解连续性优化问题的摸石头过河算法

高 尚[1],邱 玲[2],曹存根[3]

(1.江苏科技大学计算机科学与工程学院,江苏 镇江 212003)
(2.人工智能四川省高校重点实验室,四川 自贡 643000)
(3.中国科学院计算所智能信息处理重点实验室,北京 100190)

[**摘要**] 依据“摸石头过河”的思想,提出一种快速、高效的随机优化算法. 摸石头过河算法是以一个解为起点,向该起点附近邻域随机搜索若干个解,找出这些解中的最好的一个解,以此解为下次迭代的结果,然后以此点为起点,再向附近邻域随机搜索若干个解,以此类推. 解连续性优化问题时改进的方法是逐渐缩小搜索空间,对几个经典测试函数进行实验的结果表明,利用摸石头过河及其改进算法能够极大地提高收敛速度和精度.

[**关键词**] 随机优化算法,连续空间优化,快速随机优化算法,摸石头过河

[**中图分类号**]TP18 [**文献标志码**]A [**文章编号**]1001-4616(2015)01-0108-05

Solving Continuous Optimization Problem by Wading Across Stream Algorithm

Gao Shang[1],Qiu Ling[2],Cao Cungen[3]

(1.School of Computer Science and Engineering,Jiangsu University of Science and Technology,Zhenjiang 212003,China)
(2.Artificial Intelligence of Key Laboratory of Sichuan Province,Zigong 643000,China)
(3.Key Lab of Intelligent Information Processing of Chinese Academy of Sciences,
Institute of Computing Technology,Beijing 100190,China)

Abstract:According to idea of Wading across the Stream by feeling the way,a kind of fast efficient random optimization algorithm is put forward. The Wading across Stream Algorithm(WSA)acts a solution as a start point,then searches several solutions randomly near the start point,and finds the best solution of these solutions. This best solution is to take as next start point,and then several random solutions near this start point are searched,and so on. For solving continuous optimization problem,the improved Wading across Stream Algorithm shrinks the search space gradually. The experiment results of some classic benchmark functions show that the proposed optimization algorithms improve extraordinarily the convergence velocity and precision.

Key words:random optimization algorithm,continuous space optimization,fast random optimization algorithm,Wading across Stream Algorithm

对于非线性规划,目前还没有适合各种问题的一种解法,各个方法都有自己特定的适用范围. 对于传统的解析法,要求目标函数与约束函数具有连续性并且其导数存在. 但在某些实际问题中,由于目标函数很复杂,有时甚至无法写出其表达式,当然更无法求其导数,这样解析法就不再适用了. 目前采用智能优化算法来求解比较流行,基于随机搜索技术的进化算法[1]、模拟退火算法[2-3]、蚁群优化算法[4]、粒子群优化算法[5-9]和混沌优化算法等优化算法[10-11]在解决全局最优问题中已得到广泛应用. 但这些智能算法都存在某些缺陷,如需要增加一些额外的运算,进化算法中需要变异和选择操作等,这导致计算复杂性高、问题实时性差、搜索精度差,易陷于局部最优解. 本文依据“摸石头过河”的思想,提出一种快速、高效的随机优化算法-摸石头过河算法(Wading across Stream Algorithm,WSA),解决连续优化问题. 与其他优化法相比,该算法直接利用随机搜索过程进行优化,没有额外计算,降低了计算复杂度.

收稿日期:2014-08-20.
基金项目:人工智能四川省重点实验室开放基金(2012RYJ04)、中科院智能信息处理重点实验室开放课题(IIP2013-1).
通讯联系人:高尚,博士,教授,研究方向:智能计算研究. E-mail:gaoshang@ sohu. com

1　摸石头过河算法的思想

摸石头过河算法的思想来源于“摸石头过河”的思想,摸到一个“石头”后,向该“石头”周围摸索其他石头,继续摸到一个“石头”后,再向该“石头”周围摸索其他石头,以此类推进行搜索. 摸石头过河算法是以一个解为起点,向该起点附近邻域随机搜索若干个解,找出这些解中的最好的一个解,以此解为第 2 次迭代的结果. 然后以此点为起点,再向附近邻域随机搜索若干个解,找出这些解中的最好的一个解,以此解为第 3 次迭代的结果. 后面的步骤以此类推,达到最大迭代次数或其他停止条件为止. 其迭代过程如图 1 所示.

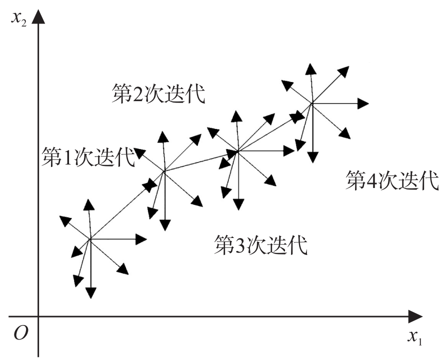

图 1　摸石头过河算法的迭代过程

Fig. 1　Iterative process of Wading across Stream Algorithm

“摸石头过河算法”与模拟退火算法有点类似,但效果比模拟退火算法好,并且算法比模拟退火算法简单. 模拟退火算法是从一个解搜索下一个解,而“摸石头过河算法”是从周围若干个解中摸索比较好的解作为下一个解,利用了“群”优势. 模拟退火算法根据 Metropolis 准则来接受或舍弃下一个解,而“摸石头过河算法”直接选取周围若干个解中比较好的解为下一个解,操作简单.

2　解连续性优化问题的摸石头过河算法

2.1　基本摸石头过河算法

连续空间优化问题可表示为:

$$\begin{aligned}&\min \quad f(x_1,x_2,\cdots,x_n),\\ &\text{s. t.} \quad a_i \leqslant x_i \leqslant b_i (i=1,2,\cdots,n).\end{aligned} \tag{1}$$

解连续空间优化问题的摸石头过河算法如下:

步骤 1　设置算法参数:搜索解的个数 m,邻域半径 L,迭代次数 $n_{\max}$.

步骤 2　随机产生一个解($x_1^*,x_2^*,\cdots,x_n^*$),计算其目标值 f^*,$k=0$.

步骤 3　在解($x_1^*,x_2^*,\cdots,x_n^*$)邻域半径 L 内产生 m 个邻域解:

$$X'=\begin{bmatrix} x'_{11} & x'_{12} & \cdots & x'_{1n}\\ x'_{21} & x'_{22} & \cdots & x'_{2n}\\ \vdots & \vdots & \ddots & \vdots\\ x'_{m1} & x'_{m2} & \cdots & x'_{mn}\end{bmatrix}. \tag{2}$$

其中 $x'_{ij}=x_j^*+L\cdot r_{ij}$,$r_{ij}$为区间[$-1,1$]的均匀分布的随机数.

步骤 4　计算这 m 个解的目标函数值$f'_1,f'_2,\cdots,f'_m$,再找出最好解($x_1^*,x_2^*,\cdots,x_n^*$)及其目标值f^*,保留最好解,$k=k+1$.

步骤 5　若 $k>n_{\max}$,算法则结束,输出保留的最好解;否则执行步骤 3.

该摸石头过河算法的时间复杂性估算如下:以计算目标函数值的操作花时间最多,所以时间复杂性大约为 $O(m\cdot n_{\max})$.

2.2　改进的摸石头过河算法

“摸石头过河”时,总要在岸边考察一下,慎重选择初始起点. 同样道理,初始选择的解对整个算法会产生影响. 改进的思路是先产生 N 随机个解,从中找出最好解作为起点解.

随着迭代次数的增加,结果越来越接近最优解,假如邻域半径 L 保持不变,到最优解附近时收敛的速度会变得很慢,改进的思路是随着迭代次数的增加,邻域半径 L 逐渐变小. 控制邻域半径 L 的方法比较多,比较简单的方法如下:

$L_k=\alpha L_{k-1}$,α 为收缩系数,一般去 0.99 左右,α 值越小,搜索速度越快.

改进的摸石头过河算法(Improved Wading across Stream Algorithm,IWSA)如下:

步骤 1　设置算法参数:搜索解的个数 m,邻域半径 L_0,迭代次数 n_{max}.

步骤 2　变量 $x_i(i=1,2,\cdots,n)$ 在 $[a_i,b_i]$ 区间均匀随机取值,共产生 N 个解:

$$X=\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N1} & x_{N2} & \cdots & x_{Nn} \end{bmatrix}. \tag{3}$$

步骤 3　计算这 N 个解的目标函数值 $f_1,f_2,\cdots,f_N$,找出最好解 $(x_1^*,x_2^*,\cdots,x_n^*)$ 及其目标值 f^*,保留最好解,$k=0$.

步骤 4　在最好解 $(x_1^*,x_2^*,\cdots,x_n^*)$ 邻域半径 L_k 内产生 m 个邻域解:

$$X'=\begin{bmatrix} x'_{11} & x'_{12} & \cdots & x'_{1n} \\ x'_{21} & x'_{22} & \cdots & x'_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x'_{m1} & x'_{m2} & \cdots & x'_{mn} \end{bmatrix}. \tag{4}$$

其中 $x'_{ij}=x_j^*+L_k\cdot r_{ij}$,$r_{ij}$ 为区间 $[-1,1]$ 的均匀分布的随机数.

步骤 5　计算这 m 个解的目标函数值 $f'_1,f'_2,\cdots,f'_m$,再找出最好 $(x_1^*,x_2^*,\cdots,x_n^*)$ 及其目标值 f^*,保留最好解,$k=k+1$,$L_k=\alpha L_{k-1}$.

步骤 6　若 $k>n_{max}$,算法则结束,输出保留的最好解;否则执行步骤 4.

2.3　数值仿真与分析

2.3.1　与随机优化算法比较

摸石头过河算法和改进的摸石头过河算法实质上计算了 $m\cdot n_{max}$ 个目标函数值,从中找出最好解. 随机优化算法(Random Optimization Algorithm,ROA)的思路从解空间中直接随机产生 $m\cdot n_{max}$ 个解,计算这 $m\cdot n_{max}$ 个目标函数值,从中找出最好解.

为了说明摸石头过河算法和改进的摸石头过河算法的优势,与随机优化算法进行比较,以下面测试函数为例:

对于测试函数

$$\begin{aligned} &\min \quad F_0=\sum_{i=1}^{4}[x_i^2-10\cos(2\pi x_i)+10], \\ &\text{s. t.} \quad -1\leqslant x_i\leqslant 1(i=1,2,3,4). \end{aligned} \tag{5}$$

当 $x_1=x_2=x_3=x_4=0$ 时,$F_{min}=0$.

对于摸石头过河算法和改进的摸石头过河算法参数设置如下:$m=200$,$n_{max}=100$,$N=1\ 000$,$L=L_0=0.1$,$\alpha=0.99$. 对于随机优化算法产生 $m\cdot n_{max}=20\ 000$ 个解,从中找出最好的解. 算法各测试 100 次,统计数据如表 1 所示.

表 1　三种算法的比较

Table 1　Comparison of three algorithms

算法	最好解	最坏解	平均值
随机优化算法(ROA)	0.071 85	2.878 20	1.537 64
摸石头过河算法(WSAA)	0.002 63	0.050 02	0.022 111
改进的摸石头过河算法(IWSA)	0.000 04	0.014 62	0.006 04

从表 1 可以看出,改进的摸石头过河算法的效果比摸石头过河算法和随机优化算法效果好很多.

2.3.2　收敛性比较

以下 4 个为经常被国内外学者用来测试优化算法有效性的测试函数[10].

$$
\begin{aligned}
&\min F_1=\sum_{i=1}^{30}x_i^2 && -1\leqslant x_i\leqslant 1(i=1,2,\cdots 30),\\
&\min F_2=\sum_{i=1}^{30}|x_i|+\prod_{i=1}^{30}|x_i| && -1\leqslant x_i\leqslant 1(i=1,2,\cdots 30),\\
&\min F_3=\max_{1\leqslant i\leqslant 30}|x_i| && -1\leqslant x_i\leqslant 1(i=1,2,\cdots 30),\\
&\min F_4=\sum_{i=1}^{30}\Big(\sum_{j=1}^{i}x_j\Big)^2 && -1\leqslant x_i\leqslant 1(i=1,2,\cdots 30).
\end{aligned}
\tag{6}
$$

为说明摸石头过河算法的优势,与当前最流行的粒子群优化算法(Particle Swarm Optimization Algorithm,PSO)作比较. 摸石头过河算法和改进的摸石头过河算法参数设置如下:$m=200$,$n_{\max}=100$,$N=1\ 000$,$L=L_0=0.1$,$\alpha=0.99$. 粒子群算法的参数如下:粒子数 50,$c_0=1$,$c_1=2$,$c_2=2$,$V_{\max}=0.5$. 3 种算法的迭代过程如图 2 所示. 从图 2 可以看出,改进的摸石头过河算法的收敛速度最快.

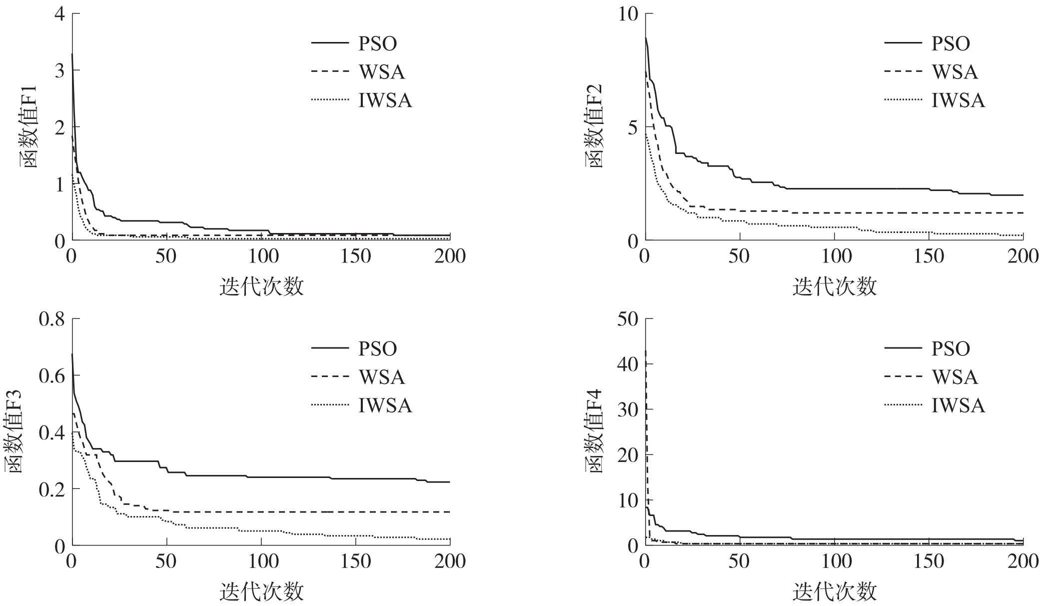

图 2　3 种算法的比较

Fig. 2　Comparisons of three algorithms

3　结束语

根据"摸石头过河"的思想,提出一种快速随机优化算法来求解连续空间优化问题,该算法易于设计和实现. 测试结果表明,该算法具有搜索速度快、精度高和不易陷入局部极值点的特点,因而具有较好的全局搜索能力,其应用前景非常广泛. 该方法具有一定潜力,值得推荐.

[参考文献]

[1]　Antamoshkin,Alexander N,Kazakovtsev,et al. Random search algorithm for the p-median problem[J]. Informatica(Slovenia),2013,37(3):267-278.

[2]　葛振振,周军,林鹏. 采用改进模拟退火算法的高速飞行器随控总体优化方法[J]. 宇航学报,2013,34(11):1 427-1 433.

[3]　黎渊,蒋江,张民选,等. 基于模拟退火算法的浮点转定点自动位宽优化工具[J]. 上海交通大学学报:中文版,2013,47(1):76-80,85.

[4]　张琦,马家辰. 基于改进蚁群算法的移动机器人路径规划[J]. 东北大学学报:自然科学版,2013,34(11):1 521-1 524.

[5]　Eberhart R C,Kennedy J. A new optimizer using particles swarm theory[C]//Proc Sixth International Symposium on Micro

Machine and Human Science. Nagoya,Japan:IEEE Press,1995:39-43.
[6] Shi Y H,Eberhart R C. A modified particle swarm optimizer[C]//IEEE International Conference on Evolutionary Computation. Anchorage. Alaska:IEEE Press,1998:69-73.
[7] Kennedy J,Eberhart R. Particle swarm optimization[C]//Proc IEEE International Conference on Neural Networks. Perth:IEEE Press,1995:1 942-1 948.
[8] 郭通,兰巨龙. 自适应的分数阶达尔文粒子群优化算法[J]. 通信学报,2014,35(4):130-140.
[9] 胡旺,GaryGYEN,张鑫. 基于 Pareto 熵的多目标粒子群优化算法[J]. 软件学报,2014,25(5):1 025-1 050.
[10] 高尚,杨静宇. 群智能算法及其应用[M]. 北京:中国水利水电出版社,2006:112-117.
[11] 高尚,杨静宇. 混沌粒子群优化算法研究[J]. 模式识别与人工智能,2006,19(2):266-270.

[责任编辑:黄　敏]

(上接第107页)

[8] Ojansivu V,Heikkilä J. Blur Insensitive Texture Classification Using Local Phase Quantization[M]//Image and Signal Processing. Heidelberg,Berlin:Springer,2008:236-243.
[9] Heikkila J,Ojansivu V,Rahtu E. Improved blur insensitivity for decorrelated local phase quantization[C]//20th International Conference on Pattern Recognition. Istanbul:IEEE,2010:818-821.
[10] Lei Z,Ahonen T,Pietikainen M,et al. Local frequency descriptor for low-resolution face recognition[C]//IEEE International Conference on Automatic Face & Gesture Recognition and Workshops. Shanghai,China:IEEE,2011:161-166.
[11] Chan C H,Tahir M A,Kittler J,et al. Multiscale local phase quantization for robust component-based face recognition using kernel fusion of multiple descriptors[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence,2013,35(5):1 164-1 177.

[责任编辑:黄　敏]