UNIVERSITY
OF TRENTO

DEPARTMENT OF INFORMATION AND COMMUNICATION TECHNOLOGY

38050 Povo – Trento (Italy), Via Sommarive 14
http://www.dit.unitn.it

SHORTEST PATHS IN CONSERVATIVE GRAPHS

Michele Conforti and Romeo Rizzi

May 2000

Technical Report # DIT-02-0053

Also: in *Discrete Mathematics* 226 (2001) 143-153.

Shortest Paths in Conservative Graphs

Michele Conforti, Romeo Rizzi
Dipartimento di Matematica Pura ed Applicata
Università di Padova
Via Belzoni 7, 35131 Padova, Italy.

May 15, 2000

Abstract

We give a polynomial algorithm to compute shortest paths in weighted undirected graphs with no negative cycles (conservative graphs). We show that our procedure gives a simple algorithm to compute optimal T-joins (and consequently all of their special cases, including weighted matchings). We finally give a direct algorithmic proof for arbitrary weights of a theorem of Sebő characterizing conservative graphs and optimal paths.

1 Conservative Graphs and T-joins

We propose an elementary and direct algorithm to find a shortest path between two nodes of an undirected graph with no cycle of negative weight. This shortest path problem can be formulated as an optimal degree-constrained subgraph problem and can therefore be solved by matching techniques, see e.g. [4]. Indeed, efficient algorithms for the minimum T-join problem (we define this concept later) had already been given by Edmonds and Johnson in [1].

Our main result is a purely combinatorial algorithm that finds a minimum weight T-join, giving a strongly polynomial algorithmic proof of a theorem of Sebö [9]. This theorem characterizes undirected graphs with no cycles of negative weight, in terms of potentials. An algorithmic proof of this theorem is given in [7], but that works only for unit weights. In Section 3, we provide an improved algorithm for unit weights, which is extended in Section 4 to a strongly polynomial algorithm for arbitrary rational weights.

We consider pairs (G, w) made up by an undirected multigraph $G = (V, E)$ with n nodes and m edges, together with a weight function $w = w(e)$, $e \in E$. G may have loops or parallel edges. The weight $w(F)$ of a set F of edges is $\sum_{e \in F} w(e)$. For $F \subseteq E$, let w_F be defined as $w_F(e) = -w(e)$ when $e \in F$ and $w_F(e) = w(e)$ when $e \in E \backslash F$. It is immediate to see that for $A, B \subseteq E$, we have that $w_A(B) = w(A \Delta B) - w(A)$.

Let T be an even subset of V. A *T-join* is a set of edges $J \subseteq E$ such that $d_J(v)$ is odd if and only if $v \in T$, where $d_J(v)$ is the degree of node v in the graph (V, J). An *Eulerian subgraph* is an $\emptyset$-join. (Hence, an empty set of edges is an Eulerian subgraph). A T-join of minimum weight is said *w-optimal* (or *optimal*, when no confusion arises). Finally, (G, w) is *conservative* if it contains no cycle whose weight is negative (*negative cycle*). Mei Gu Guan [3] has given the following *coNP*-characterization of T-joins in terms of conservative graphs. (Note that this is not a *good characterization* in the sense of [5]).

Theorem 1.1 *A T-join J is optimal in (G, w) if and only if (G, w_J) is conservative.*

This follows by noting that the symmetric difference of two T-joins is an Eulerian subgraph, hence the union of a (possibly empty) set of disjoint cycles, and conversely the symmetric difference of a T-join and a cycle is again a T-join.

2 Clean and Switch

In (G, w), fix a node $v_o \in V$. Assume given a $\{v_o, v\}$-join $J_{v_o,v}$ for every $v \in V \backslash \{v_o\}$ plus the $\emptyset$-join $J_{v_o,v_o} = \emptyset$. Then $\mathcal{J}_{v_o} := \{J_{v_o,v} : v \in V\}$ is a *family of joins rooted at v_o*. A family $\mathcal{J}_{v_o}$ is *w-optimal* if every $\{v_o, v\}$-join in $\mathcal{J}_{v_o}$ is w-optimal. Finally, $\mathcal{J}_{v_o}$ is a *clean* family if every join in $\mathcal{J}_{v_o}$ is acyclic. We now introduce two procedures, *Clean* and *Switch*.

Procedure *Clean* takes as input:

- A pair (G, w) and a family $\mathcal{J}_{v_o} = \{J_{v_o,v} : v \in V\}$.

Clean examines the joins in $\mathcal{J}_{v_o}$ one by one. Every join $J_{v_o,v}$ in $\mathcal{J}_{v_o}$ is decomposed as the disjoint union of an acyclic $\{v_o, v\}$-join $J'_{v_o,v}$ and into a (possibly empty) set of cycles $C_1, \ldots, C_k$.

If $w(C_i) < 0$ for some cycle C_i then *Clean* outputs C_i in (i) below.

Else $w(C_i) \geq 0$ for $1 \leq i \leq k$ and if $w(C_j) > 0$ for a cycle C_j in the set, then $w(J'_{v_o,v}) < w(J_{v_o,v})$ and *Clean* outputs $J'_{v_o,v}$ in (ii) below.

Otherwise $w(J'_{v_o,v}) = w(J_{v_o,v})$. If this is the case for all joins in $\mathcal{J}_{v_o}$, *Clean* outputs the clean family $\mathcal{J}'_{v_o} = \{J'_{v_o,v} \ : v \in V\}$ in (iii) below.
So the output is one of the following:

(i) A cycle C such that $w(C) < 0$.

(ii) An acyclic $\{v_o, v\}$-join $J'_{v_o,v}$ such that $w(J'_{v_o,v}) < w(J_{v_o,v})$.

(iii) A clean family $\mathcal{J}'_{v_o} = \{J'_{v_o,v} \ : v \in V\}$ with $w(J'_{v_o,v}) = w(J_{v_o,v}) \ \forall v \in V$.

Procedure *Switch* takes as input:

- A pair (G, w), a family $\mathcal{J}_{v_o} = \{J_{v_o,v} \ : v \in V\}$ and a node $v'_o \in V \backslash \{v_o\}$.

The output is the following:

- The pair $(G, w_{J_{v_o,v'_o}})$ and the family $\mathcal{J}'_{v'_o} = \{J'_{v'_o,v} = J_{v_o,v} \Delta J_{v_o,v'_o} \ \forall v \in V\}$ rooted at v'_o.

Theorem 2.1 *Let* $(G, w_{J_{v_o,v'_o}}, \mathcal{J}'_{v'_o})$ *be the output of* Switch *when applied to* $(G, w, \mathcal{J}_{v_o}, v'_o)$. *Then* (G, w) *is conservative and* $\mathcal{J}_{v_o}$ *is* w*-optimal if and only if* $(G, w_{J_{v_o,v'_o}})$ *is conservative and* $\mathcal{J}'_{v'_o}$ *is* $w_{J_{v_o,v'_o}}$*-optimal.*

Proof: Assume (G, w) is conservative and $\mathcal{J}_{v_o}$ is w-optimal. By Theorem 1.1, $(G, w_{J_{v_o,v'_o}})$ is conservative since $J_{v_o,v'_o} \in \mathcal{J}_{v_o}$ is w-optimal. Moreover, for every $J'_{v'_o,v} \in \mathcal{J}'_{v'_o}$ the pair $(G, (w_{J_{v_o,v'_o}})_{J'_{v'_o,v}}) = (G, w_{J_{v_o,v}})$ is conservative since $J_{v_o,v} \in \mathcal{J}_{v_o}$. So $\mathcal{J}'_{v'_o}$ is $w_{J_{v_o,v'_o}}$-optimal. Conversely, when *Switch* is applied to $(G, w_{J_{v_o,v'_o}}, \mathcal{J}'_{v'_o}, v_o)$, then the output is $(G, w, \mathcal{J}_{v_o})$. (Indeed, $J'_{v'_o,v_o} = J_{v_o,v_o} \Delta J_{v_o,v'_o} = J_{v_o,v'_o}$ for $J_{v_o,v_o} = \emptyset$). □

3 Unit pairs, bipartite pairs

If $w : E \mapsto \{-1, +1\}$ then (G, w) is a *unit* pair and we denote by E_+ the set of *positive* edges of weight $+1$ and by $E_- = E \backslash E_+$ the set of *negative* edges. We say (G, w) is a *bipartite pair* if every cycle of G has even weight. Note that a unit pair (G, w) is bipartite if and only if G is a bipartite loopless graph. In this case, (G, w) is a *bipartite unit* pair.

This section describes *Improve*, a polynomial algorithm which takes as input:

- A bipartite unit pair (G, w) and a clean family $\mathcal{J}_{v_o} = \{J_{v_o,v} \ : v \in V\}$.

The output of *Improve* is one of the following:

(i) A check that (G, w) is conservative and $\mathcal{J}_{v_o}$ is optimal.

(ii) A negative cycle C of (G, w).

(iii) An acyclic $\{v_o, v\}$-join $\widetilde{J}_{v_o,v}$ with $w(\widetilde{J}_{v_o,v}) < w(J_{v_o,v})$.

Improve can be employed to test conservativeness of bipartite unit pairs or to find shortest paths in conservative bipartite unit pairs. *Improve* relies on three operations: *Clean*, *Switch* and *Contract*. Procedure *Contract* takes as input:

- A unit pair (G, w) and a clean family $\mathcal{J}_{v_o}$, where $w(J_{v_o,v}) = w(v_o v) = +1$ for every neighbor v of v_o.

Contract obtains from G a new graph $G' = (V', E)$ by contracting the star $\delta(v_o)$ into a new node v_o'. This introduces loops but E and w are left uneffected. *Contract* sets $J_{v_o',v_o'} = \emptyset$ and for every node $v \in V' \backslash \{v_o'\}$, $J'_{v_o',v}$ is obtained from $J_{v_o,v}$ by removing the unique edge having v_o as endnode. (Uniqueness follows since $\mathcal{J}_{v_o}$ is clean). So the output is the following:

- A unit pair (G', w) and a family $\mathcal{J}'_{v_o'}$ of joins rooted at v_o'.

Theorem 3.1 *Let $(G', w, \mathcal{J}'_{v_o'})$ be the output of* Contract *when applied to $(G, w, \mathcal{J}_{v_o})$. Then (G, w) is conservative and $\mathcal{J}_{v_o}$ is optimal if and only if (G', w) is conservative and $\mathcal{J}'_{v_o'}$ is optimal.*

Proof: Let C be a negative cycle of (G, w). By contracting all edges in $\delta(v_o)$, C becomes an Eulerian graph of negative weight since $\delta(v_o) \subseteq E_+$. So C is the disjoint union of cycles, at least one of them is negative and (G', w) is not conservative.
Let $\widetilde{J}_{v_o,v}$ be a join of (G, w) such that $w(\widetilde{J}_{v_o,v}) < w(J_{v_o,v})$. If v is a neighbor of v_o then $w(\widetilde{J}_{v_o,v}) \leq 0$ since $w(J_{v_o,v}) = 1$. Contracting $\delta(v_o)$, $\widetilde{J}_{v_o,v}$ becomes an Eulerian graph of negative weight since $\widetilde{J}_{v_o,v}$ has at least one edge incident at v_o. Again (G', w) is not conservative. Assume v is not a neighbor of v_o. Contracting $\delta(v_o)$, $\widetilde{J}_{v_o,v}$ becomes a $\{v_o', v\}$-join $\widetilde{J}'_{v_o',v}$ with $w(\widetilde{J}'_{v_o',v}) = w(\widetilde{J}_{v_o,v}) - 1 < w(J_{v_o,v}) - 1 = w(J'_{v_o',v})$. So $\mathcal{J}'_{v_o'}$ is not optimal.

Conversely let C be a negative cycle in (G', w). In (G, w), either C is a negative cycle, or a $\{u, v\}$-join, with u and v neighbors of v_o. In the second case $\widetilde{J}_{v_o,v} = \{v_o u\} \cup C$ is a $\{v_o, v\}$-join of (G, w) with $w(\widetilde{J}_{v_o,v}) \leq 0 < 1 = w(J_{v_o,v})$.
So we assume (G', w) to be conservative. Let $\widetilde{J}'_{v_o',v}$ be a $\{v_o', v\}$-join with $w(\widetilde{J}'_{v_o',v}) < w(J'_{v_o',v})$. We can assume $\widetilde{J}'_{v_o',v}$ to be acyclic, hence $d_{\widetilde{J}'_{v_o',v}}(v_o') = 1$. So there exists a neighbor u of v_o such that $\widetilde{J}'_{v_o',v}$ is a $\{u, v\}$-join in G. Hence $\widetilde{J}_{v_o,v} = \widetilde{J}'_{v_o',v} \cup \{v_o u\}$ is a $\{v_o, v\}$-join in G and $w(\widetilde{J}_{v_o,v}) = w(\widetilde{J}'_{v_o',v}) + 1 < w(J'_{v_o',v}) + 1 = w(J_{v_o,v})$. $\square$

Improve starts the following Recursion with (G, w) and the clean family $\mathcal{J}_{v_o}$ as input.

Recursion A bipartite unit pair (G^*, w^*) and a clean family of joins $\mathcal{J}_{v_o^*} = \{J_{v_o^*,v} : v \in V(G^*)\}$ rooted at v_o^* are received as input.
If G^* contains a single node, stop: (G, w) is conservative and $\mathcal{J}_{v_o}$ is optimal.
If $w^*(J_{v_o^*,v_o'}) \leq -1$ for some neighbor v_o' of v_o^*, then apply *Switch* to $(G^*, w^*, \mathcal{J}_{v_o^*}, v_o')$.

Otherwise $w^*(J_{v_o^*,v}) \geq 1$ for all neighbors v of v_o^*. If there exists an edge $v_o^* v \in \delta(v_o^*)$ such that $w^*(v_o^* v) < w^*(J_{v_o^*,v})$, define $\widetilde{J}_{v_o^*,v} = \{v_o^* v\}$ and go to the Surface Step.

Otherwise $\delta(v_o^*) \subseteq E_+$ and $w^*(J_{v_o^*,v}) = 1$ for every neighbor v of v_o^*. Apply *Contract* to $(G^*, w^*, \mathcal{J}_{v_o^*})$.

The output $(G', w', \mathcal{J}_{v_o'})$ of either *Switch* or *Contract* is given to *Clean*. If *Clean* finds a negative cycle or an acyclic join $\widetilde{J}_{v_o',v}$ such that $w'(\widetilde{J}_{v_o',v}) < w'(J_{v_o',v})$, go to the Surface Step. Otherwise the clean family obtained and (G', w') are the input of the next Recursion.

Surface Step Let $(G, w) = (G_0, w_0), (G_1, w_1), \ldots, (G_k, w_k)$ be the sequence of pairs computed by *Switch* or *Contract* in the applications of the Recursion, where in (G_k, w_k) a negative cycle or a join $\widetilde{J}_{v_k,v}$ such that $w_k(\widetilde{J}_{v_k,v}) < w_k(J_{v_k,v})$ has been found. With k applications of Theorems 2.1 and 3.1 (whose proofs are constructive) we can find a negative cycle in (G, w) or a join $\widetilde{J}_{v_o,v}$ such that $w(\widetilde{J}_{v_o,v}) < w(J_{v_o,v})$. If an "improved" join $\widetilde{J}_{v_o,v}$ has been found, apply *Clean* one last time to obtain a negative cycle or an acyclic improved join.

Remark 3.2 *Since (G, w) is a bipartite unit pair, G is a bipartite loopless graph and this property is maintained by the above algorithm.*

Remark 3.3 *Since (G, w) is a bipartite unit pair, then $w(J_{v_o,v})$ is odd (hence distinct from 0) whenever v and v_o are neighbors. So the two cases $w(J_{v_o,v}) \leq -1$ for some neighbor v of v_o and $w(J_{v_o,v}) \geq 1$ for all neighbors v of v_o considered in the Recursion are exhaustive.*

Remark 3.4 *The polynomiality of* Improve *is straightforward: Each time we apply* Switch*, we reduce the number of negative edges. Each time we apply* Contract*, we reduce the number of nodes. In order for (G, w) to be conservative E_- has to be a forest. Therefore we can assume $|E_-| < n$ and so the number of calls to* Switch *or* Contract *is $\mathcal{O}(n)$.*

So, if (G, w) is a bipartite unit pair which is conservative and $\mathcal{J}_{v_o}$ is a clean family of joins rooted at v_o, then, with $\mathcal{O}(n^2)$ calls to *Improve*, $\mathcal{J}_{v_o}$ can be turned into a clean optimal family.

4 Finding Optimal T-joins

In this section, we show that a version of *Improve* for general weight functions can be used to compute an optimal T-join, and hence an optimal matching, in any pair (G, w).

We first describe a strongly polynomial procedure, *w-Improve*, which takes as input:

- A pair (G, w), where w is rational (hence integral). A clean family $\mathcal{J}_{v_o}$.

and whose output is one of the following:

(i) A check that (G, w) is conservative and $\mathcal{J}_{v_o}$ is optimal.

(ii) A negative cycle C of (G, w).

(iii) An acyclic $\{v_o, v\}$-join $\widetilde{J}_{v_o,v}$ with $w(\widetilde{J}_{v_o,v}) < w(J_{v_o,v})$.

w-Improve relies on *Clean*, *Switch* and *w-Contract*:

w-Contract takes as input:

- A pair (G, w), where $w(e) \neq 0$ for every $e \in E$. A clean family $\mathcal{J}_{v_o}$ such that $0 \leq w(J_{v_o,v}) \leq w(v_o v)$ for every $v_o v \in \delta(v_o)$.

Let $\overline{w} = min_{v_o v \in \delta(v_o)}\{\frac{w(J_{v_o,v})+w(v_o v)}{2}\} = \frac{w(J_{v_o,\overline{v}})+w(v_o \overline{v})}{2}$. Note that $\overline{w} > 0$ since $w(v_o \overline{v}) > 0$. Define $w'(e) = w(e)$ for every edge $e \notin \delta(v_o)$ and $w'(e) = w(e) - \overline{w}$ for every edge $e \in \delta(v_o)$. Obtain G' from G by contracting the edges e with $w'(e) = 0$. Note that loops may have been created. Let v'_o be the node of G' containing v_o. (i.e. $v_o = v'_o$ if no edge has been contracted). A family $\mathcal{J}'_{v'_o}$ of joins is obtained from $\mathcal{J}_{v_o}$ by defining $J_{v'_o,v'_o} = \emptyset$, performing the above contraction in all the joins of $\mathcal{J}_{v_o}$ and discarding all joins $J_{v_o,v}$ if $v_o v$ is contracted.

So the output is the following:

- A pair (G', w'), where $w'(e) \neq 0$ for every edge e of G', and a family $\mathcal{J}'_{v'_o}$ of joins rooted at v'_o.

The proof of the following theorem in an immediate extension of the proof of Theorem 3.1 and is left to the reader.

Theorem 4.1 *Let $(G', w', \mathcal{J}'_{v'_o})$ be the output of w-*Contract *when applied to $(G, w, \mathcal{J}_{v_o})$. Then (G, w) is conservative and $\mathcal{J}_{v_o}$ is optimal if and only if (G', w') is conservative and $\mathcal{J}'_{v'_o}$ is optimal.*

Remark 4.2 *If (G, w) is a bipartite pair, and v is a neighbor of v_o, then $w(v_o v)$ and $w(J_{v_o,v})$ have the same parity and hence $\overline{w}$ is an integer.*

Algorithm *w-Improve* first contracts all edges of zero weight (this does not affect conservativeness nor optimality of joins) as to guarantee that $w(e) \neq 0\ \forall e$. (Note that this property is maintained by *Clean*, *Switch* and *w-Contract*).
Then *w-Improve* starts the following *Recursion* with (G, w) and $\mathcal{J}_{v_o}$ as input:

Recursion A pair (G^*, w^*), where $w^*(e) \neq 0$ for every edge e of G^*, and a clean family $\mathcal{J}_{v_o^*} = \{J_{v_o^*,v} : v \in V(G^*)\}$ of joins rooted at v_o^* are received as input.

If G^* has only one node, and no loop of G^* has negative weight stop: (G, w) is conservative and $\mathcal{J}_{v_o}$ is optimal. If some loop has negative weight, go to the Surface Step.

If $w^*(J_{v_o^*,v'_o}) < 0$ for some neighbor v'_o of v_o^*, then apply *Switch* to $(G^*, w^*, \mathcal{J}_{v_o^*}, v'_o)$.

Otherwise $w^*(J_{v_o^*,v}) \geq 0$ for all neighbors v of v_o^*. If there exists an edge $v_o^* v \in \delta(v_o^*)$ such that $w^*(v_o^* v) < w^*(J_{v_o^*,v})$, define $\widetilde{J}_{v_o^*,v} = \{v_o^* v\}$ and go to the Surface Step.

Otherwise $0 \leq w^*(J_{v_o^*,v}) \leq w^*(v_o^* v)$ for all edges $v_o^* v \in \delta(v_o^*)$. Apply *w-Contract* to $(G^*, w^*, \mathcal{J}_{v_o^*})$.

The output $(G', w', \mathcal{J}'_{v'_o})$ of either *Switch* or *w-Contract* is given to *Clean*. If *Clean* finds a negative cycle or an acyclic join $\widetilde{J}_{v'_o,v}$ such that $w'(\widetilde{J}_{v'_o,v}) < w'(J'_{v'_o,v})$, go to the Surface Step. Otherwise the clean family obtained and (G', w') are the input of the next Recursion.

Surface Step Let $(G,w) = (G_0,w_0),(G_1,w_1),\dots,(G_k,w_k)$ be the sequence of pairs computed by *Switch* or *Contract* in the applications of the Recursion, where in (G_k,w_k) a negative cycle or a join $\widetilde{J}_{v_k,v}$ such that $w_k(\widetilde{J}_{v_k,v}) < w_k(J_{v_k,v})$ has been found. With k applications of Theorems 2.1 and 3.1 (whose proofs are constructive) we can find a negative cycle in (G,w) or a join $\widetilde{J}_{v_o,v}$ such that $w(\widetilde{J}_{v_o,v}) < w(J_{v_o,v})$. If $\widetilde{J}_{v_o,v}$ has been found, apply *Clean* one last time.

Theorem 4.3 *Algorithm* w-Improve *calls* w-Contract $\mathcal{O}(m)$ *times and* Switch $\mathcal{O}(mn)$ *times.*

Proof: To apply *Switch* on $(G,w,\mathcal{J}_{v_o},v'_o)$ we must have $w(J_{v_o,v'_o}) < 0$. But $w_{J_{v_o,v'_o}}(J'_{v'_o,v_o}) = -w(J_{v_o,v'_o}) > 0$ and moreover $w_{J_{v_o,v'_o}}(J'_{v'_o,v}) = w(J_{v_o,v}) - w(J_{v_o,v'_o}) > w(J_{v_o,v})$.
This means that once the root of our family leaves v_o by *Switching*, it can not reenter v_o if no w-*Contraction* is involved. Hence the number of *Switchings*, between two consecutive w-*Contractions*, is at most n.

An edge uv of G is said *monotone* for $(G,w,\mathcal{J}_{v_o})$ when $|w(uv)| = |w(J_{v_o,u}) - w(J_{v_o,v})|$. Note that if uv is monotone for $(G,w,\mathcal{J}_{v_o})$, then uv is monotone for any $(G,w',\mathcal{J}'_{v_o})$ obtained from $(G,w,\mathcal{J}_{v_o})$ by *Switching*. The same holds for *Cleaning* as long as this procedure returnes a clean family as in (i). Assume now $(G',w',\mathcal{J}'_{v'_o})$ is obtained from $(G,w,\mathcal{J}_{v_o})$ by applying w-*Contract*. If $uv \in E(G')\backslash\delta(v'_o)$ is not monotone for $(G',w',\mathcal{J}'_{v'_o})$ then uv is not monotone for $(G,w,\mathcal{J}_{v_o})$. If $v'_o u \in \delta(v'_o)$ is not monotone for $(G',w',\mathcal{J}'_{v'_o})$ then $v_o u$ is not monotone for $(G,w,\mathcal{J}_{v_o})$. Moreover, if $v_o\overline{v}$ is monotone for $(G,w,\mathcal{J}_{v_o})$, then w-*Contract* contracts $v_o\overline{v}$ and so G' has less edges than G. Otherwise, if $v_o\overline{v}$ is not monotone for $(G,w,\mathcal{J}_{v_o})$, then $w'(v'_o\overline{v}) = w(v_o\overline{v}) - \overline{w} = \frac{w(v_o\overline{v})-w(J_{v_o,\overline{v}})}{2} = \overline{w} - w(J_{v_o\overline{v}}) = -w'(J_{v'_o\overline{v}})$ and $v'_o\overline{v}$ is monotone for $(G',w',\mathcal{J}'_{v'_o})$. Thus w-*Contract* is applied $\mathcal{O}(m)$ times and therefore *Switch* is applied $\mathcal{O}(mn)$ times. □

Schulz, Weismantel and Ziegler [6] show that if we can find in strongly polynomial time a "better" solution of a 0/1-integer program, then we can solve such a program in strongly polynomial time, provided an initial solution is available. So we can derive from w-*Improve* a strongly polynomial algorithm to compute optimal $\{u,v\}$-joins in conservative pairs.

We now show that if we can find optimal $\{u,v\}$-joins in conservative pairs then we can find optimal T-joins in any conservative pair:

Remark 4.4 *Let* $T = \{u_1,v_1,\dots,u_k,v_k\}$, $k \geq 2$ *be an even subset of nodes in a conservative pair* (G,w). *For* $i = 1,\dots,k$ *let* J_i *be an optimal* $\{u_i,v_i\}$*-join in the pair* $(G,w_{J_1\Delta\dots\Delta J_{i-1}})$ $(J_0 = \emptyset)$. *Then for* $i = 1,\dots,k$ *the pair* $(G,w_{J_1\Delta\dots\Delta J_i})$ *is conservative. In particular* $J = J_1\Delta J_2\dots\Delta J_k$ *is an optimal* T*-join in* (G,w).

To conclude, the following well known fact, see e.g. [5], shows that in computing an optimal T-join for a pair (G,w), w can always be assumed non-negative, hence (G,w) conservative:

Given a set of edges $F \subseteq E$ we define $T^F = \{v \in V : d_F(v) \text{ is odd}\}$, i.e. T^F is the set of nodes such that F is a T^F-join.

Theorem 4.5 *Given a pair (G, w) let T be any even subset of V and F be any subset of E. Then a subset J of E is a w-optimal T-join if and only if $J \Delta F$ is a w_F-optimal $(T \Delta T^F)$-join.*

Proof: Note first that J is a T-join if and only if $J \Delta F$ is a $(T \Delta T^F)$-join. By Theorem 1.1 J is w-optimal if and only if $(G, w_J) = (G, (w_F)_{F \Delta J})$ is conservative. This happens if and only if $J \Delta F$ is w_F-optimal. □

Corollary 4.6 *Given a pair (G, w) and an even subset T of V, let $E_- = \{e \in E : w(e) < 0\}$. If J is an optimal $(T \Delta T^{E_-})$-join in (G, w_{E_-}) then $J \Delta E_-$ is an optimal T-join in (G, w).*

Note that $w_{E_-} \geq 0$.

5 A Theorem of Sebő

In this section, we use algorithm *Improve* to prove a characterization, due to A. Sebő [9] and conjectured by A. Frank, of the bipartite unit pairs that are conservative. This theorem implies most structural theorems about optimal T-joins and packing of T-cuts, see for instance [9] and [2], such as Seymour's result on packing T-cuts in bipartite graphs [10] and Sebő's theorem on packing T-borders [8].

We begin by formalizing the inverse of *Contract*:
Decontract applies to a bipartite unit pair (G', w') in which a node v'_o has been distinguished as *root*. Let $\{\delta_1, \ldots, \delta_k\}$ be any partition of $\delta(v'_o)$, where we allow for some of the classes $\delta_1, \ldots, \delta_k$ to be empty. Let $v_o, v_1, \ldots, v_k$ be new nodes, not in G'. For $i = 1, \ldots, k$ replace every edge $v'_o v \in \delta_i$ with an edge $v_i v$ of the same weight. Next remove v'_o. Finally for $i = 1, \ldots, k$ add any number (at least one) of positive edges between nodes v_o and v_i and designate v_o as the new root.

A direct implication of algorithm *Improve* is the following:

Lemma 5.1 *The conservative bipartite unit pairs are precisely the bipartite unit pairs which can be obtained from a graph consisting of a single node (taken as the first root) and no edge through a sequence of* Decontractions *and* Switchings *on optimal acyclic $\{v'_o, v_o\}$-joins (where v'_o is the root before and v_o will be the root after* Switching *is performed).*

By Theorem 3.1 the decontraction of a conservative bipartite unit pair (G', w') is a conservative bipartite unit pair (G, w) with no zero weight cycle going through v_o. Hence for every optimal $\{v_o, v\}$-join $J_{v_o,v}$ in (G, w), $d_{J_{v_o,v}}(v_o) \leq 1$. In particular if $v = v_o$ then $J_{v_o,v} = \emptyset$, if v is a neighbor of v_o, $J_{v_o,v}$ is obtained by adding an edge in $\delta(v_o)$ to a zero weight Eulerian subgraph of (G', w') and if $v \in V(G')$ then $J_{v_o,v}$ is obtained by adding an edge in $\delta(v_o)$ to an optimal $\{v'_o, v\}$-join of (G', w').
For a pair (G, w) the *distance function* λ *centered at* v_o is defined as:

$$\lambda(v) = \min\{w(J_{v_o,v}) \ : J_{v_o,v} \text{ is a } \{v_o, v\}\text{-join}\} \quad \forall v \in V$$

Let λ' be the distance function centered at v_o' in (G', w'). The above argument implies $\lambda(v_o) = 0$, $\lambda(v) = 1$ for every neighbor of v_o and $\lambda(v) = \lambda'(v) + 1$ for every other node.

For every integer i, let G^i be the subgraph of G induced by $V^i = \{v \in V : \lambda(v) \leq i\}$. Let $\mathcal{D} = \{D : D \text{ is the node set of a connected component of some } G^i\}$.
After defining V_i', G_i' and $\mathcal{D}'$ analogously for (G', w'), note that:

$$\{\delta(D)\ : D \in \mathcal{D}\} = \delta(v_o) \cup \{\delta(D')\ : D' \in \mathcal{D}'\} \tag{1}$$

On the other hand, by the following remark, *Switching* does not affect $\mathcal{D}$.

Remark 5.2 *Let λ be the distance function centered at v_o in a conservative pair (G, w), $J_{v_o,v_o'}$ an w-optimal $\{v_o, v_o'\}$-join and λ' the distance function centered at v_o' in $(G, w_{J_{v_o,v_o'}})$. Then for every v in V we have $\lambda'(v) = \lambda(v) - w(J_{v_o,v_o'})$.*

Proof: Let $\mathcal{J}_{v_o}$ be an w-optimal family of joins rooted at v_o with $J_{v_o,v_o'} \in \mathcal{J}_{v_o}$. Theorem 2.1 implies the $w_{J_{v_o,v_o'}}$-optimality for the family $\{J'_{v_o',v} = J_{v_o,v_o'} \Delta J_{v_o,v}\ \forall J_{v_o,v} \in \mathcal{J}_{v_o}\}$.Thus $\lambda'(v) = w_{J_{v_o,v_o'}}(J'_{v_o',v}) = w(J_{v_o,v}) - w(J_{v_o,v_o'}) = \lambda(v) - w(J_{v_o,v_o'})$. □

For every $D \in \mathcal{D}$, let $\delta_{D,v} = 0$ when $v \in D$ and $\delta_{D,v} = 1$ when $v \notin D$. A. Sebő [9] proves the following good-characterization of conservativeness for bipartite unit pairs:

Theorem 5.3 *A bipartite unit pair (G, w) is conservative if and only if:*

$$|\delta(D) \cap E_-| = \delta_{D,v_o} \qquad \forall D \in \mathcal{D} \tag{2}$$

And from it he derives a characterization for optimal $\{v_o, v\}$-joins:

Theorem 5.4 *Let $J_{v_o,v}$ be a $\{v_o, v\}$-join in a conservative bipartite unit pair (G, w). Then $J_{v_o,v}$ is optimal if and only if:*

$$w(\delta(D) \cap J_{v_o,v}) = \delta_{D,v} - \delta_{D,v_o} \qquad \forall D \in \mathcal{D} \tag{3}$$

Remark 5.5 *Condition (3) characterizes optimal $\{v_o, v_o\}$-joins (i.e. zero-weight Eulerian subgraphs) in a conservative bipartite unit pair (G, w).*

Proof of Theorems 5.3 and 5.4: We first prove the "only if" direction of both theorems. Properties (2) and (3) hold trivially when G consists of a single node.
By Lemma 5.1, we need to consider two cases:
Case 1: (G, w) is obtained from a conservative bipartite unit pair (G', w'), for which both (2) and (3) hold, by *Decontracting*. Then $\delta(v_o) \subseteq E_+$ and consequently (1) implies that (2) holds for (G, w) since it holds for (G', w'). Let $J_{v_o,v}$ be an optimal $\{v_o, v\}$-join in (G, w). If $v = v_o$ then $J_{v_o,v} = \emptyset$ and (3) holds trivially. If v is a neighbor of v_o then $J_{v_o,v}$ is obtained by adding an edge in $\delta(v_o)$ to a zero weight Eulerian subgraph of (G', w'). If $v \in V(G')$ then $J_{v_o,v}$ is obtained by adding an edge in $\delta(v_o)$ to an optimal $\{v_o', v\}$-join of (G', w'). In both cases (3) holds for $\overline{D} = \{v_o\}$, since $v_o \in \overline{D}$ but $v \notin \overline{D}$ and consequently (1) implies that (3) holds for every $D \in \mathcal{D}$ since it holds for every $D \in \mathcal{D}'$ by induction.

Case 2: (G, w) is obtained from $(G, w_{J'_{v'_o,v_o}})$ by *Switching* on the $w_{J'_{v'_o,v_o}}$-optimal join $J'_{v'_o,v_o}$, where $(G, w_{J'_{v'_o,v_o}})$ satisfies (2) and (3). In $(G, w_{J'_{v'_o,v_o}})$, let E'_+, E'_-, λ' and $\mathcal{D}'$ as in accordance with the previous notation. By Remark 5.2 $\mathcal{D} = \mathcal{D}'$. Thus for every $D \in \mathcal{D} = \mathcal{D}'$:

$$|\delta(D) \cap E_-| = |\delta(D) \cap E'_-| + w_{J'_{v'_o,v_o}}(\delta(D) \cap J'_{v'_o v_o}) =$$
$$= \delta_{D,v'_o} + (\delta_{D,v_o} - \delta_{D,v'_o}) = \delta_{D,v_o}.$$

And the necessity of Theorem 5.3 is proven.

Let $\overline{J}_{v_o,v}$ be any optimal $\{v_o, v\}$-join in (G, w). Theorem 1.1 implies that $\overline{J}'_{v'_o,v} = \overline{J}_{v_o,v} \Delta J'_{v'_o,v_o}$ is a $w_{J'_{v'_o,v_o}}$-optimal $\{v'_o, v\}$-join. Thus for every $D \in \mathcal{D} = \mathcal{D}'$, we have:

$$\begin{aligned}
&w(\delta(D) \cap \overline{J}_{v_o,v}) = \\
&= w_{\delta(D)\cap J'_{v'_o,v_o}}((\delta(D) \cap J'_{v'_o,v_o})\Delta(\delta(D) \cap \overline{J}_{v_o,v})) - w_{\delta(D)\cap J'_{v'_o,v_o}}(\delta(D) \cap J'_{v'_o,v_o}) = \\
&= w_{J'_{v'_o,v_o}}(\delta(D) \cap \overline{J}'_{v'_o,v}) - w_{J'_{v'_o,v_o}}(\delta(D) \cap J'_{v'_o,v_o}) = \\
&= (\delta_{D,v} - \delta_{D,v'_o}) - -(\delta_{D,v_o} - \delta_{D,v'_o}) = \delta_{D,v} - \delta_{D,v_o}.
\end{aligned}$$

And the necessity of Theorem 5.4 is complete.

For the "if" part of both theorems observe first that for every edge $uv \in E$, we have $|\lambda(v) - \lambda(u)| = 1$ because u and v are on different sides of the bipartition of G. This means that $\{\delta(D) : D \in \mathcal{D}\}$ is a partition of E.

Let (G, w) be a pair satisfying (2) and C be any cycle in G. For every $D \in \mathcal{D}$, $|C \cap \delta(D)|$ is even and $|\delta(D) \cap E_-| \leq 1$, hence $|C \cap \delta(D) \cap E_-| \leq |C \cap \delta(D) \cap E_+|$. From this we obtain:

$$|C \cap E_-| = \sum_{D \in \mathcal{D}} |C \cap \delta(D) \cap E_-| \leq \sum_{D \in \mathcal{D}} |C \cap \delta(D) \cap E_+| = |C \cap E_+|$$

Thus (G, w) is conservative and the sufficiency of Theorem 5.3 follows.

Let $J_{v_o,v}$ be a $\{v_o, v\}$-join satisfying (3) in a conservative bipartite unit pair (G, w). Take an optimal $\{v_o, v\}$-join $\overline{J}_{v_o,v}$. Since $\overline{J}_{v_o,v}$ satisfies (3) we have:

$$w(J_{v_o,v} \cap \delta(D)) = \delta_{D,v} - \delta_{D,v_o} = w(\overline{J}_{v_o,v} \cap \delta(D)) \quad \forall D \in \mathcal{D}$$

which implies $w(J_{v_o,v}) = w(\overline{J}_{v_o,v})$ since $\{\delta(D) : D \in \mathcal{D}\}$ is a partition of E. The sufficiency of Theorem 5.4 follows. □

Acknowledgments

Thanks are due to the anonymous referees for their suggestions and advice.

References

[1] J. Edmonds and E. Johnson, Matchings, Euler tours and the chinese postman problem *Math. Programming* 5 88–124 (1973)

[2] A. Frank and Z. Szigeti, On packing $T-$cuts *Journal of Combinatorial Theory B*, Vol.61 No.2 263–271, (1994)

[3] Mei Gu Guan, Graphic programming using odd or even points *Chinese Mathematics*, Vol.1 273–277, (1962)

[4] E.L. Lawler, *Combinatorial Optimization: Networks and Matroids* Holt, Rinehart and Winston, (1976)

[5] L. Lovász and M.D. Plummer, *Matching Theory* Akadémiai Kiadó, (1986)

[6] A.S. Schulz, R. Weismantel and G.M. Ziegler, 0/1-Integer Programming: optimization and augmentation are equivalent *Lecture Notes in Computer Science 979*, 473-483, (1995)

[7] A. Sebő, Finding the $t-$join structure of graphs *Math. Programming*, Vol.36 123–134, (1986)

[8] A. Sebő, The Schrijver system of odd join polyhedra *Combinatorica*, Vol.8 103–116, (1988)

[9] A. Sebő, Undirected distances and the postman-structure of graphs *Journal of Combinatorial Theory B*, Vol. 49 10–39, (1990)

[10] P.D. Seymour, Odd cuts and plane multicommodity flows *Proceedings of the London Mathematical Society*, Vol.42 178–192, (1981)