Linear-time Algorithms for Multiterminal Flows in Trees

Mingyu Xiao
School of Computer Science and Engineering, University of Electronic Science and Technology of China, China,
myxiao@gmail.com

Hiroshi Nagamochi
Department of Applied Mathematics and Physics, Graduate School of Informatics, Kyoto University, Japan,
nag@amp.i.kyoto-u.ac.jp

Abstract

In this paper, we study the problem of finding a multiflow which maximizes the sum of flow values between every two terminals in an undirected tree with a nonnegative edge capacity and a set of terminals. We show that a multiflow in an undirected tree can be characterized by a local structure, called "balancedness." Based on the characterization, we design a simple linear-time and linear-space algorithm for finding a maximum multiflow for our problem. Then we present a dynamic programming approach and give an linear-time and linear-space algorithm for finding a maximum integer multiflow when all edge capacities are integers. Our algorithms improve the best previous results by a factor of $\Omega(n)$. We also derive an upper and lower bounds on the number of pairs of terminals between which a positive amount of flow is sent in a given multiflow in a tree and present an linear-time and linear-space algorithm to represent the multiflow in a decomposition method (terminals pairs together with value of positive flow between them).

Key words. Multiterminal flow; Multiflow; Maximum flow; Minimum Cut; Trees; Linear-time algorithms

1 Introduction

The min-cut max-flow theorem by Ford and Fulkerson [8] is one of the most important theorems in graph theory. It catches a min-max relation between two fundamental graph problems. This theorem leads to many effective algorithms and much theory for flow problems as well as graph cut problems. Due to the great applications of it, researchers have interests to seek more similar min-max formulas in various kinds of flow and cut problems. In this paper, we consider the *maximum multiterminal flow problem*, a generalization of the basic maximum flow problem.

In the maximum flow problem, we are given two terminals (source and sink) and asked to find a maximum flow between the two terminals. A natural generalization of the maximum flow problem is the famous *maximum multicommidity flow problem*, in which, a list of pairs of source and sink for the commodities is given and the object is to maximize the sum of the simultaneous flows in all the source-sink pairs subject to the standard capacity and flow conservation requirements. The maximum multiterminal flow problem is one of the most important special cases of the maximum multicommidity flow problem. In it, a set T of more than one terminal is given and the list of source-sink pairs is given by all pairs of terminals in T. The extensions of the maximum flow problem have been extensively studied in the history. Readers are referred to a recent survey [3].

A dual problem of the maximum multiterminal flow problem is the *minimum multiterminal cut problem*, in which we are asked to find a minimum set of edges whose removal disconnects each pair of terminals in the graph. The minimum multiterminal cut problem is a generalization

[1]Technical report 2011-010, April 1, 2011.

of the minimum cut problem. When there are only two terminals, the min-cut max-flow theorem shows that the value of the maximum flow equals to the value of the minimum cut in the graph. However, when there are more than two terminals, the equivalence may not hold. Consider a star with three leaves. Each leaf is a terminal and each of the three edges has capacity 1. The flow value of a maximum multiterminal flow is 1.5 (a flow of size 1 routed between every pair of the three terminal pairs), whereas the size of a minimum multiterminal cut is 2. In fact, Cunningham [4] has proved a min-max theory for the pair of problems: The size of a minimum multiterminal cut is at most $(2 - \frac{2}{|T|})$ times of the flow value of a maximum multiterminal flow. A similar min-max theory for the maximum multicommidity flow problem and its dual problem is presented in [5].

In the maximum multiterminal flow problem, each edge is assigned a nonnegative capacity and a flow routed between a terminal pair is allowed to take any feasible fraction, whereas in the *integer multiterminal flow problem*, a flow is allowed to take a nonnegative integer and we are asked to find a maximum flow under this restriction. Clearly, we can simply assume that all edge capacities of the integer multiterminal flow problem are nonnegative integers. The integer multiterminal flow problem is different from the maximum multiterminal flow problem. We can see in the above example, the flow value of a maximum integer multiterminal flow is 1. The special case of the integer multiterminal flow problem where all edges have unit capacities is also known as the *T-path problem*, in which we are asked to find the maximum number of edge-disjointed paths between different terminal pairs.

In this paper, we study the maximum multiterminal flow problem in trees and give linear-time algorithms for both fractional and integer versions, which improve the best previous algorithms by a factor of $\Omega(n)$. Note that the maximum (integer) multicommidity flow problem in trees is NP-hard and there is a $\frac{1}{2}$-approximation algorithm for it [6].

The rest of the paper is organized as follows. Section 2 introduces basic notations on flows and reviews important min-max theorems for fractional and integer versions of maximum multiterminal flow problem. Section 3 discusses the decomposability of a "multiflow" into a set of flows for all individual pairs of terminals, and gives a characterization of multiflows by a local structure. Based on the characterization, we present a linear-time algorithm for computing a maximum multiflow in a tree. Section 4 takes a dynamic programming approach to obtain another linear-time algorithm for computing a maximum multiflow in a tree, and then extends these results to the problem of finding a maximum integer multiflow. Finally Section 5 makes some remarks.

2 Preliminaries

Let $\mathbb{R}^+$ denote the set of nonnegative reals, and $\mathbb{Z}^+$ denote the set of nonnegative integers. For two reals a, b with $a \leq b$, let $[a, b]$ denote the set of reals s with $a \leq s \leq b$. We may denote by $V(G)$ and $E(G)$ the sets of vertices and edges of an undirected graph G, respectively. An instance (G, T, c) of the multiterminal flow problem consists of a simple undirected graph $G = (V, E)$ with a vertex set V and an edge set E, a set $T \subseteq V$ of terminals, and a capacity function $c : E \to \mathbb{R}^+$. For two subsets $X, Y \subseteq V$, let $E(X, Y)$ denote the set of edges with one end-vertex in X and the other in Y, where $E(X, V - X)$ and $E(\{v\}, V - \{v\})$, $v \in V$ are denoted by $E(X)$ and $E(v)$, respectively. For a graph $G = (V, E)$ and a subset $X \subseteq V$, let $G[X]$ denote the subgraph of G induced by the vertices in X, For a function $h : E \to \mathbb{R}^+$, $h(e)$ for an edge $e = (u, v)$ is also denoted by $f(u, v)$, and $\sum_{e \in E(X,Y)} c(e)$ for subsets $X, Y \subseteq V$ is denoted by $h(X, Y)$. We denote $h(X, V - X)$ by $h(X)$. For path P and a real δ (possibly $\delta < 0$), the function $h' : E \to \mathbb{R}^+$ obtained from h by setting $h'(e) = h(e) + \delta$ $(e \in E(P))$ and $h'(e) = h(e)$

$(e \in E - E(P))$ is denoted by $h + (P, \delta)$.

We introduce "individual flows" and "multiflows" two representations of multiterminal flows. A path P in G is called a *T-path* if its end-vertices are two distinct terminals in T and none of its internal vertices belongs to T. Let $\mathcal{P}$ be the set of all T-paths in (G, T). An *individual flow* in (G, T) is defined to be a function $g : \mathcal{P} \to \mathbb{R}^+$, and its *value* $\alpha(g)$ is defined to be $\sum\{g(P) \mid P \in \mathcal{P}\}$. The *value* $\sigma_g(e)$ of g over an edge $e \in E$ is defined to be

$$\sigma_g(e) = \sum\{g(P) \mid e \in E(P),\ P \in \mathcal{P}\}.$$

An individual flow g is called *feasible* if

$$\sigma_g(e) \leq c(e) \text{ for all edges } e \in E.$$

A feasible individual flow g is called *maximum* if $\alpha(g)$ is maximized over all feasible individual flows in (G, T, c).

A function $f : E \to \mathbb{R}^+$ is called a *multiflow* if there is an individual flow g such that $f(e) = \sigma_g(e)$ for all edges $e \in E$, thus, f can be decomposed into an individual flow g. Such g is called a *decomposition* of a multiflow f. A multiflow f is called *feasible* if

$$f(e) \leq c(e) \text{ for all edges } e \in E.$$

Its *value* $\alpha(f)$ is defined to be $\frac{1}{2}\sum\{f(e) \mid e \in E(t),\ t \in T\}$, which is equal to $\alpha(g)$ of any decomposition g of f. A set $\{a_1, a_2, \ldots, a_k\}$ of nonnegative reals is called *balanced* if

$$\max\{a_i \mid i = 1, 2, \ldots, k\} \leq \frac{1}{2}\sum\{a_i \mid i = 1, 2, \ldots, k\}.$$

For a function $h : E \to \mathbb{R}^+$, an edge $e \in E(v)$ with a non-terminal vertex v is called a *heavy edge* of v if $h(e) > \frac{1}{2}h(\{v\})$, and the function h is called *balanced* in G if no non-terminal vertex has a heavy edge. It is not difficult to see that any multiflow is balanced. In a general graph G, there is a balanced function $h : E \to \mathbb{R}^+$ that admits no individual flow $g : \mathcal{P} \to \mathbb{R}^+$ such that $\alpha(g) = \alpha(h)$ and $\sigma_g(e) \leq h(e)$, $e \in E$. See Fig. 1(a) for such a balanced function h.

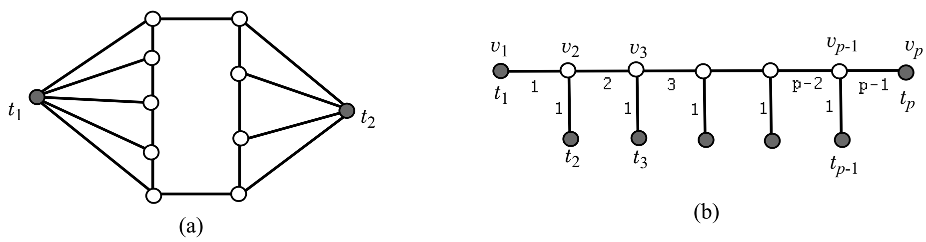

Figure 1: (a) A balanced function h in a graph $G = (V, E)$ with $T = \{t_1, t_2\}$ and $h(e) = 1$, $e \in E$. (b) A multiflow f in a tree G with $T = \{t_1, t_2, \ldots, t_p\}$, where $n = 2p - 2$, $f(v_i, v_{i+1}) = i$, $i = 1, 2, \ldots, p - 1$ and $f(v_i, t_i) = 1$, $i = 2, 3, \ldots, p - 1$.

A subset $X \subset V$ is called a *terminal-cut* if $|X \cap T| = 1$. A family $\mathcal{X} = \{X_t \mid t \in T\}$ of vertex-disjoint terminal-cuts X_t with $t \in X_t$, $t \in T$ is called a *cut-system* of T, and its cut-size $\gamma(\mathcal{X})$ is defined to be $\sum\{c(X_t) \mid t \in T\}$. Note that $\mathcal{X}$ is not required to be a partition of V. For any pair of a feasible multiflow f and a cut-system of T in (G, T, c), it holds

$$\alpha(f) \leq \frac{1}{2}\gamma(\mathcal{X}).$$

Cherkasskii [1] proved the next result.

Theorem 1 *A feasible multiflow f in (G, T, c) is maximum if and only if there is a cut-system $\mathcal{X}$ such that $\alpha(f) = \frac{1}{2}\gamma(\mathcal{X})$.*

Ibaraki *et al.* [7] proposed an $O(nm \log n)$-time algorithm for computing a maximum multiflow f in a graph G with n vertices and m edges. Hagerup *et al.* [9] proved a characterization of the maximum multiterminal flow problem and gave an $O(f(|T|)n)$-time algorithm for the maximum multiterminal flow problem in bounded treewidth graphs, where $f(|T|)$ is an exponential function of the number $|T|$ of terminals. This algorithm runs in linear time only when $|T|$ is restricted to a constant.

An integer version of multiterminal flow problem is defined as follows. Let $I = (G = (V, E), T, c)$ have integer capacities $c(e) \in \mathbb{Z}^+$, $e \in E$. An *integral multiflow* f is a multiflow which can be decomposed into *integer* individual flows g, i.e., $g : \mathcal{P} \to \mathbb{Z}^+$ (note that a multiflow f with $f(e) \in \mathbb{Z}^+$, $e \in E$ does not necessarily admit a decomposition into integer individual flows g). An instance (G, T, c) is called *inner-eulerian* if all edge capacities $c(e)$, $e \in E$ are integers and $c(v)$ is an even integer for each non-terminal vertex $v \in V - T$. It is known that any inner-eulerian instance admits a pair of a maximum integral multiflow f and a cut-system $\mathcal{X}$ with $\alpha(f) = \frac{1}{2}\gamma(\mathcal{X})$ [1]. In general, there is no pair of an integral multiflow f and a cut-system $\mathcal{X}$ with $\alpha(f) = \frac{1}{2}\gamma(\mathcal{X})$ even for trees. We review a min-max theorem on the integer version as follows.

For a family $\mathcal{X}$ of vertex subsets in an instance $I = (G = (V, E), T, c)$, let $V_{\mathcal{X}}$ denote $\cup_{X \in \mathcal{X}} X$. The vertex set W of a connected component in the graph $G[V - V_{\mathcal{X}}]$ is called an *odd set* of $\mathcal{X}$ in I if $c(W)$ is odd. Let $\kappa(\mathcal{X})$ denote the number of odd sets of $\mathcal{X}$ in I. For any pair of a feasible integral multiflow f and a cut-system $\mathcal{X}$ of T in $I = (G, T, c)$, it holds

$$\alpha(f) \leq \frac{1}{2}[\gamma(\mathcal{X}) - \kappa(\mathcal{X})].$$

Mader [10] proved the next result.

Theorem 2 *A feasible integral multiflow f in (G, T, c) is maximum if and only if there is a cut-system $\mathcal{X}$ such that $\alpha(f) = \frac{1}{2}[\gamma(\mathcal{X}) - \kappa(\mathcal{X})]$.*

For trees with n vertices, an $O(n^2)$-time algorithm for computing a maximum integral multiflow f is proposed [2], while no strongly-polynomial time algorithm is known to general graphs (e.g., see [3]).

In a rooted tree, the parent-children relationship among vertices/edges is defined. The parent of a non-root vertex v is denoted by p_v, and the edge (p_v, v) is called the *parent-edge* of v. For each edge $e = (u, v)$, where $u = p_v$, we call e the *parent-edge* of v and a *child-edge* of u, and call u and v the *parent* and the *child* of e, respectively. We treat the edge incident to the root as a non-leaf-edge.

3 Characterizing Multiflows in Trees

In this section, we prove that a given function $h : E \to \mathbb{R}^+$ in a tree is a multiflow if and only if h is balanced, and examine how small number of individual flows are necessary in a decomposition of a given multiflow f. Based on the characterization, we give a simple linear-time algorithm for computing a maximum multiflow in trees.

3.1 Decomposability of Multiflows

In a tree G, the unique path connecting two vertices u and v is denoted by $P_{u,v}$. For the two paths $P_{u,v}$ and $P_{u,v'}$ with three vertices $u, v, v' \in V$, the common vertex w in $P_{u,v}$ and $P_{u,v'}$ furthest from u is called the *branch vertex* of these paths (i.e., any two of $P_{u,w}$, $P_{w,v}$ and $P_{w,v'}$ are edge-disjoint). For a tree G with $n \geq 2$ vertices, a vertex is called a *leaf* if it has exactly one neighbour, and is called an *inner* vertex otherwise. An instance (G, T, c) of the max multiterminal cut problem in trees is called *standard* if G is a tree, $c(e) > 0$, $e \in E$, and T is given by the set of leaves in G. For each terminal $t \in T$, let e_t denote the edge incident to t.

Lemma 3 *For a standard instance (G, T, c), a function $h : E \to \mathbb{R}^+$ is a multiflow if and only if it is balanced.*

Proof. If h is a multiflow, then it has a decomposition of f into an individual flow g. The function g_P induced by a path P, i.e., $g_P(e) = g(P)$, $e \in E(P)$ and $g_P(e) = 0$, $e \in E - E(P)$ is balanced. Then we see that multiflow f, which is the sum of balanced functions g_P, is also balanced.

We prove the converse in the proof of Theorem 4. ∎

In an individual flow g, a T-path $P_{t,t'}$ with $g(P_{t,t'}) > 0$ is called a *positive path*, and a pair $\{t, t'\}$ of such terminals is called a *positive pair*. There is an example of a multiflow f such that, in any decomposition g of f, the sum of the number $|E(P_{t,t'})|$ of edges in positive T-paths $P_{t,t'}$ over all positive pairs $\{t, t'\}$ takes $\Omega(n^2)$ edges. See Fig. 1(b) for such a multiflow f in a tree instance, which has a unique decomposition g of f. As for the number of positive pairs, we shall show an upper bound $n - 1$, which is best possible, because any tree with n vertices has $n - 1$ freedom in setting capacities: the capacity of each of the $n - 1$ edges can be perturbed independently on those of other edges so that the resulting capacity remains balanced.

Theorem 4 *Any multiflow f in a standard instance (G, T, c) can be decomposed into an individual flow g such that the number of positive pairs is at most $n-1$, and a set of all positive pairs $\{t, t'\}$ together with their flow values $g(P_{t,t'}) > 0$ can be found in $O(n)$ time and space. Moreover, if the function f is inner-eulerian, then all flow values $g(P_{t,t'})$ can be chosen as integers.*

Note that once all the positive pairs are found we can construct the T-path $P_{t,t'}$ for a specified positive pair $\{t, t'\}$ in $O(|E(P_{t,t'})|)$ time. Regard the tree G as a rooted tree, and assume that the depth of each terminal in the rooted tree G has been stored. Suppose that the depth of t is larger than that of t'. Then we start from t and visit the ancestors until we reach a vertex v with the same depth of t'. After this, we visit the ancestors of v and t' one by one until we find the common ancestor of them. The edges traversed in this process give the desired path $P_{t,t'}$, and the process takes $O(|E(P_{t,t'})|)$ time.

Now we prove Theorem 4. For this, it suffices to show that any balanced capacity c in a standard instance can be decomposed into an individual flow g satisfying the conditions in the theorem, since we can apply the argument to each of the components induced by the edges e with positive $f(e)$ in a balanced function f. We show that there is a T-path P with a real $\delta > 0$ such that $c + (P, -\delta)$ remains balanced, and we set $g(P) := \delta$ and $c := c + (P, -\delta)$. By repeating this until $c \equiv 0$, we obtain a decomposition g of c.

To facilitate this, we convert a standard tree into a directed tree by assigning one or two directions to each edge. For a digraph, let $E^-(v)$ (resp., $E^+(v)$) denote the set of edges entering (resp., leaving) v. A digraph with an edge capacity c and a terminal set T is called *inner-di-eulerian* if $\sum_{e \in E^+(v)} c(e) = \sum_{e \in E^-(v)} c(e)$ for all non-terminal vertices $v \in V - T$. We treat

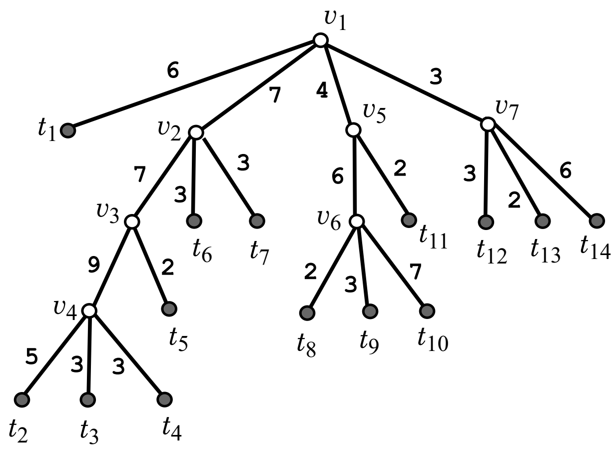

Figure 2: An example of standard tree instances (G, T, c).

a given tree instance (G, T, c) as an ordered tree rooted at an inner vertex r. Fig. 2 shows a standard instance which is rooted at $r = v_1$.

We convert $(G = (V, E), T, c)$ into an inner-di-eulerian directed tree by apply one of the following operations to each edge $e = (u, v) \in E$.

(i) give e the upward direction;

(ii) give e the downward direction; or

(iii) give e the two directions, i.e., replace $e = (u, v)$ with two copies, an upward edge $e' = (v, u)$ and a downward edge $e'' = (u, v)$ such that $c(e') + c(e'') = c(e)$.

More formally, we visit each vertex v in a breadth-first search from the root by applying (i)-(iii) to the child-edges of v so that (a) $\sum_{e \in E^+(v)} c(e) = \sum_{e \in E^-(v)} c(e)$ holds, and (b) left (resp., right) ones receive the upward (resp., downward) direction among the child-edges of v. Hence (iii) will be applied to at most one of the child-edges of v, which situates between those applied (i) and those applied (ii). Such an application of (i)-(iii) is uniquely determined as follows.

Let v be a vertex whose parent-edge e_v has been applied one of (i)-(iii), and let ε_v denote the capacity of the downward parent-edge e_v of a vertex v; i.e., $\varepsilon_v = c(e_v)$ if e_v is assigned the downward direction; $\varepsilon_v = 0$ if $v = r$ or e_v is assigned the upward direction; and $\varepsilon_v = c(e''_v)$ if e_v is replaced with a pair of an upward edge e'_v and a downward edge e''_v. By denoting the child-edges of v by $e_1, e_2, \ldots, e_p$, which appear in this order from left to right, we first find the index k such that

$$\varepsilon_v + \sum_{1 \le i < k} c(e_i) < \frac{1}{2} c(v) \le \varepsilon_v + \sum_{1 \le i \le k} c(e_i),$$

where such an index $k \le p$ exists since c is balanced and $\varepsilon_v \le 0.5c(v)$ (we interpret $\sum_{1 \le i < k} c(e_i) = 0$ when $k = 1$). For example, the index k of the root $r = v_1$ in Fig. 2 is 2, where $\varepsilon_r = 0$ and $c(r) = 20$. We next assign the upward direction to all child-edges e_i with $i < k$, and the downward direction to all child-edges e_i with $i > k$. If $\frac{1}{2}c(v) = \varepsilon_v + \sum_{1 \le i \le k} c(e_i)$, then we assign the downward direction to e_k. Otherwise, we replace e_k with a pair of an upward edge e'_k and a downward edge e''_k with $c(e'_k) = \frac{1}{2}c(v) - \varepsilon_v - \sum_{1 \le i < k} c(e_i)$ and $c(e''_k) = c(e_k) - c(e'_k)$. For example, the edge $e_k = (v_1, v_2)$ with $c(e_k) = 7$ of the root $r = v_1$ in Fig. 2 is replaced with upward edge

$e'_k = (v_2, v_1)$ with $c(e'_k) = 4$ and downward edge $e''_k = (v_1, v_2)$ with $c(e''_k) = 3$ (see Fig. 3 (a) and (b)).

After assigning directions to all edges, the resulting digraph $\tilde{G}$ is inner-di-eulerian. Fig. 3(a) and (b) shows such an inner-di-eulerian digraph $\tilde{G}$ obtained from the standard tree instance in Fig. 2. It is not difficult to observe that the given balanced capacity c can be decomposed into individual flows g by taking a positive path following the directions of edges from a terminal to other terminals. Recall that an explicit construction of all positive paths takes $O(n^2)$ time. In what follows, we show that all positive pairs can be computed in $O(n)$ time and the number of positive pairs can be bounded by $n - 1$.

A leaf-vertex in $\tilde{G}$ is called a *source* if an upward edge e is incident to it, and the *demand* of the source is defined to be $c(e)$. Analogously a leaf-vertex in $\tilde{G}$ is called a *destination* if a downward edge e is incident to it, and the *demand* of the destination is defined to be $c(e)$. The same leaf-vertex may play both roles of source and destination when (iii) is applied to its parent-edge. Let $S = \{s_1, s_2, \ldots, s_p\}$ and $D = \{d_1, d_2, \ldots, d_q\}$ denote the sets of sources and destinations in $\tilde{G}$, where they are indexed by the depth-first-search in the ordered tree G. In Fig. 3(a), the sets of terminals $\{t_1, t_2, t_3, t_4, t_8, t_9, t_{12}, t_{13}\}$ and $\{t_4, t_5, t_6, t_7, t_{10}, t_{11}, t_{13}, t_{14}\}$ give $S = \{s_1, s_2, \ldots, s_8\}$ and $D = \{d_1, d_2, \ldots, d_8\}$, respectively.

Without constructing paths concretely, we find such positive pairs of a source and a destination. For this, we first determine the amount of demands sent from sources for each upward edge by visiting upward edges in a bottom-up manner along $\tilde{G}$.

We represent the demands sent to an upward edge e by a list

$$L^+(e) = [s_h, a_h; s_{h+1}, a_{h+1}; \cdots; s_k, a_k]$$

for some integers $1 \leq h \leq k \leq p$, where a_i means that $\sum\{g(P_{s_i,d}) \mid e \in E(P_{s_i,d}), d \in D\} = a_i$. Hence the total value of individual flows passing through e is $\sum_{h \leq i \leq k} a_i$. In particular, for each upward leaf-edge $e = (s_i, v)$, we have $L^+(e) = [s_i, c(e)]$ (for example see Fig. 3(a)). Starting from $L^+(e)$ for all leaf-edges e, we compute $L^+(e)$ for other edges e recursively. For an inner vertex v such that its parent-edge e_v is upward and $L^+(e)$ for all its child-edges e have been determined, we decide $L^+(e_v)$ of the upward edge e_v. Let

$$x = \sum\{c(e') \mid \text{ downward child-edges } e' \text{ of } v\} \text{ and } y = c(e_v).$$

Then among the demands sent from sources through the upward child-edges of v, the amount y of them is sent to e_v and the rest of them (whose amount is x) is sent to the downward child-edges of v. We employ a strategy that the amount x of demands from leftmost upward child-edges is sent to the downward child-edges of v. As will be shown, this and the balancedness of c prevent from taking a positive path that visits a pair of upward and downward edges in (iii) (the two copies of the same child-edge of v). Then $L^+(e_v)$ for the amount x of demands is given by the remaining demands. For example, vertex v_4 in Fig. 3(a) has the upward parent-edge e_{v_4} with $c(e_{v_4}) = 7$, three upward child-edges (s_2, v_4), (s_3, v_4) and (s_4, v_4), and one downward child-edge (v_4, d_1), where $x = 1$ and $y = 7$ hold, and we sent the amount 1 of demand from s_2 to the downward edge (v_4, d_1), and the rest of demands from s_2, s_3 and s_4 are sent to the upward edge e_{v_4}.

More specifically, $L^+(e_v)$ is obtained as follows,

(1) For the upward child-edges $e_1, e_2, \ldots, e_p$ of vertex v which appear in this order from left to right, we let

$$L^+(v) = L^+(e_1) + L^+(e_2) + \cdots + L^+(e_p)$$

denote the list obtained by concatenating lists $L^+(e_1), L^+(e_2), \ldots, L^+(e_p)$ in this order.

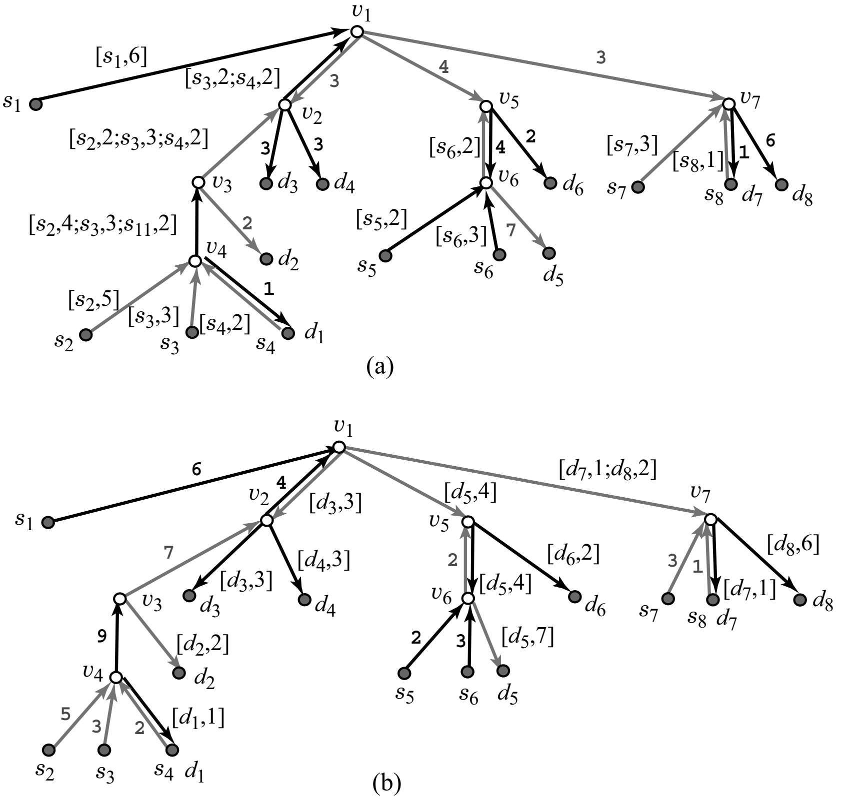

Figure 3: (a) Demands sent from sources in upward edges; (b) demands sent to sources in downward edges.

(2) For a list $L = [s_h, a_h; s_{h+1}, a_{h+1}; \cdots; s_k, a_k]$ and two reals $x, y \in \mathbb{R}^+$ with $x + y = \sum_{h \le i \le k} a_i$, we split L into two lists $L|_x$ and $L|^y$ whose total demands are x and y, respectively. For the largest index j such that $x \le \sum_{h \le i \le j} a_i$, let

$$L|_x = [s_h, a_h; \cdots; s_{j-1}, a_j; s_j, a'_j], \quad L|^y = [s_j, a_j - a'_j; s_{j+1}, a_{j+1}; \cdots; s_k, a_k],$$

where $a'_j = x - \sum_{h \le i \le j} a_i$ (we exclude the term (s_j, a'_j) with $a'_j = 0$ from $L|_x$ and the term $(s_j, a_j - a'_j)$ with $a_j - a'_j = 0$ from $L|^y$). By our strategy, we set $L^+(e_v) := L^+(v)|^y$ for $y = c(e_v)$. See Fig. 3(a) for an example of the demands from sources for all upward edges in $\tilde{G}$.

Analogously we represent the demands sent to a downward edge e by a list

$$L^-(e) = [d_h, a_h; d_{h+1}, a_{h+1}; \cdots; d_k, a_k],$$

where a_i means that $\sum\{g(P_{s,d_i}) \mid e \in E(P_{s,d_i}), s \in S\} = a_i$. Hence the total value of individual flows passing through e is $\sum_{h \le i \le k} a_i$. For the downward child-edges $e_1, e_2, \ldots, e_p$ of a vertex v which appear in this order from left to right, we let

$$L^-(v) = L^-(e_1) + L^-(e_2) + \cdots + L^-(e_p)$$

denote the list obtained by concatenating lists $L^-(e_1), L^-(e_2), \ldots, L^-(e_p)$ in this order. Then we set $L^-(e_v) := L^-(v)|_x$ for $x = c(e_v)$. Thus we let the demands from rightmost downward child-edges be sent from the upward child-edges of v. See Fig. 3(b) for an example of the demands sent to destinations for all downward edges in $\tilde{G}$.

We have determined the demands sent from sources for all upward edges and the demands sent to destinations for all downward edges. The remaining task is to find positive pairs between sources and destinations. Let $c^+(v)$ (resp., $c^-(v)$) denote the sum of capacities of leaving (resp., entering) a vertex v in the resulting digraph.

Consider the case where (iii) is applied to the parent-edge e_v of a vertex v (the other cases can be treated analogously). For the upward parent-edge e'_v and the downward parent-edge e''_v of v, let $y = c(e'_v)$, $x = c^+(v)/2 - 2y$, $x' = c(e''_v)$, and $y' = c^-(v)/2 - x'$. Then $L^+(v)$ is split into $L^+(v)|_x$ and $L^+(v)|^y = L^+(e'_v)$ whereas $L^-(v)$ is split into $L^-(v)|_{x'} = L^-(e''_v)$ and $L^-(v)|^{y'}$. Note that $L^+(v)|_x$ and $L^-(v)|^{y'}$ have the same total demands since the capacity c in $\tilde{G}$ is inner-di-eulerian. The demands from the sources in $L^+(v)|_x$ are all sent to the destinations in $L^-(v)|^{y'}$. We make positive pairs between the sources in $L^+(v)|_x$ and the destinations in $L^-(v)|^{y'}$ by choosing terms in the lists from left to right. Thus, for

$$L^+(v)|_x = [s_h, a_h; s_{h+1}, a_{h+1}; \cdots; s_{h+p}, a_{h+p}] \text{ and}$$

$$L^-(v)|^{y'} = [d_k, b_k; d_{k+1}, b_{k+1}; \cdots; d_{k+q}, b_{k+q}],$$

we first choose (s_h, a_h) and (d_k, b_k) and make a positive pair (s_h, d_k) with $g(P_{s_h,d_k}) = \delta$ for $\delta = \min\{a_h, b_k\}$. We update the lists by $a_h := a_h - \delta$ and $b_k := b_k - \delta$, eliminating term $(s_h, a_h = 0)$ or $(d_k, b_k = 0)$ if any. By repeating this, we obtain at most $p + q + 1$ positive pairs with their flow values. For any resulting positive pair (s_i, d_j), the upward child-edge $e_{i'}$ sending the demand from s_i and the downward child-edge $e_{j'}$ sending the demand into d_j satisfy $i' < j'$ since the original capacity c of G is balanced. It is easy to see that the above procedure for a vertex v can be implemented to run in $O(|E(v)|)$ time and that the positive pairs at all vertices can be determined in $O(n)$ time.

Finally we derive a tight upper bound on the maximum number of positive pairs generated by the above procedure for all vertices v. Let (t_i, t'_i) and δ_i, $i = 1, 2, \ldots, \ell$ ($\ell \leq p + q - 1$) be the resulting positive pairs and their values at vertex v. We consider the effect of removing the flow $g(P_{t_i,t'_i})$ from (G, T, c), i.e., the modification of c by $c + (P_i, -\delta_i)$, where we remove any edge e from $\tilde{G}$ whenever $c(e)$ newly becomes zero. In the process of removing positive paths $P_1, P_2, \ldots, P_\ell$ in this order, at least one of the upward and downward child-edges of v will be removed after each operation $c := c + (P_i, -\delta_i)$. Note that a newly removed edge is one of the two copies of an edge or disconnects two terminals t_i and t'_i. In the latter case, the number of connected components containing a terminal increases at least by 1. The number of times that two terminals in T are disconnects is at most $|T| - 1$. Recall that the number of edges $e \in E$ that can be applied (iii) is bounded by the number of inner vertices, i.e., $n - |T|$. Therefore, the number of positive paths is at most $n - |T| + |T| - 1 = n - 1$. As already observed, the bound $n - 1$ cannot be improved for standard instances. A star with n vertices and $n - 1$ edges, where $n - 4$ edges is of capacity 10, two edges is of capacity 11, and one edge is of capacity $10(n - 2)$, has exactly $n - 1$ positive pairs.

This finishes the proof for the property that there is a decomposition g of any multiflow f such that the number of positive pairs is at most $n - 1$, and the set of such positive pairs $\{t, t'\}$ together with their flow values $g(P_{t,t'}) > 0$ can be found in $O(n)$ time. We easily see that if a given function c in G is inner-eulerian, then all the values to determine new capacities in the above computation can be chosen as integers. Hence an inner-eulerian multiflow f is integral and can be decomposed into $n - 1$ positive T-paths with integer flow values. This completes the proof of Theorem 4.

3.2 Computing Balanced Functions in Trees

In this section, we present a simple linear-time algorithm for computing a maximum multiflow f in a tree together with a cut-system $\mathcal{X}$ with $\alpha(f) = \frac{1}{2}\gamma(\mathcal{X})$, which is a certificate of the maximality of f.

Let f be a feasible multiflow in (G, T, c). An edge e is called *saturated* if $f(e) = c(e)$. For a terminal-cut X_t with $t \in X_t$, it holds $f(e_t) \leq f(X_t) \leq c(X_t)$. We say that a terminal-cut X_t with $t \in X_t$ is *tight* in f if $f(e_t) = c(X_t)$ holds. Hence a cut-system $\mathcal{X}$ satisfies $\alpha(f) = \frac{1}{2}\gamma(\mathcal{X})$ if and only if each terminal-cut in $\mathcal{X}$ is tight. Note that $f(X_t) = c(X_t)$ holds for a tight terminal-cut X_t, but $f(X_t) = c(X_t)$ does not necessarily mean that X_t is a tight, where possibly $f(e_t) < f(X_t) = c(X_t)$. Moreover any tight terminal-cut X induces a connected graph $G[X]$ in a standard instance since $c(e) > 0$ for all edges e. If an inner vertex v has a heavy edge $e \in E(v)$, then we see that the amount $c(e) - (c(v) - c(e))$ of the capacity of e will never be used by any multiflow. Hence the set of feasible multiflows remains unchanged even if we reduce $c(e)$ to $c(v) - c(e)$. Suppose that we obtained a balanced capacity function c' by applying this procedure. Then the resulting capacity function c' is a multiflow $f = c'$ such that $2\alpha(f) = \sum_{t \in T} f(e_t) = \sum_{X_t \in \mathcal{X}} c'(t) = \gamma(\mathcal{X})$ holds for the cut-system $\mathcal{X} = \{\{t\} \mid t \in T\}$ of singletons, indicating that the multiflow f is maximum. In the following, we show not only how to reduce a given capacity function c to a balanced function f but also how to construct a cut-system $\mathcal{X}$ such that $2\alpha(f) = \gamma(\mathcal{X})$ holds for the original capacity function c.

Choose a terminal $r \in T$, and regard G as a tree rooted at r. In a rooted tree instance (G, T, c), the function f is called *upper-light* if the parent-edge of each inner vertex v is not a heavy edge in f (i.e., $f(e) \leq f(v)/2$ for the parent-edge e of v). The next algorithm first reduces the capacity of heavy parent-edges traversing the tree in a bottom-up manner, and then reduces the capacity of heavy child-edges traversing the tree in a top-down manner.

See Fig. 4(a)-(c) for an example of execution of algorithm TreeFlow.

Algorithm TreeFlow
Input: A standard tree instance $(G = (V, E), T, c)$.
Output: A multiflow f and a cut-system $\mathcal{X}$ such that $2\alpha(f) = \gamma(\mathcal{X})$.

Initialize by $f(e) := c(e)$ for all edges $e \in E$;
for each inner vertex $v \in V - T$ to be scanned in a nonincreasing order of their depths **do**
  if the parent-edge e of v is a heavy edge of v in f (i.e., $f(e) > f(v)/2$) **then**
    $f(e) := f(v) - f(e)$
  end /* if */
end; /* for */
/* the function f is upper-light */
for each inner vertex $v \in V - T$ to be scanned in a nondecreasing order of their depths **do**
  if a child-edge e' of v is a heavy edge of v in f (i.e., $f(e') > f(v)/2$) **then**
    $f(e') := f(v) - f(e')$
  end /* if */
end; /* for */
/* the function f is balanced */
For each terminal $t \in T$, let X_t be the set of vertices reachable from t via unsaturated edges (i.e., edges e with $f(e) < c(e)$) in G;
Let $\mathcal{X} := \{X_t \mid t \in T\}$.

Theorem 5 *A feasible multiflow f and a cut-system $\mathcal{X}$ with $2\alpha(f) = \gamma(\mathcal{X})$ in a standard instance (G, T, c) can be found in $O(n)$ time and space, where f is a maximum multiflow.*

Proof. We prove that algorithm TreeFlow delivers a desired pair of f and $\mathcal{X}$.

After initialization of $f := c$ in TreeFlow, the variable f is reduced to a non-negative upper-light function by the first for-loop. Then the second for-loop further reduces it to a balanced function. Note that f remains upper-light after reducing $f(e')$ of the heavy child-edge e' of each inner vertex v. Hence the resulting function is a feasible multiflow to the given instance by Lemma 3. It is easy to see that the final multiflow f can be constructed in $O(n)$ time and space.

We show that a family $\mathcal{X} = \{X_t \mid t \in T\}$ constructed by TreeFlow is a cut-system satisfying $\gamma(\mathcal{X}) = 2\alpha(f)$.

When $f(e)$ is reduced during an execution of TreeFlow, we assign an orientation to the edge e as follows.

Let E_2 be the set of edges e' for which $f(e') := f(v) - f(e')$ is executed at an inner vertex u in the second for-loop, where we treat e' as a directed edge with upward orientation from u' to u, where u' is the child of e'. (Note that an edge $e = (p_v, v)$ for which $f(e) := f(v) - f(e)$ is executed in the first for-loop may receive the upward orientation from v to p_v when p_v is scanned as an inner vertex u in the second for-loop.) Clearly the indegree of any vertex is at most one in the digraph (V, E_2).

Let E_1 be the set of edges $e \in E - E_2$ for which $f(e) := f(v) - f(e)$ is executed at an inner vertex v in the first for-loop, where we treat e as a directed edge with downward orientation from p_v to v. No edge in E_1 is incident to any terminal in T, and the indegree of any vertex is at most one in the digraph (V, E_1).

We show that the indegree of any vertex is at most one in the digraph $(V, E_1 \cup E_2)$. Assume indirectly that the indegree of some vertex is at least 2 in $(V, E_1 \cup E_2)$, i.e., there is an inner vertex v such that $e = (p_v, v) \in E_1$ and $e_1 = (v', v) \in E_2$. Since $e = (p_v, v) \in E_1$ implies that $f(e) := f(v) - f(e)$ has been executed, it must hold $f(e) = f(v)/2$ when v is scanned in the second for-loop, where no child-edge e' can be a heavy edge of v, a contradiction. Also for each edge $e = (p_v, v) \in E_1$, $f(e) = f(v)/2$ still holds after the second for-loop, since no child-edge of v can get the upward orientation.

Hence $(V, E_1 \cup E_2)$ is a disjoint set of out-branchings, i.e., directed trees with indegree at most 1, and each directed edge $e = (u, v)$ in $(V, E_1 \cup E_2)$ satisfies $f(e) = f(v)/2$. Therefore, by the construction of $\mathcal{X}$, we see that X_t is the set of vertices in the out-branching T_t starting from t. Hence two terminal-cuts X_t and $X_{t'}$ are vertex-disjoint, and $\mathcal{X}$ is a cut-system. Since T_t is an out-branching rooted at t and each directed $e = (u, v)$ satisfies $f(e) = f(v)/2$, we see that $f(X_t) = \sum_{e \in E(X_t)} f(e) = f(e_t)$ holds. Since the edges in $E(X_t)$ are saturated by f, we have $c(X_t) = f(X_t)$. Therefore, each terminal-cut $X_t \in \mathcal{X}$ satisfies $c(X_t) = f(X_t) = f(e_t)$, and is tight in f. This proves $2\alpha(f) = \sum_{t \in T} f(e_t) = \sum_{X_t \in \mathcal{X}} c(X_t) = \gamma(\mathcal{X})$.

Since two terminal-cuts X_t and $X_{t'}$ are vertex-disjoint, TreeFlow can construct it in $O(n)$ time and space. ∎

4 Dynamic Programming Approach

In this section, we take a dynamic programming approach to tree instances to obtain another linear-time algorithm for computing a maximum multiflow and a linear-time algorithm for finding a maximum integral multiflow. For a notational convenience, we assume that in a given standard instance (G, T, c), each inner vertex has exactly three neighbours. For this, we replace the two

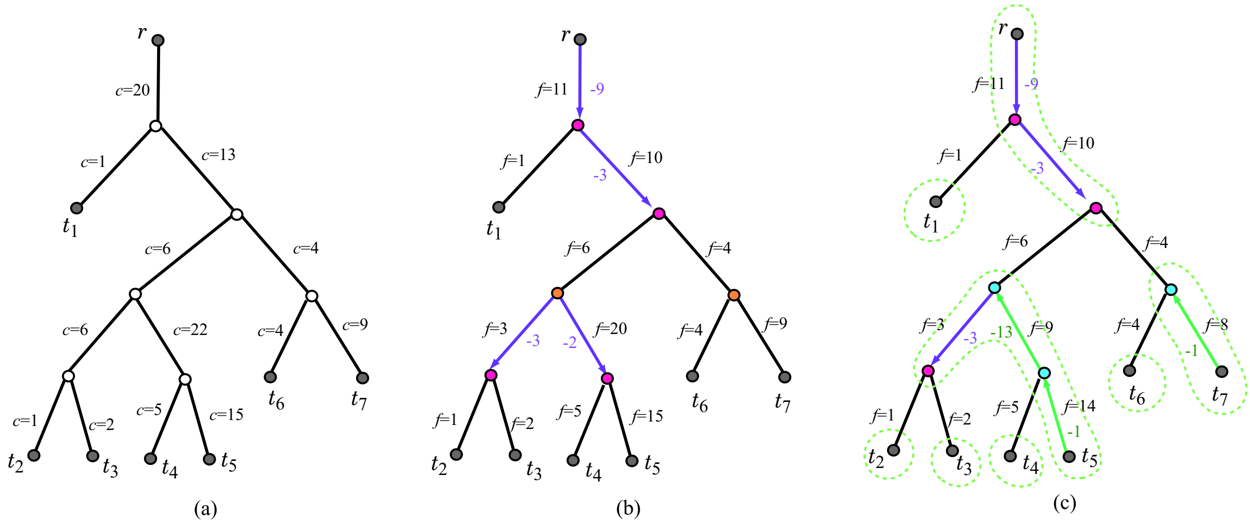

Figure 4: (a) An example of a tree rooted at a terminal $r \in T$, where the number c beside each edge e denotes its capacity $c(e)$: (b) the upper-light function f after the first for-loop in TreeFlow, where the number f beside each edge e denotes $f(e)$ and the edges e whose $f(e)$ is reduced are depicted by downward arrows: (c) the balanced function f after the second for-loop in TreeFlow, where the number f beside each edge e denotes $f(e)$, the edges e whose $f(e)$ is reduced are depicted by upward arrows, and the terminal-cut $X_t \in \mathcal{X}$ for each terminal $t \in T$ is depicted by a dashed circle.

edges (u, v) and (v, w) incident to any inner vertex of degree 2 with a single edge (u, w) with $c(u, w) = \min\{c(u, v), c(v, w)\}$, and replace any inner vertex v of degree $d \geq 4$ with new $d - 2$ vertices and $d - 3$ edges of capacity ∞ which form a subtree in the enlarged tree.

4.1 Blocking Flows in Trees

To find a maximum multiflow in a standard instance $I = (G, T, c)$, we construct a feasible multiflow f together with a cut-system such that $\alpha(f) = \frac{1}{2}\gamma(\mathcal{X})$, which implies the maximality of f. Our construction is based on a dynamic programming over subinstances I_e, $e \in E$ defined as follows.

Choose a terminal $r \in T$, and regard G as a tree rooted at r. For each edge $e = (u, v)$, where u is the parent of v, let V_e denote the set of u and the descendants of v including v itself, $G_e = (V_e, E_e)$ the graph induced from G by V_e, and $T_e = T \cap V_e$. Let $I_e = (G_e, T_e \cup \{u\}, c)$ be a standard instance obtained by adding u to T_e as a new terminal in G_e, where we call u the *root-terminal* of I_e. Let $\widetilde{I}_e$ be the instance obtained from I_e by regarding $c(e)$ as ∞.

For an instance $I_e = (G_e, T_e \cup \{u\}, c)$, we say that a feasible multiflow f in I_e *blocks* a cut-system $\mathcal{X}$ of T_e if each terminal-cut $X_t \in \mathcal{X}$ is tight in f (since u is a terminal in I_e, no terminal cut $X \in \mathcal{X}$ contains u). Such f and $\mathcal{X}$ are called *blocking* and *f-blocking* in I_e, respectively (note that $\mathcal{X}$ contains no terminal-cut for the root-terminal u). For each edge $e \in E$, we define $\Phi(e)$ to be the set of all reals $x \in \mathbb{R}^+$ such that I_e has a blocking multiflow f with $f(e) = x$. For each leaf-edge e_t with $t \in T - \{r\}$, we easily see that $\Phi(e_t) = [c(e_t), c(e_t)]$ since multiflow $f(e_t) = c(e_t)$ blocks cut-system $\{X_t = \{t\}\}$ of $\{t\}$. In what follows, we give a characterization of $\Phi(e)$ for inner edges e, based on which we design a dynamic programming algorithm.

Let us first observe that the set $\Phi(e_r)$ for the edge $e_r = (r, v) \in E$ incident to the root $r \in T$

tells how to get a maximum multiflow f in $I = I_{e_r}$.

Lemma 6 *For an instance $I = (G, T, c)$ rooted at a terminal $r \in T$, let f be a blocking multiflow with $f(e_r) = \max\{x \in \Phi(e_r)\}$. Then a cut-system $\mathcal{X}$ of T such that $\alpha(f) = \frac{1}{2}\gamma(\mathcal{X})$ can be constructed from f in linear time (hence f is a maximum multiflow in I).*

Proof. For each terminal $t \in T$, let X_t be the set of vertices reachable from t via unsaturated edges in G with f. We show that $\mathcal{X} = \{X_t \mid t \in T\}$ is a cut-system satisfying $\sum_{t \in T} f(e_t) = \gamma(\mathcal{X})$.

We first show that every two sets in $\mathcal{X}$ are disjoint, from which each set $X_t \in \mathcal{X}$ is a terminal-cut and satisfies $f(X_t) = c(X_t)$ by construction. Since f is a blocking multiflow in $I_{e_r} = I$, there is an f-blocking cut-system $\mathcal{X}' = \{X'_t \mid t \in X'_t,\ t \in T - \{r\}\}$ in I. Note that $X_t \subseteq X'_t$ holds for each $t \in T - \{r\}$ since all edges in $E(X'_t)$ are saturated. Since X_r is spanned by unsaturated edges, it is disjoint with any set $X'_t \subseteq V - \{r\}$ with $t \in T - \{r\}$. Therefore every two sets in $\mathcal{X}$ are disjoint, and $\mathcal{X}$ is a cut-system.

We next claim that each $X_t \in \mathcal{X}$ is tight, i.e., $f(e_t) = c(X_t)$. By definition of $\mathcal{X}'$, for each $t \in T - \{r\}$, it holds $f(e_t) = f(X'_t)$ and any positive path $P_{t,t'}$, $t' \in T - \{t\}$ in a decomposition g of f also passes through $E(X_t)$ since $X_t \subseteq X'_t$. Hence $f(X'_t) = f(X_t)$. Therefore, $f(e_t) = f(X'_t) = f(X_t) = c(X_t)$ for all $t \in T - \{r\}$.

We now show that $f(e_r) = c(X_r)$. If $f(e_r) = c(e_r)$, then we have $X_r = \{r\}$ and $f(e_r) = c(e_r) = c(X_r)$. Consider the case of $f(e_r) = \max\{x \in \Phi(e_r)\} < c(e_r)$. Assume indirectly that $f(e_r) < c(X_r)$. Since $f(X_r) = c(X_r)$, there must be two terminals $t, t' \in T - \{r\}$ such that $g(P_{t,t'}) > 0$ and $V(P_{t,t'}) \cap X_r \neq \emptyset$ in any decomposition g of f. Let w be the branch vertex of $P_{t,r}$ and $P_{t',r}$, and let δ be the minimum of $g(P_{t,t'})$ and $\min\{(c(e) - f(e))/2 \mid e \in E(P_{w,r})\} > 0$ (see Fig. 5(a)). The function $f' := f + (P_{t,t'}, -\delta) + (P_{t,r}, \delta) + (P_{t',r}, \delta)$ is a feasible multiflow in I. Since $f'(e) = f(e)$ for any other edges $e \in E - E(P_{r,w})$, the multiflow f' blocks $\mathcal{X}'$, and thereby f' is a blocking flow in I_{e_r}. However, $f'(e_r) > f(e_r) = \max\{x \in \Phi(e_r)\}$ contradicts the definition of $\Phi(e_r)$. Therefore, all terminal-cuts in $\mathcal{X}$ are tight in f, and it holds $\sum_{t \in T} f(e_t) = \gamma(\mathcal{X})$, as required.

Since every two sets in $\mathcal{X}$ are disjoint, $\mathcal{X}$ can be obtained in linear time. ∎

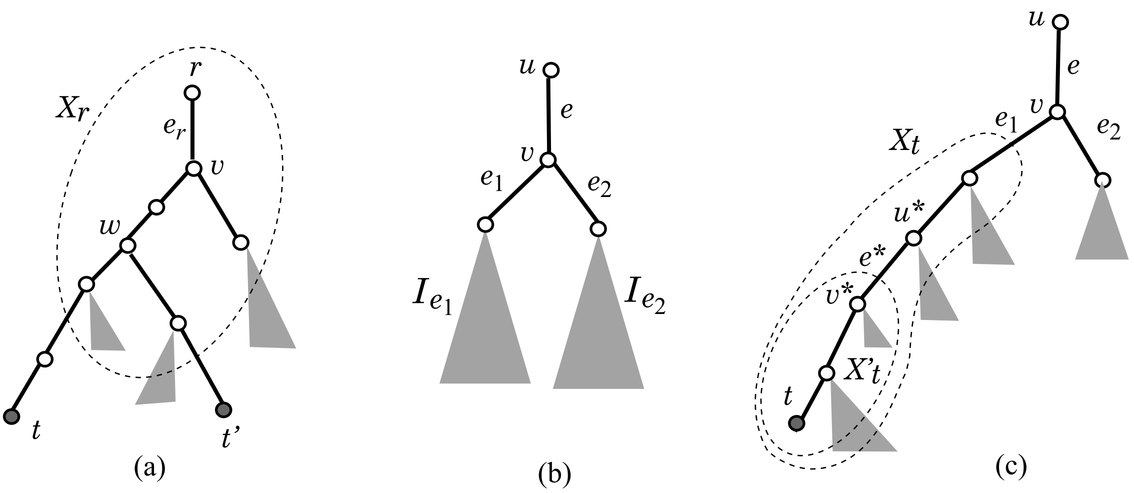

Figure 5: (a) A terminal-cut X_r for the root r: (b) two subinstances I_{e_1} and I_{e_2} in I_e: (c) A terminal-cut X_{t^*} in $\tilde{f}$.

Lemma 7 *For a blocking multiflow f in $I_e = (G_e, T_e \cup \{u\}, c)$, let $\mathcal{X}$ be an f-blocking cut-system of T_e in I_e and g be a decomposition of f. Then each positive $T_e \cup \{u\}$-path P contains at most one edge from $E(X)$ for any terminal-cut $X \in \mathcal{X}$. In particular, the terminal-cut $X_t \in \mathcal{X}$ of each terminal t is vertex-disjoint with any positive path $P_{t',t''}$ between terminals $t', t'' \in T_e \cup \{u\} - \{t\}$.*

Proof. For each terminal-cut $X_t \in \mathcal{X}$, every $T_e \cup \{u\}$-path $P_{t,t'}$ between t and other terminal $t' \in T_e \cup \{u\}$ must pass through an edge in $E(X_t)$ exactly once. Since the tightness of X_t implies $f(X_t) = f(e_t) = \sum\{g(P_{t,t'}) \mid t' \in T_e \cup \{u\} - \{t\}\}$, any path $P_{t',t''}$ between two terminals $t', t'' \in T_e \cup \{u\} - \{t\}$ contains more than one edge from $E(X_t)$ and thereby must satisfy $g(P_{t',t''}) = 0$. From this, the terminal-cut $X_t \in \mathcal{X}$ of each terminal t is vertex-disjoint with any positive path $P_{t',t''}$ between terminals $t', t'' \in T_e \cup \{u\} - \{t\}$, since otherwise $P_{t',t''}$ would contain more than one edge from $E(X_t)$. ∎

Now we show by an induction on the depth of e for $\{I_e \mid e \in E\}$ that $\Phi(e)$ is given by $[a, b]$, i.e., it is characterized by its minimum value $a = \min\{x \in \Phi(e)\}$ and maximum value $b = \max\{x \in \Phi(e)\}$. Recall that $\Phi(e) = [c(e), c(e)]$ for all leaf-edges e.

Lemma 8 *Let $e \in E$ be a non-leaf edge with its child-edges e_1 and e_2. For any reals $y \in \Phi(e_1)$, $z \in \Phi(e_2)$ and $s \in [0, \min\{y, z\}]$, it holds $x = y + z - 2s \in \Phi(e)$ when $x \le c(e)$.*

Proof. It suffices to show that I_e admits a blocking multiflow f with $f(e) = y + z - 2s$ when $y + z - 2s \le c(e)$. By definition, the instance I_{e_i} admits a blocking multiflow f_i and an f_i-blocking cut-system $\mathcal{X}_i$ of T_{e_i} for each $i = 1, 2$ such that $f_1(e_1) = y$ and $f_2(e_2) = z$ (see Fig. 5(b)). Let $f : E_e \to\to \mathbb{R}^+$ be the function defined by $f(e) = y + z - 2s$ and $f(e') = f_i(e')$ if $e' \in E_{e_i}$, $i = 1, 2$. Clearly none of edges e, e_1 and e_2 is heavy. Hence f is a balanced function, and is a feasible multiflow by Lemma 3 if $y + z - 2s \le c(e)$. Obviously $\mathcal{X} = \mathcal{X}_1 \cup \mathcal{X}_2$ is an f-blocking cut-system in I_e. This proves $y + z - 2s \in \Phi(e)$. ∎

To facilitate computing flow values of $f(e)$ from those of $f(e_1)$ and $f(e_2)$, we define an operation on two sets $A, B \subseteq \mathbb{R}^+$ of nonnegative reals. Let $A \oplus B$ denote the set of nonnegative reals $\{y + z - 2s \mid s \in [0, \min\{y, z\}]\}$ over all $y \in A$ and $z \in B$. Hence Lemma 8 tells that

$$(\Phi(e_1) \oplus \Phi(e_2)) \cap [0, c(e)] \subseteq \Phi(e). \tag{1}$$

In particular, for sets $A_1 = [a_1, b_1]$ and $A_2 = [a_2, b_2]$, we observe that

$$A_1 \oplus A_2 = \begin{cases} [a_1 - b_2, b_1 + b_2] & \text{if } b_2 < a_1 \\ [a_2 - b_1, b_1 + b_2] & \text{if } b_1 < a_2 \\ [0, b_1 + b_2] & \text{otherwise.} \end{cases}$$

We remark that, given a real $x \in [a, b] = A \oplus B$, $A = [a_1, b_1]$ and $B = [a_2, b_2]$, three reals $y \in A$, $z \in B$ and $s \in [0, \min\{y, z\}]$ such that $x = y + z - 2s$ can be found in constant time. To see this, assume $a_2 > b_1$ (other cases can be treated analogously). Then $A \oplus B$ is the union of $[b_1, b_1] \oplus [a_2 + ib_1] = [a_2 + (i-1)b_1, a_2 + (i+1)b_1]$ ($i = 0, 1, \ldots, \lfloor (b_2 - a_2)/b_1 \rfloor$) and $[b_1, b_1] \oplus [b_2, b_2] = [b_2 - b_1, b_1 + b_2]$. Hence if $x \ge b_2 - b_1$, then $x \in [b_1, b_1] \oplus [b_2, b_2]$ and we have $x = b_1 + b_2 - 2s$ (where $s = (x - b_1 - b_2)/2 \in [0, b_1]$). Otherwise $x < b_2 - b_1$ belongs to $[b_1, b_1] \oplus [a_2 + (k+1)b_1] = [a_2 + kb_1, a_2 + (k+2)b_1]$ for the integer $k = \lfloor (x - a_2)/b_1 \rfloor$, and we have $x = b_1 + (a_2 + (k+1)b_1) - 2s$ (where $s = (x - a_2 - (k+2)b_1)/2 \in [0, a_2 + (k+1)b_1]$).

Conversely we show $\Phi(e) \subseteq (\Phi(e_1) \oplus \Phi(e_2)) \cap [0, c(e)]$ or $\Phi(e) \subseteq [c(e), c(e)]$. In other words, we shall prove the next.

Lemma 9 *For each leaf edge $e \in E$, it holds $\Phi(e) = [c(e), c(e)]$. For each non-leaf edge $e \in E$ with its child-edges e_1 and e_2, let $\Phi(e_1) = [a_1, b_1]$ and $\Phi(e_2) = [a_2, b_2]$. Then for $[a, b] = [a_1, b_1] \oplus [a_2, b_2]$, it holds*

$$\Phi(e) = \begin{cases} [a,b] \cap [0, c(e)] & \text{if } c(e) \geq a \\ [c(e), c(e)] & \text{otherwise.} \end{cases}$$

We prove Lemma 9 by induction on height of edges e. Assuming that $\Phi(e')$ for each edge e' in G_e is given as an interval $[a, b]$ determined by Lemma 9, we prove the following two lemmas, from which Lemma 9 follows.

Lemma 10 *For each non-leaf edge $e \in E$ with its child-edges e_1 and e_2, let $\Phi(e_1) = [a_1, b_1]$ and $\Phi(e_2) = [a_2, b_2]$. Then for $[a, b] = [a_1, b_1] \oplus [a_2, b_2]$, it holds*

$$\Phi(e) \subseteq [\min\{c(e), a\}, \min\{c(e), b\}] = ([a, b] \cap [0, c(e)]) \cup \{c(e)\}.$$

Proof. Let u and v be the parent and child of $e = (u, v)$. We first claim that $f(e) \in \Phi(e_1) \oplus \Phi(e_2)$ holds for a blocking multiflow f in I_e if I_e has an f-blocking cut-system $\mathcal{X}$ such that there is no terminal-cut $X_t \in \mathcal{X}$ with $e \in E(X_t)$. Let $f(e_1) = y$ and $f(e_2) = z$. For each $i = 1, 2$, consider the multiflow f_i induced from f by G_{e_i}. Since e is not contained in $E(X_t)$ for any terminal-cut $X_t \in \mathcal{X}$, we see that cut-system $\mathcal{X}_i = \{X \in \mathcal{X} \mid X \subseteq V_{e_i}\}$ is f_i-blocking in I_{e_i}, and f_i is blocking in I_{e_i}. This implies $y \in \Phi(e_1)$ and $z \in \Phi(e_2)$, and $x \in \Phi(e_1) \oplus \Phi(e_2)$, as claimed. In particular, for any blocking multiflow f in I_e with $f(e) < c(e)$, its f-blocking cut-system $\mathcal{X}$ cannot contain a terminal-cut $X_t \in \mathcal{X}$ with $e \in E(X_t)$ (since e is not saturated by f), and hence $f(e) \in \Phi(e_1) \oplus \Phi(e_2)$ must hold. Therefore, $x \notin \Phi(e)$ if $x \notin \Phi(e_1) \oplus \Phi(e_2)$ and $x < c(e)$.

To prove $\Phi(e) \subseteq [\min\{c(e), a\}, \min\{c(e), b\}]$, it suffices to show that $x \notin \Phi(e)$ if $b < x = c(e)$. Let f be a blocking multiflow with $f(e) = c(e) > b$ in I_e, and assume that an f-blocking cut-system $\mathcal{X}$ in I_e contains a terminal-cut X_t with $e \in E(X_t)$ (since otherwise $f(e) \in \Phi(e_1) \oplus \Phi(e_2)$ must hold by the above claim). Without loss of generality assume that this terminal t belongs to T_{e_1}. See Fig. 5(c). By Lemma 7 and tightness of X_t, any positive T_e-path $P_{t',t''}$ with $\{t', t''\} \neq \{u, t\}$ in a decomposition g of f is vertex-disjoint with X_t. In particular, $g(P_{u,t'}) = 0$ holds for any terminal $t' \in T_e - \{t\}$, and hence $f(e') \geq f(e) > b$ for all edges e' in $P_{u,t}$. Let $\delta = \min\{c(e') - f(e') \mid e' \in E(P_{t,v})\}$, where possibly $\delta = 0$. Consider the function $f' = f + (P_{t,v}, \delta)$. Let $f'_1 : E_{e_1} \to \mathbb{R}^+$ be the function induced from f' by E_{e_1} is a feasible multiflow in I_{e_1}. Let $e^* = (u^*, v^*)$ be an edge in $E(P_{t,v})$ saturated by f', and u^* be the parent of v^*. Now we show that f'_1 is blocking in I_{e_1}. Let $\mathcal{X}_{e_1} = \{X \in \mathcal{X} \mid X \subseteq V_{e_1}\}$. We replace X_t with a new terminal-cut X'_t so that $\mathcal{X}^* = (\mathcal{X}_{e_1} - \{X_t\}) \cup \{X'_t\}$ is an f'_1-blocking cut-system in I_{e_1}. Such a set X'_t is given by the set of descendants of e^* contained in X_t, i.e., $X'_t = X_t \cap (V_{e^*} - \{u^*\})$. Since $c(X_{t'}) = f'_1(X_{t'}) = f'_1(e_{t'})$ for each $t' \in T_{e'} - \{t\}$, we see that $c(X'_t) = f'_1(X'_t) = f'_1(e_t)$, showing that $\mathcal{X}^*$ is f'_1-blocking. Hence f'_1 is a blocking multiflow in I_{e_1}, and $f'_1(e_1) \in \Phi(e_1)$, which is a contradiction that $f'_1(e_1) \geq f(e) > b = b_1 + b_2 \geq b_1$ holds for $\Phi(e_1) = [a_1, b_1]$. Therefore $\Phi(e) \subseteq [\min\{c(e), a\}, \min\{c(e), b\}]$. ∎

By this lemma and (1), we have

$$\Phi(e) = [a, \min\{c(e), b\}] \text{ if } c(e) \geq a.$$

We call an edge e' *dominating* if $a' > b''$ holds for $\Phi(e') = [a', b']$ and $\Phi(e'') = [a'', b'']$ with the sibling e'' of e'. For an edge $e = (u, v)$, where Lemma 9 holds for all edges in $E_e - \{e\}$, if a child-edge e_1 of e is dominating, then we see that there is a terminal $t^* \in T_{e_1}$ such that every edge in P_{v,t^*} is dominating, because a non-dominating edge e' in P_{v,t^*} implies $\Phi(e_p) = [a_p = 0, b_p]$ for the parent-edge e_p of e' by Lemma 9, contradicting $a_p > 0$. We call such a path P_{v,t^*} the *dominating* path of e.

Lemma 11 *For a non-leaf edge $e = (u, v) \in E$, let $\Phi(e_1) = [a_1, b_1]$ and $\Phi(e_2) = [a_2, b_2]$ for the child-edges e_1 and e_2 of e. Assume $a > 0$ holds for $[a, b] = \Phi(e_1) \oplus \Phi(e_2)$, where we assume w.l.o.g. $a_1 > b_2$ holds, and let $t^* \in T_{e_1}$ be the terminal such that P_{v,t^*} is the dominating path of e. Then any blocking multiflow $\tilde{f}$ with $\tilde{f}(e) = a$ in $\widetilde{I}_e$ satisfies the following.*

(i) *Any decomposition g of $\tilde{f}$ satisfies $g(P_{u,t}) = 0$ for all $t \in T_e - \{t^*\}$ (hence $g(P_{u,t^*}) = f(e)$).*

(ii) *If $c(e) \le a$, then the function $f := \tilde{f} + (P_{u,t^*}, -\delta_{u,t^*})$ with $\delta_{u,t^*} = a - c(e)$ is a blocking multiflow in I_e (hence $\Phi(e) \subseteq [c(e), c(e)]$ if $c(e) \le a$).*

Proof. We assume that the lemma holds for any edges in $E_e - \{e\}$. Let $\widetilde{\mathcal{X}}$ be an $\tilde{f}$-blocking cut-system of T_e in $\widetilde{I}_e$, and g be a decomposition of $\tilde{f}$.

(i) We first show that there is a terminal $t' \in T_e$ such that $g(P_{u,t}) = 0$, $t \in T_e - \{t'\}$. Assume indirectly that there are two terminals $t_1, t_2 \in T_e$ such that $g(P_{u,t_1}) > 0$ and $g(P_{u,t_2}) > 0$. Let w be the branch vertex of P_{u,t_1} and P_{u,t_2} (see Fig. 6(a)). Then the path $P_{u,w}$ is vertex-disjoint with any terminal-cut $X_t \in \widetilde{\mathcal{X}}$, since otherwise $u \notin X_t$ implies that one of paths P_{u,t_1} and P_{u,t_2} would contain two edges from $E(X_t)$ contradicting Lemma 7. For $\delta = \min\{g(P_{u,t_1}), g(P_{u,t_2})\}$, the function $f' := \tilde{f} + (P_{u,t_1}, -\delta) + (P_{u,t_2}, -\delta) + (P_{t_1,t_2}, \delta)$ is a feasible multiflow such that $f'(e') < \tilde{f}(e')$ for all edges e' along path $P_{u,w}$, and $f'(e') = \tilde{f}(e')$ for all other edges e'. Since $P_{u,w}$ is vertex-disjoint with any terminal-cut $X_t \in \widetilde{\mathcal{X}}$, we see that $\widetilde{\mathcal{X}}$ is f'-blocking and f' is a blocking multiflow with $f'(e) < a$ in $\widetilde{I}_e$. This, however, contradicts the minimality of a, proving the existence of such a special terminal $t' \in T_e$.

We next show that t' is the terminal t^* of the dominating path P_{u,t^*} of e. Note that $a > 0$ implies either $a_1 > b_2$ or $a_2 > b_1$. By assumption of $a_1 > b_2$, we see that $t^* \in T_{e_1}$, $\tilde{f}(e_1) = a_1$, $\tilde{f}(e_2) = b_2$ and $\tilde{f}(e) = a = a_1 - b_2 = \tilde{f}(e_1) - \tilde{f}(e_2)$. Since $\tilde{f}(e) = \tilde{f}(e_1) - \tilde{f}(e_2)$ means $g(P_{u,t}) = 0$ for any $t \in T_{e_2}$, it holds $t' \in T_{e_1}$. If e_1 is a leaf-edge, then $t^*, t' \in T_{e_1}$ implies that $t' = t^*$ holds. Assume that e_1 is not a leaf-edge, let e'_1 and e'_2 be the two child-edges of $e_1 = (v, v')$, and denote $[a', b'] = [a'_1, b'_1] \oplus [a'_2, b'_2]$ for $\Phi(e'_i) = [a'_i, b'_i]$, $i = 1, 2$. Recall that $a_1 > 0$. Since $a_1 = \min\{a', c(e_1)\} > 0$ and thereby $a' > 0$, we see that $a'_1 > b'_2$ holds without loss of generality. By applying Lemma 11 to edge e_1 with $a'_1 > 0$, we see that t' is the terminal for the dominating path $P_{v',t'}$ of e_1. Since $P_{v',t'}$ needs to be a subpath of the dominating path P_{u,t^*}, it must hold $t' = t^*$.

(ii) Let $Y \subseteq V_e - (X_{t^*} \cup V(P_{u,t^*}))$ be the set of vertices reachable from a vertex in P_{u,t^*} via edges unsaturated by $\tilde{f}$. We claim that any positive path P_{t_1,t_2} between terminals $t_1, t_2 \in T_e - \{t^*\}$ is vertex-disjoint with $V(P_{u,t^*}) \cup Y$. Assume indirectly that there is a positive path P_{t_1,t_2} between some terminals $t_1, t_2 \in T_e - \{t^*\}$ that contains a vertex in $V(P_{u,t^*}) \cup Y$. Let w be the branch vertex of P_{u,t_1} and P_{u,t_2}, and w' be the branch vertex of P_{u,t^*} and P_{u,t_2}, where we assume without loss of generality that the branch vertex of P_{u,t^*} and P_{u,t_1} is not a descendant of w' (see Fig. 6(b),(c)). By Lemma 7, any terminal-cut $X \in \widetilde{\mathcal{X}} - \{X_{t^*}\}$ is vertex-disjoint with P_{u,t^*}, and P_{t_1,t_2} is vertex-disjoint with X_{t^*}. Since Y is spanned by unsaturated edges, any terminal-cut $X \in \widetilde{\mathcal{X}} - \{X_{t^*}\}$ is disjoint with Y.

We consider the case where $V(P_{t_1,t_2}) \cap V(P_{u,t^*}) = \emptyset$ and $V(P_{t_1,t_2}) \cap Y \neq \emptyset$, as in Fig. 6(c) (the case where $V(P_{t_1,t_2}) \cap V(P_{u,t^*}) \neq \emptyset$ in Fig. 6(b) can be treated analogously by setting $\delta' = \infty$ below). Let $\delta' = \min\{c(e') - \tilde{f}(e') \mid e' \in E(P_{w,w'})\}$. For $\delta = \min\{a, g(P_{t_1,t_2}), \delta'/2\}$, the function $f' := \tilde{f} + (P_{t_1,t_2}, -\delta) + (P_{u,t^*}, -\delta) + (P_{u,t_1}, \delta) + (P_{t^*,t_2}, \delta)$ is a feasible multiflow in $\widetilde{I}_e$. Then we see that $\widetilde{\mathcal{X}}$ remains f'-blocking, but f' admits a decomposition g' such that $g'(P_{u,t_1}) = g(P_{u,t_1}) + \delta > 0$, a contradiction to (i). This proves our claim.

We are ready to prove that $f = \tilde{f} + (P_{t^*,u}, -\delta_{u,t^*})$ is blocking in I_e. For this, we modify the terminal-cut $X_{t^*} \in \widetilde{\mathcal{X}}$. Let X'_{t^*} be the set of vertices reachable from t via edges not saturated by f. Hence $X'_{t^*} = X_{t^*} \cup Y \cup V(P_{v,t^*})$. We show that $\mathcal{X} = (\widetilde{\mathcal{X}} - \{X_{t^*}\}) \cup \{X'_{t^*}\}$ is f-blocking. Since $f(e') = \tilde{f}(e')$ holds for all edges e' not in $P_{t^*,u}$, every terminal-cut $X_t \in \widetilde{\mathcal{X}}$ with $t \in T_e - \{t^*\}$ remains tight in f. Since Y is spanned by unsaturated edges, X'_{t^*} is disjoint with any other terminal-cuts $X \in \widetilde{\mathcal{X}} - \{X_t\}$ and $\mathcal{X}$ is a cut-system in I_e. The remaining task is to show that terminal-cut X'_{t^*} is tight in f. By the choice of Y, all edges in $E(X'_{t^*})$ are saturated by f. By Lemma 7 and the above claim, for any decomposition g of f, any positive T_e-path P_{t_1,t_2} with $t_1, t_2 \in T_e - \{t^*\}$ is vertex-disjoint with each of X_t, P_{u,t^*} and Y. This proves that $f(X'_{t^*}) = f(e_{t^*})$. Hence X'_{t^*} is tight in f, as required. ∎

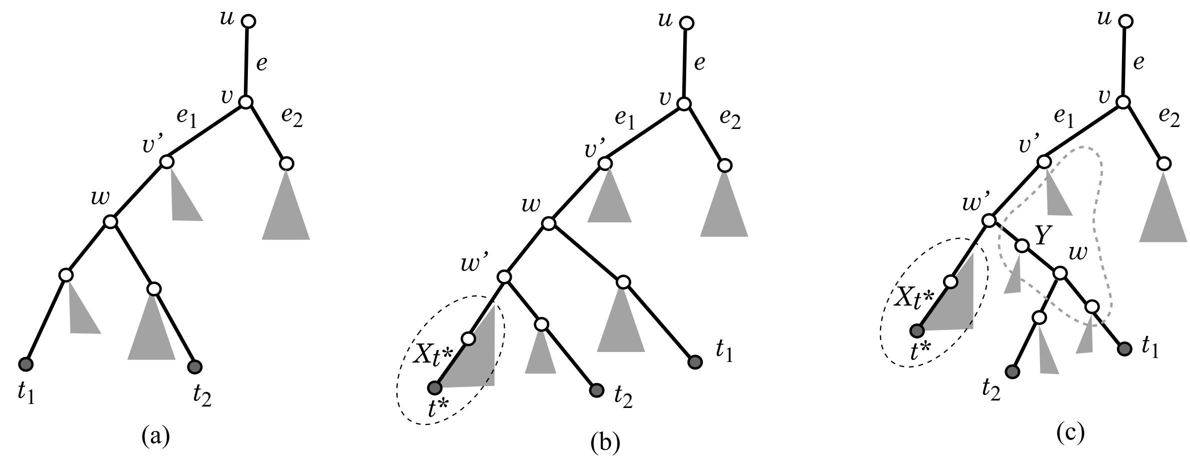

Figure 6: (a) Positive paths P_{u,t_1} and P_{u,t_2}: (b) a positive path P_{t_1,t_2} intersecting $P_{u,t}$: (c) a positive path P_{t_1,t_2} intersecting Y.

Summarizing the above arguments completes the proof of Lemma 9. Based on Lemma 9 we can easily compute $\Phi(e)$ for all edges $e \in E$ in linear time in a bottom up manner (see Fig. 7(a) for an example).

After computing $\Phi(e)$ for all edges $e \in E$ in a standard instance $I = (G, T, c)$. We construct a maximum multiflow f in I as follows. First, choose a terminal r as the root of the tree. For the edge e_r of the root $r \in T$, we set $f(e_r) := \max\{x \in \Phi(e_r)\}$ by Lemma 6. The remaining task is to determine $f(e)$ of the other edges $e \in E$ to obtain a blocking multiflow f with $f(e_r) = \max\{x \in \Phi(e_r)\}$ in I.

For the two child-edges e_1 and e_2 of e_r, if $f(e_r) \in \Phi(e_1) \oplus \Phi(e_2)$, then we set $f(e_1) := y$ and $f(e_2) := z$ for some real $y \in \Phi(e_1)$, $z \in \Phi(e_2)$ and $s \in [0, \min\{y, z\}]$ such that $f(e_r) = y + z - 2s$. Hence for all the other edges e' in the descendant, we can repeatedly determine the flow value $f(e'_i)$ of their child-edges e' in this way as long as $f(e') \in \Phi(e'_1) \oplus \Phi(e'_2)$ holds for the child-edges e'_1 and e'_2 of e'.

Now we consider the case where $c(e) < a$ holds for an edge $e = (u, v)$, its child-edges e_1 and e_2 and $[a, b] = \Phi(e_1) \oplus \Phi(e_2)$, where $f(e) \notin \Phi(e_1) \oplus \Phi(e_2)$ holds for the value $f(e)$ determined at the parent-edge e. In this case, $c(e) < a$ holds, and a blocking multiflow f with $f(e) = c(e)$ in I_e is obtained by $f := \tilde{f} + (P_{u,t^*}, -\delta_{u,t^*})$, where $\delta_{u,t^*} = a - c(e)$ and $t^* \in T_e$ is the unique terminal in Lemma 11, and $\tilde{f}$ is a blocking multiflow with $\tilde{f}(e) = a$ in $\widetilde{I}_e$. In our algorithm, we keep δ_{u,t^*} as $\delta(e)$ by setting $\delta(e) := \delta(e) + \delta_{u,t^*}$ and initially $\delta(e) := 0$ (where the old $\delta(e)$ may have received the values δ_{u',t^*} of some ancestors $e' \in E$ of e, as will be explained below), and carry it to the dominating edges in P_{u,t^*}. One of the child-edges e_1 and e_2 of e must be the first one of such dominating edges. More specifically, if $b_2 < a_1$ (resp., $b_1 < a_2$) holds for $\Phi(e_1) = [a_1, b_1]$ and $\Phi(e_2) = [a_2, b_2]$ then e_1 (resp., e_2) is the dominating edge and we set $\delta(e_1) := \delta(e)$ (resp., $\delta(e_2) := \delta(e)$). In this way, we can carry δ_{u,t^*} to all the edges in the dominating path P_{u,t^*} without explicitly finding out the special terminal t^* of the edge e. Thus we keep all such values δ_{u,t^*} as an accumulated form of $\delta(e) = \sum_{e \in E(P_{u,t^*})} \delta_{u,t^*}$ for all $e \in E$ until we fix $f(e)$ of all edges $e \in E$ in the above way. Finally we compute a blocking multiflow f^* with $f^*(e_r) := \max\{x \in \Phi(e_r)\}$ in I by $f^*(e) := f(e) - \delta(e)$, $e \in E$. See Fig. 7(b) and (c) for an example.

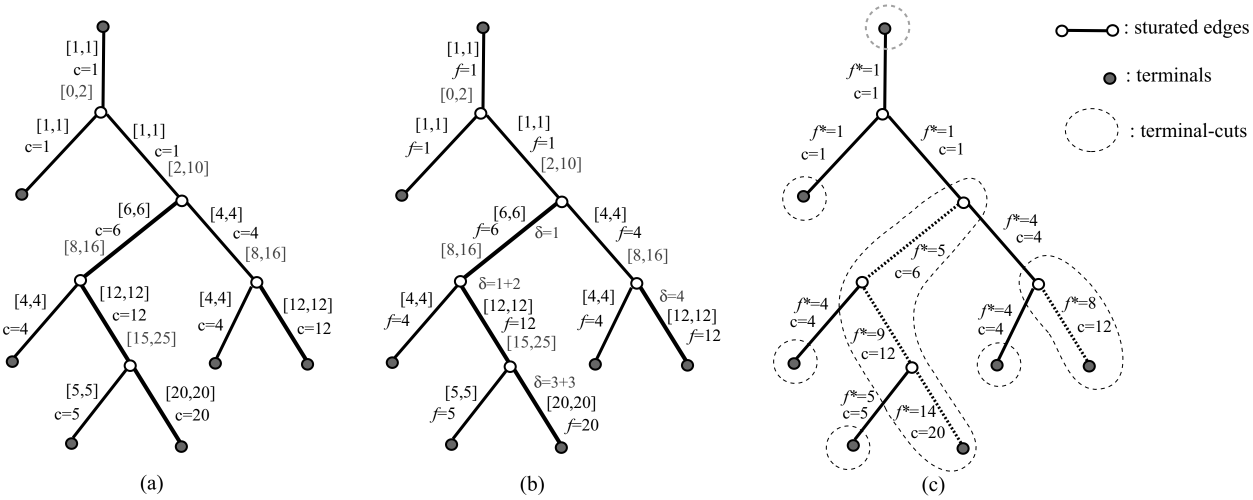

Figure 7: (a) A process of computing $\Phi(e)$, $e \in E$ in an example, where the dominating edges are drawn as thick lines, the number c beside each edge e denotes its capacity $c(e)$ and $[a, b] = \Phi(e_1) \oplus \Phi(e_2)$ for each pair of siblings e_1 and e_2 is shown in grey: (b) a process of computing $f(e)$ and $\delta(e)$, $e \in E$, where the number f beside each edge e denotes $f(e)$ and the number δ beside each dominating edge e indicates $\delta(e)$: (c) a maximum multiflow f^* and a cut-system $\mathcal{X}$ with $\alpha(f) = \gamma(\mathcal{X})/2$.

The entire algorithm is described as follows. We easily see that the algorithm can be implemented to run in linear time.

Algorithm DPTreeFlow
Input: A standard tree instance $I = (G = (V, E), T, c)$ and a terminal $r \in T$ as the root of G.
Output: A blocking multiflow f^* with $f^*(e_r) = \max\{x \in \Phi(e_r)\}$ in I.
  Let $\delta(e) := 0$ for all edges $e \in E$;
  Let $\Phi(e_t) := [c(e_t), c(e_t)]$ for all leaf-edges e_t, $t \in T - \{r\}$;
  for each non-leaf-edge $e \in E$ to be picked up in a nonincreasing order of their depths **do**
    $[a, b] := \Phi(e_1) \oplus \Phi(e_2)$ for the child-edges e_1 and e_2 of e;
    $\Phi(e) := [\min\{c(e), a\}, \min\{c(e), b\}]$
  end; /* for */
  Let $f(e_r) := b$ for $\Phi(e_r) = [a, b]$;
  for each non-leaf edge $e \in E$ to be picked up in a nondecreasing order of their depths **do**
    /* Let $\Phi(e_i) = [a_i, b_i]$ for the child-edges e_i, $i = 1, 2$ of e */

if $f(e) \le a$ for $[a,b] = \Phi(e_1) \oplus \Phi(e_2)$ **then** /* $f(e) = c(e)$ */
$\delta(e) := \delta(e) + a - c(e)$; /* $a - c(e)$ is the δ_{u,t^*} of the edge $e = (u,v)$ in Lemma 11 */
if $b_2 < a_1$ **then** /* e_1 is a dominating edge */
$f(e_1) := a_1$; $f(e_2) := b_2$; $\delta(e_1) := \delta(e)$
else /* $b_1 < a_2$, and e_2 is a dominating edge */
$f(e_1) := b_1$; $f(e_2) := a_2$; $\delta(e_2) := \delta(e)$
end /* if */
else /* $a < f(e) \in [a,b]$ */
Choose reals $y \in \Phi(e_1), z \in \Phi(e_2)$ and $s \in [0, \min\{y,z\}]$ with $f(e) = y + z - 2s$;
$f(e_1) := y$; $f(e_2) := z$
end /* if */
end; /* for */
$f^*(e) := f(e) - \delta(e)$ for all edges $e \in E$.

Theorem 12 *A feasible multiflow f and a cut-system $\mathcal{X}$ with $\gamma(\mathcal{X}) = \alpha(f)$ in a standard instance (G,T,c) can be found in $O(n)$ time and space, where f is a maximum multiflow.*

Proof. As we have observed, algorithm DPTreeFlow constructs a blocking multiflow f^* in I with $f^*(e_r) = \max\{x \in \Phi(e_r)\}$. By Lemma 6, a cut-system $\mathcal{X}$ with $\alpha(f^*) = \gamma(\mathcal{X})/2$ can be constructed from f^* in linear time, and f^* is a maximum multiflow in I. ∎

4.2 Maximum Integral Multiflows in Trees

In this section, we restrict ourselves to standard instances $I = (G = (V,E), T, c)$ with positive integer capacities $c(e) \in \mathbb{Z}^+$, $e \in E$, and integral multiflows f.

Choose a terminal $r \in T$, and regard G as a tree rooted at r. Consider a cut-system $\mathcal{X}$ in G. Let W be a vertex subset which induces such a connected component in $G[V - V_{\mathcal{X}}]$. Then the path $P_{w,r}$ from a vertex $w \in W$ to the root r contains exactly one edge from $E(W)$, which we call the *parent-edge* e_W of W. For each $X_t \in \mathcal{X}$, let *odd*(X_t) denote the family of odd sets W of $\mathcal{X}$ whose parent-edge e_W is incident to X_t (hence $r \notin W$). Fig. 8 illustrates a cut-system $\mathcal{X}$ and the family *odd*$(X_t) = \{W_1, W_2\}$.

Let f be a feasible integral multiflow in I. We say that a terminal-cut $X_t \in \mathcal{X}$ with $X_t \cap T = \{t\}$ is *semi-tight* in f if $f(e_t) = f(X_t) = c(X_t) - |odd(X_t)|$. Consider the instance $I_e = (G_e, T_e \cup \{u\}, c)$ with an edge $e = (u,v) \in E$ and a cut-system $\mathcal{X}$ of T_e in G_e.

Let f be a feasible integral multiflow in I_e. We say that a terminal-cut $X_t \in \mathcal{X}$ with $X_t \cap T = \{t\}$ is *semi-tight* in f if $f(e_t) = f(X_t) = c(X_t) - |odd(X_t)|$. We say that a feasible integral multiflow f in I_e *blocks* a cut-system $\mathcal{X}$ of T_e if each terminal-cut $X_t \in \mathcal{X}$ is semi-tight in f. Such f and $\mathcal{X}$ are called *blocking* and *f-blocking*, respectively. An f-blocking cut-system $\mathcal{X}$ satisfies the following property. For any terminal-cut $X_t \in \mathcal{X}$ with *odd*$(X_t) = \emptyset$, all edges in $E(X_t)$ are saturated by f, and we see by induction that, any terminal-cut $X_t \in \mathcal{X}$ with *odd*$(X_t) \neq \emptyset$, every unsaturated edge $e' = (u', u'') \in E(X_t)$ is the parent-edge of an odd set $W \in odd(X_t)$ and satisfies $f(e') = c(e') - 1$.

Moreover, any terminal-cut X_t in an f-blocking cut-system in I_e induces a connected subgraph, since otherwise $G_e[X_t]$ would have a component $X' \subset X_t$ such that $E(X')$ has an unsaturated edge which cannot be the parent-edge of any odd set $W \in odd(X_t)$.

Let $\Psi(e)$ denote the set of nonnegative integers $x \in \mathbb{Z}^+$ such that I_e has a blocking multiflow f such that $f(e) = x$. For each leaf-edge e_t with a terminal $t \in T - \{r\}$, we see that $\Psi(e_t) =$

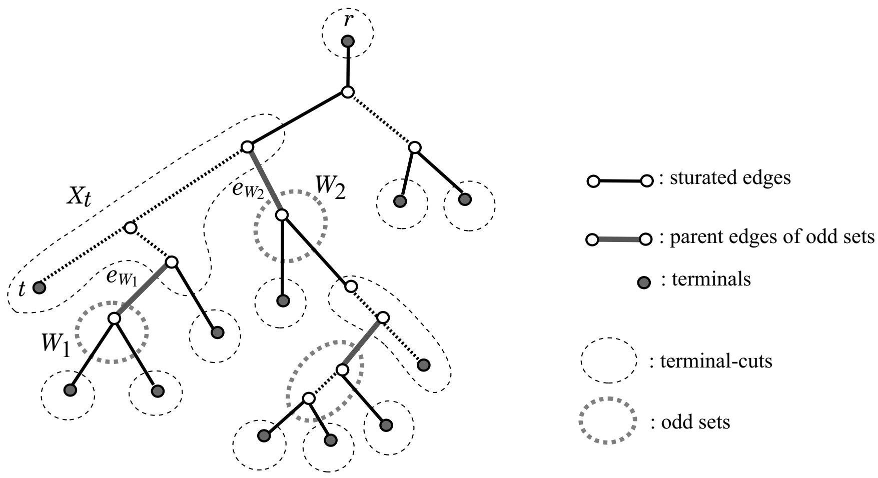

Figure 8: Illustration of a cut-system $\mathcal{X}$ and the family $odd(X_t) = \{W_1, W_2\}$ for a terminal-cut $X_t \in \mathcal{X}$.

$[c(e_t), c(e_t)]$ holds, since $f(e_t) = c(e_t)$ blocks a cut-system $\{X_t = \{t\}\}$. We first observe that the set $\Psi(e_r)$ for the edge $e_r = (r, v) \in E$ incident to the root $r \in T$ tells how to get a maximum integral multiflow f in $I = I_{e_r}$.

Lemma 13 *For an instance $I = (G, T, c)$ rooted at a terminal $r \in T$, let f be a blocking integral multiflow with $f(e_r) = \max\{x \in \Psi(e_r)\}$. Then a cut-system $\mathcal{X}$ of T such that $\alpha(f) = \frac{1}{2}[\gamma(\mathcal{X}) - \kappa(\mathcal{X})]$ can be constructed from f in linear time (hence f is a maximum integral multiflow in I).*

Proof. For the terminal $r \in T$, let X_r be the set of vertices reachable from r via unsaturated edges e with $f(e) \leq c(e) - 2$. For each terminal $t \in T - \{r\}$, let X_t be the set of vertices in $V - X_r$ reachable from t via unsaturated edges e such that $e \in E(P_{t,r})$ or $f(e) \leq c(e) - 2$. We show that $\mathcal{X} = \{X_t \mid t \in T\}$ is a cut-system satisfying $\sum_{t \in T} f(e_t) = \gamma(\mathcal{X}) - \kappa(\mathcal{X})$.

We first show that every two sets in $\mathcal{X}$ are disjoint. Since f is a blocking multiflow in $I_{e_r} = I$, there is an f-blocking cut-system $\mathcal{X}' = \{X'_t \mid t \in t \in X'_t,\ T - \{r\}\}$ in I. By the property of $\mathcal{X}'$, each edge in $E(X'_t) \cap E(P_{t,r})$ is saturated and each edge $e \in E(X'_t) - E(P_{t,r})$ satisfies $f(e) \in \{c(e) - 1, c(e)\}$. Hence, for each $t \in T - \{r\}$, we have $X_t \subseteq X'_t$ by construction of X_t. Similarly X_r is disjoint with any terminal-cut $X'_t \in \mathcal{X}'$, since the vertices in X_r are spanned with unsaturated edges e with $f(e) \leq c(e) - 2$. Hence every two sets in $\mathcal{X}$ are disjoint, and $\mathcal{X}$ is a cut-system of T, which defines $odd(X_t)$ for each $X_t \in \mathcal{X}$. By construction of $\mathcal{X}$, we have $f(X_t) = c(X_t) - |odd(X_t)|$ for any terminal $t \in T$.

We next show that $f(e_t) = c(X_t) - |odd(X_t)|$ for all terminal $t \in T$. Since $f(X_t) = c(X_t) - |odd(X_t)|$, it suffices to show $f(e_t) = f(X_t)$. Let $t \neq r$. By definition of $\mathcal{X}'$, for each $t \in T - \{r\}$, it holds $f(e_t) = f(X'_t)$ and any positive path $P_{t,t'}$, $t' \in T - \{t\}$ in a decomposition g of f also passes through an edge in $E(X_t)$ exactly one, since $X_t \subseteq X'_t$. Hence $f(X'_t) = f(X_t) = f(e_t)$.

We now show that $f(e_r) = f(X_r)$. If $f(e_r) = c(e_r)$, then we have $X_r = \{r\}$ and $f(e_r) = f(X_r)$. Consider the case where $f(e_r) = \max\{x \in \Psi(e_r)\} < c(e_r)$. Let g be a decomposition of f. We claim that any positive path P_{t_1,t_2} for $t_1, t_2 \in T - \{r\}$ is vertex-disjoint with X_r. Assume indirectly that a positive path P_{t_1,t_2} contains a vertex in X_r. Let w be the branch vertex of $P_{t_1,r}$ and $P_{t_2,r}$ (see Fig. 5(a)). The function $f' := f + (P_{t_1,t_2}, -1) + (P_{t_1,r}, 1) + (P_{t_2,r}, 1)$ is a feasible integral multiflow in I, since X_r is spanned with unsaturated edges e with $f(e) \leq c(e) - 2$.

Since $f'(e') = f(e')$ for all edges $e' \in E - E(P_{r,w})$, the multiflow f' blocks $\mathcal{X}$, and thereby f' is a blocking multiflow in I_{e_r} with $f'(e_r) > f(e_r) = \max\{x \in \Psi(e_r)\}$, which contradicts the definition of $\Psi(e_r)$. Hence any positive path P_{t_1,t_2} with $t_1, t_2 \in T - \{r\}$ is vertex-disjoint with X_r. This proves that $f(e_r) = f(X_r)$. Therefore, we have $\sum_{t \in T} f(e_t) = \sum_{t \in T}(c(X_t) - |odd(X_t)|) = \gamma(\mathcal{X}) - \kappa(\mathcal{X})$.

Since every two sets in $\mathcal{X}$ are disjoint, $\mathcal{X}$ can be obtained in linear time. ∎

For two integers $k, a \in \mathbb{Z}^+$, the set $\{a + 2i \mid i = 0, 1, \ldots, k\}$ of consecutive odd or even integers is denoted by $\langle a, b\rangle$, where $b = 2k + a$. For two sets $A, B \subseteq \mathbb{Z}^+$ of nonnegative integers, let $A \odot B$ denote the set of nonnegative integers $\{a + b - 2i \mid i = 0, 1, \ldots, \min\{a, b\}\}$ over all $a \in A$ and $b \in B$. In particular, for sets $A_1 = \langle a_1, b_1\rangle$ and $A_2 = \langle a_2, b_2\rangle$, we observe that

$$A_1 \odot A_2 = \begin{cases} \langle 0, b_1 + b_2\rangle & \text{if } A_1 \cap A_2 \neq \emptyset \\ \langle a_1 - b_2, b_1 + b_2\rangle & \text{if } b_2 < a_1 \\ \langle a_2 - b_1, b_1 + b_2\rangle & \text{if } b_1 < a_2 \\ \langle 1, b_1 + b_2\rangle & \text{if } a_2 \leq b_1,\ a_1 \leq b_2 \text{ and } A_1 \cap A_2 = \emptyset. \end{cases}$$

Analogously with the previous section, we see that, for a non-leaf edge e with its child-edges e_1 and e_2, it holds

$$(\Psi(e_1) \odot \Psi(e_2)) \cap [0, c(e)] \subseteq \Psi(e). \tag{2}$$

We will show the converse.

Lemma 14 *For each leaf edge $e \in E$, it holds $\Psi(e) = \langle c(e), c(e)\rangle$. For each non-leaf edge $e \in E$ with its child-edges e_1 and e_2, let $\Psi(e_1) = \langle a_1, b_1\rangle$ and $\Psi(e_2) = \langle a_2, b_2\rangle$. Then for $\langle a, b\rangle = \langle a_1, b_1\rangle \odot \langle a_2, b_2\rangle$, it holds*

$$\Psi(e) = \begin{cases} \langle a, b\rangle \cap [0, c(e)] & \text{if } c(e) \geq a \\ \langle c(e), c(e)\rangle & \text{otherwise.} \end{cases}$$

Note that $\langle a, b\rangle \cap [0, c(e)]$ is given by $\langle a, b'\rangle$ such that $b' = b$ if $c(e) \geq b$, $b' = c(e)$ if $c(e) \in \langle a, b\rangle$, and $b' = c(e) - 1$ if $c(e) - 1 \in \langle a, b\rangle$. We prove Lemma 14 via several lemmas in the following.

Lemma 15 *For an edge $e = (u, v) \in E$ and a blocking multiflow f in $I_e = (G_e, T_e \cup \{u\}, c)$, let $\mathcal{X}$ be an f-blocking cut-system of T_e in I_e, and g be an integer individual flow decomposed from f. Then each positive $T_e \cup \{u\}$-path P contains at most one edge from $E(X)$ for any terminal-cut $X \in \mathcal{X}$. In particular, the terminal-cut $X_t \in \mathcal{X}$ of each terminal t is vertex-disjoint with any positive path $P_{t',t''}$ between terminals $t', t'' \in T_e \cup \{u\} - \{t\}$.*

Proof. The lemma can be shown as in Lemma 7. ∎

Lemma 16 *For an edge $e = (u, v) \in E$, let f be a blocking multiflow f in $I_e = (G_e, T_e \cup \{u\}, c)$. Then for any edge $e' \in E_e$ with $f(e') = c(e')$, it holds $c(e') \in \Psi(e')$.*

Proof. Let $e' = (u', v') \in E_e$ satisfy $f(e') = c(e')$, where u' is the parent of v'. It suffices to show that the multiflow f' induced from f by $V_{e'}$ is a blocking multiflow in $I_{e'}$. For this, we modify an f-blocking cut-system $\mathcal{X}$ of T_e in I_e. Let $\mathcal{X}'$ be the family of terminal-cuts $X_t \in \mathcal{X}$ with $t \in T_{e'}$. If all terminal-cuts $X_t \in \mathcal{X}'$ are contained in $V_{e'} - \{u'\}$, then the set $odd(X_t)$ of each terminal-cut $X_t \in \mathcal{X}'$ remains unchanged, and $\mathcal{X}'$ is an f'-blocking cut-system of $T_{e'}$ in $I_{e'}$. Assume that there is a terminal-cut $X_{t^*} \in \mathcal{X}'$ such that $u' \in X_{t^*}$. Then we replace X_{t^*} with

$X'_{t^*} = X_{t^*} \cap (V_{e'} - \{u'\})$ to obtain $\mathcal{X}'' = (\mathcal{X}' - \{X_{t^*}\}) \cup \{X'_{t^*}\}$. In $\mathcal{X}''$, the set $odd(X'_{t^*})$ is given by $\{W \in odd(X_{t^*}) \mid W \subseteq V_{e'}\}$. Note that X'_{t^*} remains semi-tight in f' since e' is saturated by f'. We see that $\mathcal{X}''$ is an f'-blocking cut-system of $T_{e'}$ in $I_{e'}$. Therefore, $f(e') = f'(e') \in \Psi(e')$. ∎

Lemma 17 *For an edge $e = (u, v) \in E$, let f be a blocking multiflow f in $I_e = (G_e, T_e \cup \{u\}, c)$, and $\mathcal{X}$ be an f-blocking cut-system of T_e in I_e. Then for the parent-edge e_W of any odd set W of $\mathcal{X}$, it holds $f(e_W) = c(e_W) - 1 \in \Psi(e_W)$.*

Proof. Let $e_W = (u_W, v_W)$, where u_W is the parent of v_W. It suffices to show that the multiflow f' induced from f by V_{e_W} is a blocking multiflow in I_{e_W}. For this, we modify $\mathcal{X}$ into an f'-blocking cut-system in I_{e_W}. Let $\mathcal{X}'$ be the family of terminal-cuts $X_t \in \mathcal{X}$ with $t \in T_{e_W}$. Since vertex u_W is contained in a terminal-cut $X_{t_1} \in \mathcal{X} - \mathcal{X}'$, all terminal-cuts $X_t \in \mathcal{X}'$ are contained in $V_{e_W} - (W \cup \{u_W\})$. Hence the set $odd(X_t)$ of each terminal-cut $X_t \in \mathcal{X}'$ remains unchanged. Therefore $\mathcal{X}'$ is an f'-blocking cut-system of T_{e_W} in I_{e_W}, and $c(e_W) - 1 = f(e_W) = f'(e_W) \in \Psi(e_W)$. ∎

In what follows, we prove Lemma 14 by induction on height of edges e. Thus we assuming that $\Psi(e')$ for each edge e' in G_e is given as an interval $\langle a, b \rangle$ determined by Lemma 14. For the non-leaf edge e given in Lemma 14, we prove the following two lemmas, from which Lemma 14 follows.

Lemma 18 *For $\langle a, b \rangle = \Psi(e_1) \odot \Psi(e_2)$, it holds $\Psi(e) \subseteq (\langle a, b \rangle \cap [0, c(e)]) \cup \{c(e)\}$.*

Proof. We can carry over the same argument in the proof of Lemma 10 to show that $f(e) \in \Psi(e_1) \odot \Psi(e_2)$ holds for a blocking integral multiflow f in I_e if I_e has an f-blocking cut-system $\mathcal{X}$ such that there is no terminal-cut $X_t \in \mathcal{X}$ with $e \in E(X_t)$. Hence we see that $x \notin \Psi(e)$ if $x \notin \Psi(e_1) \odot \Psi(e_2) = \langle a, b \rangle$ and $x < c(e)$.

The remaining task is to prove that, for any integer $x > a$ with $x \notin \langle a, b \rangle$, it holds $x \notin \Psi(e)$. Let x be such an integer, and assume indirectly that I_e has a blocking multiflow f with $f(e) = x > a$. Let $\mathcal{X}$ be an f-blocking cut-system of T_e in I_e. As claimed in the proof of Lemma 10, it would hold $f(e) \in \langle a, b \rangle = \Psi(e_1) \odot \Psi(e_2)$ if $\mathcal{X}$ has no terminal-cut $X_t \in \mathcal{X}$ with $e \in E(X_t)$. Hence $\mathcal{X}$ contains a terminal-cut $X_t \in \mathcal{X}$ such that $e \in E(X_t)$. Let $e'_0, e'_1, \ldots, e'_p$ be the edges in $P_{u,t}$ that appear in this order from $e'_0 = e$ to $e'_p = e_t$ (see Fig. 9(a)). Let e''_i $(i = 1, 2, \ldots, p)$ be the other edge that shares the same parent with e'_i, where $\{e'_1, e''_1\} = \{e_1, e_2\}$. By induction hypothesis on edges with smaller height than that of e, we have $\Psi(e'_i) = \langle a'_i, b'_i \rangle$ and $\Psi(e''_i) = \langle a''_i, b''_i \rangle$ for all edges e'_i and e''_i.

For a subset $Y \subseteq V_e$, let $odd(X_t; Y)$ denote the family of odd sets $W \in odd(X_t)$ with $W \subseteq Y$. By Lemma 15 and semi-tightness of X_t, any positive $T_e \cup \{u\}$-path P in a decomposition g of f passing an edge $E(X_t)$ must connect t and other terminals. Hence we have $f(e'_{i+1}) = f(e'_i) + f(e''_{i+1})$ $(i = 1, 2, \ldots, p-1)$. Moreover, $c(X_t) - |odd(X_t)| = f(e_t) = f(X_t)$ implies that, for each $i = 1, 2, \ldots, p-1$, we have

$$c(X_t, V_{e''_i} - X_t) - |odd(X_t; V_{e''_i})| = f(e''_i),$$

$$f(e_W) = c(e_W) - 1 \text{ for the parent-edge } e_W \text{ of each odd set } W \in odd(X_t; V_{e''_i}).$$

We here prove that $f(e''_i) \geq b''_i$ for each $i = 1, 2, \ldots, p-1$. For some i, assume indirectly that $f(e''_i) < b''_i$. Since $b''_i \in \Psi(e''_i)$, the instance $I_{e''_i}$ has a blocking multiflow h with $h(e''_i) = b''_i$ (see Fig. 9(b)). Since $f(X_t, V_{e''_i} - X_t) = f(e''_i) < b''_i = h(e''_i) \leq h(X_t, V_{e''_i} - X_t)$ and $f(e') =$

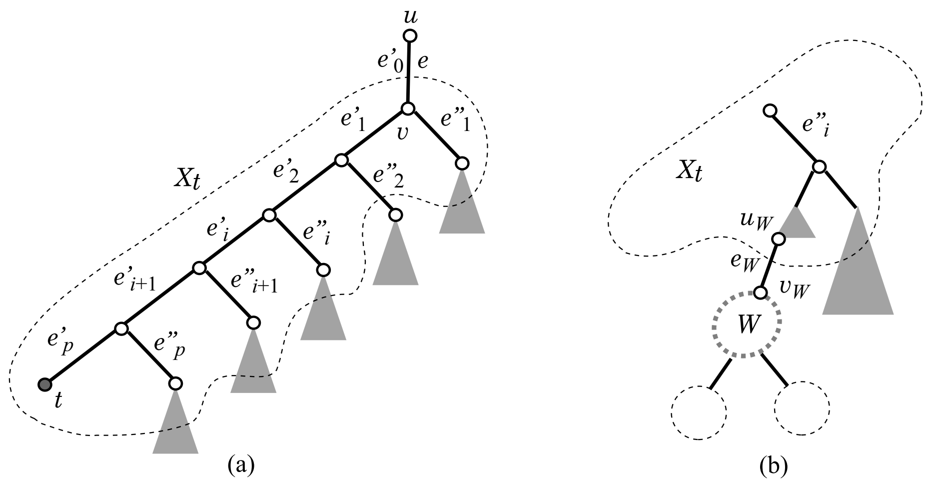

Figure 9: (a) Terminal-cut X_t with $e \in E(X_t)$ and path $P_{u,t}$: (b) an edge e''_i with $h(e''_i) = b''_i$ and an odd set $W \in odd(X_t; V_{e''_i})$.

$\min\{c(e'), c(e')-1\}$, $e' \in E(X_t, V_{e''_i} - X_t)$, we see that there must be an odd set $W \in odd(X_t; V_{e''_i})$ such that $c(e_W) - 1 = f(e_W) < h(e_W) = c(e_W)$. Now $\Psi(e_W)$ is given by $\langle a_W, b_W \rangle$ for some integers a_W and b_W. By applying Lemma 16 to the multiflow h at the saturated edge e_W, we have $c(e_W) = h(e_W) \in \Psi(e_W)$. On the other hand, by applying Lemma 17 to f at the parent edge e_W, we have $c(e_W) - 1 = f(e_W) \in \Psi(e_W)$. However, $c(e_W) - 1, c(e_W) \in \langle a_W, b_W \rangle$ is a contradiction. Therefore $f(e''_i) \geq b''_i$ for each $i = 1, 2, \ldots, p-1$.

By induction on i, we will prove that $f(e'_i) > a_i$ implies $f(e'_{i+1}) > a'_{i+1}$. For $i = 0$, it holds $f(e_0) = f(e) > a = a'_0$ by assumption. Assume $f(e'_i) \geq a'_i + 1$ for some i. Note that if $a'_{i+1} > b''_{i+1}$ then $a'_i = a'_{i+1} - b''_{i+1}$ holds, which indicates $b''_{i+1} \geq a'_{i+1} - a'_i$ holds (even if $b''_{i+1} \geq a'_{i+1}$). Then $f(e'_{i+1}) = f(e'_i) + f(e''_{i+1}) > a'_i + b''_{i+1} \geq a'_i + (a'_{i+1} - a'_i) = a'_{i+1}$, as claimed. For $i = p$, we have $f(e_t) = f(e'_p) > a'_p$. This, however, contradicts that $f(e_t) \leq c(e_t) = a'_p = b'_p$. Therefore, $x \notin \Psi(e)$ for any $x > a$ with $x \notin \langle a, b \rangle$. ∎

By Lemma 18, we see that $\Psi(e) \subseteq \langle c(e), c(e) \rangle$ if $c(e) < a$, and $\Psi(e) = \langle a, b \rangle \cap [0, c(e)] = \langle a, b' \rangle$ if $c(e) \geq a$ (since $\langle a, b' \rangle \subseteq \Psi(e)$ if $c(e) \geq a$ by (2)). Finally consider the case of $c(e) < a$, where it holds $a \geq 2$ by the assumption of $c(e) \geq 1$. The next lemma proves $c(e) \in \Psi(e)$ if $c(e) < a$.

We call an edge e' *dominating* if $a' \geq b'' + 2$ holds for $\Psi(e') = \langle a', b' \rangle$ and $\Psi(e'') = \langle a'', b'' \rangle$ with the sibling e'' of e'. For an edge $e = (u, v)$, where Lemma 14 holds for all edges in $E_e - \{e\}$, if a child-edge e_1 of e is dominating, then we see that there is a terminal $t^* \in T_{e'}$ such that every edge in P_{v,t^*} is dominating, because a non-dominating edge e' in P_{v,t^*} implies that $a_p \leq 1$ holds for $\Psi(e_p) = \langle a_p, b_p \rangle$ of the parent-edge e_p of e' by Lemma 14, contradicting $a_p \geq 2$. We call such a path P_{v,t^*} the *dominating* path of e.

Lemma 19 *For a non-leaf edge $e = (u, v) \in E$, let $\Psi(e_1) = \langle a_1, b_1 \rangle$ and $\Psi(e_2) = \langle a_2, b_2 \rangle$ for the child-edges e_1 and e_2 of e. Assume $a \geq 2$ holds for $\langle a, b \rangle = \Psi(e_1) \odot \Psi(e_2)$, where we assume w.l.o.g. $a_1 > b_2$ holds, and let $t^* \in T_{e_1}$ be the terminal such that P_{v,t^*} is the dominating path of e. Then any blocking integral multiflow $\tilde{f}$ with $\tilde{f}(e) = a$ in $\widetilde{I}_e$ satisfies the following.*

(i) *Any integer individual flow g decomposed from $\tilde{f}$ satisfies $g(P_{u,t}) = 0$ for all $t \in T_e - \{t^*\}$ (hence $g(P_{u,t^*}) = f(e)$).*

(ii) *If* $c(e) \le a$*, then the function* $f := \tilde{f} + (P_{u,t^*}, -\delta_{u,t^*})$ *with* $\delta_{u,t^*} = a - c(e)$ *is a blocking multiflow in* I_e *(hence* $\Psi(e) \subseteq \langle c(e), c(e)\rangle$ *if* $c(e) \le a$*).*

Proof. The lemma can be proven analogously with the proof for Lemma 11. Let $\widetilde{\mathcal{X}}$ be an $\tilde{f}$-blocking cut-system of T_e in $\widetilde{I}_e$, and g be an integer individual flow g decomposed from $\tilde{f}$.

(i) We first show that there is a terminal $t' \in T_e$ such that $g(P_{u,t}) = 0$, $t \in T_e - \{t'\}$. Assume indirectly that there are two terminals $t_1, t_2 \in T_e$ such that $g(P_{u,t_1}) > 0$ and $g(P_{u,t_2}) > 0$. Let w be the branch vertex of P_{u,t_1} and P_{u,t_2} (see Fig. 6(a)). Then the path $P_{u,w}$ is vertex-disjoint with any terminal-cut $X_t \in \widetilde{\mathcal{X}}$, since otherwise $u \notin X_t$ implies that one of paths P_{u,t_1} and P_{u,t_2} would contain two edges from $E(X_t)$ contradicting Lemma 15. The function $f' := \tilde{f} + (P_{u,t_1}, -1) + (P_{u,t_2}, -1) + (P_{t_1,t_2}, 1)$ is a feasible multiflow such that $f'(e') < \tilde{f}(e')$ for all edges e' along path $P_{u,w}$, and $f'(e') = \tilde{f}(e')$ for all other edges e'. Since $P_{u,w}$ is vertex-disjoint with any terminal-cut $X_t \in \widetilde{\mathcal{X}}$, we see that $\widetilde{\mathcal{X}}$ is f'-blocking and f' is a blocking multiflow with $f'(e) < a$ in $\widetilde{I}_e$. This, however, contradicts the minimality of a, proving the existence of such a special terminal $t' \in T_e$.

We next show that t' is the terminal t^* of the dominating path P_{u,t^*} of e. Note that $a \ge 2$ implies either $a_1 \ge b_2 + 2$ or $a_2 \ge b_1 + 2$. By assumption of $a_1 > b_2$, we see that $t^* \in T_{e_1}$, $\tilde{f}(e_1) = a_1$, $\tilde{f}(e_2) = b_2$ and $\tilde{f}(e) = a = a_1 - b_2 = \tilde{f}(e_1) - \tilde{f}(e_2)$. Since $\tilde{f}(e) = \tilde{f}(e_1) - \tilde{f}(e_2)$ means $g(P_{u,t}) = 0$ for any $t \in T_{e_2}$, it holds $t' \in T_{e_1}$. If e_1 is a leaf-edge, then $t^*, t' \in T_{e_1}$ implies that $t' = t^*$ holds. Assume that e_1 is not a leaf-edge, let e'_1 and e'_2 be the two child-edges of $e_1 = (v, v')$, and denote $\langle a', b'\rangle = \langle a'_1, b'_1\rangle \odot \langle a'_2, b'_2\rangle$ for $\Psi(e'_i) = \langle a'_i, b'_i\rangle$, $i = 1, 2$. Recall that $a_1 \ge 2$. Since $a_1 = \min\{a', c(e_1)\} \ge 2$ and thereby $a' \ge 2$, we see that $a'_1 \ge b'_2 + 2$ holds without loss of generality. By applying Lemma 19 to edge e_1 with $a'_1 \ge 2$, we see that t' is the terminal for the dominating path $P_{v',t'}$ of e_1. Since $P_{v',t'}$ needs to be a subpath of the dominating path P_{u,t^*}, it must hold $t' = t^*$.

(ii) Let $Y \subseteq V_e - (X_{t^*} \cup V(P_{u,t^*}))$ be the set of vertices reachable from a vertex in P_{u,t^*} via edges e' with $\tilde{f}(e') \le c(e') - 2$ or edges $e' \in E(P_{u,t^*})$ with $\tilde{f}(e') \le c(e') - 1$. We claim that any positive path P_{t_1,t_2} between terminals $t_1, t_2 \in T_e - \{t^*\}$ is vertex-disjoint with $V(P_{u,t^*}) \cup Y$. Assume indirectly that there is a positive path P_{t_1,t_2} between some terminals $t_1, t_2 \in T_e - \{t^*\}$ that contains a vertex in $V(P_{u,t^*}) \cup Y$. Let w be the branch vertex of P_{u,t_1} and P_{u,t_2}, and w' be the branch vertex of P_{u,t^*} and P_{u,t_2}, where we assume without loss of generality that the branch vertex of P_{u,t^*} and P_{u,t_1} is not a descendant of w' (see Fig. 6(b),(c)). By Lemma 15, any terminal-cut $X \in \widetilde{\mathcal{X}} - \{X_{t^*}\}$ is vertex-disjoint with P_{u,t^*}, and P_{t_1,t_2} is vertex-disjoint with X_{t^*}. By construction of Y, any terminal-cut $X \in \widetilde{\mathcal{X}} - \{X_{t^*}\}$ is disjoint with Y.

We consider the case where $V(P_{t_1,t_2}) \cap V(P_{u,t^*}) = \emptyset$ and $V(P_{t_1,t_2}) \cap Y \ne \emptyset$, as in Fig. 6(c) (the case where $V(P_{t_1,t_2}) \cap V(P_{u,t^*}) \ne \emptyset$ in Fig. 6(b) can be treated analogously). The function $f' := \tilde{f} + (P_{t_1,t_2}, -1) + (P_{u,t^*}, -1) + (P_{u,t_1}, 1) + (P_{t^*,t_2}, 1)$ is a feasible multiflow in $\widetilde{I}_e$. Then we see that $\widetilde{\mathcal{X}}$ remains f'-blocking, but f' admits a decomposition g' such that $g'(P_{u,t_1}) = g(P_{u,t_1}) + 1 > 0$, a contradiction to (i). This proves our claim.

We are ready to prove that $f = \tilde{f} + (P_{t^*,u}, -\delta_{u,t^*})$ is blocking in I_e. For this, we modify the terminal-cut $X_{t^*} \in \widetilde{\mathcal{X}}$. Let X'_t be the set of vertices reachable from t via unsaturated edges e' such that $\tilde{f}(e') \le c(e') - 2$ or $e' \in E(P_{u,t})$. Hence $X'_{t^*} = X_{t^*} \cup Y \cup V(P_{v,t^*})$. We show that $\mathcal{X} = (\widetilde{\mathcal{X}} - \{X_{t^*}\}) \cup \{X'_{t^*}\}$ is f-blocking. Since $f(e') = \tilde{f}(e')$ holds for all edges e' not in $P_{t^*,u}$, every terminal-cut $X_t \in \widetilde{\mathcal{X}}$ with $t \in T_e - \{t^*\}$ remains semi-tight in f. Since Y is spanned by unsaturated edges, X'_{t^*} is disjoint with any other terminal-cuts $X \in \widetilde{\mathcal{X}} - \{X_t\}$ and $\mathcal{X}$ is a cut-system in I_e. The remaining task is to show that terminal-cut X'_{t^*} is tight in f. By the choice of Y, each edge $e \in E(X'_t)$ is saturated by $\tilde{f}$ or the parent-edge of an odd set $W \in \mathit{odd}(X'_t)$ with $f(e) = c(e) - 1$. By Lemma 15 and the above claim, for any decomposition g of f, any positive

T_e-path P_{t_1,t_2} with $t_1, t_2 \in T_e - \{t^*\}$ is vertex-disjoint with each of X_t, P_{u,t^*} and Y. This proves that $f(X'_{t^*}) = f(e_{t^*})$. Hence X'_{t^*} is semi-tight in f, as required.

∎

Lemma 18 and Lemma 19 prove Lemma 14. Based on Lemma 14 and Lemma 19, a linear-time algorithm for finding a maximum integral flow can be designed in the similar manner with DPTreeFlow. The algorithm is described as follows.

Algorithm IntegerTreeFlow
Input: A standard tree instance $I = (G = (V, E), T, c)$ and a terminal $r \in T$ as the root of G.
Output: A blocking multiflow f^* with $f^*(e_r) := \max\{x \in \Psi(e_r)\}$ in I.

```
Let δ(e) := 0 for all edges e ∈ E;
Let Ψ(e_t) := ⟨c(e_t), c(e_t)⟩ for all leaf-edges e_t, t ∈ T − {r};
for each non-leaf-edge e ∈ E to be picked up in a nonincreasing order of their depths do
    ⟨a, b⟩ := Ψ(e_1) ⊙ Ψ(e_2) for the child-edges e_1 and e_2 of e;
    Ψ(e) := { ⟨a, b⟩ ∩ [0, c(e)]   if c(e) ≥ a
            { ⟨c(e), c(e)⟩          if c(e) < a
end; /* for */
Let f(e_r) := b for Ψ(e_r) = ⟨a, b⟩;
for each non-leaf edge e ∈ E to be picked up in a nondecreasing order of their depths do
   /* Let Ψ(e_i) = ⟨a_i, b_i⟩ for the child-edges e_i, i = 1, 2 of e */
   if f(e) ≤ a for ⟨a, b⟩ = Ψ(e_1) ⊙ Ψ(e_2) then /* f(e) = c(e) */
      δ(e) := δ(e) + a − c(e); /* a − c(e) is the δ_{u,t*} of the edge e = (u, v) in Lemma 19 */
      if b_2 < a_1 then /* e_1 is a dominating edge */
         f(e_1) := a_1; f(e_2) := b_2; δ(e_1) := δ(e)
      else /* b_1 < a_2, and e_2 is a dominating edge */
         f(e_1) := b_1; f(e_2) := a_2; δ(e_2) := δ(e)
      end /* if */
   else /* a < f(e) ∈ ⟨a, b⟩ */
      Choose integers y ∈ Ψ(e_1), z ∈ Ψ(e_2) and s ∈ [0, min{y, z}] with f(e) = y + z − 2s;
      f(e_1) := y; f(e_2) := z
   end /* if */
end; /* for */
f*(e) := f(e) − δ(e) for all edges e ∈ E.
```

Fig. 10 shows a computational process of algorithm IntegerTreeFlow. Analogously with Theorem 12, we have the next result.

Theorem 20 *A feasible integral multiflow f and a cut-system $\mathcal{X}$ with $\alpha(f) = (\gamma(\mathcal{X}) - \kappa(\mathcal{X}))/2$ in a standard instance (G, T, c) can be found in $O(n)$ time and space, where f is a maximum integral multiflow.*

Note that the maximum integral multiflow is inner-eulerian. By Theorem 4, we can also find a decomposition g of the multiflow in linear time and space such that g has at most $n - 1$ positive individual flows, which is another representation of the maximum multiterminal flow.

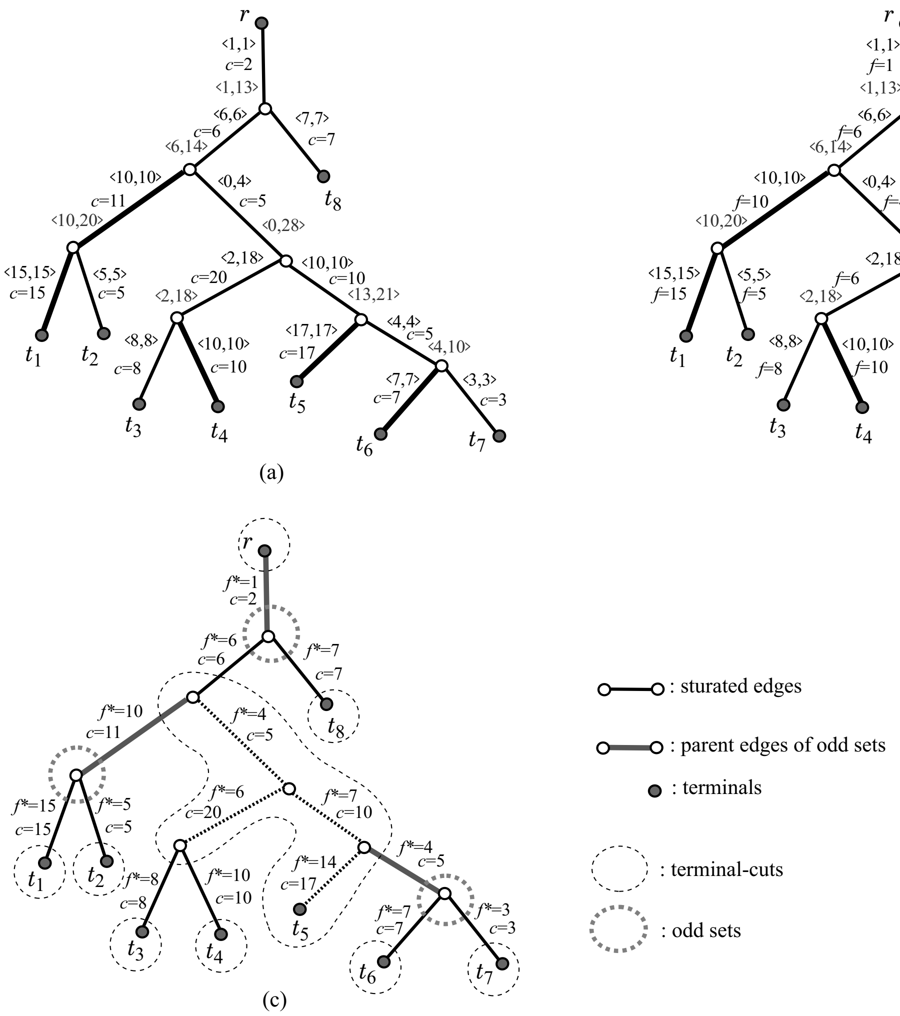

Figure 10: (a) A process of computing $\Psi(e)$, $e \in E$ in an example, where the dominating edges are drawn as thick lines, the number c beside each edge e denotes its capacity $c(e)$ and $\langle a, b\rangle = \Psi(e_1) \oplus \Psi(e_2)$ for each pair of siblings e_1 and e_2 is shown in grey: (b) a process of computing $f(e)$ and $\delta(e)$, $e \in E$, where the number f beside each edge e denotes $f(e)$ and the number δ beside each dominating edge e indicates $\delta(e)$: (c) a maximum multiflow f^* and a cut-system $\mathcal{X}$ with $\alpha(f) = (\gamma(\mathcal{X}) - \kappa(\mathcal{X}))/2$.

5 Concluding Remarks

In this paper, we gave an upper and lower bounds on the number of pairs of terminals between which a positive amount of flow is sent in a given multiflow of a multiterminal flow instance in a tree, and a characterization of multiflows as balanced functions on the edge set in the tree. The characterization provides a simple linear-time and linear-space algorithm for the maximum multiterminal flow problem in trees. To design a linear-time and linear-space algorithm for the integer version of the problem in trees, we showed a novel dynamic programming approach. It will be interesting to characterize properties and design fast algorithms for the maximum (integer) multiterminal flows in more general classes of graphs.

References

[1] Cherkasskii, B.V.: Reshenie odnoi zadachi o mnogoproduktovykh potokakh v seti [Russian; A solution of a problem of multicommodity flows in a network], Èkonomika i Matematicheskie Metody 13 (1977) 143–151

[2] Costa, M.-C., Billionnet, A.: Multiway cut and integer flow problems in trees. Electronic Notes in Discrete Mathematics **17**(20) (2004) 105–109

[3] Costa, M.-C., Létocart, L., Roupin, F.: Minimal multicut and maximal integer multiflow: a survey. European Journal of Operational Research, **162**(1) (2005) 55–69

[4] Cunningham, W.H.: The optimal multiterminal cut problem. DIMACS Series in Disc. Math. and Theor. comput. Sci. **5** (1991) 105–120

[5] Garg, N., Vazirani, V.V., Yannakakis, M.: Approximate max-flow min-(multi)cut theorems and their applications. SIAM J. Comput., **25**(2) (1996) 235–251

[6] Garg, N., Vazirani, V.V., Yannakakis, M.: Primal-Dual approximation algorithms for integral flow and multicut in trees. Algorithmica, **18**(1) (1997) 3–20

[7] Ibaraki, T., Karzanov, A. V., Nagamochi, H.: A fast algorithm for finding a maximum free multiflow in an inner eulerian network and some generalizations. Combinatorica, **18**(1) (1998) 61–83

[8] Ford, J.R. and Fulkerson, D.R.: Flows in networks, Princeton University Press, New Jersey, 1962.

[9] Hagerup, T., Katajainen, J., Nishimura, N., Ragde, P.: Characterizing Multiterminal Flow Networks and Computing Flows in Networks of Small Treewidth. J. of Computer and System Sciences, **57**(3) (1998) 366–375

[10] Mader, W.: Uber die Maximalzahl kantendisjunkter A-Wege, Archiv der Mathematik (Basel) 30 (1978) 325–336.