Parametric Power Supply Networks

Shiho Morishita[†1,a)] Takao Nishizeki[†1,b)]

Abstract: Suppose that each vertex of a graph G is either a supply vertex or a demand vertex and is assigned a supply or a demand. All demands and supplies are nonnegative constant numbers in a steady network, while they are functions of a variable λ in a parametric network. Each demand vertex can receive "power" from exactly one supply vertex through edges in G. One thus wishes to partition G to connected components by deleting edges from G so that each component has exactly one supply vertex whose supply is at least the sum of demands in the component. The "partition problem" asks whether G has such a partition. If G has no such partition, one wishes to find a maximum number r^*, $0 \le r^* < 1$, such that G has such a partition when every demand is reduced to r^* times the original demand. The "maximum supply rate problem" asks to find such a number r^*. In this paper, we deal with a network in which G is a tree, and first give a polynomial-time algorithm for the maximum supply rate problem for a steady tree network, and then give an algorithm for the partition problem on a parametric tree network, which takes pseudo-polynomial time if all the supplies and demands are piecewise linear functions of λ.

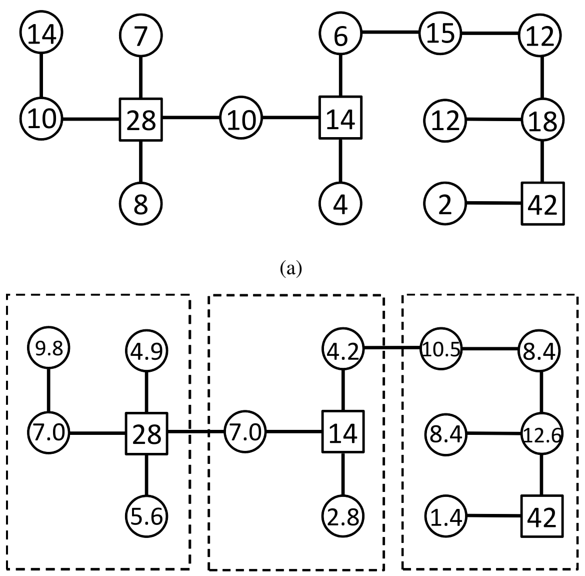
Keywords: Algorithm, Graph, Tree, Parametric network, Power supply.

1. Introduction

Consider a graph G in which each vertex is either a *supply vertex* or a *demand vertex* and is assigned a supply or a demand. Such a graph G is called a *power supply network*. All the supplies and demands are nonnegative constant numbers in an ordinary network, called a *steady network*, which has been considered so far [6], [7], [8], [9], [10]. This paper introduces a *parametric power supply network*, in which all the supplies and demands are functions of a parameter λ. The supply of a vertex v is denoted by $s_v(\lambda)$ and the demand by $d_v(\lambda)$. Figure 1 depicts steady networks; each supply vertex is drawn by a square, each demand vertex by a circle, and the supply or demand is written inside. Figure 2(a) depicts a parametric network, whose variable demands $d_{v_3}(\lambda)$ and $d_{v_4}(\lambda)$ are drawn in Fig. 2(b). Each demand vertex v must receive an amount $d_v(\lambda)$ of "power" or "commodity" from exactly one supply vertex through edges in a network G, while each supply vertex v can supply, to demand vertices, at most an amount $s_v(\lambda)$ of "power" in total. One thus wishes to partition G into connected components by deleting edges from G so that each component C has exactly one supply vertex whose supply is at least the sum of all demands in C. Such a partition is called a *feasible* partition of G. The *partition problem* asks, for each value of λ, whether G has a feasible partition. If G has no feasible partition for some value of λ, one wishes to find a maximum number r^*, $0 \le r^* < 1$, such that G has a feasible partition for the value if every demand $d_v(\lambda)$ is uniformly reduced to a new demand $d'_v(\lambda) = r^* \cdot d_v(\lambda)$. We call r^* the *maximum supply rate*, and call the problem of finding r^* the *maximum supply rate problem*. The maximum supply rate problem for a steady network is a special case of a partition problem for a parametric network, as will be observed in Section 2.

The steady network in Fig. 1(a) has no feasible partition, and the maximum supply rate r^* is 0.7; Fig. 1(b) depicts a new network with $d'_v = 0.7 \cdot d_v$ and illustrates a feasible partition by dotted lines.

The partition problem and the maximum supply rate problem have some applications to the power supply problem for power delivery networks, in which a supply or demand may depend on a parameter λ such as time, temperature, oil price, etc.[1], [11], [12], [13]. The partition problem is NP-complete



(a)

(b)

Fig. 1 (a) Steady tree network T with no feasible partition, and (b) new network constructed from T for the maximum supply rate $r^* = 0.7$.

1 Kwansei Gakuin University, 2-1 Gakuen, Sanda 669-1337, Japan
†1 Presently with Kwansei Gakuin University
a) morishita0731@gmail.com
b) nishi@kwansei.ac.jp

even for a steady network on a series-parallel graph, because the "set partition problem" [5], p. 47, can be easily reduced to the partition problem for a steady network on a complete bipartite graph $K_{2,n-2}$, which is a series-parallel graph [9]. Therefore, the maximum supply rate problem is NP-hard even for series-parallel steady networks. Hence, it is very unlikely that these problems can be solved in polynomial time even for series-parallel steady networks. However, the partition problem can be solved for steady tree networks in linear time [9].

In this thesis, we first give a polynomial-time algorithm to solve the maximum supply rate problem for a steady tree network T. It takes time $O(nL)$, where n is the number of vertices in T and L is the logarithmic size of T. We then present an algorithm to solve the partition problem for a parametric tree network. It takes pseudo-polynomial time if all supplies and demands are piecewise linear functions with integer coefficients. More precisely, it takes time $O(nW^2)$, where W is the sum of absolute values of all integer coefficients of supplies and demands.

2. Maximum Supply Rate Problem

In this section we deal with steady networks in which all supplies and demands are positive integers, and show that the maximum supply rate problem can be solved for steady tree networks in polynomial time.

Let $G = (V, E)$ be a steady network, where V is the set of vertices and E is the set of edges of G. Let V_s be the set of all supply vertices, and let V_d be the set of all demand vertices, then $V = V_s \bigcup V_d$ and $V_s \bigcap V_d = \emptyset$. Let $n = |V|$ and $n_s = |V_s|$. We denote by d_v the positive integral demand of a demand vertex v, and by s_v the positive integral supply of a supply vertex v. The *partition problem* asks whether V can be partitioned to a number n_s of subsets $V_1, V_2, \cdots, V_{n_s}$ such that each, V_i, $1 \le i \le n_s$, contains exactly one supply vertex, say u, and

$$\sum_{v \in V_i/\{u\}} d_v \le s_u.$$

We call such a partition of V a *feasible partition of* G. Ito *et al.* [9] gave an algorithm to solve the partition problem in time $O(n)$ if G is a tree. We call the algorithm **Partition**. Note that our parametric algorithm in Section 3 also runs in time $O(n)$ for a steady tree network.

The *maximum supply rate problem* for G asks to find the maximum number $r(> 0)$ such that G has a feasible partition if the demand d_v is replaced by a new demand $d'_v = r \cdot d_v$ for every demand vertex v. We call the maximum value r^* of such a number r the *maximum supply rate of* G. Thus the maximum supply rate r^* may be greater than 1. When $r^* < 1$, the value $1 - r^*$ is called the *minimum power saving rate of* G.

The maximum supply rate problem for a steady network G can be formulated as a partition problem for a parametric network G_{para}, in which the demand of every demand vertex v is a linear function $d_v(\lambda) = d_v \cdot \lambda$ and the supply of every supply vertex u is a constant function $s_u(\lambda) = s_u$. The maximum supply rate r^* of G is equal to the maximum value of λ for which G_{para} has a feasible partition.

Obviously, the following lemma holds.

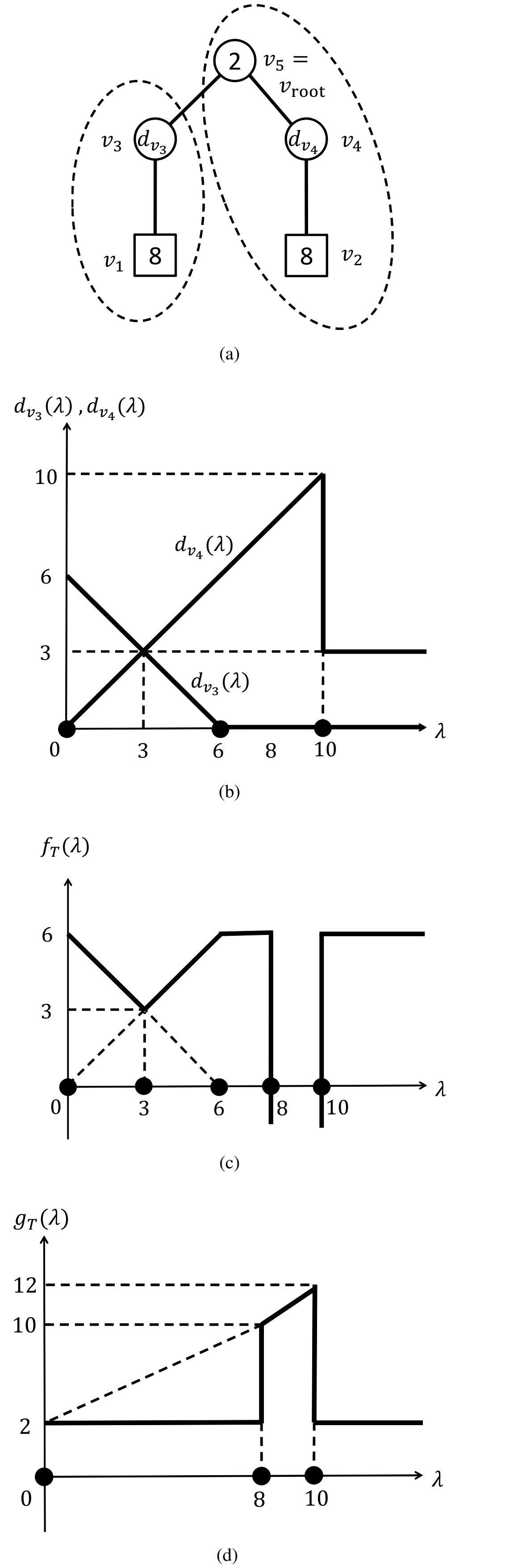


Fig. 2 (a) Parametric tree network T rooted at v_5, (b) variable demands $d_{v_3}(\lambda)$ and $d_{v_4}(\lambda)$, (c) surplus $f_T(\lambda)$, and (d) deficit $g_T(\lambda)$ of T.

Lemma. 1 Suppose that a steady network G has a feasible partition when every demand d_v is replaced by $d'_v = r \cdot d_v$ for a positive real number r. Then, for any number r' with $0 \leq r' \leq r$, G has a feasible partition when every demand d_v is replaced by $d'_v = r' \cdot d_v$.

One can thus compute r^* for a steady tree network T by a binary search on the infinite set of all positive real numbers r with the aid of the algorithm **Partition**. However, such a simple binary search either cannot exactly compute r^* or does not run in polynomial time. The idea of our algorithm is to notice that r^* is a rational number, as follows.

Lemma. 2 Let r^* be the maximum supply rate of a steady network $G = (V, E)$, and let $S = \prod_{v \in V_s} s_v$ and $D = \sum_{v \in V_d} d_v$. Then

$$r^* \in \{S/z | \, z \text{ is an integer and } S \cdot D \geq z \geq 1\}.$$

Proof. Since r^* is the maximum supply rate, there is a partition of V to subsets $V_1, V_2, \cdots, V_{n_s}$ such that each V_i, $1 \leq i \leq n_s$, contains exactly one supply vertex, say u, and

$$\sum_{v \in V_i/\{u\}} r^* \cdot d_v \leq s_u.$$

The inequality above holds in equality for some i, $1 \leq i \leq n_s$, that is,

$$r^* \sum_{v \in V_i/\{u\}} d_v = s_u;$$

otherwise, r^* would not be the maximum supply rate. Let $z^* = S/r^*$, then from the two equations above we have

$$z^* = \left(\frac{\prod_{v \in V_s} s_v}{s_u} \right) \cdot \sum_{v \in V_i/\{u\}} d_v.$$

Thus z^* is an integer, and $1 \leq z^* \leq S \cdot D$. Therefore, $r^* (= S/z^*)$ is equal to S/z for some integer z, $1 \leq z \leq S \cdot D$. □

Thus, one can find the maximum supply rate r^* of a steady tree network $T = (V, E)$ by a binary search on the finite set $\{S/z | \, S \cdot D \geq z \geq 1\}$ of rational numbers with the aid of **Partition** in time $O(n \log_2(S \cdot D))$.

The *logarithmic size* L of a steady network T is

$$L = \sum_{u \in V_s} \lceil \log_2(s_v + 1) \rceil + \sum_{v \in V_d} \lceil \log_2(d_v + 1) \rceil,$$

and clearly $\log_2(S \cdot D) \leq L$. We thus have the following theorem.

Theorem. 1 The maximum supply rate problem can be solved in time $O(nL)$ for a steady tree network T, where n is the number of vertices and L is the logarithmic size of T.

3. Parametric Networks

In this section we present an algorithm to solve the partition problem for a parametric tree network T.

3.1 Definitions

One may assume without loss of generality that a tree T is rooted at an arbitrarily chosen vertex v_{root}. We also assume that all supplies $s_v(\lambda)$ and demands $d_v(\lambda)$ in T are functions of a common nonnegative real variable $\lambda (\geq 0)$.

A *feasible partition* $\pi_\lambda = (V_1, V_2, \cdots, V_{n_s})$ *of a rooted tree network* $T = (V, E)$ *for a value* λ is a partition of V to a number n_s of subsets $V_1, V_2, \cdots, V_{n_s}$ such that

(a) the root of T is contained in V_1, that is, $v_{\text{root}} \in V_1$; and

(b) each V_i, $1 \leq i \leq n_s$, contains exactly one supply vertex, say u, and

$$\sum_{v \in V_i/\{u\}} d_v(\lambda) \leq s_u(\lambda).$$

The partition problem asks to find every value of λ for which T has a feasible partition. We actually find every interval of nonnegative real numbers such that T has a feasible partition π_λ for each value λ in the interval.

For the network T in Fig. 2(a), $v_5 = v_{\text{root}}$, $s_{v_1}(\lambda) = s_{v_2}(\lambda) = 8$, $d_{v_5}(\lambda) = 2$, and the variable demands $d_{v_3}(\lambda)$ and $d_{v_4}(\lambda)$ are drawn in Fig. 2(b). A feasible partition of T for $0 \leq \lambda \leq 6$ is indicated by dotted lines in Fig. 2(a). The solution for T is a set of two intervals $[0, 8]$ and $[10, \infty)$.

We find a feasible partition of T by the bottom-up computation on a rooted tree T. More precisely, we find a feasible partition and an extended partition, called a "root-feasible partition," from those of smaller subtrees.

A *root-feasible partition* $\pi_\lambda = (V_1, V_2, \cdots, V_{n_s+1})$ *of* T *for* λ is a partition of V to a number $n_s + 1$ of subsets $V_1, V_2, \cdots, V_{n_s+1}$ such that

(a) $v_{\text{root}} \in V_1$ and $V_1 \cap V_s = \emptyset$; and

(b) each V_i, $2 \leq i \leq n_s + 1$, contains exactly one supply vertex, say u, and

$$\sum_{v \in V_i/\{u\}} d_v(\lambda) \leq s_u(\lambda).$$

Thus, T has no root-feasible partition for any λ if v_{root} is a supply vertex. (A root-feasible partition of T in Fig. 2(a) for $0 \leq \lambda \leq 8$ is $(\{v_5\}, \{v_1, v_3\}, \{v_2, v_4\})$.)

Let $\pi_\lambda = (V_1, V_2, \cdots, V_{n_s})$ be a feasible partition of T for a value λ, and let u be the supply vertex in V_1. Then the *surplus* surp (π_λ) *of* π_λ is

$$\text{surp}(\pi_\lambda) = s_u(\lambda) - \sum_{v \in V_1/\{u\}} d_v(\lambda).$$

We now define a function $f_T(\lambda)$, called the *surplus of a parametric tree network* T, as follows:

$$f_T(\lambda) = \max_{\pi_\lambda} \text{surp}(\pi_\lambda)$$

where the maximum is taken over all feasible partitions π_λ of T for λ. Let $f_T(\lambda) = -\infty$ if T has no feasible partition for λ. Intuitively, $f_T(\lambda)$ is the maximum amount of power that can be delivered outside T through the root when all demand vertices are supplied power. (Figure 2(c) depicts $f_T(\lambda)$ for T in Fig. 2(a).)

Let $\pi_\lambda = (V_1, V_2, \cdots, V_{n_s+1})$ be a root-feasible partition of T for a value λ. Then $v_{\text{root}} \in V_1$, v_{root} is a demand vertex, and V_1 contains no supply vertex. The *deficit* $\text{def}(\pi_\lambda)$ *of* π_λ is

$$\text{def}(\pi_\lambda) = \sum_{v \in V_1} d_v(\lambda).$$

We now define a function $g_T(\lambda)$, called the *deficit of* T, as follows:

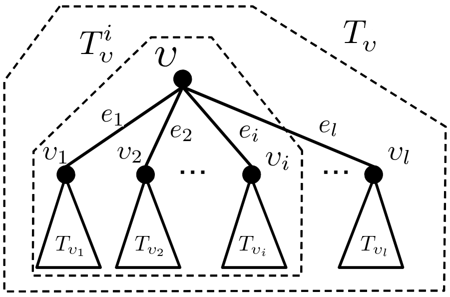


Fig. 3 Rooted subtrees.

$$g_T(\lambda) = \min_{\pi_\lambda} \ \mathrm{def}(\pi_\lambda)$$

where the minimum is taken over all root-feasible partitions π_λ of T for λ. Let $g_T(\lambda) = +\infty$ if T has no root-feasible partition for λ. Thus $g_T(\lambda) = +\infty$ for any λ if v_{root} is a supply vertex. Intuitively, $g_T(\lambda)$ is the minimum amount of power that must be delivered inside T through v_{root} when v_{root} and possibly some other demand vertices are supplied power from outside. (Figure 2(d) depicts $g_T(\lambda)$ for T in Fig. 2(a).)

We similarly define the surplus $f_{T'}(\lambda)$ and deficit $g_{T'}(\lambda)$ for a rooted subtree T' of T.

For a vertex v of T, we denote by T_v the maximum subtree of T rooted at v. Let $v_1, v_2, \cdots, v_l$ be the children of v in T, and let e_i, $1 \le i \le l$, be the edge joining v and v_i. Let T_{v_i}, $1 \le i \le l$, be the maximum subtree of T rooted at v_i. We denote by T_v^i the subtree of T_v which consists of vertex v, edges $e_1, e_2, \cdots, e_i$ and subtrees $T_{v_1}, T_{v_2}, \cdots, T_{v_i}$. In Fig. 3 T_v and T_v^i are surrounded by dotted lines. Clearly $T = T_{v_{\mathrm{root}}}$ and $T_v = T_v^l$. We denote by T_v^0 the subtree consisting of a single vertex v.

3.2 Algorithm

Our algorithm computes the surplus $f_{T_v}(\lambda)$ and deficit $g_{T_v}(\lambda)$ for each vertex v of T from leaves to the root of T by means of a dynamic programming approach, as described in (i)-(iii) below. From the surplus $f_T(\lambda)$ of $T = T_{v_{\mathrm{root}}}$, one can easily find every interval of nonnegative real numbers such that $f_T(\lambda) \ge 0$ for every value λ in the interval. We output the set of all these intervals as the solution of the partition problem of a parametric tree network T.

(i) We first compute the surplus and deficit of T_v^0 for each vertex v of T as follows. (Remember that T_v^0 consists of a single vertex v.) If v is a supply vertex, then $f_{T_v^0}(\lambda) = s_v(\lambda)$ and $g_{T_v^0}(\lambda) = +\infty$ for every λ. If v is a demand vertex, then $f_{T_v^0}(\lambda) = -\infty$ and $g_{T_v^0}(\lambda) = d_v(\lambda)$ for every λ. Since $T_v = T_v^0$ for every leaf v of T, we have thus computed f_{T_v} and g_{T_v} for every leaf v of T.

(ii) We next compute the surplus and deficit of a tree T_v^i, $1 \le i \le l$, for each internal vertex v of T from those of its subtrees T_v^{i-1} and T_{v_i}, where l is the number of the children of T_v. Note that $T_v = T_v^l$ and that T_v^i is obtained from T_v^{i-1} and T_{v_i} by joining v and v_i as illustrated in Fig. 4.

(ii-1) We first explain how to compute the surplus $f_{T_v^i}$ of T_v^i. Let n_i be the number of supply vertices in T_v^i. Assume that $f_{T_v^i}(\lambda) \neq -\infty$, that is, T_v^i has a feasible partition for λ. Let $\pi_\lambda = (V_1, V_2, \cdots, V_{n_i})$ be a feasible partition of T_v^i such that $f_{T_v^i}(\lambda) = \mathrm{surp}(\pi_\lambda)$. Then V_1 contains the root v of T_v^i, as illustrated in Fig. 4 where π_λ is indicated by dotted lines and a supply vertex is drawn by a square. There are the following three cases to consider.

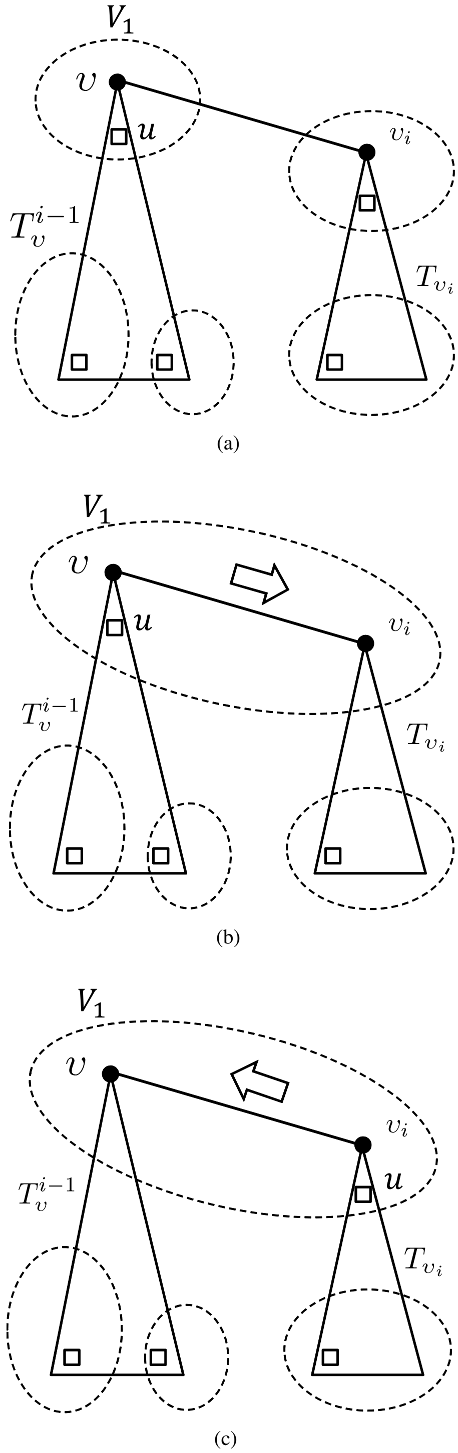


Fig. 4 Feasible partitions π_λ of T_v^i.

Case (a): $v_i \notin V_1$.

In this case, $f_{T_{v_i}}(\lambda) \ge 0$, and π_λ induces feasible partitions of T_v^{i-1} and T_{v_i}. (See Fig. 4(a).) For this case we compute the following function $f^{\mathrm{a}}_{T_v^i}$ from $f_{T_v^{i-1}}$ and $f_{T_{v_i}}$:

$$f^{\mathrm{a}}_{T^i_v}(\lambda) = \begin{cases} f_{T^{i-1}_v}(\lambda) & \text{if } f_{T_{v_i}}(\lambda) \geq 0; \\ -\infty & \text{otherwise.} \end{cases}$$

Case (b): $v_i \in V_1$, *and the supply vertex u in V_1 is contained in* T^{i-1}_v.

In this case, $f_{T^{i-1}_v}(\lambda) \geq g_{T_{v_i}}(\lambda)$, and π_λ induces a feasible partition of T^{i-1}_v and a root-feasible partition of T_{v_i}. (In Fig. 4(b) the arrow attached to edge (v, v_i) indicates the direction of power flow through it.) For this case we compute the following function $f^{\mathrm{b}}_{T^i_v}$:

$$f^{\mathrm{b}}_{T^i_v}(\lambda) = \begin{cases} f_{T^{i-1}_v}(\lambda) - g_{T_{v_i}}(\lambda) & \text{if} f_{T^{i-1}_v}(\lambda) \geq g_{T_{v_i}}(\lambda); \\ -\infty & \text{otherwise.} \end{cases}$$

Case (c): $v_i \in V_1$, *and the supply vertex u in V_1 is contained in* T_{v_i}.

In this case, $g_{T^{i-1}_v}(\lambda) \leq f_{T_{v_i}}(\lambda)$, and π_λ induces a root-feasible partition of T^{i-1}_v and a feasible partition of T_{v_i}. (See Fig. 4(c).) For this case we compute the following function $f^{\mathrm{c}}_{T^i_v}$:

$$f^{\mathrm{c}}_{T^i_v}(\lambda) = \begin{cases} -g_{T^{i-1}_v}(\lambda) + f_{T_{v_i}}(\lambda) & \text{if } g_{T^{i-1}_v}(\lambda) \leq f_{T_{v_i}}(\lambda); \\ -\infty & \text{otherwise.} \end{cases}$$

From the three functions $f^{\mathrm{a}}_{T^i_v}$, $f^{\mathrm{b}}_{T^i_v}$ and $f^{\mathrm{c}}_{T^i_v}$ above, we can now compute $f_{T^i_v}$ as follows:

$$f_{T^i_v}(\lambda) = \max\{f^{\mathrm{a}}_{T^i_v}(\lambda), f^{\mathrm{b}}_{T^i_v}(\lambda), f^{\mathrm{c}}_{T^i_v}(\lambda)\}.$$

(ii-2) We next explain how to compute the deficit $g_{T^i_v}$ of T^i_v. Assume that $g_{T^i_v}(\lambda) \neq +\infty$, that is, T^i_v has a root-feasible partition for λ. Let $\pi_\lambda = (V_1, V_2, \cdots, V_{n_i+1})$ be a root-feasible partition of T^i_v such that $g_{T^i_v}(\lambda) = \mathrm{def}(\pi_\lambda)$. Then V_1 contains the root v of T^i_v and does not contain any supply vertex, as illustrated in Fig. 5. There are the following two cases to consider.

Case (a): $v_i \notin V_1$.

In this case, $f_{T_{v_i}}(\lambda) \geq 0$, and π_λ induces a root-feasible partition of T^{i-1}_v and a feasible partition of T_{v_i}. (See Fig. 5(a).) For this case we compute the following function $g^{\mathrm{a}}_{T^i_v}$:

$$g^{\mathrm{a}}_{T^i_v}(\lambda) = \begin{cases} g_{T^{i-1}_v}(\lambda) & \text{if } f_{T_{v_i}}(\lambda) \geq 0; \\ +\infty & \text{otherwise.} \end{cases}$$

Case (b): $v_i \in V_1$.

In this case, $g_{T_{v_i}}(\lambda) \neq +\infty$, and π_λ induces root-feasible partitions of T^{i-1}_v and T_{v_i}. (See Fig. 5(b).) For this case we compute the following function $g^{\mathrm{b}}_{T^i_v}$:

$$g^{\mathrm{b}}_{T^i_v}(\lambda) = \begin{cases} g_{T^{i-1}_v}(\lambda) + g_{T_{v_i}}(\lambda) & \text{if } g_{T_{v_i}}(\lambda) \neq +\infty; \\ +\infty & \text{otherwise.} \end{cases}$$

From the two functions $g^{\mathrm{a}}_{T^i_v}$ and $g^{\mathrm{b}}_{T^i_v}$ above, we can now compute $g_{T^i_v}$ as follows:

$$g_{T^i_v}(\lambda) = \min\{g^{\mathrm{a}}_{T_{v_i}}(\lambda), g^{\mathrm{b}}_{T_{v_i}}(\lambda)\}.$$

(iii) Repeating the computation in (ii) above for each edge (v, v_i) of T, we can compute $f_T(\lambda)$ and $g_T(\lambda)$.

3.3 Computation Time

In this subsection we assume that all the supplies and demands are piecewise linear functions of a common variable $\lambda(\geq 0)$, and

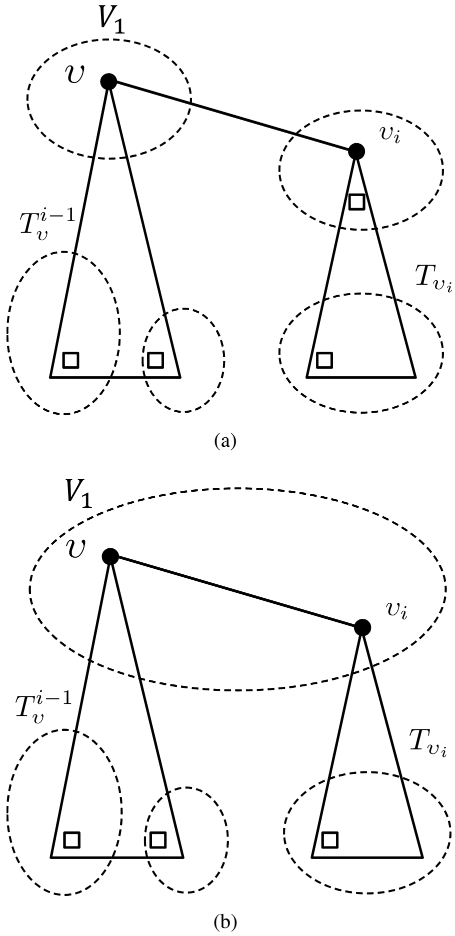


Fig. 5 Root-feasible partitions π_λ of T^i_v.

analyze the computation time of our algorithm.

A *breakpoint* of a piecewise linear function f is defined to be a point λ at which the slope of the curve of f changes, and the number of breakpoints of f is denoted by $p(f)$. For the sake of convenience, we assume that $\lambda = 0$ is a breakpoint of f. (For example, $p(d_{v_3}(\lambda)) = 2$ and $p(f_T(\lambda)) = 5$ for $d_{v_3}(\lambda)$ and $f_T(\lambda)$ in Fig. 2 where a breakpoint is drawn as a black dot.) Then the *size* N of a parametric tree network $T = (V, E)$ is

$$N = \sum_{v \in V_{\mathrm{s}}} p(s_v(\lambda)) + \sum_{v \in V_{\mathrm{d}}} p(d_v(\lambda)).$$

We define P as follows:

$$P = \max_{T'} \ \max\{p(f_{T'}(\lambda)), p(g_{T'}(\lambda))\}$$

where T' runs over all rooted subtrees of T. Note that $f_{T'}$ and $g_{T'}$ are piecewise linear functions.

Clearly one can compute the surplus $f_{T^0_v}(\lambda)$ and deficit $g_{T^0_v}(\lambda)$ of all vertices v in T in time $O(N)$ as in (i) of Sect. 3.2.

One can compute $f_{T^i_v}$ and $g_{T^i_v}$ for tree T^i_v from those for its subtrees T^{i-1}_v and T_{v_i} in time $O(P)$ as in (ii) of Sect. 3.2. (Note that the maximum and minimum of two piecewise linear functions can be computed by finding the upper and lower envelopes of the two piecewise linear curves, respectively.) The computation of (ii) occurs $n-1$ times since T has $n-1$ edges. Hence, one can compute $f_T(\lambda)$ and $g_T(\lambda)$ in time $O(nP)$. From $f_T(\lambda)$ one can find, in time $O(P)$, all intervals such that T has a feasible partition π_λ

for any value λ in each integral. Thus the partition problem can be solved in time $O(nP)$.

P may be greater than N. (For example, neither the breakpoint $\lambda = 3$ nor $\lambda = 8$ of $f_T(\lambda)$ is a breakpoint of any supply or demand of T in Fig. 2(a).) However, P is often bounded by a polynomial in N in many practical applications. In particular, if T is a steady network then $P = 1$ and hence our algorithm takes time $O(n)$. If all supplies and demands are staircase functions, then $P \le N$ and hence our algorithm takes time $O(nN)$.

3.4 Bounds on P

In this subsection we assume that all the supplies and demands of T are piecewise linear functions with integer coefficients.

Let B be the number of breakpoints of supplies and demands, and let $p_1, p_2, \cdots, p_B$ be these points. (For T in Fig. 2(a) $B = 3$ as indicated by three black dots in Fig. 2(b).) One may assume that $0 = p_1 < p_2 < \cdots < p_B$, and let $p_{B+1} = \infty$. We now assume that

(a) if v is a supply vertex and $p_i < \lambda < p_{i+1}$, $1 \le i \le B$, then

$$s_v(\lambda) = a_{vi}\lambda + b_{vi}$$

for some integers a_{vi} and b_{vi} (possibly after multiplying them by the least common multiple of denominators); and

(b) if v is a demand vertex and $p_i < \lambda < p_{i+1}$, $1 \le i \le B$, then

$$d_v(\lambda) = a_{vi}\lambda + b_{vi}$$

for some integers a_{vi} and b_{vi}.

Let

$$W = \sum_{v \in V} \sum_{i=1}^{B} (|a_{vi}| + |b_{vi}|).$$

Then we show that P is bounded by pseudo-polynomial, that is, P is bounded by a polynomial in W. More precisely, we have the following lemma.

Lemma. 3 $P = O(W^2)$.

Proof. P is the number of breakpoints of $f_{T'}$ or $g_{T'}$ for some rooted subtree T' of T. Assume that P is the number of breakpoints of $f_{T'}$. (The proof for the other case is similar.) Let n' be the number of supply vertices in T'. Let λ be a breakpoint of $f_{T'}$ which is none of the breakpoints $p_1, p_2, \cdots, p_B$ of supplies and demands. Then $p_i < \lambda < p_{i+1}$ for some i, $1 \le i \le B$. Such a breakpoint λ is a rational number expressed in terms of integer coefficients of supplies and demands, as follows.

Consider first the case where the function $f_{T'}$ is continuous at λ. Then there are two distinct feasible partitions $\pi_\lambda = (V_1, V_2, \cdots, V_{n'})$ and $\pi'_\lambda = (V'_1, V'_2, \cdots, V'_{n'})$ such that $\mathrm{surp}(\pi_\lambda)$= $\mathrm{surp}(\pi'_\lambda)$ and $V_1 \ne V'_1$; otherwise, λ would not be a breakpoint at which $f_{T'}$ is continuous. (For the continuous breakpoint λ=3 in Fig. 2(c), $\pi_\lambda = (\{v_5, v_4, v_2\}, \{v_3, v_1\})$ and $\pi'_\lambda = (\{v_5, v_3, v_1\}, \{v_4, v_2\})$.) Let u be the supply vertex in V_1, and let u' be the supply vertex in V'_1. Since $\mathrm{surp}(\pi_\lambda) = \mathrm{surp}(\pi'_\lambda)$, we have

$$s_u(\lambda) - \sum_{v \in V_1/\{u\}} d_v(\lambda) = s_{u'}(\lambda) - \sum_{v \in V'_1/\{u'\}} d_v(\lambda)$$

Since $p_i < \lambda < p_{i+1}$, we have

$$a_{ui}\lambda + b_{ui} - \sum_{v \in V_1/\{u\}} (a_{vi}\lambda + b_{vi}) = a_{u'i}\lambda + b_{u'i} - \sum_{v \in V'_1/\{u'\}} (a_{vi}\lambda + b_{vi})$$

and hence

$$\lambda = \frac{-b_{ui} + b_{u'i} + \sum_{v \in V_1/\{u\}} b_{vi} - \sum_{v \in V'_1/\{u'\}} b_{vi}}{a_{ui} - a_{u'i} - \sum_{v \in V_1/\{u\}} a_{vi} + \sum_{v \in V'_1/\{u'\}} a_{vi}}.$$

Both the numerator and denominator above are integers between $-W$ and W, and hence the number of such continuous breakpoints λ is bounded by $(2W + 1)^2$.

Consider next the case where $f_{T'}$ is discontinuous at λ. Then there is a feasible partition $\pi_\lambda = (V_1, V_2, \cdots, V_{n'})$ such that $f_{T'}(\lambda) = \mathrm{surp}(\pi_\lambda)$ and

$$s_u(\lambda) - \sum_{v \in V_j/\{u\}} d_v(\lambda) = 0$$

for some j, $1 \le i \le n'$, where u is the supply vertex in V_j. (For the discontinuous breakpoint $\lambda = 8$ in Fig. 2(c), $\pi_\lambda = (\{v_5, v_3, v_1\}, \{v_4, v_2\})$ and $j = 2$.) Since $p_i < \lambda < p_{i+1}$, we have

$$a_{ui}\lambda + b_{ui} - \sum_{v \in V_j/\{u\}} (a_{vi}\lambda + b_{vi}) = 0$$

and hence

$$\lambda = \frac{-b_{ui} + \sum_{v \in V_j/\{u\}} b_{vi}}{a_{ui} - \sum_{v \in V_j/\{u\}} a_{vi}}.$$

Therefore, the number of such discontinuous breakpoints λ is bounded by $(2W + 1)^2$.

We have thus proved that $P \le 2(2W + 1)^2 + B = O(W^2)$. □

From the lemma above we have the following theorem.

Theorem. 2 The partition problem for a parametric tree network can be solved in time $O(nW^2)$ if all supplies and demands are piecewise linear functions with integer coefficients, where W is the sum of absolute values of all coefficients of supplies and demands.

Thus, our algorithm runs in polynomial time if W is polynomial in N.

4. Conclusions

In the paper we first showed that the maximum supply rate problem for a steady tree network T can be solved in time $O(nL)$, where n is the number of vertices in T and L is the logarithmic size of T. It would be interesting to obtain a strongly polynomial-time algorithm for the problem, whose computation time is bounded by a polynomial only in n.

We then gave an algorithm to solve the partition problem for a parametric tree network. The algorithm runs in pseudo-polynomial time if all the supplies and demands are piecewise linear functions with integer coefficients. We assumed for the sake of convenience that the supplies and demands are functions of a single variable λ. However, our algorithm in Section 3.2 can be easily extended to the case where the supplies and demands are functions of two or more variables.

Kawabata and Nishizeki [10] considered a steady tree network in which each edge is also assigned a constant edge-capacity, and gave a linear algorithm to solve the partition problem. Our algorithm for the maximum supply rate problem in Section 2 can be easily extended to the case of a steady tree network with constant edge-capacity, and our algorithm for the partition problem in Section 3.2 can be extended to the case of a parametric tree network in which edge capacity is also a function of λ. Note that our problems with (constant or functional) edge-capacity cannot be formulated by the multi-source multi-sink parametric flow problem or the unsplitable parametric flow problem [2], [3], [4], [11].

If a tree network $T = (V, E)$ has no feasible partition, one would like to solve the *maximum partition problem*, which asks to find a partition of V to subsets such that

(a) each subset contains at most one supply vertex;
(b) if a subset contains a supply vertex, then the supply is no less than the sum of demands in the subsets; and
(c) the sum of demands in all subsets, each containing a supply vertex, is maximum among all these partitions of V.

There are fully polynomial-time approximation schemes (FPTAS) for the problem on a steady tree network without edge-capacity [8] and on a steady tree network with constant edge-capacity [10]. It would be interesting to obtain an FPTAS for the problem on a parametric tree network with or without edge-capacity.

Acknowledgments This research was supported by MEXT-Supported Program for the Strategic Research Foundation at Private Universities.

References

[1] N. G. Boulaxis and M.P. Papadopoulos, Optimal feeder routing in distribution system planning using dynamic programming technique and GIS facilities, IEEE Trans. Power Delivery, Vol. 17, No. 1, pp. 242-247, 2002.
[2] C. Chekuri, A. Ene and N. Korula, Unsplitable flow in paths and trees and column-restricted packing integer programs, Proc. APPROX-RANDOM 2009, Springer Lect. Notes in Computer Science, Vol. 5687, pp. 42-55, 2009.
[3] C. Chekuri, M. Mydlarz and F. B. Shepherd, Multicommodity demand flow in a tree, ACM Trans. on Algorithms, Vol. 3, Article 3, 2007.
[4] G. Gallo, M. D. Grigoriadis and R. E. Tarjan, A fast parametric maximum flow algorithm and applications, SIAM J. Comput., Vol. 18, No. 1, pp. 30-55, 1989.
[5] M. R. Garey and D. S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, Freeman, San Francisco, CA, 1979.
[6] T. Ito, E. D. Demaine, X. Zhou and T. Nishizeki, Approximability of partitioning graphs with supply and demand, Journal of Discrete Algorithms, Vol. 6, pp. 627-650, 2008.
[7] T. Ito, T. Hara, X. Zhou and T. Nishizeki, Minimum cost partitions of trees with supply and demand, Algorithmica, Vol. 64, pp. 400-415, 2012.
[8] T. Ito, X. Zhou and T. Nishizeki, Partitioning graphs of supply and demand, Discrete Applied Math., Vol. 157, pp. 2620-2633, 2009.
[9] T. Ito, X. Zhou and T. Nishizeki, Partitioning trees of supply and demand, Int. J. Found. Comput. Sci., Vol. 16, pp. 803-827, 2005.
[10] M. Kawabata and T. Nishizeki, Partitioning trees with supply, demand and edge-capacity, Proc. of ISORA 2011, Lecture Notes in Operation Research, Vol. 14, pp. 51-58, 2011, also IEICE Trans. on Fundamentals of Electronics, Communications and Computer Science, to appear.
[11] E. Minieka, Parametric network flows, Operation Research, Vol. 20, No. 6, pp. 1162-1170, 1972.
[12] A. B. Morton and I. M. Y. Mareels, An efficient brute-force solution to the network reconfiguration problem, IEEE Trans. Power Delivery, Vol. 15, pp. 996-1000, 2000.
[13] J. -H. Teng and C.-N. Lu, Feeder-switch relocation for customer interruption cost minimization, IEEE Trans. Power Delivery, Vol. 17, pp. 254-259, 2002.

