Partitioning a Multi-weighted Graph to Connected Subgraphs of Almost Uniform Size

Takehiro Ito, Kazuya Goto, Xiao Zhou, and Takao Nishizeki

Graduate School of Information Sciences, Tohoku University,
Aoba-yama 6-6-05, Sendai, 980-8579, Japan
{take, kazzg}@nishizeki.ecei.tohoku.ac.jp, {zhou, nishi}@ecei.tohoku.ac.jp

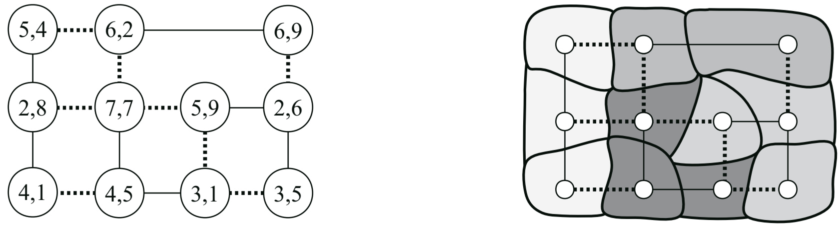
Abstract. Assume that each vertex of a graph G is assigned a constant number q of nonnegative integer weights, and that q pairs of nonnegative integers l_i and u_i, $1 \leq i \leq q$, are given. One wishes to partition G into connected components by deleting edges from G so that the total i-th weights of all vertices in each component is at least l_i and at most u_i for each index i, $1 \leq i \leq q$. The problem of finding such a "uniform" partition is NP-hard for series-parallel graphs, and is strongly NP-hard for general graphs even for $q = 1$. In this paper we show that the problem and many variants can be solved in pseudo-polynomial time for series-parallel graphs. Our algorithms for series-parallel graphs can be extended for partial k-trees, that is, graphs with bounded tree-width.

1 Introduction

Let $G = (V, E)$ be an undirected graph with vertex set V and edge set E. Assume that each vertex $v \in V$ is assigned a constant number q of nonnegative integer weights $\omega_1(v), \omega_2(v), \cdots, \omega_q(v)$, and that q pairs of nonnegative integers l_i and u_i, $1 \leq i \leq q$, are given. We call $\omega_i(v)$ the *i-th weight* of vertex v, and call l_i and u_i the i-th *lower bound* and *upper bound* on component size, respectively. We wish to partition G into connected components by deleting edges from G so that the total i-th weights of all components are almost uniform for each index i, $1 \leq i \leq q$, that is, the sum of i-th weights $\omega_i(v)$ of all vertices v in each component is at least l_i and at most u_i for some bounds l_i and u_i with small $u_i - l_i$. We call such a partition a *uniform partition* of G. Figure 1(a) illustrates a uniform partition of a graph, where $q = 2$, $(l_1, u_1) = (10, 15)$, $(l_2, u_2) = (10, 20)$, each vertex v is drawn as a circle, the two weights $\omega_1(v)$ and $\omega_2(v)$ of v are written inside the circle, and the deleted edges are drawn by dotted lines.

The problem of finding a uniform partition often appear in many practical situations such as image processing [4,6], paging systems of operation systems [8], and political districting [3,9]. Consider, for example, political districting. Let M be a map of a country, which is divided into several regions, as illustrated in Fig. 1(b). Let G be a dual-like graph of the map M, as illustrated in Fig. 1(a). Each vertex v of G represents a region, the first weight $\omega_1(v)$ represents the number of voters in the region v, and the second weight $\omega_2(v)$ represents the

D.Z. Chen and D.T. Lee (Eds.): COCOON 2006, LNCS 4112, pp. 63–72, 2006.



(a) (b)

Fig. 1. (a) A uniform partition of a graph into $p = 4$ components, and (b) electoral zoning of a map corresponding to the partition

area of the region. Each edge (u, v) of G represents the adjacency of the two regions u and v. For the political districting, one wishes to divide the country into electoral zones. Each zone must consist of connected regions, that is, the regions in each zone must induce a connected subgraph of G. Each zone must have an almost equal number of voters, and must be almost equal in area. Such electoral zoning corresponds to a uniform partition of the plane graph G for two appropriate pairs (l_1, u_1) and (l_2, u_2) of bounds.

In the paper we deal with the following three problems to find a uniform partition of a given graph G: the *minimum partition problem* is to find a uniform partition of G with the minimum number of components; the *maximum partition problem* is defined similarly; and the *p-partition problem* is to find a uniform partition of G with a given number p of components. All the problems are NP-hard for series-parallel graphs even when $q = 1$ [5]. Therefore, it is very unlikely that the three partition problems can be solved in polynomial time even for series-parallel graphs. Moreover, all the three partition problems are strongly NP-hard for general graphs even if $q = 1$ [5], and hence there is no pseudo-polynomial-time algorithm for any of the three problems on general graphs unless P = NP. Furthermore, for any $\varepsilon > 0$, there is no ε-approximation algorithm for the minimum partition problem or the maximum partition problem on series-parallel graphs unless P = NP [5], and the problems for the case $q = 1$ can be solved in pseudo-polynomial time for series-parallel graphs [5]; the minimum and maximum partition problems can be solved in time $O(u_1^4 n)$ and the p-partition problem can be solved in time $O(p^2 u_1^4 n)$ for series-parallel graphs G, where n is the number of vertices in G. However, it has not been known whether the problems can be solved in pseudo-polynomial time for the case $q \geq 2$.

In this paper, we obtain pseudo-polynomial-time algorithms to solve the three problems on series-parallel graphs for an arbitrary constant number q. More precisely, we show that the minimum and maximum partition problems can be solved in time $O(u^{4q} n)$ and hence in time $O(n)$ for any fixed constant u, and that the p-partition problem can be solved in time $O(p^2 u^{4q} n)$, where u is the maximum upper bound, that is, $u = \max\{u_i \mid 1 \leq i \leq q\}$. Our algorithms for series-parallel graphs can be extended for partial k-trees, that is, graphs with bounded tree-width [1,2].

2 Terminology and Definitions

In this section we give some definitions.

A (*two-terminal*) *series-parallel graph* is defined recursively as follows [7]:

(1) A graph G with a single edge is a series-parallel graph. The end vertices of the edge are called the *terminals* of G and denoted by $s(G)$ and $t(G)$. (See Fig. 2(a).)

(2) Let G' be a series-parallel graph with terminals $s(G')$ and $t(G')$, and let G'' be a series-parallel graph with terminals $s(G'')$ and $t(G'')$.

   (a) A graph G obtained from G' and G'' by identifying vertex $t(G')$ with vertex $s(G'')$ is a series-parallel graph, whose terminals are $s(G) = s(G')$ and $t(G) = t(G'')$. Such a connection is called a *series connection*, and G is denoted by $G = G' \bullet G''$. (See Fig. 2(b).)

   (b) A graph G obtained from G' and G'' by identifying $s(G')$ with $s(G'')$ and identifying $t(G')$ with $t(G'')$ is a series-parallel graph, whose terminals are $s(G) = s(G') = s(G'')$ and $t(G) = t(G') = t(G'')$. Such a connection is called a *parallel connection*, and G is denoted by $G = G' \parallel G''$. (See Fig. 2(c).)

The terminals $s(G)$ and $t(G)$ of G are often denoted simply by s and t, respectively. Since we deal with partition problems, we may assume without loss of generality that G is a simple graph and hence G has no multiple edges.

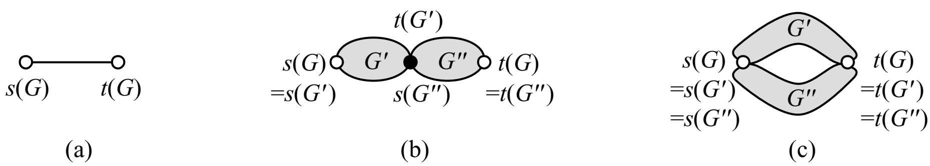


Fig. 2. (a) A series-parallel graph with a single edge, (b) series connection, and (c) parallel connection

A series-parallel graph G can be represented by a "binary decomposition tree" [7]. Figure 3(a) illustrates a series-parallel graph G, and Figure 3(b) depicts a binary decomposition tree T of G. Labels s and p attached to internal nodes in T indicate series and parallel connections, respectively. Nodes labeled s and p are called *s-* and *p-nodes*, respectively. Every leaf of T represents a subgraph of G induced by a single edge. Each node v of T corresponds to a subgraph G_v of G induced by all edges represented by the leaves that are descendants of v in T. Thus G_v is a series-parallel graph for each node v of T, and $G = G_r$ for the root r of T. Figure 3(c) depicts G_v for the left child v of the root r of T. Since a binary decomposition tree of a given series-parallel graph G can be found in linear time [7], we may assume that a series-parallel graph G and its binary decomposition tree T are given. We solve the three partition problems by a dynamic programming approach based on a decomposition tree T.

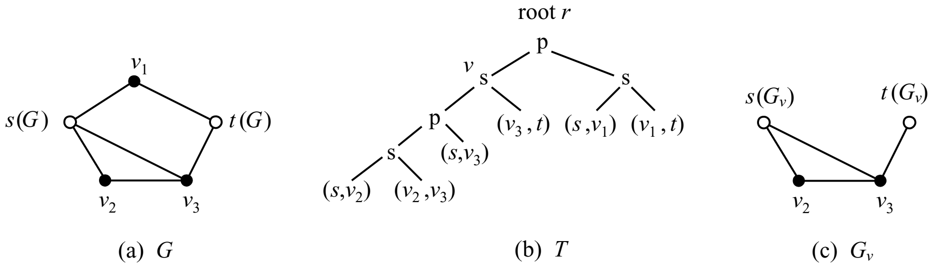


Fig. 3. (a) A series-parallel graph G, (b) its binary decomposition tree T, and (c) a subgraph G_v

3 Minimum and Maximum Partition Problems

In this section we have the following theorem.

Theorem 1. *Both the minimum partition problem and the maximum partition problem can be solved for any series-parallel graph G in time $O(u^{4q}n)$, where n is the number of vertices in G, q is a fixed constant number of weights, and u is the maximum upper bound on component size.*

In the remainder of this section we give an algorithm to solve the minimum partition problem as a proof of Theorem 1; the maximum partition problem can be similarly solved. We indeed show only how to compute the minimum number $p_{\min}(G)$ of components. It is easy to modify our algorithm so that it actually finds a uniform partition having the minimum number $p_{\min}(G)$ of components.

Every uniform partition of a series-parallel graph G naturally induces a partition of its subgraph G_v for a node v of a decomposition tree T of G. The induced partition is not always a uniform partition of G_v but is either a "connected partition" or a "separated partition" of G_v, which will be formally defined later and are illustrated in Fig. 4 where s and t represent the terminals of G_v. We denote by $\mathbb{X}$ a q-tuple $(x_1, x_2, \cdots, x_q)$ of integers with $0 \le x_i \le u_i$, $1 \le i \le q$. We introduce two functions f and h; for a series-parallel graph G_v and a q-tuple $\mathbb{X} = (x_1, x_2, \cdots, x_q)$, the value $f(G_v, \mathbb{X})$ represents the minimum number of components in some particular connected partitions of G_v; for a series-parallel graph G_v and a pair of q-tuples $\mathbb{X} = (x_1, x_2, \cdots, x_q)$ and $\mathbb{Y} = (y_1, y_2, \cdots, y_q)$, the value $h(G_v, \mathbb{X}, \mathbb{Y})$ represents the minimum number of components in some particular separated partitions of G_v. Our idea is to compute $f(G_v, \mathbb{X})$ and $h(G_v, \mathbb{X}, \mathbb{Y})$ from leaves of T to the root r of T by means of dynamic programming.

Fig. 4. (a) A connected partition, and (b) a separated partition

We now formally define the notion of connected and separated partitions of a series-parallel graph $G = (V, E)$. Let $\mathcal{P} = \{P_1, P_2, \ldots, P_m\}$ be a partition of the vertex set V of G into m nonempty subsets $P_1, P_2, \cdots, P_m$ for some integer $m \geq 1$. Thus $|\mathcal{P}| = m$. The partition $\mathcal{P}$ of V is called a *partition of* G if P_j induces a connected subgraph of G for each index j, $1 \leq j \leq m$. For a set $P \subseteq V$ and an index i, $1 \leq i \leq q$, we denote by $\omega_i(P)$ the sum of i-th weights of vertices in P, that is, $\omega_i(P) = \sum_{v \in P} \omega_i(v)$. Let $\omega_{st}(G, i) = \omega_i(s) + \omega_i(t)$. We call a partition $\mathcal{P}$ of G a *connected partition* if $\mathcal{P}$ satisfies the following two conditions (see Fig. 4(a)):

(a) there exists a set $P_{st} \in \mathcal{P}$ such that $s, t \in P_{st}$; and
(b) for each index i, $1 \leq i \leq q$, the inequality $\omega_i(P_{st}) \leq u_i$ holds, and the inequalities $l_i \leq \omega_i(P) \leq u_i$ hold for each set $P \in \mathcal{P} - \{P_{st}\}$.

Note that the inequality $l_i \leq \omega_i(P_{st})$, $1 \leq i \leq q$, does not necessarily hold for P_{st}. For a connected partition $\mathcal{P}$, we always denote by P_{st} the set in $\mathcal{P}$ containing both s and t. A partition $\mathcal{P}$ of G is called a *separated partition* if $\mathcal{P}$ satisfies the following two conditions (see Fig. 4(b)):

(a) there exist two distinct sets $P_s, P_t \in \mathcal{P}$ such that $s \in P_s$ and $t \in P_t$; and
(b) for each index i, $1 \leq i \leq q$, the two inequalities $\omega_i(P_s) \leq u_i$ and $\omega_i(P_t) \leq u_i$ hold, and the inequalities $l_i \leq \omega_i(P) \leq u_i$ hold for each set $P \in \mathcal{P} - \{P_s, P_t\}$.

Note that the inequalities $l_i \leq \omega_i(P_s)$ and $l_i \leq \omega_i(P_t)$, $1 \leq i \leq q$, do not always hold for P_s and P_t. For a separated partition $\mathcal{P}$, we always denote by P_s the set in $\mathcal{P}$ containing s and by P_t the set in $\mathcal{P}$ containing t.

We then formally define $f(G, \mathbb{X})$ for a series-parallel graph G and a q-tuple $\mathbb{X} = (x_1, x_2, \cdots, x_q)$ of integers with $0 \leq x_i \leq u_i$, $1 \leq i \leq q$, as follows:

$$\begin{aligned} f(G, \mathbb{X}) = \min\{p^* \geq 0 \mid\ & G \text{ has a connected partition } \mathcal{P} \text{ such that} \\ & x_i = \omega_i(P_{st}) - \omega_{st}(G, i) \text{ for each } i, \text{ and } p^* = |\mathcal{P}| - 1\}. \end{aligned} \tag{1}$$

If G has no connected partition $\mathcal{P}$ such that $\omega_i(P_{st}) - \omega_{st}(G, i) = x_i$ for each i, then let $f(G, \mathbb{X}) = +\infty$.

We now formally define $h(G, \mathbb{X}, \mathbb{Y})$ for a series-parallel graph G and a pair of q-tuples $\mathbb{X} = (x_1, x_2, \cdots, x_q)$ and $\mathbb{Y} = (y_1, y_2, \cdots, y_q)$ of integers with $0 \leq x_i, y_i \leq u_i$, $1 \leq i \leq q$, as follows:

$$\begin{aligned} h(G, \mathbb{X}, \mathbb{Y}) = \min\{p^* \geq 0 \mid\ & G \text{ has a separated partition } \mathcal{P} \text{ such that} \\ & x_i = \omega_i(P_s) - \omega_i(s) \text{ and } y_i = \omega_i(P_t) - \omega_i(t) \text{ for each } i, \\ & \text{and } p^* = |\mathcal{P}| - 2\}. \end{aligned} \tag{2}$$

If G has no separated partition $\mathcal{P}$ such that $\omega_i(P_s) - \omega_i(s) = x_i$ and $\omega_i(P_t) - \omega_i(t) = y_i$ for each i, then let $h(G, \mathbb{X}, \mathbb{Y}) = +\infty$.

Our algorithm computes $f(G_v, \mathbb{X})$ and $h(G_v, \mathbb{X}, \mathbb{Y})$ for each node v of a binary decomposition tree T of a given series-parallel graph G from leaves to the root r of T by means of dynamic programming. Since $G = G_r$, one can compute the minimum number $p_{\min}(G)$ of components from $f(G,\mathbb{X})$ and $h(G,\mathbb{X},\mathbb{Y})$ as follows:

$$\begin{aligned} p_{\min}(G) = \min\Big\{ & \min\{f(G,\mathbb{X})+1 \mid l_i \le x_i + \omega_{st}(G,i) \le u_i \text{ for each } i\}, \\ & \min\{h(G,\mathbb{X},\mathbb{Y})+2 \mid l_i \le x_i + \omega_i(s) \le u_i \text{ and} \\ & \qquad l_i \le y_i + \omega_i(t) \le u_i \text{ for each } i\}\Big\}. \end{aligned} \quad (3)$$

Note that $p_{\min}(G) = +\infty$ if G has no uniform partition.

We first compute $f(G_v,\mathbb{X})$ and $h(G_v,\mathbb{X},\mathbb{Y})$ for each leaf v of T, for which the subgraph G_v contains exactly one edge. We thus have

$$f(G_v,\mathbb{X}) = \begin{cases} 0 & \text{if } \mathbb{X} = (0,0,\cdots,0); \\ +\infty & \text{otherwise,} \end{cases} \quad (4)$$

and

$$h(G_v,\mathbb{X},\mathbb{Y}) = \begin{cases} 0 & \text{if } \mathbb{X} = \mathbb{Y} = (0,0,\cdots,0); \\ +\infty & \text{otherwise.} \end{cases} \quad (5)$$

By Eq. (4) one can compute $f(G_v,\mathbb{X})$ in time $O(u^q)$ for each leaf v of T and all q-tuples $\mathbb{X} = (x_1,x_2,\cdots,x_q)$, where u is the maximum upper bound on component size, that is, $u = \max\{u_i \mid 1 \le i \le q\}$. Similarly, by Eq. (5) one can compute $h(G_v,\mathbb{X},\mathbb{Y})$ in time $O(u^{2q})$ for each leaf v and all pairs of q-tuples $\mathbb{X} = (x_1,x_2,\cdots,x_q)$ and $\mathbb{Y} = (y_1,y_2,\cdots,y_q)$. Since G is a simple series-parallel graph, the number of edges in G is at most $2n-3$ and hence the number of leaves in T is at most $2n-3$. Thus one can compute $f(G_v,\mathbb{X})$ and $h(G_v,\mathbb{X},\mathbb{Y})$ for all leaves v of T in time $O(u^{2q}n)$.

We next compute $f(G_v,\mathbb{X})$ and $h(G_v,\mathbb{X},\mathbb{Y})$ for each internal node v of T from the counterparts of the two children of v in T.

We first consider a parallel connection.

[Parallel connection]

Let $G_v = G' \parallel G''$, and let $s = s(G_v)$ and $t = t(G_v)$. (See Figs. 2(c) and 5.)

We first explain how to compute $h(G_v,\mathbb{X},\mathbb{Y})$ from $h(G',\mathbb{X}',\mathbb{Y}')$ and $h(G'',\mathbb{X}'',\mathbb{Y}'')$. The definitions of a separated partition and $h(G,\mathbb{X},\mathbb{Y})$ imply that if $\omega_i(P_s) = x_i + \omega_i(s) > u_i$ or $\omega_i(P_t) = y_i + \omega_i(t) > u_i$ for some index i, then $h(G_v,\mathbb{X},\mathbb{Y}) = +\infty$. One may thus assume that $x_i + \omega_i(s) \le u_i$ and $y_i + \omega_i(t) \le u_i$ for each index i, $1 \le i \le q$. Then every separated partition $\mathcal{P}$ of G_v can be obtained by combining a separated partition $\mathcal{P}'$ of G' with a separated partition $\mathcal{P}''$ of G'', as illustrated in Fig. 5(a). We thus have

$$\begin{aligned} h(G_v,\mathbb{X},\mathbb{Y}) = \min\{ & h(G',\mathbb{X}',\mathbb{Y}') + h(G'',\mathbb{X}-\mathbb{X}',\mathbb{Y}-\mathbb{Y}') \mid \\ & \mathbb{X}' = (x'_1,x'_2,\cdots,x'_q) \text{ and } \mathbb{Y}' = (y'_1,y'_2,\cdots,y'_q) \\ & \text{such that } 0 \le x'_i, y'_i \le u_i \text{ for each } i\}, \end{aligned} \quad (6)$$

where $\mathbb{X}-\mathbb{X}' = (x_1-x'_1, x_2-x'_2, \cdots, x_q-x'_q)$ and $\mathbb{Y}-\mathbb{Y}' = (y_1-y'_1, y_2-y'_2, \cdots, y_q-y'_q)$.

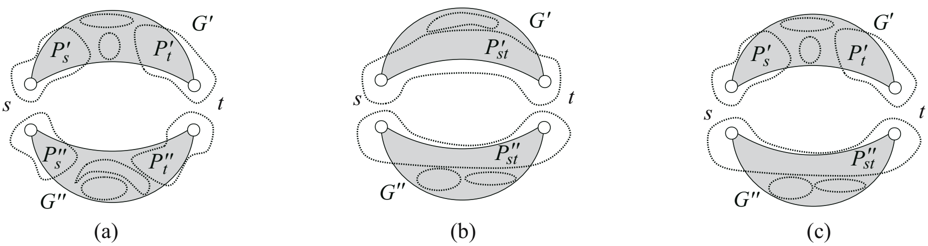


Fig. 5. The combinations of a partition $\mathcal{P}'$ of G' and a partition $\mathcal{P}''$ of G'' for a partition $\mathcal{P}$ of $G_v = G' \parallel G''$

We next explain how to compute $f(G_v, \mathbb{X})$ from $f(G', \mathbb{X}')$, $f(G'', \mathbb{X}'')$, $h(G', \mathbb{X}', \mathbb{Y}')$ and $h(G'', \mathbb{X}'', \mathbb{Y}'')$. If $\omega_i(P_{st}) = x_i + \omega_{st}(G_v, i) > u_i$ for some index i, then $f(G_v, \mathbb{X}) = +\infty$. One may thus assume that $x_i + \omega_{st}(G_v, i) \le u_i$ for each index i, $1 \le i \le q$. Then every connected partition $\mathcal{P}$ of G_v can be obtained by combining a partition $\mathcal{P}'$ of G' with a partition $\mathcal{P}''$ of G'', as illustrated in Figs. 5(b) and (c). There are the following two Cases (a) and (b), and we define two functions f^a and f^b for the two cases, respectively.

Case (*a*): *both $\mathcal{P}'$ and $\mathcal{P}''$ are connected partitions.* (See Fig. 5(b).)

Let

$$f^a(G_v, \mathbb{X}) = \min\{f(G', \mathbb{X}') + f(G'', \mathbb{X} - \mathbb{X}') \mid \mathbb{X}' = (x'_1, x'_2, \cdots, x'_q) \text{ such that } 0 \le x'_i \le u_i \text{ for each } i\}. \quad (7)$$

Case (*b*): *one of $\mathcal{P}'$ and $\mathcal{P}''$ is a separated partition and the other is a connected partition.*

One may assume without loss of generality that $\mathcal{P}'$ is a separated partition and $\mathcal{P}''$ is a connected partition. (See Fig. 5(c).) Let

$$f^b(G_v, \mathbb{X}) = \min\{h(G', \mathbb{X}', \mathbb{Y}') + f(G'', \mathbb{X} - \mathbb{X}' - \mathbb{Y}') \mid \mathbb{X}' = (x'_1, x'_2, \cdots, x'_q) \text{ and } \mathbb{Y}' = (y'_1, y'_2, \cdots, y'_q) \text{ such that } 0 \le x'_i, y'_i \le u_i \text{ for each } i\}. \quad (8)$$

From f^a and f^b above, one can compute $f(G_v, \mathbb{X})$ as follows:

$$f(G_v, \mathbb{X}) = \min\{f^a(G_v, \mathbb{X}), f^b(G_v, \mathbb{X})\}. \quad (9)$$

By Eq. (6) one can compute $h(G_v, \mathbb{X}, \mathbb{Y})$ in time $O(u^{4q})$ for all pairs of q-tuples $\mathbb{X} = (x_1, x_2, \cdots, x_q)$ and $\mathbb{Y} = (y_1, y_2, \cdots, y_q)$ with $0 \le x_i, y_i \le u_i$, $1 \le i \le q$. By Eqs. (7)–(9) one can compute $f(G_v, \mathbb{X})$ in time $O(u^{3q})$ for all q-tuples $\mathbb{X} = (x_1, x_2, \cdots, x_q)$ with $0 \le x_i \le u_i$, $1 \le i \le q$. Thus one can compute $f(G_v, \mathbb{X})$ and $h(G_v, \mathbb{X}, \mathbb{Y})$ for each p-node v of T in time $O(u^{4q})$.

We next consider a series connection.

[Series connection]

Let $G_v = G' \bullet G''$, and let w be the vertex of G identified by the series connection, that is, $w = t(G') = s(G'')$. (See Figs. 2(b) and 6.)

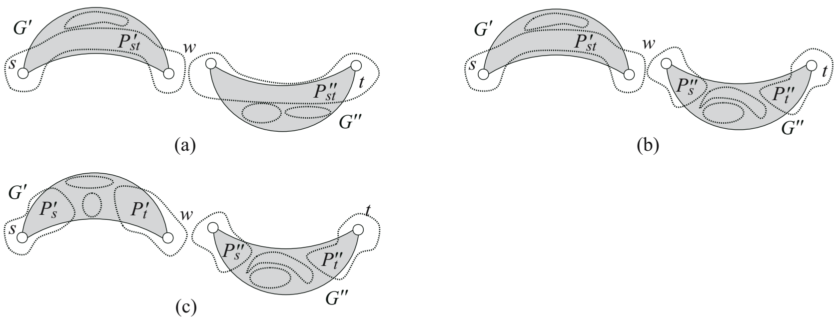


Fig. 6. The combinations of a partition $\mathcal{P}'$ of G' and a partition $\mathcal{P}''$ of G'' for a partition $\mathcal{P}$ of $G_v = G' \bullet G''$

We first explain how to compute $f(G_v, \mathbb{X})$. If $x_i + \omega_{st}(G_v, i) > u_i$ for some index i, then $f(G_v, \mathbb{X}) = +\infty$. One may thus assume that $x_i + \omega_{st}(G_v, i) \leq u_i$ for each index i, $1 \leq i \leq q$. Then every connected partition $\mathcal{P}$ of G_v can be obtained by combining a connected partition $\mathcal{P}'$ of G' with a connected partition $\mathcal{P}''$ of G'', as illustrated in Fig. 6(a). We thus have

$$\begin{aligned} f(G_v, \mathbb{X}) = \min\{ f(G', \mathbb{X}') + f(G'', \mathbb{X}'') \mid \mathbb{X}' = (x'_1, x'_2, \cdots, x'_q) \text{ and} \\ \mathbb{X}'' = (x''_1, x''_2, \cdots, x''_q) \text{ such that } 0 \leq x'_i, x''_i \leq u_i \text{ and} \\ x'_i + x''_i + \omega_i(w) = x_i \text{ for each } i\}. \end{aligned} \tag{10}$$

We next explain how to compute $h(G_v, \mathbb{X}, \mathbb{Y})$. If $x_i + \omega_i(s) > u_i$ or $y_i + \omega_i(t) > u_i$ for some index i, then $h(G_v, \mathbb{X}, \mathbb{Y}) = +\infty$. One may thus assume that $x_i + \omega_i(s) \leq u_i$ and $y_i + \omega_i(t) \leq u_i$ for each index i, $1 \leq i \leq q$. Then every separated partition $\mathcal{P}$ of G_v can be obtained by combining a partition $\mathcal{P}'$ of G' with a partition $\mathcal{P}''$ of G'', as illustrated in Figs. 6(b) and (c). There are the following two Cases (a) and (b), and we define two functions h^a and h^b for the two cases, respectively.

Case (a): one of $\mathcal{P}'$ and $\mathcal{P}''$ is a connected partition and the other is a separated partition.

One may assume without loss of generality that $\mathcal{P}'$ is a connected partition and $\mathcal{P}''$ is a separated partition. (See Fig. 6(b).) Let

$$\begin{aligned} h^a(G_v, \mathbb{X}, \mathbb{Y}) = \min\{ f(G', \mathbb{X}') + h(G'', \mathbb{X}'', \mathbb{Y}) \mid \mathbb{X}' = (x'_1, x'_2, \cdots, x'_q) \text{ and} \\ \mathbb{X}'' = (x''_1, x''_2, \cdots, x''_q) \text{ such that } 0 \leq x'_i, x''_i \leq u_i \text{ and} \\ x'_i + x''_i + \omega_i(w) = x_i \text{ for each } i\}. \end{aligned} \tag{11}$$

Case (b): both $\mathcal{P}'$ and $\mathcal{P}''$ are separated partitions. (See Fig. 6(c).)

Let

$$\begin{aligned} h^b(G_v, \mathbb{X}, \mathbb{Y}) = \min\{ & h(G', \mathbb{X}, \mathbb{Y}') + h(G'', \mathbb{X}'', \mathbb{Y}) + 1 \mid \\ & \mathbb{Y}' = (y'_1, y'_2, \cdots, y'_q) \text{ and } \mathbb{X}'' = (x''_1, x''_2, \cdots, x''_q) \\ & \text{such that } 0 \leq y'_i, x''_i \leq u_i \text{ and} \\ & l_i \leq y'_i + x''_i + \omega_i(w) \leq u_i \text{ for each } i\}. \end{aligned} \tag{12}$$

From h^a and h^b above one can compute $h(G_v, \mathbb{X}, \mathbb{Y})$ as follows:

$$h(G_v, \mathbb{X}, \mathbb{Y}) = \min\{h^a(G_v, \mathbb{X}, \mathbb{Y}), h^b(G_v, \mathbb{X}, \mathbb{Y})\}. \tag{13}$$

By Eq. (10) one can compute $f(G_v, \mathbb{X})$ in time $O(u^{2q})$ for all q-tuples $\mathbb{X} = (x_1, x_2, \cdots, x_q)$ with $0 \le x_i \le u_i$, $1 \le i \le q$. By Eqs. (11)–(13) one can compute $h(G_v, \mathbb{X}, \mathbb{Y})$ in time $O(u^{4q})$ for all pairs of q-tuples $\mathbb{X} = (x_1, x_2, \cdots, x_q)$ and $\mathbb{Y} = (y_1, y_2, \cdots, y_q)$ with $0 \le x_i, y_i \le u_i$, $1 \le i \le q$. Thus one can compute $f(G_v, \mathbb{X})$ and $h(G_v, \mathbb{X}, \mathbb{Y})$ for each s-node v of T in time $O(u^{4q})$.

In this way one can compute $f(G_v, \mathbb{X})$ and $h(G_v, \mathbb{X}, \mathbb{Y})$ for each internal node v of T in time $O(u^{4q})$ regardless of whether v is a p-node or an s-node. Since T is a binary tree and has at most $2n-3$ leaves, T has at most $2n-4$ internal nodes. Since $G = G_r$ for the root r of T, one can compute $f(G, \mathbb{X})$ and $h(G, \mathbb{X}, \mathbb{Y})$ in time $O(u^{4q}n)$. By Eq. (3) one can compute the minimum number $p_{\min}(G)$ of components in a uniform partition of G from $f(G, \mathbb{X})$ and $h(G, \mathbb{X}, \mathbb{Y})$ in time $O(u^{2q})$. Thus the minimum partition problem can be solved in time $O(u^{4q}n)$. This completes our proof of Theorem 1.

4 *p*-Partition Problem

In this section we have the following theorem.

Theorem 2. *The p-partition problem can be solved for any series-parallel graph G in time $O(p^2u^{4q}n)$, where n is the number of vertices in G, q is a fixed constant number of weights, u is the maximum upper bound on component size, and p is a given number of components.*

The algorithm for the p-partition problem is similar to the algorithm for the minimum partition problem in the previous section. So we present only an outline.

For a series-parallel graph G and an integer p^*, $0 \le p^* \le p-1$, we define a set $F(G, p^*)$ of q-tuples $\mathbb{X} = (x_1, x_2, \cdots, x_q)$ as follows:

$$\begin{aligned} F(G, p^*) = \{\mathbb{X} \mid\ & G \text{ has a connected partition } \mathcal{P} \text{ such that} \\ & x_i = \omega_i(P_{st}) - \omega_{st}(G, i) \text{ for each } i, \text{ and } p^* = |\mathcal{P}| - 1\}. \end{aligned}$$

For a series-parallel graph G and an integer p^*, $0 \le p^* \le p-2$, we define a set $H(G, p^*)$ of pairs of q-tuples $\mathbb{X} = (x_1, x_2, \cdots, x_q)$ and $\mathbb{Y} = (y_1, y_2, \cdots, y_q)$ as follows:

$$\begin{aligned} H(G, p^*) = \{(\mathbb{X}, \mathbb{Y}) \mid\ & G \text{ has a separated partition } \mathcal{P} \text{ such that} \\ & x_i = \omega_i(P_s) - \omega_i(s) \text{ and } y_i = \omega_i(P_s) - \omega_i(t) \text{ for each } i, \\ & \text{and } p^* = |\mathcal{P}| - 2\}. \end{aligned}$$

Clearly $|F(G, p^*)| \le (u+1)^q$ and $|H(G, p^*)| \le (u+1)^{2q}$.

We compute $F(G_v, p^*)$ and $H(G_v, p^*)$ for each node v of a binary decomposition tree T of a given series-parallel graph G from leaves to the root r of T by means of dynamic programming. Since $G = G_r$, the following lemma clearly holds.

Lemma 1. *A series-parallel graph G has a uniform partition with p components if and only if the following condition* (a) *or* (b) *holds*:

(a) $F(G, p-1)$ *contains at least one q-tuple* $\mathbb{X} = (x_1, x_2, \cdots, x_q)$ *such that* $l_i \leq x_i + \omega_{st}(G, i) \leq u_i$ *for each index i,* $1 \leq i \leq q$; *and*

(b) $H(G, p-2)$ *contains at least one pair of q-tuples* $\mathbb{X} = (x_1, x_2, \cdots, x_q)$ *and* $\mathbb{Y} = (y_1, y_2, \cdots, y_q)$ *such that* $l_i \leq x_i + \omega_i(s) \leq u_i$ *and* $l_i \leq y_i + \omega_i(t) \leq u_i$ *for each index i,* $1 \leq i \leq q$.

One can compute in time $O(p)$ the sets $F(G_v, p^*)$ and $H(G_v, p^*)$ for each leaf v of T and all integers p^* $(\leq p-1)$, and compute in time $O(p^2 u^{4q})$ the sets $F(G_v, p^*)$ and $H(G_v, p^*)$ for each internal node v of T and all integers p^* $(\leq p-1)$ from the counterparts of the two children of v in T. Since $G = G_r$ for the root r of T, one can compute the sets $F(G, p-1)$ and $H(G, p-2)$ in time $O(p^2 u^{4q} n)$. By Lemma 1 one can know from the sets in time $O(u^{2q})$ whether G has a uniform partition with p components. Thus the p-partition problem can be solved in time $O(p^2 u^{4q} n)$.

5 Conclusions

In this paper we obtained pseudo-polynomial-time algorithms to solve the three uniform partition problems for series-parallel graphs. Both the minimum partition problem and the maximum partition problem can be solved in time $O(u^{4q} n)$. On the other hand, the p-partition problem can be solved in time $O(p^2 u^{4q} n)$.

One can observe that the algorithms for series-parallel graphs can be extended for partial k-trees, that is, graphs with bounded tree-width [1,2].

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. B. Bozkaya, E. Erkut and G. Laporte, A tabu search heuristic and adaptive memory procedure for political districting, *European J. Operational Research*, Vol. 144, pp. 12–26, 2003.
4. R. C. Gonzalez and P. Wintz, Digital Image Processing, *Addison-Wesley, Reading, MA*, 1977.
5. T. Ito, X. Zhou and T. Nishizeki, Partitioning a graph of bounded tree-width to connected subgraphs of almost uniform size, *J. Discrete Algorithms*, Vol. 4, pp. 142–154, 2006.
6. M. Lucertini, Y. Perl and B. Simeone, Most uniform path partitioning and its use in image processing, *Discrete Applied Mathematics*, Vol. 42, pp. 227–256, 1993.
7. 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.
8. D. C. Tsichritzis and P. A. Bernstein, Operating Systems, *Academic Press, New York*, 1974.
9. J. C. Williams Jr., Political redistricting: a review, *Papers in Regional Science*, Vol. 74, pp. 13–40, 1995.