Approaching $\frac{3}{2}$ for the s-t-path TSP

Vera Traub Jens Vygen

Research Institute for Discrete Mathematics, University of Bonn
{traub,vygen}@or.uni-bonn.de

Abstract

We show that there is a polynomial-time algorithm with approximation guarantee $\frac{3}{2} + \varepsilon$ for the s-t-path TSP, for any fixed $\varepsilon > 0$.

It is well known that Wolsey's analysis of Christofides' algorithm also works for the s-t-path TSP with its natural LP relaxation except for the *narrow cuts* (in which the LP solution has value less than two). A fixed optimum tour has either a single edge in a narrow cut (then call the edge and the cut *lonely*) or at least three (then call the cut *busy*). Our algorithm "guesses" (by dynamic programming) lonely cuts and edges. Then we partition the instance into smaller instances and strengthen the LP, requiring value at least three for busy cuts. By setting up a k-stage recursive dynamic program, we can compute a spanning tree (V, S) and an LP solution y such that $(\frac{1}{2} + O(2^{-k}))y$ is in the T-join polyhedron, where T is the set of vertices whose degree in S has the wrong parity.

1 Introduction

An instance of the s-t-path TSP consists of a finite metric space (V, c) and $s, t \in V$. The goal is to compute a path (V, F) with endpoints s and t (or a circuit if $s = t$) that contains all elements of V. Christofides [1976] and Hoogeveen [1991] proposed to compute a cheapest spanning tree (V, S), let $T := \{v \in V : |S \cap \delta(v)| \text{ odd}\} \triangle \{s\} \triangle \{t\}$ be the set of vertices with wrong parity, compute a cheapest T-join J and an Eulerian trail from s to t in $(V, S \mathbin{\dot\cup} J)$, and shortcut whenever a vertex is visited more than once. This algorithm has approximation guarantee $\frac{3}{2}$ for $s = t$ (Christofides [1976]), but only $\frac{5}{3}$ for $s \neq t$ (Hoogeveen [1991]).

Let us briefly explain our notation. Let $E = \binom{V}{2}$; so (V, E) is the complete graph on V. For $U \subseteq V$ let $E[U]$ denote the set of edges with both endpoints in U, $\delta(U)$ the set of edges with exactly one endpoint in U, and $\delta(v) := \delta(\{v\})$ for $v \in V$. For $x \in \mathbb{R}^E$ and $F \subseteq E$ we write $x(F) := \sum_{e \in F} x_e$, $c(x) := \sum_{e=\{v,w\} \in E} c(v, w) x_e$, and $c(F) := \sum_{e=\{v,w\} \in F} c(v, w)$. By $[m]$ we denote the index set $[m] := \{1, 2, \ldots, m\}$. An A-B-cut is an edge set $\delta(U)$ for some vertex set $A \subseteq U \subseteq V \setminus B$. As usual, $\triangle$ and $\mathbin{\dot\cup}$ denote symmetric difference and disjoint union. An *s-t-tour* is an edge set F such that (V, F) is an s-t-path (containing all vertices).

As all previous works, we use a classical idea of Wolsey [1980] for analyzing Christofides' algorithm. The following LP is obviously a relaxation of the s-t-path TSP (incidence vectors of s-t-tours are feasible solutions):

$$\begin{aligned} &\min c(x) \\ &s.t. \qquad x(\delta(U)) \geq 2 \quad \text{for } \emptyset \subset U \subseteq V \setminus \{s, t\}, \\ &\qquad\quad x(\delta(U)) \geq 1 \quad \text{for } \{s\} \subseteq U \subseteq V \setminus \{t\}, \\ &\qquad\quad\; x(e) \geq 0 \quad \text{for } e \in E. \end{aligned} \tag{1}$$

Let x^* be an optimum solution to the LP. Call a cut $\delta(U)$ (for $\emptyset \neq U \subset V$) *narrow* if $x^*(\delta(U)) < 2$. For the special case $s = t$ there are no narrow cuts, and thus the vector $\frac{1}{2}x^*$ is in the T-join polyhedron

$$\big\{x \geq 0 : x(\delta(U)) \geq 1 \text{ for } U \subset V \text{ with } |U \cap T| \text{ odd}\big\}. \tag{2}$$

Hence $c(J) \leq \frac{1}{2} c(x^*)$. This is Wolsey's [1980] analysis of Christofides' [1976] algorithm.

If $s \neq t$, this argument fails because of the narrow cuts. An, Kleinberg and Shmoys [2015] observed that the narrow cuts form a chain:

Proposition 1
Let $x \in \mathbb{R}^E_{\geq 0}$ be a feasible solution to the linear program (1). Then there are $m \geq 0$ sets $X_1, \ldots, X_m$ with $\{s\} \subseteq X_1 \subset X_2 \subset \cdots \subset X_m \subseteq V \setminus \{t\}$ such that

$$\{\delta(X_i) : i \in [m]\} = \{\delta(U) : \emptyset \neq U \subset V,\, x(\delta(U)) < 2\}.$$

Moreover, all of these sets can be computed by n^2 minimum cut computations in the graph (V, E) and thus in polynomial time.

reference	ratio
Hoogeveen [1991]	1.667
An, Kleinberg and Shmoys [2015]	1.618
Sebő [2013]	1.6
Vygen [2016]	1.599
Gottschalk and Vygen [2016]	1.566
Sebő and van Zuylen [2016]	1.529

Table 1: Previous approximation guarantees (rounded).

Proof: Let $X, Y \subseteq V$ with $x^*(\delta(X)) < 2$, $x^*(\delta(Y)) < 2$ and $s \in X \cap Y$. By the LP constraints we have $t \notin X$ and $t \notin Y$. Suppose neither $X \subseteq Y$ nor $Y \subseteq X$. Then, $X \setminus Y$ and $Y \setminus X$ are both nonempty and contain none of the vertices s and t. Thus,

$$4 > x^*(\delta(X)) + x^*(\delta(Y)) \geq x^*(\delta(X \setminus Y)) + x^*(\delta(Y \setminus X)) \geq 4,$$

a contradiction. To prove that the narrow cuts can be computed efficiently, we observe that for each narrow cut $C \in \mathcal{N}$ a pair $\{v, w\}$ of vertices exists such that C is the only narrow cut separating v and w. Thus, by computing a minimum v-w-cut for all pairs $\{v, w\}$ of vertices we will find all narrow cuts. □

Note that An, Kleinberg and Shmoys [2015] added degree constraints to the LP, but we do not need them.

Narrow cuts were the focus of all previous approximation algorithms (cf. Table 1). They all also proved upper bounds on the integrality ratio. Our approach is completely different. It yields the approximation ratio $\frac{3}{2} + \varepsilon$ for any $\varepsilon > 0$, but it does not yield an upper bound on the integrality ratio. The best known bound on the integrality ratio is $\frac{26}{17} \approx 1.529$, due to Sebő and van Zuylen [2016].

2 Outline of our algorithm

We will compute a spanning tree (V, S) and a parity correction vector in the T-join polyhedron (2) for $T := \{v \in V : |S \cap \delta(v)| \text{ odd}\} \triangle \{s\} \triangle \{t\}$. The parity correction vector will be a nonnegative combination of LP solutions. If x_1^* is an optimum solution to the LP (1), $\frac{1}{2}x_1^*$ would be good, but it is insufficient for narrow cuts C with $|C \cap S|$ even. Note that s-t-cuts $C = \delta(U)$ with $|C \cap S|$ odd are irrelevant because for these sets $|\{v \in U : |S \cap \delta(v)| \text{ odd}\}|$ is odd and thus $|U \cap T| = |\{v \in U : |S \cap \delta(v)| \text{ odd}\} \triangle \{s\} \triangle \{t\}|$ is even.

Let F be a fixed optimum s-t-tour. As all narrow cuts are s-t-cuts, we have for each narrow cut C that $|C \cap F|$ is odd. Suppose we know the partition $\mathcal{N}_1 = \mathcal{L} \,\dot\cup\, \mathcal{B}$ of the narrow cuts into *lonely* cuts (cuts $C \in \mathcal{N}_1$ with $|C \cap F| = 1$) and *busy* cuts (cuts $C \in \mathcal{N}_1$ with $|C \cap F| \geq 3$). Then we can compute a cheapest spanning tree (V, S) with $|S \cap C| = 1$ for all lonely cuts $C \in \mathcal{L}$. However, $\frac{1}{2}x_1^*$ is still insufficient for busy cuts.

Knowing the busy cuts, we can add the constraint $x(C) \geq 3$ for all $C \in \mathcal{B}$ to the LP

level l	fraction of x_l^* in parity correction vector	lower bound on LP value $x_l^*(C)$ of busy cuts C for			
		$C \in \mathcal{N}_1$	$C \in \mathcal{N}_2$	$C \in \mathcal{N}_3$	$C \in \mathcal{N}_4$
1	$\frac{8}{29}$	1	2	2	2
2	$\frac{4}{29}$	3	1	2	2
3	$\frac{2}{29}$	3	3	1	2
4	$\frac{1}{29}$	3	3	3	1

Table 2: Let x_l^* be the LP solution on level l, and $\mathcal{N}_l$ its narrow cuts. If we enforce $x(C) \geq 3$ for all busy cuts $C \in \mathcal{N}_i$ on all levels $l > i$, a nonnegative combination of the LP solutions x_l^* with the coefficients in the second column is a cheap parity correction vector for any tree (V, S) with $|S \cap C| = 1$ for every lonely cut C.

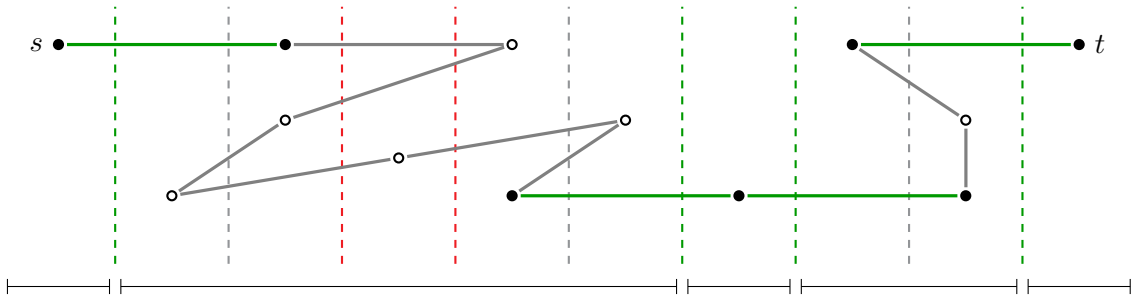

Figure 1: The dashed vertical lines show the narrow cuts. The solid lines show an optimum s-t-tour. The green edges and the green cuts are lonely. The intervals at the bottom indicate the sub-instances of the next recursion level, where the filled vertices serve as s' and/or t'. All other narrow cuts are busy, but only the red busy cuts will be passed to the next level because they have s' on the left and t' on the right. The gray busy cuts will automatically have value at least 3 as the proof will reveal.

and obtain a second solution x_2^*. Since $x_2^*(C)$ is big where $x_1^*(C)$ was insufficient, we can combine the two vectors; for example, $\frac{2}{3}x_1^* + \frac{1}{3}x_2^*$ is an LP solution with value at least $\frac{5}{3}$ at every cut $C \notin \mathcal{L}$ (while x_1^* could only guarantee ≥ 1). The second LP solution x_2^* has new narrow cuts, which again can be lonely or busy. Adding additional constraints $x(C) \geq 3$ for the new busy cuts, we get a third LP solution x_3^*, and so on. Table 2 shows how these LP solutions can be combined to a cheap parity correction vector.

To ensure that we can still find a spanning tree (V, S) with $|S \cap C| = 1$ for all lonely cuts C, they should form a chain, but this is not guaranteed with the procedure described above. Therefore we partition the original instance at the lonely cuts, solve separate LPs for the sub-instances, and combine the solutions. For this we also need to know the *lonely edges*, that is the edge $e \in C \cap F$ for every $C \in \mathcal{L}$. See Figure 1.

Of course, the main difficulty is that we do not know which cuts are lonely and which are busy, and we do not know the lonely edges. However, for each possibility of two subsequent

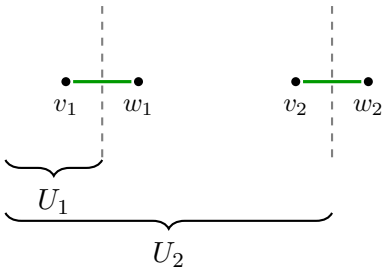

Figure 2: A possible sub-instance with vertex set $U_2 \setminus U_1$, $s' = w_1$, and $t' = v_2$. This sub-instance will be represented by the arc $((U_1, v_1, w_1), (U_2, v_2, w_2))$ in the digraph D. Note that the vertices w_1 and v_2 might be identical.

lonely cuts $\delta(U_1)$ and $\delta(U_2)$ with $\{s\} \subseteq U_1 \subset U_2 \subseteq V \setminus \{t\}$ and lonely edges $\{v_1, w_1\}$ and $\{v_2, w_2\}$ with $v_1 \in U_1$, $w_1, v_2 \in U_2 \setminus U_1$ and $w_2 \in V \setminus U_2$, we can consider the instance with vertex set $U_2 \setminus U_1$ and $s' = w_1$ and $t' = v_2$. See Figure 2. There are $O(n^6)$ such instances (due to Proposition 1). For each such instance we compute a spanning tree and an LP solution (recursively), and we combine these by dynamic programming.

The output of the dynamic program is a spanning tree (V, S) and an LP solution y. We set $T := \{v \in V : |\delta(v) \cap S| \text{ odd}\} \triangle \{s\} \triangle \{t\}$, compute a cheapest T-join J, find an Eulerian trail from s to t in $(V, S \dot\cup J)$, and shortcut. To bound the cost of J we will show that $(\frac{1}{2} + O(2^{-k}))y$ is a parity correction vector, where k denotes the number of levels in our recursive dynamic program.

Before we get into the details, let us mention one more subtle point. The busy cuts of previous levels can intersect several sub-instances. For a sub-instance on $U_2 \setminus U_1$ with $s' = w_1$ and $t' = v_2$, we will only pass a busy cut $C = \delta(U)$ to this sub-instance if $U_1 \cup \{s'\} \subseteq U \subseteq U_2 \setminus \{t'\}$. For the other busy cuts C (gray in Figure 1), the inequality $x(C) \geq 3$ will follow automatically from combining the LP solutions returned by the sub-instances.

3 The recursive dynamic program

In this section we describe the dynamic programming algorithm in detail. We call the algorithm recursively with a fixed recursion depth k. Moreover, we have fixed coefficients $\lambda_1 > \lambda_2 > \cdots > \lambda_k > 0$. We explain the choice of these constants depending on ε in Section 4.

The input to the dynamic program consists of

- sets $W_s, W_t \subseteq V$ with $W_s \cap W_t = \emptyset$;
- vertices $s', t' \in W := V \setminus (W_s \cup W_t)$ (see Figure 3; note that $s' = t'$ is possible);
- a collection $\mathcal{B}$ of busy $(W_s \cup \{s'\})$-$(W_t \cup \{t'\})$-cuts; and
- a level $l \in [k]$.

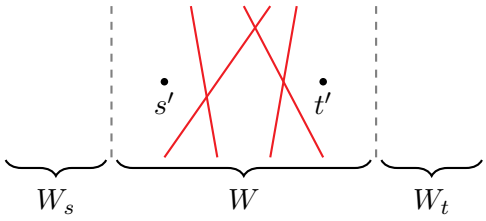

Figure 3: The input to the dynamic program. The dashed lines are the cuts $\delta(W_s)$, and $\delta(W_t)$. The red lines are possible busy cuts, i.e. elements of $\mathcal{B}$.

The output of the dynamic program is

- a tree (W, S);
- a vector $y \in \mathbb{R}^E_{\geq 0}$, which will contribute to the parity correction vector; and
- a chain $\mathcal{L}$ of $(W_s \cup \{s'\})$-$(W_t \cup \{t'\})$-cuts with $|S \cap C| = 1$ for all $C \in \mathcal{L}$.

We remark that for computing an s-t-tour it is sufficient to return the tree (W, S) and the cost of the vector y. The chain $\mathcal{L}$ and the explicit vector y are added only for the purpose of analysis.

The dynamic programming algorithm first computes an optimum solution x^* to the following linear program:

$$\begin{aligned} &\min c(x) \\ &s.t. \qquad x(\delta(U)) \geq 2 \quad \text{for } U \text{ with } \emptyset \neq U \subseteq W \setminus \{s', t'\} \\ &\qquad\quad x(\delta(U)) \geq 1 \quad \text{for } U \text{ with } \{s'\} \subseteq U \subseteq W \setminus \{t'\} \\ &\qquad\quad x(C) \geq 3 \quad \text{for } C \in \mathcal{B} \\ &\qquad\quad x(e) \geq 0 \quad \text{for } e \in E[W] \\ &\qquad\quad x(e) = 0 \quad \text{for } e \in E \setminus E[W]. \end{aligned} \tag{3}$$

The vector x^* restricted to edges $e \in E[W]$ is a feasible solution of linear program (1) for the instance of the metric s-t-path TSP with vertex set W and $s = s'$ and $t = t'$. By Proposition 1 the set of *narrow cuts*

$$\mathcal{N} := \left\{\delta(U) : x^*(\delta(U)) < 2, W_s \cup \{s'\} \subseteq U \subseteq V \setminus \left(W_t \cup \{t'\}\right)\right\},$$

forms a chain, i.e. there exist sets $W_s \cup \{s'\} \subseteq X_1 \subset X_2 \subset \cdots \subset X_m \subseteq V \setminus (W_t \cup \{t'\})$ such that $\mathcal{N} = \{\delta(X_i) : i \in [m]\}$.

If we have $l = k$, i.e. we are on the final level k, we return the vector $y := x^*$ and a minimum cost tree (W, S). Moreover, we return $\mathcal{L} = \emptyset$.

Otherwise, i.e. if $l < k$, we construct a directed auxiliary graph D with vertices

$$V(D) := \left\{(U, v, w) : \delta(U) \in \mathcal{N}, s' \in U, v \in U \cap W, w \in W \setminus U\right\} \dot\cup \left\{(W_s, \emptyset, s'), (V \setminus W_t, t', \emptyset)\right\}$$

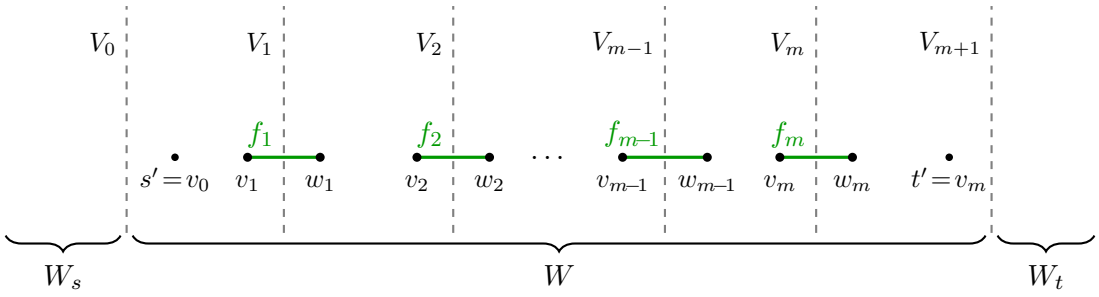

Figure 4: The dashed lines show the cuts $\delta(V_j)$ for $j = 0, 1, \ldots, m+1$, where the sets V_j are the sets left of the dashed lines. The partition of the vertex set into W_s, W_t and W is shown at the bottom of the picture. The edges f_j are drawn in green. We remark that the vertices w_j and v_{j+1} might be equal for $j = 0, 1, \ldots, m$.

and arcs

$$E(D) := \big\{((U_1, v_1, w_1), (U_2, v_2, w_2)) : U_1 \subset U_2, w_1, v_2 \in U_2 \setminus U_1\big\}.$$

Figure 2 illustrates the sets and vertices defining an arc $a \in E(D)$.

The next step of the algorithm is to compute weights for the arcs of D. For an arc $a = ((U_1, v_1, w_1), (U_2, v_2, w_2)) \in E(D)$ we call the dynamic program with

- $W_s = U_1$ and $W_t = V \setminus U_2$,
- $s' = w_1$ and $t' = v_2$,
- $\mathcal{B} = \mathcal{B}^a := \big\{\delta(U) \in \mathcal{N} \cup \mathcal{B} : U_1 \cup \{w_1\} \subseteq U \subseteq U_2 \setminus \{v_2\}\big\}$, and
- the level $l + 1$.

Let the output of this application of the dynamic program be the tree $(U_2 \setminus U_1, S^a)$, the vector $y^a \in \mathbb{R}^E_{\geq 0}$, and the chain $\mathcal{L}^a$ of cuts C. Then we define the cost of the arc $a \in E(D)$ to be

$$d(a) := \begin{cases} c\left(S^a\right) + \lambda_{l+1} \cdot c\left(y^a\right) + (1 + \lambda_{l+1}) \cdot c(v_2, w_2), & \text{if } w_2 \neq \emptyset \\ c\left(S^a\right) + \lambda_{l+1} \cdot c\left(y^a\right), & \text{if } w_2 = \emptyset. \end{cases} \tag{4}$$

Now we compute a shortest $(W_s, \emptyset, s')$-$(V \setminus W_t, t', \emptyset)$-path P in the auxiliary digraph D. We then define

$$S := \{e \in S^a : a \in E(P)\} \cup \{\{v, w\} : (U, v, w) \in V(P), v \neq \emptyset, w \neq \emptyset\}.$$

Let $(W_s, \emptyset, s') = (V_0, v_0, w_0), (V_1, v_1, w_1), (V_2, v_2, w_2), \ldots, (V_m, v_m, w_m), (V_{m+1}, v_{m+1}, w_{m+1}) = (V \setminus W_t, t', \emptyset)$ be the vertices of the path P visited in exactly this order. We define $a_j \in E(P)$ to be $a_j := ((V_j, v_j, w_j), (V_{j+1}, v_{j+1}, w_{j+1}))$ $(j = 0, \ldots, m)$. Moreover, for every $j \in [m]$ let $f_j := \{v_j, w_j\}$ (see Figure 4).

We then set the vector y' to be

$$y' := \sum_{a \in E(P)} y^a + \sum_{j=1}^{m} \chi^{f_j},$$

where χ^{f_j} is the incidence vector of f_j (i.e., $\chi^{f_j}_{f_j} = 1$ and $\chi^{f_j}_e = 0$ for $e \in E \setminus \{f_j\}$).

Define y to be the following convex combination of x^* and y':

$$y := \frac{\lambda_l - \lambda_{l+1}}{\lambda_l} \cdot x^* + \frac{\lambda_{l+1}}{\lambda_l} \cdot y'.$$

We set

$$\mathcal{L} := \{C : C \in \mathcal{L}^a \text{ for some } a \in E(P)\} \cup \{\delta(V_j) : j \in [m]\}$$

and return the edge set S, the vector y and the set $\mathcal{L}$.

4 Properties of the dynamic program

In this section we show several important properties of the output of the dynamic program. We show all these properties by induction on $k-l$, i.e. to prove them we assume that they hold for all levels l' with $l < l' \leq k$.

Lemma 2
$\mathcal{L}$ is a chain of $(W_s \cup \{s'\})$-$(W_t \cup \{t'\})$-cuts.

Proof: If $l = k$, we have $\mathcal{L} = \emptyset$. So we may assume $l < k$. If a cut C belongs to $\mathcal{L}$, it is a cut $\delta(V_j)$ for some $j \in [m]$ or is contained in $\mathcal{L}^a$ for some $a \in E(P)$. Recall that

$$W_s = V_0 \subset V_1 \subset V_2 \subset \cdots \subset V_m \subset V_{m+1} = V \setminus W_t.$$

Moreover, all cuts $\delta(V_j)$ for $j \in [m]$ are narrow cuts, i.e. $\delta(V_j) \in \mathcal{N}$, which implies $W_s \cup \{s'\} \subseteq V_j \subseteq V \setminus (W_t \cup \{t'\})$.

Now consider the cuts $\mathcal{L}^{a_j}$ for $j \in \{0, 1, \ldots, m\}$. By induction on $k-l$, the cuts in $\mathcal{L}^{a_j}$ are a chain of cuts of the form $\delta(U)$ for a vertex set U with $V_j \cup \{s'\} \subset U \subset V_{j+1} \setminus \{t'\}$. Thus, $\{\delta(V_j) : j \in [m]\} \subseteq \mathcal{N}$ remains a chain when adding the sets $\mathcal{L}^a$ for all $a \in E(P)$. □

Lemma 3
For $l < k$, an edge f_j with $j \in [m]$ is not contained in any cut $C \in \mathcal{L}^a$ for $a \in E(P)$.

Proof: Assume an edge f_j for $j \in [m]$ is contained in a cut $C \in \mathcal{L}^a$ for some $a \in E(P)$. As the edge f_j is contained in neither $\delta(V_{j-1})$ nor $\delta(V_{j+1})$, one endpoint is in $V_j \setminus V_{j-1}$ and the other endpoint is in $V_{j+1} \setminus V_j$. Using Lemma 2, this implies $a = a_{j-1}$ or $a = a_j$. If $a = a_{j-1}$, the endpoint v_j of f_j is contained in V_j and plays the role of t' in the dynamic

program computing the tree S^a. This implies by Lemma 2 that for a cut $C \in \mathcal{L}^a$ we have $C = \delta(U)$ for some U with $V_{j-1} \subseteq U \subseteq V_j \setminus \{v_j\}$, and hence $f_j \notin C = \delta(U)$. For the case $a = a_j$ a symmetric argument shows $f_j \notin C$ for $C \in \mathcal{L}^{a_j}$. □

Lemma 4
The graph (W, S) is a tree. For every cut $C \in \mathcal{L}$ we have $|S \cap C| = 1$.

Proof: For level $l = k$ the chain $\mathcal{L}$ is empty, and hence the statement is trivial. So assume $l < k$.

By the construction of the digraph D we have $W_s = V_0 \subset V_1 \subset V_2 \subset \cdots \subset V_m \subset V_{m+1} = V \setminus W_t$. We have $f_j \in \delta(V_j)$ and $f_j \notin \delta(V_h)$ for $h \neq j$. By induction, $(V_{j+1} \setminus V_j, S^{a_j})$ is a tree for every $j \in \{0, 1, \ldots, m\}$. The edges f_j (for $j \in [m]$) connect these trees to a tree spanning W. We observe that $S \cap \delta(V_j) = \{f_j\}$ for every V_j with $j \in [m]$.

By induction we have $|S^a \cap C| = 1$ for all $a \in E(P)$ and $C \in \mathcal{L}^a$. Moreover, note that edges of S^a are not contained in any cut $C \in \mathcal{L} \setminus \mathcal{L}^a$. As observed above, the tree (W, S) is constructed such that $S \cap \delta(V_j) = \{f_j\}$ for every $j \in [m]$. Thus, it only remains to show that an edge f_j for $j \in [m]$ can not be contained in a cut $C \in \mathcal{L}^a$ for any $a \in E(P)$ which is precisely the statement of Lemma 3. □

Lemma 5
For levels $l < k$ the cost $d(P)$ of the path P equals the cost $c(S) + \lambda_{l+1} \cdot c(y')$ of the tree S and the vector $\lambda_{l+1} \cdot y'$.

Proof: We have

$$c(S) = \sum_{a \in E(P)} c(S^a) + \sum_{j=1}^{m} c(f_j),$$

and

$$\lambda_{l+1} \cdot c(y') = \lambda_{l+1} \cdot \sum_{a \in E(P)} c\left(y^a\right) + \lambda_{l+1} \cdot \sum_{j=1}^{m} c(f_j).$$

Together with the definition (4) of the arc cost in D this shows $d(P) = c(S) + \lambda_{l+1} \cdot c(y')$. □

We fix an optimum s-t-tour F. We say an input $W_s, W_t, s', t', \mathcal{B}$ to the dynamic program is *consistent* with the tour F if F contains an s'-t'-path containing exactly the vertices in $V \setminus (W_s \cup W_t)$ and $|F \cap C| \neq 1$ for every cut $C \in \mathcal{B}$. We say that a path $\bar{P}$ in the auxiliary digraph D is *consistent* with the tour F if

- for every $(U, v, w) \in V_{\text{in}}(\bar{P})$ we have $\delta(U) \cap F = \{\{v, w\}\}$, and
- $|F \cap C| \neq 1$ for every cut $C \in \mathcal{N} \setminus \{\delta(U) : (U, v, w) \in V_{\text{in}}(\bar{P})\}$,

where $V_{\text{in}}(\bar{P})$ denotes the set of inner vertices of the path $\bar{P}$. Note that for parity reasons

$|F \cap C| \neq 1$ implies $|F \cap C| \geq 3$ for every s-t-cut C.

We denote by $F_{[s',t']}$ the edge set of the unique path from s' to t' that is contained in the path (V, F).

Lemma 6
If the input to the dynamic program is consistent with the tour F, we have

$$c(S) + \lambda_l \cdot c(y) \leq (1 + \lambda_l) \cdot c\left(F_{[s',t']}\right).$$

Proof: If the input of the dynamic program is consistent with the tour F, the incidence vector of $F_{[s',t']}$ is a feasible solution to the linear program (3) and thus

$$c(x^*) \leq c\left(F_{[s',t']}\right). \tag{5}$$

For $l = k$ we therefore have $c(y) = c(x^*) \leq c(F_{[s',t']})$; moreover $(W, F_{[s',t']})$ is a tree and hence $c(S) \leq c(F_{[s',t']})$.

Now assume $l < k$. Let $\bar{P}$ be the unique $(W_s, \emptyset, s')$-$(V \setminus W_t, t', \emptyset)$-path in D whose set of inner vertices is exactly the set of vertices $(U, v, w) \in V(D)$ with $\{\{v, w\}\} = F \cap \delta(U)$. Then $\bar{P}$ is consistent with the tour F.

For $a = ((U_1, v_1, w_1), (U_2, v_2, w_2)) \in E(\bar{P})$ let $s^a := w_1$ and $t^a := v_2$. The tour F is the disjoint union of the $F_{[s^a,t^a]}$ for $a \in E(\bar{P})$ and the edges $\{v, w\}$ for $(U, v, w) \in V_{\text{in}}(\bar{P})$. By induction on $k - l$, we have

$$c\left(S^a\right) + \lambda_{l+1} \cdot c\left(y^a\right) \leq (1 + \lambda_{l+1}) \cdot c\left(F_{[s^a,w^a]}\right).$$

Hence,

$$\begin{aligned} d(\bar{P}) &= \sum_{a \in E(\bar{P})} c\left(S^a\right) + \lambda_{l+1} \sum_{a \in E(\bar{P})} c\left(y^a\right) + (1 + \lambda_{l+1}) \sum_{(U,v,w) \in V_{\text{in}}(\bar{P})} c(v, w) \\ &\leq \sum_{a \in E(\bar{P})} (1 + \lambda_{l+1}) \cdot c\left(F_{[s^a,t^a]}\right) + \sum_{(U,v,w) \in V_{\text{in}}(\bar{P})} (1 + \lambda_{l+1}) \cdot c(v, w) \\ &= (1 + \lambda_{l+1}) \cdot c\left(F_{[s',t']}\right). \end{aligned}$$

Using Lemma 5 and the fact that P is no longer than $\bar{P}$ we get

$$c(S) + \lambda_{l+1} \cdot c(y') = d(P) \leq d(\bar{P}) \leq (1 + \lambda_{l+1}) \cdot c\left(F_{[s',t']}\right).$$

Using also (5) and

$$\lambda_l \cdot y = \lambda_{l+1} \cdot y' + (\lambda_l - \lambda_{l+1}) \cdot x^*$$

we get

$$\begin{aligned} c(S) + \lambda_l \cdot c(y) &= c(S) + \lambda_{l+1} \cdot c(y') + (\lambda_l - \lambda_{l+1}) \cdot c(x^*) \\ &\leq (1 + \lambda_{l+1}) \cdot c\left(F_{[s',t']}\right) + (\lambda_l - \lambda_{l+1}) \cdot c\left(F_{[s',t']}\right) \\ &= (1 + \lambda_l) \cdot c\left(F_{[s',t']}\right). \end{aligned}$$

□

Lemma 7
If $l < k$, the support of the vector y' is a subset of $E[W]$ and we have $y'(\delta(V_j)) = 1$ for every cut $\delta(V_j)$ with $j \in [m]$.

Proof: The vector y' is defined as the sum of vectors with support contained in $E[W]$. Thus, also the support of y' is a subset of $E[W]$. Next, we prove $y'(\delta(V_j)) = 1$ for every cut $\delta(V_j)$ with $j \in [m]$. We have $E(P) = \{a_j : j \in \{0, 1, \ldots, m\}\}$ and for every edge a_j the support of y^{a_j} is contained in $E[V_{j+1} \setminus V_j]$. Thus, for every pair of indices $j, r \in \{0, 1, \ldots, m\}$ we have $y^{a_j}(\delta(V_r)) = 0$. As an edge f_r is contained in $\delta(V_r)$, but not in any other cut $\delta(V_j)$ with $j \neq r$, we have $y'(\delta(V_j)) = y'(f_j) = 1$. □

Lemma 8
The vector y' (for $l < k$) and the vector y are feasible solutions to the following linear program:

$$\begin{aligned} &\min c(x) \\ &s.t. \qquad x(\delta(U)) \geq 2 \quad \text{for } U \text{ with } \emptyset \neq U \subseteq W \setminus \{s', t'\} \\ &\qquad\quad x(\delta(U)) \geq 1 \quad \text{for } U \text{ with } \{s'\} \subseteq U \subseteq W \setminus \{t'\} \\ &\qquad\quad x(e) \geq 0 \quad \text{for } e \in E[W] \\ &\qquad\quad x(e) = 0 \quad \text{for } e \in E \setminus E[W]. \end{aligned} \tag{6}$$

Proof: The vector x^* is a feasible solution to the linear program (3), and hence, also a solution to (6). If $l = k$, we have $y = x^*$, completing the proof for this case. We now assume $l < k$ and show, that also y' is a solution to (6). As y is a convex combination of x^* and y', this implies the statement of the Lemma.

The vector y' is defined as the sum of nonnegative vectors with support contained in $E[W]$, so $y' \geq 0$ and $y'(e) = 0$ for $e \in E \setminus E[W]$. It remains to check the cut constraints.

First consider $\delta(U)$ with $\{s'\} \subseteq U \subseteq W \setminus \{t'\}$. If there exists an index $j \in \{0, 1, \ldots, m\}$ such that $(V_{j+1} \setminus V_j) \cap U$ and $(V_{j+1} \setminus V_j) \setminus U$ are both not empty, we have $y'(\delta(U)) \geq y^{a_j}(\delta(U)) \geq 1$ by induction. Otherwise, there exists an index $j \in [m]$ such that $\delta(U)$ separates the sets $V_{j+1} \setminus V_j$ and $V_j \setminus V_{j-1}$. Then, the edge f_j is contained in $\delta(U)$, implying $y'(\delta(U)) \geq \chi^{f_j}(C) \geq 1$.

Now consider $\delta(U)$ with $\emptyset \neq U \subseteq W \setminus \{s', t'\}$.

Let $j_{\min} \in \{0, 1, \ldots m\}$ minimal such that $(V_{j_{\min}+1} \setminus V_{j_{\min}}) \cap U$ is nonempty and $j_{\max} \in \{0, 1, \ldots m\}$ maximal such that $(V_{j_{\max}+1} \setminus V_{j_{\max}}) \cap U$ is nonempty (see Figure 5).

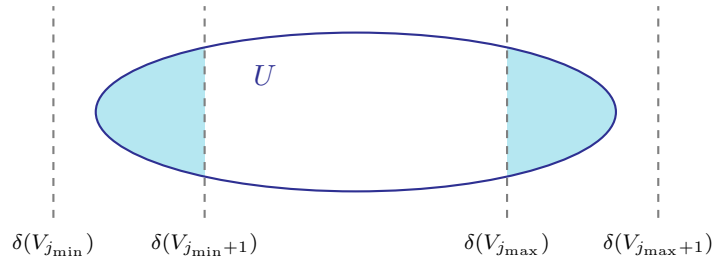

Figure 5: The picture illustrates the definition of $j_{\min}$ and $j_{\max}$. The dashed lines show the cuts written below. The indices $j_{\min}$ and $j_{\max}$ are chosen such that the two light blue sets are both nonempty.

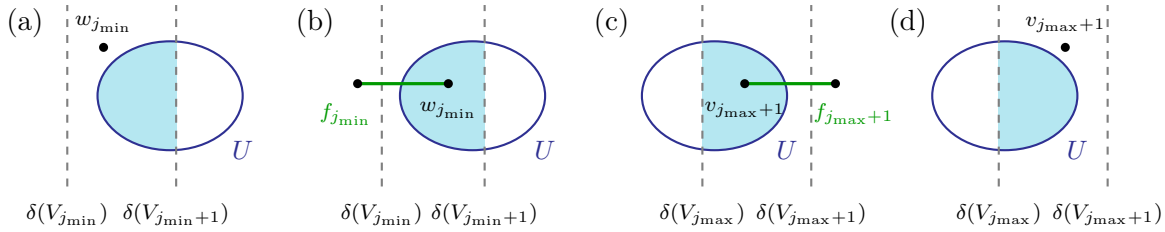

Figure 6: Different cases occurring in the proof of Lemma 8. The dashed vertical lines indicate the cuts written below. The set U is shown in blue. The light blue set is nonempty.

If $w_{j_{\min}}$ is not contained in U (Figure 6 (a)), the set $(V_{j_{\min}+1} \setminus V_{j_{\min}}) \setminus U$ is nonempty, and thus, we have $y^{a_{j_{\min}}}(\delta(U)) \geq 1$. This shows

$$y^{a_{j_{\min}}}(\delta(U)) + |\{w_{j_{\min}}\} \cap U| \geq 1. \tag{7}$$

Similarly, if $v_{j_{\max}+1}$ is not contained in U (Figure 6 (d)), we have $y^{a_{j_{\max}}}(\delta(U)) \geq 1$. This shows

$$y^{a_{j_{\max}}}(\delta(U)) + |\{v_{j_{\max}+1}\} \cap U| \geq 1. \tag{8}$$

If $|\{w_{j_{\min}}\} \cap U| = 1$, we have $j_{\min} \neq 0$ and $\chi^{f_{j_{\min}}}(\delta(U)) = 1$ (Figure 6 (b)). If $|\{v_{j_{\max}+1}\} \cap U| = 1$, we have $j_{\max} < m$ and $\chi^{f_{j_{\max}+1}}(\delta(U)) = 1$ (Figure 6 (c)). As we have $j_{\min} \leq j_{\max} < j_{\max}+1$ the edges $f_{j_{\min}}$ (for $j_{\min} > 0$) and $f_{j_{\max}+1}$ (for $j_{\max} < m$) are distinct edges. Thus, unless $j_{\max} = j_{\min}$ and

$$|\{w_{j_{\min}}\} \cap U| = |\{v_{j_{\max}+1}\} \cap U| = 0,$$

the inequalities (7) and (8) imply $y'(\delta(U)) \geq 2$.

So it remains to consider the case when U is a subset of $V_{j_{\max}+1} \setminus V_{j_{\max}} = V_{j_{\min}+1} \setminus V_{j_{\min}}$ and contains neither $w_{j_{\min}}$ nor $v_{j_{\max}+1}$. But then

$$y'(\delta(U)) \geq y^{a_{j_{\max}}}(\delta(U)) \geq 2.$$

□

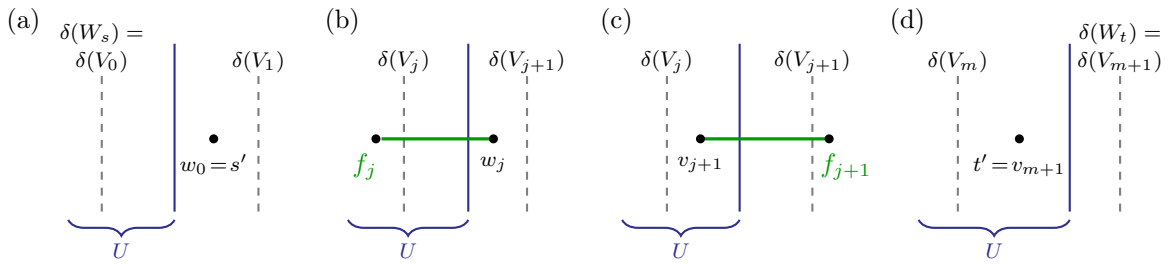

Figure 7: Different situations occurring in Case 1 of the proof of Lemma 9. The pictures (a) and (b) show the situation where $w_j \notin U$. Then, $j = 0$ (picture (a)) or $j \neq 0$ and $f_j \in \delta(U)$ (picture (b)). The pictures (c) and (d) show the situation where $v_{j+1} \in U$. Then, $j = m$ (picture (d)) or $j \neq m$ and $f_{j+1} \in \delta(U)$ (picture (c)).

Lemma 9
For every cut $C \in \mathcal{B}$ we have

$$y(C) \geq 3. \tag{9}$$

For every U with $W_s \subset U \subset V \setminus W_t$ with $s' \notin U$ or $t' \in U$ we have

$$y(\delta(U)) + |\{s'\} \setminus U| + |\{t'\} \cap U| \geq 3. \tag{10}$$

Proof: First note that for $y = x^*$ the claimed inequalities follow easily from the LP constraints (3): for $C \in \mathcal{B}$ we have $x^*(C) \geq 3$. For $W_s \subset U \subset V \setminus W_t$ we have $x^*(\delta(U)) \geq 1$, and if $s', t' \in U$ or $s', t' \notin U$ we have $x^*(\delta(U)) \geq 2$.

This proves the lemma for $l = k$. We again use induction on $k - l$.

Let now $l < k$. We fix a cut $C = \delta(U)$ with $W_s \subset U \subset V \setminus W_t$. We assume that $C \in \mathcal{B}$ or $s' \notin U$ or $t' \in U$. We will show

$$y'(\delta(U)) + |\{s'\} \setminus U| + |\{t'\} \cap U| \geq 3. \tag{11}$$

This implies $y'(C) \geq 3$ if $C \in \mathcal{B}$ because busy cuts are $(W_s \cup \{s'\})$-$(W_t \cup \{t'\})$-cuts. As y is a convex combination of y' and x^*, this will complete the proof. To show (11), we consider two cases.

Case 1: $V_j \subset U \subset V_{j+1}$ for some $j \in \{0, \ldots, m\}$

If $C \in \mathcal{B}$, we pass C as a busy cut to the next level, i.e. we have $C \in \mathcal{B}^{a_j}$, or we have $w_j \notin U$ or $v_{j+1} \in U$. If $s' \notin U$, we have $j = 0$ and $w_j = s' \notin U$. If $t' \in U$, we have $j = m$ and $v_{j+1} = t' \in U$. Thus, we can apply the induction hypothesis to the sub-instance corresponding to a_j. We use (9) if $C \in \mathcal{B}^{a_j}$ and (10) otherwise. In both cases we get

$$y^{a_j}(C) + |\{w_j\} \setminus U| + |\{v_{j+1}\} \cap U| \geq 3.$$

If $|\{w_j\} \setminus U| = 1$, then either $w_j = s'$ or ($j \neq 0$ and $\chi^{f_j}(C) = 1$). If $|\{v_{j+1}\} \cap U| = 1$, then either $v_{j+1} = t'$ or ($j \neq m$ and $\chi^{f_{j+1}}(C) = 1$). See Figure 7. This implies (11) by the

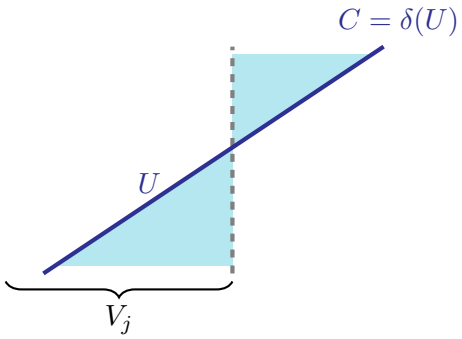

Figure 8: Case 2 of the proof of Lemma 9, where the cut C is crossing the cut $\delta(V_j)$, i.e. the two light blue sets $U \setminus V_j$ and $V_j \setminus U$ are nonempty. Note that $s' \in V_j$ and $t' \notin V_j$.

definition of y'.

Case 2: $V_j \subset U \subset V_{j+1}$ holds for no $j \in \{0, \dots, m\}$.

Then the cut C must cross some cut $\delta(V_j)$ with $j \in [m]$, i.e., $U \setminus V_j$ and $V_j \setminus U$ are nonempty (see Figure 8). Recall that $s' \in V_j$ and $t' \notin V_j$.

Since $t' \notin V_j \setminus U$, we have by Lemma 8

$$y'(\delta(V_j \setminus U)) \geq 2 - |\{s'\} \setminus U|.$$

Since $s' \notin U \setminus V_j$, we have by Lemma 8

$$y'(\delta(U \setminus V_j)) \geq 2 - |\{t'\} \cap U|.$$

Now by Lemma 7, we have $y'(\delta(V_j)) = 1$. Hence

$$\begin{aligned} y'(\delta(U)) + 1 &= y'(\delta(U)) + y'(\delta(V_j)) \\ &\geq y'(\delta(V_j \setminus U)) + y'(\delta(U \setminus V_j)) \\ &\geq 2 - |\{s'\} \setminus U| + 2 - |\{t'\} \cap U|. \end{aligned}$$

This shows (11). □

We now fix the constants $\lambda_1, \dots, \lambda_k$. We set the scaling constant Λ to be $\Lambda := 2^{k+1} - 3$. For $l \in [k]$ we set

$$\lambda_l := \frac{2^{k-l+1} - 1}{\Lambda}.$$

Let $0 < \varepsilon \leq \frac{1}{2}$. We choose the recursion depth k to be

$$k := \lceil \log_2 (1/\varepsilon) \rceil .$$

Then we have $k \geq \log_2\left(\frac{3}{2} + \frac{1}{4\varepsilon}\right)$ and thus,

$$\lambda_1 = \frac{2^k - 1}{\Lambda} = \frac{2^k - 1}{2^{k+1} - 3} = \frac{1}{2} + \frac{1/2}{2^{k+1} - 3} \leq \frac{1}{2} + \frac{1}{4 \cdot \left(\frac{3}{2} + \frac{1}{4\varepsilon}\right) - 6} = \frac{1}{2} + \varepsilon.$$

Lemma 10
If $y(C) < 2 - \frac{1}{\Lambda \cdot \lambda_l}$ for some $(W_s \cup \{s'\})$-$(W_t \cup \{t'\})$-cut, then $C \in \mathcal{L}$.

Proof: If $l = k$, we have $y(C) \geq 1 = 2 - \frac{1}{\Lambda \cdot \lambda_k}$ by Lemma 8. Let now $l < k$.

Let $W_s \cup \{s'\} \subseteq U \subseteq V \setminus (W_t \cup \{t'\})$ with $y(\delta(U)) < 2 - \frac{1}{\Lambda \cdot \lambda_l}$. By Lemma 8, the vector y is a feasible solution to the linear program (6). Hence, the set

$$\mathcal{N}_y := \left\{\delta(U') : y(\delta(U')) < 2,\ W_s \cup \{s'\} \subseteq U' \subseteq V \setminus (W_t \cup \{t'\})\right\}$$

of narrow cuts is a chain (by Proposition 1). By definition of the sets V_j, all cuts $\delta(V_j)$ (for $j \in [m]$) are contained in the set $\mathcal{N}$ of narrow cuts of the vector x^*. In particular, we have $x^*(\delta(V_j)) < 2$. By Lemma 7, we have $y'(\delta(V_j)) = 1$. As y is a convex combination of x^* and y', this shows $y(\delta(V_j)) < 2$, and thus, $\delta(V_j) \in \mathcal{N}_y$ for all $j \in [m]$. From this we can conclude that either $\delta(U) = \delta(V_j)$ for some $j \in [m]$, or $V_j \subset U \subset V_{j+1}$ for some $j \in \{0, 1, \ldots, m\}$.

If $\delta(U) = \delta(V_j)$ for some $j \in [m]$, we have $\delta(U) \in \mathcal{L}$ by construction of $\mathcal{L}$. Otherwise, we have $V_j \subset U \subset V_{j+1}$ for some $j \in \{0, 1, \ldots, m\}$. We distinguish two cases.

Case 1: $C \notin \mathcal{N}$ and $w_j \in U$ and $v_{j+1} \notin U$

If $C \in \mathcal{L}^{a_j}$, we have $C \in \mathcal{L}$. Otherwise we have $y^{a_j}(C) \geq 2 - \frac{1}{\Lambda \cdot \lambda_{l+1}}$ by induction. Moreover, $x^*(C) \geq 2$. As

$$y = \frac{\lambda_l - \lambda_{l+1}}{\lambda_l} \cdot x^* + \frac{\lambda_{l+1}}{\lambda_l} \cdot y',$$

this implies

$$y(C) \geq \frac{\lambda_l - \lambda_{l+1}}{\lambda_l} \cdot 2 + \frac{\lambda_{l+1}}{\lambda_l} \cdot \left(2 - \frac{1}{\Lambda \cdot \lambda_{l+1}}\right) = 2 - \frac{1}{\Lambda \cdot \lambda_l}.$$

Case 2: $C \in \mathcal{N}$ or $w_j \notin U$ or $v_{j+1} \in U$

Then $C \in \mathcal{B}^{a_j}$ or $w_j \notin U$ or $v_{j+1} \in U$. By Lemma 9 applied to this call of the dynamic program, we have

$$y^{a_j}(C) + |\{w_j\} \setminus U| + |\{v_{j+1}\} \cap U| \geq 3.$$

If $|\{w_j\} \setminus U| = 1$, then $j \neq 0$ and $\chi^{f_j}(C) = 1$. If $|\{v_{j+1}\} \cap U| = 1$, then $j \neq m$ and $\chi^{f_{j+1}}(C) = 1$. Hence,

$$y'(C) \geq 3.$$

By the LP constraints (3), we have $x^*(C) \geq 1$, and hence,

$$\begin{aligned} y(C) &\geq \frac{\lambda_l - \lambda_{l+1}}{\lambda_l} \cdot x^*(C) + \frac{\lambda_{l+1}}{\lambda_l} \cdot y'(C) \\ &\geq \frac{\lambda_l - \lambda_{l+1}}{\lambda_l} + 3 \cdot \frac{\lambda_{l+1}}{\lambda_l} \\ &= 2 + \frac{2 \cdot \lambda_{l+1} - \lambda_l}{\lambda_l} \\ &= 2 + \frac{2 \cdot \left(2^{k-l} - 1\right) - \left(2^{k-l+1} - 1\right)}{\Lambda \cdot \lambda_l} \\ &= 2 - \frac{1}{\Lambda \cdot \lambda_l}. \end{aligned}$$

□

5 The $\frac{3}{2} + \varepsilon$ approximation ratio

In this section we prove the approximation ratio of $\frac{3}{2} + \varepsilon$ for any fixed $\varepsilon > 0$. Let S^* be the spanning tree, $y^* \in \mathbb{R}^E$ the parity correction vector, and $\mathcal{L}^*$ the chain of cuts returned by the dynamic program with input $W_s = W_t = \emptyset$, $s' = s$, $t' = t$, $\mathcal{B} = \emptyset$, and level $l = 1$.

Lemma 11
If OPT denotes the cost of an optimum s-t-tour, we have

$$c(S^*) + \lambda_1 \cdot c(y^*) \leq \left(\frac{3}{2} + \varepsilon\right) \cdot OPT.$$

Proof: The input of the dynamic program computing S^* and y^* is consistent with any s-t tour. Thus, we get from Lemma 6 that

$$c(S^*) + \lambda_1 \cdot c(y^*) \leq (1 + \lambda_1) \cdot c(F)$$

for every s-t tour F. By the choice of k we have

$$1 + \lambda_1 \leq 1 + \frac{1}{2} + \varepsilon \leq \frac{3}{2} + \varepsilon,$$

implying

$$c(S^*) + \lambda_1 \cdot c(y^*) \leq \left(\frac{3}{2} + \varepsilon\right) \cdot \mathrm{OPT}.$$

□

Lemma 12
For $T = \{v \in V : |\delta(v) \cap S^*| \text{ odd}\} \triangle \{s\} \triangle \{t\}$ the vector $\lambda_1 \cdot y^*$ is contained in the T-join polyhedron $\{x \in \mathbb{R}^E_{\geq 0} : x(\delta(U)) \geq 1$ for $|U \cap T|$ odd, $\emptyset \neq U \subset V\}$.

Proof: From Lemma 4 we get that $|S^* \cap C| = 1$ for every cut $C \in \mathcal{L}^*$. Moreover, we have that all cuts $C \in \mathcal{L}^*$ are s-t-cuts. Thus, none of the cuts in $\mathcal{L}^*$ is a T-cut, i.e. we have $|U \cap T|$ even for every cut $\delta(U) \in \mathcal{L}^*$. Hence, it suffices to show $y^*(C) \geq 1$ for all cuts $C \notin \mathcal{L}^*$. Consider such a cut C. By Lemma 10, we have $y^*(C) \geq 2 - \frac{1}{\Lambda \cdot \lambda_1}$. Thus,

$$\lambda_1 \cdot y^*(C) \geq 2 \cdot \lambda_1 - \frac{1}{\Lambda} = 2 \cdot \frac{2^k - 1}{\Lambda} - \frac{1}{\Lambda} = \frac{2^{k+1} - 3}{\Lambda} = 1.$$

□

Theorem 13
Let $0 < \varepsilon \leq \frac{1}{2}$. Denote by $p(n,k)$ an upper bound on the time needed to solve a linear program (3) with $|V| = n$ and $|\mathcal{B}| \leq k \cdot n$. Then there exists a $\left(\frac{3}{2} + \varepsilon\right)$-approximation algorithm with runtime $O\left(n^{6\lceil \log_2(1/\varepsilon) \rceil} \cdot p\left(n, \lceil \log_2(1/\varepsilon) \rceil\right)\right)$.

Proof: We call the dynamic programming algorithm with level $l = 1$, $W_s = \emptyset$, $W_t = \emptyset$, $s' = s$, $t' = t$, and $\mathcal{B} = \emptyset$. Let (V, S^*) be the returned spanning tree and y^* the returned parity correction vector. We set $T := \{v \in V : |\delta(v) \cap S^*| \text{ odd}\} \triangle \{s\} \triangle \{t\}$, compute a cheapest T-join J and an Eulerian trail in $(V, S^* \dot\cup J)$, and shortcut. By Lemma 12 the cost $c(S^*) + c(J)$ is at most $c(S^*) + c(y^*)$. By Lemma 11 this is at most $\left(\frac{3}{2} + \varepsilon\right) \cdot \text{OPT}$, where OPT denotes the cost of an optimum s-t-tour.

Calling the dynamic program with level $l = k$ requires solving the linear program (3) once. For $l < k$, the digraph D has at most n^6 edges. Thus, calling the dynamic program with level $l < k$ requires solving the linear program (3) once, computing the narrow cuts (cf. Proposition 1), and calling at most n^6 times the dynamic program with level $l+1$. In every recursion step we add only (a subset of the) narrow cuts of the computed LP solution to the set $\mathcal{B}$. As the narrow cuts form a chain, these are at most n cuts. Thus, for the recursion depth $k = \lceil \log_2(1/\varepsilon) \rceil$ we have $|\mathcal{B}| \leq \lceil \log_2(1/\varepsilon) \rceil \cdot n$. By induction on the level l, we obtain an overall runtime of $O\left(n^{6(k-l)} \cdot p\left(n, \lceil \log_2(1/\varepsilon) \rceil\right)\right)$. □

Note that $p(n,k)$ can be chosen as a polynomial because the busy cut constraints can be checked explicitly, and the separation problem for the other cut constraints reduces to $O(n)$ minimum cut computations. Hence, we have a polynomial-time algorithm for any fixed $\varepsilon > 0$. We remark that we do not need the explicit LP solutions for our algorithm. The only property we use from the LP solutions is the LP value and the set of narrow cuts.

References

An, H.-C., Kleinberg, R., and Shmoys, D.B. [2015]: Improving Christofides' algorithm for the s-t path TSP. Journal of the ACM 62 (2015), Article 34

Christofides, N. [1976]: Worst-case analysis of a new heuristic for the traveling salesman problem. Technical Report 388, Graduate School of Industrial Administration, Carnegie-Mellon University, Pittsburgh 1976

Gottschalk, C., and Vygen, J. [2016]: Better s-t-tours by Gao trees. In: Integer Programming and Combinatorial Optimization; Proceedings of the 18th IPCO Conference; LNCS 9682 (Q. Louveaux, M. Skutella, eds.), Springer, Cham 2016, pp. 126–137

Hoogeveen, J.A. [1991]: Analysis of Christofides' heuristic: some paths are more difficult than cycles. Operations Research Letters 10 (1991), 291–295

Sebő, A. [2013]: Eight fifth approximation for TSP paths. In: Integer Programming and Combinatorial Optimization; Proceedings of the 16th IPCO Conference; LNCS 7801 (J. Correa, M.X. Goemans, eds.), Springer 2013, pp. 362–374

Sebő, A., and van Zuylen, A. [2016]: The salesman's improved paths: 3/2+1/34 integrality gap and approximation ratio. Proceedings of the 57th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2016), 118–127

Vygen, J. [2016]: Reassembling trees for the traveling salesman. SIAM Journal on Discrete Mathematics 30 (2016), 875–894

Wolsey, L.A. [1980]: Heuristic analysis, linear programming and branch and bound. Mathematical Programming Study 13 (1980), 121–134