模拟退火算法收敛性的研究

冯玉蓉, 朱均燕
(福建农林大学计算机与信息学院 福建 福州 350002)

【摘 要】: 模拟退火算法是一个全局最优的有效算法。本文在传统的模拟退火算法的基础上, 引入 Marquardt-Levenberg 方法,来加快该算法的计算速度,增强其算法的收敛性。

【关键词】: 模拟退火算法;Marquardt-Levenberg 方法;收敛性;计算速度

1. 引言

数学建模是20世纪的理学基石。这不仅是因为通过数学建模可以推进当前的科技进步,更重要的是有助于理解新的、复杂的现象或过程。尤其是在人们通过数学建模这种方法,运用相应的算法所获得的最优值与人们通过经验所得到的值相符时,更多的数学建模方法进入了理学领域。众所周知,现实世界是非线性的、随机性的,所以一些数理统计方法、非线性和非凸的函数形式等常用来作为数据的处理工具。本文所提到的"模拟退火算法"的建立就是基于这一观点。

2. 模拟退火算法

模拟退火算法是根据固态或液态材料中粒子的统计力学与复杂组合最优化问题的求解过程的相似之处而提出来的[1]。它是一种随机性的近似算法, 通过采用随机模拟物体退火过程来完成问题的求解,也就是在控制参数(温度)的作用下对参数的值进行调整, 直到所选取的参数值最终使能量函数达到全局极小值。该算法与其它搜索方法相比,具有如下的特点:(1)以一定的概率接受恶化解;(2)引进算法控制参数;(3)使用对象函数值进行搜索;(4)搜索复杂区域。

该算法的具体实施方案如下:

(1) 初始时刻 t=1,在足够高的初始温度 T(t)下,随机选取一个初始态 S_0,计算 $E(X_t)$;

(2) 根据某种跃迁分布,随机地从 X_t 跳到 X_{t+1};

(3)计算 $E(X_{t+1})$,如果 $E(X_{t+1})< E(X_t)$,则用 X_{t+1} 取代 X_t,否则计算其接受几率 $p=\exp\{-[E(X_{t+1})- E(X_t)]/T(t)\}$,且产生一个(0,1)内均匀分布的随机数 r,比较 p 与 r,如果 p>r,则用 X_{t+1} 取代 X_t,否则保持不变;

(4)重复(2)、(3)步骤,直到在该温度下找到一个全局最优解;

(5)根据降温公式,计算下一步的温度 T(t+1),回到(2)重复以上步骤,直到 E 达到某个预定的精度范围之内,算法结束。

可见, 模拟退火算法是一个在同一温度下寻找最优解和在降温过程中寻求在预定精度范围内的最优解的双重循环的算法,因此可以认为跃迁分布函数、接受准则和降温函数是该算法的核心部分。

3. 改进的模拟退火算法

3.1 传统的模拟退火算法

传统的模拟退火算法是指具有 Gaussian 分布和 Markovian 链的系统。其跃迁函数为 Gaussian 分布,表达式如下:

$$g(x)=(2\pi T)^{-D/2}\exp[-\Delta x^2/(2T)] \quad (1)$$

这里,$\Delta x=x_{t+1}-x_t$,T 是当前温度,D 是状态变量 x 的维数。

是否接受当前状态的解由 Metropolis 准则来判定, 其公式如下:

$$p=\min[1,\frac{1}{1+\exp(\Delta E/T)}] \quad (2)$$

算法中的降温函数如下式所示:

$$T_{k+1}=\alpha T_k \quad (3)$$

式中 α 通常接近 1,科克派特里克等选用 0.9。[1]

给定跃迁分布函数 g(x)、可以证实在温度函数 T_{k+1} 的 度下降的速度比较慢的情况下、我们可以得到一个全局最优 ,可见,为了获得这个最优解是以收敛性的速度作为代价的。同时不能排斥该算法可能局限于在某一温度下的能量局部最小而不能跳出,而找到一个伪最优解。为了避免这种情况的出现,以及加快该算法的收敛性,有必要对其做进一步的研究,笔者试着从这个方向对该算法做了改进。

3.2 改进的模拟退火算法

跃迁分布函数、接受准则和降温函数是模拟退火算法的三大支柱,在许多的参考资料中,都探讨了降温函数的选择对该算法速度的改进,Metropolis 准则是大家所公认的接受准则, 唯独对跃迁分布函数的探讨相对较少。本文试从对跃迁分布函数的选择分析,试图找到提高该算法收敛性的速度。

而在改变该算法的计算速度方面已经有了方法, 比如将 Gaussian 分布用文献[2]中提到的 Canchy-Lorentz 分布代替,表述如下:

$$g(\Delta x)\propto\frac{T}{(T^2+(\Delta x)^2)^{\frac{D+1}{2}}} \quad (4)$$

该分布是一个半局域分布,在同样的温度下,与 Gaussian 分布相比较,Canchy-Lorentz 分布有更大的机会进行较长距离的跃迁,从而使得在该一温度下所进行的搜索时间变短了, 加快了其计算速度。虽然其分布函数改变了,但依然要确定其 Markovian 链长,这仍然是一件让人很繁闷的事情。

降温函数对于模拟退火算法来说是及其重要的。它可以控制算法的收敛速度, 构造新的降温函数能解决算法搜索较缓慢的问题。在传统的模拟退火算法中其降温函数采用的是下一次的温度值是当前温度值与一常数的乘积, 虽然对于常数的取值做了规定, 但是不能很好解决搜索中算法可能长期隐于局部点而无法跳出的情况。有一些资料提到可以采用如下的降温函数:

$$T_{k+1}=\frac{T_0}{t} \quad (5)$$

式中 t 为降温时间,T_0 为初始降温的温度。

这里为了提高模拟退火算法的收敛性, 引入了 Marquardt-Levenberg 方法[3],该方法是在牛顿迭代法的基础上进行改进的。它克服了牛顿迭代法的两个缺陷,即:1)避免了初始点的选择一定要在最优点的邻域内,这让我们可以随机产生一个初始点,而不必为初始解大伤其神;2) 不要求目标函数的二阶导数阵是非奇异的。在该方法中,其迭代公式的方向为:

$$P_k=-[G^{(k)}]^{-1}\nabla f(x^{(k)}) \quad (6)$$

其中

$$G^{(k)}=[\nabla^2 f(x^{(k)})+\beta_k Q^{(k)}] \quad (7)$$

在这里,取步长为 1,β_k 是一个非负数,$Q^{(k)}$为一个给定矩阵,当然它也可以是一个与 k 无关的矩阵, 只是它至少应该是半正定的。

Marquardt-Levenberg 方法是利用目标函数的导数进行最优值寻优的迭代算法。如果在模拟退火算法中引入该方法作为跃迁函数,首先它避免了寻找 Markovian 链长,其次由于该算法的快速收敛性可以缩短最优解的寻找时间, 提高模拟退火算法的收敛性。

3.3 算法的实现步骤

引入 Marquardt-Levenberg 方法没有改变模拟退火算法的

双重循环性质，它仍然是要进行两重循环计算来找到目标值。对于$\min_{x\in R^n} f(x)$的全局优化问题，其算法实现步骤如下：

1) 给定初始温度 T_0，初始解 $S_0,\beta_0,Q,\varepsilon,\alpha>1$；

2) 在初始温度 T_0 下，用 Marquardt-Levenberg 迭代法进行搜索，其迭代步骤如下：

a) $k \Leftarrow 0$ ；

b)求出 $\nabla f(x^{(k)}),\nabla^2 f(x^{(k)}),\beta\Leftarrow\beta_0$；

c) $G=\nabla^2 f(x^{(k)})+\beta Q$ ；

d)若 G-1 存在，则求出 G-1 转向 e；若 G-1 不存在，则 转向 c；

e)求 $P_k=-G^{-1}\nabla f(x^{(k)})$ ；

f)求 $x^{(k+1)}=x^{(k)}+\lambda_k P_k$，其中 λ_k 满足 $f(x^{(k)}+\lambda_k P_k)=\min_{\lambda}(x^{(k)}+\lambda P_k)$，若 $\lambda_k=0$，则 $\beta\Leftarrow\alpha\beta$ 转向 c；若 $\lambda_k\neq 0$，则转向 g；

g) 若 $\|x^{(k+1)}-x^{(k)}\|\leq\varepsilon$，则求出最优解 $x^*=x^{(k+1)}$，若 $\|x^{(k+1)}-x^{(k)}\|>\varepsilon$，则令 $k\Leftarrow k+1$，并转向 b。

3) 计算 $f(x^*)$的值，并令 $S^*\Leftarrow f(x^*)$；

4) 计算增量 $\Delta s=S^*-S_0$；

5) 若 $\Delta s<0$，则接受 S^*；否则以概率 $\exp(-\Delta k/T)$接受 S^* 作为新的当前解；

6) 如果 S^* 满足终止条件则输出当前解作为最优解，终止程序；

7) 将 $S^*\Rightarrow S_0$，按照降温函数降温，重复 2)，3)，4)，5)，6)。

对步骤 2)中的$\nabla f(x^{(k)})$和$\nabla^2 f(x^{(k)})$的说明，它们分别为目标函数的一阶、二阶导数，由于这里的 $x\in R^n$，所以它们应该是函数对矩阵的导数，故它们是矩阵，其行数、列数与 x 的维数有关。

4. 仿真实验

以

$$f_1(x_1,x_2)=3/2x_1^2+1/2x_2^2-x_1x_2-2x_1$$

$$f_2(x_1,x_2)=e^{x_1}(4x_1^2+2x_2^2+4x_1x_2+2x_2+1)$$

$$f_3(x_1,x_2,x_3)=(x_1-x_2+x_3)+(-x_1+x_2+x_3)^2+(x_1+x_2-x_3)^2$$

为目标函数，求其最小值。用本文提出的改进的模拟退火算法与传统的模拟退火算法进行比较。给定初始温度 $T_0=10^6$，$S_0=(-2,4)$，，$\beta_0=1$，$Q=[1,0;0,1]$，$\varepsilon=1e-4$，其仿真结果见下表。

模拟退火算法 / 参数 / 函数	运算时间（s）		未收敛次数		调用函数次数	
	传统	改进	传统	改进	传统	改进
f_1	45	37	3	0	85	60
f_2	108	80	8	0	94	50
f_3	97	55	5	0	60	36

表一、算法仿真结果

5. 小节

从仿真结果看，改进的模拟退火算法在运算时间上比传统的模拟退火算法少，未收敛次数为 0，调用函数的次数也较少，可见确实加快了其运算时间和增强了算法的收敛性。下一步应该将该算法与实际的应用结合起来，使该算法可用于多变量，且每一变量区域的限制是不同的目标函数中，加快其算法时间和其收敛性。

参考文献：

1. Nirwan Ansari Edwin Hou 著.用于最优化的计算智能[M].李军，边肇祺译.北京，清华大学出版社，1999.P27-40.

2. 向阳，龚新高.推广模拟退火方法及其应用[J].物理学进展，Vol.20,No.3,Sept.,2000

3. 席少霖，赵凤治.最优化计算方法[M].上海，上海科学技术出版社，1983.P74-95.

4. Lester Ingber,Very Fast Simulated Re-Annealing,Mathl.Comput.Modelling,V.12,p967-973,1989

5. Global Optimization of Statistical Functions with Simulated Annealing, Journal of econometrics, vol. 60, no. 1/2, Jan./Feb. 1994

6. Yang ruoli and Gu jifa. An efficient simulated annealing algorithm for global optimization. Journal of System Engineering Theory and Practic. 1997,17(5)

（上接第 49 页）

常构件只提供接口，测试者无法调用构件的其它方法。但通过触发器，测试者可任意调用构件的方法。构件的属性可通过探头的数据通道访问。

4. 总结

探针模型是构件测试的关键技术，它解决了被测构件与外界交互的问题，使得构件测试成为可能。本文对 COM 构件测试的模型进行分析与研究。根据 COM 构件的特点及探针模型在 JavaBean 上的实现原理，提出并实现了在 Visual C++环境下的 COM 构件测试的探针模型。并针对探针模型的现有缺点对探针结构和功能作如下改进：

1. 传输的数据类型有限、结构化等复杂的数据类型的探针制作繁琐及探针无法复用的缺点，提出相应的解决方法，把原子的数据类型编写成可复用的标准数据通道，以利于探针的复用。把结构、联合等复杂的数据类型原子化，即探针传输的是复杂数据类型的变量的原子分量。

2. 对于部分检测结果直接以消息的形式，由消息发送器向外传输。

3. 为解决探头与构件相互访问的递归性问题，本文把触发器与探头分离。由触发器访问构件，再由构件调用探头。

4. 编写标准的复用率高的数据通道代替一次传输多个数据的探针形式。为便于探针的复用同时也为探针自动插装作准备，用多个基本数据类型的数据通道代替每个插装点一个探针。

探针模型解决了构件测试的瓶颈，但探针模型并不完善。首先，探针模型必须人工插装探针；其二，探针的数据传输部分还需人工完成。其三、探针模型与被测构件捆绑在一起，无论哪个出现错误，则整个系统崩溃，这对于以测试为目的的探针模型极其不利。因此，探针模型推广应用，还要解决探针的插装的自动化技术、探针的标准化和复用、探针与构件的自动结合和剥离以及改变探针模型被动激发状态，转变为探针模型激发构件，当被测构件出错时，不至于要重启系统。目前，这些技术还在研究当中。随着构件技术的广泛应用，基于探针模型的构件测试技术的前景更好美好！

参考文献：

1. Robert V.Binder 面向对象系统的测试 [M] 人民邮电出版社，2001，4 华庆一 王斌君 陈莉 译

2. 景 涛 白成刚 胡庆培 蔡开元 构件软件的测试问题综述 [J] 计算机工程与应用，2002,24:1-6

3. David J.Kruglinski Scot Wingo Programming Microsoft Visual C++ 6.0 技术内幕(第五版) [M] Microsoft Press

4. 潘爱民 COM 原理与应用 [M] 清华大学出版社，1999.12

5. Dale Rogerson COM 技术内幕-微软组件对象模型 [M] Microsoft Press，1998.12

6. Richard C.Leinecker COM+技术大全[M] 机械工业出版社，2001.8 高智勇等译

7. John Paul Mueller COM+开发指南[M] 清华大学出版社，2000.12 天宏工作室内 译

8. Abernethy.R COM/DCOM 技术内幕[M] 汪浩电子工业出版社，2000.1

9. 方艳 Java 软件分析与测试工具的研究与开发[D] 北京航空航天大学，20010301

10. 周俊 JavaBean 构件测试与分析技术的研究和实现[D] 北京航空航天大学，20030228

11. WatKins.J 实用软件测试过程 [M] 机械工业出版社，2004.1 贺红卫等译

12. Patton.R 软件测试[M] 北京工业出版社，2002.2

13. McGregor.J.D 面向对象的软件测试[M] 机械工业出版社，2002.8