

基于多尺度分数阶多重记忆与学习的粒子群算法^{*}

董立军, 蒲亦非, 周激流

(四川大学 计算机学院, 成都 610065)

摘要: 针对粒子群优化算法在搜索高维多峰问题时容易出现种群多样性减少, 导致算法早熟收敛, 陷入局部最优等缺点, 提出了一种基于多尺度分数阶多重记忆与学习的粒子群算法。该算法将分数阶微积分引入标准粒子群速度、位置更新公式, 通过记忆粒子的历史速度、位置轨迹、个体最优轨迹、种群最优轨迹, 借助分数阶微积分具备的长期记忆特性来充分利用寻优过程中的历史信息, 增强算法的收敛速度和收敛精度。同时针对种群进化过程中出现的一些特殊情况, 采用多尺度分数阶和轨迹纠错学习策略来保护种群多样性, 减少算法陷入局部最优的可能。通过与其他粒子群改进算法的对比实验, 结果表明, 该算法具有较快的收敛速度和较高的收敛精度。

关键词: 粒子群优化算法; 多尺度分数阶; 多重记忆; 学习策略

中图分类号: TP18; TP301.6

文献标志码: A

文章编号: 1001-3695(2018)03-0661-05

doi:10.3969/j.issn.1001-3695.2018.03.005

Particle swarm optimization algorithm based on multi-scale fractional order multiple memory and learning

Dong Lijun, Pu Yifei, Zhou Jiliu

(College of Computer Science, Sichuan University, Chengdu 610065, China)

Abstract: In order to improve the problem that the particle swarm optimization algorithm is easy to reduce the population diversity, leading to premature convergence, and fall into the local optimum especially when the particle swarm optimization algorithm(PSO) searches the multi-peak problem in a high dimensional space, this paper developed an improved algorithm which called multi-scale fractional multiple memory and learning PSO(MML-PSO). It introduced the fractional calculus into the velocity and position updating formula, through remembering particle's historic velocity, position trajectory and individual optimal trajectory, population optimal trajectory, it used the long-term memory characteristics of fractional calculus to make full use of the historical information in the process of optimization and improved the convergence speed and convergence accuracy of the algorithm. At the same time, in order to solve some special cases in the process of population evolution, it put forward multi-scale fractional order and correcting trajectories learning strategy to protect the diversity of population and reduce the possibility of falling into local optimum. By comparing MML-PSO with other modified PSO algorithms, the results show that the proposed algorithm has fast convergence speed and high convergence precision.

Key words: particle swarm optimization(PSO); multi-scale fractional order; multiple memory; learning strategy

0 引言

粒子群优化算法(particle swarm optimization, PSO)是由 Eberhart 等人^[1]在 1995 提出的一种群体智能优化算法, 通过种群中粒子个体的简单运动行为和种群内信息的共享与交互来实现全局优化搜索。粒子群优化算法容易实现, 操作简单, 收敛速度快, 但是由于粒子都向最优粒子学习, 容易陷入局部最优, 导致早熟收敛。针对粒子群算法存在的一些缺点, 国内外的研究者提出了很多有效的改进措施, 主要可以分为以下四类:

a) 通过调整粒子群算法的参数来平衡算法的全局探测和局部开发能力。Shi 等人^[2-4]将惯性权重 w 引入速度更新公式, 使用惯性权重来调整算法收敛速度和收敛精度。Yang 等人^[5]通过对粒子群更新公式使用差分方程来分析算法的收敛条件, 根据收敛条件来指导惯性权重和加速因子的调整。Zhan

等人^[6]根据粒子群的进化状态, 自适应调整惯性权重、加速因子等参数。

b) 将其他进化算法或搜索策略等与粒子群结合。例如将遗传算法^[7]、差分进化^[8]等引入粒子群算法。Liu 等人^[9]将混沌搜索与粒子群结合, 增强粒子群的局部搜索能力。Pires 等人^[10]将分数阶微积分引入粒子速度更新公式, 使用分数阶阶次来控制算法的收敛速度。Gao 等人^[11]提出分数阶粒子群, 使用分数阶微积分来更新粒子速度和位置。Wang 等人^[12]引入多样性增强机制和邻域搜索策略来保护种群多样性, 减少算法的早熟收敛。

c) 使用不同的邻域拓扑结构、优化粒子学习方法来增强算法的性能。Kennedy 等人^[13]提出了几种基本的邻域结构, 如环形结构、星型结构和冯诺依曼结构等。Mendes 等人^[14]提出综合利用所有邻居粒子的最佳位置信息, 充分利用粒子间的信息。Liang 等人^[15]提出一种综合学习粒子群算法, 综合学习所

收稿日期: 2016-11-14; 修回日期: 2017-01-05 基金项目: 国家自然科学基金资助项目(61571312)

作者简介: 董立军(1991-), 男, 安徽池州人, 硕士研究生, 主要研究方向为智能信息处理(1598217726@qq.com); 蒲亦非(1975-), 男, 四川成都人, 教授, 博导, 主要研究方向为分数阶微积分在信号处理中的应用、模式识别、图像处理; 周激流(1963-), 男, 四川成都人, 教授, 博导, 主要研究方向为图像处理、人脸识别、无线网络、计算智能。

有粒子的不同维个体极值来更新速度。Chen 等人^[16]将老化机制引入粒子群算法,在保留原始粒子群算法快速收敛特性的同时,克服原始算法早熟收敛的缺点。Cheng 等人^[17]将社会学习机制引入粒子群算法,使用比当前粒子好的其他所有粒子的信息来更新当前粒子。

d)使用多种群方法来增强算法性能。Gülcü 等人^[18]在 CLPSO 的基础上提出了多种群并行的 CLPSO,多个种群基于主从模式,多个种群之间共享信息,协同合作。Liang 等人^[19]提出一种动态多种群粒子群优化算法(DMS-PSO),将整个粒子群分成若干个较小的粒子群,子种群之间相互交换信息并重组,动态改变种群拓扑结构,使得种群多样性增加,有利于解决复杂的多峰问题。

由此可见,针对粒子群算法的优化主要集中在如何克服粒子群算法的早熟收敛并保持快速收敛的特性。本文分数阶微积分引入粒子群算法,提出一种改进的粒子群算法——基于分数阶多重记忆与学习的粒子群算法(multiple memory and learning PSO, MML-PSO)。利用分数阶的长期记忆特性,记忆粒子的历史速度、位置轨迹、个体最优位置轨迹、种群最优位置轨迹,综合使用这些历史轨迹信息,利用分数阶来更新粒子的速度和位置。同时采用多尺度分数阶和轨迹纠错学习策略能够进一步保护种群多样性,减少陷入局部最优的概率,并且保留原始算法快速收敛的特性。

1 粒子群优化算法

粒子群优化算法是从自然界中鸟类觅食行为的模型中抽象出来的一种群体智能优化算法。粒子群中的每一个粒子都有速度和位置两个属性。速度表示粒子飞行的方向和位置改变的大小;位置则表示搜索空间的一个候选解。假定粒子群的种群大小为 N ,搜索空间的维数为 D ,则第 $i(i \leq N)$ 个粒子的速度和位置分别表示为 $V_i(v_i^1, v_i^2, \dots, v_i^D)$ 、 $X_i(x_i^1, x_i^2, \dots, x_i^D)$ 。在每一次的迭代过程中,粒子都根据自身的历史最优位置和种群最优位置来更新自己的速度和位置。第 $t+1$ 次迭代时,第 i 个粒子的速度和位置更新公式如式(1)所示。

$$V_i^j(t+1) = w \times V_i^j(t) + c_1 \times r_1 \times (P_i^j(t) - X_i^j(t)) + c_2 \times r_2 \times (G^j(t) - X_i^j(t)) \quad (1)$$

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

其中: $V_i^j(t)$ 表示第 t 次迭代时的第 i 个粒子第 j 维的速度; $X_i^j(t)$ 表示第 i 个粒子第 j 维的位置; $P_i^j(t)$ 表示第 i 个粒子历史个体最优位置的第 j 维; $G^j(t)$ 表示种群历史最优位置的第 j 维; w 表示惯性权重; c_1 、 c_2 分别为自身学习因子和社会学习因子; r_1 、 r_2 为服从 $[0,1]$ 均匀分布的随机数。

2 基于多尺度分数阶多重记忆粒子群

2.1 分数阶微分

分数阶微积分^[20]是整数阶微积分的自然推广。如果说整数阶微积分通常被看做是一种离散化的操作,那么相对的分数阶微积分则可以看做是一种连续性的操作。因为分数阶微分固有的长期记忆特性,相比整数阶微分,更适合应用于许多自然现象和不可逆的连续性过程^[21]。

分数阶微分的 Grünwald-Letnikov 定义如下:

$$D^\alpha[x(t)] = \lim_{h \rightarrow 0} \left[\frac{1}{h} \sum_{k=0}^{+\infty} \frac{(-1)^k \Gamma(\alpha+1)x(t-kh)}{\Gamma(k+1)\Gamma(\alpha-k+1)} \right] \quad (3)$$

从上面的定义就可以看出,分数阶微分与之前所有的历史状态有关,具有长期连续记忆特性。

离散形式的 Grünwald-Letnikov 定义如下:

$$D^\alpha[x(t)] = \frac{1}{T^\alpha} \sum_{k=0}^t \frac{(-1)^k \Gamma(\alpha+1)x(t-kT)}{\Gamma(k+1)\Gamma(\alpha-k+1)} \quad (4)$$

其中: T 为采样周期,在粒子群算法中,迭代一次粒子状态改变一次,因此 $T=1$; r 为截断阶次,实际应用中根据需要选择记忆前 r 次的历史状态,在本文算法中取 $r=4$ 。

2.2 分数阶多重记忆

由于粒子群算法是基于个体的生物种群长期动态进化寻优算法,分数阶微分固有的长期记忆特性十分适合描述粒子群的搜索寻优过程,所以文献^[10,11]将分数阶微积分引入粒子速度、位置更新公式,使用分数阶阶次来控制算法的收敛速度和搜索精度。根据式(1)可得

$$V_i(t+1) - V_i(t) = (w-1) \times V_i(t) + c_1 \times r_1 \times (P_i(t) - X_i(t)) + c_2 \times r_2 \times (G(t) - X_i(t)) \quad (5)$$

式(5)就是一阶微分形式,将其推广到分数阶微分则为

$$D^\alpha[V_i(t+1)] = (w-1) \times V_i(t) + c_1 \times r_1 \times (P_i(t) - X_i(t)) + c_2 \times r_2 \times (G(t) - X_i(t)) \quad (6)$$

根据式(4)(6)则有

$$V_i(t+1) = (w-1) \times V_i(t) + c_1 \times r_1 \times (P_i(t) - X_i(t)) + c_2 \times r_2 \times (G(t) - X_i(t)) + VV_i(t) \quad (7)$$

其中(取 $T=1, r=4$):

$$VV_i(t) = \alpha V_i(t) + \frac{1}{2}\alpha(1-\alpha)V_i(t-1) + \frac{1}{6}\alpha(1-\alpha)(2-\alpha)V_i(t-2) + \frac{1}{24}\alpha(1-\alpha)(2-\alpha)(3-\alpha)V_i(t-3) \quad (8)$$

若令惯性权重 $w=1$ (即原始的 PSO 算法,没有惯性权重),此时式(7)变为

$$V_i(t+1) = c_1 \times r_1 \times (P_i(t) - X_i(t)) + c_2 \times r_2 \times (G(t) - X_i(t)) + VV_i(t) \quad (9)$$

在式(9)中综合使用粒子前若干次的历史速度信息来更新当前的速度,具有明显的长期记忆能力,并且分数阶次 α 可以调节粒子历史速度对当前速度的影响程度,与惯性权重 w 具有相似的作用。同理对于式(2)的分数阶位置更新为

$$X_i(t+1) = V_i(t+1) + XX_i(t) \quad (10)$$

$$XX_i(t) = \beta X_i(t) + \frac{1}{2}\beta(1-\beta)X_i(t-1) + \frac{1}{6}\beta(1-\beta)(2-\beta)X_i(t-2) + \frac{1}{24}\beta(1-\beta)(2-\beta)(3-\beta)X_i(t-3) \quad (11)$$

从式(11)可以看出,粒子的位置更新也不仅仅依赖于前一次的粒子位置,而是依赖于粒子前若干次的历史位置,并且速度项也是由长期记忆的结果更新而来,即位置的更新实际上是粒子速度和位置双重长期记忆的结果。这样可以在一定程度上保护粒子种群的分布多样性,但是粒子的趋向同一性主要是由自身认知和社会认知部分决定的,粒子个体极值和种群极值会对粒子的运动产生强大的吸引力,种群多样性仍然会丧失。

因此考虑对粒子速度更新中的自身认知和社会认知部分引入分数阶,利用分数阶的长期记忆特性,记录粒子寻优轨迹,综合学习之前 r 次迭代的历史个体极值和历史种群极值,而并不是只由当前的个体极值和种群极值决定学习方向。

对于式(9)中的自身认知部分,令

$$PP_i(t) = P_i(t+1) - X_i(t+1) \quad (12)$$

根据式(4)(12),令 $D^\alpha[PP_i(t)] = 0$,即取 $P_i(t+1) - X_i(t+1)$ 的极值点,若使得下一次迭代时粒子位置与个体最优位置之间距离取得极值时应该满足何种关系,则

$$PP_i(t) = \gamma[P_i(t) - X_i(t)] + \frac{1}{2}\gamma(1-\gamma)[P_i(t-1) - X_i(t-1)] + \frac{1}{6}\gamma(1-\gamma)(2-\gamma)[P_i(t-2) - X_i(t-2)] + \frac{1}{24}\gamma(1-\gamma)(2-\gamma)(3-\gamma)[P_i(t-3) - X_i(t-3)] \quad (13)$$

若由式(13)中的 $PP_i(t)$ 替换式(9)中的 $P_i(t) - X_i(t)$, 此时速度更新方程为

$$V_i(t+1) = VV_i(t) + c_1 \times r_1 \times PP_i(t) + c_2 \times r_2 \times (G(t) - X_i(t)) \quad (14)$$

从式(14)中可以看出,此时粒子的自身认知是综合利用之前若干代的历史自身认知,而不仅仅是只根据当前的个体最优进行学习,充分利用粒子以前的经验进行学习。另外第 $t+1$ 次进行速度更新时, $p_i(t+1)$ 、 $X_i(t+1)$ 仍然是未知的,式(14)中则是相当于使用粒子以往的粒子经验、轨迹和学习方向来简单地预测未来的学习方向和位置,可以被看做是超前自身认知。同理,对于式(9)中的社会认知部分(下面 φ 为分数阶阶次):

$$GG_i(t) = \varphi[G(t) - X_i(t)] + \frac{1}{2}\varphi(1-\varphi)[G(t-1) - X_i(t-1)] + \frac{1}{6}\varphi(1-\varphi)(2-\varphi)[G(t-2) - X_i(t-2)] + \frac{1}{24}\varphi(1-\varphi)(2-\varphi)(3-\varphi)[G(t-3) - X_i(t-3)] \quad (15)$$

与粒子自身认知部分相同,若由式(15)中的 $GG_i(t)$ 替换式(14)中的 $G(t) - X_i(t)$, 此时粒子速度更新方程为

$$V_i(t+1) = VV_i(t) + c_1 \times r_1 \times PP_i(t) + c_2 \times r_2 \times GG_i(t) \quad (16)$$

从式(16)中可看出,粒子也综合利用之前若干代的历史社会认知,充分学习种群的历史经验。同时 $G(t+1)$ 、 $X_i(t+1)$ 也是未知的,相当于使用以往的粒子种群经验、轨迹和学习方向来简单地预测未来的学习方向,可以被看做是超前社会认知。

因此,本文提出的改进粒子群算法充分利用分数阶微分的记忆特性来实现对粒子寻优过程的速度、位置、个体极值、种群极值的多重记忆,更加切合粒子种群寻优过程。

2.3 轨迹纠错学习策略

如果当前粒子 i 为种群最优粒子,即 $X_i(t) = G(t)$, 则该粒子以前的搜索方向是正确的,继续学习自己的历史搜索方向,此时粒子速度更新公式变为

$$V_i(t+1) = VV_i(t) \quad (17)$$

如果当前粒子 i 不是种群最优粒子,即 $X_i(t) \neq G(t)$, 则该粒子之前的搜索路径和自身历史经验可能是错的,因此以一定的概率 Lp 学习比自己好的粒子或种群最优粒子。

若 $\text{rand} > Lp$, 则认为该粒子之前的搜索路径和历史经验是正确的,按式(19)来更新速度。

若 $\text{rand} < Lp$, 则认为该粒子之前的学习方向轨迹是错的,需要学习比自己好的其他粒子的个体历史最优轨迹。根据之前的 r 次迭代的所有粒子的个体历史最优轨迹,对于 $t_1(t_1 = t-1, t-2, \dots, t-r)$ 次迭代时粒子 i 的个体最优值用 t_1 次迭代时更好的某一粒子 j 的个体最优值去替换,即 $P_i(t-u) = P_{f(\text{rand})}(t-u)$, $u=0, 1, \dots, r-1$ 。其中 $f(\text{rand})$ 表示在第 $t-u$ 次迭代时个体最优比第 i 个粒子个体最优好的所有粒子中随机选取一个粒子。这样用其他较好的粒子最优轨迹重组、替换当前粒子的个体最优轨迹。显然,通过学习其他比自己好的粒子的历史个体极值,可以纠正该粒子之前已搜索过的错误的路线,使得该粒子之前的每一步都比自己原来的学习方向更加正确。

$$V_i(t+1) = VV_i(t) + c_1 \times r_1 \times PP_i(t) \cdots (\text{rand} < Lp) \quad (18)$$

$$V_i(t+1) = VV_i(t) + c_1 \times r_1 \times PP_i(t) + c_2 \times r_2 \times GG_i(t) \cdots (\text{rand} > Lp) \quad (19)$$

2.4 多尺度分数阶策略

某些特殊情况下,某一分数阶阶次可能使得 $VV_i(t)$ 、 $XX_i(t)$ 的值为零,导致粒子群收敛速度缓慢甚至停滞。随着种群搜索状态的改变,导致其值为零的阶次也在不断地改变。并且算法中含有 α 、 β 、 γ 、 φ 四个分数阶次参数,难以确定各个阶次之间的最优组合。

针对这两个问题,本文使用了多尺度分数阶策略。对于这四个阶次参数,因为 α 、 β 是通过控制迭代步长来调节算法性能,作用类似; γ 、 φ 是通过控制学习的程度来调节算法性能,作用相似,所以取 $\alpha = \beta$ 、 $\gamma = \varphi$ 即可视为两个阶次参数。为使算法每次迭代时都有不同的分数阶次,本文设计了两种动态阶次:

$$\text{alfa1} = 0.9 - 0.8 \times (t/\text{max_iter}) \quad (20)$$

$$\text{alfa2} = 0.1 + 0.8 \times (t/\text{max_iter}) \quad (21)$$

alfa1 、 alfa2 两两之间共有 $(\text{alfa1}, \text{alfa1})$ 、 $(\text{alfa1}, \text{alfa2})$ 、 $(\text{alfa2}, \text{alfa1})$ 、 $(\text{alfa2}, \text{alfa2})$ 四种组合,并随迭代过程动态变化,能够满足粒子群整个迭代过程中不同进化状态对分数阶阶次的需求。种群中每个粒子每次迭代时($\alpha = \beta$ 、 $\gamma = \varphi$)参数组合分别使用以上四种阶次组合来更新速度和位置,然后使用贪心策略选择其中最优化阶次组合的更新结果。不同阶次组合和多尺度策略对算法搜索的影响如图1所示。从图中可以看出,多尺度分数阶策略对于单峰和多峰函数的综合优化性能较好,收敛快、精度高。本文提出的基于多尺度分数阶多重记忆与学习的粒子群算法(MML-PSO),使用分数阶微分来更新粒子的速度和位置,利用分数阶固有的记忆特性,长期记忆粒子搜索过程中的速度、位置、个体历史最优、种群历史最优等轨迹。通过多重记忆,充分利用迭代过程中的历史搜索信息,并综合使用轨迹纠错学习和多尺度策略,减少算法陷入局部最优的可能性,提升算法的性能。MML-PSO 的算法流程如图2所示。

3 实验与分析

3.1 测试函数

为了测试本文提出的 MML-PSO 算法的性能,对六个标准测试函数^[16,22](表1)进行了对比实验。这些测试函数可以分为两组:第一组,单峰函数 $f_1 - f_3$, 函数较为简单,主要用来测试算法的收敛速度;第二组,多峰函数 $f_4 - f_6$, 这类函数有很多局部极小点,用来测试算法的全局收敛能力和跳出局部最优、避免早熟收敛的能力。表1中的搜索范围是指测试函数每一维变量的范围;最优值指的是测试函数的理论最小值;误差指的是算法优化结果与最优值的所允许的最大误差,若算法优化结果与最优值之差小于误差,则认为本次优化是成功的。

3.2 对比算法及参数设置

为了更直观地评价本文算法的性能,将本文算法与其他五种改进算法进行了对比实验。对比算法及参数设置如下:a)GPSO^[3],惯性权重 $0.9 \sim 0.4$;b)FIPS^[14];c)CLPSO^[15];d)ALC-PSO^[16];e)DMS-PSO^[19]。这五种改进算法包含了前文所述四个不同类别的 PSO 改进算法。

本文算法的参数设置为: $c_1 = c_2 = 2$, α 、 β 、 γ 、 φ 四个阶次使用多尺度策略,学习概率 $Lp = 0.9 - (0.9 - 0.4) \times (t/\text{max_iter})$,截断阶次(记忆长度) $r = 4$,粒子最大限制速度为搜索范围的15%。对比改进算法的参数设置与原文献保持一致。为

公平起见,所有对比算法的最大评价次数 $FEs = 200\ 000$,种群粒子数 $N = 20$,测试函数维数 $D = 30$,每个测试函数运行 30 次。单峰、多峰函数对比实验如表 2、3 所示。表 2、3 中的

mean 表示 30 次运行结果的平均值。其中,best 为 30 次中的最佳结果,dev 为标准差,suc% 为 30 次运行结果达到指定最大误差的成功率。

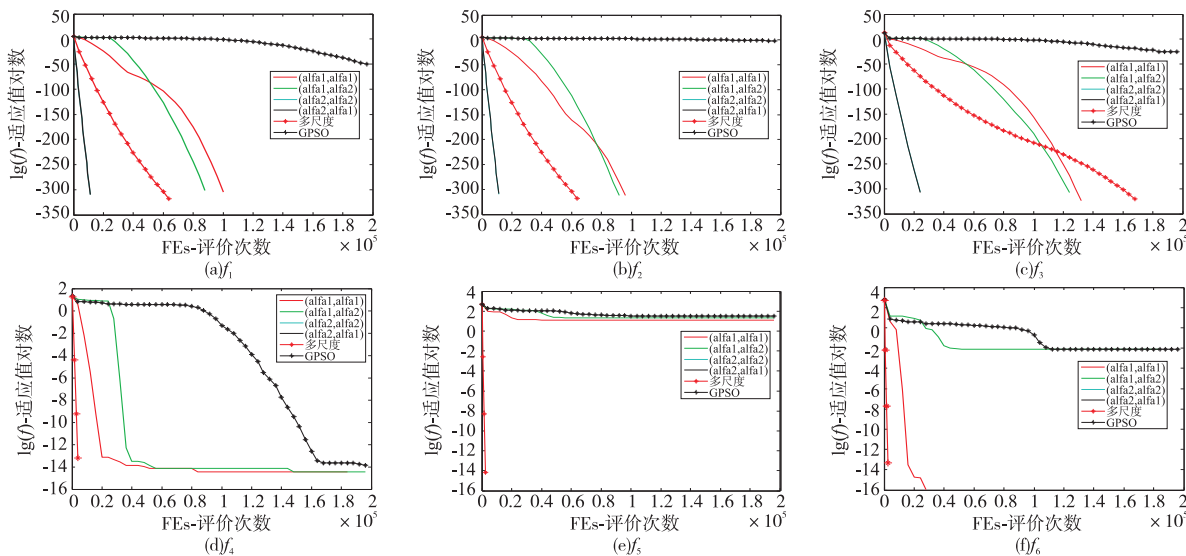


图1 不同阶次组合下不同测试函数的收敛曲线 ($N=20, D=30$)

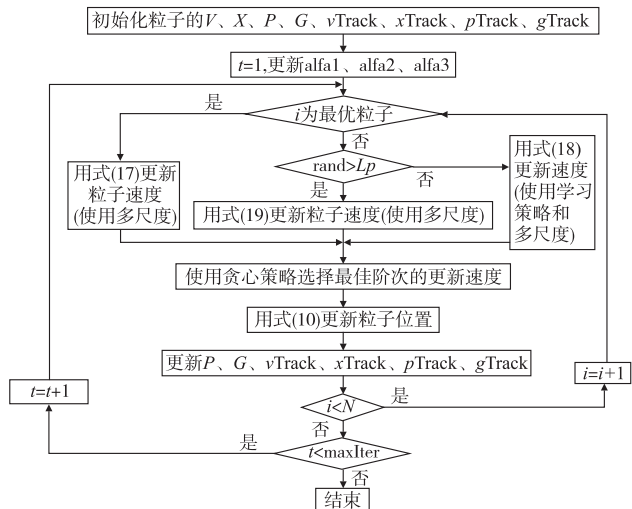


图2 MML-PSO算法流程

3.3 实验结果及分析

1) 单峰函数对比实验 主要测试算法的收敛速度和收敛精度。a) 收敛速度上:从图 2(a) ~ (c) 可以看出,本算法在单峰函数上的收敛速度明显快于 GPSO,可见在单峰函数

上本算法不仅保留,甚至还增强了标准 PSO 算法的快速收敛特性;b) 收敛精度上:从表 2 中可以看出,本文提出的 MML-PSO 算法在 f_1 、 f_2 、 f_3 单峰函数上均取得了理论最小值,显然 MML-PSO 在单峰函数优化上具有良好的性能,而且在三个单峰测试函数上都能以 100% 的成功率达到指定的误差精度,甚至在 30 次测试实验中均能达到最小值 0。因此可以说 MML-PSO 在单峰函数的优化上既能保持快速收敛,同时大幅提高收敛精度。

2) 多峰函数对比实验 主要测试算法的收敛精度以及跳出局部最优,避免早熟收敛的性能。a) 在收敛速度上:从图 1(d) ~ (f) 可以看出,本算法对于多峰函数优化的收敛速度明显比 GPSO 要快得多,可见 MML-PSO 在多峰函数优化上依然保持并增强了粒子群算法的快速收敛特性;b) 收敛精度上:从表 3 可以看出,MML-PSO 在选取的三个多峰函数上均取得了理论最优值,明显优于其他算法,所提算法的收敛精度高且不易陷入局部最优,且 30 次测试实验中每次均能收敛到最小值 0,尤其在 f_5 、 f_6 测试函数上,某些对比算法收敛精度较差,寻优结果精度达到指定误差的成功率低于 100% 甚至为 0。

表 1 标准测试函数

	测试函数	搜索范围	维数	误差	最优值
Sphere	$f_1(x) = \sum_{i=1}^n x_i^2$	$[-100, 100]$	30	0.01	0
Quadric	$f_2(x) = \sum_{i=1}^n (\sum_{j=1}^i x_j)^2$	$[-100, 100]$	30	100	0
Schwefel's P2.22	$f_3(x) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	$[-10, 10]$	30	0	0
Ackley	$f_4(x) = -20 \exp(-0.2 \sqrt{1/n \sum_{i=1}^n x_i^2}) - \exp(1/n \sum_{i=1}^n \cos(2\pi x_i)) + 20 + e$	$[-32, 32]$	30	0.01	0
Rastrigin	$f_5(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	$[-5.12, 5.12]$	30	10	0
Griewank	$f_6(x) = 1/4000 \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos(x_i/\sqrt{i}) + 1$	$[-600, 600]$	30	0.01	0

4 结束语

为了更好地平衡粒子群算法的全局搜索和局部搜索能力,缓解收敛速度与早熟收敛之间的矛盾,进一步提升算法的性能,

本文提出了 MML-PSO 算法。实验结果表明,本文提出的算法在单峰和多峰测试函数上均有较快的收敛速度和较高的收敛精度,部分函数能以较高的概率收敛到理论最优。后续将继续研究本算法中各分数阶次之间的关系以及对算法性能的影响。

表2 单峰函数对比实验

测试函数		MML-PSO	GPSO	ALC-PSO	CLPSO	FIPS	DMS-PSO
f_1	mean	0	7.08e-53	1.68e-161	1.39e-27	2.60e-30	1.95e-54
	best	0	2.02e-56	1.14e-172	2.57e-29	5.63e-31	2.08e-61
	dev	0	1.71e-52	8.21e-161	2.05e-27	3.24e-30	8.43e-54
	suc/%	100	100	100	100	100	100
f_2	mean	0	5.76e-02	1.79e-11	1.06e+02	3.29e+00	2.38e+01
	best	0	2.94e-03	8.27e-14	3.19e+01	3.46e+00	3.09e-01
	dev	0	6.16e-02	3.54e-11	5.04e+01	2.30e+00	2.84e+01
	suc/%	100	100	100	100	100	100
f_3	mean	0	3.13e-34	1.16e-90	3.58e-17	1.61e-17	2.39e-27
	best	0	8.57e-38	1.12e-98	1.89e-17	5.18e-18	4.15e-34
	dev	0	6.41e-34	4.15e-90	1.12e-17	8.59e-18	1.19e-26
	suc/%	100	100	100	100	100	100

表3 多峰函数对比实验

测试函数		MML-PSO	GPSO	ALC-PSO	CLPSO	FIPS	DMS-PSO
f_4	mean	0	1.10e-14	1.15e-14	2.49e-14	7.58e-15	9.23e-15
	best	0	7.69e-15	7.69e-15	1.84e-14	4.14e-15	7.69e-15
	dev	0	2.27e-15	2.94e-15	4.18e-15	6.49e-16	1.79e-15
	suc/%	100	100	100	100	100	100
f_5	mean	0	2.52e+01	2.53e-14	2.44e-14	2.87e+01	2.78e+01
	best	0	1.49e+01	7.11e-15	0	7.71e+00	1.29e+01
	dev	0	5.21e+00	1.38e-14	5.98e-14	1.46e+01	7.57e+00
	suc/%	100	100	100	100	100	100
f_6	mean	0	1.65e-02	1.22e-02	2.01e-14	2.47e-04	1.07e-02
	best	0	0	0	0	0	0
	dev	0	1.69e-02	1.58e-02	8.67e-14	2.47e-04	1.60e-02
	suc/%	100	53.3	60	100	100	70

参考文献:

- [1] Eberhart R C, Kennedy J. A new optimizer using particle swarm theory[C]//Proc of the 6th International Symposium on Micro Machine and Human Science. 1995;39-43.
- [2] Shi Yuhui, Eberhart R. A modified particle swarm optimizer[C]//Proc of IEEE International Conference on Evolutionary Computation Proceedings & IEEE World Congress on Computational Intelligence. 1998; 69-73.
- [3] Shi Yuhui, Eberhart R C. Empirical study of particle swarm optimization[C]//Proc of Congress on Evolutionary Computation. 1999.
- [4] Shi Yuhui, Eberhart R C. Fuzzy adaptive particle swarm optimization [C]//Proc of Congress on Evolutionary Computation. 2001; 101-106.
- [5] Yang Qiaohu, Tian Jinpeng, Si Wen. An improved particle swarm optimization based on difference equation analysis [J]. Journal of Difference Equations & Applications, 2016,181(6):1-18.
- [6] Zhan Zhihui, Zhang Jun, Li Yun, *et al.* Adaptive particle swarm optimization[J]. IEEE Trans on Systems, Man, and Cybernetics, Part B: Cybernetics, 2009,39(6):1362-1381.
- [7] Juang C F. A hybrid of genetic algorithm and particle swarm optimization for recurrent network design[J]. IEEE Trans on Systems, Man, and Cybernetics, Part B: Cybernetics, 2004,34(2):997-1006.
- [8] Zhang Wenjun, Xie Xiaofeng. DEPSO: hybrid particle swarm with differential evolution operator[C]//Proc of IEEE International Conference on Systems Man and Cybernetics. 2003;3816-3821.
- [9] Liu Bo, Wang Ling, Jin Yihui, *et al.* Improved particle swarm optimization combined with chaos[J]. Chaos, Solitons & Fractals, 2005,25(5):1261-1271.
- [10] Pires E J S, Machado J A T, De Moura O P B, *et al.* Particle swarm optimization with fractional-order velocity [J]. Nonlinear Dynamics, 2010,61(1-2):295-301.
- [11] Gao Zhe, Wei Junxiu, Liang Chao, *et al.* Fractional-order particle swarm optimization[C]//Proc of the 26th Chinese Control and Decision Conference. 2014;1284-1288.
- [12] Wang Hui, Sun Hui, Li Changhe, *et al.* Diversity enhanced particle swarm optimization with neighborhood search[J]. Information Sciences, 2013,223:119-135.
- [13] Kennedy J, Mendes R. Population structure and particle swarm performance[C]//Proc of Congress on Evolutionary Computation. [S. l.]: IEEE Computer Society, 2002;1671-1676.
- [14] Mendes R, Kennedy J, Neves J. The fully informed particle swarm: simpler, maybe better[J]. IEEE Trans on Evolutionary Computation, 2004,8(3):204-210.
- [15] Liang Jing, Qin Kai, Suganthan P N, *et al.* Comprehensive learning particle swarm optimizer for global optimization of multimodal functions[J]. IEEE Trans on Evolutionary Computation, 2006,10(3):281-295.
- [16] Chen Weineng, Zhang Jun, Lin Ying, *et al.* Particle swarm optimization with an aging leader and challengers[J]. IEEE Trans on Evolutionary Computation, 2013,17(2):241-258.
- [17] Cheng Ran, Jin Yaochu. A social learning particle swarm optimization algorithm for scalable optimization[J]. Information Sciences, 2015,291:43-60.
- [18] Gülcü S, Kodaz H. A novel parallel multi-swarm algorithm based on comprehensive learning particle swarm optimization[J]. Engineering Applications of Artificial Intelligence, 2015,45:33-45.
- [19] Liang Jing, Suganthan P N. Dynamic multi-swarm particle swarm optimizer with local search[C]//Proc of IEEE Congress on Evolutionary Computation. 2005; 522-528.
- [20] Podlubny I. Fractional differential equations: an introduction to fractional derivatives, fractional differential equations, to methods of their solution and some of their applications [M]. [S. l.]: Academic Press, 1998.
- [21] Sabatier J, Agrawal O P, Machado J A T. Advances in fractional calculus[M]. Dordrecht: Springer, 2007.
- [22] Suganthan P N, Hansen N, Liang Jing, *et al.* Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization[R]. 2005;341-357.