第6卷第7期 软 件 学 报 Vol. 6, No. 7
1995年7月 JOURNAL OF SOFTWARE Jul. 1995

货郎担问题的几何解法*

周培德

(北京理工大学计算机系,北京 100081)

摘要 本文提出货郎担问题的一种新的求解方法,即几何解法.它的时间复杂性为:求距离运算次数为$O(nm)$,比较次数为$O(\max(nm, n\log n))$,求夹角次数为$O(\frac{n^2}{m})$,其中n为点集中点的数目,m为点集的凸包顶点数.

关键词 几何算法,算法复杂性,货郎担问题*.

货郎担问题是一个NP完全问题,现已有许多近似方法求解它.本文提出的算法主要是依据几何中的求点线距离,两点间的距离,距离值的比较、求和,求两线的夹角,点集的凸包等基本几何运算和算法完成的.该算法摆脱了分支限界法的思想,也不同于动态规划的分级求解和最小权匹配算法(MM算法)的思想,而是利用几何方法求解.基本作法是,先求出点集的凸包,以点集中的点(除凸包顶点外)与凸包各边界的距离的最小值为标准划分点集中的点为不同的类,然后依次在各类中不断以三角形两斜边代替底边的方法把三角形顶点及两斜边分别加入点集及子路径集.划分新的点集类与三角形中两斜边代替底边的工作继续下去,直至所有的新点集类为空.此时便得到一条完整的路径.该算法易于修改为并行算法.

1 算法描述

输入:n个给定点的点集B(包括n个点的坐标$P_i(x_i, y_i)$及各点之间的距离矩阵$D=(d_{ij})$)

输出:经过n个点的一条旅行路线T及此路线的长度

步1:求n个点的凸包.设$A=\{P_1,\cdots,P_m\}$是凸包顶点集,$C=\{\overrightarrow{P_1P_2},\overrightarrow{P_2P_3},\cdots,\overrightarrow{P_mP_1}\}$是凸包的边界集.

步2:除凸包顶点外的$n-m$个点分成m个类$T_{i,i+1}(i=\overline{1,m}, T_{m,m+1}=T_{m,1})$

2.1:计算点$P_k\in B-A$至凸包各边界的距离$d^k_{i,i+1}(k=\overline{1,n-m}, i=\overline{1,m})$

2.2: if $D^k_{j,j+1}=\min_i(d^k_{i,i+1})$ then $T_{j,j+1}\leftarrow P_k$

2.3: if $T_{j,j+1}=\varnothing$ then $T\leftarrow\overrightarrow{P_jP_{j+1}}$

2.4: while $d^k_{i-1,i}=d^k_{i,i+1}$ do

* 本文1993-10-23收到,1994-03-15定稿

作者周培德,1941年生,副教授,主要研究领域为算法设计与分析,计算理论.

本文通讯联系人:周培德,北京100081,北京理工大学计算机系

if　$(P_{t_1},P_{t_2}\in T_{i,i+1})\wedge(d(P_k,P_{t_1})<d(P_k,P_l)\wedge d(P_k,P_{t_2})<d(P_k,P_l)$,
$(l=\overline{1,n-m-3})$)　then　$T_{i,i+1}\leftarrow P_k$

2.5: while　$d^u_{i-1,i}\approx d^u_{i,i+1}$(点团 $P_u\in P_{\overline{1,n-m}}$)　do
if　$\Sigma_u d^u_{i-1,i}<\Sigma_u d^u_{i,i+1}$　then　$T_{i-1,i}\leftarrow$点团 P_u
else　$T_{i,i+1}\leftarrow P_u$

步3: $i\leftarrow 1$, $j\leftarrow 1$

步4: 处理类 $T_{i,i+1}$中的点,设 $T_{i,i+1}$中有 V_i 个点.

4.1: if　$P_e\in T_{i,i+1}\wedge d^e_{i,i+1}<T_{i,i+1}$中其它点到$\overrightarrow{P_iP_{i+1}}$的距离$\wedge d(P_i,P_e)\approx d(P_e,P_{i+1})$
then　用$\overrightarrow{P_iP_e}$与$\overrightarrow{P_eP_{i+1}}$代替边$\overrightarrow{P_iP_{i+1}}$, $T_{i,i+1}\leftarrow T_{i,i+1}-P_e$
else if　$d(P_i,P_e)\not\approx d(P_e,P_{i+1})\wedge P_{e_1}\in T_{i,i+1}\wedge d^{e_1}_{i,i+1}\approx d^e_{i,i+1}$
then　比较 $d(P_i,P_e)+d(P_e,P_{i+1})$与 $d(p_i,P_{e_1})+d(P_{e_1},P_{i+1})$
if　$d(P_i,P_e)+d(P_e,P_{i+1})>d(p_i,P_{e_1})+d(P_{e_1},P_{i+1})$
then　用$\overrightarrow{P_iP_{e_1}}$与$\overrightarrow{P_{e_1}P_{i+1}}$代替$\overrightarrow{P_iP_{i+1}}$, $T_{i,i+1}\leftarrow T_{i,i+1}-P_{e_1}$
else　用$\overrightarrow{P_iP_e}$与$\overrightarrow{P_eP_{i+1}}$代替$\overrightarrow{P_iP_{i+1}}$, $T_{i,i+1}\leftarrow T_{i,i+1}-P_e$

4.2: $P_h\in T_{i,i+1}$时,以$\angle P_hP_iP_e$, $\angle P_hP_eP_i$(或$\angle P_hP_{i+1}P_e$, $\angle P_hP_eP_{i+1}$)是否为锐角将类 $T_{i,i+1}$中的点分成两个子类 $T'_{i,i+1}$与 $T'_{i+1,i+2}$. 然后按4.1的方法在下一轮($j+1$)中处理各子类中的点. 某些点可能不能划入两个子类中的任何一类,这种点待下一轮分类时考虑划入哪个子类. 如果 j 循环临近终止时仍有点 P_t 不能划入任何一个子类,则用穷举法求该点与 $T_{i,i+1}$中已找到的子路径各弧端点 P_w 的距离 $d(P_t,P_w)$, $w=i,\cdots,j$, $\overrightarrow{P_iP_{i+1}},\cdots,\overrightarrow{P_{j-1}P_j}$是 $T_{i,i+1}$中已找到的子路径.

if　$d(P_u,P_t)+d(P_t,P_{u+1})-d(P_u,P_{u+1})=\min_w(d(P_t,P_w)+d(P_t,P_{w+1})$
$-d(P_w,P_{w+1}))$
then　用弧$\overrightarrow{P_uP_t}$与$\overrightarrow{P_tP_{u+1}}$代替弧$\overrightarrow{P_uP_{u+1}}$

4.3: $i\leftarrow i+1$　goto 步4,直至 $i=2^{j-1}m+1$

4.4: $j\leftarrow j+1$, $i\leftarrow 1$　goto 步4,直至所有 $T_{i,i+1}$为空集(此时便求得 $T_{i,i+1}$中的一条路径).

步5: m 个类 $T_{i,i+1}$中的 m 条子路径在凸包顶点处连接,便组成一条完整路径 T.

2　算法的正确性与复杂性

引理1. 给定等腰$\triangle ABC$, AD 为底边中垂线,顶点 A 沿平行于底边 BC 的直线 l 移动到 A_1与 A_2处,则有$|\overline{BA_1}|+|\overline{A_1C}|<|\overline{BA_2}|+|\overline{A_2C}|$.

证明:以 D 点为坐标原点,底边为 X 坐标轴,$\overrightarrow{DA}$为 Y 轴,各点坐标如图1所示.

$$|\overline{BA_1}|=\sqrt{(x_1+a)^2+(y_0-0)^2}=\sqrt{(x_1+a)^2+y_0^2}$$

$$|\overline{A_1C}|=\sqrt{(x_1-a)^2+y_0^2}$$

$$|\overline{BA_2}|=\sqrt{(x_2+a)^2+y_0^2}$$

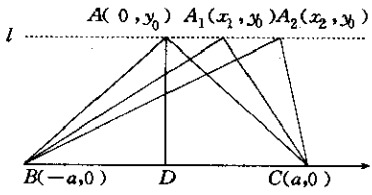

图1

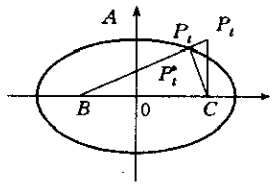

图2

$$|\overline{A_2C}|=\sqrt{(x_2-a)^2+y_0^2}$$

$$|\overline{BA_1}|+|\overline{A_1C}|=\sqrt{(x_1+a)^2+y_0^2}+\sqrt{(x_1-a)^2+y_0^2}$$

$$|\overline{BA_2}|+|\overline{A_2C}|=\sqrt{(x_2+a)^2+y_0^2}+\sqrt{(x_2-a)^2+y_0^2}$$

由于 $x_2>x_1$，所以有

$$|\overline{BA_1}|+|\overline{A_1C}|<|\overline{BA_2}|+|\overline{A_2C}|$$ 证毕

特别地，当$\angle A_2CB>\frac{\pi}{2}$时，两斜边长度之和增长速度加快，而$|\overline{BA}|+|\overline{AC}|$最短. 因此算法的步4.2要求$\angle P_hP_iP_e$和$\angle P_hP_eP_i$为锐角，步4.1要求$d(P_i,P_e)\approx d(P_e,P_{i+1})$.

引理2. 当点P_i分别位于以$\triangle ABC$的顶点B、C为焦点的椭圆内、椭圆上和椭圆外时(见图2)，该点与B、C的距离之和为单调递增.

证明：由椭圆的定义及简单的几何知识即知. 证毕

由引理2可确定步4.1中点P_e、P_{e_1}的选取方法：如果P_e与P_{e_1}都在椭圆上，则可任意选取；如果P_e在椭圆上，而P_{e_1}在椭圆内(外)，则先取$P_{e_1}(P_e)$. 这样可保证步4.1中的不等式成立.

引理3. 设P_t是折线$P_iP_{i+1}\cdots P_{j-1}P_j$外任一点，当下式成立时，新折线(原折线加入点$P_t$后所得折线)的长度增加最少.

$$|\overrightarrow{P_wP_t}|+|\overrightarrow{P_tP_{w+1}}|=\min_w(d(P_t,P_w)+d(P_t,P_{w+1})-d(P_w,P_{w+1})),w=i,\cdots,j \quad (1)$$

证明：(1)式表明w遍历i与j之间所有正整数时，要求距离的代数和(下式)最小，即点P_t插入折线后使折线的长度增加最少.

$$d(P_t,P_w)+d(P_t,P_{w+1})-d(P_w,P_{w+1})$$ 证毕

定理1. 算法正确地求出了货郎担问题的较优解.

证明：算法首先把$n-m$个点分成m个类，在此分类过程中考虑了每个点应归于哪一类，没有丢失点的可能性发生. 然后，算法轮流考查每个类中的所有的点，直至最后一个点进入子路径，而且每个点只考查一次，即已进入子路径的点不可能再次进入子路径. 因此，这样求得的解满足货郎担问题解的要求，即算法正确地求解了货郎担问题.

算法的步2.2，是以$\min_i(d^k_{i,i+1})$为标准划分点P_k应归于哪一类，这是以贪心法的思想划分点类，在大多数情况下，为后继步骤能求得较优解奠定了基础.

算法的第4.1步执行以后，由引理1、2知，点P_e加入子路径后，子路径的长度增加最少. 步4.2中要求$\angle P_hP_iP_e$，$\angle P_hP_eP_i$为锐角，由引理1知，子路径增加新点后长度增加较少.

算法的步4.2中最后处理点P_t时，由引理3知，子路径长度增加最少.

由于该算法每次加入新点至子路径时，都是以子路径长度增长最少为目标，因此算法求

得的解是最优的或较优的.

定理2. 设平面上 n 个点呈均匀分布，则算法的时间复杂性为：求距离运算次数为 $O(n\cdot m)$，求夹角次数为 $O(n^2/m)$，比较次数为 $O(n\cdot m)$，其中 m 为点集凸包的顶点数.

证明：算法的步1要求 $O(n\log n)$ 次比较. 步2.1要求 $(n-m)m$ 次距离. 步2.2要进行 $(m-1)(n-m)$ 次比较. 步2.3用常数时间. 步2.4至多要求 $(n-m-1)$ 次距离，$(n-m-2)$ 次比较便可判定 P_k 是否划入 $T_{i,i+1}$. 步2.5至多要求 $2(n-m-1)$ 次加法，m 次比较就可确定点团 P_u 应划入哪一类.

步2总共至多要求 $(n-m)(m+1)-1$ 次求距离运算，$((m-1)(n-m)+n-2)$ 次比较和 $2(n-m-1)$ 次加法.

步4.1(i 循环)耗费 $\sum_{i=1}^{m}(U_i-1)$ 次比较，即 $n-2m$ 次比较，可求出 $T_{i,i+1}$ 中的点 P_e，然后通过 $2m$ 次求距离和 m 次比较可以确定 P_e 是否进入子路径点集.

步4.2通过 $(n-2m)2^{j+1}$ 次求夹角运算可以完成重新分类点的工作.

$j\leftarrow j+1$ 之后，步4.1要求 $n-m2^{j-1}$ 次点线距离运算，$n/(m2^{j-1})-2$ 次比较，$2(n-m2^{j-1})$ 次点点距离运算及 $n-m2^{j-1}$ 次比较可以确定 P_e 是否进入子路径点集. 步4.2要求 $(n-m2^{j-1})2^j$ 次夹角运算.

步4.2中处理点 P_t，设 $T_{i,i+1}$ 中已找到的子路径有 V_j 条弧，则需要 $3V_j$ 次求距离运算，(V_j-1) 次比较和 $2V_j$ 次加减法可以完成 P_t 插入子路径的工作.

j 循环终止时，$j=\left\lceil\log_2\left(\frac{n-m}{m}\right)\right\rceil\approx\left\lceil\log_2\left(\frac{n}{m}\right)\right\rceil$，所需比较次数为

$$
\begin{aligned}
&(n-2m)+m+\sum_{j=2}^{\log_2(\frac{n}{m})}\left[(n-m2^{j-1})+\left(\frac{n}{m2^{j-1}}\right)-2\right]\\
&\leqslant n-m+n\log_2\left(\frac{n}{m}\right)-m\sum_{j=2}^{\log_2(\frac{n}{m})}2^{j-1}+\frac{n}{m}\left(1+\frac{1}{2}+\cdots+\frac{1}{2}\log_2\left(\frac{n}{m}\right)\right)-\sum_{j=2}^{\log_2(\frac{n}{m})}2\\
&\leqslant n-m+n\log_2\left(\frac{n}{m}\right)-2m\left(\frac{2n}{m}-1\right)+2\frac{n}{m}-2\log_2\left(\frac{n}{m}\right)\\
&=O\left(n\log_2\left(\frac{n}{m}\right)\right)
\end{aligned}
$$

所需求距离次数为

$$
2m+\sum_{j=2}^{\log_2\frac{n}{m}}3(n-m2^{j-1})=2m+3n\log_2\left(\frac{n}{m}\right)-3m\left(\frac{2n}{m}-1\right)=O\left(n\log_2\left(\frac{n}{m}\right)\right)
$$

求夹角次数为

$$
\begin{aligned}
&(n-2m)2^2+\sum_{j=2}^{\log_2\frac{n}{m}}(n-m2^{j-1})2^j\\
&=4(n-2m)+n\cdot 2^2\left(\frac{2n}{m}-1\right)-m\sum_{j=2}^{\log_2\frac{n}{m}}2^{2j-1}\\
&=4(n-2m)+4n\left(\frac{2n}{m}-1\right)-\frac{8}{3}m\left(\frac{2n}{m}-1\right)=O\left(\frac{n^2}{m}\right)
\end{aligned}
$$

因此，算法所需求距离次数为

$$(n-m)(m+1)-1+2m+3n\log_2\frac{n}{m}-3m(\frac{2n}{m}-1)=O(n\cdot m)$$

比较次数为

$$O(n\log n)+(m-1)(n-m)+n-2+n-m+n\log_2\frac{n}{m}+\frac{2n}{m}$$

$$-2(\frac{2n}{m}-1)m-2\log_2(\frac{n}{m})=O(\max(nm,n\log n))$$

求夹角次数为$O(\frac{n^2}{m})$. 证毕

3 应 用

对文献[3]提供的数据,利用该算法求得一条旅行路线:

北京$\overset{263}{—}$石家庄$\overset{394}{—}$呼和浩特$\overset{341}{—}$太原$\overset{516}{—}$西安$\overset{521}{—}$银川$\overset{346}{—}$兰州$\overset{192}{—}$西宁$\overset{1440}{—}$乌鲁木齐$\overset{1592}{—}$拉萨$\overset{1257}{—}$成都$\overset{641}{—}$昆明$\overset{434}{—}$贵阳$\overset{449}{—}$南宁$\overset{373}{—}$海口$\overset{455}{—}$广州$\overset{563}{—}$长沙$\overset{299}{—}$武汉$\overset{270}{—}$南昌$\overset{441}{—}$福州$\overset{252}{—}$台北$\overset{596}{—}$杭州$\overset{160}{—}$上海$\overset{269}{—}$南京$\overset{141}{—}$合肥$\overset{463}{—}$郑州$\overset{374}{—}$济南$\overset{271}{—}$天津$\overset{605}{—}$沈阳$\overset{281}{—}$长春$\overset{232}{—}$哈尔滨$\overset{1061}{—}$北京

总里程15 492公里,比文献[3]求得的最佳旅行路线(长度为15904公里)短412公里.

参考文献

1 周培德. 求凸包顶点的一种算法. 北京理工大学学报,1993.

2 周培德. 算法设计与分析. 机械工业出版社,1992.

3 靳蕃,范俊波,谭永东. 神经网络与神经计算机. 西南交通大学出版社,1991.

GEOMETRIC METHOD FOR SOLVING TS PROBLEM

Zhou Peide

(*Department of Computer Science, Beijing Institute of Technology, Beijing* 100081)

Abstract In this paper, a new geometric method for solving TS problem is presented. Let n be the number of points in the point set, and m be the number of vertexes in convex hulls of the point set. The time complexity of the algorithm is: the number of computation distance is $O(nm)$, the number of comparisons is $O(\max(nm, n\log n))$ and the number of computation included angle is $O(\frac{n^2}{m})$.

Key words Geometric algorithm, algorithmic complexity, travel salesman problem*.