ELSEVIER

Discrete Applied Mathematics 68 (1996) 17–32

DISCRETE
APPLIED
MATHEMATICS

A heuristic with worst-case analysis for minimax routing of two travelling salesmen on a tree

Igor Averbakh[a], Oded Berman[b,*]

[a] *University of Toronto, Division of Management and Economics at Scarborough, 1265 Military Trail, Scarborough, Ont., Canada M1C 1A4*

[b] *University of Toronto Division of Management and Economics at Scarborough and the Faculty of Management, 246 Bloor Street, Toronto, Ont., Canada M5S 1V4*

Received 22 April 1993; revised 28 November 1994

Abstract

Suppose two travelling salesmen must visit together all points/nodes of a tree, and the objective is to minimize the maximal length of their tours. Home locations of the salesmen are given, and it is required to find optimal tours. For this NP-hard problem a heuristic with complexity O(n) is presented. The worst-case relative error for the heuristic performance is $\frac{1}{3}$ for the case of equal home locations for both servers and $\frac{1}{2}$ for the case of different home locations.

Keywords: Travelling salesman problem; Algorithms; Worst-case analysis

1. Problem formulation and notation

We consider the 2-travelling-salesmen allocation problem (2-TSP) on a tree with minimax criterion that can be interpreted as follows. There are two identical service units (servers), initially situated at some nodes A_1, A_2 of the tree (home locations). The servers are required to visit together some set DP of demand points and return back to their home locations; DP is either the set of all nodes or the set of all points of the tree. The objective is to minimize the maximum of the lengths of their tours.

Problems of this type arise in many services such as repair and maintenance, delivery and customer pick-up. Demand points are interpreted as customers that need some service. The tree represents a transportation network connecting the customers. The minimax objective is motivated, first, by the desire to distribute the workload to the servers in a "fair" way, second, by natural restrictions such as limited working day of the servers. Some approximation algorithms for minimax multiserver routing problems on general networks have been studied in [1].

*Corresponding author.

SSDI 0166-218X(95)00054-2

Let $T = (V, E)$ be a tree with V the set of nodes and E the set of edges, $|V| = n$. Also, T will denote the continuum set of all points of the tree. Throughout the paper, the term "subtree" is used in the topological sense: T' is a subtree of T iff T' is a connected subset of T. For any subtree $T' \subset T$, $L(T')$ denotes the length of T'; L denotes the length of T (the sum of the lengths of all the edges). Subtree visited (served) by a server in his service tour is referred to as *allocation* for that server.

TSP for the case of a single server on a tree is trivial: it is well-known that any depth-first route solves the problem, and the length of the optimal tour is equal to twice the length of the tree. Without any loss of generality, we assume that the service tour of each server is a depth-first route within his allocation, with length equal to twice the length of the allocation. With this assumption, the considered problems can be formulated in graph-theoretic terms, and we use lengths of allocations instead of lengths of service tours.

Problem 1 (Allocation minimax 2-TSP). Given home locations $A_1, A_2 \in V$, set $DP \in \{V, T\}$, find closed subtrees $F_1, F_2 \subset T$ (allocations) such that $DP \subset F_1 \cup F_2$ and $A_i \in F_i$, $i = 1, 2$, so as to minimize $\max\{L(F_1), L(F_2)\}$.

We distinguish between two variants of Problem 1 called "Problem 1-V", if $DP = V$, and "Problem 1-E", if $DP = T$. When a variant of the problem is not specified, reported results pertain to both variants.

In this paper, we present a heuristic with complexity $O(n)$ for solving (NP-hard) Problem 1. The worst-case relative error for the heuristic performance is $\frac{1}{3}$ for the case of equal home locations for both servers and $\frac{1}{2}$ for the case of different home locations. We note that for the case of equal home locations the problem can also be solved approximately using the tour partitioning heuristic developed in [1] for general networks, but our heuristic has a better worst-case bound – $\frac{1}{3}$ instead of $\frac{1}{2}$ for the tour partitioning heuristic (since we use tree structure of the network).

We use the following notation and definitions. For any two points $a, b \in T$ let $d(a, b)$ denote the distance between a and b. For an edge (v_1, v_2), let $x(v_1, v_2; r)$ denote the point of edge (v_1, v_2) which is r units away from $v_1 (0 < r < d(v_1, v_2))$, $x(v_1, v_2; 0) = v_1$, $x(v_1, v_2; d(v_1, v_2)) = v_2$. For any edge (a, b) it is assumed that points a, b do not belong to that edge; $[a, b]$ denotes edge (a, b) with points a, b. For any two points $c, d \in T$ let $P(c, d)$ denote the path between c and d.

Let X be a subtree of T. Connected components of set $T \backslash X$ are referred to as X-branches (note that X can be a single point). For any $b \notin X$ let $B(X, b)$ denote the unique X-branch containing b. Let L_*^V and L_*^E denote optimal values for problems 1-V and 1-E, respectively. For $(a, b) \in E$ let $W(a, b)$ denote the length of a-branch $B(a, b)$.

2. Complexity of the problem

Theorem 1. *Problem* 1 *is* NP-*complete even for stars* (*trees where all edges have a common node*).

Proof. It is evident that the recognition version of Problem 1 belongs to class NP. To prove NP-hardness we use reduction from the following problem.

Two-processor scheduling problem (TPSP). Tasks $t_1, \ldots, t_n$, and a positive integer length c_i for each task t_i are given. Find a two-processor schedule, i.e. a function $\delta: \{t_1, \ldots, t_n\} \to \{1,2\}$, so as to

$$\text{minimize} \quad \max_{j=1,2} \sum_{i:\delta(t_i)=j} c_i.$$

TPSP is known to be NP-hard [2].

Reduction. For an instance of TPSP $\{c_1, \ldots, c_n\}$ consider the instance of Problem 1 on star T' with a common vertex b and n edges (b, d_i), $d(b, d_i) = c_i$, $i = 1, \ldots, n$, and home locations $A_1 = A_2 = b$. It is easy to see that both instances of the problems are equivalent.

The theorem is proved. □

However, it is not clear whether Problem 1 is NP-complete in the strong sense or there exists a pseudopolynomial algorithm to solve it. The main difficulty is that in some cases optimal allocations F_1, F_2 inevitably have intersections of non-zero lengths. Consider, for example, the tree in Fig. 1: both home locations are at node a. Then edges (c, a) and (a, b) belong to both optimal allocations F_1, F_2.

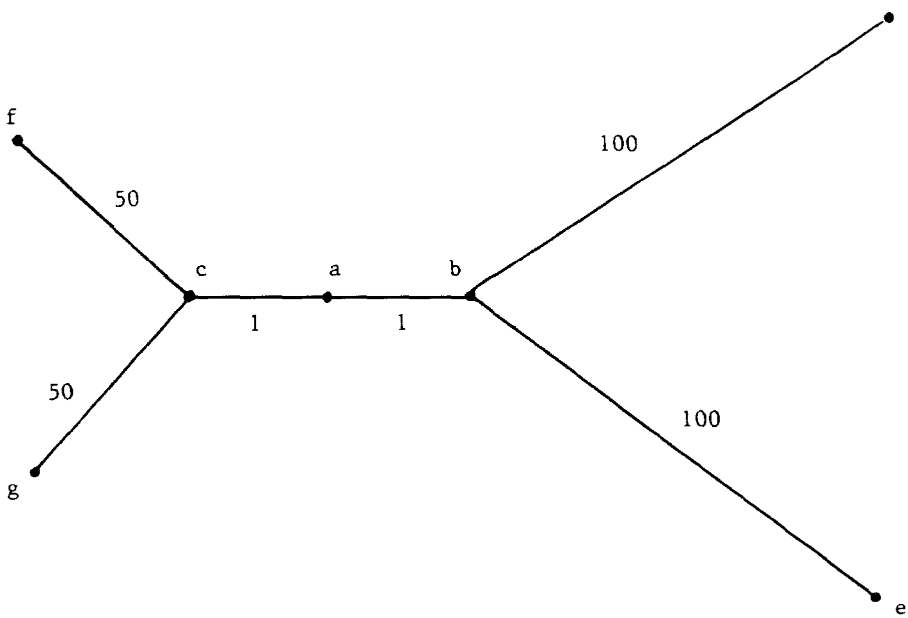

Fig. 1. Example of the problem where optimal allocations have intersection of nonzero length.

Consider a modification of Problem 1, where it is additionally required that total length $L(F_1) + L(F_2)$ of allocations F_1, F_2 must be minimal; this modification will be referred to as modified Problem 1. For the case $DP = T$ the additional restriction is equivalent to $L(F_1) + L(F_2) = L$, or $L(F_1 \cap F_2) = 0$. Therefore, for modified Problem 1-E set $F_1 \cap F_2$ is a single point f. If $f = F_1 \cap F_2$ is known, then modified Problem 1-E is trivially reduced to TPSP. Therefore, a pseudopolynomial algorithm for modified Problem 1-E can be constructed using a pseudopolynomial algorithm for TPSP [2] and the following observations:

(a) f belongs to path $P(A_1, A_2)$;
(b) f is either a node or the $(L/2)$-dividing point of T (point x is called $(L/2)$-dividing point if all x-branches have lengths not greater than $L/2$);
(c) the $(L/2)$-dividing point of a tree can be obtained in $O(n)$ time.

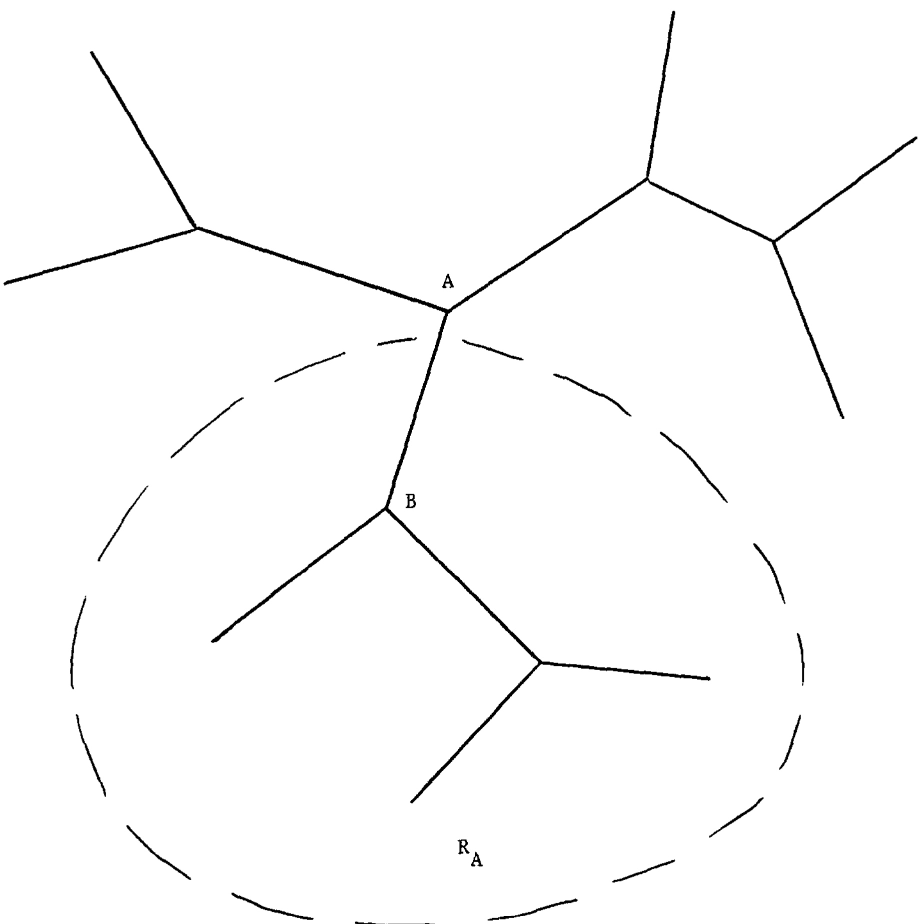

Fig. 2. Illustration required for the description of Step 1 of Heuristic H1.

A pseudopolynomial algorithm for modified Problem 1-V can be constructed using analogous ideas and the following observation: either $F_1 \cap F_2$ is a single point, or $L \backslash (F_1 \cup F_2)$ is an edge belonging to path $P(A_1, A_2)$.

3. A $\frac{4}{3}$-optimal heuristic for the case of a single home location

Let $A_1 = A_2 = A$. Problems 1-V and 1-E are equivalent for this case, $L_*^V = L_*^E = L_*$. Let Y denote the intersection of allocations F_1, F_2. At each step of the heuristic either a $\frac{4}{3}$-optimal solution to Problem 1 is found (a solution is called $(1+\varepsilon)$-optimal if its relative error is not greater than ε), or a new edge is included in common set Y; at the beginning $Y = \{A\}$.

Heuristic H1. Tree $T = (V, E)$ of length L and locations $A_1 = A_2 = A$ are given. Set $Y$is initially equal to $\{A\}$.

Step 1: Consider all A-branches. (1) If there is an A-branch with length φ such that $\frac{1}{3}L \leqslant \varphi \leqslant \frac{2}{3}L$ then assign this A-branch to one server, assign all other A-branches to the other server and STOP: a $\frac{4}{3}$-optimal solution is obtained (since $L_* \geqslant \frac{1}{2}L$).

(2) If all the A-branches have lengths smaller than $\frac{1}{3}L$, then assign consecutively A-branches to the servers as follows: an A-branch is assigned to that server which can accept it without exceeding the limit $\frac{2}{3}L$. When all the A-branches are assigned (obviously all the A-branches will be assigned), then STOP: a $\frac{4}{3}$-optimal solution is obtained.

(3) If there is an A-branch R_A of length greater than $\frac{2}{3}L$, consider node B adjacent to A, $B \in R_A$ (Fig. 2). Memorize $L(\mathrm{R}_A)$ as a record value and allocations $F_1 = R_A \cup \{A\}$, $F_2 = T \backslash R_A$ as a record solution.

If $d(A,B) \geqslant \frac{1}{2}L$, then STOP: the record solution is $\frac{4}{3}$-optimal. If $d(A,B) < \frac{1}{2}L$, then include edge $[A, B]$ in common set Y and go to step 2.

Step k, $k = 2, 3, \ldots$. At step $k-1$ a new edge was included in common set Y. Suppose it was edge $[C, D]$, and node D had not been in Y before. According to the description, Y is a path from A to D of length smaller than $\frac{1}{2}L$.

Consider all D-branches, which do not contain point C. These branches are called appropriate D-branches.

(1) If there is an appropriate D-branch R_D of length φ, $\frac{1}{3}(L - L(Y)) \leqslant \varphi \leqslant \frac{2}{3}(L - L(Y))$, assign this branch and set Y to one server, assign set $T \backslash R_D$ to the other server. Compare this solution with the record solution, take the best one and STOP: the obtained solution is $\frac{4}{3}$-optimal.

(2) If all the appropriate D-branches have lengths smaller than $\frac{1}{3}(L - L(Y))$ (note that the situation where there is no appropriate D-branches corresponds to this case), then all Y-branches have lengths smaller than $\frac{2}{3}(L - L(Y))$ (since $L(Y) < L/2$ and all Y-branches according to the description have lengths smaller than $L/3$). Now, if there is a Y-branch of length φ, $\frac{1}{3}(L - L(Y)) \leqslant \varphi \leqslant \frac{2}{3}(L - L(Y))$ then assign this branch and set Y to one server, all other Y-branches and set Y to the other server. Compare

this solution with the record solution, take the best one and STOP: a $\frac{4}{3}$-optimal solution is obtained. If all Y-branches have lengths smaller than $\frac{1}{3}(L - L(Y))$, then assign consecutively Y-branches to the servers as follows: a Y-branch is assigned to that server which can accept it without exceeding the limit $\frac{2}{3}(L - L(Y)) + L(Y)$ (set Y is assumed to be assigned to both servers in advance). When all the Y-branches are assigned (obviously all the Y-branches will be assigned), then compare the obtained solution with the record solution, take the best one and STOP: a $\frac{4}{3}$-optimal solution is obtained.

(3) If there is an appropriate D-branch R_D of length greater than $\frac{2}{3}(L - L(Y))$, consider node E adjacent to D, $E \in R_D$ (Fig. 3). Take solution $F_1 = R_D \cup Y$, $F_2 = T \backslash R_D$, compare this solution of value $L(Y) + L(R_D)$ with the record solution, take the best one as a new record solution. Now, if $L(Y) + d(D,E) \geqslant L/2$, then STOP: the record solution is $\frac{4}{3}$-optimal. If $L(Y) + d(D,E) < L/2$, then include $[D,E]$ into Y and go to step $k + 1$.

Theorem 2. *Heuristic H1 obtains a $\frac{4}{3}$-optimal solution to Problem 1 for the case of equal home locations $A_1 = A_2 = A$.*

Proof. Parts 1, 2 of the description of the first step need no explanations. Consider part 3: there is an A-branch R_A of length greater than $\frac{2}{3}L$. There are two possibilities for an optimal solution:

(1) edge (A,B) is served by only one server,

(2) edge (A,B) is served by both servers.

In the first case, A-branch R_A of length greater than $\frac{2}{3}L$ is entirely served by only one server and solution $F_1 = R_A \cup \{A\}$, $F_2 = T \backslash R_A$ of value $L(R_A)$ is obviously the best one. Therefore, the record solution is the best solution from all allocations such that (A,B) is served by only one server.

In the second case, if $d(A,B) \geqslant L/2$, then $L_* \geqslant \frac{3}{4}L$. Therefore, if edge (A,B) is served by both servers and has length not smaller than $\frac{1}{2}L$, then any allocations are $\frac{4}{3}$-optimal (since any allocations have value not greater than L), and the record solution is $\frac{4}{3}$-optimal too.

After step $k - 1$, if a $\frac{4}{3}$-optimal solution has not been obtained during this step, the record solution is the best one from all allocations such that current set Y is not entirely served by both servers.

Consider step k, $k = 2, 3, \ldots$. Parts 1, 2 need no explanations. Consider part 3: there is an appropriate D-branch R_D of length greater than $\frac{2}{3}(L - L(Y))$. Note that we should take into account only solutions such that current common set Y is served by both servers, since the current record solution is the best one from all other solutions. There are two possibilities for an optimal solution:

(1) edge (D,E) is served by only one server;

(2) edge (D,E) is served by both servers.

In the first case, D-branch R_D of length greater than $\frac{2}{3}(L - L(Y))$ is served by only one server and solution $F_1 = R_D \cup Y$, $F_2 = T \backslash R_D$ is the best one from all allocations

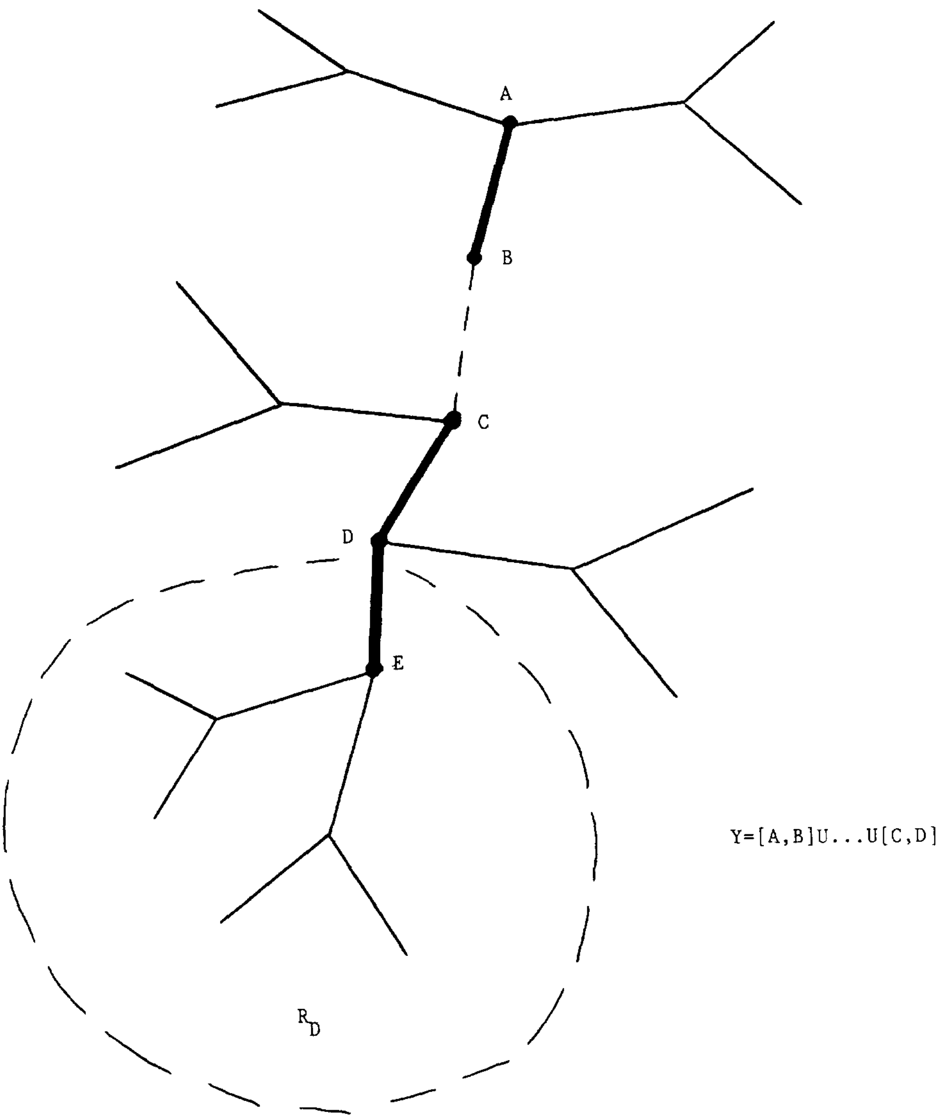

Fig. 3. Illustration required for the description of Step k of Heuristic H1.

such that set Y is served by both servers but edge (D, E) is served by only one of them. Therefore, the best one from this solution and the current record solution, taken as a new record solution, is the best solution from all allocations in which set $Y \cup (D, E)$ is not served entirely by both servers.

In the second case, if $L(Y) + d(D, E) \geqslant L/2$, then $L_* \geqslant \frac{3}{4}L$ (since set $Y \cup (D, E)$ of length not smaller than $L/2$ is served by both servers, and the servers must serve together the remaining part $T \backslash (Y \cup (D, E))$); therefore, if Y and (D, E) are served by both servers and $L(Y) + d(D,E) \geqslant L/2$, then any feasible solution (and the record one too) is $\frac{4}{3}$-optimal.

The number of steps of the algorithm is not greater than $|V| - 1 = n - 1$, since at each step either a $\frac{4}{3}$-optimal solution is found or a new edge is included in common set Y.

The theorem is proved. □

Theorem 3. *Heuristic $H1$ can be implemented in time* $O(n)$.

Proof. Note that values $W(a,b)$, $W(b,a)$ for all edges $(a,b) \in E$ can be calculated in total time $O(n)$.

At each step of Heuristic H1, except the last step, only appropriate D-branches are examined, where D is the new node included in set Y at step k-1 (if $k = 1$, then $D = A$). Therefore, if the number of steps of Heuristic H1 is M, then the number of branches examined at first M-1 steps is not greater than $|E| = n$-1 and each branch is examined only once. Examination of each appropriate D-branch takes constant (unit) time if values $W(a,b)$, $W(b,a)$ are given for all $(a,b) \in E$. Therefore, the first M-1 steps can be implemented in time $O(n)$. The last step consumes time $O(n)$. The theorem is proved. □

4. A $\frac{3}{2}$-optimal heuristic for the case of different home locations

Let $A_1 \neq A_2$. Consider Problem 1-E. Let $b_1, \ldots, b_r$ be the nodes of path $P(A_1, A_2)$ in consecutive order from A_1 to A_2, $b_1 = A_1$, $b_r = A_2$ (Fig. 4).

Heuristic H2. Let $L_1 = W(b_2, b_1) - d(b_1, b_2)$, $L_2 = W(b_{r-1}, b_r) - d(b_{r-1}, b_r)$. $L_1(L_2)$ is the total length of all A_1-branches (A_2-branches) which do not include any edge from path $P(A_1, A_2)$.

Case 1: If $\frac{1}{3}L \leqslant L_1 \leqslant \frac{2}{3}L$ ($\frac{1}{3}L \leqslant L_2 \leqslant \frac{2}{3}L$), then assign all A_1-branches (A_2-branches) which do not contain edge (b_1, b_2) (edge (b_{r-1}, b_r)) to one server and the only remaining A_1-branch $B(b_1, b_2)$ (A_2-branch $B(b_r, b_{r-1})$) to the other server and STOP: a $\frac{4}{3}$-optimal solution is obtained (since optimal value L_*^E is not smaller than $L/2$).

Case 2: If $L_1 > \frac{2}{3}L$ ($L_2 > \frac{2}{3}L$) then obviously there exists an optimal solution such that point $A_1(A_2)$ is served by both servers. Consider the auxiliary problem, where basic tree T is the same, but both servers are situated in $A_1(A_2)$, and apply Heuristic H1. Let F_1, F_2 be the $\frac{4}{3}$-optimal solution for this auxiliary problem, obtained by Heuristic H1; then F_1, F_2 is a $\frac{4}{3}$-optimal solution for the original problem as well. STOP.

Case 3: Now consider the remaining case $L_1 < \frac{1}{3}L$, $L_2 < \frac{1}{3}L$. Consider values $W(b_{i+1}, b_i)$ for edges (b_i, b_{i+1}), $i = 1, 2, \ldots, r-1$, of path $P(A_1, A_2)$.

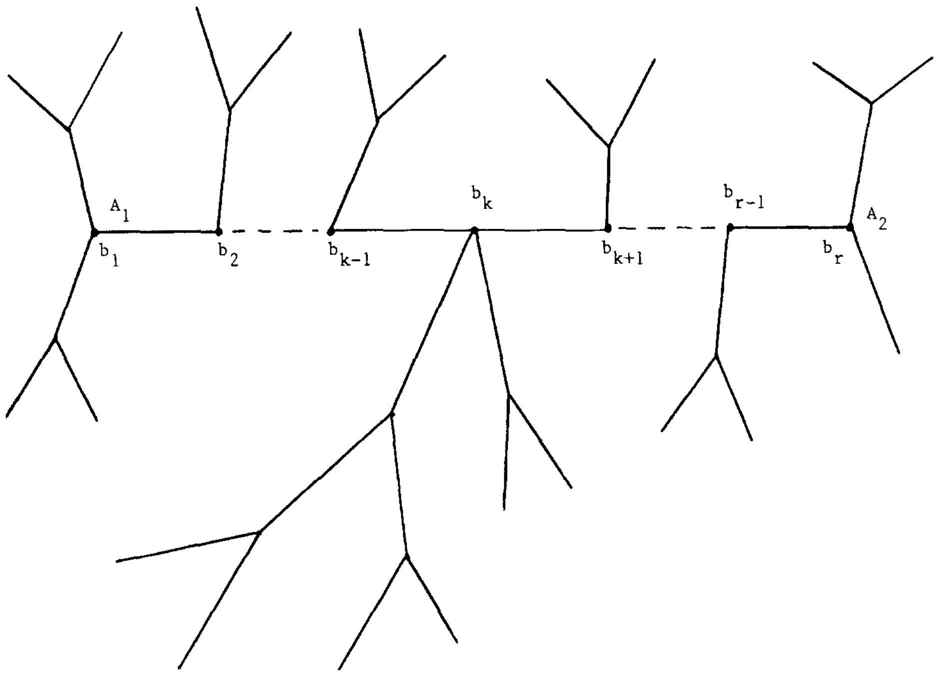

Fig. 4. Illustration required for the description of Heuristic H2.

Subcase 3.1: If for some t

$$\tfrac{1}{2}L \leqslant W(b_{t+1}, b_t) \leqslant \tfrac{1}{2}L + d(b_{t+1}, b_t), \tag{1}$$

then assign set $[b_t, x(b_{t+1}, b_t;\ W(b_{t+1}, b_t) - \frac{1}{2}L)]$ and all b_t-branches which do not include point b_{t+1} to one server, set $[b_{t+1}, x(b_{t+1}, b_t;\ W(b_{t+1}, b_t) - L/2)]$ and all b_{t+1}-branches which do not include point b_t to the other server and STOP: an optimal solution of value $\frac{1}{2}L$ is obtained.

Subcase 3.2: If there is no t such that condition (1) holds, then there is k, $1 < k < r$, such that

$$W(b_{k+1}, b_k) - d(b_k, b_{k+1}) \geqslant \tfrac{1}{2}L, \tag{2}$$

$$W(b_{k-1}, b_k) - d(b_{k-1}, b_k) \geqslant \tfrac{1}{2}L. \tag{3}$$

Subcase 3.2.1: If $\frac{1}{4}L \leqslant \mathrm{W}(b_k, b_{k+1}) \leqslant \frac{1}{2}L$ $(\frac{1}{4}L \leqslant \mathrm{W}(b_k, b_{k-1}) \leqslant \frac{1}{2}\mathrm{L})$, then assign branch $B(b_k, b_{k+1})$ (branch $B(b_k, b_{k-1})$) to one server, all other b_k-branches to the other server and STOP: a $\frac{3}{2}$-optimal solution is obtained.

Note that since conditions (2) and (3) hold, values $W(b_k, b_{k+1})$, $W(b_k, b_{k-1})$, cannot be greater than $\frac{1}{2}L$.

Subcase 3.2.2: If $W(b_k, b_{k+1}) < \frac{1}{4}L$, $W(b_k, b_{k-1}) < \frac{1}{4}L$, then obviously there exists an optimal solution F_1, F_2 such that $b_k \in F_1 \cap F_2$. Let $T_1 = W(b_k, b_{k-1})$, $T_2 = W(b_k, b_{k+1})$. Without loss of generality, suppose $T_1 \leqslant T_2$.

Perform the following:

Procedure R. Y will denote the intersection of allocations F_1, F_2. At the beginning $Y = \{b_k\}$. At each step of Procedure R either a $\frac{3}{2}$-optimal solution is found, or a new edge is included in set Y.

Step 1: Consider all b_k-branches, which do not include points b_{k-1}, b_{k+1}; we call them appropriate b_k-branches at the first step.

(1) If there is an appropriate b_k-branch B with length φ such that $\frac{1}{4}L \leqslant \varphi \leqslant \frac{3}{4}L - T_1$, then assign this branch and branch $B(b_k, b_{k-1})$ to one server, branch $B(b_k, b_{k+1})$ and other appropriate b_k branches to the other server and STOP: the obtained solution $F_1 = B \cup B(b_k, b_{k-1}) \cup \{b_k\}$, $F_2 = (T \backslash F_1) \cup \{b_k\}$ is $\frac{3}{2}$-optimal.

(2) If all appropriate b_k-branches have lengths smaller than $\frac{1}{4}L$, then assign branch $B(b_k, b_{k-1})$ to one server, assign branch $B(b_k, b_{k+1})$ to the other server, and assign all appropriate b_k-branches consecutively to the servers as follows: a b_k-branch is assigned to that server which can accept it without exceeding the limit $\frac{3}{4}L$. When all the b_k-branches are assigned (obviously all the b_k-branches will be assigned), then STOP: a $\frac{3}{2}$-optimal solution is obtained.

(3) If there is an appropriate b_k-branch R_{b_k} of length greater than $\frac{3}{4}L - T_1$, then consider node B adjacent to b_k, $B \in R_{b_k}$ (Fig. 5). Memorize $L(R_{b_k}) + T_1$ as a record value and allocations $F_1 = R_{b_k} \cup B(b_k, b_{k-1}) \cup \{b_k\}$, $F_2 = (T \backslash F_1) \cup \{b_k\}$ as a record solution. If $d(b_k, B) \geqslant \frac{1}{3}L$, then STOP: the record solution is $\frac{3}{2}$-optimal. If $d(b_k, B) < \frac{1}{3}L$, then include edge $[b_k, B]$ in set Y and go to step 2.

Step k, $k = 2, 3, \ldots$. At step $k - 1$ a new edge was included in common set Y. Let it be edge $[C, D]$, and node D had not been in Y before. Y is the path from b_k to D of length less than $\frac{1}{3}L$.

Consider all D-branches, which do not contain point C; these branches will be called appropriate D-branches.

(1) If there is an appropriate D-branch R_D of length φ, $\frac{1}{4}(L - L(Y)) \leqslant \varphi \leqslant \frac{3}{4}(L - L(Y)) - T_1$, assign this branch, set Y and branch $B(b_k, b_{k-1})$ to one server, and assign set $T \backslash (R_D \cup B(b_k, b_{k-1}))$ to the other server. Compare this solution with the record solution, take the best one and STOP: the obtained solution is $\frac{3}{2}$-optimal.

(2) If all appropriate D-branches have lengths smaller than $\frac{1}{4}(L - L(Y))$ (note that the situation where there is no appropriate D-branches corresponds to this case), then all other Y-branches have lengths smaller than $\frac{3}{8}(L - L(Y))$ (since $L(Y) < \frac{1}{3}L$ and all Y-branches have lengths smaller than $\frac{1}{4}L$). Now:

(a) If $T_2 \geqslant \frac{1}{4}(L - L(Y))$ then compare solution $F_2 = B(b_k, b_{k+1}) \cup \{b_K\}$, $F_1 = T \backslash B(b_K, b_{K+1})$ with the record solution, take the best one and STOP: a $\frac{3}{2}$-optimal solution is obtained.

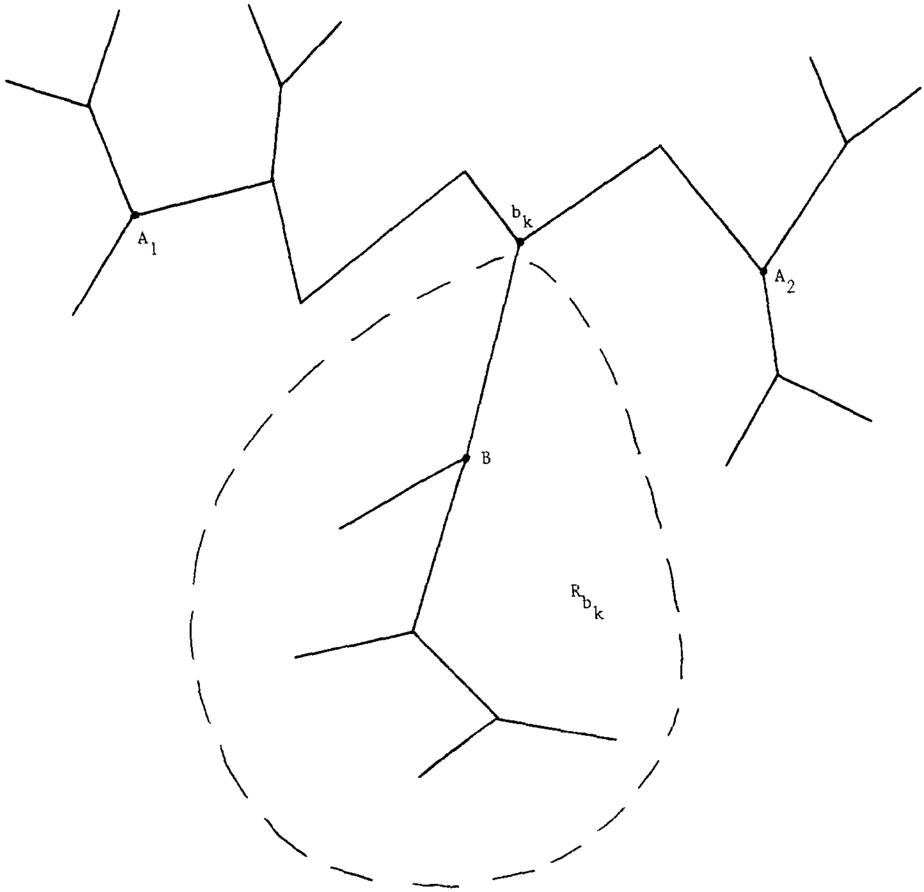

Fig. 5. Illustration for the description of Heuristic H2.

(b) If $T_2 < \frac{1}{4}(L - L(Y))$, but there is a Y-branch of length φ such that $\frac{1}{4}(L - L(Y)) \leqslant \varphi \leqslant \frac{3}{8}(L - L(Y))$, then assign this Y-branch, set Y and branch $B(b_k, b_{k-1})$ to one server, all other Y-branches and set Y to the other server. Compare this solution with the record solution, take the best one and STOP: a $\frac{3}{2}$-optimal solution is obtained.

(c) If all Y-branches have lengths smaller than $\frac{1}{4}(L - L(Y))$, then assign $B(b_k, b_{k-1})$ and set Y to one server, assign $B(b_k, b_{k+1})$ and set Y to the other server and assign consecutively all other Y-branches to the servers as follows: a Y-branch is assigned to that server which can accept it without exceeding the limit $\frac{3}{4}(L - L(Y)) + L(Y)$. When all the Y-branches are assigned (obviously all the branches will be assigned), then compare the obtained solution with the record solution, take the best one and STOP: a $\frac{3}{2}$-optimal solution is obtained.

(3) If there is an appropriate D-branch R_D of length greater than $\frac{3}{4}(L - L(Y)) - T_1$ then consider node E adjacent to D, $E \in R_D$ (Fig. 6). Take solution

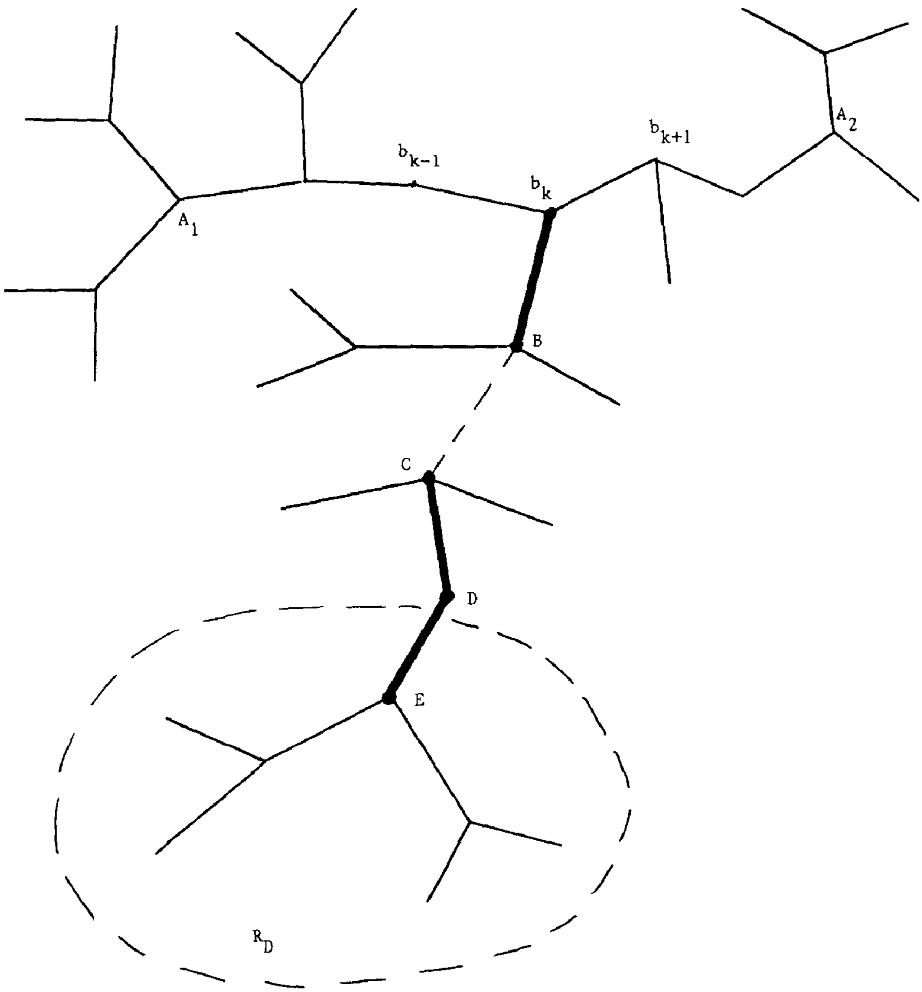

Fig. 6. Illustration for the description of Heuristic H2.

$F_1 = R_D \cup Y \cup B(b_k, b_{k-1})$, $F_2 = T \backslash (R_D \cup B(b_k, b_{k-1}))$, compare this solution of value $L(Y) + L(R_D) + T_1$ with the record solution, take the best one as a new record solution. Now, if $L(Y) + d(D,E) \geqslant \frac{1}{3}L$, then STOP: the record solution is $\frac{3}{2}$-optimal. If $L(Y) + d(D,E) < L/3$, then include edge $[D,E]$ into Y and go to step $k+1$.

Theorem 4. *Heuristic H2 obtains a $\frac{3}{2}$-optimal solution for Problem 1-E for the case of different home locations.*

Proof. In the description of Heuristic H2, only Procedure R in Subcase 3.2.2 needs explanation.

If Procedure R stops at part 1 of step 1, then the obtained solution is $\frac{3}{2}$-optimal since its value is not greater than $\frac{3}{4}L$ and optimal value L_*^E is not smaller than $\frac{1}{2}L$. The same about part 2 of step 1.

Consider part 3 of step 1: there is a b_k-branch R_{b_k} of length greater than $\frac{3}{4}L - T_1$. There are two possibilities for an optimal solution:

(1) edge (b_k, B) is served by only one server;

(2) edge (b_k, B) is served by both servers.

In the first case, b_k-branch R_{b_k} of length greater than $\frac{3}{4}L - T_1 > \frac{1}{2}L$ (since $T_1 < \frac{1}{4}L$) is served by only one server, therefore $L_*^E > L(R_{b_k})$ and solution $F_1 = R_{b_k} \cup B(b_k, b_{k-1}) \cup \{b_k\}$, $F_2 = (T \setminus F_1) \cup \{b_k\}$ of value $L(R_{b_k}) + T_1$ is $\frac{3}{2}$-optimal (since $(L(R_{b_k}) + T_1)/L_*^E \leqslant (L(R_{b_k}) + T_1)/L(R_{b_k}) \leqslant (L/2 + T_1)/(L/2) \leqslant 3/2$).

Therefore the record solution is $\frac{3}{2}$-optimal among all allocations such that edge (b_k, B) is served by only one server.

In the second case, if $d(b_k, B) \geqslant L/3$ then $L_*^E \geqslant \frac{2}{3}L$. Therefore, if edge (b_k, B) is served by both servers and has length not smaller than $\frac{1}{3}L$, then any feasible solution is $\frac{3}{2}$-optimal, and the record solution is $\frac{3}{2}$-optimal too.

After step $k-1$, if a $\frac{3}{2}$-optimal solution was not obtained at this step, the record solution is $\frac{3}{2}$-optimal among all allocations such that current common set Y is not entirely served by both servers.

Consider step k, $k = 2, 3, \ldots$. If procedure R stops at part 1 of step k, then the obtained solution is $\frac{3}{2}$-optimal, since if set Y is served by both servers, then optimal value L_*^E is not smaller than $\frac{1}{2}(L - L(Y)) + L(Y)$. The same about part 2 of step k. Consider part 3 of step k: there is an appropriate D-branch R_D of length greater than $\frac{3}{4}(L - L(Y)) - T_1$. We should take into account only solutions such that current set Y is served by both servers, since the current record solution is $\frac{3}{2}$-optimal among all other solutions. There are two possibilities for an optimal solution:

(1) edge (D, E) is served by only one server;

(2) edge (D, E) is served by both servers.

In the first case, D-branch R_D of length greater than $\frac{3}{4}(L - L(Y)) - T_1$ is served by only one server and solution $F_1 = R_D \cup Y \cup B(b_k, b_{k-1})$, $F_2 = T \setminus (R_D \cup B(b_k, b_{k-1}))$ of value $L(R_D) + L(Y) + T_1$ is $\frac{3}{2}$-optimal among all allocations such that set Y is served by both servers but edge (D, E) is served by only one of them. This is because

$$\frac{L(R_D) + L(Y) + T_1}{L_*^E} \leqslant \frac{L(R_D) + L(Y) + T_1}{L(R_D) + L(Y)} \leqslant \frac{\frac{3}{4}(L - L(Y)) - T_1 + L(Y) + T_1}{\frac{3}{4}(L - L(Y)) - T_1 + L(Y)}$$

$$\leqslant \frac{\frac{3}{4}L}{\frac{3}{4}L - T_1} \leqslant \frac{3}{2}.$$

Therefore the best one from this solution and the current record solution, taken as a new record solution, is a $\frac{3}{2}$-optimal solution among all allocations in which set $Y \cup (D, E)$ is not served entirely by both servers.

In the second case, if $L(Y) + d(D,E) \geqslant \frac{1}{3}L$, then $L_*^E \geqslant \frac{2}{3}L$ (since set $Y \cup (D,E)$ of length not smaller than $\frac{1}{3}L$ is served by both servers, and the servers must serve together the remaining part $T \backslash (Y \cup (D,E))$); thus, if Y and (D,E) are served by both servers and $L(Y) + d(D,E) \geqslant \frac{1}{3}L$, then any solution (and the record one too) is $\frac{3}{2}$-optimal.

The number of steps of procedure R is not greater than $|E| = n - 1$, since at each step either a $\frac{3}{2}$-optimal solution is found or a new edge is included in common set Y. □

Theorem 5. *Heuristic H2 can be implemented in time* $O(n)$.

Proof is analogous to the proof of Theorem 3.

Problem 1-V, $A_1 \neq A_2$, can be solved using Heuristic H2 as follows. Consider values $W(b_{i+1}, b_i)$ for edges (b_i, b_{i+1}), $i = 1, 2, \ldots, r-1$, of path $P(A_1, A_2)$ (Fig. 4). If for some t, $1 \leqslant t < r$, condition (1) holds, then $F_1 = T \backslash B(b_t, b_{t+1})$, $F_2 = T \backslash B(b_{t+1}, b_t)$ is our optimal solution. If there is no t such that condition (1) holds, then $L_*^V \geqslant \frac{1}{2}L$ and there is an optimal solution to the problem such that $F_1 \cup F_2 = T$. Therefore, Heuristic H2 in this case obtains a $\frac{3}{2}$-optimal solution.

5. Worst-case analysis

The obtained bounds for the heuristics' performance errors cannot be improved, as the following theorem demonstrates.

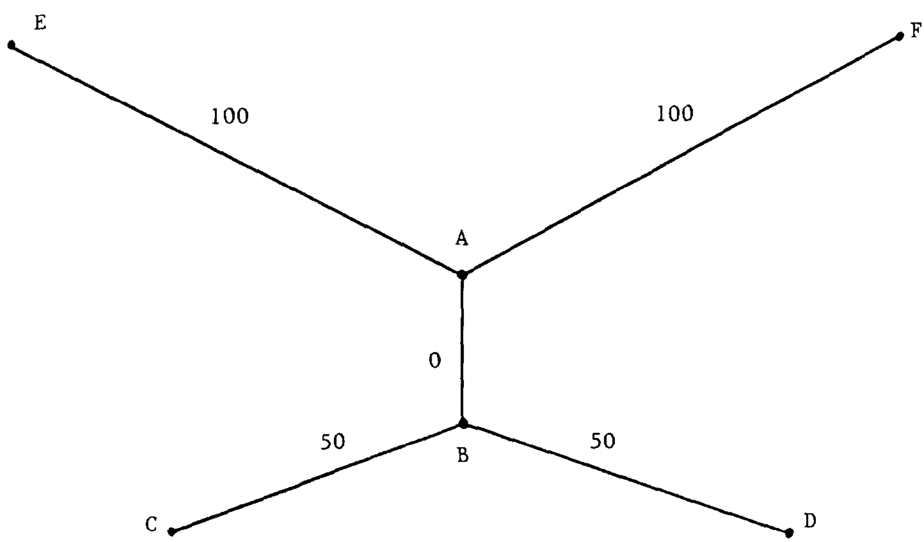

Fig. 7. Example of tightness for Heuristic H1.

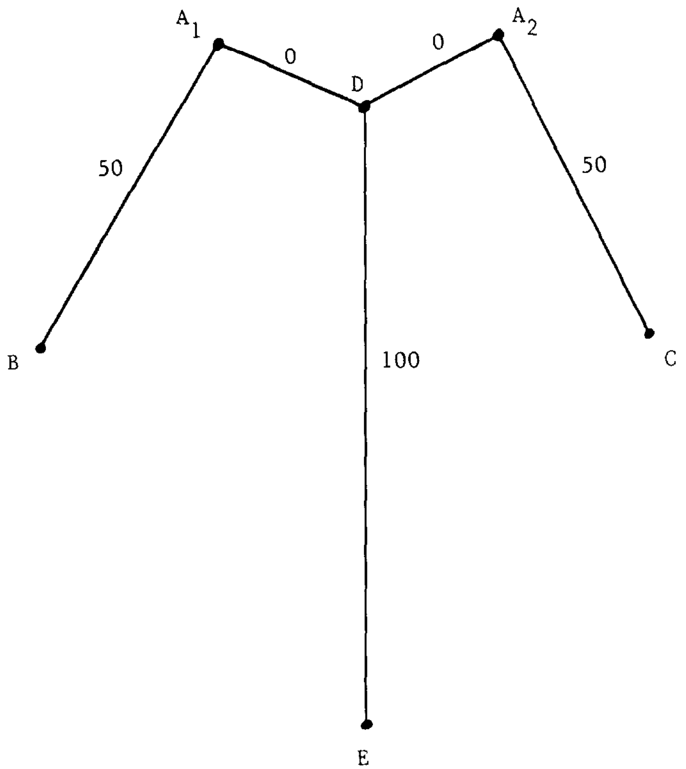

Fig. 8. Example of tightness for Heuristic H2.

Theorem 6. *The worst-case relative errors for heuristics H1 and H2 are* $\frac{1}{3}$ *and* $\frac{1}{2}$ *respectively.*

Proof. An example of tightness for heuristic H1 is demonstrated in Fig. 7 (both servers are located at node A), $L = 300$. Heuristic H1 obtains a solution of value 200, but the optimal value is $L_* = 150$ (for example, $F_1 = [E,A] \cup [A,B] \cup [B,C]$, $F_2 = [F,A] \cup [A,B] \cup [B,D]$) and the relative error is $(200 - 150)/150 = \frac{1}{3}$.

An example of tightness for Heuristic H2 is demonstrated in Fig. 8 (the servers are located at A_1 and A_2), $L = 200$. Heuristic H2 obtains a solution of value 150, but the optimal value is $L_*^E = L_*^V = 100$ (for example, $F_1 = [A_1, D] \cup [D, E]$, $F_2 = [C, A_2] \cup [A_2, D] \cup [D, A_1] \cup [A_1, B]$) and the relative error is $(150 - 100)/100 = \frac{1}{2}$.

Edges of zero lengths can be considered as edges of lengths equal to ε, $\varepsilon \to 0$. $\square$

6. Conclusions and future research

The main result of the paper is a linear-time heuristic for the (NP-complete) allocation minimax 2-TSP on a tree. The worst-case relative error for the heuristic performance is $\frac{1}{3}$ for the case of equal home locations for both servers and $\frac{1}{2}$ for the case of different home locations.

Future research is suggested in the following directions:

(1) Is the allocation minimax 2-TSP on a tree NP-complete in the strong sense or there exists a pseudopolynomial algorithm?

(2) Try to find polynomial heuristics for the problem with lower worst-case relative errors.

(3) Find polynomial heuristics for the case of p servers, $p \geqslant 3$, with non-trivial worst-case analysis.

References

[1] G.N. Frederickson, M.S. Hecht and C.E. Kim, Approximation algorithms for some routing problems, SIAM J. Comput. 7 (1978) 178–193.

[2] M. Garey and D. Johnson, Computers and Intractability (Freeman, San Francisco, 1979).