

基于二次优化的随机优化算法结果的改进

王东风, 黄金山

(华北电力大学 自动化系, 河北 保定 071003)

摘要: 统计研究发现, 随机优化算法多次运行后的优化结果满足正态分布, 且期望值更接近最优解. 为此, 提出一种基于统计学理论并结合牛顿法的二次优化方法来改进随机优化算法的求解结果, 以克服将多次优化结果的平均值作为最优解时不能满足精度要求的缺陷. 以遗传算法对4个经典测试函数的多次优化为例, 分别运用平均法和二次优化法来综合其优化结果. 多次实验表明, 二次优化法在处理多次随机运行结果时, 比平均法精度更高、稳定性更好.

关键词: 正态分布; 遗传算法; 随机优化算法; 牛顿法

中图分类号: TP18

文献标志码: A

Improving results of stochastic optimization algorithms via secondary optimization

WANG Dong-feng, HUANG Jin-shan

(Department of Automation, North China Electric Power University, Baoding 071003, China. Correspondent: WANG Dong-feng, E-mail: wangdongfeng@ncepubd.edu.cn)

Abstract: Through analyzing the stochastic optimization results, it is been found that optimization results satisfy normal distribution, and the expected value is the optimal solution. A secondary optimization method based on the statistical theory and the Newton method is proposed to improve the optimization results of stochastic optimization algorithms, which can overcome the average method's shortcoming that the precision requirements are often can not be met. Taking multiple optimization results of four classic test functions optimized by genetic algorithm as examples, the average method and the secondary optimization method are respectively used to synthesize the optimization results. Experiments show that, in dealing with multiple stochastic optimization results, the secondary optimization method has higher accuracy and better stability than those of the average method.

Keywords: normal distribution; genetic algorithm; stochastic optimization algorithm; Newton method

0 引言

在日常生产生活中经常会遇到许多规划问题, 人们将这些问题归结为函数的优化问题, 并提出了多种优化算法解决这类问题, 例如梯度法、牛顿法、单纯形法等. 随着人类认知水平的发展, 探索的未知世界越来越复杂, 也就遇到了越来越多的复杂优化问题, 这些优化问题往往呈现多维、非线性、多峰值、多目标等特征^[1]. 这时传统优化算法已不能满足精度要求. 传统优化方法是基于数学知识的, 往往有精确的数学原理、数学公式, 以及精确的寻优方向、寻优步距, 对一些简单的单峰值优化问题具有非常好的优化效果. 但大量的复杂优化问题会使数学计算量很大, 甚至无

法进行解析式的优化求解. 这时科学家们需要突破数学发展滞后的局限性. 他们发现可以借鉴自然界的许多现象和规律来解决优化问题, 从而提出了许多启发式的智能优化算法^[2], 如遗传算法、模拟退火算法、蚁群算法、粒子群算法等. 这些启发式算法虽然是对不同自然现象和规律的描述, 但有一个共同点, 即都基于随机搜索. 这些优化算法都是以一组可行解为初始搜索点, 在可行域里进行随机化、结构化的全局并行搜索, 对好的优化解进行保留并以此寻找更好的优化解, 因此也可以将它们统称为随机优化算法. 随机化的搜索机制导致算法的优化结果也具有随机性特点, 这些结果只是最优解附近的随机解而不是真正的最

收稿日期: 2013-09-22; 修回日期: 2014-01-16.

基金项目: 高等学校博士学科点专项科研基金项目(20120036120013); 中央高校基本科研业务费专项资金项目(11MG49).

作者简介: 王东风(1971—), 男, 教授, 博士, 从事先进控制、群智能计算等研究; 黄金山(1989—), 男, 硕士生, 从事群智能优化算法的研究.

优解. 随机优化算法对目标函数要求不高, 对大型多维多峰值复杂问题的优化尤为适用, 因此在科学和工程领域获得了广泛应用. 文献 [2-7] 列举了这些随机优化算法在众多工程领域的研究应用, 结果表明它们都能很好地解决实际难题. 但对某些大型工程优化问题, 如能进一步提高优化精度, 将会大大地节约成本或提高效益, 因此, 进一步提高优化精度意义重大.

随机优化算法的搜索机制是基于概率的随机化搜索, 因此单次搜索结果也必然是一个最优解附近的随机化结果, 而很难得到理论上的最优解. 为了提高解的精确性, 许多学者将多次优化运行结果的平均值作为问题的最优解^[8-10]. 作者在对多次优化运行的结果进行统计研究后发现, 随机优化结果的各个解变量相互独立, 且服从正态分布, 即每个解变量满足 $X \sim N(\mu, \sigma^2)$, 其中 μ 为正态分布的期望值, 对应优化问题的最优解. 依据最大似然估计 (MLE) 原理, 正态分布的期望值 μ 可用样本均值 $\bar{X}$ 来近似, 所以之前学者将多次优化结果的平均值作为优化问题的最优解有一定的可行性. 但对于有限的样本数据, 即多次独立优化运行的次数不是无穷大时, 这种近似方法的精度就显得不够, 为此本文提出一种对优化问题进行二次优化来求得最优解的方法. 以优化算法中的典型代表——遗传算法为例, 通过多个测试函数的优化实验验证了所提出的方法能够不同程度地提高最优解的精度, 也从另一个角度提高了随机优化算法的优化精度, 具有很高的实际应用价值.

1 遗传算法简介

遗传算法是美国 Holland 教授于 1975 年提出的一种智能优化方法. 遗传算法借鉴了自然界生物遗传进化过程中“适者生存, 优胜劣汰”的进化机制, 通过模拟生物进化过程来寻找问题的最优解. 遗传算法的寻优过程如下^[11-12]:

1) 前期准备. 确定算法参数, 包括群体规模数 M , 迭代次数 N , 交叉概率 P_c , 变异概率 P_m , 适应度相差限 ϵ ; 选取适应度函数, 适应度函数的设计一般要根据具体的优化问题而定, 往往直接影响遗传算法的优化

性能; 变量编码, 常用的编码策略有二进制编码、浮点数编码、字符编码等.

2) 种群初始化.

3) 终止迭代判断. 若算法迭代次数已满足最大迭代次数 N , 或者群体中个体的最大适应度和最小适应度之差不超过适应度相差限 ϵ , 则停止迭代, 输出最大适应度个体的解变量值, 否则继续进行遗传操作.

4) 遗传操作. 包括选择、交叉和变异操作.

2 基于正态分布分析的重复运行遗传算法优化经典函数的结果综合

2.1 用于本文测试的经典函数介绍

为了验证优化算法的优化性能, 本文采用遗传算法对 4 个经典测试函数分别进行多次优化运行^[13-14], 获得多次优化结果用于后续分析. 这 4 个经典测试函数的解析式、理论最优函数值 f^* 、理论最优解 x^* 和解的可行域如表 1 所示 (为说明问题而又不失一般性, 每个测试函数的变量维数取为 2).

2.2 重复运行随机优化算法优化结果的正态分布特性分析

正态分布概念是由德国的数学家和天文学家 Moivre 于 1733 年首次提出并由德国数学家 Gauss 率先将其应用于天文学研究, 故又称高斯分布. 它是概率论中最重要的一种分布.

正态分布记作 $N(\mu, \sigma^2)$, 其中 μ 和 σ 分别表示正态分布的期望值和标准差^[15], 是正态分布的两个重要参数, μ 决定了分布的集中位置, σ 决定了分布均匀变动的缓急程度. 正态分布的概率密度函数为

$$f(x|\mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \tag{1}$$

通过它可以计算出随机变量 x 在某点出现的概率.

正态分布的特征主要有:

1) 集中性. 随机数据都集中在某一固定值附近, 且越靠近固定值数据越密集.

2) 对称性. 随机数据的分布都以那个固定值为对称点, 固定值左右相同距离处数据出现的次数几乎等同.

表 1 本文使用的经典测试函数

函数	函数解析式	最优解 x^* 和最优值 f^*	解的可行域
F_1 : Schaffer1 Function	$F_1 = \sqrt[4]{x_1^2 + x_2^2} \cdot (\sin(50 \cdot \sqrt[10]{x_1^2 + x_2^2}))^2 + 1$	$x^* = (0, 0), f^* = 0$	$x \in [-10, 10]$
F_2 : Schaffer2 Function	$F_2 = 0.5 + \frac{(\sin \sqrt{x_1^2 + x_2^2})^2 - 0.5}{[1 + 0.1 \cdot (x_1^2 + x_2^2)]^2}$	$x^* = (0, 0), f^* = 0$	$x \in [-10, 10]$
F_3 : Sphere Function	$F_3 = x_1^2 + x_2^2$	$x^* = (0, 0), f^* = 0$	$x \in [-100, 100]$
F_4 : Generalized Rastrigin's Function	$F_4 = x_1^2 - 10 \cos(2\pi x_1) + x_2^2 - 10 \cos(2\pi x_2) + 20$	$x^* = (0, 0), f^* = 0$	$x \in [-5.12, 5.12]$

3) 均匀变动性. 随机数据在离固定值越远的地方出现的次数成逐渐递减趋势.

正态分布是统计学里面最重要的一种概率分布, 日常生活中很多现象都服从正态分布规律. 例如: 随机测量误差, 人群的身高、智商, 物件的使用寿命等. 本文采用遗传算法 (GA) 对上述 4 个测试函数分别进行优化实验, 其中对测试函数 Schaffer1 Function 独立运行 1000 次, 得变量 x_1 的优化结果如图 1 所示.

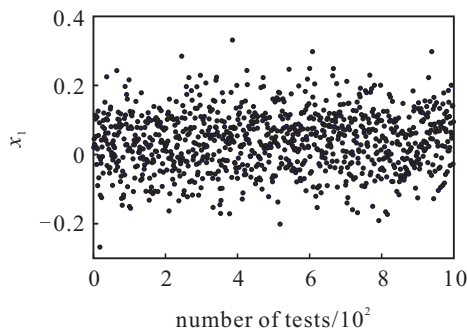


图 1 GA 独立运行 1000 次优化函数 F_1 的结果分布

对每个函数的每个变量的 1000 次优化结果数据进行统计研究, 发现这些结果数据的分布都满足正态分布的上述 3 大特征, 可按正态分布规律处理, 即 $X \sim N(\mu, \sigma^2)$.

由正态分布的概率密度函数也可以发现, 随机变量 x 在 μ 点附近出现的概率最大, 所以 μ 值可以看作优化函数的最优解, 因此提出一种改善 μ 值估计精度的方法具有实际意义. 通常采用最大似然估计法来计算正态分布的两个重要参数 μ 和 σ . 最大似然估计将结果平均值作为期望值 μ 的估计, 在有限样本的情况下, 这种估计往往不能令人满意, 所以本文提出一种新的 μ 值计算方法. 通过实验分析表明, 新的计算方法具有更高的精度和更好的稳定性. 本文使用样本概率密度计算方法, 以最大化的 μ 值作为新的期望值, 即最大化函数

$$S(\mu) = \max_{\mu} \sum_{i=1}^n \frac{1}{\sqrt{2\pi}\sigma_0} e^{-\frac{(x_i - \mu)^2}{2\sigma_0^2}}. \quad (2)$$

其中: N 为样本数即随机优化次数; σ_0 为最大似然估计法对标准差的估计结果, 为一固定值. 所以式 (2) 可简化为

$$S(\mu) = \max_{\mu} \sum_{i=1}^n e^{-\frac{(x_i - \mu)^2}{2\sigma_0^2}}. \quad (3)$$

2.3 基于二次优化的结果综合

牛顿法是求解非线性优化问题的有效方法之一, 由于其简单有效, 在工程上有很多应用实例. 牛顿法的基本思想是: 用二次函数逼近目标函数 $f(X)$, 用二次函数的极小值点逼近目标函数 $f(X)$ 的极小值点. 将 $f(X)$ 在第 n 次优化点 X_n 展成二阶泰勒级数, 有

$$f(X) \approx p(X) = f(X_n) + \nabla f(X_n)^T (X - X_n) + \frac{1}{2} (X - X_n)^T H(X_n) (X - X_n), \quad (4)$$

其中: $\nabla f(X_n)^T, H(X_n)$ 分别是 $f(X)$ 在 X_n 点处的梯度和 Hessian 矩阵. 求 $f(X)$ 的极小值点, 即使 $\nabla p(X) = 0$ 的点, 令 $\nabla p(X) = 0$, 则有

$$\nabla f(X_n) + H(X_n)(X - X_n) = 0. \quad (5)$$

若 $H(X_n)$ 正定, 按式 (5) 求解出 X , 并将它记作 X_{n+1} , 即得到牛顿法的迭代公式

$$X_{n+1} = X_n - H(X_n)^{-1} \nabla f(X_n). \quad (6)$$

牛顿法对二次严格凸函数非常有效, 收敛速度很快. 一般目标函数在最优值附近呈现为二次函数, 所以在最优值附近采用牛顿迭代法可以很快找到最优值. 牛顿法优化步骤如下:

1) 选定初始点 X_0 , 优化精度 $\epsilon > 0$, 令 $n = 0$;

2) 计算 $P_n = -H(X_n)^{-1} \nabla f(X_n)$, 若 $\|P_n\| < \epsilon$, 则停止迭代, 取 $X_{\min} = X_n$, 否则转 3);

3) 令 $X_{n+1} = X_n + P_n$, 求 X_{n+1} , 转 2).

综上所述, 采用本文所提的二次优化方法来求解优化问题的过程描述如下:

1) 将优化问题归结为一优化函数模型;

2) 采用某一随机优化算法 (本文采用遗传算法) 对优化函数进行多次优化, 记录优化结果;

3) 采用最大似然估计法计算出结果样本的正态分布参数 μ_0 和 σ_0 ;

4) 固定 σ_0 , 以 μ_0 为初始解, 采用牛顿法对指标函数 $S(\mu) = \max_{\mu} \sum_{i=1}^n e^{-\frac{(x_i - \mu)^2}{2\sigma_0^2}}$ 进行迭代优化, 找到 μ 的最优解 μ^* , 将 μ^* 作为待求优化问题的最优解.

采用最大似然法和本文的二次优化方法计算得到测试函数 Schaffer1 Function 的变量 x_1 的概率密度函数, 其直方图 (hist) 和概率密度函数图 (pdf) 合二为一, 如图 2 所示. 由图 2 可知, 随机优化的结果分布符合正态分布, 且采用本文方法计算得到的正态分布期望值更接近理论最优解.

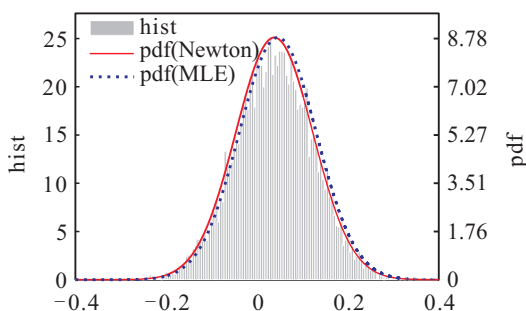


图 2 GA 独立运行 1000 次优化函数 F_1 的结果

采用遗传算法对所有测试函数进行优化实验,将变量 x_1 的优化结果分别采用取平均和本文方法来求得最优解并进行比较. 将上述过程重复执行 100 次,这 100 组最优解的比较结果如图 3 所示. 分析图 3 可知,采用本文方法求多次随机优化的最优解精确度更高,且稳定性更好.

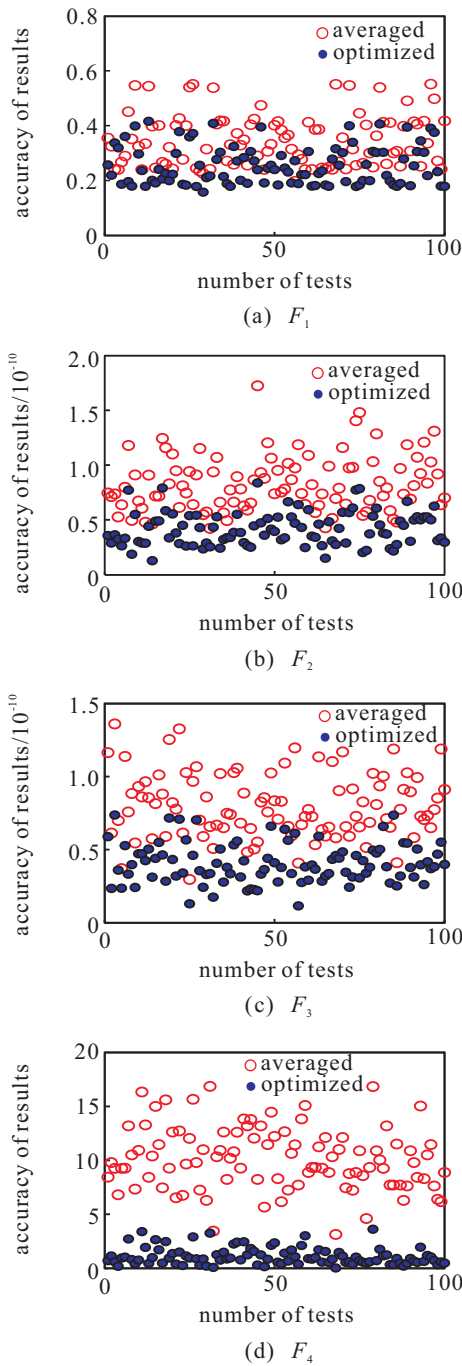


图3 平均值法与优化法所求结果的精度对比

表1给出了所有函数的理论极值及其对应的变量值,表2给出了对所有函数重复优化 100 次后得到的变量平均值及其对应的函数值、优化 100 次后采用本文方法得到的最优变量值及其对应的函数值. 由采用两种方法计算得到的最优函数值可知,采用本文方法能够找到更好的优化解,提高优化问题的求解精度.

表2 100次运行优化值在不同方法综合时的平均结果比较

函数	平均方法 $\frac{x_1, x_2}{f}$	本文方法 $\frac{x_1, x_2}{f}$	函数值改进/%
F_1	$\frac{3.90e^{-02}, 3.80e^{-02}}{2.58e^{-01}}$	$\frac{2.29e^{-02}, 2.21e^{-02}}{1.78e^{-01}}$	24.03
F_2	$\frac{6.10e^{-06}, 5.93e^{-06}}{7.97e^{-11}}$	$\frac{4.19e^{-06}, 4.26e^{-06}}{3.93e^{-11}}$	49.97
F_3	$\frac{6.27e^{-06}, 6.15e^{-06}}{7.72e^{-11}}$	$\frac{4.35e^{-06}, 4.41e^{-06}}{3.84e^{-11}}$	49.49
F_4	$\frac{1.65e^{-01}, 1.64e^{-01}}{9.86e^{+00}}$	$\frac{4.72e^{-02}, 4.76e^{-02}}{8.87e^{-01}}$	90.17

本文通过 4 个测试函数的优化实验验证了本文二次优化方法不仅可行,而且比多次随机优化结果平均值法具有更高的优化精度,优化结果更加稳定. 另外,还对压力管设计以及伸缩绳设计等典型优化问题^[16-17]进行了求解,结果均表明二次优化算法的运行结果都得到了不同程度的改善,限于篇幅,结果略. 因此,本文方法可以应用到最优值与最优解未知的实际工程优化问题,具有很高的实际应用价值.

3 结 论

通过对多组随机优化结果样本的研究,发现随机优化算法结果服从正态分布的统计学规律. 针对采用平均法处理多次随机优化算法的结果最优解精度不够的问题,提出了基于牛顿法原理的二次优化方法. 针对多个函数的多次测试实验表明,此方法能够有效地提高随机优化算法的精度,改善算法的性能,且算法简单容易实现,因而具有一定的理论意义、较高的应用价值.

参考文献(References)

[1] 刘淳安. 动态多目标优化进化算法及其应用[M]. 北京: 科学出版社, 2011: 1-5.
(Liu C A. Dynamic multi-object optimization algorithms and their applications[M]. Beijing: Science Press, 2011: 1-5.)

[2] 雷秀娟. 群智能优化算法及其应用[M]. 北京: 科学出版社, 2012: 1-20.
(Lei X J. Swarm intelligent optimization algorithms and their applications[M]. Beijing: Science Press, 2012: 1-20.)

[3] 王凌, 钱斌. 混合差分进化与调度算法[M]. 北京: 清华大学出版社, 2012: 51-252.
(Wang L, Qian B. Hybrid difference evolutionary optimization and scheduling algorithms[M]. Beijing: Tsinghua University Press, 2012: 51-252.)

[4] Juang C F, Chang Y C, Hsiao C M. Evolving gaits of a hexapod robot by recurrent neural networks with symbiotic species-based particle swarm optimization[J]. IEEE Trans on Industrial Electronics, 2011, 58(7): 3110-3119.

- [5] Zhou Y R. Runtime analysis of an ant colony optimization algorithm for TSP instances[J]. IEEE Trans on Evolutionary Computation, 2009, 13(5): 1083-1092.
- [6] 邓林义, 林焰. 粒子群算法求解任务可拆分项目调度问题[J]. 控制与决策, 2008, 23(6): 681-684.
(Deng L Y, Lin Y. Particle swarm optimization for resource-constrained project scheduling problems with activity splitting[J]. Control and Decision, 2008, 23(6): 681-684.)
- [7] 李倩, 宫俊, 唐加福. 多目标粒子群算法在交叉培训规划中的应用[J]. 控制理论与应用, 2013, 30(1): 17-22.
(Li Q, Gong J, Tang J F. Multi-objective particle swarm optimization algorithm for cross-training programming[J]. Control Theory & Applications, 2013, 30(1): 17-22.)
- [8] 王凌, 刘波. 微粒群优化与调度算法[M]. 北京: 清华大学出版社, 2008: 49-51.
(Wang L, Liu B. Particle swarm optimization and scheduling algorithms[M]. Beijing: Tsinghua University Press, 2008: 49-51.)
- [9] 肖人彬. 面向复杂系统的群集智能[M]. 北京: 科学出版社, 2013: 79-361.
(Xiao R B. Complex system oriented swarm intelligence[M]. Beijing: Science Press, 2013: 79-361.)
- [10] 李碧, 郝志峰. 协同进化算法及其应用[M]. 北京: 科学出版社, 2013: 85-97.
(Li B, Hao Z F. Co-evolutionary optimization algorithms and their applications[M]. Beijing: Science Press, 2013: 85-97.)
- [11] 戴晓晖, 李敏强, 寇纪淞. 遗传算法理论研究综述[J]. 控制与决策, 2000, 15(3): 263-268.
(Dai X H, Li M Q, Kou J S. Survey on the theory of genetic algorithms[J]. Control and Decision, 2000, 15(3): 263-268.)
- [12] 席裕庚, 柴天佑, 恽为民. 遗传算法综述[J]. 控制理论与应用, 1996, 13(6): 697-708.
(Xi Y G, Chai T Y, Yun W M. Survey on genetic algorithm[J]. Control Theory & Applications, 1996, 13(6): 697-708.)
- [13] Sahu A, Panigrahi S K, Pattnaik S. Fast convergence particle swarm optimization for functions optimization[J]. Procedia Technology, 2012, 4(1): 319-324.
- [14] 阎岭, 蒋静坪. 进化学习策略收敛性和逃逸能力的研究[J]. 自动化学报, 2005, 31(6): 61-68.
(Yan L, Jiang J P. Convergence and escape capacity research of evolution learning strategies[J]. Acta Automatica Sinica, 2005, 31(6): 61-68.)
- [15] 赵瑛, 孙王杰. 概率论与数理统计[M]. 北京: 高等教育出版社, 2013: 89.
(Zhao Y, Sun W J. Probability and statistics[M]. Beijing: Higher Education Press, 2013: 89.)
- [16] He Q, Wang L. An effective coevolutionary particle swarm optimization for constrained engineering design problems[J]. Engineering Applications of Artificial Intelligence, 2007, 20(1): 89-99.
- [17] Coello C A, Mantes E M. Constraint-handling in genetic algorithms through the use of dominance-based tournament selection[J]. Advanced Engineering Informatics, 2002, 16(1): 193-203.

(责任编辑: 孙艺红)