Available online at www.sciencedirect.com
SCIENCE DIRECT®
Discrete Applied Mathematics 140 (2004) 103–114
DISCRETE APPLIED MATHEMATICS
www.elsevier.com/locate/dam

A faster 2-approximation algorithm for the minmax p-traveling salesmen problem on a tree

Hiroshi Nagamochi[a], Kohei Okada[b]

[a]*Department of Information and Computer Sciences, Toyohashi University of Technology, Toyohashi, Aichi 441-8580, Japan*
[b]*Matsushita Electric Industrial Co., Ltd., Kadoma 1006, Osaka 571-8501, Japan*

Received 25 June 2002; received in revised form 25 June 2003; accepted 30 June 2003

Abstract

Given an edge-weighted tree T and an integer $p \geqslant 1$, the minmax p-traveling salesmen problem on a tree T asks to find p tours such that the union of the p tours covers all the vertices. The objective is to minimize the maximum of length of the p tours. It is known that the problem is NP-hard and has a $(2-2/(p+1))$-approximation algorithm which runs in $O(p^{p-1}n^{p-1})$ time for a tree with n vertices. In this paper, we consider an extension of the problem in which the set of vertices to be covered now can be chosen as a subset S of vertices and weights to process vertices in S are also introduced in the tour length. For the problem, we give an approximation algorithm that has the same performance guarantee, but runs in $O((p-1)! \cdot n)$ time.

Keywords: Traveling salesman problem; Vehicle routing; Tree; Graph partition; NP-hard; Approximation algorithm

1. Introduction

Given a graph, the p-traveling salesmen problem (p-TSP) asks to find a set of p tours that cover all vertices in the graph, minimizing a given objective function. This type of problems arises in many applications such as the multi-vehicle scheduling problem [7]. Graphs are restricted as paths or trees in some applications such as the task sequencing problem [6], the delivery scheduling by ships on a shoreline [13] and the scheduling of automated guided vehicles. Thus the 1-TSP or p-TSP on these graphs and related problems have been studied extensively (e.g., [1,2,8–14]).

E-mail address: naga@ics.tut.ac.jp (H. Nagamochi).

0166-218X/$ - see front matter
doi:10.1016/j.dam.2003.06.001

In this paper, we consider the minmax p-traveling salesmen problem (the minmax p-TSP for short) on an edge-weighted tree. There are p identical service units (called vehicles), initially situated at some points of the tree (called home locations). A set DP of demand points is prescribed (where a demand point is a vertex or an intermediate position on an edge); each demand point in DP must be served by being visited by at least one vehicle. Each vehicle must return back to its home location before finishing its service. The objective is to minimize the maximum length of the tours.

Averbakh and Berman [3] studied the minmax 2-TSP with given home locations on a tree, and gave a $\frac{4}{3}$-approximation algorithm for the case of equal home locations and $\frac{3}{2}$-approximation for the case of different home locations. Averbakh and Berman [4] considered the minmax p-TSP, where home locations of vehicles are not given in advance and should be chosen along with tours for vehicles. More formally the problem is described as follows. Let HL denote a set of points that are allowed to be chosen as home locations of vehicles. We denote by $HL = V$ (resp., $HL = E$) if home locations can be chosen from the set of all vertices (resp., the set of all vertices and all points on edges). Similarly $DP = V$ (resp., $DP = E$) means that a set of demand points is given as a set of all vertices in the tree (resp., the set of all vertices and all points on edges). Thus, there are four possible problems by the choice of $HL = V$ or E, and $DP = V$ or E. But the case with $HL = E$ and $DP = V$ needs no special consideration since it is easily reduced to the problem with $HL = V$ and $DP = V$. Averbakh and Berman [4] presented a $(2-2/(p+1))$-approximation algorithm for each of these three problems (excluding the reducible one), which are all shown to be NP-hard. The run time of their algorithms in a tree with n vertices is $O(p+n)$ to the problem with $HL = E$, and $O(p^{p-1}n^{p-1})$ to the problem with $HL = V$ and $DP = V$.

In this paper, we reduce the time bound to the problem with $HL = V$ and $DP = V$, which is currently extremely high complexity. We first extend the minmax p-TSP with $HL = V$ and $DP = V$ in such a way that a nonnegative vertex-weight is introduced and DP is allowed to be a subset of vertices. In what follows, we give a formal definition of our problem. Since an edge in a tour is traversed exactly twice, the problem will be described as a problem of finding a set of p subtrees that covers S.

Let T be a tree with edges weighted by nonnegative reals, where the weight of an edge e is denoted by $w(e)$. The vertex set and the edge set of a tree T are denoted by $V(T)$ and $E(T)$, respectively. Let S be a nonempty subset of $V(T)$, where each vertex u in S has a nonnegative weight $h(u)$; for convenience we use $h(v) = 0$ for vertices $v \in V(T) - S$. A vertex with degree 1 is called a *leaf*. A connected subgraph T' of T is called a *subtree* of T, and we denote this by $T' \subseteq T$. For a set $\mathscr{T}$ of subtrees $T_i \subseteq T$, $i = 1, 2, \ldots, h$, we denote by $V(\mathscr{T})$ and $E(\mathscr{T})$ the sets of vertices and edges in subtrees in $\mathscr{T}$, respectively. The sum of vertex weights (resp., edge weights) in a subtree T' is denoted by $H(T')$ (resp., $W(T')$). Let $WH(T')$ denote $W(T') + H(T')$. For a subset $X \subseteq V(T)$ of vertices, let $T - X$ denote the graph obtained from T by removing the vertices in X together with all adjacent edges, and let $T\langle X\rangle$ denote the minimal subtree of T that contains X (where the leaves of $T\langle X\rangle$ will be vertices in X). For a subset $S \subseteq V(T)$, a collection $\mathscr{S}$ of subsets $S_1, S_2, \ldots, S_k$ (not necessarily nonempty) of S is called a *partition* of S if their union is S, and is called a *p-partition* of S if $|\mathscr{S}| = p$. The cost of a subset $S_i \in \mathscr{S}$ is defined by $cost(S_i) = W(T\langle S_i\rangle) + H(S_i)$ (implying the travel cost of $T\langle S_i \rangle$ plus the processing cost of S_i). The cost of a partition $\mathscr{S}$ is defined by $cost(\mathscr{S}) = \max_{S_i \in \mathscr{S}} cost(S_i)$. Then our problem is described as follows:

Minmax p-subtrees cover problem (MSCP for short):
Input: A tree T, a subset $S \subseteq V(T)$ and an integer p with $1 \leqslant p \leqslant |S|$.
Feasible solution: A p-partition $\mathscr{S} = \{S_1, S_2, \ldots, S_p\}$ of S.
Goal: Minimize $cost(\mathscr{S})$.

The result of the minmax p-TSP by Averbakh and Berman [4] implies that a $(2 - 2/(p+1))$-approximate solution to the MSCP can be found in $O(p^{p-1}n^{p-1})$ time in the special case of $S = V(T)$ and $h(v) = 0$, $v \in V(T)$. In this paper, partly following their approach, we present an algorithm that finds a $(2 - 2/(p+1))$-approximate solution to the MSCP in the time complexity of $O((p-1)! \cdot n)$, which is a linear time for a fixed p. In our algorithm, a new idea is exploited to design an $O((p-1)! \cdot n)$ time algorithm for computing a lower bound on an optimal solution.

The paper is organized as follows. In Section 2, we give a procedure for decomposing a tree into a set of edge-disjoint subtrees such that the weight of each subtree is bounded by a certain value. In Section 3, we introduce a lower bound on the optimal value of the MSCP by modifying the one introduced by Averbakh and Berman [4]. We then design a new and fast algorithm for computing the lower bound. Based on the results in Sections 2 and 3, we in Section 4 present an approximation algorithm for the MSCP. As an application, we in Section 5 use our algorithm to design algorithms for the multi-vehicle scheduling problem in trees. In Section 6, we describe some concluding remarks.

2. Decomposing a tree

In this section, we prove the next lemma, which will be the basis of our approximation algorithm given in Section 4. Let $h_{\max}$ denote $\max_{v \in V(T)} h(v)$.

Lemma 1. *For any instance (T, S, p) of the MSCP, there exists a p-partition $\mathscr{S}$ of S with*

$$cost(\mathscr{S}) \leqslant \max\left\{\left(2 - \frac{2}{p+1}\right) \cdot \frac{WH(T)}{p},\ h_{\max}\right\}$$

such that for any two $S_i, S_j \in \mathscr{S}$, subtrees $T\langle S_i \rangle$ and $T\langle S_j \rangle$ are edge-disjoint. Such an $\mathscr{S}$ can be obtained in $O(p+n)$ time. □

The lemma in the case that $S = V(T)$ and $h(u) = 0$, $u \in S$ has been obtained by Averbakh and Berman [4]. Based on their idea of chopping a maximally eligible subtree off the current tree, we in the below give a procedure for constructing such an $\mathscr{S}$ in the lemma. (We remark that unlike their procedure [4] ours can take a vertex weight h into account, treat a given tree as a rooted one, and split no edge into two edges with fractional weights; a distinct solution $\mathscr{S}$ may be produced by ours even for the case that $S = V(T)$ and $h(u) = 0$, $u \in V(T)$.)

In a rooted tree T, a subtree of T rooted at a vertex v is denoted by T_v, and an edge $e=(u,v)$ such that v is a child of u is called the *parent-edge of* v or a *child-edge* of u. For a real $\delta \geqslant 0$, a tree T is called a δ*-pseudo-star* centered at a vertex u if T can be rooted at u so that $WH(T_v)+w(e) \leqslant \delta$ holds for all child-edges $e=(u,v)$ of the root u.

Lemma 2. *Let T' be a tree with an edge weight w and a vertex weight h. If T' is a Δ- pseudo-star centered at a vertex u for some $\Delta > 0$, then for $p' = \max\{1, \lfloor WH(T')/\Delta \rfloor\}$, there is a p'-partition $\mathscr{S} = \{S_1, S_2, \ldots, S_{p'}\}$ of $V(T')$ such that $cost(\mathscr{S}) \leqslant \max\{2\Delta, h(u)\}$.*

Proof. If $WH(T') < 2\Delta$ then $p'=1$ and the lemma holds. Assume that $WH(T') \geqslant 2\Delta$ and $p' \geqslant 2$. Let $V_0=\{u\}$ and $a(V_0)=h(u)$. For child-edges $e_i=(u,v_i)$, $i=1,2,\ldots,k$ of u, let $V_i = V(T_{v_i})$ and $a(V_i) = WH(T_{v_i}) + w(e_i)$. Let $A=\{V_0, V_1, \ldots, V_k\}$. Since $a(V_i) \leqslant \Delta$ $(1 \leqslant i \leqslant k)$ by T' being a Δ-pseudo-star, we can repeat choosing a maximal subset $B \subseteq A$ such that $\Delta \leqslant \sum_{V_i \in B} a(V_i) \leqslant \max\{2\Delta, h(u)\}$ and setting $A := A - B$ until $A=\emptyset$ holds or such B is obtained $p'-1$ times. Let $B_1, B_2, \ldots, B_t$ be the chosen subsets, and let $S_j = \bigcup_{V_i \in B_j} V_i$ for $j=1,2,\ldots,t$, where $cost(S_j) \leqslant \max\{2\Delta, h(u)\}$ holds by construction. If $A=\emptyset$ holds when the procedure halts, then $\{S_1, S_2, \ldots, S_t\}$ (together with $p'-t$ empty subsets if $t < p'$) is a desired p'-partition. Consider the latter case (i.e., $t = p' - 1$). Let $S_{p'} = \bigcup_{V_i \in A} V_i$ for the resulting set A. Hence, $cost(S_{p'})$ is at most $WH(T') - (p' - 1)\Delta \leqslant WH(T') - (\lfloor WH(T')/\Delta \rfloor - 1)\Delta < WH(T') - (WH(T')/\Delta - 2)\Delta = 2\Delta$. Thus, $\{S_1, S_2, \ldots, S_{p'-1}, S_{p'}\}$ satisfies the lemma. □

For a rooted tree T and a given $\Delta \geqslant 0$, we call an edge (u,v) between a vertex u and its child v *light* if $WH(T_v) + w(e) \leqslant \Delta$ and *heavy* otherwise. We call a vertex v in T *admissible* if its parent-edge (if any) is heavy and its child-edges (if any) are all light. Note that there exists at least one admissible vertex v in any rooted tree T, and that T_v for an admissible vertex v is a Δ-pseudo-star. We now give a procedure for decomposing a given tree T rooted at r. For a given tree T rooted at r and $p \geqslant 2$, we set $\Delta := [1/(p+1)]WH(T)$ (where $WH(T) > \Delta$), $T' := T$ and $i := 1$, and apply the next.

DECOMPOSE(T', i)
case-1) If $WH(T') \leqslant 2\Delta$, then halt outputting $T_i := T'$ and $\alpha_i := WH(T')(>\Delta)$.
case-2) Otherwise $(WH(T') > 2\Delta)$, find an admissible vertex v closest to root r in T'.
  (i) If $WH(T') - WH(T_v) \leqslant \Delta$ (i.e., the current T' is a Δ-pseudo-star centered at v), then halt outputting $T_i := T'$ and $\alpha_i := WH(T')(>\Delta)$.
  (ii) Otherwise $(WH(T') - WH(T_v) > \Delta)$, let e_v be the parent-edge of v (where e_v is heavy in T').
    (a) If $WH(T') - WH(T_v) - w(e_v) \leqslant \Delta$, then halt outputting
    $T_i := T' - V(T_v)$, $\alpha_i := \Delta(\geqslant WH(T_i))$,
    $T_{i+1} := T_v$ and $\alpha_{i+1} := WH(T') - \Delta$ $(> \max\{\Delta, WH(T_v)\})$.
    (b) Otherwise $(WH(T') - WH(T_v) - w(e_v) > \Delta)$,output
    $T_i := T_v$ and $\alpha_i := WH(T_v) + w(e_v)(>\Delta)$, and DECOMPOSE$(T', i)$
    after setting $i := i+1$ and $T' := T' - V(T_v)$
    (note that $WH(T') > \Delta$).

Let $T_1, T_2, \ldots, T_k$ and $\alpha_1, \alpha_2, \ldots, \alpha_k$ be the subtrees and the values associated with them computed by the procedure, where any two of these subtrees are vertex-disjoint. Moreover, T_i is a Δ-pseudo-star unless $WH(T_i) \leqslant 2\Delta$. Observe that α_i satisfies

$$\sum_{1 \leqslant i \leqslant k} \alpha_i = WH(T), \quad \text{and} \quad \alpha_i \geqslant \max\{\Delta, WH(T_i)\} \text{ for } i = 1, 2, \ldots, k.$$

In particular, $\alpha_k > \Delta$. We further decompose each T_i into an adequate number $p(i)$ of subtrees. The number $p(i)$ is given as follows.

Lemma 3. *Let* $p(i) = \lfloor \alpha_i / \Delta \rfloor$, $i = 1, 2, \ldots, k$. *Then we have* $\max\{1, \lfloor WH(T_i)/\Delta \rfloor\} \leqslant p(i)$, $i = 1, 2, \ldots, k$, *and* $\sum_{i=1}^{k} p(i) \leqslant p$.

Proof. By $\alpha_i \geqslant \max\{\Delta, WH(T_i)\}$, $p(i) \geqslant \max\{1, \lfloor WH(T_i)/\Delta \rfloor\}$. By noting $\alpha_k > \Delta$, we obtain $\sum_{i=1}^{k} p(i) < \sum_{i=1}^{k} \alpha_i/\Delta \leqslant (1/\Delta) WH(T) = p + 1$. □

For each T_i, we find a $p(i)$-partition $\mathscr{S}_i$ of $V(T_i)$ such that $cost(\mathscr{S}_i) \leqslant \max\{2\Delta, h_{\max}\}$. Such an $\mathscr{S}_i$ can be found by Lemmas 2 and 3 if T_i is a Δ-pseudo-star; $\mathscr{S}_i$ can be chosen as any $p(i)$-partition of $V(T_i)$ otherwise (since $WH(T_i) \leqslant 2\Delta$ holds). Therefore, the union $\mathscr{S}$ of the resulting partitions $\mathscr{S}_1, \mathscr{S}_2, \ldots, \mathscr{S}_k$ (adding empty subsets if necessary) gives a p-partition of $V(T)$ and satisfies $cost(\mathscr{S}) \leqslant \max\{2\Delta, h_{\max}\} = \{2p/(p+1) \cdot WH(T)/p, h_{\max}\}$. Furthermore, by construction, for any two $S', S'' \in \mathscr{S}$, the subtrees $T\langle S' \rangle$ and $T\langle S'' \rangle$ are edge-disjoint. Thus, $\mathscr{S}$ satisfies the conditions in Lemma 1. It is not difficult to see that $\mathscr{S}$ can be obtained in $O(p+n)$ time.

It should be noted that Lemma 1 does not immediately imply the $(2 - 2/(p+1))$-approximability of the MSCP since $WH(T)/p$ is not a lower bound in general (i.e., some edge may not be used in any subtree $T\langle S_i \rangle$ in an optimal solution $\{S_1, S_2, \ldots, S_p\}$).

3. Computing a lower bound

In this section, we give an algorithm for computing a lower bound on the optimal value to the MSCP. The lower bound is a modification of the one by Averbakh and Berman [4], but the time to compute the lower bound will be reduced significantly by our new idea below. We start with a simple observation of a lower bound on the optimal value.

Lemma 4. *For an instance* (T, S, p) *of the MSCP*, $\max\{WH(T)/p, h_{\max}\}$ *is a lower bound on the optimal value provided that each edge is contained in some subtree* $T\langle S_i \rangle$ *for an optimal solution* $\mathscr{S}^* = \{S_1, S_2, \ldots, S_p\}$.

For a p-partition $\mathscr{S} = \{S_1, S_2, \ldots, S_p\}$ of S, the subtrees $T\langle S_i \rangle$, $i = 1, 2, \ldots, p$ are uniquely determined, where possibly $V(T\langle S_i \rangle) \cap V(T\langle S_j \rangle) \neq \phi$ for some i and j. There may be an edge $e \in E(T)$ that is not contained in any subtree $T\langle S_i \rangle$. Also there may be a vertex $v \in V(T) - \bigcup_{S_i \in \mathscr{S}} V(T\langle S_i \rangle)$. We call such edges and vertices *unused*. By removing all unused edges from T, we have several vertex-disjoint subtrees of T, each of which is given by either the subtree $T\langle S_{i_1} \cup S_{i_2} \cup \cdots \cup S_{i_k}\rangle$ for some union of some subsets in $\mathscr{S}$ or a subtree consisting of a single unused vertex. We denote by $\mathscr{T}\langle\mathscr{S}\rangle$ the set of such subtrees with the form of $T\langle S_{i_1} \cup S_{i_2} \cup \cdots \cup S_{i_k}\rangle$. For each $T' \in \mathscr{T}\langle\mathscr{S}\rangle$, let $p(T')$ denote the number of subsets $S_i \in \mathscr{S}$ such that $S_i \subseteq V(T')$.

Let $\mathscr{S}^*$ be an optimal p-partition of S to the problem. With the set $\mathscr{T} = \mathscr{T}\langle\mathscr{S}^*\rangle$ of subtrees and the set $\{p(T') \mid T' \in \mathscr{T}\langle\mathscr{S}^*\rangle\}$ of numbers (but without knowing $\mathscr{S}^*$ itself), we can construct a $(2 - 2/(p+1))$-approximation solution by computing a $p(T')$-partition of $V(T')$ for each subtree $T' \in \mathscr{T}$ with number $p(T')$ by Lemma 1. Hence, by generating all possible sets $\mathscr{T}$ of subtrees and sets $\{p(T')\}$ of numbers and by computing such a solution for each pair of $\mathscr{T}$ and $\{p(T')\}$, we can obtain a $(2 - 2/(p+1))$-approximation solution. This, however, takes time complexity of $O(p^{p-1}n^{p-1})$, as already observed in [4]. To avoid this brute-force search, we in this section introduce a new idea for computing a lower bound. For this, we define new notions such as valued subtree collections, rational edges, and rational points.

For a tree T, a subset $S \subseteq V(T)$ and an integer $p \leqslant |S|$, we define a *valued subtree collection* of (T, S, p) as a set $\mathscr{T}$ of vertex-disjoint subtrees $T_1, T_2, \ldots, T_k \subseteq T$ such that $S \subseteq V(\mathscr{T})$ holds and a positive integer p_{T_i} with $\sum_{T_i \in \mathscr{T}} p_{T_i} = p$ is associated with each T_i. We define

$$\lambda(\mathscr{T}) = \max\left\{ \frac{WH(T_i)}{p_{T_i}} \;\middle|\; T_i \in \mathscr{T} \right\}$$

and

$$\lambda^*(T, S, p) = \min\{\lambda(\mathscr{T}) \mid \text{all valued subtree collections } \mathscr{T} \text{ of } (T, S, p)\}.$$

Lemma 5. *$\lambda^*(T, S, p)$ is a lower bound on the optimal value to the MSCP.*

Proof. Let $\mathscr{S}^*$ be an optimal p-partition of S, and $\mathscr{T} = \mathscr{T}\langle\mathscr{S}^*\rangle = \{T_1, T_2, \ldots, T_q\}$ with $\{p(T') \mid T' \in \mathscr{T}\langle\mathscr{S}^*\rangle\}$ be the valued subtree collection induced by $\mathscr{S}^*$. Since all edges in $E(T_i)$ are used in T_i, it holds $WH(T_i)/p(T_i) \leqslant cost(S_j)$ for subsets $S_j \in \mathscr{S}^*$ with $S_j \subseteq V(T_i)$. Thus $\lambda(\mathscr{T}) \leqslant cost(\mathscr{S}^*)$. On the other hand, we have $\lambda^*(T, S, p) \leqslant \lambda(\mathscr{T})$ since $\mathscr{T}$ with $\{p(T')\}$ is a valued subtree collection. Therefore $\lambda^*(T, S, p) \leqslant cost(\mathscr{S}^*)$. $\square$

A recursive form of $\lambda^*(T, S, p)$ is given by

$$\lambda^*(T, S, p) = \min\left\{ \frac{WH(T)}{p}, \right.$$
$$\min\{\max\{\lambda^*(T_u, S \cap V(T_u), p'), \lambda^*(T_v, S \cap V(T_v), p'')\}$$
$$\left. |\, e = (u, v) \in E(T), p' + p'' = p\} \right\},$$

where T_u and T_v denote the subtrees obtained from T by removing $e = (u, v)$. Computing $\lambda(T, S, p)$ by the recursion still takes $O(p^{p-1}n^{p-1})$ time. In this section, we prove the next result.

Theorem 6. *For a tree T, a subset $S \subseteq V(T)$ and an integer $p \leqslant |S|$, $\lambda^*(T,S,p)$ and a valued subtree collection $\mathscr{T}$ of (T,S,p) with $\lambda(\mathscr{T}) = \lambda^*(T,S,p)$ can be computed in* $O((p-1)! \cdot n)$ *time.* $\square$

To prove the theorem, an edge $e=(u,v)$ in a tree T may be treated as a segment that consists of infinitely many *points*. A point x in e is specified by the distance from one of the end vertices of e. The point with distance t ($0 \leqslant t \leqslant w(e)$) from u on e is denoted by $x_{e,u}(t)$.

An edge $e=(u,v)$ in a tree T is called a *p-rational edge* of T if the subtrees $T_u, T_v \subseteq T$ obtained from T by removing e satisfy

$$WH(T_u) \leqslant p' \frac{WH(T)}{p} \quad \text{and} \quad WH(T_v) \leqslant p'' \frac{WH(T)}{p}$$

for some positive integers p' and p'' with $p'+p''=p$, where a point $x_{e,u}(t)$ in e with $WH(T_u)+t = p'WH(T)/p$ is called a *p-rational point* of T (note that a p-rational edge may contain more than one p-rational point).

Lemma 7. *Let T be a tree, and S be a nonempty subset of $V(T)$. Assume that $T = T\langle S\rangle$. Let $\mathscr{T}$ be a valued subtree collection of (T,S,p). Then if $\lambda(\mathscr{T}) < WH(T)/p$, then $E(T)-E(\mathscr{T})$ contains at least one p-rational edge.*

Proof. A vertex or edge not contained in any $T' \in \mathscr{T}$ is called *unused*. Hence, $E(T)-E(\mathscr{T})$ is the set of unused edges. By $\lambda(\mathscr{T}) < WH(T)/p$, $|E(T)-E(\mathscr{T})| \geqslant 1$. We show the lemma by induction on $k=|E(T)-E(\mathscr{T})|$. We consider T as a rooted tree by choosing an arbitrary vertex as the root. Let T^u be a subtree in $\mathscr{T}$ which is the farthest one from the root, and $e=(u,v) \in E(T)-E(\mathscr{T})$ be the edge connecting T^u and $T-V(T^u)$ in T, where $u \in V(T^u)$ is assumed without loss of generality. Then let T^v be the subtree containing v among the subtrees obtained from T by removing $E(T)-E(\mathscr{T})$, where T^v may not belong to $\mathscr{T}$ (i.e., it may consist of a single unused vertex).

(i) Let $k=|E(T)-E(\mathscr{T})|=1$. By $T=T\langle S\rangle$, T^v also belongs to $\mathscr{T}$. By $\lambda(\mathscr{T}) < WH(T)/p$, we have $WH(T^v) < p_{T^v} \cdot WH(T)/p$, i.e., $WH(T) - WH(T^u) - w(e) < (p - p_{T^u})WH(T)/p$ by $WH(T) = WH(T^u) + WH(T^v) - w(e)$ and $p = p_{T^u} + p_{T^v}$. From this,

$$\frac{WH(T)}{p} < \frac{WH(T^u)+w(e)}{p_{T^u}}.$$

With this and $WH(T^u)/p_{T^u} < WH(T)/p$, there is a real $t \in (0, w(e))$ such that

$$\frac{WH(T)}{p} = \frac{WH(T^u)+t}{p_{T^u}}$$

implying that e is a p-rational edge.

(ii) Let $k>1$ and assume that the lemma holds for $k-1$. Now $WH(T^u)/p_{T^u} < WH(T)/p$. If

$$\frac{WH(T)}{p} \leqslant \frac{WH(T^u)+w(e)}{p_{T^u}}$$

then there is a real $t \in (0, w(e)]$ such that

$$\frac{WH(T)}{p} = \frac{WH(T^u) + t}{p_{T^u}}$$

implying that e is a p-rational edge. Then assume

$$\frac{WH(T^u) + w(e)}{p_{T^u}} < \frac{WH(T)}{p}. \tag{1}$$

We first assume that $T^v \notin \mathscr{T}$ (i.e., v is an unused vertex and $h(v) = 0$ by $v \notin S$). Construct the subtree $\hat{T}^u = T\langle V(T^u) \cup \{v\}\rangle$ with the associated integer $p_{\hat{T}^u} = p_{T^u}$. Then it satisfies

$$\frac{WH(\hat{T}^u)}{p_{\hat{T}^u}} = \frac{WH(T^u) + w(e)}{p_{T^u}} < \frac{WH(T)}{p}.$$

Note that $\mathscr{T}' = (\mathscr{T} - \{T^u\}) \cup \{\hat{T}^u\}$ still satisfies the assumption of the lemma, having one less unused edge, and by the induction hypothesis, it has a p-rational edge $e' \in E(T) - E(\mathscr{T}')(\subset E(T) - E(\mathscr{T}))$.

We next assume that $T^v \in \mathscr{T}$. Construct the subtree $T^{uv} = T\langle V(T^u) \cup V(T^v)\rangle$ with the associated integer $p_{T^{uv}} = p_{T^u} + p_{T^v}$. With (1) and $WH(T^v)/p_{T^v} < WH(T)/p$, T_{uv} satisfies

$$\frac{WH(T^{uv})}{p_{T^{uv}}} < \frac{WH(T)}{p}.$$

Since $\mathscr{T}' = (\mathscr{T} - \{T^u, T^v\}) \cup \{T^{uv}\}$ satisfies the assumption of the lemma, we have a p-rational edge $e' \in E(T) - E(\mathscr{T}')(\subset E(T) - E(\mathscr{T}))$ by the induction hypothesis. □

Lemma 8. *For a tree T and an integer $p \geqslant 2$, the number of p-rational points in T is at most $p-1$, and the number of p-rational edges in T is at most* $\min\{p-1, n-1\}$.

Proof. It suffices to show the former part of the lemma. The case of $p = 2$ is trivial. We can prove the case of $p > 2$ by an inductive proof. Let T be rooted at some vertex r, and choose a vertex v closest to r such that T_v contains no p-rational edge. Thus, the edge $e = (u, v)$ joining v and its parent u contains a p-rational point $x_{e,u}(t)$; we choose x as the one closest to v. Construct the tree T' obtained from T by contracting vertices in T_v into a single vertex v^* and by setting weight $h(v^*) = 0$ for the vertex v^* and weight $w(e^*) = t$ for the adjacent edge $e^* = (u, v^*)$. Then $WH(T') = WH(T) - (1/p)WH(T)$ holds. Thus, any p-rational point in T (except for the $x_{e,u}(t)$) is a $(p-1)$-rational point in T'. By the induction hypothesis, the number of such $(p-1)$-rational points is at most $p-2$. Thus, T has at most $p-1$ p-rational points. □

Lemma 9. *All p-rational edges and p-rational points in a tree T can be found in* $O(p+n)$ *time.*

Proof. Choose an arbitrary vertex $r \in V(T)$ as the root of T. All $WH(T_u)$, $u \in V(T)$ can be computed in $O(n)$ time by computing them in a bottom-up manner. Let $\Delta =$

$WH(T)/p$. For each edge $e=(u,v)$ between a vertex u and its parent v, compute the maximum integer $k_e \geqslant 0$ such that

$$\left(\left\lfloor \frac{WH(T_u)}{\Delta} \right\rfloor + k_e\right)\Delta \leqslant WH(T_u) + w(e).$$

Then e is a p-rational edge if and only if $k_e \geqslant 1$ or $\lfloor WH(T_u)/\Delta \rfloor \Delta = WH(T_u)$ holds. Moreover, e contains k_e p-rational points $x_{e,u}((\lfloor WH(T_u)/\Delta\rfloor + j)\Delta - WH(T_u))$, $j = 0,1,\dots,k_e$. It is easy to see that each edge e can be processed in $O(k_e+1)$ time. Thus, the entire running time is $O(\sum_{e\in E(T)}(k_e+1)) = O(p+n)$ by Lemma 8. $\square$

Let $RP(T,p)$ denote the set of all p-rational points in T. For a point $x \in RP(T,p)$, let $e_x=(u_x,v_x)$ be the p-rational edge containing x, ℓ_x be the distance of x from u_x, and T_{u_x} and T_{v_x} be the subtrees obtained from T by removing e_x. Let p_{u_x} and p_{v_x} be the integers assigned to T_{u_x} and T_{v_x}, i.e.,

$$p_{u_x} = \frac{(WH(T_{u_x})+\ell_x)p}{WH(T)} \quad \text{and} \quad p_{v_x} = \frac{(WH(T_{v_x})+w(e_x)-\ell_x)p}{WH(T)} = p - p_{u_x},$$

respectively. Assume that $T=T\langle S\rangle$ (otherwise we can reset T by $T\langle S\rangle$ without losing the optimality of the given instance (T,S,p)). Then let $S_{u_x}=S\cap V(T_{u_x})$, $S_{v_x}=S\cap V(T_{v_x})$, $T'_{u_x}=T\langle S_{u_x}\rangle$ and $T'_{v_x}=T\langle S_{v_x}\rangle$. By Lemma 7, another recursive form of $\lambda^*(T,S,p)$ is given by

$$\lambda^*(T,S,p) = \begin{cases} \dfrac{WH(T)}{p} & \text{if } RP(T,p)=\emptyset, \\ \min\left\{\max\{\lambda^*(T'_{u_x},S_{u_x},p_{u_x}),\lambda^*(T'_{v_x},S_{v_x},p_{v_x})\}\right. & \\ \quad \left.| x\in RP(T,p)\right\} & \text{otherwise.} \end{cases}$$

We consider the time complexity for computing $\lambda^*(T,S,p)$ in this form. It is a simple matter to see that the total number of recursive calls is bounded by $(p-1)!$ from above. For each call $\lambda^*(T'_{u_x},S_{u_x},p_{u_x})$, we can compute the set of p_{u_x}-rational points in $O(n)$ time by Lemma 9. Thus the time to compute $\lambda^*(T,S,p)$ is $O((p-1)!n)$. Also a valued subtree collection $\mathscr{T}$ of (T,S,p) attaining the $\lambda^*(T,S,p)$ can be retrieved in the same time complexity, proving Theorem 6.

4. Approximation algorithm

Based on Theorem 6, we obtain the following algorithm for computing a p-partition $\mathscr{S}$ to an instance (T,S,p) of the MSCP.

APPROX(T,S,p)

Step 1. Set $T := T\langle S\rangle$, and compute a valued subtree collection $\mathscr{T}$ of (T,S,p) attaining $\lambda(\mathscr{T}) = \lambda^*(T,S,p)$.

Step 2. For each subtree $T'\in\mathscr{T}$ and the associated integer $p_{T'}$, construct an instance $(T', S\cap V(T'), p_{T'})$ of the MSCP, and compute a $p_{T'}$-partition $\mathscr{S}_{T'}$ of $S\cap V(T')$ such that

$$cost(\mathscr{S}_{T'}) \leqslant \max\left\{\left(2-\frac{2}{p_{T'}+1}\right)\frac{WH(T')}{p_{T'}}, \max_{v\in V(T')} h(v)\right\}.$$

Step 3. Output the union $\cup_{T' \in \mathcal{T}} \mathcal{S}_{T'}$ as a solution to the instance (T, S, p).

Step 1 can be executed in $O((p-1)!n)$ time by Theorem 6. By Lemma 1, we can obtain in $O(p_{T'}+n)$ time such a solution $\mathcal{S}_{T'}$ in Step 2. For a solution $\mathcal{S}=\cup_{T' \in \mathcal{T}} \mathcal{S}_{T'}$ output by APPROX(T, S, p), it holds $cost(\mathcal{S}) \leqslant \max\{(2-2/(p+1))\lambda^*(T, S, p), h_{\max}\}$. Therefore the next is established.

Theorem 10. *For an instance (T, S, p) of the MSCP, there exists a p-partition of S such that $\mathrm{cost}(\mathcal{S}) \leqslant \max\{(2-2/(p+1))\lambda^*(T, S, p), h_{\max}\}$ holds and subtrees $T\langle S'\rangle$ and $T\langle S''\rangle$ are edge-disjoint for any two distinct $S', S'' \in \mathcal{S}$. Such an $\mathcal{S}$ can be obtained in $O((p-1)!n)$ time.*

Since the subtrees $T\langle S_i\rangle$, $S_i \in \mathcal{S}$ in the theorem are *edge-disjoint*, we can enjoy the same approximability even when the condition that $T\langle S_i\rangle$, $S_i \in \mathcal{S}$ are edge-disjoint is additionally imposed on the MSCP. We remark that the vertex-disjoint version of the MSCP with $S = V(T)$ (i.e., the case where $T\langle S_i\rangle$, $S_i \in \mathcal{S}$ are required to be *vertex-disjoint*) has been studied as a tree partition problem, for which an $O(p^3 n)$ time algorithm has been obtained [5]. We close this section by showing that there is a tight example to algorithm APPROX.

Example 1. Let T be a tree with vertex set $V(T)=\{u_1, u_2, \ldots, u_p, u', v', v_1, v_2, \ldots, v_p\}$ and edge set $E(T) = \{(u_i, u') \mid i = 1, 2, \ldots, p\} \cup \{(u', v')\} \cup \{(v', v_i) \mid i = 1, 2, \ldots, p\}$, where for a positive ε edges are weighted by $w(e)=1$ for $e=(u_i, u')$, $i=1, 2, \ldots, p$, $w(e)=1-2\varepsilon$ for $e=(u', v')$, and $w(e)=\varepsilon$ for $e=(v', v_i)$, $i=1, 2, \ldots, p$. Let $S=V(T)$ and $h(u)=0$ for all $u \in V(T)$. Then $\lambda^*(T, S, p) \leqslant WH(T)/p=(p+1+(p-2)\varepsilon)/p$. An optimal solution to this instance of the MSCP is $\mathcal{S}^*=\{S_i=\{u_i, v_i\} \mid i=1, 2, \ldots, p\}$ with $cost(\mathcal{S}^*)=2-\varepsilon$. Thus $cost(\mathcal{S}^*)/\lambda^*(T, S, p) \geqslant (2-\varepsilon)/[(p+1+(p-2)\varepsilon)/p]$, which approaches $(2-2/(p+1))$ as $\varepsilon \to 0$.

This example shows that the approximation guarantee $(2-2/(p+1))$ in Theorem 10 cannot be improved unless the current lower bound $\lambda^*(T, S, p)$ is strengthened. However, we are not aware of any example with the same gap which has an optimal solution $\mathcal{S}^* = \{S_1, S_2, \ldots, S_p\}$ such that any two subtrees $T\langle S_i\rangle$ and $T\langle S_j\rangle$ are edge-disjoint.

5. Application to multi-vehicle scheduling in trees

As an application of Theorem 10, we in this section consider the following multi-vehicle scheduling problem in trees. There are p vehicles $(1 \leqslant p \leqslant n)$ in a tree T with n vertices. Travel time $w(u, v)$ $(=w(v, u))$ is associated with each edge $(u, v) \in E(T)$. Each vertex $v \in V(T)$ has a job (denoted also by v), and any job must be processed by exactly one vehicle. Each job v has release time $r(v)$ and handling time $h(v)$. That is, a vehicle cannot start processing job v before time $r(v)$, and it takes $h(v)$ time units to process job v (no interruption of the processing is allowed). A vehicle at vertex v may wait until time $r(v)$ to process job v, or move to other vertices without processing job v if it is more advantageous (in this case, the vehicle must come back to v later to process job v, or another vehicle must come to v to process it). The problem asks to find an optimal schedule of p vehicles that minimizes the makespan (i.e., the maximum completion time) of all the jobs. The initial location for each vehicle is also chosen so as to minimize the makespan. The problem is known to be NP-hard for any fixed $p \geqslant 2$. Let b be the number of leaves in T. It is shown [9] that the problem with fixed p and b admits a polynomial time approximation scheme that finds a $(1+\epsilon)$-approximation algorithm in time $O((1+2/\epsilon)n^p p^{p+1}(n^2+2^{1+2/\epsilon})(8pn^{2+b}/\epsilon)^{p(1+2/\epsilon)})$. Currently, it is left open whether there is a constant factor approximation algorithm that runs in polynomial in both p and b (such an approximation algorithm is known if a given graph is a path [8]). By using Theorem 10 we obtain a constant factor approximation algorithm that runs in $O((p-1)!n)$ time.

Corollary 1. *A* $(5-4/(p+1))$*-approximation solution to the multi-vehicle scheduling problem in a tree can be obtained in* $O((p-1)!n)$ *time. If each vehicle is required to return to the initial location, then a* $(3-2/(p+1))$*-approximation solution can be obtained in* $O((p-1)!n)$ *time.*

Proof. Let $r_{\max} = \max_{v\in V(T)} r(v)$ and $h_{\max} = \max_{v\in V(T)} h(v)$. Ignoring release times r, we find in $O((p-1)!n)$ time a p-partition $\mathscr{S}$ of $S=V(T)$ such that $cost(\mathscr{S}) \leqslant \max\{(2-2/(p+1))\lambda^*(T,S,p), h_{\max}\}$ by Theorem 10. For each $S_i \in \mathscr{S}$, the i-th vehicle can process all jobs in S_i along the subtree $T\langle S_i\rangle$ until time $r_{\max} + 2(W(T\langle S_i\rangle) + H(S_i)) \leqslant r_{\max} + \max\{2(2-2/(p+1))\lambda^*(T,S,p), 2h_{\max}\}$ at latest. Since $\max\{r_{\max}, \lambda^*(T,S,p), h_{\max}\}$ is a lower bound on the optimal value to the multi-vehicle scheduling problem, the scheduling along these subtrees is a $(5-4/(p+1))$-approximation solution.

We consider the case that each vehicle is required to return to the initial location. Then each edge must be traversed even number of times. For $S=V(T)$, we compute the lower bound $\lambda^*(T,S,p)$ after doubling all the edge weights in T. We then find a p-partition $\mathscr{S}$ of $S=V(T)$ such that $cost(\mathscr{S}) \leqslant \max\{(2-2/(p+1))\lambda^*(T,S,p), h_{\max}\}$. For each $S_i \in \mathscr{S}$, the ith vehicle can process all jobs in S_i along the subtree $T\langle S_i\rangle$ until time $r_{\max} + W(T\langle S_i\rangle) + H(S_i) \leqslant r_{\max} + \max\{(2-2/(p+1))\lambda^*(T,S,p), h_{\max}\}$ at latest. This gives a $(3-2/(p+1))$-approximation solution. □

6. Concluding remarks

We improved the time complexity of finding a $(2-2/(p+1))$-approximation solution to the MSCP by designing an efficient procedure of computing a lower bound on the optimal value. However, the time complexity is not polynomial in p. On the other hand the MSCP with $S=V(T)$ is known to be solvable if subtrees $T\langle S_i\rangle$ are required to be vertex-disjoint. It is left open to reduce the complexity of a $(2-2/(p+1))$-approximation algorithm to the one polynomial in both p and n and to obtain a better approximation algorithm, say, to the case where subtrees $T\langle S_i\rangle$ are required to be edge-disjoint.

Acknowledgements

This research was partially supported by the Scientific Grant-in-Aid from Ministry of Education, Culture, Sports, Science and Technology of Japan. The authors would like to thank anonymous referees for their valuable comments.

References

[1] T. Asano, N. Katoh, K. Kawashima, A new approximation algorithm for the capacitated vehicle routing problem on a tree, J. Combin. Optim. 5 (2001) 213–231.

[2] I. Averbakh, O. Berman, Sales-delivery man problems on treelike networks, Networks 25 (1995) 45–58.

[3] I. Averbakh, O. Berman, A heuristic with worst-case analysis for minmax routing of two traveling salesmen on a tree, Discrete Appl. Math. 68 (1996) 17–32.

[4] I. Averbakh, O. Berman, $(p-1)/(p+1)$-approximate algorithm for p-traveling salesmen problems on a tree with minmax objective, Discrete Appl. Math. 75 (1997) 201–216.

[5] R. Becker, Y. Perl, The shifting algorithm technique for the partitioning of trees, Discrete Appl. Math. 62 (1995) 15–34.

[6] J. Bruno, P. Downey, Complexity of task sequencing with deadlines, set-up times and changeover costs, SIAM J. Comput. 7 (1978) 393–581.

[7] J. Desrosiers, Y. Dumas, M.M. Solomon, F. Soumis, Time constrained routing and scheduling, in: M.O. Ball, T.L. Magnanti. C.L. Monma, G.L. Nemhauser (Eds.), Handbooks in Operations Research and Management Science Volume 8: Network Routing, North-Holland, Amsterdam, 1995, pp. 35–139.

[8] Y. Karuno, H. Nagamochi, A 2-approximation algorithm for the multi-vehicle scheduling on a path with release and handling times, in: Proc. ESA'01, Lecture Notes in Computer Science, Vol. 2161, Springer, Berlin, 2001, pp. 218–229.

[9] Y. Karuno, H. Nagamochi, A polynomial time approximation scheme for the multi-vehicle scheduling on a path with release and handling times, in: Proc. ISAAC'01, Lecture Notes in Computer Science, Vol. 2223, Springer, Berlin, 2001, pp. 36–47.

[10] Y. Karuno, H. Nagamochi, T. Ibaraki, Vehicle scheduling on a tree to minimize maximum lateness, J. Oper. Res. Soc. Japan 39 (1996) 345–355.

[11] Y. Karuno, H. Nagamochi, T. Ibaraki, Vehicle scheduling on a tree with release and handling times, Ann. Oper. Res. 69 (1997) 193–207.

[12] H. Nagamochi, K. Mochizuki, T. Ibaraki, Complexity of the single vehicle scheduling problem on graphs, Inform. Systems Oper. Res. 35 (1997) 256–276.

[13] H. Psaraftis, M. Solomon, T. Magnanti, T. Kim, Routing and scheduling on a shoreline with release times, Manage. Sci. 36 (1990) 212–223.

[14] J.N. Tsitsiklis, Special cases of traveling salesman and repairman problems with time windows, Networks 22 (1992) 263–282.