The Salesman's Improved Paths: A 3/2+1/34 Approximation

András Sebő
Optimisation Combinatoire (G-SCOP)
CNRS, Univ. Grenoble Alpes
Grenoble, France
andras.sebo@grenoble-inp.fr

Anke van Zuylen
Department of Mathematics
College of William & Mary
Williamsburg VA, USA
anke@wm.edu

***Abstract*—We give a new, strongly polynomial algorithm and improved analysis of the metric s-t path TSP. It finds a tour of cost less than 1.53 times the optimum of the subtour elimination LP, while known examples show that 1.5 is a lower bound for the integrality gap.**

A key new idea is the deletion of some edges of Christofides' trees, and we show that the arising "reconnection" problems can be solved for a minor extra cost. On the one hand our algorithm and analysis extend previous tools, at the same time simplifying the framework. On the other hand new tools are introduced, such as a flow problem used for analyzing the reconnection cost, and the use of a set of more and more restrictive minimum cost spanning trees, each of which can still be found by the greedy algorithm. The latter leads to a simple Christofides-like algorithm completely avoiding the computation of a convex combination of spanning trees. Furthermore, the $3/2$ target-bound is easily reached in some relevant new cases.

keywords: path traveling salesman problem (TSP), approximation algorithm, integrality gap, Christofides' heuristic, matching theory, the Chinese postman problem, T-joins, polyhedra, matroids

I. Introduction

In the Traveling Salesman Problem (TSP), we are given a set V of n "cities", a cost function $c : \binom{V}{2} \to \mathbb{Q}_{>0}$, and the goal is to find a tour of minimum cost that starts and ends in the same city and visits each city exactly once. This "minimum length Hamiltonian circuit" problem is one of the most well-known problems of combinatorial optimization. It is not only *NP*-hard to solve but also to approximate, and even for quite particular lengths, since the Hamiltonian cycle problem in 3-regular graphs is *NP*-hard [10].

A condition on the cost function that helps in theory and is often satisfied in practice is known as the triangle inequality in complete graphs. A nonnegative function on the edges that satisfies this inequality is called a *metric* function. For a thoughtful and entertaining account of the difficulties and successes of the TSP, see Bill Cook's book [4].

If the cost function is a metric, we may relax the problem into finding a Eulerian sub-multigraph (V, E^*) of minimum cost $\sum_{e\in E^*} c(e)$, since an Eulerian trail can be shortcut to a tour without increasing the cost, with edge-multiplicities 0, 1 or 2.

Christofides [3] gave a very simple $\frac{3}{2}$-approximation algorithm for the metric TSP: he separated the problem into finding a minimum cost connected subgraph (minimum cost spanning tree) and completing this tree to an Eulerian subgraph, by correcting the parity of the constructed tree, S, i.e., by adding a minimum cost T_S-join, where T_S is the set of vertices that have odd degree in S (Chinese postman problem). Wolsey [20] observed that Christofides actually finds a solution that can be similarly bounded with the optimum of the well-known subtour elimination linear program (LP) that was introduced by Dantzig, Fulkerson and Johnson [5]:

$$\text{Min} \qquad c(x) := \sum_{e\in\binom{V}{2}} c(e)x(e)$$

subject to:

$$\sum_{e\in\delta(\{i\})} x(e) = f(\{i\}), \quad \text{for all } i \in V,$$

$$\sum_{e\in\delta(U)} x(e) \geq f(U), \quad \text{for all } \emptyset \subsetneq U \subsetneq V,$$

$$x(e) \geq 0, \qquad \text{for all } e \in \binom{V}{2},$$

where $f(U) \equiv 2$ for all sets U, and $\delta(U) = \{\{i,j\} \in E : i \in U, j \notin U\}$. The first set of constraints impose the condition that every vertex is visited once, and the second set of constraints, known as the subtour elimination constraints, ensure that every subset is visited at least once. Let x^* denote an optimal solution to the subtour elimination LP, and $\textsc{opt}_{LP} := \textsc{opt}_{LP}(c) := c(x^*)$ the optimal objective value. The integrality gap is the worst-case ratio, over all metric cost functions, of the cost of the optimal tour to $\textsc{opt}_{LP}$. Wolsey [20] observed that $\frac{n-1}{n}x^*$ is in the the spanning tree polytope, and that $x^*/2$ is in the T-join polyhedron for any set T of even size, which implies that the integrality gap of the subtour elimination LP is at most $\frac{3}{2}$.

Despite significant effort, no improvement is known for either the approximation ratio or the integrality gap.

A relevant generalization of the (metric) TSP is the (metric) $s-t$ path TSP, where $s, t \in V$ are part of the input.

The salesman starts in s, ends in t, and needs to visit every city once. In other words, the goal is to find a Hamiltonian path from s to t of minimum cost. The special case $s = t$ is the regular traveling salesman problem. When $s \neq t$, the subtour elimination LP for the $s-t$ path TSP is as above, but now defining $f(U) = 1$ if $|U \cap \{s,t\}| = 1$ and $f(U) = 2$ otherwise.

In this paper, we describe a new, combinatorial, strongly polynomial algorithm and improved analysis with the new approximation and integrality ratio $3/2+1/34$ for the metric $s-t$ path traveling salesman problem. To achieve this, we also need *to delete edges from Christofides' trees*, in the hope that they will be automatically reconnected during parity correction; however, this is not always the case, and whenever it is not, we also need to invest separately in *reconnection*. We deal with this in the algorithm by anticipating the cost of reconnection in the cost of the parity connecting T-joins we anticipate the cost of reconnection; when analyzing it, we use an LP for defining the distribution of a random choice for the reconnecting edges.

To the best of our knowledge, the first relevant occurrence of $s \neq t$ is in an exercise in [13]. Hoogeveen [12] provides a Christofides-type approximation algorithm for the metric case, with an approximation ratio of $5/3$ rather than $3/2$. There had been no improvement until An, Kleinberg and Shmoys [1], [2] improved this ratio to $\frac{1+\sqrt{5}}{2} < 1.618034$ with a simple algorithm and an ingenious new framework for the analysis. The algorithm in [1], [2] is the *best-of-many* version of Christofides' algorithm. It first determines a minimum cost solution x^* of the subtour elimination LP. Then writing x^* as a convex combination of spanning trees and adding Christofides' *parity correction* for each, the algorithm outputs the best of the arising solutions. The best-of-many algorithm was used by subsequent publications and is also used in the present work with some modification; mainly, we alter Christofides' algorithm by deleting certain edges from the spanning tree (see Section II), and by then correcting both parity and connectivity. Deletion has been already used for other versions of the TSP by Mömke and Svensson in a different, ingenious way avoiding disconnection [14], [15]. The novel method we develop disconnects and reconnects each of the occurring trees.

To analyze the best-of-many algorithm, An, Kleinberg and Shmoys [1] come up with a certain "master formula" to bound the average parity correction cost, that was also used in subsequent publications. Sebő [17] improved the analysis further by separating each spanning tree in the convex combination into an $s-t$ path and its complement. This separation makes it possible to use a simple and transparent master formula (see (5) in Section III), but, surprisingly, until now the original, quite involved one was used by all the papers on $s-t$ path TSP following [1].

Moreover, p^* can be further decomposed: it bounds from above a sum of naturally arising vectors important for parity correction [17] (denoted by x^Q, see the details later on). These vectors are getting an extended role in the present paper for analyzing the economies of deletion (Section III).

For the $s-t$ path TSP on "graph metrics", that is, cost functions that are defined as the shortest path distances on a given unweighted graph, Gao [8] proves the existence of a tree in the support of LP solution x^* that has exactly one edge in so-called narrow cuts (see Section II below). For such a tree, it is possible to bound the cost of the parity correction by $c(x^*)/2$, as in Wolsey's version of Christofides' analysis. This allows Gao to give a very elegant proof of the approximation ratio $3/2$ for graph metrics, a result that was first shown by [18] with a combinatorial, but more difficult proof.

For arbitrary metrics, the interesting idea of *choosing the convex combination of spanning trees in a particular way* was introduced in Vygen [19]. The claimed progress in the ratio was only 0.001, but the reassambling of trees $S \in \mathcal{S}$ which participate in the convex combination by local changes is further developed by Gottschalk and Vygen [11].

They generalize the concept of Gao-trees, and the reassambling leads to a powerful result: a convex combination using generalized Gao-trees. Although the use of local changes in Gottschalk and Vygen's proof of the existence of the convex combination is constructive, the algorithm it implies is not a polynomial time algorithm. Kanstantsin Pashkovich pointed out that this convex combination can be found (in weakly polynomial time) using the ellipsoid method [11]. Gottschalk and Vygen show that the best-of-many algorithm applied to this convex combination gives a 1.566-approximation and integrality ratio for the $s-t$ path TSP.

The convex combination of Gottschalk-Vygen will also play a role in our analysis, namely in bounding the reconnection cost (Section III-D). We provide a new interpretation of this convex combination in terms of matroid partition, which immediately implies a polynomial time algorithm. However, it turns out we only need the existence of this convex combination, with the added matroid-basis property of generalized Gao-trees: at the end of Section II, we sketch how, using this property, our algorithm can work with a set of spanning trees found by a greedy algorithm rather than a convex combination requiring heavy computations.

In the next section we provide more details for the approaches described above in the form we need them in this paper, along with the novel algorithm.

II. The Best-of-Many-with-Deletion Algorithm

Given a finite set V and costs $c : \binom{V}{2} \to \mathbb{Q}_{>0}$ satisfying the triangle inequality, let x^* be the minimum cost solution to the subtour elimination LP for the $s-t$ path TSP, and $E = \{e : x^*(e) > 0\}$. We will slightly abuse notation, and use the same notation for a (multi) subset of E and its own incidence vector in $\mathbb{Z}^E_{\geq 0}$. For a vector $z \in \mathbb{R}^E$ and any (multi) subset H of E, the usual notation $z(H) := \sum_{e \in H} z(e)$ is then just the scalar product of two vectors, as well as $c(z) := \sum_{e \in E} c(e)z(e)$.

We can write x^* as a convex combination of spanning trees; that is, there exist a collection of spanning trees $\mathcal{S}$ and coefficients $\lambda_S > 0$ for $S \in \mathcal{S}$ such that $\sum_{S \in \mathcal{S}} \lambda_S = 1$ and $x^* = \sum_{S \in \mathcal{S}} \lambda_S S$.

For a set of edges S, we let T_S be the set of odd degree vertices of S, i.e. those that have an odd number of incident edges from S. For two sets A, B, let $A \triangle B := (A \setminus B) \cup (B \setminus A)$ be the *symmetric difference* operation, which corresponds to the mod 2 sum of the incidence vectors. Then the vertex set with the "wrong" degree parity is $T_S \triangle \{s,t\}$. A T-join ($T \subseteq V$, $|T|$ is even) is a set of edges J such that $|J \cap \delta(\{i\})|$ is odd for $i \in T$ and $|J \cap \delta(\{i\})|$ is even for $i \in V \setminus T$. The operation "+" between sets means the disjoint union (sum of the multiplicity vectors). If S is a spanning tree and J_S a $T_S \triangle \{s,t\}$-join, $(V, S + J_S)$ is a spanning connected multigraph in which s and t have odd degree and every other vertex has even degree. We call the edge set of such a graph an $\{s,t\}$-tour.

The best-of-many Christofides' (BOMC) algorithm computes a minimum cost $T_S \triangle \{s,t\}$-join for every tree S with $\lambda_S > 0$, and outputs the resulting $\{s,t\}$-tour of minimum total cost. The cost of the BOMC solution is at most the cost of the $\{s,t\}$-tour we obtain by adding a $T_{\mathcal{S}} \triangle \{s,t\}$-join to a randomly chosen spanning tree $\mathcal{S}$ with $\Pr(\mathcal{S} = S) = \lambda_S$. Clearly, the expected cost of $\mathcal{S}$ is equal to $c(x^*)$; the main difficulty for the analysis of the s-t path TSP version of Christofides' algorithm is the "parity correction" part. It follows from Edmonds and Johnson [7] that the $T_S \triangle \{s,t\}$-join polyhedron is

$$\begin{array}{ll} \sum_{e \in \delta(U)} y(e) \geq 1 & \text{for all } U \text{ such that} \\ & |U \cap (T_S \triangle \{s,t\})| \text{ is odd,} \\ y(e) \geq 0 & \text{for all } e \in E. \end{array}$$

As noted above, Wolsey [20] observed for the TSP (i.e. for $s = t$) and a solution x^* to the subtour elimination LP for the TSP that $x^*/2$ is in the T-join polyhedron for any set $T \subseteq V$, $|T|$ even. If $s \neq t$ this is not true anymore for a solution x^* to the corresponding subtour elimination LP since for $U \subseteq V$ containing exactly one of s and t, $f(U) = 1$ in the LP, and hence we are only guaranteed that $x^*(\delta(U))/2 \geq \frac{1}{2}$. If $U \subset V$ and $|U \cap \{s,t\}| = 1$, we call the edge set $\delta(U)$ an $s-t$ *cut*.

Following [1], [2], we say that an $s-t$ cut Q is *narrow* if $x^*(Q) < 2$. We let $\mathcal{Q}$ be the set of all narrow cuts, that is, $\mathcal{Q} = \{Q \subset E : \ Q \text{ is an } s-t \text{ cut}, x^*(Q) < 2\}$. Figure 1 shows an example of an optimal solution x^* to the subtour elimination LP for an $s-t$ path TSP and the narrow cuts $\mathcal{Q}$. An, Kleinberg and Shmoys [1], [2], observe that $\mathcal{Q}$ is defined by a chain:

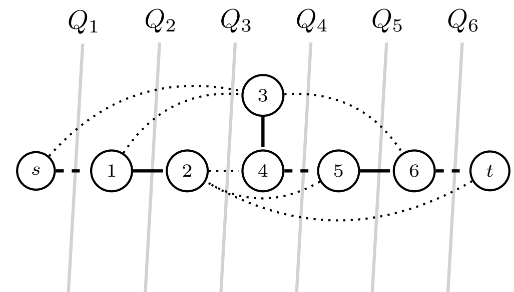

Figure 1. An example from Gao [9]: for full edges $x^*(e) = 1$, for dashed edges $x^*(e) = 2/3$ and for dotted edges $x^*(e) = 1/3$. The narrow cuts are indicated by gray lines and labeled $Q_1, \ldots, Q_6$.

Lemma 1 (An, Kleinberg, Shmoys [1], [2]): Let U_1, U_2 such that $s \in U_1 \cap U_2$, and $Q_i = \delta(U_i)$ is in $\mathcal{Q}$ for $i = 1, 2$. Then either $U_1 \subseteq U_2$, or $U_2 \subseteq U_1$.

We include a brief proof.

Proof: The lemma is a consequence of the submodular inequality that will be helpful in the following well-known equality form:

$$\begin{aligned} x^*(\delta(A)) + x^*(\delta(B)) &= x^*(\delta(A \cap B)) \\ + \quad & x^*(\delta(A \cup B)) + 2x^*(A \setminus B, B \setminus A), \end{aligned} \tag{1}$$

where (X, Y) is the set of edges with one endpoint in X, the other in Y. Now to show Lemma 1 apply (1) to $A := U_1$, $B := V \setminus U_2$. If the lemma does not hold, $A \cap B$ and $A \cup B$ are both non-empty, strict subsets of V. Furthermore, note that $A \cap B$ contains neither of s, t and $A \cup B$ contains both of s, t. Hence, by constraint (1), the first and second terms in the right hand side of (1) are at least 2, contradicting that the left hand side is strictly smaller than 4 since $\delta(U_1), \delta(U_2) \in \mathcal{Q}$. $\square$

An, Kleinberg and Shmoys [1], [2] exploit the fact that the $T_S \triangle \{s,t\}$-join polyhedron has a constraint for a narrow cut $Q = \delta(U)$ only if the spanning tree S has an even number of edges in Q, which allows them to remedy the fact that $x^*(Q)/2 < 1$ in an economic way only when necessary. Later works [9], [11], [17], [19] suggest improved ways of completing $x^*/2$ to a vector in the $T_S \triangle \{s,t\}$-join polyhedron. In Section III we introduce these approaches, and transform them to a novel method using several new ideas.

We recall now more precisely Gao's beautiful theorem [8] mentioned in the introduction: *there always exists a spanning tree S in the support of x^* (that is, in $S \subseteq \{e : x^*(e) > 0\}$) such that $|S \cap Q| = 1$ for every narrow cut Q.* Such a spanning tree will be called a *Gao-tree*. Whenever there exists a minimum cost spanning tree which is a Gao-tree (like for "graph-metrics") the $3/2$ ratio and integrality gap follow straightforwardly. Let us also recall Gottschalk and Vygen [11]'s generalization: a subtour LP solution x^* can be decomposed into "generalized Gao-trees", an ordered collection of spanning trees, such that the “first” $2 - x^*(Q)$ trees have a single edge in cut Q, and this holds for every narrow cut Q.

More precisely, let the different cut-sizes of narrow cuts be $2 - \zeta_1 > 2 - \zeta_1 - \zeta_2 > \ldots > 2 - \zeta_1 - \ldots - \zeta_k = 1$, and

$$\mathcal{Q}_i := \{Q \in \mathcal{Q} : x^*(Q) \leq 2 - \zeta_1 - \ldots - \zeta_i\}.$$

Note that $\sum_{i=1}^{k} \zeta_i = 1$. Let $\mathcal{B}_i := \{S \subseteq E : S$ is a spanning tree of G, $|S \cap Q| = 1$ for all $Q \in \mathcal{Q}_i\}$, and let P^i be the convex hull of $\mathcal{B}_i$ $(i = 1, \ldots, k)$. We state the theorem in a form that supports the interpretation we need:

Theorem 2 (Gottschalk and Vygen [11]): There exist $x^i \in P^i$ for $i = 1, \ldots, k$ such that $\sum_{i=1}^{k} \zeta_i x^i = x^*$.

For convenience, and to show the equivalence to the original form of the theorem, observe that x^i can be expressed as a convex combination of spanning trees in $\mathcal{B}_i$. We can thus write

$$\zeta_i x^i = \sum_{j=\ell_{i-1}+1}^{\ell_i} \lambda_j S_j, \ (i = 1, \ldots, k)$$

where $0 = \ell_0 < \ldots < \ell_k$, S_j $(j = 1, \ldots, \ell_k)$ are spanning trees, $\sum_{j=\ell_{i-1}+1}^{l_i} \lambda_i = \zeta_i$ and $S_j \in \mathcal{B}_i$ for $j = \ell_{i-1} + 1, \ldots, \ell_i$.

If for $Q \in \mathcal{Q}$ we have $x^*(Q) = 2 - \zeta_1 - \ldots - \zeta_i$, then we write $\ell_Q := \ell_i$, and we have

$$|S_j \cap Q| = 1 \text{ for } j = 1, \ldots, \ell_Q, \text{ and}$$

$$\sum_{j=1}^{\ell_Q} \lambda_j = 2 - x^*(Q). \tag{2}$$

We call a convex combination of trees with these properties *layered*. See Figure 2 for an example.

We now interpret this particular convex combination in a way that immediately yields a faster algorithm for obtaining it:

We say that edge $e \in E$ is in *layer* i $(i \in \{1, \ldots, k\})$, if there is a unique cut $Q \in \mathcal{Q}_i$ such that $e \in Q$. Edges may also be in several or no layers. Since $\mathcal{Q}_1 \supseteq \ldots \supseteq \mathcal{Q}_k$, the layers in which an edge is contained are consecutive numbers. (If it is contained exactly in layers $i \in [a, b] \cap \mathbb{N}$, then it is contained in at least 2 cuts of $\mathcal{Q}_i$ for $i < a$, and in none of the cuts $\mathcal{Q}_i$ for $i > b$.) Denote $L_i := \{e \in E : e$ is in layer $i\}$.

Now, let $\mathcal{I}_i = \{F_i \cup C_i : F_i$ is a forest in $E \setminus L_i$ and $C_i \subseteq L_i$, $|C_i \cap Q| \leq 1$ for each $Q \in \mathcal{Q}_i\}$. It is straighforward to check that $\mathcal{I}_i$ form the independent sets of a matroid, and that $\mathcal{B}_i$ is exactly the set of bases of this matroid. Hence, Theorem 2 states that it is possible to *partition* or decompose x^* into bases of the matroids $M_1, \ldots, M_k$ with coefficients given by $\zeta_1, \ldots, \zeta_k$. Therefore, using Edmonds' matroid

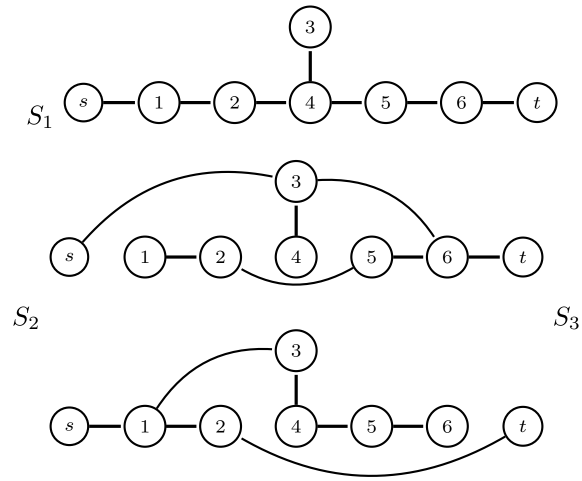

Figure 2. x^* from Figure 1 expressed as a layered convex combination of three spanning trees S_1, S_2, S_3, each with multiplier $\lambda_{S_i} = 1/3$. Note that S_1 is a so-called Gao-tree. All edges but $\{3, 4\}$ are lonely in S_1, and in S_2, S_3 only the edges incident to s and t are lonely.

partition algorithm [6], a layered convex combination can be found in strongly polynomial time.

In this paper, we will take a layered convex combination of x^* and modify the best-of-many Christofides' algorithm by *deleting* certain edges called *lonely* from the spanning trees in $\mathcal{S} = \{S_1, \ldots, S_{\ell_k}\}$. We denote by $L(S_j)$ the set of lonely edges of tree S_j, defined as

$$L(S_j) \;=\; \big\{e \in S_j : \text{there exists } Q \in \mathcal{Q} \text{ such that } Q \cap S_j = \{e\} \text{ and } \ell_Q \geq j\big\}. \tag{3}$$

For a tree $S \in \mathcal{S}$, denote by $\mathcal{Q}(S) \subseteq \mathcal{Q}$ the set of *lonely cuts* of S: cuts Q such that $|Q \cap S| = 1$ and for which the unique edge in $Q \cap S$ is in $L(S)$. In other words, $\mathcal{Q}(S_j) = \{Q \in \mathcal{Q} : \ell_Q \geq j\}$. If $\mathcal{Q}(S) = \mathcal{Q}$, then S is a Gao-tree; in a layered convex combination the first ℓ_1 spanning trees are Gao-trees.

Note that the definition of the layered convex combination guarantees the following “monotonocity property”:

$$\text{for each } Q \in \mathcal{Q}(S) \text{ and cut } Q' \text{ such that } x^*(Q') \leq x^*(Q) \text{ we have } Q' \in \mathcal{Q}(S), \tag{4}$$

that is, Q' is also lonely in S if it is smaller than another cut Q lonely in S. This is the crucial property that we exploit in Section III-D where we bound the reconnection cost, and the very reason why layered convex combinations are necessary.

For each tree $S \in \mathcal{S}$, we delete its lonely edges to obtain a forest $F(S) := S \setminus L(S)$. We write F instead of $F(S)$ whenever S is clear from the context. We then add a $T_F \triangle \{s, t\}$-join J_F to obtain a graph in which each vertex except for s and t has even degree and finally, we add a minimum cost doubled spanning tree on the components of this graph to get an $\{s, t\}$-tour. Note that adding a spanning

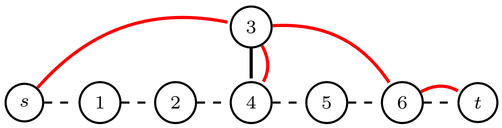

Figure 3. The forest $F_1 := F(S_1)$ obtained by removing the lonely edges from S_1 in Figure 2 contains only the edge $\{3,4\}$ indicated in (solid) black, the dashed edges are the lonely edges of S_1. The (solid) red edges give a possible $T_{F_1} \triangle \{s,t\}$-join J_{F_1}, where edges $\{s,3\}$ and $\{3,6\}$ are bad edges. An upper bound on $2D_{F_1+J_{F_1}}$ can be obtained by adding two copies of two out of the three lonely edges $\{s,1\}, \{1,2\}, \{2,4\}$ and adding two copies of either $\{4,5\}$ or $\{5,6\}$.

tree between the components suffices to make the graph connected, and we double it in order to keep the degree parities.

If we were to compute a minimum cost $T_F \triangle \{s,t\}$-join J_F, then $(V, F + J_F)$ may have many components, and the doubled spanning tree between the components could be very expensive. In order to prevent this, we compute a minimum cost join with respect to a modified cost, which, for every edge e, takes the sum of its cost and the anticipation of (or, in fact, an upper bound on) the reconnection cost related to its presence in J_F.

We explain now a way to upper bound the reconnection cost, which determines how to modify the costs.

For $Q \in \mathcal{Q}(S)$, we denote by e_S^Q the (incidence vector of) the unique edge of $Q \cap S$. For each edge $e \in E$, let $\mathcal{Q}(S,e)$ be the set of lonely cuts that contain e. Note that $\mathcal{Q}(S,e)$ may be empty. Now, the set of doubled edges is obtained by taking two copies of

$$\bigcup_{e \in \mathcal{J}_F} \{e_S^Q : \text{ for all but at most one } Q \text{ in } \mathcal{Q}(S,e)\}.$$

In other words, each edge e in J_F will add two copies of the lonely edges of all but one of the lonely cuts that e is contained in. See Figure 3 for an illustration.

To see that the set of doubled edges does indeed contain a doubled spanning tree between the components of $(V, F + J_F)$, note that if, for each $e \in J_F$, we were to add two copies of e_S^Q for *every* $Q \in \mathcal{Q}(S,e)$, then the set would contain at least two copies of *all* lonely edges in $L(S)$. This is because J_F must contain at least one edge in every cut $Q \in \mathcal{Q}(S)$, and thus $Q \in \mathcal{Q}(S,e)$ for at least one $e \in J_F$. Moreover, each edge $e \in J_F$ will be in a cycle with the edges in $F + \{e_S^Q : Q \in \mathcal{Q}(S,e)\}$, and hence we can choose one of the cuts $Q \in \mathcal{Q}(S,e)$ and not include e_S^Q. The best choice is of course to omit the edge with the highest cost and thus the cost of including e in the $T_F \triangle \{s,t\}$-join will be set to

$$c_F(e) = c(e) + \sum_{Q \in \mathcal{Q}(S,e)} 2c(e_S^Q) - \max_{Q \in \mathcal{Q}(S,e)} 2c(e_S^Q).$$

Note that $c_F(e) > c(e)$ only if e is contained in more than one lonely cut in S.

In addition to computing a *forest-based* $\{s,t\}$*-tour* for each $S \in \mathcal{S}$, the algorithm also computes *tree-based* $\{s,t\}$*-tours* as in the best-of-many Christofides' algorithm, and returns the least expensive among all constructed $\{s,t\}$-tours.

Algorithm 1 summarizes the description of the algorithm.

Let x^* be an optimal solution to the subtour elimination LP for the $s-t$ path TSP.
Write x^* as a layered convex combination of spanning trees.
Let $\mathcal{S}$ be the collection of trees with positive coefficient.
for $S \in \mathcal{S}$ **do**
  Construct forest-based $\{s,t\}$**-tour:**
  $F := F(S) := S \setminus L(S)$.
  For all $e \in E$, let $c_F(e) = c(e) + \sum_{Q \in \mathcal{Q}(S,e)} 2c(e_S^Q) - \max_{Q \in \mathcal{Q}(S,e)} 2c(e_S^Q)$.
  Let J_F^* be a minimum cost $T_F \triangle \{s,t\}$-join with respect to costs c_F.
  Contract the components of $(V, F + J_F^*)$, and let $2D_{F+J_F^*}$ be the edge set of a doubled minimum cost spanning tree in the contracted graph.
  $P_1(S) = F + J_F^* + 2D_{F+J_F^*}$.
  Construct tree-based $\{s,t\}$**-tour:**
  $P_2(S) = S + J_S^*$, where J_S^* is a minimum cost $T_S \triangle \{s,t\}$-join with respect to costs c.
end for
Return a minimum cost $\{s,t\}$-tour from among $\bigcup_{S \in \mathcal{S}} \{P_1(S), P_2(S)\}$.

Algorithm 1: Best-of-Many With Deletion

A variant of this algorithm avoids the computation of a layered convex combination for x^*, but uses a collection of k spanning trees easy to compute from x^* instead. The first of these trees, B_1, is a minimum cost Gao tree (having exactly one edge in each narrow cut in $\mathcal{Q} = \mathcal{Q}_1$), and analogously B_i is a minimum cost tree containing exactly one edge in each narrow cut $Q \in \mathcal{Q}_i$. By the observations below Theorem 2, B_i is a minimum cost basis of a matroid, and can thus be found using a greedy algorithm, in fact several spanning tree minimizations in subgraphs.

Now, let $x' := \sum_{i=1}^k \zeta_i B_i$. Then $c(x^*) \geq c(x') = \sum_i \zeta_i c(B_i)$ by Theorem 2 and the fact that B_i is a minimum cost basis in $\mathcal{B}_i$. Furthermore, although x' is not necessarily a feasible solution to the subtour elimination LP for the $s-t$ path TSP, we can view $\sum_{i=1}^k \zeta_i B_i$ as a layered convex combination of x'. This implies that the analysis below can be applied to x' and this revised convex combination, which gives a bound that is not worse than the one of Theorem 6. The details, including the precise computational complexity of this simplified version is deferred to the full version of this paper.

III. The Analysis

Given $x^* = \sum_{S\in\mathcal{S}} \lambda_S S$, where $\mathcal{S}$ is a set of spanning trees, $\lambda_S > 0$ for $S \in \mathcal{S}$, and $\sum_{S\in\mathcal{S}} \lambda_S = 1$, recall that this coefficient vector can be interpreted as a probability distribution. We will slightly abuse notation and let $\mathcal{S}$ denote both the support of the convex combination, and also a tree-valued random variable with $\Pr(\mathcal{S} = S) = \lambda_S$. (See Figure 2 for such a convex combination, where x^* is from Figure 1.)

Our analysis will bound the c_F-cost of a minimum cost $T_F\triangle\{s,t\}$-join J_F^* by constructing *a parity correction vector* y_F in the $T_F\triangle\{s,t\}$-join polyhedron. We separately bound $c(y_F)$ (referred to as the cost of parity correction) and $c_F(y_F) - c(y_F)$ (referred to as the cost of reconnection).

We let $\mathbb{E}[\cdot]$ denote the expectation or average of a random vector in $\mathbb{R}^E$; for example $\mathbb{E}[\mathcal{S}] = x^*$. Similarly, we view y_F as the expectation of a random $T_F\triangle\{s,t\}$-join. Various notations for trees S will be inherited by $\mathcal{S}$ (and similarly for T-joins): for instance, for a tree S, we denote $S(s,t)$ the $s-t$ path of S, and then $\mathcal{S}(s,t)$ is the path-valued random variable equal to $S(s,t)$ on the event $\mathcal{S} = S$.

A. Bounding the cost of tree-based $\{s,t\}$-tours

To analyze the average cost of the tree-based $\{s,t\}$-tours, we recall a simple well-known observation that we adopt:

Observation 3: Since $S = S(s,t) + S \setminus S(s,t)$, and for any $S \in \mathcal{S}$, $S + S \setminus S(s,t)$ is an $\{s,t\}$-tour, defining $p^* := \mathbb{E}[\mathcal{S}(s,t)]$, $q^* := \mathbb{E}[\mathcal{S}\setminus\mathcal{S}(s,t)]$, $x^* = \mathbb{E}[\mathcal{S}] = p^* + q^* \in \mathbb{R}^E$, the cost of the solution found by the best-of-many Christofides' algorithm can be bounded by

$$\mathbb{E}[c(\mathcal{S}) + c(\mathcal{S}\setminus\mathcal{S}(s,t))] = c(x^*) + c(q^*) = 2c(x^*) - c(p^*).$$

Similar to the analysis of [17], when $c(q^*)$ is sufficiently small compared to $c(p^*)$, then this bound provides the best approximation guarantee; when it is large, the other bound we will derive is better.

In fact, since we will be taking the minimum of this tree-based $\{s,t\}$-tour and another, forest-based $\{s,t\}$-tour, this observation allows us to "erase" $\mathcal{S}\setminus\mathcal{S}(s,t)$ from further consideration for parity correction, i.e., for analyzing the cost of the forest-based $\{s,t\}$-tours, we will only take the complementary $\mathcal{S}(s,t)$ into account.

B. Basic parity correction

Concretely, to analyze the parity correction for the forests, we will always use

$$\frac{x^*}{2} + \gamma\, S(s,t), \tag{5}$$

as a so-called *basic parity correction*, where γ is a chosen parameter between 0 and $1/2$. The basic parity correction plays a similar role to the vector $\beta\frac{x^*}{2} + (1-\beta)S$ that was used in the analysis of [1], which was also adopted by subsequent work [9], [11], [17], [19].

For $\gamma = 1/2$ the basic parity correction $\frac{x^*}{2} + \frac{1}{2}S(s,t)$ for any $S \in \mathcal{S}$ gives a feasible solution to the T-join polyhedron for any set T of even size, so in particular it allows us to bound the cost of the $T_S\triangle\{s,t\}$-join that corrects the parity for the tree S. On average, the basic parity correction with $\gamma = 1/2$ will equal $\frac{x^*}{2} + \frac{p^*}{2}$, and combined with Observation 3, we get the integrality gap and approximation ratio $\min\left\{\frac{3}{2}c(x^*) + \frac{1}{2}c(p^*), 2c(x^*) - c(p^*)\right\} \le \frac{5}{3}c(x^*)$ equal to Hoogeveen's ratio. We can similarly use $\frac{x^*}{2} + \frac{1}{2}S(s,t)$ as a fractional $T_F\triangle\{s,t\}$-join for the forest $F = S \setminus L(S)$; however, the summand $\frac{1}{2}\mathcal{S}(s,t)$ is too expensive to get a good bound on the parity correction cost.

To improve the ratio further, we will need $\gamma < 1/2$, but then the basic parity correction vector may not be feasible for the $T_F\triangle\{s,t\}$-join polyhedron. Hence, as in previous analyses of the best-of-many algorithm, we will need to add a *parity completion*[1] vector to the basic parity correction vector so as to ensure the constraints of the $T_F\triangle\{s,t\}$-join polyhedron are satisfied for the narrow cuts Q with $x^*(Q) < 2 - 2\gamma$ in which F contains an even number of edges.

However, we cannot reduce γ to 0, since the completion vectors are too expensive for large narrow cuts. The threshold for being "large" will be chosen optimally at the end of Section IV.

C. Parity completion

The parity completion vector will add (fractional) edges for each narrow cut Q if $|Q \cap F|$ is even. For each narrow cut Q we introduce a vector $x^Q \in \mathbb{R}^E$, where

$$x^Q(e) := \Pr(\mathcal{S}\cap Q = \{e\} \text{ and } e \in L(\mathcal{S})). \tag{6}$$

Note that $x^Q(Q) = \Pr(Q \in \mathcal{Q}(\mathcal{S}))$, and by (2):

$$x^Q(Q) = Pr(Q \in \mathcal{Q}(\mathcal{S})) \ge 2 - x^*(Q). \tag{7}$$

holds with equality (this inequality also holds for an arbitrary convex combination where the definition is $x^Q(e) = \Pr(\mathcal{S}\cap Q = \{e\})$ [17]).

Hence, we have that $\frac{x^*(Q)+x^Q(Q)}{2} = 1$, on every cut. In other words, adding $x^Q/2$ to the basic parity correction vector for every Q such that $|Q \cap F|$ is even gives a suitable vector to *complete* parity correction, that is, for *parity completion*.

On the other hand, if we let $\mathcal{F}$ be the random forest obtained by deleting $L(\mathcal{S})$ from $\mathcal{S}$, then we also have that $x^Q(e)$ is the *deletion probability* for edge e and cut Q. So by deleting the lonely edges from $\mathcal{S}$, we "save up" for parity completion. In Section IV, we show how to put these ingredients together to construct an appropriate

[1] Our terminology differs from that in [1], where the term "correction vector" is used for what we call the "parity completion vector". We use the term "parity correction vector" for a vector that is in the $T_F\triangle\{s,t\}$-join polyhedron, and it will be composed of basic parity correction and parity completions.

parity correction vector y_F, and we bound the mean parity correction cost by computing $\mathbb{E}[c(y_F(S)]$.

D. Reconnection

It remains to analyze $c_F(y_F) - c(y_F)$; the cost of the doubled edges that give an upper bound on the doubled spanning tree on the components of $(V, F + \mathcal{J}_F)$, where $\mathcal{J}_F$ is a $T_F \triangle \{s,t\}$-join, with $\mathbb{E}[\mathcal{J}_F] = y_F$. We will call the edges e with $c_F(e) > c(e)$ the *bad edges* for S, and denote them by $B(S)$, that is, $B(S) =$

$$\{e \in E : \text{there exist } Q_1 \neq Q_2 \in \mathcal{Q}(S),\, e \in Q_1 \cap Q_2\}.$$

Figure 3 shows a forest F and a $T_F \triangle \{s,t\}$-join J_F with two bad edges.

In order to bound $c_F(y_F) - c(y_F)$, we will need two ingredients. First, we will need to bound $\Pr(e \in \mathcal{J}_F) = y_F(e)$ for the bad edges $e \in B(S)$. As sketched above, our fractional $T_F \triangle \{s,t\}$-join y_F is going to consist of the basic parity correction $\frac{x^*}{2} + \gamma S(s,t)$ plus a parity completion vector for which we will use the x^Q vectors for $Q \in \mathcal{Q}$. There is no way to avoid that the basic parity correction vector contains bad edges, but the fact that we used a layered convex combination and the accompanying definition of the lonely edges for each $S \in \mathcal{S}$ guarantees that parity completion does not add more bad edges:

Lemma 4: Let $\mathcal{S}$ be the support of a layered convex combination, $S \in \mathcal{S}$, and $Q \in \mathcal{Q} \setminus \mathcal{Q}(S)$. Then $x^Q(B(S)) = 0$.

Proof: We prove $x^Q(Q') = 0$ for each $Q' \in \mathcal{Q}(S)$. Indeed, by (4), $x^*(Q') < x^*(Q)$, for otherwise $Q \in \mathcal{Q}(S)$. Also, by (4), Q' is lonely in every tree $S' \in \mathcal{S}$ where Q is lonely, so Q and Q' cannot have a common edge in such a tree S', proving $x^Q(Q') = 0$.

Therefore, the edges e with $x^Q(e) > 0$ are not contained in any cut of $\mathcal{Q}(S)$ at all, whereas the edges of $B(S)$ are contained in at least two cuts of $\mathcal{Q}(S)$. □

This lemma will allow us to bound the probability that y_F contains a bad edge e by $x^*(e)/2$, and thus $c_F(y_F) - c(y_F) = c_F(x^*/2) - c(x^*/2)$. The second ingredient will be the following lemma, which is proved by analyzing the combinatorial structure of the inequalities satisfied by x^*:

Lemma 5: Let $S \in \mathcal{S}$, let $F = S \setminus L(S)$ and let $\mathcal{Q}(S)$ be the lonely cuts in S. Then

$$c_F(x^*/2) - c(x^*/2) \leq \sum_{Q \in \mathcal{Q}(S)} (x^*(Q) - 1)c(e_S^Q).$$

Proof: Observe that

$$\begin{aligned} c_F(e) - c(e) &= \sum_{Q \in \mathcal{Q}(S,e)} 2c(e_S^Q) - \max_{Q \in \mathcal{Q}(S,e)} 2c(e_S^Q) \\ &\leq \sum_{Q \in \mathcal{Q}(S,e)} 2c(e_S^Q)(1 - x(e,Q)), \end{aligned}$$

for any values $x(e,Q) \geq 0$ for $e, \in E, Q \in \mathcal{Q}(S)$ such that $\sum_{Q \in \mathcal{Q}(S,e)} x(e,Q) \leq 1$. Hence, given such values, we can write $c_F(x^*/2) - c_F(x^*/2)$

$$\begin{aligned} &= \sum_{e \in E} \frac{x^*(e)}{2} \left(\sum_{Q \in \mathcal{Q}(S,e)} 2c(e_S^Q) - \max_{Q \in \mathcal{Q}(S,e)} 2c(e_S^Q) \right) \\ &\leq \sum_{e \in E} \frac{x^*(e)}{2} \left(\sum_{Q \in \mathcal{Q}(S,e)} 2c(e_S^Q)(1 - x(e,Q)) \right) \\ &= \sum_{Q \in \mathcal{Q}(S)} \sum_{e \in \mathcal{Q}(S)} \frac{x^*(e)}{2} \left(2c(e_S^Q)(1 - x(e,Q)) \right) \\ &= \sum_{Q \in \mathcal{Q}(S)} c(e_S^Q) \left(x^*(Q) - \sum_{e \in \mathcal{Q}(S)} x^*(e)x(e,Q) \right). \end{aligned}$$

So in order to prove the lemma, it suffices to prove that the following system of inequalities has a solution:

$$\begin{aligned} \sum_{Q \in \mathcal{Q}(S,e)} x(e,Q) &\leq 1 \qquad \text{for all } e \in E \\ \sum_{e \in \mathcal{Q}(S)} x(e,Q)x^*(e) &\geq 1 \qquad \text{for all } Q \in \mathcal{Q}(S) \\ x(e,Q) &\geq 0 \qquad \text{for all } e \in E, Q \in \mathcal{Q}(S,e) \end{aligned}$$

By multiplying the first set of inequalities by $x^*(e)$, and letting $f(e,Q) = x^*(e)x(e,Q)$, this gives the following flow problem.

$$\begin{aligned} \sum_{Q \in \mathcal{Q}(S,e)} f(e,Q) &\leq x^*(e) \qquad \text{for all } e \in E \\ \sum_{e \in \mathcal{Q}(S)} f(e,Q) &\geq 1 \qquad \text{for all } Q \in \mathcal{Q}(S) \\ f(e,Q) &\geq 0 \qquad \text{for all } e \in E, Q \in \mathcal{Q}(S,e) \end{aligned}$$

A solution to this flow problem exists if and only for any $\mathcal{Q}' \subseteq \mathcal{Q}$

$$x^*(\cup_{Q \in \mathcal{Q}'} Q) \geq |\mathcal{Q}'|.$$

Indeed, the necessity and sufficiency of this condition follows from the max flow min cut theorem (it is actually just a variant of the Kőnig-Hall theorem), and to prove that the condition is satisfied denote $k := |\mathcal{Q}'|$, and let $C_0, \ldots, C_k$ the vertex sets of components of $S \setminus \cup_{Q \in \mathcal{Q}'} e_S^Q$, where $s \in C_0, t \in C_k$. Since $\{C_0, \ldots, C_k\}$ is a partition of V, every edge is counted twice in the sum of their coboundaries:

$$\begin{aligned} x^*(\cup_{Q \in \mathcal{Q}'} Q) &= \frac{1}{2} \sum_{j=0}^{k} x^*(\delta(C_i)) \\ &\geq \frac{1}{2} \left(x^*(\delta(C_0)) + \sum_{j=1}^{k-1} x^*(\delta(C_j)) + x^*(\delta(C_k)) \right) \\ &\geq \frac{1}{2}(1 + 2(k-1) + 1) = k. \end{aligned}$$

exactly as needed. □

E. Balancing

To bound the average cost of the forest-based $\{s,t\}$-tours, we now simply add up the average cost of the forest plus parity correction (consisting of basic parity correction and parity completion) plus reconnection. By setting the parameter γ in the basic parity correction vector appropriately, we show that the average cost of the forest-based $\{s,t\}$-tours is expressable in terms of x^* and p^* only. Finally, we take the minimum of the tree-based $\{s,t\}$-tours (which, by Observation 3, has cost decreasing in $c(p^*)$) and the cost of the forest-based $\{s,t\}$-tours is increasing in $c(p^*)$. The worst case for the minimum of the two is when the two are *balanced*, that is equal.

IV. The New Ratio

In this section we prove the main result.

Theorem 6: The Best-of-Many With Deletion (BOMD) algorithm returns a solution to the $s-t$ path TSP of cost at most $\left(\frac{3}{2}+\frac{1}{34}\right) OPT_{LP}$.

Proof: We analyze the different parts of the forest-based $\{s,t\}$-tour $P_1(S)$ averaged over $S \in \mathcal{S}$:

Forest: Let $\mathcal{F} := \mathcal{S} \setminus L(\mathcal{S})$.
Expected Value of Forest : $\mathbb{E}[\mathcal{F}] = x^* - \sum_{Q\in\mathcal{Q}} x^Q$.

Parity Correction (PC): For each forest F, we define y_F as the sum of the following vectors:

Basic Parity Correction (BP): $\frac{1}{2}x^* + \gamma\, S(s,t)$.
Note that (BP) is enough to ensure that y_F satisfies the constraints of the $T_F \triangle \{s,t\}$-join polyhedron for all $Q=\delta(U)$ with $x^*(Q) \geq 2-2\gamma$.
Expected Value of (BP): $\dfrac{1}{2}x^* + \gamma\, p^*$.

Parity Completion for Empty Cuts (PCE): $\left(1-\dfrac{1}{2}x^*(Q)-\gamma\right) e_S^Q$.
We recall that e_S^Q is the (indicator vector of) the lonely edge of S in Q. We add this (PCE) vector to y_F for every $Q \in \mathcal{Q}(S)$ such that $x^*(Q) \leq 2-2\gamma$. Note that then (BP)+(PCE) suffices to ensure that y_F satisfies the constraint of the $T_F \triangle \{s,t\}$-join polyhedron for each narrow cut Q such that $Q \cap F = \emptyset$. Since $\mathbb{E}\left[e_{\mathcal{S}}^Q\right] = x^Q$, we have:
Expected Value of (PCE):
$\sum_{Q\in\mathcal{Q}:\, x^*(Q)\leq 2-2\gamma}\left(1-\frac{1}{2}x^*(Q)-\gamma\right)x^Q$.

Parity Completion for Large (Non-Empty Even) Cuts (PCL): $\left(\dfrac{1-\frac{1}{2}x^*(Q)-\gamma}{2-x^*(Q)}\right)x^Q$.
We add this vector to y_F for every Q such that $x^*(Q) \leq 2-2\gamma$ and $|F\cap Q|$ is even and at least 2. By (7), y_F satisfies then the constraints of the $T_F \triangle \{s,t\}$-join polyhedron also for such cuts. Contrary to (PCE), (PCL) does not depend on (the outcome of the random variable) $\mathcal{S}$, but only on the cut $Q \in \mathcal{Q}$. However, the choice of adding it or not, depends on $\mathcal{S}$: we add it only if $|\mathcal{F}\cap Q|$ is even and at least two, the probability of which can be bounded by $\Pr(|\mathcal{S}\cap Q|\geq 2) \leq 1-(2-x^*(Q))$ by (7), so we add (PCL) with probability at most $x^*(Q)-1$:
Expected Value of (PCL):
$\sum_{Q\in\mathcal{Q}:x^*(Q)\leq 2-2\gamma}(x^*(Q)-1)\left(\frac{1-\frac{1}{2}x^*(Q)-\gamma}{2-x^*(Q)}\right)x^Q$.

The combination (BP) + (PCE) + (PCL) gives a fractional solution to the $T_F \triangle \{s,t\}$-join polyhedron for any fixed tree S and its uniquely determined forest $F = S \setminus L(S)$:

$$\begin{aligned} y_F &= \frac{1}{2}x^* + \gamma\, S(s,t) \\ &\quad + \sum_{\substack{Q\in\mathcal{Q}:x^*(Q)\leq 2-2\gamma \\ |Q\cap F|=0}} \left(1-\frac{1}{2}x^*(Q)-\gamma\right)e_S^Q \\ &\quad + \sum_{\substack{Q\in\mathcal{Q}:x^*(Q)\leq 2-2\gamma \\ |Q\cap F|\geq 2,\text{even}}} \left(\frac{1-\frac{1}{2}x^*(Q)-\gamma}{2-x^*(Q)}\right)x^Q. \end{aligned}$$

Reconnection: Given $S \in \mathcal{S}$, define the random $T_F \triangle \{s,t\}$-join $\mathcal{J}_F$ with $\mathbb{E}[\mathcal{J}_F] = y_F$. As explained in Section II, $\mathbb{E}\left[c(2D_{F+\mathcal{J}_F})\right] \leq c_F(y_F) - c(y_F)$ (where F is fixed and the expectation is over $\mathcal{J}_F$).

Recall that $B(S)$ is the set of edges that are contained in more than one cut in $\mathcal{Q}(S)$, and that $c_F(e)-c(e)=0$ for $e \notin B(S)$. We claim that $y_F(e) = x^*(e)/2$ for $e \in B(S)$: $\gamma S(s,t)$ and (PCE) contain only edges from the $S(s,t)$ and thus contain no bad edges for S, since they are by definition contained in at most one cut of $\mathcal{Q}(S)$, while the edges in $B(S)$ are those contained in at least two such cuts. The fact that this also holds for (PCL) is exactly the conclusion of Lemma 4.

Hence, we have $c_F(y_F) - c(y_F) = c(x^*/2) - c(x^*/2)$, which is bounded by $\sum_{Q\in\mathcal{Q}(S)}(x^*(Q)-1)c(e_S^Q)$ by Lemma 5. By taking the expectation over $S \in \mathcal{S}$ and noting that $\mathbb{E}\left[e_{\mathcal{S}}^Q\right] = x^Q$, we get the following bound.
Expected Value of Reconnection Cost:
$\mathbb{E}\left[c(2D_{\mathcal{F}+\mathcal{J}_\mathcal{F}})\right] \leq \sum_{Q\in\mathcal{Q}}(x^*(Q)-1)c(x^Q)$.

TOTAL: If we add up the costs of the bounds on the different parts of the solution, we get $\frac{3}{2}c(x^*)+\gamma c(p^*)$ plus the sum over all $Q \in \mathcal{Q}$ of some multiple (that depends on $x^*(Q)$) of $c(x^Q)$. In particular, if $2-2\gamma < x^*(Q) < 2$, the multiplier of $c(x^Q)$ is $-1+x^*(Q)-1 = x^*(Q)-2 \leq 0$, and if $x^*(Q) \leq 2-2\gamma$ it is

$$\begin{aligned} &-1+\left(1-\frac{1}{2}x^*(Q)-\gamma\right)\left(1+\frac{x^*(Q)-1}{2-x^*(Q)}\right)+ \\ &+x^*(Q)-1 = \frac{2-x^*(Q)-2\gamma}{2(2-x^*(Q))} + x^*(Q)-2. \end{aligned}$$

We choose γ so that this multiplier is nonpositive for all $Q \in \mathcal{Q}$, that is, we want

$$2-x^*(Q)-2\gamma+2(2-x^*(Q))(x^*(Q)-2) \leq 0$$

that is,

$$(2 - x^*(Q) - \frac{1}{4})^2 + \gamma - \frac{1}{16} \geq 0.$$

The minimum of γ for which this is satisfied for all $1 \leq x^*(Q) \leq 2$ is $\gamma = \frac{1}{16}$, where we note that equality holds if and only if $x^*(Q) = \frac{7}{4}$.

We have thus bounded the expectation of $c(P_1(\mathcal{S}))$ by $\frac{3}{2}c(x^*) + \frac{1}{16}c(p^*)$. By Observation 3, the expectation of $c(P_2(\mathcal{S}))$ is at most $2c(x^*) - c(p^*)$, and therefore the cost of the solution returned by (BOMD) is at most

$$\min\left\{\frac{3}{2}c(x^*) + \frac{1}{16}c(p^*), 2c(x^*) - c(p^*)\right\}$$
$$= \left(\frac{3}{2} + \frac{1}{34}\right)c(x^*).$$

□

V. Tight Bound for Special Cases

We show some simple cases where the upper bound of $3/2$ on the approximation ratio and integrality gap can be proved.

A. All Narrow Cuts Are Small

If $x^*(Q) \leq \frac{3}{2}$ for all $Q \in \mathcal{Q}$, then choosing $\gamma = 0$ suffices to guarantee that $(2 - x^*(Q) - \frac{1}{4})^2 + \gamma - \frac{1}{16} \geq 0$. Hence, by the proof of Theorem 6, our algorithm achieves the bound of $\frac{3}{2}OPT_{LP}$ for any instance of the $s-t$ path TSP in which the optimal subtour LP solution has no cuts Q with $x^*(Q) \in (\frac{3}{2}, 2)$.

B. Disjoint and Almost-Disjoint Narrow Cuts

Let $G = (V, E)$ be a graph and denote $\mathbf{1}$ be the all-1 vector on the edges. A naturally arising open question relaxing the integrality gap of the subtour elimination LP is to determine the smallest constant α so that $\alpha\mathbf{1}$ a convex combination of T-tours[2] [16]. In this case, we will say that $\alpha\mathbf{1}$ is *feasible*.

When $T = \{s, t\}$, the conjecture that the integrality gap of the subtour elimination LP for the $s-t$ path TSP is $3/2$ implies that $\mathbf{1}$ is feasible for graphs in which all non s-t cuts are of size at least 3: Indeed, the conjecture that the integrality gap of the subtour elimination LP is $3/2$ for s-t path TSP is well-known to be equivalent to the feasibility of $\frac{3}{2}x$ provided x is in the subtour polytope (for the LP with degree lower bound constraints rather than degree equality constraints). Since setting $2/3$ on all the edges is in the subtour polytope, if this conjecture is true, $\alpha = \frac{3}{2} \cdot \frac{2}{3} = 1$ gives a feasible vector for $T = \{s, t\}$. Our techniques allow us to prove that this is indeed true.

Theorem 7: Let $G = (V, E)$ be a connected graph, $s, t \in V$ and let $|\delta(U)| \geq 3$ for any $|U \cap \{s, t\}| \neq 1$. Then, $\mathbf{1}$ can be expressed as a convex combination of $\{s, t\}$-tours.

[2] A T-tour is a connected graph in which the set of odd degree vertices is T.

Proof: (Sketch) We can suppose that G is 2-edge-connected, since a cut-edge necessarily separates s and t, and we can proceed by induction, separating the problem into two. Then $x^*(e) = \frac{2}{3}$ on every $e \in E$ satisfies the subtour elimination constraints of the LP for the $s-t$ path TSP, and the narrow cuts are exactly the $s-t$ cuts of G containing two edges. Now, the fact that the non-$s-t$-cuts of G have at least three edges implies that *the narrow cuts are pairwise disjoint*:

Indeed, suppose there exist narrow cuts $Q_1 = \delta(U_1), Q_2 = \delta(U_2)$ (each containing exactly two edges), $s \in U_1$, $t \in U_2$ and $Q_1 \cap Q_2 \neq \emptyset$. Then the left hand side of (1) is $2 \cdot \frac{2}{3} + 2 \cdot \frac{2}{3}$, and the first term of the right hand side is $3 \cdot \frac{2}{3} = 2$ by the condition of the theorem; the last term on the right hand side is at least $2 \cdot \frac{2}{3}$ because a common edge, that exists by the non-emptiness asumption, has one endpoint in $U_1 \setminus U_2$ and the other in $U_2 \setminus U_1$. Since $\frac{8}{3} < \frac{10}{3}$ this is a contradiction.

This observation allows us to simplify the BOMD algorithm, since there are no bad edges: such an edge would be in the intersection of two narrow cuts. Rather than requiring a layered convex combination, we allow now an arbitrary decomposition of x^* into spanning trees $S \in \mathcal{S}$, and define $L(S)$ to be simply the set of edges $e \in S$ such that there exists a narrow cut in which e is the unique edge of S.

Furthermore, in the analysis, we take y_F to be the basic parity correction vector with $\gamma = 0$, and we add as parity completion $x^Q/2$ for every narrow cut Q such that $|Q \cap F|$ is even. As observed in Section III-C, this indeed gives a feasible parity correction vector, with an expected value of at most $\frac{1}{2}x^* + \frac{1}{2}\sum_{Q \in \mathcal{Q}} x^Q$. Adding this to the expected value of the forest $x^* - \sum_{Q \in \mathcal{Q}} x^Q$ and noting that no reconnection is needed shows that $\frac{3}{2}x^* - \frac{1}{2}\sum_{Q \in \mathcal{Q}} x^Q \leq \mathbf{1}$ is already in the convex hull of $\{s, t\}$-tours. □

We actually proved that $\frac{3}{2}x^* - \frac{1}{2}\sum_{Q \in \mathcal{Q}} x^Q$ *is a convex combination of* $\{s, t\}$ *tours, under the more general condition that the narrow cuts are disjoint.* This can actually be strengthened as follows.

Theorem 8: Let x^* be a feasible solution to the subtour elimination LP for the $s-t$ path TSP, and let $G = (V, E)$ be the support graph of x^*. If no edge in G is contained in more than two narrow cuts, then $\frac{3}{2}x^*$ is a convex combination of $\{s, t\}$-tours.

Proof: (Sketch) By Lemma 1, we know that there exist $s \subseteq U_1 \subset U_2 \subset U_2 \subset \ldots U_k$ such that the narrow cuts are exactly $\delta(U_i)$ for $i = 1, \ldots, k$. We will call a narrow cut $Q = \delta(U_i)$ odd-numbered or even-numbered, depending on whether i is odd or even. As in the proof of Theorem 7, we take an arbitrary decomposition of x^* into spanning trees and call an edge lonely in a tree if it is the unique edge in some narrow cut. However, we will now only remove lonely edges from either the odd-numbered narrow cuts, or the even-numbered narrow cuts, each with probability $\frac{1}{2}$. This changes the expectation of the forest to $x^* - \frac{1}{2}\sum_{Q\in\mathcal{Q}} x^Q$.

As in the proof of Theorem 7, we can upper bound the expectation of the parity correction vector by $\frac{1}{2}x^* + \frac{1}{2}\sum_{Q\in\mathcal{Q}} x^Q$, and we are again guaranteed that the reconnection cost is 0, since a bad edge can be contained in only two consecutive narrow cuts, and we never delete the lonely edges of both of these. The sum of the two nonzero contributions is exactly $\frac{3}{2}x^*$. $\square$

ACKNOWLEDGMENTS

This research was initiated during the Hausdorff Trimester Program in Combinatorial Optimization at the Hausdorff Institute of Mathematics, Bonn, and was partially supported by LabEx PERSYVAL-Lab (ANR 11-LABX-0025) and by a grant from the Simons Foundation (#359525, Anke Van Zuylen).

REFERENCES

[1] H. An, R. Kleinberg, and D. B. Shmoys. Improving Christofides' algorithm for the s-t path TSP. In *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012, New York, NY, USA, May 19 - 22, 2012*, pages 875–886, 2012. See [2] for journal version.

[2] H. An, R. D. Kleinberg, and D. B. Shmoys. Improving Christofides' algorithm for the s-t path TSP. *J. ACM*, 62(5):34, 2015.

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

[4] W. J. Cook. *In Pursuit of the Traveling Salesman: Mathematics at the Limits of Computation*. Princeton University Press, December 2011.

[5] G. Dantzig, R. Fulkerson, and S. Johnson. Solution of a large-scale traveling-salesman problem. *Operations Research*, 2:393–410, 1954.

[6] J. Edmonds. Submodular functions, matroids, and certain polyhedra. In *Combinatorial Structures and their Applications (Proc. Calgary Internat. Conf., Calgary, Alta., 1969)*, pages 69–87. Gordon and Breach, New York, 1970.

[7] J. Edmonds and E. L. Johnson. Matching, euler tours and the chinese postman. *Math. Program.*, 5(1):88–124, 1973.

[8] Z. Gao. An LP-based 3/2-approximation algorithm for the *s-t* path graph Traveling Salesman Problem. *Oper. Res. Lett.*, 41(6):615–617, 2013.

[9] Z. Gao. On the metric s-t path traveling salesman problem. *SIAM J. Discrete Math.*, 29(3):1133–1149, 2015.

[10] M. R. Garey, D. S. Johnson, and R. E. Tarjan. The planar hamiltonian circuit problem is NP-complete. *SIAM Journal on Computing*, 5(4):704–714, 1976.

[11] C. Gottschalk and J. Vygen. Better s-t-tours by Gao trees. In *Proceedings of the 18th International Conference on Integer Programming and Combinatorial Optimization (IPCO)*, pages 126–137, 2016.

[12] J. Hoogeveen. Analysis of Christofides' heuristic: Some paths are more difficult than cycles. *Operations Research Letters*, 10(5):291 – 295, 1991.

[13] D. S. Johnson and C. H. Papadimitriou. Performance guarantees for heuristics. In E. Lawler, J. Lenstra, A. R. Kan, and D. Shmoys, editors, *The Traveling Salesman Problem – A Guided Tour of Combinatorial Optimization*, pages 145–180. Wiley, Chichester, 1985.

[14] T. Mömke and O. Svensson. Approximating graphic TSP by matchings. pages 560–569, 2011. See [15] for journal version.

[15] T. Mömke and O. Svensson. Removing and adding edges for the traveling salesman problem. *J. ACM*, 63(1):2, 2016.

[16] A. Sebő. Problems about uniform covers, with tours and detours. *Oberwolfach Report*, 51/2014.

[17] A. Sebő. Eight-fifth approximation for the path TSP. In *Proceedings of the 16th International Conference on Integer Programming and Combinatorial Optimization (IPCO)*, pages 362–374, 2013.

[18] A. Sebő and J. Vygen. Shorter tours by nicer ears: 7/5-approximation for the graph-TSP, 3/2 for the path version, and 4/3 for two-edge-connected subgraphs. *Combinatorica*, 34(5):597–629, 2014.

[19] J. Vygen. Reassembling trees for the traveling salesman. *SIAM J. Discrete Math.*, 30(2):875–894, 2016.

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