项目设计三：设计和实现求解 TSP 问题的模拟退火法

姓名：项津旭　学号：3150104848 班级：数学与应用数学 1501 班级序号：38

摘要：本文使用模拟退火法实现求解 TSP 问题，并用已知 TSP 问题最优值的路径测算不同数量的节点下，最终结果误差在1%以内的退火次数，从而拟合获得系数与节点数的函数关系。进而运用上面的系数计算 TSP 问题的解、时间复杂度，并求出在[0,1]x[0,1]内随机投点数与其 TSP 问题的路程的函数关系式。

一．构建模拟退火法求解 TSP 问题

TSP 问题的模拟退火法算法如下：

步骤 1：随机产生[0,1]x[0,1]内的N个点和一条起始路径P，给定系数L_k和概率p_{back}。

步骤 2：多次随机设置两条路径并计算两条路径长度的差值D_l，并计算D_l的最大值D_{max}。通过此差值得到初始温度$T_0 > 0$。由T_0、给定的退火公式及后文计算出的退火次数函数得到终止温度$T_{min} > 0$。此时$i = 0$。

步骤 3：如果当前温度$T_i > T_{min}$，则继续，否则转至步骤 8。

步骤 4：使用$inverse$、$insert$、$swap$三种方式改变路径P，得到新的三个路径$P_j, j = 1,2,3$,并计算路径改变的差值$D_j, j = 1,2,3$。选择差值最小(D_t)的路径(P_t)进行下一步计算。

步骤 5，如果$D_t < 0$，则$P = P_t$，否则以概率e^{-D_t/T_i}接受 $P = P_t$。

步骤 6，以p_{back}的概率使得当前温度T_i升高一倍，并转至步骤 4，以此提高改变路径的随机性。

步骤 7，实现降温过程将T_i改变为T_{i+1}，$i = i + 1$，并转至步骤 3。

步骤 8，获取当前路径P，根据路径画出图像，计算 TSP 问题的距离。

二．三种改变路径方式$inverse$、$insert$、$swap$的详解

假设 TSP 问题中的节点数为 n，其路径为$(p_0, p_1, \dots p_{n-2}, p_{n-1})$，任取两不相等数字$s, t$满足$s, t < n - 1$，旋转路径保证$s < t$，新的路径与数字仍记为$(p_0, p_1, \dots p_{n-2}, p_{n-1})$与$s, t$。

1. $inverse$：在当前路径的基础上，使得p_s与p_t之间（包含$p_\mathrm{s}, p_\mathrm{t}$）的路径顺序全部逆序排列，记作$inverse(\mathrm{path,s,t})$。
2. $insert$：在当前路径的基础上，将p_s插入到p_t后一个节点的位置上，记作$insert(\mathrm{path,s,t})$。
3. $swap$：在当前路径的基础上，仅交换p_s与p_t两节点的位置，记作$swap(\mathrm{path,s,t})$。[1]

上述三种方式均可以由一步或两步$inverse$得到，相较于仅使用一种方法，三种并用可以提高 TSP 问题收敛速度。

当$s = t - 1$时，$inverse(\mathrm{path,s,t}) = insert(\mathrm{path,s,t}) = swap(\mathrm{path,s,t})$

其他情况下，$insert(\mathrm{path,s,t}) = inverse(\mathrm{path,s,t+1}) \circ inverse(\mathrm{path,s,t})$

$$swap(\mathrm{path,s,t}) = inverse(\mathrm{path,s,t}) \circ inverse(\mathrm{path,s+1,t-1})$$

其中$\mathrm{A} \circ \mathrm{B}$表示进行A变换后再进行B变换。

三．系数的选取

在模拟退火法的使用中，主要依赖于两个循环嵌套进行多次计算。第一个循环是在不同温度下的计算，这里的计算次数主要依赖于参数T_0, T_{min}和降温公式。第二个循环是在同一温度下的计算，其计算次数主要依赖于L_k。

T_0的选择是模拟退火中最重要的地方，因为它关系到上文步骤 5 中接受路程变大这个改变的概率。如果T_0过大，则接受改变的概率近似于1，这会导致模型中的随机性过大，前若干次的计算无法得到良好结果。如果T_0过小，会导致接受概率近似于 0，这会导致模型的随机性过小，无法跳出某些局部最优值。考虑到接受概率为e^{-D_j/T_i}，T_0的选取应与距离差D有关。所以本文随机选取一百次两条路径并计算两条路径长度的差值$D_l, l = 0, \dots, 99$，并计算D_l的最大值D_{max}。令$T_0 = 2 * D_{max}$，这样初始的接受概率会接近$e^{-1/2} \approx 60\%$。

降温公式的选择有很多种，但是它们达到的效果都是类似的，故本文使用$T_i = T_0/(1 + i)$作为降温公式进行计算。下一节将讲述在此降温公式下降温次数的选择方式，在给定降温次数M的情况下，$T_{min} = T_0/M$。

多个变量的计算是复杂的，如果不断细化降温温差，使得降温的循环中的次数变得越来越多，这样就类似于在相近的温度下进行多次计算，从而达到类似于改变L_k的效果。所以可以简化得将L_k的取值设定为当前温度T_i相关，在后面的计算中全部使用$L_k = L_0 ln(1 + 1/T_i)$，本文中取$L_0 = 15$。

四．降温次数函数的拟合

经过多次尝试可知，若降温次数为常数，则在随机选取的点的个数N较大时，可以明显看出计算得到的 TSP 问题的路程图交叉次数过多，其并非较优解。所以降温次数函数应为一个随N增大而增大的函数。计算此函数的步骤如

下：

步骤 1：选定节点数N_i，生成已知 TSP 最优解的节点图，随机设置初始路径。最优解的值记作C_{min}。

步骤 2：选取足够小的降温次数R_0，使得在此R_0的计算下无法得到 TSP 问题的较优解。

步骤 3：计算使用R_j时得到的 TSP 问题的路程值，多次计算取平均得到C_j。

步骤 4：若$C_j < C_{min} * (1 + err)$，则将此$R_j$记作$M_i$，并继续至步骤 5。否则，将$R_j$增大为$R_{j+1}$后转至步骤 3。这里的$err$是可接受误差，本文在计算中使用$err = 1\%$。

步骤 5：选择不同的N_i，并将坐标（N_i, M_i）绘制到平面上，拟合计算出M关于N的函数关系式。

按照上面的步骤，得到的降温次数近似函数关系式为：$M = 1.5N * (lnN)^2$，拟合图像见图 1。

五．时间复杂度分析

使用模拟退火法求解 TSP 问题的时间复杂度主要集中在上文提到的两个循环处，第一层循环的计算次数为$M = 1.5N * (lnN)^2$，其时间复杂度为$\mathrm{O}(N(lnN)^2)$。第二个循环中的计算次数为$L_k = L_0 ln(1 + 1/T_i)$，由于$T_i = T_0/(1 + i), \mathrm{i} = 0, \ldots \mathrm{M}$，故其时间复杂度为$\mathrm{O}(L_0 lnN)$。

在内层中，计算经过$inverse$、$insert$、$swap$改变后的路径与原路径的差值并不需要将全部的路径计算出来，对于原路径与改变后的路径仅有 4 段路程改变，设改变的点为s, t：

$$dist\ =\ \mathrm{PLength(points, path, s-1, s+1)}\ +\ \mathrm{PLength(points, path, t-1, t+1)}$$

$$dist1\ =\ \mathrm{PLength(points, path1, s-1, s+1)}\ +\ \mathrm{PLength(points, path1, t-1, t+1)}$$

$$dist2\ =\ \mathrm{PLength(points, path2, s-1, s)}\ +\ \mathrm{PLength(points, path2, t-1, t+1)}\ +\ \mathrm{PLength(points, path, t-1, t)}$$

$$dist3\ =\ \mathrm{PLength(points, path3, s-1, s+1)}\ +\ \mathrm{PLength(points, path3, t-1, t+1)}$$

其中$\mathrm{PLength(points, path, a, b)}$是计算点列$\mathrm{points}$按照路程$\mathrm{path}$从节点$a$到节点$b$中的路程长度。在<—>步骤 4 中使用的$D_{inverse} = dist1 - dist$，$D_{insert} = dist2 - dist$，$D_{swap} = dist3 - dist$。

所以内层的时间复杂度为$O(1)$，综上所述，本文使用的算法所用的时间复杂度为$\mathrm{O}(L_0 N(lnN)^3)$，考虑到本文中L_0为常数，故时间复杂度为$\mathrm{O}(N(lnN)^3)$。后经过图 2 验算，与分析结果一致。

六．补充内容

图 3 展现了在[0,1]x[0,1]内随机产生的N个节点 TSP 问题的最优值，这是一条随N增大的曲线，一阶导数为正、二阶导数为负。使用lnN拟合后，lnN函数从左至右穿插原函数而过，说明原函数增长速率大于lnN而小于N。

图 4 展现了200个随机点的 TSP 问题路程图，在增大err而减小M的过程中，计算速度会显著性提高。事实上，由图 5 可知，在200个点的情况下，进行当前状况约$1/3$的计算，即使用约$1/3$的时间，即可得到非常好的 TSP 问题的解。而通过计算，在50个点的时候，这个比值是大于$1/3$的，在500个点的时候，这个比值是小于$1/3$。这说明了本文中算法的时间复杂度并非最优，其仍有一定提升范围。同时，这也说明了本算法没有节点数上限，但是会在计算更多节点时花费过多的时间。图 6 展现了通过本算法计算出的2000个随机点的 TSP 路径。

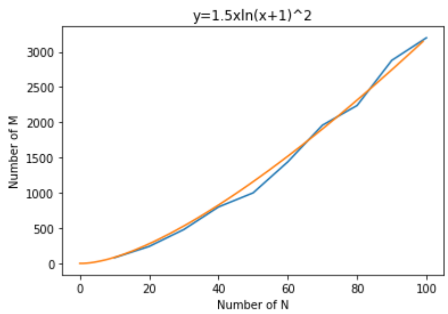

图 1

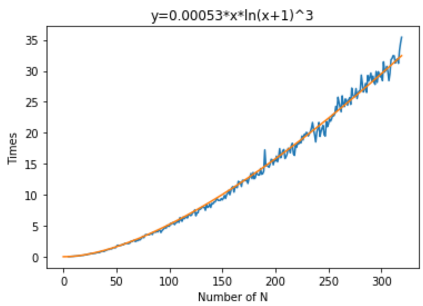

图 2

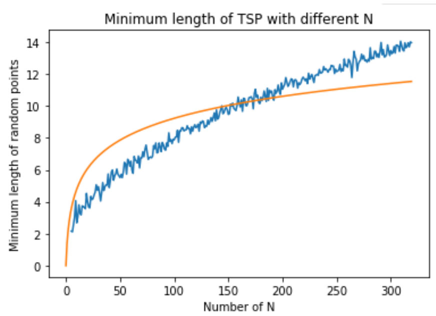

图 3

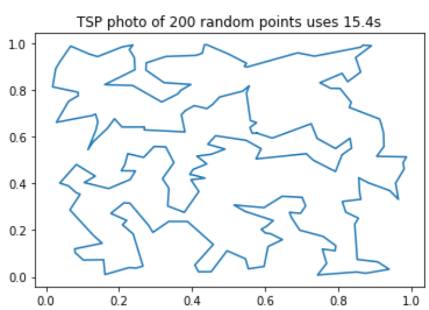

图 4

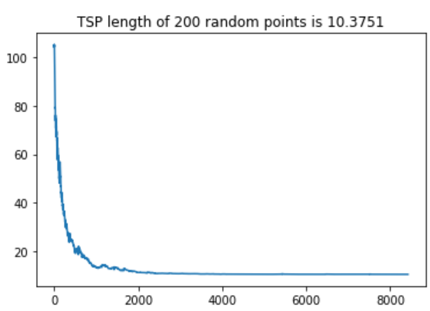

图 5

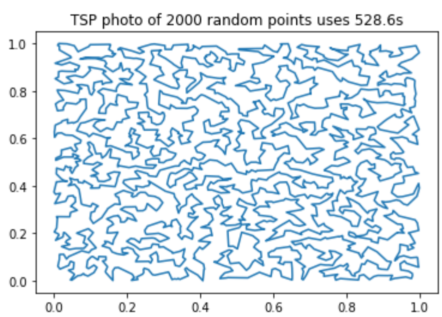

图 6

致谢

我在完成本次 project 时，参考了 CSDN 网上下面几篇文章的内容：

1. 模拟退火算法与其 python 实现（一）——TSP 问题 作者：WFRainn
2. 模拟退火算法与其 python 实现（二）——TSP 问题 作者：WFRainn

从中我学习到了<一>步骤 6 中p_{back}、<三>中T_0的选取，并从中获取了部分书写代码的思路。十分感谢作者无私的分享。

参考文献

[1] 求解 TSP 问题的带记忆的模拟退火算法及其并行设计 万星，王长缨 福建电脑 2018 年 01 期 P46-P59 福建省计算中心，福建省计算机学会