

模拟退火算法改进综述及参数探究

卢宇婷, 林禹攸, 彭乔姿, 王颖喆

(北京师范大学 数学科学学院, 北京 100875)

[摘 要] 回顾模拟退火算法流程,对模拟退火算法现有的改进方法进行了系统的总结与评价.将模拟退火算法应用于求解 Sobol'g 函数的最小值,进而对模拟退火算法的三个关键参数:降温函数、初末温和马氏链长度进行探究.

[关键词] 模拟退火算法;改进;总结;Sobol'g 函数;参数

[中图分类号] O29 [文献标识码] C [文章编号] 1672-1454(2015)06-0096-08

1 引 言

模拟退火算法 (Simulated Annealing Algorithm, SAA) 是一种应用广泛的随机智能优化算法,最早的思想是由 Metropolis 等人于 1953 年提出.1983 年, Kirkpatrick 等成功地将退火思想引入到组合优化领域.在求解组合优化问题时,模拟退火算法能概率性地跳出局部最优解并最终趋于全局最优.

模拟退火算法发展至今,已广泛应用于各个领域求解组合优化问题.当下这个时代是大数据的时代,面对海量的数据,如何处理数据、提取信息,是一个重要的问题,这也对优化算法有着越来越高的要求,不断有国内外学者对模拟退火算法进行研究改进,以提高传统模拟退火算法的性能.模拟退火算法的改进方法非常纷杂,至今没有一个较为系统完整的归纳总结,本文提供的较为系统的模拟退火算法总结综述以及算法中重要参数的研究将为今后的学者学习研究模拟退火算法带来极大的便利.本文将传统模拟退火算法应用求解 Sobol'g 函数以得出算法中关键参数对算法性能的影响. Sobol'g 函数有如下形式:

$$Y = \prod_{i=1}^5 g_i(X_i),$$

其中 $g_i(X_i) = \frac{|4X_i - 2| + a_i}{1 + a_i}$, $(a_1, \dots, a_5) = (0, 1, 4.5, 9, 99)$, $X_i \in [0, 1]$.

Sobol'g 函数为复杂天气系统中一个重要的实验函数^[1].

2 模拟退火算法

2.1 算法概述^[2]

在给定的控制参数初值下,算法随机地从可行解出发,持续进行“产生新解—判断—接受/舍弃”的迭代过程,在迭代递减时产生一系列的 Markov 链,通过计算系统的时间演化过程,逐步逼近问题的最优解.停止准则达到后,根据控制参数衰减函数减小控制参数的值,重复进行上述步骤,就可以在控制参数达到终止时,求得组合优化问题的整体最优解.

[收稿日期] 2015-09-29

[基金项目] 985 资助项目(国家级大学生创新创业训练计划(3010400040))

2.2 算法流程

设目标函数为 $y = f(x)$, 传统模拟退火算法流程如下:

Step1: 选定初始控制温度 T_0 , 马氏链长度 L_0 , 在可行解空间中随机选取一个初始解 i_0 , 此时, 最优解 $i = i_0$, 迭代次数 $k = 0$, 降温函数(即控制参数衰减函数) $T_k = h(k)$;

Step2: 产生一次随机扰动, 在可行解空间中得到一个新解 j ;

Step3: 判断是否接受新解, 判断准则为 Metropolis 准则:

(i) 若 $f(i) \geq f(j)$, 则接受新解 j , 此时最优解 $i = j$,

(ii) 若 $f(i) < f(j)$, 则依概率接受新解 j , 即 $\exp\left(\frac{f(i) - f(j)}{T_k}\right) > \text{random}[0, 1)$ 时, 接受新解 j ,

此时最优解 $i = j$, 否则, 拒绝 j , 此时最优解仍为 i ;

Step4: 重复执行 L_0 次 step2 和 step3, 得到链长为 L_0 的马氏过程下的一个最优解;

Step5: 判断是否满足停止准则, 若满足则输出最优解, 算法停止, 否则执行 step6;

Step6: 迭代次数 $k = k + 1$, 最优解更新为 step4 得到的解, 温度函数变成 T_{k+1} , 马氏链长度变为 L_{k+1} , 回到 step2.

3 模拟退火算法改进综述

虽然传统的模拟退火算法已经有着较为良好的收敛性, 但是, 随着数据处理量的日益增大以及所需结果的日益精细, 对随机优化算法有着越来越高的要求, 越来越多的学者深入学习模拟退火算法并对其进行适当地改进, 使得模拟退火算法具备更优良的性质. 下面介绍九种常见的, 广为接受的改进算法.

3.1 改进算法

3.1.1 加温退火法^[3]

改进初温和初始解的选取, 改进流程如下:

先令 step1 中的 $T_0 = 0$, 执行 step2, 然后确定是否接受新解, 接受准则为当且仅当 $f(i) < f(j)$ 时接受新解 j , 同时令 T_0 按照某个事先确定的升温函数 $h^*(t)$ 增加, 初始解变为 j , 否则仍为 i , 如此重复, 直至满足升温停止准则, 输出所得温度 T_0^* 和初始解 i^* , 分别以 T_0^* 和 i^* 作为 step1 中的初始控制温度和初始解开始退火, 即执行 step2—step6, 每一次接受新解时都令温度按照 $h^*(t)$ 增加.

改进效果: 提高算法效率(即缩短运行时间), 降低多次重复实验所得解的离散性.

不足: 多次重复实验下所得解的最优解略差.

3.1.2 有记忆的模拟退火算法^[3]

在 step1 中增加一个记忆矩阵 I 和一个记忆函数 F , 开始时, I 中只有一个元素 i_0 , $F = f(i_0)$, 在 step3 中, 每一次接受新解 j 时, 都将 F 与 $f(j)$ 作对比, 若 $f(j) < F$, 则令 $F = f(j)$, 同时将 j 存入 I 中. 在算法停止后, 将所得最优解与 I 中记录的解作对比, 选出最优者作为算法的最终解.

改进效果: 提高解的质量.

不足: 可能导致算法效率降低.

3.1.3 带返回搜索的模拟退火算法^[3]

先用有记忆的模拟退火算法对初始求解, 在退火结束时, 再对所得最终解施行局部搜索算法, 直至局部搜索过程结束, 输出最终解.

改进效果: 提高解的质量.

不足: 可能导致算法效率降低.

3.1.4 回火退火法^[3]

在 step2 中, 仅在当前解的一个 q 领域内选取新解 j , q 是可变的, 当温度降至较低时, 跳出局部最优变得困难, 此时可增大温度值, 再继续退火.

改进效果: 提高解的质量.

不足: 可能导致算法效率降低.

3.1.5 保持温度不下降改进最优解的选取方法^[4]

此方法适用于离散情形的优化问题,在整个算法的执行中,保持温度不下降,即降温函数使用常值函数 $T_k = T > 0$,估计最优解的方法有两种,一种是记录下所有迭代中算法最常访问的点,将这个点作为最优解,另一种是用具有最优目标函数估计值的状态作为最优解的状态,用这个值作为最优解估计值.

改进效果:提高算法效率.

3.1.6 改进新解产生方式^[5]

在 step2 中,每次独立产生 n 次随机扰动,得到 n 个解,选取这 n 个解中最好解 j ,令 $k = k + n$,接着执行 step3—step6.

改进效果:弥补冷却进度表选择不当导致的算法的不稳定性.

不足:可能会降低算法效率.

3.1.7 混合模拟退火—遗传算法^[6-8]

在遗传算法的基础上,将遗传算法^[9]的新解接受机制改为模拟退火算法中的 Metropolis 准则,将变异操作与模拟退火的结合,由此模拟退火算法和遗传算法共同决定优化问题解的搜索范围.

改进效果:提高解的质量.

不足:可能导致算法效率降低.

3.1.8 一种高效的模拟退火全局优化算法^[10]

改进 step6 中温度函数,一方面提出确定温度函数的启发式准则:令 A 为解空间中任意给定的包含坐标原点的有界区域,令正整数 k' 为任意迭代次数,温度函数应使得当所有迭代次数 k 满足 $k > k'$ 时,所产生的新解全部落在 A 内的概率等于零,温度函数应与 k 的幂函数成反比,与优化问题的变量维数无关.另一方面,构造产生随机向量的概率密度函数.依照这一启发式准则及特定的产生随机向量的概率密度函数确定出的温度函数 $T_k = \frac{T_0}{k^m}$ (其中 T_0 为初始温度, $m \geq 1$ 为给定常数),再结合它的产生随机向量的概率密度函数得到的新算法能有效地提高算法运行效率.

改进效果:提高算法效率.

3.1.9 一种改进的模拟退火组合优化算法^[11]

改进算法的停止准则.首先在每一给定温度下,由 $ss(k) = \{s(k) \text{ 和 } s(k-1) \text{ 中目标函数值更小的那个}\}$, $k = 1, 2, \dots$ 构造序列 $ss(k)$, 则 $ss(k)$ 的目标函数值随 k 增大是一个单调递减序列. 其次定义用于记录每个温度下的最优解的序列 $S(T_i) = \{T_i \text{ 下 } ss(k) \text{ 的最优值}\}$ ($i = 1, 2, \dots$). 改进的停止准则为:事先给定一个阈值 p , 当温度连续下降 p 次,而 $S(T_i)$ 没有改变时,算法停止,输出最优解 $S(T_i)$.

改进效果:提高解的质量.

3.2 结 论

先介绍模拟退火算法中的两个重要概念:

1. 冷却进度表,包含以下参数:初温 T_0 ,降温函数 T_k ,末温(即停止准则),Markov 链长度 L_k ;
2. 移动策略:接受新解的准则.

从 3.1 中九种改进算法可以直观地看出,算法的改进集中在改进冷却进度表和移动策略.此外,从改进效果和不足可以直观地看出,解的精度与算法效率往往不可兼得.提高解的质量往往会伴随着算法效率降低,提高算法效率往往会伴随着解的质量降低.因此在对模拟退火算法进行改进时,应着重考虑改进冷却进度表和移动策略,并且根据所需改进效果,适当牺牲解的精度或者解的质量,以寻求更有效的应用.

4 对模拟退火算法中冷却进度表有关参数的探究

模拟退火算法实施的关键之一是冷却进度表.冷却进度表规定了控制参数 T 的初值 T_s 、控制参数 T 的衰减函数、控制参数 T 的终值 T_e (停止准则)和 Markov 链的长度 L_k .

下文中简记控制参数 T 的初值 T_s 为“初温”,控制参数 T 的终值 T_e 为“末温”,控制参数 T 的衰减函数为“降温函数”,解的精度为“解质”.此外,下文中提到的解和运行时间为重复多次试验取平均得到的平均解和平均运行时间.

4.1 降温函数

算法的运行时长与控制参数 T 的变化次数及马氏链的长度 L_k 成正比.较为常见的控制参数 T 的降温函数为

$$T_k = \frac{T_s}{k}, T_k = T_s \times 0.99^k \text{ 及 } T_k = \frac{T_s}{\ln(k)}.$$

通过观察图 1,可比较得出上述三种情况下控制参数变化的速度快慢情况——降温函数为 $T_k = \frac{T_s}{k}$ 时控制参数变化的最快, $T_k = \frac{T_s}{\ln(k)}$ 在 k 较小的情况下变化速度比 $T_k = T_s \times 0.99^k$ 更快, k 较大的情况下 $T_k = T_s \times 0.99^k$ 的变化速度比 $T_k = \frac{T_s}{\ln(k)}$ 和 $T_k = \frac{T_s}{k}$ 快.

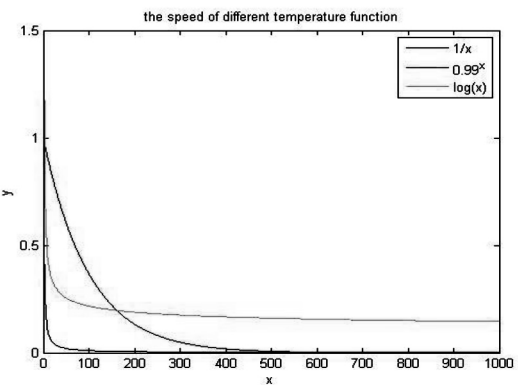


图 1 不同降温函数的变化速度

通过图 1,对不同降温函数的衰减速度进行了粗略地定性分析.为了进一步了解上述降温函数的衰减速度,我们对其进行定量分析,详见表 1.

表 1 不同降温函数在不同初末温下变化次数比较

变化次数 $T_s : T_e$	降温函数	$T_k = \frac{T_s}{k}$	$T_k = T_s \times 0.99^k$	$T_k = \frac{T_s}{\ln(k)}$
1000 : 1		1000	688	Inf
500 : 1		500	618	1.4036×10^{217}
300 : 1		300	568	1.942×10^{130}
200 : 1		200	528	7.226×10^{86}
200 : 2		100	459	2.688×10^{43}
100 : 2		50	390	5.185×10^{21}
100 : 1		100	459	2.688×10^{43}
50 : 1		50	390	5.185×10^{21}

由表 1 可知,控制参数 T 的变化次数与 $m = T_s : T_e$ (控制参数 T 的初值/终值)呈正相关.衰减函数为 $T_k = \frac{T_s}{\ln(k)}$ 时,控制参数的变化次数为 e^m 次,变化次数远远多于其余两种降温函数,算法运行时间特别长(多于 4 个小时).此外还可知 m 较大时,降温函数为 $T_k = T_s \times 0.99^k$ 时的运行时长小于 $T_k = \frac{T_s}{k}$ 的运行时长;而当 m 较小时,降温函数为 $T_k = T_s \times 0.99^k$ 时的运行时长大于 $T_k = \frac{T_s}{k}$ 的运行时长.用 Sobol' g 函数进行实例验证:

由图 2,图 3 可知,三种降温函数进行对比, $T_k = \frac{T_s}{k}$ 的解质基本上保持在 $10^{-4} \sim 10^{-5}$, $T_k = T_s \times 0.99^k$ 的解质保持在 10^{-5} ,在初温较低时优于 $T_k = \frac{T_s}{k}$ 的解质,初温较高时没有 $T_k = \frac{T_s}{k}$ 的解质好.降温函数为 $T_k = \frac{T_s}{k}$ 时解质受温度比影响很大,而降温函数为 $T_k = T_s \times 0.99^k$ 时的解受温度比影响较小.当初温比较高时,两种降温函数算得的解差不多,都缓慢趋于 0.前两种降温函数的解对应的最小温度比都是在 50 : 1,但是 $T_k = \frac{T_s}{\ln(k)}$ 在温度比为 10 : 1 时的解质就已经有 10^{-7} 了,远好于前两种降温函数,所以可

以推测,初末温比为 $100:1$ 时, $T_k = \frac{T_s}{\ln(k)}$ 的解质将远优于相同条件下 $T_k = \frac{T_s}{k}$ 和 $T_k = T_s \times 0.99^k$ 的解质. 关于运行时间,在温度比小于 700 时,运行时间 $T_k = \frac{T_s}{k} < T_k = T_s \times 0.99^k$; 在温度比大于 700 时,运行时间 $T_k = \frac{T_s}{k} > T_k = T_s \times 0.99^k$. 而对于温度函数 $T_k = \frac{T_s}{\ln(k)}$ 来说,温度比仅为 $10:1$ 时的运行时间就达到 29.9041 s,已经远远超过试验中前两种函数的最大运行时间,若初温更高, $T_k = \frac{T_s}{\ln(k)}$ 的运行时间将更长,难以应用于解决复杂的实际问题.

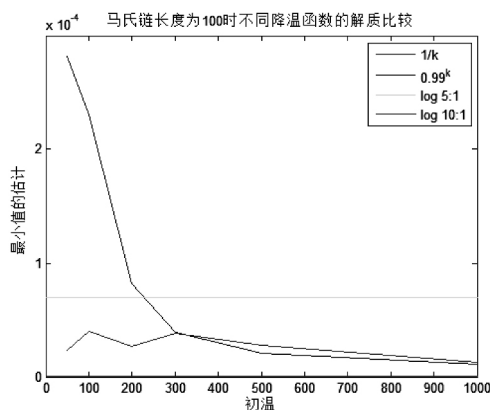


图2 不同降温函数解质比较

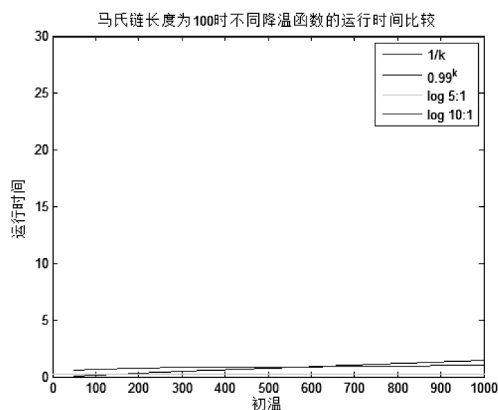


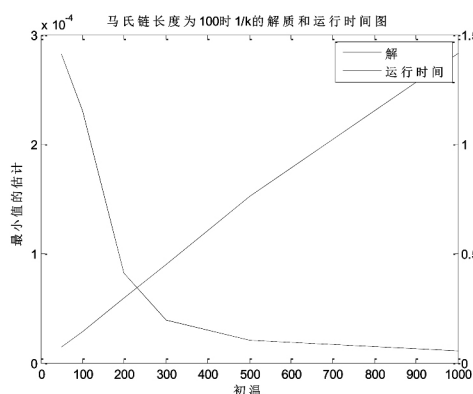
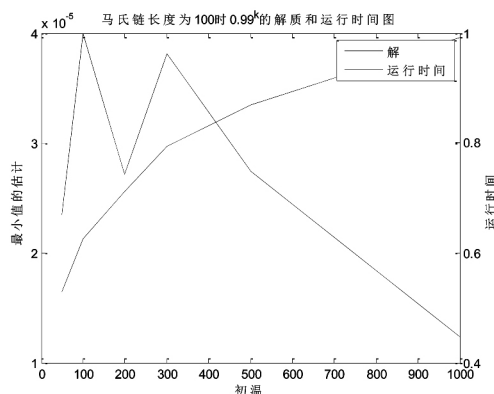
图3 不同降温函数运行时间比较

4.2 初温、末温

冷却进度表中的另一关键参数就是初末温的设置. 根据模拟退火算法解的收敛性理论可知,当初温足够高,降温过程足够慢时,算法才可以收敛到全局最优解. 温差越大,解越精确,但相应的运行时间也会增加. 我们试图探究初温末温设置和解质、运行时间之间的关系,下面继续用 Sobol'g 函数来进行实验.

(i) 给定马氏链长度为 100 ,末温为 1 ,初温分别为 $50, 100, 200, 300, 500, 1000$:

降温函数分别是 $T_k = \frac{T_s}{k}$ 和 $T_k = T_s \times 0.99^k$,解质和运行时间如图4,图5:

图4 $T_k = \frac{T_s}{k}$ 的解质与运行时间图图5 $T_k = T_s \times 0.99^k$ 的解质与运行时间图

由图4可知,当初温升高时,降温函数为 $T_k = \frac{T_s}{k}$ 时解质越来越好,有数量级上的改变. 初温小于 300 时解质下降的速度较快, 300 之后解质下降的速度较慢. 可知降温函数为 $T_k = \frac{T_s}{k}$ 的解质对初温变化很敏感. 由图5可知,当初温升高时,降温函数为 $T_k = T_s \times 0.99^k$ 时解质一开始呈现出波动,后来逐渐减小,但没有数量级上的改变. 可知降温函数 $T_k = T_s \times 0.99^k$ 的解质在变化初温时比较稳定. 运行时

间方面,两种降温函数的运行时间均与温度比呈正比,温度比越高,运行时间越长.

降温函数是 $T_k = \frac{T_s}{\ln(k)}$,解质和运行时间见表 2(由于运行时间实在太长,只做了两组数据,末温为 1):

表 2 $T_k = \frac{T_s}{\ln(k)}$ 的解质与运行时间表

初末温比	解质	运行时间/(s)
10 : 1	6.0234e-007	29.9041
5 : 1	6.9544e-005	0.20123

由上面两组数据可以观察到,初温越高,解质越好,呈现出数量级上的改变.运行时间大幅增加.

(ii) 马氏链长度固定为 200,初温为 200,末温分别为 1、2、5、10、20、50,降温函数分别是 $T_k = \frac{T_s}{k}$ 和 $T_k = T_s \times 0.99^k$ 时,解质和运行时间如图 6,图 7 所示:

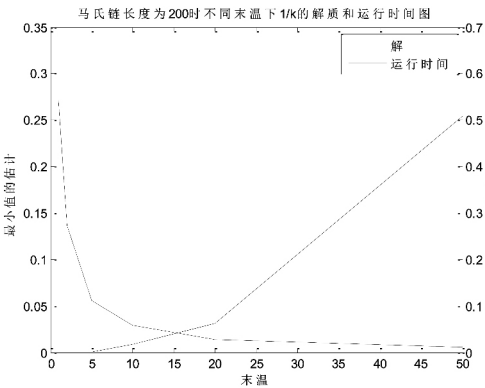


图 6 $T_k = \frac{T_s}{k}$ 的解质与运行时间图

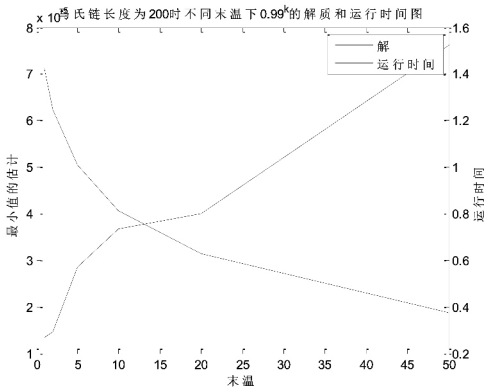


图 7 $T_k = T_s \times 0.99^k$ 的解质与运行时间图

由图 6 可知,降温函数为 $T_k = \frac{T_s}{k}$ 时末温越高,解质就越差,变化幅度较大,数量级从 10^{-5} 变化至 10^{-1} ,可见该降温函数的解质受末温影响大.由图 7 可知,降温函数是 $T_k = T_s \times 0.99^k$ 时,解质呈现先快再缓后快的上升趋势,但是幅度不是很大,数量级始终保持在 10^{-5} .两种降温函数下运行时间均随末温增加而减少.

马氏链长度是 200,降温函数是 $T_k = \frac{T_s}{\ln(k)}$,初温是 10,末温分别是 1、2、3、4、5 时,解质和运行时间如图 8 所示:

随着末温的升高,解质变差,速度先慢后快,幅度较大;运行时间减少,先快后慢,幅度也有较大的变化.

4.3 马氏链长度

在每一温度下,可以用一个马氏链来刻画状态空间中的状态转移,其中的转移概率就是每一个当前解会转移到新解的概率.在每一温度下,我们重复多次产生新解并去判断是否转移,而产生新解的次数就是对应的马氏链的长度.下面来探究马氏链长度对解质和运行时间的影响.

温度为 100 : 1 时,随马氏链长度增加,三种降温函数解质及运行时间变化如图 9、图 10 所示(log 的是温度为 10 : 1 的结果):

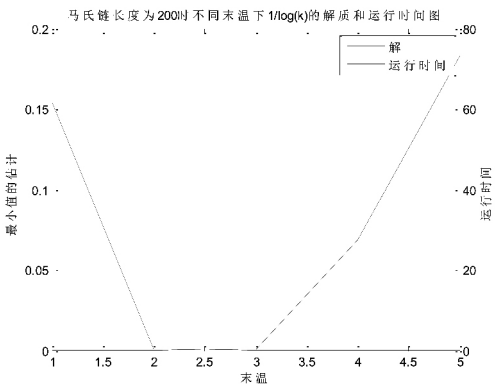


图 8 $T_k = \frac{T_s}{\ln(k)}$ 的解质与运行时间图

由图 9,图 10 可知,随马氏链长度的增大,解的精确度就越高.对降温函数 $T_k = \frac{T_s}{k}$ 和 $T_k = T_s \times 0.99^k$,马氏链长度小于 1000 时解质的变化较快,马氏链长度大于 1000 时解质的变化很缓慢.运行时间随马氏链长度增大呈线性增长.

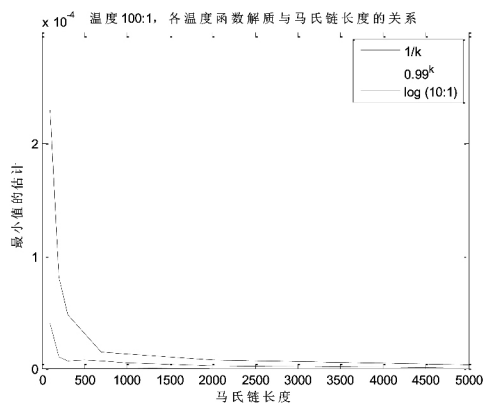


图9 解质与马氏链长度的关系

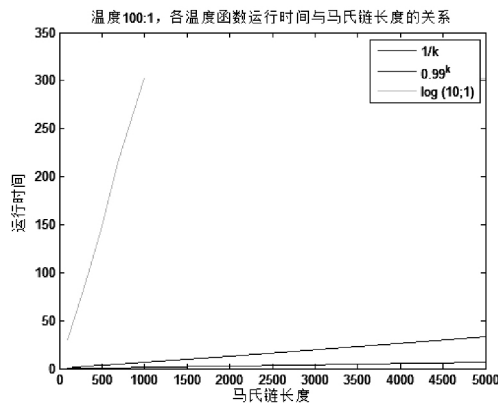


图10 运行时间与马氏链长度的关系

4.4 结论

(i) 降温函数的比较中,初温较低时解质(“>”表示“优于”)

$$T_k = \frac{T_s}{\ln(k)} > T_k = T_s \times 0.99^k > T_k = \frac{T_s}{k},$$

初温较高时 $T_k = \frac{T_s}{\ln(k)}$ 解质最好, $T_k = T_s \times 0.99^k$, $T_k = \frac{T_s}{k}$ 解质差不多,运行时间

$$T_k = \frac{T_s}{\ln(k)} > T_k = T_s \times 0.99^k > T_k = \frac{T_s}{k}.$$

若想要高效的得到精确度较高的最优解估计值,可以选择 $T_k = \frac{T_s}{\ln(k)}$ 并且设置较小的初末温比,或者选择 $T_k = T_s \times 0.99^k$ 或 $T_k = \frac{T_s}{k}$ 并且设置相对较高的初末温比.还可以选择比 $T_k = T_s \times 0.99^k$ 变化速度更慢但又快于 $T_k = \frac{T_s}{\ln(k)}$ 的温度函数,控制初末温后也可以在较短时间内得到解质较好的估计值.但降温函数 $T_k = \frac{T_s}{\ln(k)}$ 在应用时对初末温的选择需要对具体问题进行分析.

(ii) 初末温的比较中,固定马氏链长度和初温,末温越高,解质越差,运行时间越短,解质变差的速度大体呈现出先快后慢的趋势;固定末温,初温越高,解质越好,运行时间越长,解质的变化情况是先慢后快.在实际应用中,初温不需要选择过高;末温选择过高或过低都会影响解质.

(iii) 马氏链长度的比较中,受马氏链长度的影响由大到小分别是 $T_k = \frac{T_s}{k}$, $T_k = T_s \times 0.99^k$ 和 $T_k = \frac{T_s}{\ln(k)}$.运行时间从长到短分别是 $T_k = \frac{T_s}{\ln(k)}$, $T_k = T_s \times 0.99^k$ 和 $T_k = \frac{T_s}{k}$.解质随马氏链长度增加都有所提高,但在马氏链长度大于 1000 时解质不会有较大的变化了,因此马氏链长度依问题复杂程度进行适度选择即可,无需过长.

[参 考 文 献]

- [1] 刘来福,黄海洋,等. 数学建模实验[M]. 北京:北京师范大学出版社,2014.
- [2] Alfonsas MISEVICIUS. A Modified Simulated Annealing Algorithm for the Quadratic Assignment Problem[J]. INFORMATICA, 2003, 14(4): 497—51.
- [3] 康立山,等. 非数值并行算法第一册:模拟退火算法[M]. 北京:科学出版社,1994:29—30;84—124.
- [4] Mahmoud H. ALREFAEI, Sigrun ANDRADO-TTIR. A Simulated Annealing Algorithm with Constant Temperature for Discrete Stochastic Optimization[J]. Management Science, 1999, 45(5): 748—764.
- [5] Hisao ISHIBUCHI, Shinta MISAKI, Hideo TANAKA. Modified simulated annealing algorithms for the flow shop sequencing problem[J]. European Journal of Operational Research, 1995, 81(2): 388—398.

- [6] 陈章潮,顾洁,孙纯军. 改进的混合模拟退火—遗传算法应用于电网规划[J]. 电力系统与自动化[J],1999,23(10):28—40.
- [7] 姚明海,王娜,赵连朋. 改进的模拟退火和遗传算法求解 TSP 问题[J]. 计算机工程与应用,2013,49(14):60—65.
- [8] 王雪梅,王义和. 模拟退火算法与遗传算法的结合[J]. 计算机学报,1997,20(4):381—384.
- [9] 刘勇,康立山,等. 非数值并行算法第二册:遗传算法[M]. 北京:科学出版社,1995:75—85.
- [10] 杨若黎,顾基发. 一种高效的模拟退火全局优化算法[J]. 系统工程理论与实践,1997,5(5):29—35.
- [11] 徐雷. 一种改进的模拟退火组合优化算法[J]. 信息与控制,1990,3:1—7.

A Review of Improvement and Research on Parameters of Simulated Annealing Algorithm

LU Yu-ting, LIN Yu-you, PENG Qiao-zi, WANG Ying-zhe

(School of Mathematical Science, Beijing Normal University, Beijing 100875, China)

Abstract: We reviewed the process of simulated annealing algorithm(SAA) and made a comprehensive summary and evaluation of the existing improvement of the SAA. And then we explored the three important key parameters: cooling function, initial temperature and final temperature, Markov chain length by applying the SAA to determine the minimum value of Sobol’g function.

Key words: simulated annealing algorithm; improvement; summary; Sobol’g function; parameters