

基于改进 PSO 算法的 Rosenbrock 函数优化问题的研究

邵 鹏 吴志健

(武汉大学计算机学院 武汉 430072) (武汉大学软件工程国家重点实验室 武汉 430072)

摘 要 Rosenbrock 函数优化属于无约束函数优化问题,其全局极小值位于一条平滑而狭长的抛物线形状的山谷底部,且为优化算法提供的信息很少,因此找到其全局极小值就显得很困难。根据 Rosenbrock 函数的这种特性,专门提出了一种改进的 PSO 算法(PSO-R),该算法引入三角函数因子,利用三角函数具有的周期振荡性,使每个粒子获得较强的振荡性,扩大每个粒子的搜索空间,引导粒子向全局极小值附近靠近,避免算法过早地收敛,陷入局部最优,从而找到 Rosenbrock 函数的全局极小值。大量实验结果表明,该算法具有很好的优化性能,为某些领域某些特定的类似于 Rosenbrock 函数的优化问题提供了一种新的思路。

关键词 无约束优化, Rosenbrock 函数, 粒子群算法, 三角函数因子

中图分类号 TP18 **文献标识码** A

Rosenbrock Function Optimization Based on Improved Particle Swarm Optimization Algorithm

SHAO Peng WU Zhi-jian

(School of Computer, Wuhan University, Wuhan 430072, China)

(State Key Lab of Software Engineering, Wuhan University, Wuhan 430072, China)

Abstract Rosenbrock function optimization belongs to unconstrained optimization problems, and its global minimum value is located in the bottom of a smooth and narrow valley of the parabolic shape. It is very difficult to find the global minimum value because of little information the function optimization algorithm provided. According to the characteristics of the Rosenbrock function, this paper specifically proposed an improved particle swarm optimization(PSO) algorithm(PSO-R) which introduces trigonometric factor with periodic oscillations of trigonometric functions so that each particle gets strong oscillation to expand the search space of each particle and guide particles to close nearly the optimal value and avoid converging prematurely and local optimum, and find the global minimum of the Rosenbrock function. A large number of experimental results show that the algorithm has good performance of function optimization, and provides a new idea for optimization problems similar with the Rosenbrock function for some problems of special fields.

Keywords Unconstrained optimization, Rosenbrock function, PSO algorithm, Trigonometric factor

1 引言

无约束函数优化是当今信息时代应用领域很广泛的一类优化问题,对此问题的研究也随着人们实际的需要以及计算机的快速发展而逐渐深入。经过大量的研究,学者们已经提出了许多无约束最优化的方法,如基于导数的最速下降法、共轭梯度法、牛顿法等;基于群智能的粒子群算法、遗传算法等搜索算法,这些算法都具有很好的优化性能且都是鲁棒性较好的算法。寻找具有鲁棒性的算法解决复杂的现实问题是一种常见的方法,但许多领域中的问题都是某一领域具有该领域特征的问题,当具有较好鲁棒性的算法难以解决或无法解决该问题时可以尝试从该问题在该领域中所具有的特性出发寻找解决问题的新方法、新思路。

在函数极小化问题中,目标函数等高面内经常出现(至少

在局部范围内)“超山谷”的现象,这种形状类似于向上开口的抛物线,目标函数的全局极小值位于狭长山谷的底部^[1]。优化算法应该具有沿着狭长的山谷逐渐逼近目标函数全局极小值的能力,而由 Rosenbrock 设计的 Rosenbrock 函数是测试优化算法是否具有这种能力的一个很典型的测试函数^[2]。现在许多优化算法对大部分测试函数能达到很好的优化性能,但大量文献表明许多高效的算法甚至许多智能搜索算法对 Rosenbrock 函数的优化都难以找到全局极小值,如梯度法、遗传算法^[1]、由 Kennedy J 和 Eberhart RC. 提出的粒子群算法(PSO)^[3]、由王晖和吴志健等提出的基于一般反向学习的 PSO 算法(GOPSO)^[4]等。

目前,针对 Rosenbrock 函数的特性设计优化算法的研究很少,梁春艳等人于 1997 年提出基于遗传算法的 Rosenbrock 函数的优化问题的研究^[1]。根据该函数的特性,提出了一种

到稿日期:2012-11-13 返修日期:2013-03-1 本文受国家自然科学基金(61070008),中央高校基本科研业务费专项资金(2012211020205)资助。

邵 鹏(1983-),博士生,主要研究方向为智能计算、智能信息处理,E-mail:sp198310@163.com;吴志健(1963-),教授,博士生导师,主要研究方向为演化计算、智能计算、反问题、并行计算。

改进的 PSO 算法 (PSO-R) 并对该函数进行优化。实验结果表明,该算法具有很好的优化性能,为类似于该函数的优化问题提供了一种新的思路,所以对该函数的优化是很有现实意义的。

2 Rosenbrock 函数特性分析

Rosenbrock 函数是无约束最优化理论与方法中一个非常经典的函数测试问题,它是衡量无约束算法优劣的一个重要工具。在数值优化领域中,该函数是由 Howard H. Rosenbrock 于 1960 年提出来的,是一个典型的非凸函数,主要用于优化算法的性能测试,并以 Rosenbrock 山谷函数或 Rosenbrock 香蕉函数的别名而闻名^[2]。

2.1 Rosenbrock 函数定义

Rosenbrock 函数常见的定义分为两个变量和随机化高维扩展两种。

两个变量 Rosenbrock 函数定义如下:

$$f(x, y) = (1-x)^2 + 100(y-x^2)^2 \quad (1)$$

式中,函数的全局极小值点 $(x, y) = (1, 1)$,第二项 $100(y-x^2)^2$ 中系数 100 是可变的,并且它的改变不会影响该函数的全局极小值^[2]。两个变量的 Rosenbrock 函数的三维图如图 1 所示。

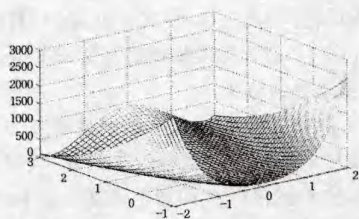


图 1 Rosenbrock 函数三维图

Rosenbrock 函数有许多随机化高维扩展函数的定义形式,常见的高维扩展的定义^[5]如下:

$$f(x_i) = \sum_{i=1}^{n-1} [(1-x_i)^2 + 100(x_{i+1}-x_i^2)^2] \quad (2)$$

式中, $x_i \in [-2.048, 2.048]$ 。

Rosenbrock 函数在 $(x_1, x_2, \dots, x_n) = (1, 1, \dots, 1)$ 处可找到全局极小值 0,但是由于它随机的特性,任何基于下降梯度的优化算法都没能找到全局极小值^[1]。高维扩展的 Rosenbrock 函数等高线图如图 2 所示。

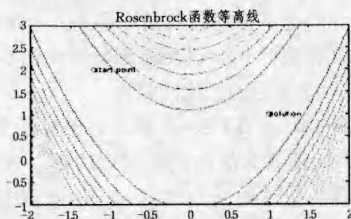


图 2 Rosenbrock 函数等高线图

另外,其高维扩展函数还具有“驼峰”结构^[6],如图 3 所示。

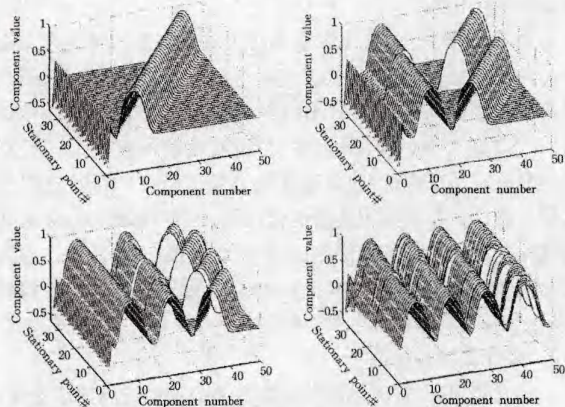


图 3 Rosenbrock 函数的“驼峰”结构图¹⁾

2.2 Rosenbrock 函数的特性分析

Rosenbrock 函数是一种无约束函数优化问题,在维数大于 3 的情况下呈现出多峰函数的特性,而在其它维数下呈现出单峰不可分函数的特性^[2]。

结合图 1 和图 2 可以看出,Rosenbrock 函数的全局极小值位于一条平滑而狭窄的抛物线形状的山谷底部。由于该函数为优化算法提供的搜索信息比较少,使一般的优化算法很难辨别搜索方向,查找到全局极小值也就变得困难。从图 3 可以看出,Rosenbrock 函数具有“驼峰”结构,“驼峰”之间有一条狭长的山谷,从侧面能看到形如图 2 的等高线的形状,函数的全局极小值点位于山谷底部。在算法对 Rosenbrock 函数进行优化时,很容易在山谷某一侧面陷入局部最优而很难跳出。为了使算法在陷入局部最优后跳出,可以扩大算法的搜索空间并增加搜索信息。对于粒子群优化算法来说,可以通过增强每个粒子的振荡性来扩大每个粒子的搜索空间,使粒子有更大的概率搜索到全局极小值所在的狭长的山谷底部,找到全局极小值。

3 PSO 算法及其改进算法

3.1 PSO 算法

粒子群算法 (particle swarm optimization, PSO) 最早是在 1995 年由美国社会心理学家 Kennedy 和电气工程师 Eberhart 共同提出的^[3]。PSO 算法的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结果的启发^[7]。PSO 算法的数学模型由速度更新方程和位置更新方程两部分组成。

速度更新方程如下:

$$v_i(t+1) = \omega v_i(t) + c_1 r_1 (pbest_i - x_i(t)) + c_2 r_2 (gbest - x_i(t)) \quad (3)$$

位置更新方程如下:

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (4)$$

式(3)中, $v_i(t+1)$ 为粒子 i 在第 $t+1$ 代的速度; $x_i(t+1)$ 为粒子 i 在第 $t+1$ 代的位置; c_1 和 c_2 分别为认知系数和社会系数 (加速因子); ω 为惯性权重; r_1 和 r_2 为 $[0, 1]$ 之间服从均匀分布的随机数; $pbest_i$ 为第 i 个个体历史最优位置, $gbest$ 为群体

1) 该图作者 Chthonicdaemon。

历史最优位置。

在 PSO 算法中,每个粒子具有位置 $x_i(t+1)$ 和速度 $v_i(t+1)$ 两个参量,每个粒子代表搜索空间中一个可能解,粒子通过自身的经验值和其它粒子的经验值来调整自身的搜索策略。PSO 算法简单、容易实现,在初期收敛速度快但在后期很容易陷入局部最优,收敛速度慢且精度低。PSO 算法对于大部分测试函数具有很好的优化性能,但对 Rosenbrock 函数的优化却易陷入局部最优,难以找到全局极小值,甚至许多改进的高效的 PSO 算法对 Rosenbrock 函数的优化也难以找到全局极小值,所以设计一种有效的改进 PSO 算法很有必要。

3.2 PSO 算法的改进

在现实中,经常看见附着在弹簧上的物体的周期性振荡、挂在绳子上物体的小角度周期性摆动,这些自然现象在物理学上称之为简谐运动,而在数学上可以用三角函数来描述这种运动,即三角函数具有周期振荡的特性^[8]。

为了有效地避免 PSO 算法陷入局部最优,引入三角函数因子,利用三角函数所具有的周期性振荡的特性使每个粒子获得振荡,从而扩大粒子的搜索空间,跳出局部最优,找到全局极小值。

根据前文所述算法原理并结合 Rosenbrock 函数所具有的特性,提出一种针对该函数优化的改进 PSO 算法,简称为 PSO-R 算法。在 PSO-R 算法中引入两种简单的三角函数,它们分别为正弦函数 $\sin x$ 和余弦函数 $\cos x$ 。对于正弦函数 $\sin x$,在一个周期内其函数曲线具有一个极大值和一个极小值,如图 4 所示。

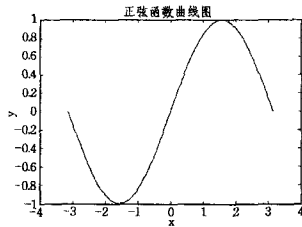


图 4 正弦函数曲线图

在 PSO-R 算法中, $\sin x$ 中的 x 的取值是在一定的范围内随机产生的,因此,结合图 4 可以推出,随机的 x 值相对应的 $\sin x$ 的值跳跃性大,即在该算法中粒子获得的振荡性大,从而增加粒子跳出局部最优的概率。然而,通过实验表明,过于强烈的振荡性会使粒子难以找到全局极小值,实验中采用缩小 x 取值范围的方法来减缓粒子的振荡性。由于 $\cos x$ 可以转为 $\sin x$ 故不讨论。

考虑以下 PSO 算法模型:

速度更新方程如下:

$$v_i(t+1) = wv_i(t) + c_1 r_1 (pbest_i - x_i(t)) + c_2 r_2 (gbest - x_i(t)) \quad (5)$$

位置更新方程如下:

$$x_i(t+1) = x_i(t)(1 + \cos(h)) + v_i(t+1)\sin(h) + 1 \quad (6)$$

式(6)中, $\sin(h)$ 、 $\cos(h)$ 分别称为正弦函数因子和余弦函数因子,统称三角函数因子, h 为待定参数。式中第一项为对粒子的位置振荡;第二项为对粒子的速度振荡;第三项为修正因子,主要对优化精度进行修正,其作用是粒子在参数 h 的引导下向全局极小值靠近。

PSO-R 算法优化 Rosenbrock 函数的实现流程与 PSO 算法类似,简述如下:

- 步骤 1 初始化,包括 w 、 c_1 、 c_2 、最大速度上限 $V_{\max}$ 、 r_1 、 r_2 、 h 等参数的初始化并使粒子随机均匀地分布于搜索空间中;
- 步骤 2 由式(3)或(5)计算粒子下一代的的速度;
- 步骤 3 根据式(6)计算粒子下一代的位置;
- 步骤 4 根据式(2)计算每个粒子的适应值;
- 步骤 5 更新各粒子的个体历史最优位置和群体历史最优位置:比较粒子 i 当前适应值与该粒子历史适应值,记录最优位置 $pbest_i$;比较粒子 i 当前适应值与群体历史最优位置,记录最优位置 $gbest$;
- 步骤 6 达到控制条件或迭代次数,停止计算;否则,转步骤 2。

4 实验结果与性能分析

4.1 参数设置

实验中,为了与所提算法的性能进行比较,选用 5 个已有的对 Rosenbrock 函数的优化都不是很理想的改进 PSO 算法与 PSO-R 共 6 个算法进行性能测试比较,它们分别是:标准权重的 PSO 算法(PSO-w)^[9]、统一的粒子群优化算法(UPSO)^[10]、综合学习 PSO 算法(CLPSO)^[11]、带柯西变异的反向学习 PSO 算法(OPSO)^[12]、一般反向学习的 PSO 算法(GOPSO)^[4] 和 PSO-R 算法。前面 3 种算法中的参数设置与参考文献[9]中的设置一样,后 3 种算法参数设置如下: $c_1 = c_2 = 1.49618$,惯性权重 $w = 0.72984$,最大速度 $V_{\max} = 2.0$; r_1 和 r_2 为 $[0, 1]$ 之间服从均匀分布的随机数;所有算法中种群大小为 40,最大迭代次数为 200000,运行次数为 30,维数为 30;本算法中参数 h 初始范围为 $(-\pi, \pi)$ (不包括 0),其中 π 为圆周率,初始值为 3.141592653;误差精度为 $1.27e-7$ 。

4.2 实验结果与性能分析

4.2.1 参数 h 取值范围为 $(-\pi, \pi)$ 的实验

本次实验中,参数的设置如前文所述。PSO-R 算法对 Rosenbrock 函数的优化以及其它 5 种算法对 Rosenbrock 函数的优化结果如表 1 所列。表中最好结果加粗并用下划线标记。

表 1 6 种函数对 Rosenbrock 函数的优化结果

算法	平均值	标准方差
PSO-W	2.93e+001	2.51e+001
UPSO	1.51e+001	8.14e-001
CLPSO	2.01e+001	2.98e+000
OPSO	9.78e+001	6.33e+000
GOPSO	1.48e+001	9.57e-001
PSO-R	<u>2.55e-003</u>	<u>4.78e-003</u>

参考文献[4,9-12]表明对应的算法对大部分函数的优化性能都很好,但对 Rosenbrock 函数的优化存在问题,从表 1 中可以看出,无论是从平均值的角度还是从标准方差的角度来分析,PSO-R 算法对 Rosenbrock 函数的优化性能明显优于其它 5 种函数对该函数的优化性能。

同时,大量实验表明 PSO-R 算法的鲁棒性弱,但对于类似于 Rosenbrock 函数的优化性能相对较好。为了验证该结论,实验中选择两个结构上类似 Rosenbrock 函数的函数,它们分别为 Extended Tridiagonal-1 函数和 Extended White & Holst 函数,其函数形式如式(7)和式(8)所示。

$$f(x_i) = \sum_{i=1}^{n/2} [(x_{2i-1} + x_{2i} - 3)^2 + (x_{2i-1} - x_{2i} + 1)^4] \quad (7)$$

$$f(x_i) = \sum_{i=1}^{n/2} [100(x_{2i} - x_{2i-1}^3)^2 + (1 - x_{2i-1})^2] \quad (8)$$

式(7)的函数的极值点为 $(1, 2, 1, 2, \dots, 1, 2)^T$,极小值为

0;式(8)的函数的极值点为 $(1,1,\dots,1)^T$,极值点为0。

实验中选择两个代表性的算法(PSO-w 算法和 GOPSO 算法)以及 PSO-R 算法分别对上述两个函数进行测试,结果如表 2 所列。

表 2 不同函数的实验结果

函数 算法	Extended Tridiagonal-1		Extended White & Holst	
	平均值	标准方差	平均值	标准方差
PSO-W	8.29e+000	3.53e+000	7.72e+000	3.85e+000
GOPSO	9.88e+000	3.77e+000	7.51e+000	3.81e+000
PSO-R	4.27e+000	1.46e+000	5.25e-002	6.88e-002

从表 2 中可以看出,PSO-R 算法对两个函数的优化相对来说均好于其它两种算法,且对 Extended White & Holst 函数的优化性能明显好于 Extended Tridiagonal-1 函数,这是因为 Extended White & Holst 函数的结构更类似于 Rosenbrock 函数。实验结果表明,PSO-R 算法虽然鲁棒性弱但对于结构类似于 Rosenbrock 函数的函数具有较好的优化性能。

4.2.2 参数 h 的实验分析

在测试函数性能的大量实验中发现,参数 h 的不同取值对算法的性能有很大的影响。因此,为了测试参数的不同取值对 PSO-R 算法性能的影响,在 $(-\text{II}, \text{II})$ 范围内取 9 个点,分别为-1.0, 1.0, 1.5, 1.8, 2.0, 2.5, 3.0, 3.14, 3.141592653。每个 h 值对应的 PSO-R 算法对 Rosenbrock 函数进行优化,运行 30 次,实验结果如表 3 所列。

表 3 不同 h 值下 PSO-R 算法的实验结果

参数 h 取值	平均值	标准方差	最好适应值
-1.0	1.81e+003	3.31e+002	1.18e+003
1.0	1.87e+003	2.39e+002	1.41e+003
1.5	1.08e+003	1.66e+002	7.22e+002
1.8	9.34e+002	1.68e+002	7.08e+002
2.0	1.00e+003	2.78e+002	4.96e+002
2.5	1.80e+002	3.13e+001	8.59e+001
3.0	2.90e-001	8.71e-003	2.64e-001
3.14	4.58e-009	9.13e-011	4.33e-009
3.1415926	0.0	0.0	0.0

从表 3 中可以看到,随着 h 取值的不断增大,平均值、标准方差和最好适应值都在不断减小,平均值从 1810 减至 0,标准方差从 3310 减至 0,最好适应值也从 1180 减至 0。实验结果表明,参数 h 的取值对 PSO-R 算法的性能有很大的影响。为了使 PSO-R 算法具有更好的优化性能,分别取 h 的范围为 $[3.0, \text{II})$ 、 $[3.14, \text{II})$ 。

在 h 的取值范围为 $[3.0, \text{II})$ 、 $[3.14, \text{II})$ 的情况下,分别再次采用 PSO-R 算法对 Rosenbrock 函数进行优化,实验结果如表 4 所列。

表 4 h 在不同取值范围下 PSO-R 算法的实验结果

h 取值范围	平均值	标准方差	最好适应值
$(-\text{II}, \text{II})$	2.55e-003	4.78e-003	2.07e-011
$[3.0, \text{II})$	6.90e-010	1.20e-009	3.28e-016
$[3.14, \text{II})$	8.53e-022	1.05e-021	1.84e-027

从表 4 中看到,当 h 的取值范围从 $(-\text{II}, \text{II})$ 压缩到 $[3.0, \text{II})$ 时,平均值数量级从 10^{-3} 到 10^{-10} 、标准方差从 10^{-3} 到 10^{-9} 、最好适应值从 10^{-11} 到 10^{-16} 等 3 个性能评价指标均大幅提高。进一步压缩范围到 $[3.14, \text{II})$,平均值数量级从 10^{-10} 到 10^{-22} 、标准方差从 10^{-9} 到 10^{-21} 、最好适应值从 10^{-16} 到 10^{-27} 等 3 个性能评价指标均再次大幅提高,从而表明参数 h 的范围较精确的定位能够较大幅度提高 PSO-R 算法的性能。

结合表 3 和表 4 还可以看到,当 $h=3.1415926$,却在范围 $[3.0, \text{II})$ 内,平均值、标准方差、最好适应值都达到全局极小值 0,取得了较好的优化性能,PSO-R 算法已经找到了全局极小值。为了不失一般性,PSO-R 算法中 h 取值范围为 $[3.14, \text{II})$ 。

另外,从实验中还可以看到, h 的取值范围的缩小引导粒子向全局极小值靠近,同时证明了所设计算法的正确性。

结束语 鉴于 Rosenbrock 函数全局极小值位于一条狭长的山谷底部致使许多优化算法很难找到其全局极小值,专门提出了一种改进的 PSO 算法(PSO-R)。实验研究表明,该改进算法对 Rosenbrock 函数的优化具有非常好的性能,且为解决类似于该函数的优化问题提供了一种新的思路。该算法对 Rosenbrock 函数具有很好的优化性能,同时,Rosenbrock 函数也证明该算法是一种性能很好的优化算法。然而,该算法也存在不足,即振荡性强度过大可能会导致收敛速度变慢;该算法对于 Rosenbrock 函数具有很好的优化性能,但鲁棒性弱,对于其它函数的优化性能相对差些,原因是该算法是针对 Rosenbrock 函数的特性设计的;下一步的研究工作是如何控制算法的振荡强度以及按照算法思想将其推广到所有优化问题。

参 考 文 献

- [1] 梁艳春,周春光,李寿范.基于遗传算法的 Rosenbrock 函数优化问题的研究[J].软件学报,1997;701-708
- [2] Rosenbrock H H. An automatic method for finding the greatest or least value of a function [J]. Computer Journal, 1960,3(3): 175-184
- [3] Kennedy J, Eberhart R C. Particle swarm optimization [C]// Proc of the IEEE Int Conf on Neural Networks. Piscataway, 1995;1942-1948
- [4] Wang Hui, Wu Zhi-jian, Rahnamayan S, et al. Enhancing particle swarm optimization using generalized opposition-based learning [J]. Information Sciences, 2011,181(20):4699-4714
- [5] Yang X-S, Deb S. Engineering optimization by cuckoo search [J]. Int. J. Math. Modelling Num, Optimization, 2010,1(4):330-343
- [6] Dixon L C W, Mills D J. Effect of rounding errors on the Variable Metric Method [J]. Journal of Optimization Theory and Applications, 1994,80(1):175-179
- [7] 崔志化,曾建潮.微粒群优化算法[M].北京:科学出版社,2004
- [8] 陈入云.高振荡函数积分的高效数值算法及实现研究[D].长沙:中南大学,2009
- [9] Shi Y, Eberhart R C. A modified particle swarm optimizer [C]// Proc. Congr. Evol. Comput. 1998;69-73
- [10] Parsopoulos K E, Vrahatis M N. UPSO-A unified particle swarm optimization scheme [Z]. Lect. Ser. Comput. Sci., 2004; 868-873
- [11] Liang J J, Qin A K, Suganthan P N, et al. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions[J]. IEEE Trans. Evolut. Comput, 2006,10:281-295
- [12] Wang Hui, Liu Yong, Zeng San-you, et al. Opposition-based particle swarm algorithm with Cauchy mutation[C]//Proc. Congr. Evol. Comput. 2007;4750-4756