

# Approximability of Partitioning Graphs with Supply and Demand

## Extended Abstract

Takehiro Ito<sup>1</sup>, Erik D. Demaine<sup>2</sup>, Xiao Zhou<sup>1</sup>, and Takao Nishizeki<sup>1</sup>

<sup>1</sup> Graduate School of Information Sciences, Tohoku University,  
Aoba-yama 6-6-05, Sendai, 980-8579, Japan

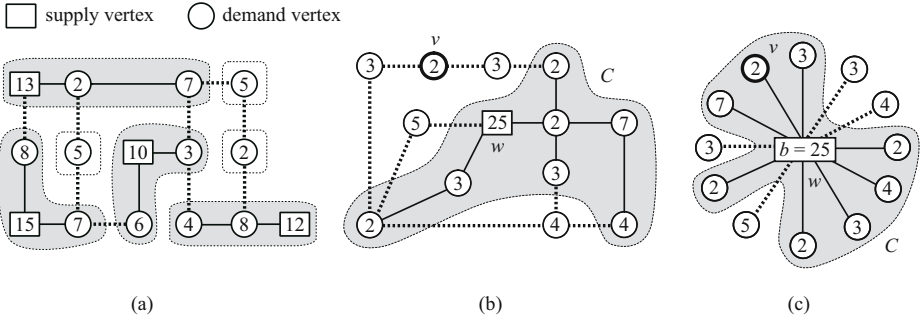
<sup>2</sup> MIT Computer Science and Artificial Intelligence Laboratory,  
32 Vassar St., Cambridge, MA 02139, USA

take@nishizeki.ecei.tohoku.ac.jp, edemaine@mit.edu,  
{zhou, nishi}@ecei.tohoku.ac.jp

**Abstract.** Suppose that each vertex of a graph  $G$  is either a supply vertex or a demand vertex and is assigned a positive real number, called the supply or the demand. Each demand vertex can receive “power” from at most one supply vertex through edges in  $G$ . One thus wishes to partition  $G$  into connected components so that each component  $C$  either has no supply vertex or has exactly one supply vertex whose supply is at least the sum of demands in  $C$ , and wishes to maximize the fulfillment, that is, the sum of demands in all components with supply vertices. This maximization problem is known to be NP-hard even for trees having exactly one supply vertex and strongly NP-hard for general graphs. In this paper, we focus on the approximability of the problem. We first show that the problem is MAXSNP-hard and hence there is no polynomial-time approximation scheme (PTAS) for general graphs unless  $P = NP$ . We then present a fully polynomial-time approximation scheme (FPTAS) for series-parallel graphs having exactly one supply vertex. The FPTAS can be easily extended for partial  $k$ -trees, that is, graphs with bounded treewidth.

## 1 Introduction

Consider a graph  $G$  such that each vertex is either a supply vertex or a demand vertex. Each vertex  $v$  is assigned a positive real number; the number is called the *supply* of  $v$  if  $v$  is a supply vertex; otherwise, it is called the *demand* of  $v$ . Each demand vertex can receive “power” from at most one supply vertex through edges in  $G$ . 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 demands of all demand vertices in  $C$ . However, such a partition does not always exist. So we wish to partition  $G$  into connected components so that each component  $C$  either has no supply vertex or has exactly one supply vertex whose supply is at least the sum of demands of all demand vertices in  $C$ , and wish to maximize the “fulfillment,” that is, the sum



**Fig. 1.** (a) Partition of a graph with maximum fulfillment, (b) partition of a series-parallel graph  $G$  having exactly one supply vertex, and (c) a star  $S$  with a supply vertex at the center

of demands of the demand vertices in all components with supply vertices. We call this problem the *maximum partition problem* [4]. The maximum partition problem has some applications to the power supply problem for power delivery networks [4,6]. Figure 1(a) illustrates a solution of the maximum partition problem for a graph, whose fulfillment is  $(2 + 7) + (8 + 7) + (3 + 6) + (4 + 8) = 45$ . In Fig. 1(a) each supply vertex is drawn as a rectangle and each demand vertex as a circle, the supply or demand is written inside, the deleted edges are drawn by thick dotted lines, and each connected component with a supply vertex is shaded.

Given a set  $A$  of integers and an upper bound (integer)  $b$ , the *maximum subset sum problem* asks to find a subset  $C$  of  $A$  such that the sum of integers in  $C$  is no greater than the bound  $b$  and is maximum among all such subsets  $C$ . The maximum subset sum problem can be reduced in linear time to the maximum partition problem for a particular tree, called a star, with exactly one supply vertex at the center, as illustrated in Fig. 1(c) [4]. Since the maximum subset sum problem is NP-hard, the maximum partition problem is also NP-hard even for stars. Thus it is very unlikely that the maximum partition problem can be exactly solved in polynomial time even for trees. However, there is a fully polynomial-time approximation scheme (FPTAS) for the maximum partition problem on trees [4]. One may thus expect that the FPTAS for trees can be extended to a larger class of graphs, for example series-parallel graphs and partial  $k$ -trees, that is, graphs with bounded treewidth [1,2].

In this paper, we study the approximability of the maximum partition problem. We first show that the maximum partition problem is MAXSNP-hard, and hence there is no polynomial-time approximation scheme (PTAS) for the problem on general graphs unless  $P = NP$ . We then present an FPTAS for series-parallel graphs having exactly one supply vertex. The FPTAS for series-parallel graphs can be extended to partial  $k$ -trees. (The details are omitted from this extended abstract.) Figure 1(b) depicts a series-parallel graph together with a connected component  $C$  found by our FPTAS. One might think that it would be straightforward to extend the FPTAS for the maximum subset sum problem in [3] to an

FPTAS for the maximum partition problem with a single supply vertex. However, this is not the case since we must take a graph structure into account. For example, the vertex  $v$  of demand 2 in Fig. 1(b) cannot be supplied power even though the supply vertex  $w$  has marginal power  $25 - (2 + 3 + 2 + 2 + 3 + 7 + 4) = 2$ , while the vertex  $v$  in Fig. 1(c) can be supplied power from the supply vertex  $w$  in the star having the same supply and demands as in Fig. 1(b). Indeed, we not only extend the “scaling and rounding” technique but also employ many new ideas to derive our FPTAS.

The rest of the paper is organized as follows. In Section 2 we show that the maximum partition problem is MAXSNP-hard. In Section 3 we present a pseudo-polynomial-time algorithm for series-parallel graphs. In Section 4 we present an FPTAS based on the algorithm in Section 3.

## 2 MAXSNP-Hardness

Assume in this section that a graph  $G$  has one or more supply vertices. (See Figs. 1(a) and 2(b).) The main result of this section is the following theorem.

**Theorem 1.** *The maximum partition problem is MAXSNP-hard for bipartite graphs.*

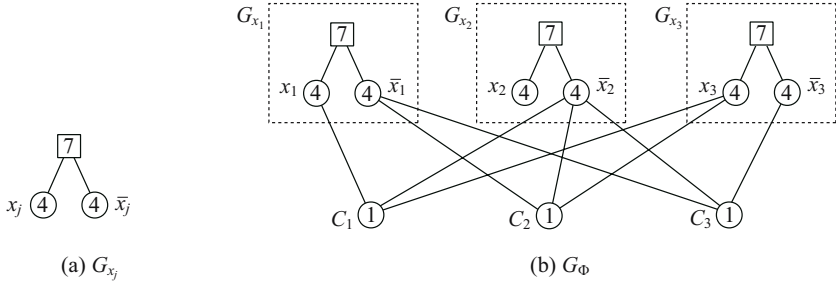
*Proof.* As in [7,8], we use the concept of “L-reduction” which is a special kind of reduction that preserves approximability. Suppose that both  $A$  and  $B$  are maximization problems. Then we say that  $A$  can be  $L$ -reduced to  $B$  if there exist two polynomial-time algorithms  $R$  and  $S$  and two positive constants  $\alpha$  and  $\beta$  which satisfy the following two conditions (1) and (2) for each instance  $I_A$  of  $A$ :

- (1) the algorithm  $R$  returns an instance  $I_B = R(I_A)$  of  $B$  such that  $OPT_B(I_B) \leq \alpha \cdot OPT_A(I_A)$ , where  $OPT_A(I_A)$  and  $OPT_B(I_B)$  denote the maximum solution values of  $I_A$  and  $I_B$ , respectively; and
- (2) for each feasible solution of  $I_B$  with value  $c_B$ , the algorithm  $S$  returns a feasible solution of  $I_A$  with value  $c_A$  such that  $OPT_A(I_A) - c_A \leq \beta \cdot (OPT_B(I_B) - c_B)$ .

Note that, by condition (2) of the L-reduction,  $S$  must return the optimal solution of  $I_A$  for the optimal solution of  $I_B$ .

We show that a MAXSNP-hard problem, called “3-occurrence MAX3SAT” [7,8], can be L-reduced to the maximum partition problem for bipartite graphs. However, due to the page limitation, we only show in this extended abstract that condition (1) of the L-reduction holds.

We now show that condition (1) of the L-reduction holds. An instance  $\Phi$  of 3-occurrence MAX3SAT consists of a collection of  $m$  clauses  $C_1, C_2, \dots, C_m$  on  $n$  variables  $x_1, x_2, \dots, x_n$  such that each clause has exactly three literals and each variable appears at most three times in the clauses. The problem 3-occurrence MAX3SAT is to find a truth assignment for the variables which satisfies the maximum number of clauses. Then it suffices to show that, from each instance



**Fig. 2.** (a) Variable gadget  $G_{x_j}$ , and (b) the bipartite graph  $G_{\Phi}$  corresponding to an instance  $\Phi$  with three clauses  $C_1 = (x_1 \vee \bar{x}_2 \vee x_3)$ ,  $C_2 = (\bar{x}_1 \vee \bar{x}_2 \vee x_3)$  and  $C_3 = (\bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3)$

$\Phi$  of 3-occurrence MAX3SAT, one can construct in polynomial time a bipartite graph  $G_{\Phi}$  as an instance of the maximum partition problem such that

$$OPT_{MPP}(G_{\Phi}) \leq 26 \cdot OPT_{SAT}(\Phi), \quad (1)$$

where  $OPT_{MPP}(G_{\Phi})$  and  $OPT_{SAT}(\Phi)$  are the maximum solution values of  $G_{\Phi}$  and  $\Phi$ , respectively: condition (1) of the L-reduction holds for  $\alpha = 26$ .

We first make a *variable gadget*  $G_{x_j}$  for each variable  $x_j$ ,  $1 \leq j \leq n$ ;  $G_{x_j}$  is a binary tree with three vertices as illustrated in Fig. 2(a); the root is a supply vertex of supply 7, and two leaves  $x_j$  and  $\bar{x}_j$  are demand vertices of demands 4. The graph  $G_{\Phi}$  is constructed as follows. For each variable  $x_j$ ,  $1 \leq j \leq n$ , put the variable gadget  $G_{x_j}$  to the graph, and for each clause  $C_i$ ,  $1 \leq i \leq m$ , put a demand vertex  $C_i$  of demand 1 to the graph. Finally, for each clause  $C_i$ ,  $1 \leq i \leq m$ , join a demand vertex  $x_j$  (or  $\bar{x}_j$ ) in  $G_{x_j}$  with the demand vertex  $C_i$  if and only if the literal  $x_j$  (or  $\bar{x}_j$ ) is in  $C_i$ , as illustrated in Fig. 2(b). Clearly,  $G_{\Phi}$  can be constructed in polynomial time, and is a bipartite graph. It should be noted that the degree of each demand vertex in the variable gadget for  $x_j$  is at most four since  $x_j$  appears at most three times in the clauses. Therefore, each supply vertex in the variable gadget  $G_{x_j}$  has enough “power” to supply all demand vertices  $C_i$  whose corresponding clauses have  $x_j$  or  $\bar{x}_j$ . Then one can verify Eq. (1), whose proof is omitted from this extended abstract.  $\square$

### 3 Pseudo-polynomial-time Algorithm

Since the maximum partition problem is strongly NP-hard [5], there is no pseudo-polynomial-time algorithm for general graphs unless  $P = NP$ . However, Ito *et al.* presented a pseudo-polynomial-time algorithm for the maximum partition problem on series-parallel graphs [5]. In this section we present another pseudo-polynomial-time algorithm, which is suited to an FPTAS presented in Section 4. More precisely, we have the following theorem.

**Theorem 2.** *The maximum partition problem for a series-parallel graph  $G$  with a single supply vertex can be solved in time  $O(F^2n)$  if the demands and the supply*

are integers, where  $n$  is the number of vertices and  $F$  is the sum of all demands in  $G$ .

### 3.1 Terminology and Definitions

Suppose that there is exactly one supply vertex  $w$  in a graph  $G = (V, E)$ , as illustrated in Figs. 1(b) and (c). Let  $\text{sup}(w)$  be the supply of  $w$ . For each demand vertex  $v$ , we denote by  $\text{dem}(v)$  the demand of  $v$ . Let  $\text{dem}(w) = 0$  although  $w$  is a supply vertex. Then, instead of finding a partition of  $G$ , we shall find a set  $C \subseteq V$ , called a *supplied set* for  $G$ , such that

- (a)  $w \in C$ ;
- (b)  $\sum_{v \in C} \text{dem}(v) \leq \text{sup}(w)$ ; and
- (c)  $C$  induces a connected subgraph of  $G$ .

The *fulfillment*  $f(C)$  of a *supplied set*  $C$  is  $\sum_{v \in C} \text{dem}(v)$ . A supplied set  $C$  is called the *maximum supplied set* for  $G$  if  $f(C)$  is maximum among all supplied sets for  $G$ . Then the *maximum partition problem* is to find a maximum supplied set for a given graph  $G$ . The *maximum fulfillment*  $f(G)$  of a graph  $G$  is the fulfillment  $f(C)$  of the maximum supplied set  $C$  for  $G$ . For the series-parallel graph  $G$  in Fig. 1(b), the supplied set  $C$  shaded in the figure has the maximum fulfillment, and hence  $f(G) = f(C) = 23$ , while  $f(S) = 25$  for the star  $S$  in Fig. 1(c).

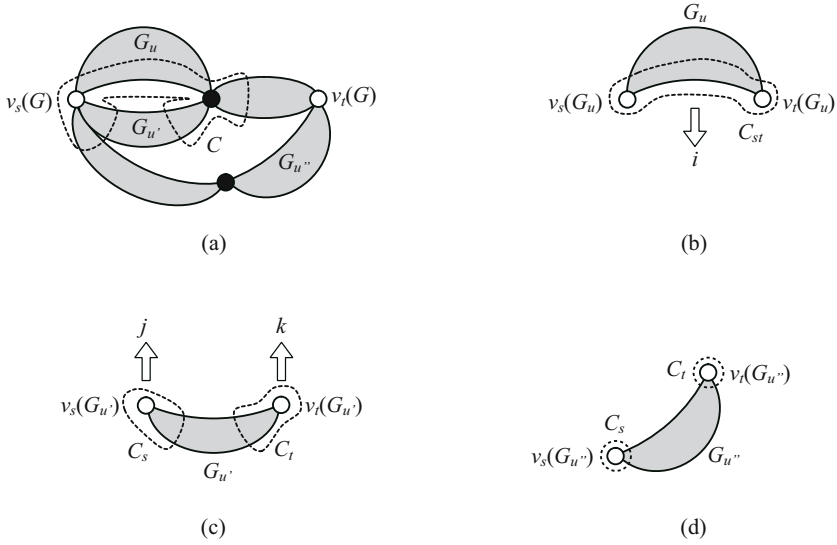
A (*two-terminal*) *series-parallel graph*  $G$  is defined recursively as a graph obtained from two series-parallel graphs by the so-called series or parallel connection [9]. The terminals of  $G$  are denoted by  $v_s(G)$  and  $v_t(G)$ . Since we deal with the maximum partition problem, we may assume without loss of generality that  $G$  is a simple graph.

A series-parallel graph  $G$  can be represented by a “binary decomposition tree”  $T$  [9]. Every leaf of  $T$  represents a subgraph of  $G$  induced by a single edge. Each node  $u$  of  $T$  corresponds to a subgraph  $G_u = (V_u, E_u)$  of  $G$  induced by all edges represented by the leaves that are descendants of  $u$  in  $T$ .  $G_u$  is a series-parallel graph for each node  $u$  of  $T$ , and  $G = G_r$  for the root  $r$  of  $T$ . Since a binary decomposition tree of a given series-parallel graph  $G$  can be found in linear time [9], we may assume that a series-parallel graph  $G$  and its binary decomposition tree  $T$  are given.

### 3.2 Algorithm

In this subsection we give an algorithm to solve the maximum partition problem in time  $O(F^2n)$  as a proof of Theorem 2.

Let  $G$  be a series-parallel graph, let  $u, u'$  and  $u''$  be nodes of a binary decomposition tree  $T$  of  $G$ , and let  $G_u = (V_u, E_u)$ ,  $G_{u'} = (V_{u'}, E_{u'})$  and  $G_{u''} = (V_{u''}, E_{u''})$  be the subgraphs of  $G$  for nodes  $u, u'$  and  $u''$ , respectively, as illustrated in Fig. 3(a). Every supplied set  $C$  for  $G$  naturally induces subsets of  $V_u, V_{u'}$  and  $V_{u''}$ . The supplied set  $C$  for  $G$  in Fig. 3(a) induces a single subset  $C_{st}$  of  $V_u$  in Fig. 3(b) such that  $C_{st} = C \cap V_u$  and  $v_s(G_u), v_t(G_u) \in C_{st}$ . On the other



**Fig. 3.** (a) A supplied set  $C$  for a series-parallel graph  $G$ , (b) a connected set  $C_{st}$  for  $G_u$ , (c) a separated pair  $(C_s, C_t)$  of sets for  $G_{u'}$ , and (d) a separated pair  $(C_s, C_t)$  of isolated sets for  $G_{u''}$

hand,  $C$  induces a pair of subsets  $C_s$  and  $C_t$  of  $V_{u'}$  in Fig. 3(c) such that  $C_s \cup C_t = C \cap V_{u'}$ ,  $C_s \cap C_t = \emptyset$ ,  $v_s(G_{u'}) \in C_s$  and  $v_t(G_{u'}) \in C_t$ . A set  $C_{st}$ ,  $C_s$  or  $C_t$  is not always a supplied set for  $G_u$  or  $G_{u'}$ , because it may not contain the supply vertex  $w$ .  $C_{st}$  is a “connected set” for  $G_u$ , that is,  $C_{st}$  induces a connected subgraph of  $G_u$ , while the pair  $(C_s, C_t)$  is a “separated pair of sets” for  $G_{u'}$ , that is,  $C_s$  and  $C_t$  induce vertex-disjoint connected subgraphs of  $G_{u'}$ . The set  $C$  contains no terminals of  $G_{u''}$  in Fig. 3(a). In such a case, we regard that  $\text{dem}(v_s(G_{u''})) = \text{dem}(v_t(G_{u''})) = 0$  and  $C$  induces a separated pair of singleton sets  $(C_s, C_t)$  such that  $C_s = \{v_s(G_{u''})\}$  and  $C_t = \{v_t(G_{u''})\}$ , as illustrated in Fig. 3(d).

If a set  $C_{st}$ ,  $C_s$  or  $C_t$  contains the supply vertex  $w$ , then the set may have the “marginal” power, the amount of which is no greater than  $\text{sup}(w)$ . If a set does not contain  $w$ , then the set may have the “deficient” power, the amount of which is no greater than  $\text{sup}(w)$ . Thus we later introduce five functions  $g$ ,  $h_1$ ,  $h_2$ ,  $h_3$  and  $h_4$ ; for a series-parallel graph  $G_u$  and a real number  $x$ , the value  $g(G_u, x)$  represents the maximum marginal power or the minimum deficient power of connected sets for  $G_u$ ; for a series-parallel graph  $G_u$  and a real number  $x$ , the value  $h_i(G_u, x)$ ,  $1 \leq i \leq 4$ , represents the maximum marginal power or the minimum deficient power of separated pairs of sets for  $G_u$ . Our idea is to compute  $g(G_u, x)$  and  $h_i(G_u, x)$ ,  $1 \leq i \leq 4$ , from the leaves of  $T$  to the root  $r$  of  $T$  by means of dynamic programming.

We now formally define the notion of connected sets and separated pair of sets for a series-parallel graph  $G$ . Let  $G_u = (V_u, E_u)$  be a subgraph of  $G$  for a node  $u$  of a binary decomposition tree  $T$  of  $G$ , and let  $v_s = v_s(G_u)$  and  $v_t = v_t(G_u)$ .

We call a set  $C \subseteq V_u$  a *connected set for  $G_u$*  if  $C$  satisfies the following three conditions (see Fig. 3(b)):

- (a)  $v_s, v_t \in C$ ;
- (b)  $C$  induces a connected subgraph of  $G_u$ ; and
- (c)  $\sum_{v \in C} \text{dem}(v) \leq \sup(w)$  if  $w \in C$ .

A pair of sets  $C_s, C_t \subseteq V_u$  is called a *separated pair (of sets) for  $G_u$*  if  $C_s$  and  $C_t$  satisfy the following four conditions (see Fig. 3(c)):

- (a)  $C_s \cap C_t = \emptyset$ ,  $v_s \in C_s$  and  $v_t \in C_t$ ;
- (b)  $C_s$  and  $C_t$  induce connected subgraphs of  $G_u$ ;
- (c)  $\sum_{v \in C_s} \text{dem}(v) \leq \sup(w)$  if  $w \in C_s$ ; and
- (d)  $\sum_{v \in C_t} \text{dem}(v) \leq \sup(w)$  if  $w \in C_t$ .

We then classify connected sets and separated pairs further into smaller classes. Let  $\mathbb{R}_w = \{x \in \mathbb{R} : |x| \leq \sup(w)\}$ , where  $\mathbb{R}$  denotes the set of all real numbers. For each real number  $i \in \mathbb{R}_w$ , we call a connected set  $C$  for  $G_u$  an *i-connected set* if  $C$  satisfies the following two conditions:

- (a) if  $i > 0$ , then  $w \in C$  and  $i + \sum_{x \in C} \text{dem}(x) \leq \sup(w)$ ; and
- (b) if  $i \leq 0$ , then  $w \notin C$  and  $\sum_{x \in C} \text{dem}(x) \leq |i| = -i$ .

An *i-connected set*  $C$  for  $G_u$  with  $i > 0$  is a supplied set for  $G_u$ , and hence corresponds to some supplied set  $C_r$  for the whole graph  $G = G_r$  such that  $w \in C \subseteq C_r$ , where  $r$  is the root of  $T$ ; an amount  $i$  of the remaining power of  $w$  can be delivered outside  $G_u$  through  $v_s$  or  $v_t$ ; and hence the “margin” of  $C$  is  $i$ . On the other hand, an *i-connected set*  $C$  for  $G_u$  with  $i \leq 0$  is not a supplied set for  $G_u$ , but may correspond to a supplied set  $C_r$  for  $G = G_r$  such that  $w \notin C \subset C_r$  and  $w \in C_r$ ; an amount  $|i|$  of power must be delivered to  $C$  through  $v_s$  or  $v_t$ , and hence the “deficiency” of  $C$  is  $|i|$ . For an *i-connected set*  $C$  for  $G_u$ , let

$$f(C, i) = \sum_{x \in C} \text{dem}(x).$$

Then  $f(C, i) = f(C)$  if  $0 < i \leq \sup(w)$ . On the other hand, if  $-\sup(w) \leq i \leq 0$ , then  $f(C, i)$  represents the fulfillment of  $C$  when an amount  $|i|$  of power is delivered to  $C$  from  $w$  in the outside of  $G_u$ .

Let  $\sigma \notin \mathbb{R}_w$  be a symbol. For each pair of  $j$  and  $k$  in  $\mathbb{R}_w \cup \{\sigma\}$ , we call a separated pair  $(C_s, C_t)$  for  $G_u$  a *(j, k)-separated pair* if  $(C_s, C_t)$  satisfies the following seven conditions:

- (a) if  $j > 0$ , then  $w \in C_s$  and  $j + \sum_{x \in C_s} \text{dem}(x) \leq \sup(w)$ ;
- (b) if  $j \leq 0$ , then  $w \notin C_s$  and  $\sum_{x \in C_s} \text{dem}(x) \leq -j$ ;
- (c) if  $j = \sigma$ , then  $C_s = \{v_s\}$ ;
- (d) if  $k > 0$ , then  $w \in C_t$  and  $k + \sum_{x \in C_t} \text{dem}(x) \leq \sup(w)$ ;
- (e) if  $k \leq 0$ , then  $w \notin C_t$  and  $\sum_{x \in C_t} \text{dem}(x) \leq -k$ ;
- (f) if  $k = \sigma$ , then  $C_t = \{v_t\}$ ; and
- (g) if  $j + k \leq 0$ , then  $j \leq 0$  and  $k \leq 0$ .

Since  $G$  has only one supply vertex  $w$ , there is no  $(j, k)$ -separated pair  $(C_s, C_t)$  for  $G$  such that  $j > 0$  and  $k > 0$ . A  $(j, k)$ -separated pair  $(C_s, C_t)$  for  $G_u$  with  $j > 0$  corresponds to a supplied set  $C_r$  for the whole graph  $G$  such that  $w \in C_s \subseteq C_r$ ; an amount  $j$  of the remaining power of  $w$  can be delivered outside  $C_s$  through  $v_s$ , and hence the margin of  $C_s$  is  $j$ . A  $(j, k)$ -separated pair  $(C_s, C_t)$  for  $G_u$  with  $j \leq 0$  may correspond to a supplied set  $C_r$  for  $G$  such that  $C_s \subset C_r$  and either  $w \in C_t$  or  $w \in C_r - C_s \cup C_t$ ; an amount  $|j|$  of power must be delivered to  $C_s$  through  $v_s$ , and hence the deficiency of  $C_s$  is  $|j|$ . A  $(j, k)$ -separated pair  $(C_s, C_t)$  for  $G_u$  with  $j = \sigma$  corresponds to a supplied set  $C_r$  for  $G$  such that  $v_s \notin C_r$ , that is,  $v_s$  is never supplied power. (See Figs. 3(a) and (d).) A  $(j, k)$ -separated pair  $(C_s, C_t)$  for  $G_u$  with  $k > 0$ ,  $k \leq 0$  or  $k = \sigma$  corresponds to a supplied set  $C_r$  for  $G$  similarly as above. For a  $(j, k)$ -separated pair  $(C_s, C_t)$  for  $G_u$ , let

$$f(C_s, C_t, j, k) = \begin{cases} \sum_{x \in C_s \cup C_t} \text{dem}(x) & \text{if } j, k \in \mathbb{R}_w; \\ \sum_{x \in C_s} \text{dem}(x) & \text{if } j \in \mathbb{R}_w \text{ and } k = \sigma; \text{ and} \\ \sum_{x \in C_t} \text{dem}(x) & \text{if } j = \sigma \text{ and } k \in \mathbb{R}_w. \end{cases}$$

Let

$$f(C_s, C_t, \sigma, \sigma) = \max\{f(C_u) \mid C_u \text{ is a supplied set for } G_u \\ \text{such that } v_s, v_t \notin C_u\};$$

let  $f(C_s, C_t, \sigma, \sigma) = 0$  if  $G_u$  has no supplied set  $C_u$  such that  $v_s, v_t \notin C_u$ .

We now formally define a function  $g$  as follows: for a series-parallel graph  $G_u$  and a real number  $x \in \mathbb{R}$ ,

$$g(G_u, x) = \max\{i \in \mathbb{R}_w \mid G_u \text{ has an } i\text{-connected set } C \text{ such that } f(C, i) \geq x\}.$$

If  $G_u$  has no  $i$ -connected set  $C$  with  $f(C, i) \geq x$  for any number  $i \in \mathbb{R}_w$ , then let  $g(G_u, x) = -\infty$ . We then formally define a function  $h_1$  as follows: for a series-parallel graph  $G_u$  and a real number  $x \in \mathbb{R}$ ,

$$h_1(G_u, x) = \max\{j + k \mid G_u \text{ has a } (j, k)\text{-separated pair } (C_s, C_t) \text{ such that} \\ j, k \in \mathbb{R}_w, |j + k| \leq \sup(w), \text{ and } f(C_s, C_t, j, k) \geq x\}.$$

If  $G_u$  has no  $(j, k)$ -separated pair  $(C_s, C_t)$  with  $f(C_s, C_t, j, k) \geq x$  for any pair of numbers  $j$  and  $k$  in  $\mathbb{R}_w$ , then let  $h_1(G_u, x) = -\infty$ . It should be noted that a  $(j, k)$ -separated pair  $(C_s, C_t)$  for  $G_u$  with  $j, k \in \mathbb{R}_w$  corresponds to a supplied set  $C_r$  for  $G$  such that  $C_s \cup C_t \subseteq C_r$ , and hence we can simply take the summation of  $j$  and  $k$  as the marginal power or the deficient power of  $C_s \cup C_t$ . We next formally define a function  $h_2$  as follows: for a series-parallel graph  $G_u$  and a real number  $x \in \mathbb{R}$ ,

$$h_2(G_u, x) = \max\{j \in \mathbb{R}_w \mid G_u \text{ has a } (j, \sigma)\text{-separated pair } (C_s, \{v_t\}) \\ \text{such that } f(C_s, \{v_t\}, j, \sigma) \geq x\}.$$

If  $G_u$  has no  $(j, \sigma)$ -separated pair  $(C_s, \{v_t\})$  with  $f(C_s, \{v_t\}, j, \sigma) \geq x$  for any number  $j \in \mathbb{R}_w$ , then let  $h_2(G_u, x) = -\infty$ . We then formally define a function  $h_3$  as follows: for a series-parallel graph  $G_u$  and a real number  $x \in \mathbb{R}$ ,



$$h_3(G_u, x) = \max\{k \in \mathbb{R}_w \mid G_u \text{ has a } (\sigma, k)\text{-separated pair } (\{v_s\}, C_t) \\ \text{such that } f(\{v_s\}, C_t, \sigma, k) \geq x\}.$$

If  $G_u$  has no  $(\sigma, k)$ -separated pair  $(\{v_s\}, C_t)$  with  $f(\{v_s\}, C_t, \sigma, k) \geq x$  for any number  $k \in \mathbb{R}_w$ , then let  $h_3(G_u, x) = -\infty$ . We finally define a function  $h_4$  as follows: for a series-parallel graph  $G_u$  and a real number  $x \in \mathbb{R}$ ,

$$h_4(G_u, x) = \begin{cases} 0 & \text{if } G_u \text{ has a } (\sigma, \sigma)\text{-separated pair } (\{v_s\}, \{v_t\}) \\ & \text{such that } f(\{v_s\}, \{v_t\}, \sigma, \sigma) \geq x; \\ -\infty & \text{otherwise.} \end{cases}$$

Our algorithm computes  $g(G_u, x)$  and  $h_i(G_u, x)$ ,  $1 \leq i \leq 4$ , for each node  $u$  of a binary decomposition tree  $T$  of a given series-parallel graph  $G$  from the leaves to the root  $r$  of  $T$  by means of a dynamic programming. Since  $G = G_r$ , one can compute the maximum fulfillment  $f(G)$  of  $G$  from  $g(G_r, x)$  and  $h_i(G_r, x)$ ,  $1 \leq i \leq 4$ . However, due to the page limitation, we omit the details of our algorithm.

We now show that our algorithm takes time  $O(F^2n)$ . Since all demands and the supply of vertices in a given series-parallel graph  $G$  are integers,  $f(C_u)$  is an integer for any supplied set  $C_u$  for  $G_u$ . Similarly,  $f(C, i)$  and  $f(C_s, C_t, j, k)$  are integers for any  $i$ -connected set  $C$  and any  $(j, k)$ -separated pair  $(C_s, C_t)$ , respectively. Then one can easily observe that it suffices to compute values  $g(G_u, x)$  and  $h_i(G_u, x)$ ,  $1 \leq i \leq 4$ , only for all integers  $x$  such that  $0 \leq x \leq F$ , because  $f(G) \leq F = \sum_{v \in V} \text{dem}(v)$ . We compute  $g(G_u, x)$  and  $h_i(G_u, x)$ ,  $1 \leq i \leq 4$ , for each internal node  $u$  of  $T$  from the counterparts of the two children of  $u$  in  $T$ . This is called *combining operation*, and can be done in time  $O(F^2)$ . Since  $T$  has at most  $2n - 4$  internal nodes, the combining operation is executed no more than  $2n$  times and hence one can compute  $g(G, x)$  and  $h_i(G, x)$ ,  $1 \leq i \leq 4$ , in time  $O(F^2n)$ . Our algorithm thus takes time  $O(F^2n)$ .

## 4 FPTAS

Assume in this section that the supply and all demands are positive real numbers which are not always integers. Since the maximum partition problem is MAXSNP-hard, there is no PTAS for the problem on general graphs unless  $P = NP$ . However, using the pseudo-polynomial-time algorithm in Section 3, we can obtain an FPTAS for series-parallel graphs having exactly one supply vertex, and have the following theorem.

**Theorem 3.** *There is a fully polynomial-time approximation scheme for the maximum partition problem on a series-parallel graph having exactly one supply vertex.*

We give an algorithm to find a supplied set  $C$  for a series-parallel graph  $G$  with  $f(C) \geq (1 - \varepsilon)f(G)$  in time polynomial in  $n$  and  $1/\varepsilon$  for any real number  $\varepsilon$ ,

$0 < \varepsilon < 1$ . Thus our approximate maximum fulfillent  $\bar{f}(G)$  of  $G$  is  $f(C)$ , and hence the error is bounded by  $\varepsilon f(G)$ , that is,

$$f(G) - \bar{f}(G) = f(G) - f(C) \leq \varepsilon f(G). \quad (2)$$

We now outline our algorithm and the analysis. We extend the ordinary “scaling and rounding” technique for the knapsack problem [3] and the maximum partition problem on trees [4] and apply it to the maximum partition problem for a series-parallel graph with a single supply vertex. For some scaling factor  $t$ , we consider the set  $\{\dots, -2t, -t, 0, t, 2t, \dots\}$  as the range of functions  $g$  and  $h_i$ ,  $1 \leq i \leq 4$ , and find the approximate solution  $\bar{f}(G)$  by using the pseudo-polynomial-time algorithm in Section 3. Then we have

$$f(G) - \bar{f}(G) < 4nt. \quad (3)$$

Intuitively, Eq. (3) holds because the combining operation is executed no more than  $2n$  times and each combining operation adds at most  $2t$  to the error  $f(G) - \bar{f}(G)$ . Let  $m_d$  be the maximum demand, that is,  $m_d = \max\{\text{dem}(v) \mid v \in V_d\}$ . Taking  $t = \varepsilon m_d / (4n)$  and claiming  $f(G) \geq m_d$ , we have Eq. (2). One can observe that the algorithm takes time

$$O\left(\left(\left\lceil \frac{F}{t} \right\rceil + 1\right)^2 n\right) = O\left(\frac{n^5}{\varepsilon^2}\right),$$

because  $F \leq nm_d$  and hence we have  $F/t \leq 4n^2/\varepsilon$ .

## Acknowledgments

We thank MohammadTaghi Hajiaghayi for fruitful discussions.

## References

1. S. Arnborg, J. Lagergren and D. Seese, Easy problems for tree-decomposable graphs, *J. Algorithms*, Vol. 12, pp. 308–340, 1991.
2. H. L. Bodlaender, Polynomial algorithms for graph isomorphism and chromatic index on partial  $k$ -trees, *J. Algorithms*, Vol. 11, pp. 631–643, 1990.
3. O. H. Ibarra and C. E. Kim, Fast approximation algorithms for the knapsack and sum of subset problems, *J. ACM*, Vol. 22, pp. 463–468, 1975.
4. T. Ito, X. Zhou and T. Nishizeki, Partitioning trees of supply and demand, *International J. of Foundations of Computer Science*, Vol. 16, pp. 803–827, 2005.
5. T. Ito, X. Zhou and T. Nishizeki, Partitioning graphs of supply and demand, *Proc. of the 2005 IEEE Int'l Symposium on Circuits and Syst.*, pp. 160–163, 2005.
6. A. B. Morton and I. M. Y. Mareels, An efficient brute-force solution to the network reconfiguration problem, *IEEE Trans. on Power Delivery*, Vol. 15, pp. 996–1000, 2000.
7. C. H. Papadimitriou, *Computational Complexity*, Addison-Wesley, 1994.
8. C. H. Papadimitriou and M. Yannakakis, Optimization, approximation, and complexity classes, *J. Computer and System Sciences*, Vol. 43, pp. 425–440, 1991.
9. K. Takamizawa, T. Nishizeki and N. Saito, Linear-time computability of combinatorial problems on series-parallel graphs, *J. ACM*, Vol. 29, pp. 623–641, 1982.