

1-8-2019

## Multi-strategy Cooperative Evolutionary PSO Based on Cauchy Mutation Strategy

Yongji Wang

*National Key Laboratory of Science and Technology on Multispectral Information Processing, Automation College, Huazhong University of Science and Technology, Wuhan 430074, China;*

Tingting Su

*National Key Laboratory of Science and Technology on Multispectral Information Processing, Automation College, Huazhong University of Science and Technology, Wuhan 430074, China;*

Liu Lei

*National Key Laboratory of Science and Technology on Multispectral Information Processing, Automation College, Huazhong University of Science and Technology, Wuhan 430074, China;*

Follow this and additional works at: <https://dc-china-simulation.researchcommons.org/journal>



Part of the Artificial Intelligence and Robotics Commons, Computer Engineering Commons, Numerical Analysis and Scientific Computing Commons, Operations Research, Systems Engineering and Industrial Engineering Commons, and the Systems Science Commons

---

This Paper is brought to you for free and open access by Journal of System Simulation. It has been accepted for inclusion in Journal of System Simulation by an authorized editor of Journal of System Simulation.

---

## Multi-strategy Cooperative Evolutionary PSO Based on Cauchy Mutation Strategy

### Abstract

**Abstract:** For improving the performance of particle swarm optimization (PSO) in optimization simulation, a *multi-strategy cooperative evolutionary PSO based on Cauchy mutation strategy* is proposed. The new algorithm divides the whole swarm into three sub-swarms. A part of particles is selected to *Cauchy mutation* with a certain probability, and the rest of particles adjust their *exploitation* and *exploration* by different evolutionary strategies (*large-scale search strategy*, *local search strategy*, and *adaptive velocity updating strategy*). The sub-swarms share their information to achieve cooperation. Three strategies are used to optimize three test functions, and the result shows the advantages of three strategies. The simulation experiment uses the soft lunar landing problem as the simulation model to optimize the trajectory. Simulation results indicate that the performance of improved PSO is superior to other PSO. *The simulation uses OpenMP to parallelization optimization*, which improves the efficiency of the algorithm.

### Keywords

particle swarm optimization, multi-strategy, exploitation, exploration, Cauchy mutation strategy, OpenMP parallelization

### Recommended Citation

Wang Yongji, Su Tingting, Liu Lei. Multi-strategy Cooperative Evolutionary PSO Based on Cauchy Mutation Strategy[J]. Journal of System Simulation, 2018, 30(8): 2875-2883.

## 基于柯西变异的多策略协同进化粒子群算法

王永骥, 苏婷婷, 刘磊

(华中科技大学自动化学院多谱信息处理技术国家级重点实验室, 湖北 武汉 430074)

**摘要:** 为了提高粒子群算法在优化仿真中的性能, 提出基于柯西变异的多策略协同进化粒子群算法。该方法将种群划分为3个子群, 以一定的概率选中粒子进行柯西变异, 未被选中的粒子采用不同的子群进化策略(大范围搜索策略, 精细搜索策略, 自适应速度更新策略)来调节自身的开发能力和探测能力, 子群间通过共享信息来达成协同合作。仿真中将3种策略应用在3种测试函数的优化中, 验证3种策略的优点, 同时对月球软着陆进行仿真寻优, 结果表明改进PSO的优化性能优于其它改进算法, 采用OpenMP对改进PSO进行并行化仿真, 提高并行化PSO算法的效率。

**关键词:** 粒子群算法; 多策略; 开发; 探测; 柯西变异; OpenMP并行化

中图分类号: TP391.9 文献标识码: A 文章编号: 1004-731X(2018)08-2875-09

DOI: 10.16182/j.issn1004731x.joss.201808008

## Multi-strategy Cooperative Evolutionary PSO Based on Cauchy Mutation Strategy

Wang Yongji, Su Tingting, Liu Lei

(National Key Laboratory of Science and Technology on Multispectral Information Processing, Automation College, Huazhong University of Science and Technology, Wuhan 430074, China)

**Abstract:** For improving the performance of particle swarm optimization (PSO) in optimization simulation, a multi-strategy cooperative evolutionary PSO based on Cauchy mutation strategy is proposed. The new algorithm divides the whole swarm into three sub-swarms. A part of particles is selected to Cauchy mutation with a certain probability, and the rest of particles adjust their exploitation and exploration by different evolutionary strategies (large-scale search strategy, local search strategy, and adaptive velocity updating strategy). The sub-swarms share their information to achieve cooperation. Three strategies are used to optimize three test functions, and the result shows the advantages of three strategies. The simulation experiment uses the soft lunar landing problem as the simulation model to optimize the trajectory. Simulation results indicate that the performance of improved PSO is superior to other PSO. The simulation uses OpenMP to parallelization optimization, which improves the efficiency of the algorithm.

**Keywords:** particle swarm optimization; multi-strategy; exploitation; exploration; Cauchy mutation strategy; OpenMP parallelization

## 引言

源于对鸟群捕食过程中迁徙和群居行为的模



收稿日期: 2016-11-11 修回日期: 2017-07-20;  
基金项目: 国家自然科学基金面上项目(61473124);  
作者简介: 王永骥(1955-), 男, 江西吉安, 博士, 教授, 博导, 研究方向为飞行器制导控制, 智能优化与智能控制; 苏婷婷(1993-), 女, 浙江温州, 硕士生, 研究方向为飞行器轨迹优化, 智能优化算法。

拟, 粒子群优化算法(Particle Swarm Optimization, PSO)是由心理学研究人员 Kenned 和计算机智能研究人员 Eberhart 在 1995 年提出的一种基于群智能方法的演化计算技术, 有着简单容易实现同时又有智能背景的特点。

粒子群优化算法自提出以来, 受到众多学者的研究和改进, 运用范围广泛, 其中越来越多地被运

用到飞行器轨迹优化的领域中。文献[1-6]分别使用 PSO 算法对飞行器轨迹优化问题进行求解,表明了 PSO 算法可靠性强,收敛速度快,验证了算法应用到飞行器轨迹优化上的有效性和通用性,提高了最优轨迹的质量。

粒子群优化算法存在两个主要缺点,其一是粒子群优化算法容易陷入到局部极值点中,导致得不到全局最优解;其二是粒子群优化算法的收敛速度比较慢。因此针对粒子群优化算法的不足,学者们对其进行改进。

近年来,众多学者将进化算法中的变异操作引入 PSO 算法。文献[7-9]分别提出了基于“变异”思想的不同改进方法,提高了算法的寻优性能和稳定性。另外也有学者通过引入分群多策略协同进化的思想来改进 PSO 算法。文献[10]提出基于多种群多模型协同进化的粒子群优化算法,通过分群,每分群采用不同的进化模型,从而具有全局性能好、寻优精度高的优点。因此,可以研究增加变异操作和多策略分群搜索思路来改进标准粒子群算法应用到轨迹优化问题中,从而提高算法的寻优性能。

针对标准 PSO 算法的不足,本文引入“变异”和“多策略”思想,将整个种群划分为3个子群,通过改进的柯西变异公式挑选粒子进行变异,同时提出3种新的进化策略,通过策略更新信息使3个种群分别具有强探测能力、强开发能力和快搜索速度能力,从而提出了一种基于柯西变异的多策略协同进化粒子群算法,同时分析算法的收敛性。本文仿真先分别采用3种新策略对3种典型测试函数进行优化,验证了3种策略各自的优点,再以月球软着陆模型为例,进行最优控制寻优,验证了改进算法的有效性,最后进行基于 OpenMP 的并行化仿真,提高了并行化 PSO 算法的优化效率。

## 1 基于柯西变异的多策略协同进化粒子群算法

### 1.1 多策略进化模式

在粒子群算法寻优过程中,惯性权重  $w$  的大

小可以用来控制算法的探测能力和开发能力。较大的惯性权重具有较强的探测能力,能够搜索以前未能达到的区域,较小的惯性权重具有较强的开发能力,将使粒子在当前解的附近区域进行精细搜索,有利于局部搜索和算法收敛。

考虑到同一个种群的粒子,我们希望在不同时期的搜索中,粒子的探测能力和开发能力进行动态调整,因此在种群内可以通过采用线性递减策略,即在迭代过程中线性地减小惯性权重,并表示为:

$$\omega = \omega_{\max} + (\omega_{\min} - \omega_{\max}) \cdot t / t_{\max} \quad (1)$$

考虑到不同种群的粒子,我们希望同一时期的粒子群具有不同的探测能力和开发能力,因此选择不同的惯性权重最大值和最小值来达到效果,这里提出以下策略。

(1) 大范围搜索策略:

$$\omega = 0.9 + (0.6 - 0.9) \cdot t / t_{\max} \quad (2)$$

(2) 精细搜索策略:

$$\omega = 0.4 + (0.2 - 0.4) \cdot t / t_{\max} \quad (3)$$

标准 PSO 算法中,某粒子即使在本次寻优中寻到比上一代更优的解,按照标准 PSO 算法的速度更新公式,它依旧会在下一次寻优中更改寻优方向,这样不加区分统一对待粒子进行速度更新会降低算法的收敛速度。因此提出自适应速度更新策略,区分寻到更优解粒子和普通粒子的速度更新公式,自适应速度更新策略公式如下:

$$v_{id}(t+1) = \begin{cases} \omega \cdot v_{id}(t), & \text{if } f(x_i(t)) < f(x_i(t-1)) \\ \omega \cdot v_{id}(t) + c_1 \cdot \text{rand}() \cdot (p_{id} - x_{id}(t)) + c_2 \cdot \text{rand}() \cdot (p_{gd} - x_{id}(t)), & \text{others} \end{cases} \quad (4)$$

$$\omega = 0.6 + (0.4 - 0.6) \cdot \frac{t}{t_{\max}} \quad (5)$$

### 1.2 多种群协同进化

一般情况下,粒子的探测能力和开发能力是相互矛盾的,若粒子能在一个区域内进行精细地搜索,那么它就较难从这个区域跳出到另一个区域进

行大范围搜索, 反之亦然。这也就导致粒子群陷入局部最优。因此为了协调粒子群算法寻优过程中粒子的探测能力和开发能力, 这里提出采用多种群的思路来进行协作搜索, 基本思想是将整个种群划分为多个子群, 子群之间通过共享当前各子群中最好 gbest 的信息来完成协作进化。根据上小节提出的三种策略, 本文将整个种群平均划分为三个子群, 记为  $S_i$ , 子群  $S_1$  采用大范围搜索策略, 具有强探测能力, 子群  $S_2$  采用精细搜索策略, 具有强开发能力, 子群  $S_3$  采用自适应速度更新策略策略, 具有快搜索速度能力。子群间合作模式如图 1 所示。

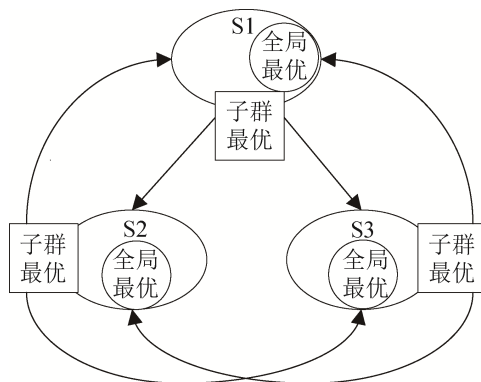


图 1 子群合作模式

Fig. 1 Cooperation model of sub-swarms

### 1.3 柯西变异

PSO 算法与遗传算法比较, 有较多共同特点, 比如: 都是基于种群的; 都需要适应度函数; 都是随机计算技术。因此借鉴遗传算法, 在 PSO 算法中引入柯西变异算子, 即通过以一定的概率  $P$  选中粒子, 对其进行柯西变异, 从而来增加种群的多样性和避免陷入局部最优。其中,  $P$  取值 0.1。

一维的柯西分布的概率密度函数为:

$$f_i(x) = (1/\pi) \cdot t / (t^2 + x^2), -\infty < x < +\infty \quad (6)$$

已知的柯西变异的公式有:

$$X_{ij} = X_{ij} + \eta \cdot C(0,1) \quad (7)$$

式中:  $\eta$  为一个控制变异步长的常数,  $C(0,1)$  是由  $t=1$  的柯西分布函数产生的随机数。

变异步长参数  $\eta$  为一个定值并不利于搜索最优解。在 PSO 算法初中期, 需要较大步长的变异

因子跳出局部最优; 在 PSO 算法后期, 需要小步长变异因子来加速收敛。为此, 变异步长参数改变为一个与位置、迭代数相关的参数, 随着迭代数的增加逐步减小, 改进的柯西变异公式如下:

$$X_{ij} = X_{ij} \cdot (1 + n \cdot C(0,1)) \quad (8)$$

式中:  $n = (\text{MaxIter} - t) / \text{MaxIter}$ 。

### 1.4 改进粒子群算法的描述及流程

在基于柯西变异的多策略协同进化粒子群算法中, 把粒子随机初始化为 3 个子群, 记为  $S_i$ , 在粒子  $p$  每一次迭代, 产生一个(0,1)范围的随机数, 如果随机数大于概率  $P$ , 则该粒子  $p$  按照自己所属种群的进化策略进行进化, 即: 若  $p \in S_1$ , 采用式(2); 若  $p \in S_2$ , 采用公式(3); 若  $p \in S_3$ , 采用式(4)。如果随机数小于或等于概率  $P$  则按照柯西变异式(8)计算新位置和新速度。在每一次迭代中子群保持信息交流, 通过共享各自子群的历史最优产生全局历史最优, 通过协同合作找到最优解。具体流程如下:

Step1. 在搜索空间内, 对种群中的 3 个子群随机初始化, 求出每一个粒子的初始位置和速度。

Step2. 根据适应度函数计算出每一个粒子的适应值, 用来判断粒子在该位置的好坏程度。

Step3. 根据适应值, 找出每一个粒子到本次迭代为止搜索到的最佳解  $pbest$ , 通过共享各自子群的历史最优来找出整个种群中到本次迭代为止搜索到的最佳解  $gbest$ 。

Step4. 产生(0,1)范围的随机数, 若随机数小于或等于概率  $P$ , 则采用柯西变异公式对该粒子的位置进行柯西变异, 否则第一个子群采用公式(2), 第二个子群采用公式(3), 第三个子群按采用公式(4)。

Step5. 返回 Step2 继续执行, 直到获得一个满意的解或是达到某个固定的迭代次数, 算法结束。

## 2 收敛性分析

由参考文献[11]得到随机搜索算法的收敛性的相关分析, 假设 1, 假设 2 和定理 1:

假设 1: 目标函数  $f$  的解空间是  $R^n$  到  $R$ ,  $S$  是  $R^n$  的一个子集, 算法目的是在  $S$  的搜索范围内找到一个解  $z$ , 使得目标函数  $f$  最小化。当存在  $f(D(z, \xi)) \leq f(z)$ , 如果  $\xi \in S$ , 则  $f(D(z, \xi)) \leq f(\xi)$ 。其中  $D$  是算法迭代方式, 用来产生下一代个体。

假设 2: 对于  $S$  的任意 Borel 子集  $A$ , 若其测度  $\nu(A) > 0$ , 则有  $\prod_{i=0}^{\infty} (1 - \mu_i[A]) = 0$ , 其中  $\mu_i[A]$  是由测度  $\mu_i$  所得到  $A$  的概率。

定理 1: 假设目标函数  $f$  为可测函数, 区域  $S$  是可测子集, 能够满足假设 1 和假设 2, 设  $\{z_k\}_{k=1}^{+\infty}$  为算法所生成的解序列, 则  $\lim_{k \rightarrow \infty} P[z_k \in R_\varepsilon] = 1$ 。

分析改进粒子群算法的收敛性, 首先需要证明其符合假设 1、2。

引理 1: 改进粒子群算法满足假设 1。

证明: 由 Step3 可知, 当前粒子最好位置的适应值与全局最好位置的适应值比较, 若比较好, 则将当前粒子最好位置更新为全局最好位置, 则  $D$  函数在改进 PSO 算法中的描述可定义为:

$$D(P_{g,t}, X_{i,t}) = \begin{cases} P_{g,t}, f(P_{g,t}) \leq f(X_{i,t}) \\ X_{i,t}, f(P_{g,t}) \geq f(X_{i,t}) \end{cases}。$$

因此分两种情况进行分析:

若  $f(P_{g,t}) \leq f(X_{i,t})$ , 则  $D(P_{g,t}, X_{i,t}) = P_{g,t}$ , 所以  $f(D(P_{g,t}, X_{i,t})) = f(P_{g,t}) \leq f(X_{i,t})$ 。

若  $f(P_{g,t}) \geq f(X_{i,t})$ , 则  $D(P_{g,t}, X_{i,t}) = X_{i,t}$ , 所以  $f(D(P_{g,t}, X_{i,t})) = f(X_{i,t}) \leq f(X_{i,t})$ 。

综上所述, 改进 PSO 算法满足假设 1。

引理 2: 改进 PSO 算法满足假设 2。

证明: 子群 1 中采用大范围搜索策略的粒子:

$$\begin{aligned} X(t+1) &= \\ X(t) + wv(t) - x(t)(\phi_1 + \phi_2) + P\phi_1 + P_g\phi_2 &= \\ X(t) + w(X(t) - X(t-1)) - X(t)(\phi_1 + \phi_2) + & \\ P\phi_1 + P_g\phi_2 &= \\ (1 + w - \phi_1 - \phi_2)X(t) - wX(t-1) + P\phi_1 + P_g\phi_2 & \end{aligned}$$

因此, 代入可以得到  $M_{it}$ , 为:

$$\begin{aligned} M_{it} &= \\ (1 + w - \phi_1 - \phi_2)X_{i,j,t-1} - wX_{i,j,t-2} + \phi_1P + \phi_2P_g &= \\ X_{i,j,t-1} + (0.9 - 0.3 \cdot t / t_{\max})(X_{i,j,t-1} - X_{i,j,t-2}) + & \\ \phi_1(P - X_{i,j,t-1}) + \phi_2(P_g - X_{i,j,t-1}) & \end{aligned}$$

同理可以得到子群 2 中采用精细搜索策略的粒子:

$$\begin{aligned} M_{it} &= \\ X_{i,j,t-1} + (0.4 - 0.2 \cdot t / t_{\max})(X_{i,j,t-1} - X_{i,j,t-2}) + & \\ \phi_1(P - X_{i,j,t-1}) + \phi_2(P_g - X_{i,j,t-1}) & \end{aligned}$$

子群 3 中采用自适应速度更新策略的粒子:

$$\begin{aligned} M_{it} &= \\ \begin{cases} X_{i,j,t-1} + (0.6 - 0.2 \cdot t / t_{\max})(X_{i,j,t-1} - X_{i,j,t-2}) + & \\ \phi_1(P - X_{i,j,t-1}) + \phi_2(P_g - X_{i,j,t-1}) & \\ \text{if } f(x_i(t)) < f(x_i(t-1)), X_{i,j,t-1} + & \\ (0.6 - 0.2 \cdot t / t_{\max})(X_{i,j,t-1} - X_{i,j,t-2}), \text{ others.} & \end{cases} \end{aligned}$$

显然, 随迭代次数的增加,  $\nu(M_{i,t})$  在不断减少, 其并  $\nu(\bigcup_{i=1}^N M_{i,t})$  也在减少。但是由于改进 PSO

算法在 3 种策略的基础上运用了柯西变异:

$X(t+2) = X(t+1) \cdot (1 + n \cdot C(0,1))$ , 由于柯西变异的随机性, 种群中采用柯西变异的粒子  $\nu(M_{i,t})$  在不断增大, 其并  $\nu(\bigcup_{i=1}^N M_{i,t})$  也在增大。

从而表明, 采用改进 PSO 算法时, 种群中柯西变异过的粒子支撑集的并集为  $\alpha$ ,  $k$  为粒子柯西变异的次数, 则存在整数  $k'$ , 当  $k' > k$  时,  $\alpha \supseteq S$ 。种群中其它无变异、运用三种策略进化的粒子, 设其支撑集的并集分别为  $\beta_1, \beta_2, \beta_3$ , 则改进 PSO 算法搜索结束时,  $\alpha \cup \beta_1 \cup \beta_2 \cup \beta_3 \supseteq S$ , 即改进 PSO 算法的粒子群样本空间的并集包含  $S$ 。

因此, 定义  $S$  的任意 Borel 子集  $A = M_{i,t}$ , 则  $\nu(A) > 0$ , 此时有  $\mu_i[A] = \sum_{i=1}^N \mu_{i,t}[A] = 1$ , 即

$$\prod_{i=0}^{\infty} (1 - \mu_i[A]) = 0。$$

综上所述, 改进 PSO 算法满足假设 2。

引理 3: 改进 PSO 算法是一个全局收敛算法。

证明: 设  $\{P_{g,t}\}_{t=0}^{\infty}$  是算法所生成的解序列, 改进 PSO 算法满足假设 1 和假设 2, 由定理 1 可知:

$\lim_{t \rightarrow \infty} P[P_{g,t} \in R_{\varepsilon}] = 1$ , 因此改进 PSO 算法是一个全局收敛算法。

### 3 算例仿真

#### 3.1 经典测试函数

通过经典的函数优化问题来比较不同  $w$  取值下的 3 种策略的优化结果。测试的二维复杂 Schaffer 函数 1、10 维单峰 Sphere 函数 2、10 维非线性多模态 Rastrigrin 函数 3 数学描述分别如下。

$$\begin{cases} f(x) = 0.5 + \frac{(\sin\sqrt{x_1^2 + x_2^2})^2 - 0.5}{(1 + 0.001(x_1^2 + x_2^2))^2}, |x_i| \leq 10 \\ f(x) = \sum_{i=1}^n x_i^2, |x_i| \leq 100 \\ f(x) = \sum_{i=1}^n (x_i^2 - 10\cos(2\pi x_i) + 10), |x_i| \leq 10 \end{cases} \quad (10)$$

式(10)作为算法的适应度函数, 算法参数设置:  $c_1=1.8$ ,  $c_2=2.2$ , 最大迭代次数为 1000, 粒子数为 100。根据上述条件, 分别对  $w_{\max}=0.9$ ,  $w_{\min}=0.6$  的大范围搜索策略(策略 1),  $w_{\max}=0.4$ ,  $w_{\min}=0.2$  的精细搜索策略(策略 2),  $w_{\max}=0.6$ ,  $w_{\min}=0.4$  的自适应速度更新策略(策略 3)的 PSO 算法, 进行 200 次优化仿真, 得到的统计结果如表 1。

表 1 3 种测试函数仿真结果  
Tab. 1 Results of three test functions

仿真结果	最小寻优精度	平均寻优精度	收敛效果	
函数 1	策略 1	9.3e-7	0.004 91	好
	策略 2	3.7e-7	0.005 91	一般
	策略 3	0	0.005 56	较好
函数 2	策略 1	0.000 16	0.060 29	好
	策略 2	0.000 08	0.107 72	较好
	策略 3	0	0.055 67	一般
函数 3	策略 1	3.354 88	11.712 0	较好
	策略 2	3.148 56	12.014 9	一般
	策略 3	0.000 17	1.758 46	好

对表中不同(高/低维, 单/多峰)函数的 3 种策略仿真结果进行比较, 可以发现, 3 种策略单次寻优精度优劣: 3>2>1; 平均寻优精度优劣: 1>2; 收敛性能优劣: 1>2。策略 3 若保持一定的速度进行快速搜索的时候是在最优解范围附近, 则加快了收敛速度, 提高了搜索的精度, 否则收敛性能差, 搜索精度低。

以 Sphere 为例, 对图 2 中 3 种策略在搜索过程中适应值的变化进行分析。可得: 策略 2、策略 3 的下降速度远大于策略 1。采用策略 2 时, 粒子在第 105 代时集中在较好的点附近进行局部搜索, 提高了搜索精度, 最终在 284 代找到最优解。在采用策略 3 时, 改进的速度更新公式提高了粒子朝着最优解的方向进行等速搜索的可能性, 从而使得粒子在 506 代寻到最优解, 搜索速度和搜索精度都得到提高。

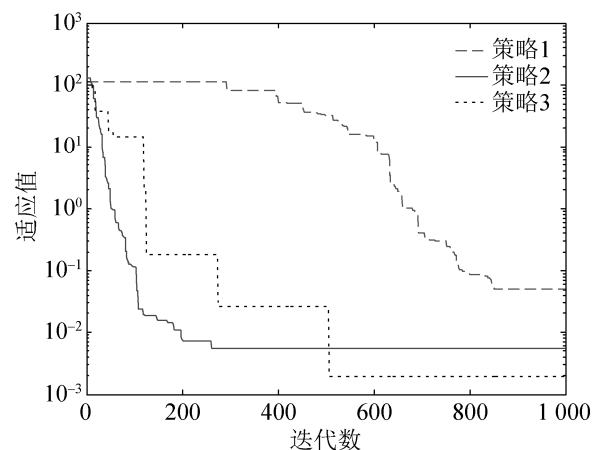


图 2 Sphere 中的对比情况

Fig. 2 Comparisons in sphere

因此, 在 3 个种群中分别采用大范围搜索策略(策略 1), 精细搜索策略(策略 2), 自适应速度更新策略(策略 3)进行协同搜索的改进 PSO 算法, 能够结合 3 种策略各自的优点, 弥补它们的缺点, 改善优化性能。

#### 3.2 月球软着陆仿真

我们将改进 PSO 算法运用到实际复杂问题上, 故采用月球软着陆问题进行仿真。月球软着



陆问题是非线性最优控制问题，不能直接应用粒子群算法，它的求解主要分为两个部分，问题转化和问题求解。首先，采用较简单的直接打靶法将原最优控制问题进行转化成参数为切比雪夫多项式插值点上的控制变量的非线性规划问题。然后利用粒子群优化算法进行求解转化后的问题即可得到解。

由参考文献[12]，月球软着陆模型表示为：

$$\begin{cases} \dot{r} = v \\ \dot{v} = \frac{F}{m} \sin \psi - \frac{\mu}{r^2} + r\omega^2 \\ \dot{\theta} = \omega \\ \dot{\omega} = -(\frac{F}{m} \cos \psi + 2v\omega) / r \\ \dot{m} = -F / C \end{cases} \quad (11)$$

式中： $r$  是着陆器与月球中心的距离； $v$  是着陆器在半径矢量方向上的速度； $\theta$  是着陆器方位角； $\omega$  是角速度； $m$  是着陆器质量； $\mu$  是月球的引力常数； $C$  是制动火箭的排气速度，是一个常数； $F$  是推力的大小，取 1500 N； $\psi$  是推力与当地水平面之间的夹角，取锐角； $\psi$  是控制变量，边界条件为： $0 \leq \psi \leq 0.785 \text{ rad}$ 。

最佳软着陆应满足如下条件： $r(0) = r_0$ ， $v(0) = 0$ ， $\theta(0) = 0$ ， $\omega(0) = \omega_0$ ， $m(0) = m_0$ ， $r(t_f) = r_f$ ， $v(t_f) = v_f$ ， $\omega(t_f) = 0$ ， $|r(t_f) - r_f| - \varepsilon_r \leq 0$ ， $|v(t_f) - v_f| - \varepsilon_v \leq 0$ ， $|\omega(t_f)| - \varepsilon_\omega \leq 0$ 。

具体参数如下： $\mu = 4.90275 \times 10^{12} \text{ m}^3 / \text{s}^2$ ， $C = 300 \times 9.8 \text{ m/s}$ ， $r_0 = 1753000 \text{ m}$ ， $m_0 = 600 \text{ kg}$ ， $\omega_0 = 9.64 \times 10^{-4} \text{ rad/s}$ ， $r_f = 1738000 \text{ m}$ ， $v_f = 0 \text{ m/s}$ ， $\varepsilon_r = 10 \text{ m}$ ， $\varepsilon_v = 0.5 \text{ m/s}$ ， $\varepsilon_\omega = 5 \times 10^{-4} \text{ rad/s}$ 。

根据上述条件进行优化仿真，以最小燃油消耗质量为性能指标，即  $J = \max m(t_f)$ ，考虑终端约束，适应度函数定义为：

$$c(1) = \begin{cases} 0, & \text{if } |r(t_f) - r_f| - \varepsilon_r < 0 \\ |r(t_f) - r_f| - \varepsilon_r \end{cases}$$

$$c(2) = \begin{cases} 0, & \text{if } |v(t_f) - v_f| - \varepsilon_v < 0 \\ |v(t_f) - v_f| - \varepsilon_v \end{cases}$$

$$c(3) = \begin{cases} 0, & \text{if } |\omega(t_f)| - \varepsilon_\omega < 0 \\ |\omega(t_f)| - \varepsilon_\omega \end{cases}$$

$$f(x) = -k_1 \cdot m(t_f) + k_2 \cdot c(1) + k_3 \cdot c(2) + k_4 \cdot c(3)$$

采用改进 PSO 算法进行优化，参数设置如下： $NP=60$ ， $MaxIter=300$ ， $c_1=1.8$ ， $c_2=2.2$ ， $k_1=0.002$ ， $k_2=10$ ， $k_3=200$ ， $k_4=100000$ 。

由于粒子群算法是随机搜索方法，将该算法多次运行，希望得到具有统计性的结果。再采用文献[9]基于变异策略改进、文献[10]多种群多策略改进的两种算法求解上述轨迹优化问题，得到文献[9-10]算法的统计结果。

3种算法所得到的统计结果如表2所示，其中， $m(t_f)_{\max}$  为终端质量的最大值， $m(t_f)_{\min}$  为终端质量的最小值， $\bar{m}(t_f)$  为终端质量的平均值。

表2 3种算法统计优化结果  
Tab.2 Statistical optimal results of three algorithms

结果	$m(t_f)_{\max}$	$m(t_f)_{\min}$	$\bar{m}(t_f)$
本文算法	420.918	386.735	406.562
文献[9]	405.612	344.898	371.33
文献[10]	414.796	362.245	389.8

分别选取3种算法中的单次优化结果，所得主要状态量与控制量曲线如图3~5所示，所得结果数据如表3所示。

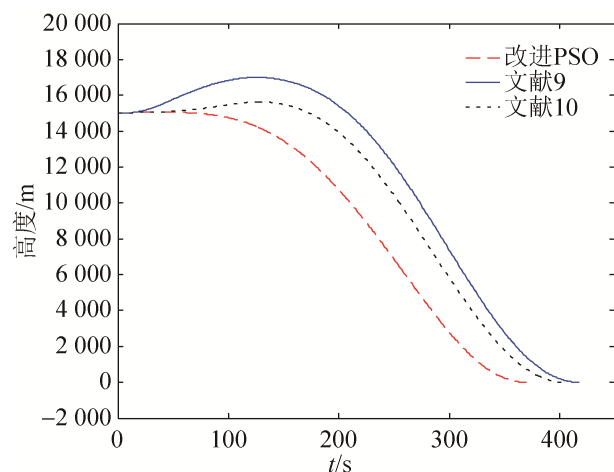


图3 3种算法高度对比图  
Fig.3 Height comparison chart of three algorithms



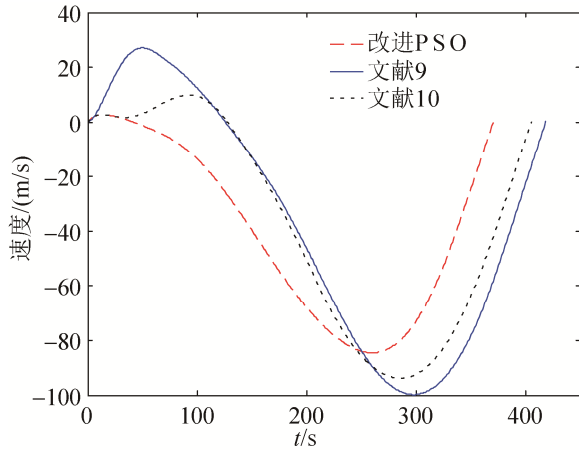


图4 3种算法速度对比图

Fig. 4 Velocity comparison chart of three algorithms

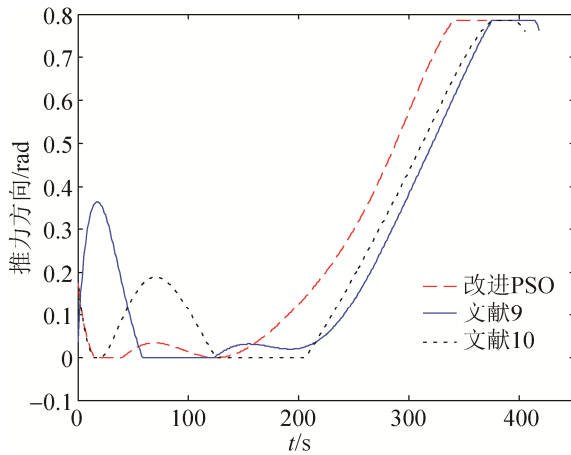


图5 3种算法控制量对比图

Fig. 5 Control variable comparison chart of three algorithms

表3 3种算法仿真结果

Tab. 3 Results of three algorithms

仿真结果	改进 PSO	文献[9]	文献[10]	要求
高度/m	0.007 60	0.097 01	0.063 57	0±0.1
速度/(m/s)	0.014 29	0.349 31	0.272 23	0±0.5
角速度/(rad/s)	0.000 39	0.000 29	0.000 32	0±0.000 5
质量/kg	411.224	386.735	393.367	/
飞行时间/s	370	418	405	/
优化耗时/s	537	639	512	/

通过优化仿真结果可以发现,在月球软着陆最小燃油消耗的轨迹优化问题上,本文提出的改进 PSO 算法和文献[9-10]的算法都能完成轨迹优化的各项指标要求。与文献[9-10]的算法相比,改进 PSO 算法消耗的质量最少,说明了改进 PSO 算法搜索最优解的能力强于文献[9-10]算法。并且改进 PSO

算法的收敛速度也更快,改进 PSO 算法在第 16 代已经快速找到较好的点,最终在 77 代找到最优解。综上所述,改进 PSO 算法的优化性能更好,验证了优化策略的有效性。

### 3.3 并行化仿真

利用改进 PSO 算法进行轨迹优化的时候,需要进行数值积分计算每一时刻的状态量,从而来计算粒子的适应值。这导致优化时间很长。由于 PSO 算法在计算适应值的过程中,各个粒子相互独立,所以本文提出采用 OpenMp 来实现 PSO 算法的并行化,从而减少优化耗时。

使用 OpenMP<sup>[13]</sup>的库函数必须将其头文件 <omp.h>包含在相应的程序中。编译指导语句的格式如下

```
#pragma omp<具体的编译指导语句>[clause[[], clause]...]
```

其中具体的编译指导语句包括 for, parallel, parallel for, sections, parallel sections, master, critical, flush, single, ordered 和 atomic。这些编译指导语句或者用来分配任务(比如 master, 分配主线程),或者用来同步(比如 parallel)。其中这 5 个编译指导语句(master, critical, flush, ordered 和 atomic)不能跟相应的子句。粒子群算法并行化采用的编译指导语句主要是 for。

PSO 算法首先对种群中所有粒子进行初始化,通过计算适应值选出个体最优解和全局最优解,通过公式进行位置、速度更新,迭代得到最优解。在这过程中,计算适应值选出个体最优解和全局最优解这一部分是通过大量循环迭代得到的,十分耗时。循环并行化是使用 OpenMP 并行化程序的最重要部分,在 OpenMP 编程模式下,使用编译指导语句能将循环中工作分配到一个线程组中,线程组中的每一个线程分担循环中的一部分内容,实现整个程序的并行化,即多个粒子同时进行适应值的计算和最优解的更新。种群中粒子更新局部最优和全局最优的并行实现如图 6 所示。

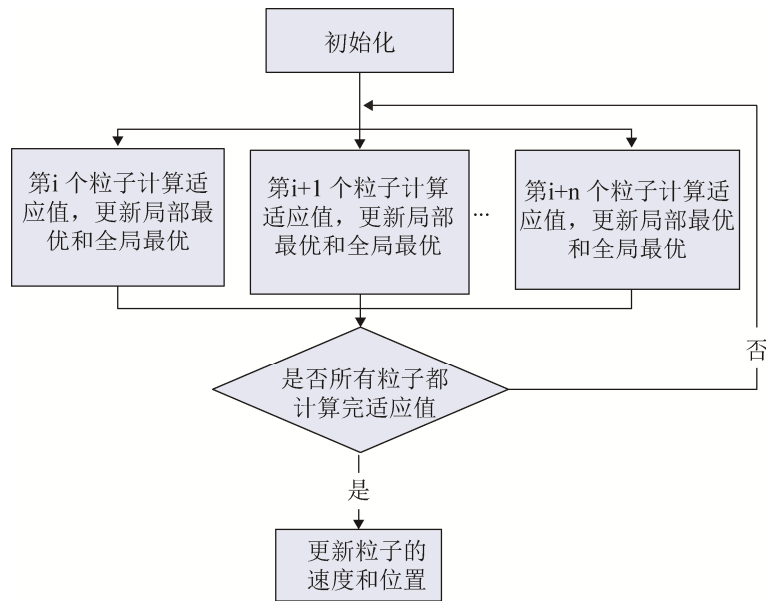


图 6 并行化流程图

Fig.6 Flow chart of parallelization

采用并行的基于柯西变异的多策略协同进化粒子群算法对上节月球软着陆最优控制问题分别采用单线程、多线程进行优化, 所得到的统计结果如表 4 所示。

表 4 并行化仿真结果

Tab. 4 Results of parallelization

仿真结果	单线程	多线程	要求
高度/m	0.061 35	0.053 18	0±0.1
速度/(m/s)	0.324 91	0.279 15	0±0.5
角速度/(rad/s)	0.000 367	0.000 367	0±0.0005
质量/kg	405.653 1	405.602 0	/
优化耗时/s	536	325.9	/

通过优化结果的对比可知, 采用不同的线程的粒子群算法对同一个轨迹优化问题进行优化, 所得到的结果都在目标允许的范围内, 并且精度相当, 但是多线程的算法在优化时间上有了很大的提高, 提高了算法的效率。

## 4 结论

为了提高粒子群算法的优化性能, 提出了一种基于柯西变异的多策略协同进化 PSO 算法, 算法通过一定的概率选中粒子进行柯西变异, 有助于粒

子跳出局部最优解, 未被选中的粒子按照所属种群的 3 种新进化策略更新位置和速度, 多种群粒子间共享信息, 提高了算法的收敛速度。算例仿真中, 首先应用到简单函数的优化上, 通过对三种典型函数的优化仿真, 验证了 3 种新策略分别具有开发能力强, 探测能力强, 快速探索的优点。再将本文提出的改进 PSO 算法和文献[7,10]的算法运用到实际复杂问题上, 通过对月球软着陆的优化仿真, 结果表明了改进算法优化性能有了较大的提高, 验证了策略的可行性。最后考虑到改进 PSO 算法的优化耗时, 对改进 PSO 算法进行基于 OpenMP 的并行化仿真, 结果表明并行化处理提高了算法的效率。

## 参考文献:

- [1] CR Bessette, DB Spencer. Optimal Space Trajectory Design: A Heuristic-Based Approach[C]//AAS/AIAA Space Flight Mechanics Meeting, 2006: 1-8.
- [2] 苏茂, 王永骥, 刘磊, 等. 基于改进粒子群算法的再入飞行器轨迹优化[J]. 计算技术与自动化, 2011, 30(4): 55-59.  
SU Mao, WANG Yongji, LIU Lei, et al. Design of Reentry Vehicle Trajectory Optimization Based on Improved Particle Swarm Optimization Algorithm[J]. Computer Technology and Automation, 2011, 30(4): 55-59.

- [3] 闫斌斌, 叶俊, 闫杰. 基于粒子群的高超声速飞行器模糊控制方法[J]. 科学技术与工程, 2012, 12(14): 3493-3496.  
YAN Binbin, YE Jun, YAN Jie. Fuzzy Control Method of Hypersonic Vehicle Based on Particle Swarm Optimization [J]. Science Technology and Engineering, 2012, 12(14): 3493-3496.
- [4] 王路, 邢清华, 毛艺帆. 再入高超声速滑翔飞行器轨迹快速优化[J]. 现代防御技术, 2015, 43(6): 74-80.  
WANG Lu, XING Qinghua, MAO Yifan. Fast Trajectory Optimization for Reentry Hypersonic Glide Vehicle[J]. MODERN DEFENCE TECHNOLOGY, 2015, 43(6): 74-80.
- [5] 陈上上, 何英姿, 刘贺龙. 基于粒子群优化的再入飞行器在线轨迹优化[J]. 上海航天, 2015, 32(6): 1-7.  
CHEN Shangshang, HE Yingzi, LIU Helong. Onboard Trajectory Planning for Entry Vehicle Based on Particle Swarm Optimization [J]. AEROSPACE SHANGHAI, 2015, 32(6): 1-7.
- [6] 冯林爽, 刘磊, 王永骥. 基于 PSO 和 SQP 混合优化算法的高超声速飞行器轨迹优化[C]. 第 27 届中国控制与决策会议, 2015: 4538-4542.  
Feng Linshuang, Liu Lei, Wang yongji. Trajectories optimization of hypersonic vehicle based on hybrid optimization algorithm of PSO and SQP[C]// Proceeding of 27th Chinese Control and Decision Conference (CCDC), 2015: 4538-4542.
- [7] 高立群, 吴沛峰, 邹德旋. 基于变异策略的粒子群算法[J]. 东北大学学报(自然科学版), 2010, 31(11): 1530-1533.  
GAO Liqun, WU Peifeng, ZOU Dexuan. Particle Swarm Optimization Based on Mutation Strategy [J]. Journal of Northeastern University(Natural Science), 2010, 31(11): 1530-1533.
- [8] 康岚兰, 董文永, 田降森. 一种自适应柯西变异的反向学习粒子群优化算法[J]. 计算机科学, 2015, 42(10): 226-231.  
KANG Lanlan, DONG Wenyong, TIAN Jiangsen. Opposition-based Particle Swarm Optimization with Adaptive Cauchy Mutation [J]. Computer Science, 2015, 42(10): 226-231.
- [9] 吴润秀, 孙辉, 朱德刚, 等. 具有高斯扰动的局部引导粒子群优化算法[J]. 计算机工程与科学, 2016, 36(6): 1183-1192.  
WU Runxiu, SUN Hui, ZHU Degang, et al. A particle swarm optimization algorithm based on local guidance and Gauss perturbation [J]. Computer Engineering and Science, 2016, 36(6): 1183-1192.
- [10] 徐冰纯, 葛洪伟, 王燕燕. 基于多种群多模型协同进化的粒子群优化算法[J]. 计算机工程, 2013, 39(5): 200-208.  
XU Bingchun, GE Hongwei, WANG Yanyan. Particle Swarm Optimization Algorithm Based on Multi-swarm and Multi-model Cooperative Evolution[J]. Computer Engineering, 2013, 39(5): 200-208.
- [11] 曾建潮, 介婧, 崔志华. 微粒群算法[M]. 北京: 科学出版社, 2004: 61-69.  
ZENG Chaojian, JIE Jing, CUI Zhihua. Particle Swarm Optimization [M]. Beijing: Science Press, 2004: 61-69.
- [12] X L Liu, G R Duan, K L Teo. Optimal soft landing control for moon lander [J]. Automatica (S0005-1098), 2008, 44(4): 1097-1103.
- [13] 刘凯, 寇正. OpenMP 在并行计算中的应用[J]. 微型机与应用, 2003, 22(12): 12-14.  
LIU Kai, KOU Zheng. Application of OpenMP in parallel computing [J]. Microcomputer and Application, 2003, 22(12): 12-14.