Mathematical Programming 36 (1986) 123-134
North-Holland

FINDING THE t-JOIN STRUCTURE OF GRAPHS

András SEBŐ
Computer and Automation Institute of the Hungarian Academy of Sciences, Budapest, Hungary

Received 27 September 1984
Revised manuscript received 19 February 1986

t-joins are generalizations of postman tours, matchings, and paths; t-cuts contain planar multicommodity flows as a special case. In this paper we present a polynomial time combinatorial algorithm that determines a minimum t-join and a maximum packing of t-cuts and that ends up with a Gallai–Edmonds type structural decomposition of (G, t) pairs, independent of the running of the algorithm. It only uses simple combinatorial steps such as the symmetric difference of two sets of edges and does not use any shrinking operations.

Key words: t-joins, t-cuts, matchings, algorithmic proof, structure theorem, Chinese postman problem.

1. Introduction

Let G be an undirected graph and $t: V(G) \to \mathbb{Z}$. $V(G)$ is the vertex set and $E(G)$ the edge-set of the graph G. ($\mathbb{Z}$ is the set of integers.) $F \subset E(G)$ is called a *t-join* if $d_F(x) \equiv t(x) \bmod 2\ \forall x \in V(G)$. ($d_F(x) := |\{e \in F$: e is adjacent to $x\}|$. Since all congruences will be "mod 2," we shall omit "mod 2" in the notation.) It is not difficult to see that G possesses a t-join if and only if $t(V(G')) \equiv 0$ for each connected component G'. (See Section 2 for the construction of a t-join. If $f: X \to \mathbb{Z}$, then $f(X) := \sum \{f(x)$: $x \in X\}$.) Suppose G is connected, $t(V(G)) \equiv 0$ and let $\tau := \tau(G, t) := \min\{|F|$: F is a t-join$\}$. If $|F| = \tau(G, t)$, then F is called a *minimum* t-join.

For $X \subset V(G)$, $G(X)$ denotes the subgraph of G induced by X, $E(X) := E(G(X))$ and $\delta(X) := \{xy \in E(G)$: $x \in X, y \notin X\}$. $\delta(X)$ is called the *coboundary* of X. $K \subset E(G)$ is a *cut* if $K = \delta(X)$ for some $X \dot\subset V(G)$. If $t(X) \equiv 1$, then X is called a *t-odd set* and $\delta(X)$ is called a *t-cut*. Obviously, $|F \cap K| \geq 1$ for any t-join F and any t-cut K.

A *k-packing* ($k \in \mathbb{Z}$, $k \geq 0$) of t-cuts is a family $\mathscr{K}$ of t-cuts with $|\{K \in \mathscr{K}$: $e \in K\}| \leq k$ for each $e \in E(G)$. Repetitions are allowed in $\mathscr{K}$. Let $\nu_k := \nu_k(G, t) := \max\{|\mathscr{K}|$: $\mathscr{K}$ is a k-packing of t-cuts$\}$. A 1-packing is simply called a packing and $\nu := \nu_1$. It is easy to see that $\tau \geq \nu_2/2 \geq \nu$. The following minimax theorems hold:

Theorem 1.1 [10]. $\tau(G, t) = \nu_2(G, t)/2$.

Theorem 1.2 [16]. *If G is bipartite, then* $\tau(G, t) = \nu(G, t)$.

These are sharpened, respectively, by the following results:

Theorem 1.3 [6]. $\tau(G, t) = \frac{1}{2}\max\{\sum_{i=1}^{k} q_t(X_i): \{X_1, \ldots, X_k\} \in \mathcal{P}(V(G))\}$ *where* $\mathcal{P}(X)$ *denotes the set of partitions of* X *and* $q_t(X)$ *for* $X \subset V(G)$ *denotes the number of t-odd components of* $G - X$.

Theorem 1.4 [6]. *If G is bipartite with classes A and B, then* $\tau(G, t) = \max\{\sum_{i=1}^{k} q_t(X_i): \{X_1, \ldots, X_k\} \in \mathcal{P}(A)\}$.

Note that Theorem 1.4 trivially implies all the previous theorems and the Berge–Tutte theorem on matchings (cf. [6]). A surprisingly short proof of Theorem 1.4 is given in [14] (cf. also [6]). A fifth minimax theorem is presented in [15], which provides the minimal TDI description (Schrijver system) of t-join polyhedra. This theorem contains Theorems 1.1 and 1.2 but seems to be independent of Theorems 1.3 and 1.4. On the other hand, Theorems 1.1–1.4 and the "Schrijver system" are implied by a "structure theorem" of t-joins proved in [12]. None of these theorems will be used in the present paper. Quite the contrary—*an algorithmic proof is provided for the "structure theorem" and hence for all of these results.*

A *path* in this paper is considered to be a set of edges. When a repetition of vertices and edges is allowed, we use the term *walk*. If the two endpoints of a path (walk) coincide, then it is called a *circuit* (*closed walk*). The *length* of a path, walk, etc., is the number of its edges (with multiplicity). The vertex set of a path P is denoted by $V(P)$. If $x, y \in V(P)$, then $P(x, y)$ is the *subpath* of P joining x and y. "$\triangle$" denotes the symmetric difference operation.

A *postman tour* is a closed walk in G that contains each edge of G at least once. It is easy to see that there is a one-to-one correspondence between minimum-length postman tours and minimum d_G-joins ($d_G := d_{E(G)}$). Further applications (e.g., matchings and ± 1-weighted paths) are summarized in [13].

The problem of finding minimum-length postman tours (the Chinese postman problem) was posed by Mei Gu Guan in [7], where an algorithm is also suggested. This algorithm proceeds by finding an arbitrary d_G-join F first and then achieving improving steps. An improving step can be achieved if and only if there exists a circuit C with $|C \setminus F| < |C \cap F|$. Such a circuit will be called an *improving circuit*. If C is an improving circuit, then $|C \triangle F| < |F|$, and since $C \triangle F$ is also a t-join, we can decrease the size of the current t-join. Conversely, if F' is a t-join, $|F'| < |F|$, then $F' \triangle F$ has all degrees even and contains an improving circuit. This idea is appealing, but Guan does not give a polynomial algorithm to find an improving circuit. As Lawler [9] remarks: "The only trouble with these observations, as Edmonds pointed out, is that it is not apparent how one should detect negative circuits in an undirected network. The ordinary shortest path computations do not apply to undirected networks in which some arcs have negative length. And any apparent process of enumeration involves a lengthy computation." (Improving circuits are negative circuits if we put weight -1 on $e \in F$ and weight $+1$ on $e \in E(G) \setminus F$.)

Polynomial algorithms for solving the Chinese postman problem have been presented in [1, 3, 4, 8]. Let us remark that each known algorithm for finding the minimum *cardinality* postman tour makes use of or is an adaptation of Edmonds's *weighted* matching algorithm and works with a linear programming framework which is quite strange for the cardinality case. Fractional (dual) solutions may occur and may be used in the course of solving the problem. The packing determined by these algorithms depends on the algorithmic execution.

In this paper we turn back to Guan's original approach by defining and using a generalization of the improving circuits. We describe a direct combinatorial algorithm that finds a minimum t-join and a maximum packing of t-cuts through elementary improving steps, in polynomial time. The algorithm ends up with a packing that does not depend on the execution of the algorithm and is actually the unique "canonical" maximum packing of t-cuts (cf. [12] and below).

Let us now say a few words about the origins of this paper. The essential step of all versions of Edmonds's matching algorithm is a certain "blossom shrinking" operation, presented in his celebrated paper [2], and the same shrinking operation occurs in the algorithms that solve generalizations of the matching problem (cf., e.g., [1, 3, 5, 8, 11]). In the present paper our starting point is a new principle presented by Lovász [11], which yields an entirely new interpretation of the matching algorithm. Lovász's version keeps in each step a list of equal size matchings at hand, and constructs a "tentative" "Gallai–Edmonds partition" corresponding to this list. If the current partition happens to be the Gallai–Edmonds partition of the graph, then the algorithm stops by concluding that each matching of the list is maximum. If it is not, then either a matching of greater cardinality is determined starting a new list, or a matching of the same cardinality is added to the list. In the latter case, the tentative partition corresponding to the new list turns out to be better in some sense.

The "structure theorem" of t-joins proved in [12] generalizes the Gallai–Edmonds theorem. It claims the existence of a unique "canonical" packing of t-cuts that characterizes the set of all minimum t-joins. It will be stated here in the form of an "optimality criterion" and will be proved algorithmically. It will play the same role as the Gallai–Edmonds theorem in Lovász's algorithm. That is, the algorithm does not rely on the theorem; the knowledge of the theorem serves merely as a motivation for the algorithm. Thus the paper is self-contained and provides an independent algorithmic proof of the structure theorem. This algorithm contains Lovász's algorithm as a special case, but the case analysis of the latter becomes considerably simpler at the level of t-joins.

The paper is organized as follows. In Section 2 we define the notions needed to describe the algorithm and state the optimality criterion that controls the algorithm.

Sections 3 and 4 present and explain the main steps. In Section 5 a brief summary of the algorithm is given together with some comments.

2. Preliminary remarks

The pair (G, t), where G is an arbitrary graph and $t: V(G) \to \mathbb{Z}$, $t(V(G)) \equiv 1$, will be called a *tower*. If (G, t) is a tower and $a \in V(G)$, set

$$t^a(x) := \begin{cases} t(x) & \text{if } x \neq a, \\ t(x)+1 & \text{if } x = a. \end{cases}$$

The input of the algorithm will be a tower (G, t), and the output will be a minimum t^x-join for all $x \in V(G)$. This output is enough to obtain a canonical packing of t^x-cuts for all $x \in V(G)$ (Section 5).

Towers enable us to generalize improving circuits. Clearly, if F^a is a t^a-join and P is an (a, b) path, then $F^a \triangle P$ is a t^b-join. If F^b is a t^b-join and $|F^a \triangle P| < |F^b|$, then we say that P is an F^b*-improving path.* If $a = b$, we get back Guan's improving circuits.

We shall often use the trivial fact that the symmetric difference of a t^a-join and a t^b-join is the disjoint union of an (a, b) path and circuits. Assignments will be denoted by "$\leftarrow$".

Before getting into the details of improving paths, we simplify our problem. We show that it is enough to deal with towers (G, t) where G is bipartite; these will be called *bipartite towers.* Let (G, t) be an arbitrary tower, and divide each edge $e \in E(G)$ into two edges with a new vertex v_e. Denote the result by G', and define

$$t'(x) := \begin{cases} t(x) & \text{if } x \in V(G), \\ 0 & \text{if } x = v_e,\ e \in E(G). \end{cases}$$

(G', t') is a bipartite tower. It is straightforward to see that the natural one-to-one correspondence between t^x-joins of G and t'^x-joins of G', and between 2-packings of t^x-cuts of G and packings of t'^x-cuts of G', doubles the cardinality of t^x-joins and preserves the cardinality of packings of t^x-cuts. So it preserves optimality, and we may therefore suppose in the following that (G, t) is a bipartite tower. (It might, of course, be useful in practice to work directly on the nonbipartite tower without doubling the edges. The corresponding algorithm can be deduced from the bipartite case.) For bipartite towers some statements become sharper (compare e.g., Theorem 1.2 with Theorem 1.1, and Theorem 1.4 with Theorem 1.3), and many technical details become simpler to describe.

The algorithm starts by determining an arbitrary t^x-join F^x, for each $x \in V(G)$. It is enough to consider one $x \in V(G)$ since we can get F^y from $F^y \leftarrow F^x \triangle P$, where P is an (x, y) path. It is easy to construct a t^x-join: If $F \subset E(G)$ and $a \neq b \in V(G)$, $d_F(a) \not\equiv t^x(a)$, $d_F(b) \not\equiv t^x(b)$, then $F \leftarrow F \triangle P$ where P is an (a, b) path increases the number of vertices $v \in V(G)$ with $d_F(v) \equiv t^x(v)$. Starting from $F = \emptyset$ and repeating this step, we get a t^x-join. Let $\pi(x) := |F^x|$. In each step of the algorithm $|F^x|$

will be decreased for some $x \in V(G)$, until for each $x \in V(G)$, F^x is a minimum t^x-join. When F^x changes, $\pi(x)$ must be changed accordingly.

Now we return to the improving paths. The following two propositions are immediate consequences of their definition:

Proposition 2.1. *If $ab \in E(G)$ and $\pi(b) < \pi(a) - 1$, then the edge ab is an F^a-improving path.*

Proposition 2.2. *Let $P \subset F^a \triangle F^b$ be an (a, b) path and $p \in V(P)$. $P(a, p)$ is an F^p-improving path if and only if*

$$|F^a \cap P(a, p)| - |F^b \cap P(a, p)| > \pi(a) - \pi(p).$$

Let us perform inproving steps with improving paths of length 1, while the condition of Proposition 2.1 holds for some $ab \in E(G)$. If it does not, then $|\pi(x) - \pi(y)| \leq 1$ for all $xy \in E(G)$. Since G is bipartite, $xy \in E(G)$ implies $\pi(x) \neq \pi(y)$. ($F^x \triangle F^y$ is the disjoint union of an (x, y) path and circuits, but the path is odd because $xy \in E(G)$, and the circuits are even. Consequently, $|F^x|$ and $|F^y|$ have different parity.) If the condition of Proposition 2.1 does not hold, then

$$|\pi(y) - \pi(x)| = 1 \quad \text{for all } xy \in E(G). \tag{2.1}$$

When π is decreased in any way during the algorithm, (2.1) can be restored by improving with $ab \in E(G)$ if necessary (see Proposition 2.1). Similarly, whenever we deal with a path $P \subset F^x \triangle F^y$, we can check for each vertex $p \in V(P)$ whether $P(x, p)$ or $P(y, p)$ is an F^p-improving path or not. If it is, we improve. If $P(x, p)$ is not F^p-improving, then according to Proposition 2.2 we have:

$$|F^x \cap P(x, p)| - |F^y \cap P(x, p)| \leq \pi(x) - \pi(p). \tag{2.2}$$

In particular, if P itself is neither F^x-improving nor F^y-improving, then

$$|F^x \cap P| - |F^y \cap P| = \pi(x) - \pi(y), \quad \text{i.e., } |F^x| = |F^y \triangle P|, \quad \text{and}$$
$$|F^y| = |F^x \triangle P|. \tag{2.3}$$

Each step of the algorithm will work towards finding an improving path. If any of (2.1), (2.2), or (2.3) does not hold, this goal is reached at once. Thus we can always assume that they do hold.

We now introduce the optimality criterion. Given a function $\pi : V(G) \to \mathbb{Z}$ let us introduce the following notations: $m := m(\pi) := \min\{\pi(x) : x \in V(G)\}$, $M := M(\pi) := \max\{\pi(x) : x \in V(G)\}$, $G^i := G^i(\pi) := G(\{x \in V(G) : \pi(x) \leq i\})$ $(m \leq i \leq M)$, $\mathscr{D} := \mathscr{D}(\pi) := \{D : D$ is the vertex set of a component of G^i for some $i\}$. This set-system $\mathscr{D}(\pi)$ will always be at hand. It must be reconstructed each time π is changed. It will play a crucial role. If (2.1) is satisfied, then $\{\delta(D) : D \in \mathscr{D}\}$ *is a partition* of $E(G)$. When the algorithm stops, then for all $x \in V(G)$ $\pi(x)$ is the cardinality of a minimum t^x-join, and $\{\delta(D) : D \in \mathscr{D}\}$ will turn out to be a "canonical partition" of $E(G)$with respect to t (Section 5).

We are able now to state the optimality criterion:

Optimality Criterion. *Let (G, t) be a bipartite tower, and suppose a t^x-join F^x is given for all $x \in V(G)$. Set $\pi(x) := |F^x|$. Then the following statements are equivalent:*

(i) *F^x is a minimum t^x-join for all $x \in V(G)$.*

(ii) *(2.1) holds, and there exists $a \in V(G)$ such that:*

a. *$F^a \cap \delta(D) = \emptyset$, provided $a \in D \in \mathscr{D}$.*

b. *$|F^a \cap \delta(D)| \leqslant 1$, provided $a \notin D \in \mathscr{D}$.*

(iii) *(2.1) holds, and for all $x \in V(G)$ and $D \in \mathscr{D}$,*

a. *$F^x \cap \delta(D) = \emptyset$, provided $x \in D$.*

b. *$|F^x \cap \delta(D)| = 1$, provided $x \notin D$.*

Proof. By Proposition 2.1, (i) implies (2.1). The rest of "(i)$\Rightarrow$(ii)" will follow from the algorithm; in Sections 3 and 4 we shall show how some $\pi(x)$ can be decreased if (ii)a or (ii)b, respectively, does not hold for some a and D. This proves the seemingly stronger statement that (ii)a and (ii)b hold for all $a \in V(G)$ and $D \in \mathscr{D}$.

To prove "(ii)$\Rightarrow$(iii)" let $a \in V(G)$ satisfy (ii)a and (ii)b, and let $x \in V(G)$ be arbitrary. $F^x \triangle F^a$ is the edge-disjoint union of an (x, a) path P, and circuits $C_1, \ldots, C_k$. Since $|C_i \cap \delta(D)| = |C_i \cap F^x \cap \delta(D)| + |C_i \cap F^a \cap \delta(D)|$ is even and $|C_i \cap F^a \cap \delta(D)| \leqslant |F^a \cap \delta(D)| \leqslant 1$, we have that, for all $D \in \mathscr{D}$,

$$|C_i \cap F^x \cap \delta(D)| \geqslant |C_i \cap F^a \cap \delta(D)| \quad (i = 1, \ldots, k). \tag{2.4}$$

Similarly, if $D \in \mathscr{D}$ is such that $\{a, x\} \cap D = \emptyset$ or $\{a, x\} \subset D$, then

$$|P \cap F^x \cap \delta(D)| \geqslant |P \cap F^a \cap \delta(D)|. \tag{2.5}$$

On the other hand, if $a \in D$, $x \notin D$, then

$$|P \cap F^x \cap \delta(D)| - |P \cap F^a \cap \delta(D)| \geqslant 1 \tag{2.6}$$

because $|P \cap \delta(D)|$ is odd, and $F^a \cap \delta(D) = \emptyset$. If $a \notin D$, $x \in D$

$$|P \cap F^x \cap \delta(D)| - |P \cap F^a \cap \delta(D)| \geqslant -1 \tag{2.7}$$

directly by (ii). Using the fact that $\{\delta(D) \colon D \in \mathscr{D}\}$ is a partition,

$$\begin{aligned}\pi(x) - \pi(a) &= |F^x \setminus F^a| - |F^a \setminus F^x| = |P \cap F^x| - |P \cap F^a| \\ &\quad + \sum_{i=1}^{k} |C_i \cap F^x| - |C_i \cap F^a| \\ &= \sum_{D \in \mathscr{D}} \{|P \cap F^x \cap \delta(D)| - |P \cap F^a \cap \delta(D)| \\ &\quad + \sum_{i=1}^{k} (|C_i \cap F^x \cap \delta(D)| - |C_i \cap F^a \cap \delta(D)|)\}.\end{aligned}$$

So, applying (2.4), (2.5) first and (2.6), (2.7) thereafter:

$$\begin{aligned}\pi(x)-\pi(a) &\geq \sum\{|P\cap F^x\cap\delta(D)| \\ &\qquad -|P\cap F^a\cap\delta(D)|\colon D\in\mathcal{D}, |\{a,x\}\cap D|=1\} \\ &\geq |\{D\in\mathcal{D}\colon a\in D, x\notin D\}|-|\{D\in\mathcal{D}\colon a\notin D, x\in D\}| \\ &= |\{D\in\mathcal{D}\colon a\in D\}|-|\{D\in\mathcal{D}\colon x\in D\}| \\ &= M-\pi(a)+1-(M-\pi(x)+1)=\pi(x)-\pi(a).\end{aligned}$$

Consequently, equality must hold throughout. Let $D\in\mathcal{D}$ be such that $a\notin D$ and choose $x\in D$. Equality in (2.7) implies that $|F^a\cap\delta(D)|=1$. Now, for arbitrary $x\in V(G)$ and $D\in\mathcal{D}$, equalities in (2.4)-(2.7) imply that $|F^x\cap\delta(D)|=0$ when $x\in D$ and $|F^x\cap\delta(D)|=1$ when $x\notin D$. Thus (ii)$\Rightarrow$(iii) is proved.

Finally, let us prove (iii)$\Rightarrow$(i). Let $x\in V(G)$ be arbitrary, and assume that (iii) holds. Then $|F^x|=\sum_{D\in\mathcal{D}}|F^x\cap\delta(D)|=|\{D\in\mathcal{D}\colon x\notin D\}|$. (iii) implies that $\delta(D)$ is a t^x-cut provided $x\notin D$. Thus we have a packing of t^x-cuts with cardinality $|F^x|$. $\square$

Note that the proof of the essential part "(i)$\Rightarrow$(ii)" is postponed.

The statement "(i)$\Rightarrow$(iii)" is trivially equivalent to the main result of [12] (cf. [12, Theorem 2.6]) and is a version of what we call a "structure theorem" of t-joins.

In the following we shall only use systems $\mathcal{D}(\pi)$ with π satisfying (2.1). Our view of such systems should include the following:

Proposition 2.3. *If* $D\in\mathcal{D}$, *and* $cd\in\delta(D)$, $c\notin D$, $d\in D$ *then* $\pi(d)=\max\{\pi(x)\colon x\in D\}=\pi(c)-1$.

Proof. By definition, D is a component of the graph G^i for some i, $m\leq i\leq M$. $\pi(c)>i$ and $\pi(d)\leq i$ follows. (2.1) implies now $\pi(c)=i+1$, $\pi(d)=i=\max\{\pi(x)\colon x\in D\}$. $\square$

3. The bubble step

In this section we show how to find an improving path if (ii)a of the optimality criterion is not satisfied for some a and D, i.e.,

$$a\in D\in\mathcal{D}(\pi)\quad\text{and}\quad F^a\cap\delta(D)\neq\emptyset. \tag{3.1}$$

We shall find an improving path.

Let $cd\in F^a\cap\delta(D)$, $c\in V(G)\backslash D$, $d\in D$. By Proposition 2.3 $\pi(c)=\pi(d)+1$. Let $Q\subset E(D)$ be a minimum (in edge cardinality) (a,d) path in $G(D)$.

Case 1. $|Q|=0$, i.e., $a=d$. In this case the edge cd is clearly an F^c-improving path. ($F^c\leftarrow F^d\backslash\{cd\}$ decreases $\pi(c)$ by 2.)

Case 2. $|Q|>0$. Denote by b the neighbor of a on Q. (ab is the first edge of Q.) (2.1) implies that either Case 2.1 or Case 2.2 holds:

Case 2.1. $\pi(b)=\pi(a)+1$. If $ab\in F^a$, then the edge ab is an improving path. $F^b\leftarrow F^a\backslash\{ab\}$. If $ab\notin F^a$, then assign $F^b\leftarrow F^a\cup\{ab\}$. $\pi(b)$ does not change. This implies that $\mathscr{D} = \mathscr{D}(\pi)$ is unchanged. (3.1) also holds now with b instead of a, and $|Q(b, d)| < |Q|$. Repeat, the algorithm of this section, with b in the place of a. (After $|Q|$ repetitions of Case 2, Case 1 holds.)

Case 2.2. $\pi(b) = \pi(a) - 1$. If $ab \in F^a$, then $F^b \leftarrow F^a \setminus \{ab\}$. Repeat the algorithm of this section with b instead of a. ($\pi(b)$ does not change, (3.1) still holds, and $|Q|$ decreases.) If $ab \notin F^a$, then consider an (a, b) path $P \subset F^a \triangle F^b$. By (2.3),

$$|F^a \cap P| - |F^b \cap P| = 1 (= \pi(a) - \pi(b)), \text{ i.e., } |F^a \triangle P| = |F^b|. \tag{3.2}$$

If $V(P) \not\subset D$, then we can immediately construct an improving path: Let $pr \in P$, $p \in V(P) \setminus D$, $r \in D$. By Proposition 2.3 $\pi(p) > \pi(r) \geq \pi(a)$. Let $C := P \cup \{ab\}$. As $ab \notin F^a$, (3.2) can be written in the form $|C \cap F^a| - |C \setminus F^a| = 0$. It follows that, for one of the two (a, p) subpaths $C(a, p) \subset C$ of the circuit C,

$$|C(a, p) \setminus F^a| - |C(a, p) \cap F^a| \leq 0 < \pi(p) - \pi(a)$$

holds. Hence $|F^a \triangle C(a, p)| < |F^p|$, i.e., $C(a, p)$ is an F^p-improving path.

Thus, $V(P) \subset D$, i.e., $P \cap \delta(D) = \emptyset$ can be assumed. $cd \notin P$ follows. After $F^b \leftarrow F^a \triangle P$, $\pi(b)$ does not change (see Case 2.2), and (3.1) holds for b. ($cd \in F^b \cap \delta(D)$, $|Q|$ is decreased again.) Repeat the algorithm of this section with b instead of a.

Vertex a in (3.1) can be visualized to be a "bubble" in the "water" D, trying to come up to the surface of D. Sometimes it goes up (Case 2.1), and sometimes down, which is more difficult (Case 2.2). Either it disappears before reaching the surface (an improving path occurs before Case 1), or it disappears only when reaching the surface at the "wrong" edge cd (Case 1).

The procedure described in this section will be referred to as the "bubble step." The algorithm is meant to repeat the bubble step until (ii)a holds, i.e., while at least one bubble exists.

4. The straw step

In this section we show how to find an improving path if in the optimality criterion (ii)a is satisfied for all $a \in V(G)$, but (ii)b is not. Note that this will complete the proof of (i)$\Rightarrow$(ii) in the optimality criterion.

Let $a \in V(G)$, and let $D \in \mathscr{D}$ be such that $a \notin D$ and $|F^a \cap \delta(D)| \geq 2$. Let $cd \in F^a \cap \delta(D)$, $c \notin D$, $d \in D$. We shall use the fact that $F^a \cap \delta(D)$ contains another edge as well.

Consider the (a, d) path $P \subset F^a \triangle F^d$, and let p be the first vertex of P in D. Let us apply (2.2):

$$|F^a \cap P(a, p)| - |F^d \cap P(a, p)| \leq \pi(a) - \pi(p). \tag{4.1}$$

Case 1. Equality is satisfied in (4.1). In this case $F^p \leftarrow F^a \triangle P(a, p)$ is a t^p-join, and $\pi(p)$ is not changed. Consequently, $\mathscr{D}(\pi)$ remains the same. Since p is the first vertex of P in D, $|P(a, p) \cap \delta(D)| = 1$; because of $|F^a \cap \delta(D)| \geq 2$ we have $|(F^a \triangle P(a, p)) \cap \delta(D)| \geq 1$. Thus, after the assignment above, p is a bubble in D(p satisfies (3.1)). A bubble step is to be executed now.

Case 2. Equality is not satisfied in (4.1). $p \neq d$ follows, since otherwise we would have equality by (2.3). Inequality in (4.1) means (using the trivial fact $F^a \cap P(a, p) = P(a, p) \setminus F^d$):

$$|P(a, p) \setminus F^d| - |P(a, p) \cap F^d| < \pi(a) - \pi(p). \tag{4.2}$$

Consider a (d, p) path $Q \subset F^d \triangle F^p$. As (ii)a is satisfied, $F^d \cap \delta(D) = \emptyset = F^p \cap \delta(D)$. It follows that $Q \cap \delta(D) = \emptyset$, and since Q is a (d, p) path, $d, p \in D$ we have $V(Q) \subset D$. On the other hand, p is the first vertex of P in D; hence $V(P(a, p)) \cap V(Q) = \{p\}$, and $R := P(a, p) \cup Q$ is an (a, d) path. By (2.3):

$$|Q \setminus F^d| - |Q \cap F^d| = \pi(p) - \pi(d). \tag{4.3}$$

Adding (4.2) and (4.3) we get $|R \setminus F^d| - |R \cap F^d| < \pi(a) - \pi(d)$. This means that R is an F^a-improving path, as we desired.

The path $P(a, p)$ can be visualized to be a "straw" through which either a bubble is blown into D (Case 1), or the "wrong" edge cd is "sucked away" (Case 2).

The straw step has to be repeated while (ii)a holds, but (ii)b does not hold.

5. Summary and comments

First we summarize the algorithm. We shall also indicate improvements arising from Propositions 2.1 and 2.2. ((2.1), (2.2) and (2.3) will not be assumed to hold automatically.)

Main algorithm

Input: A bipartite tower (G, t).

Output: A minimum t^x-join for each $x \in V(G)$, and a system $\mathcal{D}$.

0. For all $x \in V(G)$ determine a t^x-join F^x. **go to** 1.
1. While there exists $e \in E(G)$ which is an improving path, improve. Afterwards (2.1) holds (Proposition 2.1). Determine the system $\mathcal{D} \leftarrow \mathcal{D}(\pi)$, $\pi(x) \leftarrow |F^x|$ (for all $x \in V(G)$).

   If (ii) of the optimality criterion is true, **stop**. Otherwise, one of the following two conditions holds:

   If there exists $a \in D \in \mathcal{D}$, $cd \in F^a \cap \delta(D)$ $(c \notin D, d \in D)$, then **go to** 2.

   If $x \in D \in \mathcal{D}$ implies $F^x \cap \delta(D) = \emptyset$ but there exists a, D, $a \notin D \in \mathcal{D}$, $cd \neq c'd' \in F^a \cap \delta(D)$ $(c, c' \notin D;\ d, d' \in D)$, then **go to** 3.
2. Determine an (a, d) path $Q \subset D$ and call the subroutine BUBBLE(a, Q, cd). Then **go to** 1.
3. Determine an (a, d) path $P \subset F^a \triangle F^d$. Check whether there exists $p \in V(P)$ such that $P(a, p)$ or $P(d, p)$ is an F^p-improving path.

   If yes, then improve and **go to** 1.

   If no, then (2.2) and (2.3) hold with a, d in the role of x, y (Proposition 2.2). Call the subroutine STRAW$(a, P, cd, c'd')$; then **go to** 1.

end

The summary of the subroutines follows now. We do not repeat the conditions that must be satisfied by their parameters. Everything true in the calling environment is supposed to hold true.

BUBBLE(a, Q, cd)

1. If $|Q|=0$, then the edge cd is an improving path.
   Improve and **stop**.
   Otherwise, **go to** 2.
2. If $|Q|>0$, then let $ab \in Q$.
   If $\pi(b)=\pi(a)+1$, then **go to** 2.1.
   If $\pi(b)=\pi(a)-1$, then **go to** 2.2.

2.1. $F^b \leftarrow F^a \cup \{ab\}$ and call BUBBLE(b, $Q(b, d)$, cd). Then **stop**. ($ab \notin F^a$, since $ab \in F^a$ would imply that ab is an improving path, contradicting 1 of the main algorithm.)

2.2. If $ab \in F^a$, then $F^b \leftarrow F^a \backslash \{ab\}$. Call BUBBLE($b$, $Q(b, d)$, cd). **stop**. If $ab \notin F^a$, then consider an (a, b) path $P \subset F^a \triangle F^b$. $C \leftarrow P \cup \{ab\}$. Check whether there exists $p \in V(P)$ such that either of the two distinct (a, p) subpaths of C is improving.
   If yes, then improve and **stop**.
   If no, then $F^b \leftarrow F^a \triangle P$. Call BUBBLE($b$, $Q(b, d)$, cd). **stop**.

end

Clearly, the subroutine BUBBLE finds an improving path after at most $|Q|$ recursive calls.

STRAW(a, P, cd, $c'd'$)

1. Let $p \in V(P)$ be the first vertex of P in D, starting from a. One of the following alternatives holds. (If $P(a, p)$ were an improving path, we would have improved in 3 of the main algorithm.)
   If $|F^a \triangle P(a, p)| = F^p$, then $F^p \leftarrow F^a \triangle P(a, p)$. Call BUBBLE($p$, Q, cd (or $c'd'$)), where Q is a (p, d) path (or (p, d') path), $Q \subset D$. Then **stop**. (If the last edge of $P(a, p)$ is cd, $d = p$, then after the above assignment $cd \notin F^p \cap \delta(D)$, and we have to call BUBBLE with $c'd'$.) Otherwise **go to** 2.
2. If $|F^a \triangle P(a, p)| > |F^p|$, then determine the (d, p) path $Q \subset F^d \triangle F^p$. It turns out that $P(a, p)$ and Q only have their endpoint p in common, and $P(a, p) \cup Q$ is an F^a-improving (d, a) path. Improve and **stop**.

end

Let us now estimate the running time of the algorithm. If a t^x-join contains a circuit, then deleting this circuit we obtain a t^x-join with smaller cardinality. Thus F^x $(x \in V(G))$ can be assumed to be a forest in the beginning, and consequently $0 \leqslant \pi(x) \leqslant n-1$. Thus $0 \leqslant \sum_{x \in V(G)} \pi(x) \leqslant n^2$. It follows that the number of improving steps is at most n^2. It is easy to see that each subroutine finds an improving path in $O(n^2)$ time. Thus the algorithm has $O(n^4)$ worst-case running time.

Clearly, when the algorithm stops, the system $\mathscr{D}$ is identical to $\mathscr{D}(\pi)$, where $\pi(x) := \tau(G, t^x)$. It follows that the output $\mathscr{D}$ does not depend on the algorithm, only on (G, t). Luckily enough, (iii)b of the optimality criterion implies that $\{\delta(D)\colon D \in \mathscr{D}, x \notin D\}$ is a maximum packing of t^x-cuts. (See the proof of (iii)$\Rightarrow$(i) of the optimality criterion.) Theorem 1.2 immediately follows, and with little work the tighter Theorems 1.3 and 1.4 can also be proved. (To prove Theorem 1.4 choose $\mathscr{P} := \{\{x \in D\colon \pi(x) = \max_{y \in D} \pi(y)\} \cap A\colon D \in \mathscr{D}\}$.) Note that this maximum packing of t^x-cuts is also defined with (G, t) only.

We have thus made a particular system $\mathscr{D}(G, t)$ correspond to each tower (G, t). We call $\{D \in \mathscr{D}(G, t)\colon x \notin D\}$ a *canonical packing* of t^x-cuts in the tower (G, t). In [12] and [13] properties characterizing $\mathscr{D}(G, t)$ are mentioned. One of these properties is that the set of minimum t^x-joins can be described in terms of the system $\mathscr{D}(G, t)$. The essential part of this description is that (iii) of the optimality criterion holds.

The algorithm and the system $\mathscr{D}(G, t)$ are also new in the special case of postman tours and minimum weight paths in graphs without negative circuits (cf. [12]). Specializing them to matchings (see the reduction in [6, 12, or 13]), the bubble step becomes a step of Lovász's algorithm, and the system $\mathscr{D}(G, t)$ gives the Gallai–Edmonds decomposition of graphs. Thus it does not give a new algorithm in this case, although it provides some simplifications.

Further applications are mentioned in [13] with more details.

Acknowledgment

I wish to express my thanks to András Frank, who encouraged me to do this work. He and Éva Tardos helped me a lot with their useful remarks. I am grateful to the anonymous referees and J. Gyenese for improving the clarity and accuracy of the presentation.

References

[1] F. Barahona, R. Maynard, R. Rammal and J.P. Uhry, "Morphology of ground states of two-dimensional frustration model," *Journal of Physics A, Mathematical and General* 15 (1982) 673–679.

[2] J. Edmonds, "Paths, trees and flowers," *Canadian Journal of Mathematics* 17 (1965) 449–467.

[3] J. Edmonds and E. Johnson, "Matching: a well solved class of integer linear programs," in: R. Guy et al., eds., *Combinatorial Structures and Their Applications* (Gordon and Breach, New York, 1976) pp. 89–92.

[4] J. Edmonds, "The Chinese postman problem," *Operations Research* 13 Suppl. 1 (1965) 373.

[5] J. Edmonds and E.L. Johnson, "Matching, Euler tours and the Chinese postman," *Mathematical Programming* 5 (1973) 88–124.

[6] A. Frank, A. Sebő and É. Tardos, "Covering directed and odd cuts," *Mathematical Programming Study* 22 (1984) 99–112.
[7] Mei Gu Guan, "Graphic programming using odd or even points," *Chinese Mathematics* 1 (1962) 273–277.
[8] E. Korach, "Packings of T-cuts and other aspects of dual integrality," Ph.D. thesis, Waterloo University (1982).
[9] E. Lawler, *Combinatorial Optimization: Networks and Matroids* (Holt, Rinehart, and Winston, New York, 1976).
[10] L. Lovász, "2-matchings and 2-covers of hypergraphs," *Acta Mathematica, Academiae Scientiarum Hungaricae*, 26 (1975) 433–444.
[11] L. Lovász and M.D. Plummer, *Matching Theory* (Akadémiai Kiadó, to appear).
[12] A. Sebő, "Undirected distances and the structure of odd joins," *Journal of Combinatorial Theory*, to appear.
[13] A. Sebő, "The Chinese postman problem: Algorithms, structure and applications," Working Paper MO/61 MTA SZTAKI (1985) *Discrete Applied Mathematics*, to appear.
[14] A. Sebő, "A quick proof of Seymour's theorem on t-joins," Technical Report 85383-OR, Institut für Ökonometrie und Operations Research, University of Bonn (1985), to appear in *Discrete Mathematics*.
[15] A. Sebő, "The Schrijver system of odd join polyhedra," Technical Report 85394-OR, Institut für Ökonometrie und Operations Research, University of Bonn (1985), to appear in *Combinatorica*.
[16] P.D. Seymour, "On odd cuts and plane multicommodity flows," *Proceedings of the London Mathematical Society* 3, 42 (1981) 178–192.