第25卷第8期(总第164期) 系 统 工 程 Vol 25, No. 8
2007年8月 Systems Engineering Aug., 2007

文章编号: 1001-4098(2007)08-0049-05

城市交通网络中商业街的选址分析与计算*

肖 鹏[1,2], 徐寅峰[1,2], 代文强[3]

(1. 西安交通大学 管理学院, 陕西 西安 710049;
2. 西安交通大学 机械制造系统工程国家重点实验室, 陕西 西安 710049;
3. 电子科技大学 管理学院, 四川 成都 610054)

摘 要: 商业街在城市交通网络体系中的选址是商业街布局的重要方面。本文提出交通网络商业街选址问题并建立数学模型, 给出了两种选址原则: (1) 城市中所有节点至商业街(只需要达到商业街的其中一个节点)之间的最短路径之和最小。(2) 城市中所有点对之间过商业街的最短路径之和最小。针对以上两个方面, 分别给出其计算方法, 并进行了算法的时间和空间复杂性分析。

关键词: 商业街; 选址问题; 双子树; 双子合并树

中图分类号: O221; TB114 **文献标识码**: A

1 引言

商业街是指由大量零售业、服务业商店构成的具有一定长度的街区。商业街在城市商业的发展中起着支柱作用, 商业街在城市交通网络体系中的选址是商业街布局的重要方面。因此, 商业街(边)选址是城市规划的热点问题。以往关于选址问题的研究主要有两个方面的内容: (1) 点的选址问题。如经典的 k-median 问题, 既在具有 n 个点的网络中寻求 k 个点, 使得这 k 个点到其余的点的距离之和最小, 这个问题已经被证明是NP-hard的[1-3]。(2) 路径选址(path location)和树形结构选址(tree location)问题[4-9]。路径选址或树形结构选址问题是选择一条路径或树, 使得建设费用加上所有节点至这条路径或树的费用之和最小。Hedetniemi和Minieka讨论了在一颗树上选择一条路径或一颗子树[4,5]。Hakimi证明了不同类别的路径选址和树形结构选址问题的复杂性[6]。Wang研究了给定长度的条件下的路径选址问题, 并给出了最优的平行算法[7]。Tamir研究了网络图上点具有权重的路径选址问题[8], 并进一步考虑了连续的路径选址问题研究[9]。因此, 在以往的研究中, 并没有涉及边的选址问题。

本文从商业街交通空间网络体系建设的角度出发, 提出城市商业街(边)的选址问题, 给出了商业街的两种选择原则: (1) 方便城市中所有节点上的用户能够很快到达商业街进行购物, 因此, 第一种原则是使得城市中所有节点上的用户至商业街(只需要达到商业街的其中一个节点)之间的最短路径之和最小。(2) 方便城市中的所有用户从商业街购买货物后能尽快送货到达另外一个节点上的用户, 因此, 第二种原则是使得城市中所有点对之间过商业街的最短路径之和最小。针对以上两种问题, 分别给出其计算方法, 并进行了算法的时间和空间复杂性分析。

2 问题描述和建模

给定一个城市交通网络, 节点代表用户, 边代表城市街道。在城市网络中需要建设一条商业街, 研究以下两类商业街选址问题: (1) 城市中所有节点上的用户至商业街(只需要达到商业街的其中一个节点)之间的最短路径之和最小。(2) 城市中所有点对之间过商业街的最短路径之和最小。如果按照第一原则选择商业街, 则该商业街选址问题称为问题1(P1); 如果按照第二原则选择商业街, 则该商业街选址问题称为问题2(P2)。

根据这两类问题描述建立下列数学模型:

假设城市交通网络为图 $G=(V,E)$, 图 G 为一边加权无向图, 其中 V 为顶点集合, E 为边集合, 令 $|V|=n$, $|E|=m$.

P1: 任取点 $v_j\in V$ 和边 $e_i=(x_i,y_i)$, 定义该点 v_j 至边的距离 $d(v_j,e_i)$ 为点 v_j 分别至边的两个端点 x_i 和 y_i 的距

* 收稿日期: 2007-06-09
基金项目: 国家自然科学基金资助项目(70525004; 70121001; 70471035); 博士点基金资助项目(20050698048)
作者简介: 肖鹏(1979-), 男, 湖北汉川人, 西安交通大学管理学院博士研究生, 研究方向: 交通网络优化与算法设计。

© 1994-2007 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

离中的最小值，即 $d(v_j, e_i) = \min\{d(v_j, x_i), d(v_j, y_i)\}$，对任意边 $e_i = (x_i, y_i)$，定义其费用是 $C_1(e_i) = \sum_{v_j \in V} d(v_j, e_i)$。问题 P1的目标是在 G 中选择一条边 e^*，使得其费用满足 $C_1(e^*) = \min_{e_i \in E}\{\sum_{v_j \in V} d(v_j, e_i)\}$。

P2：任取两点 s_l、t_l 和边 $e_i = (x_i, y_i)$，定义两点 s_l 和 t_l 之间过边 $e_i = (x_i, y_i)$ 的最短路径为 $SP_{e_i}(s_l, t_l)$，其路径长记为 $d_{e_i}(s_l, t_l)$，对任意边 $e_i = (x_i, y_i)$，定义其费用是 $C_2(e_i) = \sum_{s_l, t_l \in V} d_{e_i}(s_l, t_l)$。问题P2的目标是在 G 中选择一条边 e^*，使得其费用满足 $C_2(e^*) = \min_{e_i \in E}\{\sum_{s_l, t_l \in V} d_{e_i}(s_l, t_l)\}$。

3 商业街选址问题 P1的求解

根据问题 P1中点至边的距离的定义可知，在计算的过程中必须明确所有点对之间的最短路径才能求解任何点至边的距离，根据这一思想可以用以下的算法进行求解。

步骤一：利用 Dijkstra 算法给出网络上任意两点的距离。

步骤二：对于给定边 $e_i = (x_i, y_i)$，比较每个点 v_j 至 x_i 和 y_i 的距离，得出点至边的距离为 $d(v_j, e_i)$。

步骤三：计算得出每条边的费用 $C_1(e_i)$，$i = 1, 2, \cdots, m$.

步骤四：计算费用最小的边，使 $C_1(e^*) = \min_{e_i \in E}\{C_1(e_i)\}$。

算法复杂性分析：

步骤一可以通过 $O(n^3)$ 时间计算得出[10]；步骤二计算所有点至给定边的距离的时间复杂度为 $O(n)$，共需要计算 m 条边，则时间复杂度为 $O(mn)$；步骤四的时间复杂度为 $O(m)$。因此，算法的时间复杂度为 $O(n^3)$。在进行计算的过程中，在计算一条边的费用时，所需要的空间为 $O(n)$，而共需要计算 $O(m)$ 条边，储存 $O(m)$ 个数据，所以该算法的空间复杂度为 $O(m)$。

4 商业街选址问题 P2的求解

根据问题 P2的数学模型，可以通过直接法进行求解：首先，计算通过边的所有点对的最短路径，即可求出该边的费用；然后循环求出所有边的费用，取得费用最小的边。由于网络图上所有点对共有 $O(n^2)$ 个，因此，计算一条边的费用为 $O(n^2)$，则计算 m 条边的费用时间复杂度为 $O(mn^2)$。因为该算法需要记录 $O(n^2)$ 个点对的数据，则空间复杂度为 $O(n^2)$。直接法的时间复杂度和空间复杂度都很大，下面给出更快的计算方法。算法思想：点对之间过边的最短路径是由边组成的，那么只需记录这些最短路径通过网络图中边的次数，边的次数与边的权重相乘即为这条边在最终费用中所占的份额，最后将所有边所占的份额相加即为边的费用 $C_2(e_i)$。

4.1 相关定义

为了计算方便，给出如下的一些定义：

(1) 最短路径树 $SPT(x_i)$：所有节点与节点 x_i 的最短路径组合生成的一颗树，节点 x_i 为树的根节点；

(2) 双子树 $Bi\text{-}Tree(e_i)$（简写为 $BT(e_i)$）：所有节点与边 e_i 的最短路径（与边的其中一个端点的最短路径）生成的一颗树（如图1所示）。对于每条边 e_i 可以得到对应一个 $BT(e_i)$。

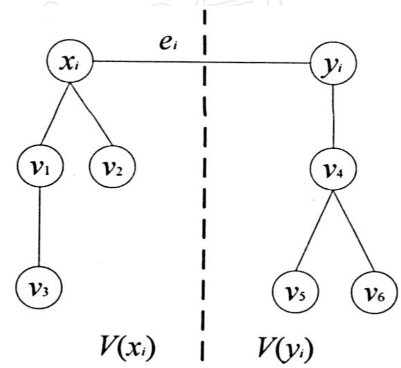

图1 双子树 $BT(e_i)$

(3) $V(x_i)$、$V(y_i)$：双子树中由边 e_i 中断分割为两个点集合，其中 $V(x_i)$ 是指包含 x_i 的点集合，$V(y_i)$ 是指包含 y_i 的点集合，且令 $|V(x_i)| = n_x$，$|V(y_i)| = n_y$；

(4) 父代节点：节点 v_j 的上一级连接的节点。x_i 和 y_i 无父代节点。

(5) 子代节点：节点 v_j 的下一级连接的节点；

(6) 后代节点集：节点 v_j 下的所有节点，不包括自身节点。节点 v_j 的所有后代的数量记为 $OS(v_j)$；

(7) 同构点：在 $SPT(x_i)$ 中是 y_i 后代或者在 $SPT(y_i)$ 中是 x_i 后代的节点，称为同构点。

(8) 异构点：在 $SPT(x_i)$ 中不是 y_i 后代，而在双子树中是 y_i 后代的节点，或者在 $SPT(y_i)$ 中不是 x_i 后代，而在双子树中是 x_i 后代的节点，称为异构点。

(9) 复制点：当节点 v_j 为异构点时，节点 v_j 在另一个点集合中仍然保留与边的另一端点相连的最短路径，保留的节点称为复制点 v_j'。若 $v_j \in V(x_i)$，则节点 $v_j' \in V(y_i)$，同理若 $v_j \in V(y_i)$，则节点 $v_j' \in V(x_i)$。

(10) $\Delta(v_j)$：当节点 v_j 为异构点时，节点 v_j 的偏移距离。若 $v_j \in V(x_i)$，则节点的偏移距离为 $\Delta(v_j) = d(v_j', y_i) - d(v_j, x_i)$。

(11) 双子合并树 $Combining\ Bi\text{-}Tree(e_i)$（简写为 $CBT(e_i)$）：在双子树 $BT(e_i)$ 中增加了所有的复制点，称此树为双子合并树（如图2所示）。

根据双子合并树的定义，可以得到以下结论：

引理1 增加复制点后的双子合并树是由 $SPT(x_i)$ 和

© 1994-2007 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

$SPT(y_i)$合并生成的, 合并规则:

(1) 在$SPT(x_i)$中y_i的子树中节点的位置不变, 在$SPT(y_i)$中x_i的子树中节点的位置不变;

(2) 其余的节点根据与x_i和y_i的距离大小进行判断, 距离较近的节点保持不变, 而较远的节点作为复制点。

根据异构点和偏移量的定义可以得到以下的性质:

性质1　异构点的后代也是异构点。

性质2　假如异构点v_{j1}是异构点v_{j2}的后代, 则$\Delta(v_{j1})\geq\Delta(v_{j2})$。

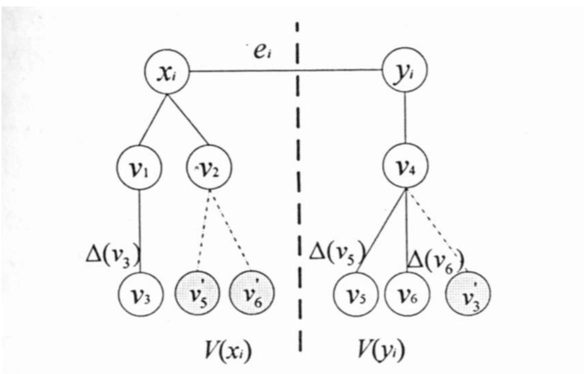

图2　双子合并树$CBT(e_i)$

4.2　双子合并树上最短路径分析

在双子合并树上, 两个节点之间过边的最短路径有以下两类情况:

情形1: 两个节点s_l和t_l分别位于$V(x_i)$、$V(y_i)$, 如图3所示, 则最短路径为粗线部分, 即为$s_l\rightarrow\cdots\rightarrow x_i\rightarrow y_i\rightarrow\cdots\rightarrow t_l$. 同理, 若两个节点$s_l$和$t_l$分别位于$V(y_i)$、$V(x_i)$, 则最短路径为$s_l\rightarrow\cdots\rightarrow y_i\rightarrow x_i\rightarrow\cdots\rightarrow t_l$.

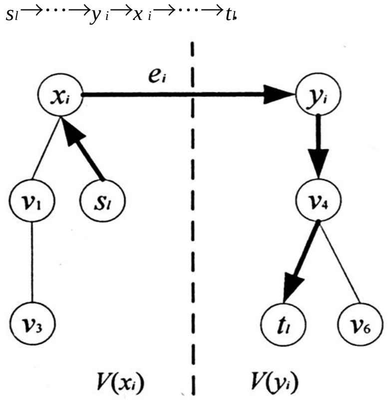

图3　情形1

情形2: 两个节点s_l和t_l位于同一个点集合中, 不妨设均在$V(y_i)$点集合中, 则分别有以下三种情形:

情形2-1: 当两个节点为同构点, 则最短路径为图4中粗线部分, 即$s_l\rightarrow\cdots\rightarrow y_i\rightarrow x_i\rightarrow y_i\rightarrow\cdots\rightarrow t_l$.

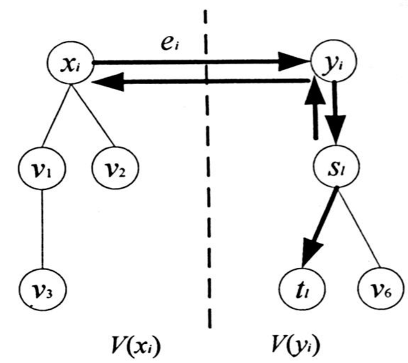

图4　情形2-1

情形2-2: 当其中一个节点s_l为同构点, 另一个节点t_l为异构点, 则最短路径为图5中粗线部分, 即$s_l\rightarrow\cdots\rightarrow y_i\rightarrow x_i\rightarrow\cdots\rightarrow t_l'$.

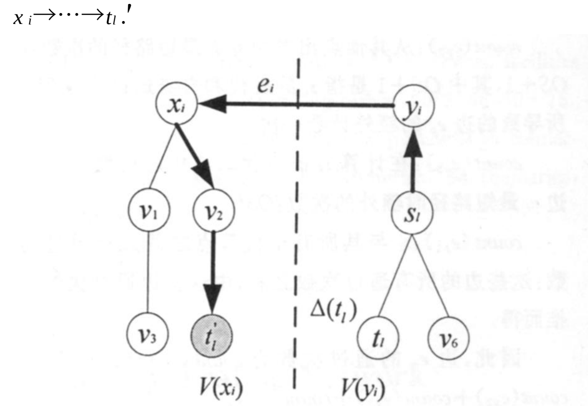

图5　情形2-2

情形2-3: 当两个节点为异构点, 则有两条路径均可能为最短路径, 如图6中粗线和点粗线所示, 分别为$s_l\rightarrow\cdots\rightarrow y_i\rightarrow x_i\rightarrow\cdots\rightarrow t_l$和$s_l'\rightarrow\cdots\rightarrow x_i\rightarrow y_i\rightarrow\cdots\rightarrow t_l$. 选择偏移量大的节点为不动点, 另一个节点选择其复制点, 则两个节点之间的最短路径即如第(1)类情形所示(一个异构点与另一个异构点的复制点分别处于两个集合中)。

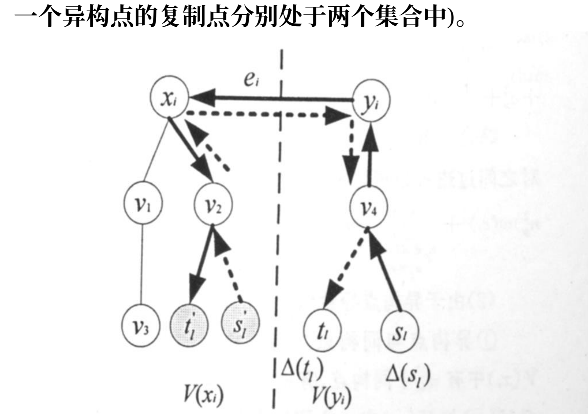

图6　情形2-3

根据上述四种情形下任意点对间的最短路径和异构

© 1994-2007 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net

点的偏移量，可以计算出任何两个节点之间过边 e_i 的最短路径。

4.3 运用双子合并树计算边费用

下面介绍如何根据双子合并树来计算所有两点之间过边的最短路径之和。思想是：将所有的节点作为同构点进行计算边的费用，然后减去由于异构点产生的所有偏移量。

(1) 所有节点均为同构点时所有两点之间过边的最短路径之和。

①任取双子合并树中的一个节点 s_l，则此节点与父代节点之间的边记为 e_p（非 e_i），边 e_p 通过的次数 $count(e_p)$ 由以下四种部分组成：

$count(e_{p1})$：从 s_l 出发到其他点的最短路径的次数：$n+OS+1$，其中 $OS+1$ 是指 s_l 通过边 e_i 到达自身以及 s_l 的后代的最短路径所导致的边 e_p 的额外计数一次。

$count(e_{p2})$：从其他点出发到 s_l 的最短路径的次数：$n+OS+1$，其中 $OS+1$ 是指 s_l 的后代和自身通过边 e_i 到达 s_l 所导致的边 e_p 的额外计数一次。

$count(e_{p3})$：在计算 s_l 的后代节点中任何两点计算过边 e_i 最短路径时额外的次数：OS^2。

$count(e_{p4})$：s_l 与其所有子代节点之间边所通过的次数：这些边的所有通过次数之和，由这些边通过次数的递推而得。

因此，边 e_p 的通过次数为 $count(e_p) = count(e_{p1}) + count(e_{p2}) + count(e_{p3}) + count(e_{p4})$。

②边 e_i 的通过次数 $count(e_i)$ 由两部分组成：

$count(e_{i1})$：集合 $V(x_i)$ 中节点到 $V(y_i)$ 中节点之间过边的最短路径，则通过 e_i 次数为 $n_x n_y$；同理，集合 $V(y_i)$ 中节点到 $V(x_i)$ 中节点之间过边的最短路径，则通过 e_i 次数也是 $n_x n_y$。

$count(e_{i2})$：集合 $V(x_i)$ 中任何两点之间过边的最短路径，则通过 e_i 次数为 $2n_x^2$；同理，集合 $V(y_i)$ 中任何两点，则通过次数为 $2n_y^2$。

因此，边 e_i 的通过次数为 $2n_x n_y + n_x^2 + n_y^2 = (n_x + n_y)^2 + n_x^2 + n_y^2 = n^2 + n_x^2 + n_y^2$。

综合①和②，则假设所有节点均为同构点时所有点对之间过边 e_i 最短路径之和为：$Com\_C(e_i) = (n^2 + n_x^2 + n_y^2)\,\omega(e_i) + \sum_{\substack{e_p \in BT(e_i) \\ e_p \neq e_i}} count(e_p)\,\omega(e_p)$。

(2) 由于异构点导致偏移量的计算。

①异构点与同构点之间偏移量的计算：假设集合 $V(x_i)$ 中有 n_{x1} 个同构点，有 n_{x2} 个异构点，则一个异构点 $v_j \in V(x_i)$ 与 $V(x_i)$ 中一个同构点的最短路径的偏移量（比两点均为同构点时两点之间最短路径少的部分）为 $\omega(e_i) - \Delta(v_j)$，因此，集合 $V(x_i)$ 中异构点所导致的总的偏移量为 $\Delta(x_{i1}) = \sum_{j=1}^{n_{x2}} n_{x1}(\omega(e_i) - \Delta(v_j))$；同理，假设集合 $V(y_i)$ 中有 n_{y1} 个同构点，有 n_{y2} 个异构点，则集合 $V(y_i)$ 中异构点所导致的总的偏移量为 $\Delta(y_{i1}) = \sum_{j=1}^{n_{y2}} n_{y1}(\omega(e_i) - \Delta(v_j))$。

②异构点与异构点之间偏移量的计算：假设集合 $V(x_i)$ 中有 n_{x2} 个异构点，将这些异构点的偏移量进行排序，令 $\Delta(v_1) \leq \Delta(v_2) \leq \cdots \leq \Delta(v_j) \leq \cdots \leq \Delta(v_{n_{x2}})$，由于任何两个异构点之间的最短路径一定是偏移量较大节点不变，偏移量较小节点进行偏移（即选择复制点），则根据双子合并树上最短路径的情形2-3，可知集合 $V(x_i)$ 中异构点与异构点之间产生的总的偏移量为：$\Delta(x_{i2}) = \sum_{j=1}^{n_{x2}} (n_{x2} - j + 1)(\omega(e_i) - \Delta(v_{n_{x2}-j+1}))$。

同理，集合 $V(y_i)$ 中有 n_{y2} 个异构点，则集合 $V(y_i)$ 中异构点与异构点之间产生的总的偏移量为：$\Delta(y_{i2}) = \sum_{j=1}^{n_{y2}} (n_{y2} - j + 1)(\omega(e_i) - \Delta(v_{n_{y2}-j+1}))$。

综合①和②，则由于异构点导致的偏移量为：$\Delta(e_i) = \Delta(x_{i1}) + \Delta(x_{i2}) + \Delta(y_{i1}) + \Delta(y_{i2})$。

通过(1)和(2)的计算可以得到最终的费用 $C_2(e_i)$ 为：

$$C_2(e_i) = Com\_C(e_i) - \Delta(e_i)$$

4.4 FPLC 算法及复杂性分析

利用双子合并树可以快速计算边的费用 $C_2(e_i)$，FPLC 算法（first partial and late combination algorithm）详细步骤如下：

(1) 问题 P1 求解后，可生成所有边的双子树，时间复杂度为 $O(n^3)$；

(2) 对于一条边 e_i 生成双子合并树，时间复杂度为 $O(n)$，共有 m 条边，故时间复杂度为 $O(mn)$；

(3) 对一个双子合并树中的两个子集 $V(x_i)$、$V(y_i)$ 中异构点的偏移量分别进行排序，时间复杂度为 $O(n\log n)$，共有 m 个双子合并树，故时间复杂度为 $O(mn\log n)$；

(4) 对一个双子合并树 $CBT(e_i)$，计算其费用 $C_2(e_i)$，时间复杂度为 $O(n)$，共有 m 个双子合并树，故时间复杂度为 $O(mn)$；

(5) 计算费用最小的边使得 $C_2(e^*) = \min_{e_i \in E}\{C_2(e_i)\}$，时间复杂度为 $O(m)$；

因此，FPLC 算法的时间复杂度为 $O(\max\{mn\log n, n^3\})$。FPLC 算法在求解过程中仅需要 $O(m)$ 的空间进行记录数据，因此 FPLC 算法的空间复杂度为 $O(m)$。

5 结束语

商业街的选址问题是城市规划和城市商业网络结构体系建设中的重要组成部分。本文所讨论的商业街选择

© 1994-2007 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

问题的原则有两个: (1) 所有节点(用户)至边(商业街)的距离之和最小, (2) 所有点对过边的最短路径(商户经由商业街提货送至另一用户)之和最小。对于第一个选择原则的单边选址问题, 给出了时间复杂度为 $O(n^3)$ 的算法; 对于第二个选择原则的单边选址问题, 给出了时间复杂度为 $O(\max\{mn\log n, n^3\})$ 的算法。两个算法的空间复杂度均为 $O(m)$。

对于文中所引入的问题, 有许多有待进一步深入研究的理论问题: (1) 特殊网络上 k-边选址问题的研究; (2) 加入初期建设费用的边选址问题研究; (3) 网络图上 k-边选址问题的研究等等。

参考文献:

[1] Garey M R, et al. Computers and intractability: a guide to the theory of NP-completeness[M]. San Francisco, CA: Freeman, 1979.

[2] Kariv O, et al. An algorithm approach to network location problems. Part II: The p-medians[J]. SIAM J. Appl Math., 1979, 37: 539~560.

[3] Drezner Z, Hamacher H. Facility location: application and theory[M]. Springer, 2002.

[4] Hedetniemi S M, Cockaine E J, Hedetniemi S T. Linear algorithms for finding the Jordan center and path center of a tree [J], Transport. Sci, 1981, 15: 98~114.

[5] Minieka E. The optimal location of a path or tree in a tree network[J]. Networks, 1985, 15: 309~321.

[6] Hakimi S L, Schmeichel E F, Labbe M. On locating path or tree shaped facilities on networks[J]. Networks, 1993, 23: 543~555.

[7] Wang B F. Efficient parallel algorithms for optimally locating a path and a tree of a specified length in a weighted tree network [J]. Journal of Algorithm, 2000, 34: 90~108.

[8] Tamir A, Puerto J, Perez-Brito D. The centdian subtree on tree networks[J]. Disc. Appl Math., 2002, 118: 263~278.

[9] Tamir A, Puerto J, Mesa J A, Rodriguez-Chia A M. Conditional location of path and tree shaped facilities on trees[J]. Journal of Algorithm, 2005, 56: 50~75.

[10] Dijkstra E W. A note on two problems in connection with graphs [J]. Numerical Mathematics, 1959, 1.

The Location of Business-Street in a City Traffic Network

XIAO Peng[1,2], XU Yin-feng[1,2], DAI Wen-qiang[3]

(1. School of Management, Xi'an Jiaotong University, Xi'an 710049, China;

2. The State Key Lab for Manufacturing Systems Engineering, Xi'an 710049, China;

3. School of Management, University of Electronic Science and Technology, Chengdu 610054, China)

Abstract: The location of Business-Streets in transportation network is important. In this paper, we formulate the problem of Business-Street location, and give two principles: one objective function is to minimize the sum of the distances which are from all nodes to the Business-Street, and the other objective function is to minimize the sum of the distances of all-pairs paths which have to pass through the Business-Street. For two Business-Street locating problems, we show two algorithms and an analysis of the time complexities and space complexities, respectively.

Key words: Business-Street; Location Problem; Bi-root Tree; Combining Bi-root Tree

© 1994-2007 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net