第25卷 第2期
1999年3月

自 动 化 学 报
ACTA AUTOMATICA SINICA

Vol. 25, No. 2
Mar., 1999

研究简报

TSP 问题分层求解算法的复杂度研究[1)]

卢 欣 李衍达

(清华大学自动化系 北京 100084)

关键词 TSP,局部搜索算法,动态聚类,计算复杂度.

COMPLEXITY ANALYSIS OF THE MULTI-LAYERED CLUSTERING ALGORITHMS IN TSP

LU Xin LI Yanda

(*Department of Automation, Tsinghua University, Beijing* 100084)

Key words Traveling salesman problem(TSP), local search algorithm, multi-layered clustering, complexity of computation.

1 TSP 问题及其区域划分求解算法

TSP(traveling salesman problem)问题已被证明是 NP 问题,用现有的优化算法,如分支定界、动态规划等求最优解,需要问题规模的指数阶时间[1,2]. 在问题规模增大时,往往由于计算时间的限制而丧失可行性,只能用一定的策略对解空间进行启发式搜索,期望在合理的时间内得到一个满意解. 比较算法时主要以解的质量以及运算时间为标准[3].

在 TSP 问题中,问题与解的结构完全由城市之间的相对距离矩阵 $D_{ij}(i,j=1,\cdots,N)$ 来决定. 因此,利用距离矩阵的信息,将城市聚类为一些小的区域,这样就可以将问题分解,以提高算法的效率.

在问题规模增大时,可以通过多层区域划分,使每一个层次的问题规模限制在一定的范围内,降低问题的复杂度. 此外,采用并行技术实现 TSP 问题分层优化,其复杂度以及导致复杂度的原因也会产生根本性的变化,从而导致算法研究指向与以往不同的方向.

2 动态聚类分层求解大规模 TSP 问题及性能分析

算法可简述如下:

1) 若城市数小于预先给定的数目,则直接调用局部搜索算法求解.

1)国家自然科学基金资助项目.

收稿日期 1997-04-09 收修改稿日期 1998-04-28

2) 将城市划分为 P 个区域,求 P 个区域的重心,调用局部搜索算法,求一条遍历 P 个重心的最优路径.

3) 对于每一个区域,以这个区域在区域遍历路径上的相邻点为固定点,递归调用动态聚类分层算法.

对于 N 点规模的 TSP 问题,一般估计局部搜索算法的计算复杂度不低于 $O(N^2)$,具体估计还有许多争论[1,2,3].

一次 N 个样本的 C 均值聚类划分,算法复杂度估计为 $O(N)$;而在 P 个区域间及 P 个区域内分别用局部搜索算法寻优,相当于解一个 P 规模和 P 个(N/P)规模的 TSP 问题.所以,以局部搜索算法的计算复杂度下界来分析,不难证明,进行一次划分,将 N 城市 TSP 划分为 P 个区域时,假设城市均匀分布,每一个区域城市的数目大致相等,则其计算复杂度估计不低于

$$O(N)+O(P^2)+P^*O[(N/P)^2]. \tag{1}$$

当以固定的区域数目 P 进行 k 次划分时,计算复杂度估计不低于

$$O(kN+P^2+P^3+\cdots+P^{k+1}+N^2/P^k). \tag{2}$$

令
$$f(N,P,k)=kN+P^2+P^3+\cdots+P^{k+1}+N^2/P^k, \tag{3}$$

可以求出最佳划分层数 k

$$\frac{\partial f(N,P,k)}{\partial k}=N+\left(\frac{P^{k+2}}{P-1}-\frac{N^2}{P_k}\right)\cdot \ln P=0,$$

$$k=\log_P\left[\left[\frac{\sqrt{\left(\frac{1}{\ln P}\right)^2+4\frac{P^2}{P-1}}-\frac{1}{\ln P}}{2\frac{P^2}{P-1}}\right]N\right]\approx$$

$$\log_P N+\log_P\left[\frac{\sqrt{2P}-1/\ln P}{2P}\right]\approx \log_P N-1/2. \tag{4}$$

当 $\log_P N$ 为整数时,可取 $k=\log_P N-1$,这时

$$P^k=P^{\log_P N-1}=N/P, \tag{5}$$

$$f(N,P,k)=kN+\frac{P^{k+2}-P^2}{P-1}+\frac{N^2}{P^k}=kN+\frac{PN-P^2}{P-1}+PN. \tag{6}$$

当 $N\to\infty$ 时
$$f(N,P,k)\approx(k+1+P)N, \tag{7}$$

即 k 层划分后整体算法的计算复杂度估计为

$$O[f(N,P,k)]=O(N\ln N). \tag{8}$$

与局部搜索算法相比,其计算复杂度降低了 $O(N/\ln N)$数量级.问题规模越大,算法效率改进越明显.

3 仿真实例

用一台 Pentium 166 PC 机进行仿真实验.所需计算时间在不同的实现环境下可能会有较大差别,但几种算法间的相对比较还是有意义的.

利用两类问题对动态聚类分层优化算法进行了仿真研究.一个是 CHN144实例,另一个是随机生成,均匀分布的 TSP 问题.每次分10个区域,每个区域最多20个城市,每个层

次上分别执行10次(动态聚类10)和100次(动态聚类100)局部搜索算法.同时也执行了局部搜索算法与之相比较,每次只搜索一遍.

表1是对于 CHN144问题,每种算法执行100次,取其平均值与最优的结果作对比.

表1　CHN144问题解的质量及时间比较

算法	最优			平均		
	时间(秒)	结果(km)	解质(%)	时间(秒)	结果(km)	解质(%)
动态聚类100	40.110	32 013	94.9	46.901	32 835	92.5
动态聚类10	2.473	32 412	93.7	3.351	33 337	91.1
局部搜索	40.934	32 089	94.7	52.022	33 113	91.7

从表1中可以看出,三种算法所获得的解质量基本相当,最主要的差别在于运算时间.由于动态聚类100是在每个层次上执行100次搜索,而局部搜索算法只执行了一次,所以,动态聚类方法的速度是局部搜索算法的大约100倍,速度提高的数量级与分析的基本吻合.

表2是对随机生成,均匀分布样本的 TSP 问题仿真结果.这里只是比较动态聚类分层划分方法在各种问题规模下的计算速度,以验证对于其计算复杂度的估计.下面比较中用的是动态聚类10的方法,其中144规模的问题就是表1中的 CHN144问题,其它每个规模用十个随机生成的实例,每个实例寻优100次,随问题规模的增长,寻优速度的增长与前面的分析相符合.

表2　不同问题规模下动态聚类方法的计算时间比较

问题规模	最优(秒)	平均(秒)	最差(秒)
100	1.264	1.527	1.648
144	2.473	3.351	3.901
1 000	15.549	16.318	17.640
10 000	230.752	237.812	252.589

4　算法的并行实现及其复杂度研究

这里,给出对于大规模 TSP 问题的动态聚类分层优化算法的一种主从式并行实现形式,并初步分析其计算复杂度.

动态聚类分层优化算法中,高层完成了区域划分后,低层中每个区域的路径寻优实际上只利用了相邻区域的重心坐标,而这一信息是在高层区域划分过程中获得的.所以,低层每个区域之间的寻优是相互无关的,可以很容易地实现并行处理.

算法的并行实现是一棵 P 叉树的形式.根节点进行 N 个城市的区域划分,并将 P 个区域分给第二层的 P 个节点同步进行各区域的寻优.这 P 个节点进一步进行区域划分,并将任务向下分解.在算法完成时,遍历这棵 P 叉的所有叶节点,就可以建立最终的结果路径.按照上文的方法进行并行算法的复杂度分析,可以看出,将 N 城市问题分为 $k=\log_P N-1$ 层时,其计算复杂度可以表示为

$$O(N+P^2)+O(N/P+P^2)+\cdots+O(N/P^{k-1}+P^2)+O(P^2)=$$
$$O(P^2\log_P N)+O(N). \tag{9}$$

现在计算复杂度主要由动态聚类算法决定,而局部搜索算法的影响相对变成了次要问题.

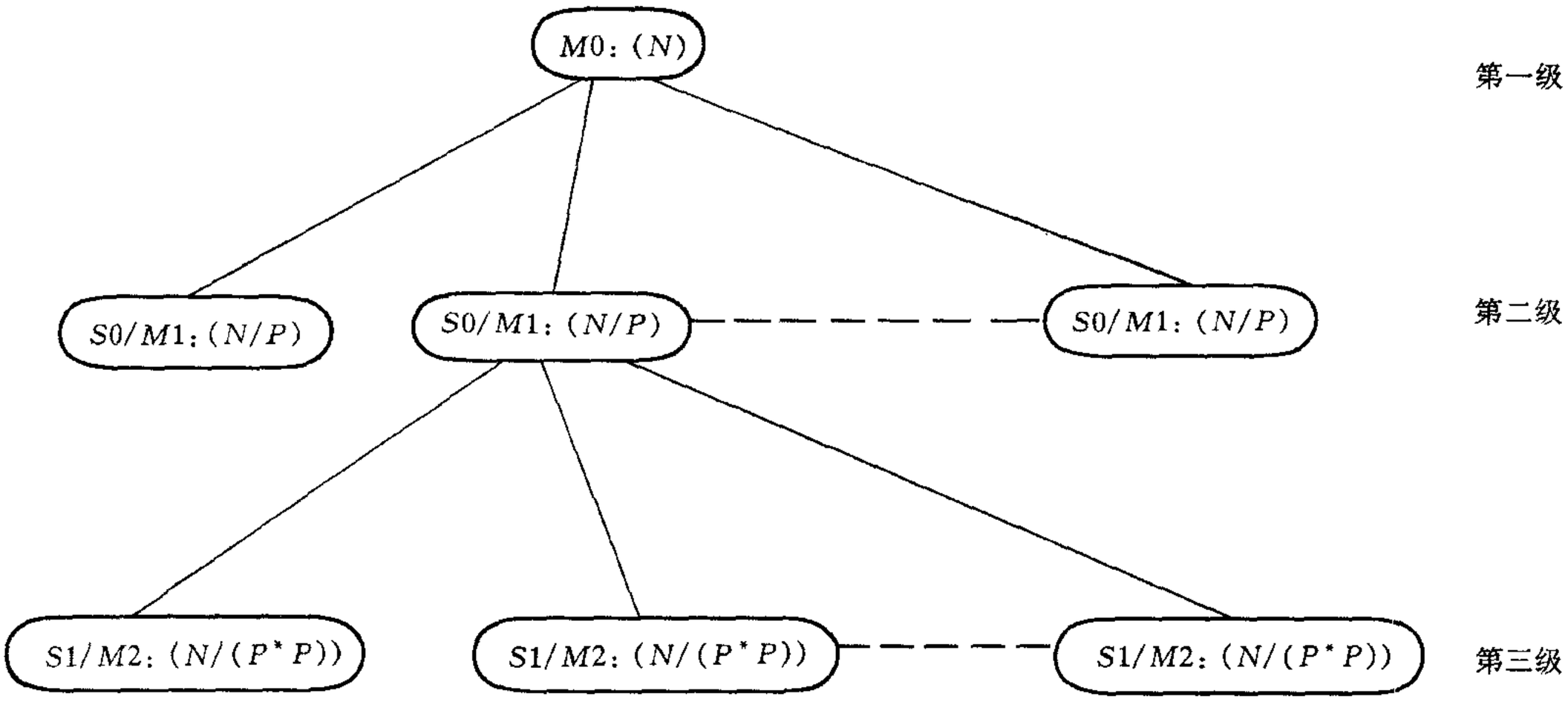

图1 动态聚类分层优化算法的并行实现结构图

这也促使研究的重点转向了动态聚类算法.寻找更高效的聚类算法或算法的并行实现,以进一步降低算法复杂度,将成为下一步研究的方向.

参 考 文 献

1 Johnson D S, Papadimitriou C H, Yannakakis M. How easy is local search? *J. Comput. Sys. Sci.* 1988,37(1):79—100

2 Papadimitriou C H, Yannakakis M. Optimization, Approximation and complexity classes. *J. Comput. Sys. Sci.* 1991,43(3):425—440

3 Ausiello G, Protasi M. Local search, reducibility and approximability of NP-optimization problem. *Information Processing Letters*,1995,54(2):73—79

卢 欣 1996年毕业于清华大学,获学士学位.现为清华大学自动化系博士生.研究兴趣为生物信息学、计算智能技术、组合优化及算法等.

李衍达 清华大学自动化系教授、信息学院院长、中国科学院院士.长期从事生物信息学、信号处理和计算智能理论及应用的研究.