J]. 系统工程与电子技术, 2006,28(6):922 - 924. (Gao Y, Yao Z J, Xie S L. Particle swarm optimization algorithm based on population desity[J]. *Systems Engineering and Electronics*, 2006,28(6):922 - 924.)

[7] 倪庆剑. 一种基于可变多簇结构的动态概率粒子群优化算法[J]. 软件学报, 2009,20(2):339 - 349.

[8] 张雷,王道波,段海滨. 基于粒子群优化算法的无人战斗机路径规划方法[J]. 系统工程与电子技术, 2008,30(3):506 - 510. (Zhang L, Wang D B, Duan H B. Study on uninhabited combat arial vehicle path planning method based on particle swarm optimization algorithm[J]. *Systems Engineering and Electronics*, 2008,30(3):506 - 510.)

[9] 王东, 吴湘滨. 利用粒子群算法优化 SVM 分类器的超参数[J]. 计算机应用, 2008,28(1):134 - 135.

[10] Coello C A, Pulido G T, Lechuga M S. Handling multiple objectives with particle swarm optimization[J]. *IEEE Trans. on Evolutionary Computation*, 2004,8(3):256 - 279.

[11] 贾兆红. 面向多目标的自适应动态概率粒子群优化算法[J]. 系统仿真学报, 2008, 20(18):4959 - 4963.

[12] Cai T, Pan F, Chen J. Adaptive particle swarm optimization algorithm[C]// *Proc. of Fifth World Congress on Intelligent Control and Automation*, 2004:2245 - 2247.

[13] Elshamy W, Emara H M, Bahgat A. Clubs-based particle swarm optimization[C]// *Proc. of IEEE Swarm Intelligence Symposium*, 2007:289 - 296.

[14] Janson S, Middendorf M. A hierarchical particle swarm optimizer[C]// *Proc. of the Congress on Evolutionary Computation*, 2003:770 - 776.

[15] Xu J J, Xin Z H. An extended particle swarm optimizer[C]// *Proc. of 19th IEEE International Parallel and Distributed Processing Symposium*, 2005:193 - 197.

[16] Stacey A, Jancic M, Grundy I. Particle swarm optimization with mutation[C]// *Proc. of the Congress on Evolutionary Computation*, 2003:1425 - 1430.

[17] 罗亚中. 空间最优交会路径规划策略研究[D]. 长沙:国防科学技术大学, 2007.

[18] 骆晨钟, 邵惠鹤. 用混沌搜索求解非线性约束优化问题[J]. 系统工程与理论实践, 2000,20(8):54 - 58.

[19] Homaifar A. Constrained optimization via genetic algorithms[J]. *Simulation*, 1994,62(4):242 - 254.

[20] Fogel D B. A comparison of evolutionary programming and genetic algorithms on selected constrained problems[J]. *Simulation*, 1995,64(6):399 - 406.

[21] Luo Y Z, Tang G J. Parallel simulated annealing using simplex method[C]// *Proc. of 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, 2004:2004 - 4584.

[22] Ray T. Golinski's speed reducer problem revisited[J]. *AIAA Journal*, 2003,41(3):556 - 558.

[23] Azarm S, Li W C. Multi-level design optimization using global monotonicity analysis[J]. *Journal of Mechanism, Transmissions, and Automation in Design*, 1989:259 - 263.

[24] Rao S S. *Engineering optimization*[M]. 3rd ed. New York: Wiley, 1996.

[25] Kuang J K, Rao S S, Chen L. Taguchi-aided search method for design optimization of engineering systems[J]. *Engineering Optimization*, 1998, 30(1):1 - 23.

[26] Li H L, Papalambros P A. Production system for use of global optimization knowledge[J]. *Journal of Mechanism, Transmissions, and Automation in Design*, 1985,17(2):277 - 284.

多邻域改进粒子群算法

作者： 杨雪榕， 梁加红， 陈凌， 尹大伟， YANG Xue-rong， LIANG Jia-hong， CHEN Ling， YIN Da-wei

作者单位： 杨雪榕,梁加红,陈凌,YANG Xue-rong,LIANG Jia-hong,CHEN Ling(国防科学技术大学机电工程与自动化学院,湖南,长沙,410073)， 尹大伟,YIN Da-wei(国防科学技术大学航天与材料工程学院,湖南,长沙,410073)

刊名： 系统工程与电子技术 ISTIC EI PKU

英文刊名： SYSTEMS ENGINEERING AND ELECTRONICS

年，卷(期)： 2010,32(11)

被引用次数： 9次

参考文献(26条)

1. Kennedy J;Eberhart R Particle swarm optimization[外文会议] 1995

2. 谭皓;沈春林;李锦 混合粒子群算法在高维复杂函数寻优中的应用[期刊论文]-系统工程与电子技术 2005(08)

3. Nakagawa N;Ishigame A;Yasuda K Particle swarm optimization with velocity control 2009(01)

4. Shi Y;Eberhart R C Fuzzy Adaptive particle swarm optimization[外文会议] 2001

5. Li J;Xiao X P Multi-swarm and multi-best particle swarm optimization algorithm[外文会议] 2008

6. 高鹰;姚振坚;谢胜利 基于种群密度的粒子群优化算法[期刊论文]-系统工程与电子技术 2006(06)

7. 倪庆剑 一种基于可变多簇结构的动态概率粒子群优化算法[期刊论文]-软件学报 2009(02)

8. 张雷;王道波;段海滨 基于粒子群优化算法的无人战斗机路径规划方法[期刊论文]-系统工程与电子技术 2008(03)

9. 王东;吴湘滨 利用粒子群算法优化SVM分类器的超参数[期刊论文]-计算机应用 2008(01)

10. Coello C A;Pulido G T;Lechuga M S Handling multiple objectives with particle swarm optimization [外文期刊] 2004(03)

11. 贾兆红 面向多目标的自适应动态概率粒子群优化算法[期刊论文]-系统仿真学报 2008(18)

12. Cai T;Pan F;Chen J Adaptive particle swarm optimization algorithm[外文会议] 2004

13. Elshamy W;Emara H M;Bahgat A Clubs-based particle swarm optimization[外文会议] 2007

14. Janson S;Middendorf M A hierarchical particle swarm optimizer 2003

15. Xu J J;Xin Z H An extended particle swarm optimizer[外文会议] 2005

16. Stacey A;Jancic M;Grundy I Particle swarm optimization with mutation 2003

17. 罗亚中 空间最优交会路径规划策略研究[学位论文] 2007

18. 骆晨钟;邵惠鹤 用混沌搜索求解非线性约束优化问题[期刊论文]-系统工程与理论实践 2000(08)

19. Homaifar A Constrained optimization via genetic algorithms[外文期刊] 1994(04)

20. Fogel D B A comparison of evolutionary programming and genetic algorithms on selected constrained problems[外文期刊] 1995(06)

21. Luo Y Z;Tang G J Parallel simulated annealing using simplex method 2004

22. Ray T Golinski's speed reducer problem revisited[外文期刊] 2003(03)

23. Azarm S;Li W C Multi-level design optimization using global monotonicity analysis 1989

24. Rao S S Engineering optimization 1996

25. Kuang J K;Rao S S;Chen L Taguchi-aided search method for design optimization of engineering systems 1998(01)

26. Li H L;Papalambros P A Production system for use of global optimization knowledge 1985(02)

本文读者也读过(2条)

1. 申元霞. 王国胤. 曾传华. SHEN Yuan-xia. WANG Guo-yin. ZENG Chuan-hua PSO模型种群多样性与学习参数的关系研究[期刊论文]-电子学报2011, 39(6)

2. 雷阳. 李树荣. 张强. 张晓东. LEI Yang. LI Shu-rong. ZHANG Qiang. ZHANG Xiao-dong 基于骨干粒子群的混合遗传算法及其应用[期刊论文]-计算机工程与应用2010, 46(36)

引证文献(9条)

1. 谢永强. 陈建军. 徐亚兰 基于仿射算法的确定性全局优化算法[期刊论文]-华南理工大学学报（自然科学版）2012(5)

2. 左旭坤. 苏守宝 粒距反馈的S函数粒子群权值调整策略[期刊论文]-计算机应用 2012(10)

3. 孙文新. 穆华平 自适应群体结构的粒子群优化算法[期刊论文]-智能系统学报 2013(4)

4. 于胜龙. 薄煜明. 陈志敏. 吴盘龙. 朱凯. 尹明锋 基于混沌粒子群优化的新型VRP求解算法[期刊论文]-计算机工程与科学 2012(12)

5. 秦全德. 李丽. 程适. 李荣钧 交互学习的粒子群优化算法[期刊论文]-智能系统学报 2012(6)

6. 陈志敏. 薄煜明. 吴盘龙. 于胜龙 混沌粒子群优化粒子滤波算法[期刊论文]-电光与控制 2013(1)

7. 王兴元. 张鹏 基于细致化仿生的改进粒子群优化算法[期刊论文]-系统工程与电子技术 2012(7)

8. 张鼎逆. 刘毅 基于混合粒子群法的RLV上升段轨迹优化[期刊论文]-江苏大学学报（自然科学版） 2013(1)

9. 姚旭. 王晓丹. 张玉玺. 邢雅琼 基于自适应t分布变异的粒子群特征选择方法[期刊论文]-系统工程与电子技术 2013(6)