A Proof of the Boyd-Carr Conjecture

Frans Schalekamp David P. Williamson* Anke van Zuylen†

Abstract

Determining the precise integrality gap for the subtour LP relaxation of the traveling salesman problem is a significant open question, with little progress made in thirty years in the general case of symmetric costs that obey triangle inequality. Boyd and Carr [3] observe that we do not even know the worst-case upper bound on the ratio of the optimal 2-matching to the subtour LP; they conjecture the ratio is at most 10/9.

In this paper, we prove the Boyd-Carr conjecture. In the case that a fractional 2-matching has no cut edge, we can further prove that an optimal 2-matching is at most 10/9 times the cost of the fractional 2-matching.

1 Introduction

The traveling salesman problem (TSP) is the most famous problem in discrete optimization. Given a set of n cities and the costs $c(i,j)$ of traveling from city i to city j for all i,j, the goal of the problem is to find the least expensive tour that visits each city exactly once and returns to its starting point. An instance of the TSP is called *symmetric* if $c(i,j) = c(j,i)$ for all i,j; it is *asymmetric* otherwise. Costs obey the *triangle inequality* if $c(i,j) \leq c(i,k) + c(k,j)$ for all i,j,k. The TSP is known to be NP-hard, even in the case that instances are symmetric and obey the triangle inequality. From now on we consider only these instances unless otherwise stated.

Because of the NP-hardness of the traveling salesman problem, researchers have considered approximation algorithms for the problem. The best approximation algorithm currently known is a $\frac{3}{2}$-approximation algorithm given by Christofides in 1976 [7]. Better approximation algorithms are known for special cases. Exciting progress has been made recently in the case of graph-TSP, in which costs $c(i,j)$ are given by shortest path distances in an unweighted graph; we will discuss these results shortly. However, to date, Christofides' algorithm has the best known performance guarantee for the general case.

There is a well-known, natural direction for making progress which has also defied improvement for nearly thirty years. The following linear programming relaxation of the traveling salesman problem was used by Dantzig, Fulkerson, and Johnson [8] in 1954. For simplicity of notation, we let $G = (V,E)$ be a complete undirected graph on n nodes. In the LP relaxation, we have a variable $x(e)$ for all $e = (i,j)$ that denotes whether we travel directly between cities i and j on our tour. Let $c(e) = c(i,j)$, and let $\delta(S)$ denote the set of all edges with exactly one endpoint in $S \subseteq V$. Then the relaxation is

$$\text{Min} \quad \sum_{e \in E} c(e)x(e)$$

subject to:

$$(1.1) \quad \sum_{e \in \delta(i)} x(e) = 2, \quad \forall i \in V, \qquad (SUBT)$$

$$(1.2) \quad \sum_{e \in \delta(S)} x(e) \geq 2, \quad \forall S \subset V,\ 3 \leq |S| \leq |V|-3,$$

$$(1.3) \quad 0 \leq x(e) \leq 1, \quad \forall e \in E.$$

The first set of constraints (1.1) are called the *degree constraints*. The second set of constraints (1.2) are sometimes called *subtour elimination constraints* or sometimes just *subtour constraints*, since they prevent solutions in which there is a subtour of just the nodes in S. As a result, the linear program is sometimes called the *subtour LP*. It is known that the equality sign in the first set of constraints may be replaced by $\geq$ in case the costs obey the triangle inequality (Goemans and Bertsimas [12]; see also Williamson [21]).

The LP is known to give excellent lower bounds on TSP instances in practice, coming within a percent or two of the length of the optimal tour (see, for instance, Johnson and McGeoch [13]). However, its theoretical worst-case is not well understood. In 1980, Wolsey [22] showed that Christofides' algorithm produces a solution whose value is at most $\frac{3}{2}$ times

*Address: School of Operations Research and Information Engineering, Cornell University, Ithaca, NY 14853, USA. Email: dpw@cs.cornell.edu. This work was carried out while the author was on sabbatical at TU Berlin. Supported in part by the Berlin Mathematical School, the Alexander von Humboldt Foundation, and NSF grant CCF-1115256.

†Address: Max-Planck-Institut für Informatik, Department 1: Algorithms and Complexity, Campus E1 4, Room 311c, 66123 Saarbrücken, Germany. Email: anke@mpi-inf.mpg.de.

the value of the subtour LP (also shown later by Shmoys and Williamson [20]). This proves that the *integrality gap* of the subtour LP is at most $\frac{3}{2}$; the integrality gap is the worst-case ratio, taken over all instances of the problem, of the value of the optimal tour to the value of the subtour LP, or the ratio of the optimal integer solution to the optimal fractional solution. The integrality gap of the LP is known to be at least $\frac{4}{3}$ via a specific class of instances. However, no instance is known that has integrality gap worse than this, and it has been conjectured for some time that the integrality gap is at most $\frac{4}{3}$ (see, for instance, Goemans [11]).

Stronger bounds on the integrality gap are known in the case of graph-TSP. Oveis Gharan, Saberi, and Singh [17] show that graph-TSP can be approximated to within $\frac{3}{2} - \epsilon$ for a small constant $\epsilon > 0$, and this implies a bound on the integrality gap of $\frac{3}{2} - \epsilon$ for such instances as well. Mömke and Svensson [14] show that the integrality gap is at most 1.461. Mucha [15] improved this result to $\frac{35}{24}$. If the graph is cubic, Boyd, Sitters, van der Ster, and Stougie [6] show that the gap is $\frac{4}{3}$, and Mömke and Svensson extend the bound of $\frac{4}{3}$ to subcubic graphs as well.

There is some evidence that the conjectured gap of $\frac{4}{3}$ might be true. Benoit and Boyd [2] have shown via computational methods that the conjecture holds for $n \leq 10$, and Boyd and Elliot-Magwood [5] have extended this to $n \leq 12$. In a 1995 paper, Goemans [11] showed that adding any class of valid inequalities known at the time to the subtour LP could increase the value of the LP by at most $\frac{4}{3}$; this is necessary for the conjecture to be true. Somewhat weaker evidence is as follows. A *2-matching* is an integer solution to the subtour LP obeying only the degree constraints (1.1) and the bounds constraints (1.3).[1] A *fractional 2-matching* is a 2-matching without the integrality constraints. Boyd and Carr [4] have shown that the integrality gap for the 2-matching problem is at most $\frac{4}{3}$. Furthermore, Boyd and Carr [3] have shown that if the subtour LP solution is half-integral (that is, $x(i,j) \in \{0, \frac{1}{2}, 1\}$ for all $i, j \in V$) and has a particular structure then there is a tour of cost at most $\frac{4}{3}$ times the value of the subtour LP.

Not only do we not know the integrality gap of the subtour LP, Boyd and Carr have observed that we don't even know the worst-case ratio of the optimal 2-matching to the value of the subtour LP, which is surprising because 2-matchings are well understood and well characterized. They make the following conjecture.

CONJECTURE 1. (BOYD AND CARR [3]) *The worst-case ratio of an optimal 2-matching to an optimal solution to the subtour LP is at most* $\frac{10}{9}$.

It is known that there are cases for which the cost of an optimal 2-matching is at least $\frac{10}{9}$ times the optimal solution to the subtour LP; see Figure 1. Boyd and Carr have shown that the conjecture is true if the solution to the subtour LP has a very special structure: namely, all variables $x(e) \in \{0, \frac{1}{2}, 1\}$, the cycles formed by the edges e with $x(e) = \frac{1}{2}$ all have the same odd size k, and the support is $(k-1)$-edge-connected.[2] In the general case, the only bound on this ratio that we know of is the Boyd and Carr bound on the integrality gap of 2-matchings; since the constraints of the subtour LP are a superset of the fractional 2-matching constraints, this implies the ratio is at most $\frac{4}{3}$.

The work of Goemans [11] has some bearing on this conjecture. He studies the following linear program which is essentially same as the subtour LP in the case edge costs obey triangle inequality:

$$\text{Min} \quad \sum_{e \in E} c(e)x(e)$$

subject to:

$$(1.4) \qquad \sum_{e \in \delta(S)} x(e) \geq 2, \quad \forall S \subset V, S \neq \emptyset, \quad (SUBT')$$

$$(1.5) \qquad x(e) \geq 0, \qquad \forall e \in E.$$

Goemans shows (among other things) that adding comb inequalities to this LP can increase the LP value by at most $\frac{10}{9}$; more precisely, he shows that if x is a feasible solution to $(SUBT')$, then $\frac{10}{9}x$ is feasible for the LP obtained by adding comb inequalities to $(SUBT')$. It is known that adding a subset of the comb inequalities to the degree constraints (1.1) and bounds (1.3) gives the 2-matching polytope. This would imply the Boyd-Carr conjecture if it were known that there is an optimal solution that obeys the degree constraints when the comb inequalities are added to $(SUBT')$; as mentioned above, it can be shown that there is an optimal solution for $(SUBT')$ that obeys the degree constraints when the edge costs obey the triangle inequality. But we do not know whether there is an optimal solution that obeys the degree constraints if the comb inequalities are added.[3]

[1] We note that what we refer to here as 2-matchings, are also sometimes called 2-factors.

[2] In fact, they show in this case the optimal 2-matching has cost at most $\frac{3k+1}{3k}$ times the subtour LP.

[3] To quote Goemans [11, p. 348]: "One might wonder whether the worst-case improvements remain unchanged when one adds the degree constraints $x(\delta\{i\}) = 2$ for all $i \in V$ and restricts one's attention to cost functions satisfying the triangle inequality. We believe so but have been unable to prove it.

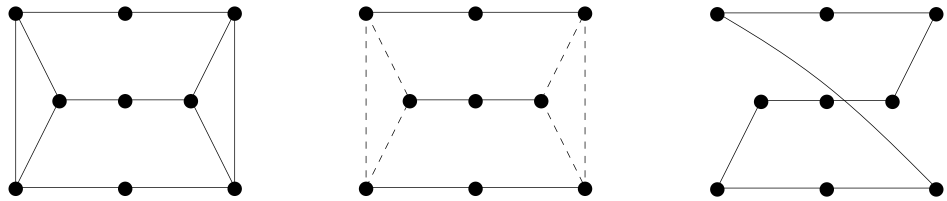

Figure 1: Illustration of the worst example known for the ratio of 2-matchings to the subtour LP. The figure on the left shows the instance; all edges in the graph have cost 1, all other edges have cost 2. The figure in the center gives the subtour LP solution, in which the dotted edges have value $\frac{1}{2}$, and the solid edges have value 1; this is also an optimal fractional 2-matching. The figure on the right gives an optimal 2-matching, which is also the optimal tour.

The contribution of this paper is to improve our state of knowledge for the subtour LP by proving Conjecture 1.

We start by showing that in some cases the cost of an optimal 2-matching is at most $\frac{10}{9}$ the cost of a fractional 2-matching, which is a stronger statement than Conjecture 1; in particular, we show this is true whenever the support of the fractional 2-matching has no cut edge. The example in Figure 1 shows that the ratio can be at least $\frac{10}{9}$ in such cases, so this result is tight. As the first step in this proof, we give a simplification of the Boyd and Carr result bounding the integrality gap for 2-matchings by $\frac{4}{3}$. In the case that the support of an optimal fractional 2-matching has no cut edge, the proof becomes quite simple. The perfect matching polytope plays a crucial role in the proof: we use the matching edges to show us which edges to remove from the solution in addition to showing us which edges to add. We note that this idea was independently developed in the recent work of Mömke and Svensson, but also previously appeared in the reduction of the 2-matching polytope to the matching polytope; see, for instance, Schrijver [19, Section 30.7]. We also use a notion from Boyd and Carr [4] of a *graphical* 2-matching: in a graphical 2-matching, each node has degree either 2 or 4, each edge has 0, 1, or 2 copies, and each component has size at least three. Given the triangle inequality, we can shortcut any graphical 2-matching to a 2-matching of no greater cost.

To obtain our proof of the Boyd-Carr conjecture, we give a polyhedral formulation of the graphical 2-matching problem, and use it to prove Conjecture 1. If x is a feasible solution for the subtour LP, then, roughly speaking, we show that $\frac{10}{9}x$ is feasible for the graphical 2-matching polytope. Our previous results give us intuition for the precise mapping of variables that we need. Using the graphical 2-matching polytope allows us to overcome the issues with the degree constraints faced in trying to use Goemans' results.

All the results above can be made algorithmic and have polynomial-time algorithms, though we do not explicitly determine running times.

We conclude by posing a new conjecture, namely that the worst-case integrality gap is achieved for solutions to the subtour LP that are fractional 2-matchings (that is, for instances such that adding the subtour constraints to the degree constraints and the bounds on the variables does not change the objective function value).

In a companion paper, Qian, Schalekamp, Williamson, and van Zuylen [18] show that the proof of the Boyd-Carr conjecture can be used to help bound the integrality gap of the subtour LP for the 1,2-TSP. They show that the gap is at most $\frac{106}{81} \approx 1.3086 < \frac{4}{3}$. They also give a proof that the cost of the optimal 2-matching is at most $\frac{10}{9}$ times the cost of a fractional 2-matching in the case that $c(i,j) \in \{1,2\}$, which gives an alternate proof of the Boyd-Carr conjecture in this case.

Our paper is structured as follows. We introduce basic terms and notation in Section 2. In Section 3, we rederive the Boyd-Carr integrality gap for 2-matchings, and show that the gap is at most $\frac{10}{9}$ in the case the fractional 2-matching has no cut edge. In Section 4, we give the polytope for graphical 2-matchings and show how to use it to prove the Boyd-Carr conjecture. Finally, we close with our new conjecture in Section 5. Some proofs are omitted from this abstract; a full version of the paper can be found at `http://arxiv.org/abs/1107.1628`.

The result would follow immediately if one could prove that the degree constraints never affect the value of the relaxation when the cost function satisfies the triangle inequality."

2 Preliminaries

We will work extensively with fractional 2-matchings; that is, optimal solutions x to the LP ($SUBT$) with only constraints (1.1) and (1.3). For convenience we will abbreviate "fractional 2-matching" by F2M and "2-matching" by 2M. F2Ms have the following well-known structure (attributed to Balinski [1]). Each connected component of the support graph (that is, the edges e for which $x(e) > 0$) is either a cycle on at least three nodes with $x(e) = 1$ for all edges e in the cycle, or consists of odd-sized cycles with $x(e) = \frac{1}{2}$ for all edges e in the cycle connected by paths of edges e with $x(e) = 1$ for each edge e in the path (the center figure in Figure 1 is an example). We call the former components *integer components* and the latter *fractional components*. Many of our results focus on transforming an F2M into a 2M, in which all components are integer. For that reason, we will often focus solely on how to transform the fractional components into integer components. We then call the edges of fractional components for which $x(e) = \frac{1}{2}$ *cycle edges* and the edges for which $x(e) = 1$ *path edges*. Note that removing a cycle edge can never disconnect a fractional component. If removing a path edge disconnects a fractional component, we call it a *cut edge*. The associated path of the path edge we will call a *cut path*, since every edge in it will be a cut edge. We will say that a fractional 2-matching is *connected* if it has a single component.

We will use a concept introduced by Boyd and Carr [4] of a *graphical* 2-matching (G2M). As stated above, in a graphical 2-matching, each node has degree either 2 or 4, each edge has 0, 1, or 2 copies, and each component has size at least three. Given the triangle inequality, we can shortcut any G2M to a 2M of no greater cost. Our techniques for transforming an F2M to a 2M actually find G2Ms.

We will often need to find minimum-cost perfect matchings. By a result of Edmonds [9], the perfect matching polytope is defined by the following linear program (M):

$$\text{Min} \quad \sum_{e \in E} c(e)x(e)$$

subject to:

$$(2.6) \qquad \sum_{e \in \delta(i)} x(e) = 1, \quad \forall i \in V, \qquad\qquad (M)$$

$$(2.7) \qquad \sum_{e \in \delta(S)} x(e) \geq 1, \quad \forall S \subset V,\ |S| \text{ odd},$$

$$(2.8) \qquad x(e) \geq 0, \qquad \forall e \in E.$$

3 2-matching Integrality Gaps

In this section, we bound the cost of a G2M in terms of an F2M via combinatorial methods. We start by giving a proof of a result of Boyd and Carr [4] that there is a G2M of cost at most $\frac{4}{3}$ the cost of an F2M. Our proof is somewhat simpler than theirs, but more importantly, it introduces the main ideas that we will need to obtain other results. We then show that if the F2M has no cut edges, we can improve the bound from $\frac{4}{3}$ to $\frac{10}{9}$. The main idea of this section is that given an F2M, we define a matching problem and compute a perfect matching. The perfect matching tells us how to modify the fractional components by either duplicating or removing edges so that we obtain a G2M. We then relate the cost of the perfect matching found to the F2M by providing a feasible solution to the perfect matching LP (M). We will need the following result of Naddef and Pulleyblank [16] .

Lemma 3.1. (Naddef and Pulleyblank [16]) *Let G be a cubic, 2-edge-connected graph with edge costs $c(e)$ for all $e \in E$. Then there exists a perfect matching in G of cost at most $\frac{1}{3}\sum_{e \in E} c(e)$.*

We note that Lemma 3.1 also holds for cubic, 2-edge-connected multigraphs.

Theorem 3.1. *There exists a G2M of cost at most $\frac{4}{3}$ times the cost of an F2M if the F2M has no cut edge.*

Proof. As described above, it is sufficient to focus on a single fractional component of the F2M. Let G be the support graph of this component.

To find the G2M, we find a minimum-cost perfect matching on the (multi)graph G' we obtain by replacing each path in G by a single edge, which we will call (at the risk of some confusion) a path edge. We set the cost of this edge to be the cost of the path in G, and we set the cost of a cycle edge in G' to the *negative* of the cost of the cycle edge in G. Note that G' is cubic and 2-edge-connected because the support graph G of the F2M has no cut edge.

Given a minimum-cost perfect matching in G', we construct a G2M in G by first including all paths from G. If a path edge is in the matching in G', we double the path in G. If a cycle edge is *not* in the matching in G', then we include the cycle edge in the G2M in G, otherwise we omit the cycle edge.

We first show that this indeed defines a G2M: for each node, the degree is four if the perfect matching contains the path edge incident to the node (since in that case, the two cycle edges on the node cannot be in the perfect matching, and hence both are added to the G2M together with two copies of the path), and it is two otherwise (since one cycle edge is in the perfect matching and hence only the other cycle edge and one copy of the path are added to the graphical 2-matching). Note that any connected component indeed has at least three nodes: Every connected component contains (the edges corresponding to) at least one path edge; now, either a path edge corresponds to a path of length at least 2, or the path edge corresponds to a single edge, but then at least two cycle edges incident to the endpoints of the path edge are also contained in the connected component, and these cycle edges are distinct from the edge that is the path, since G has no doubled edges.

We let C denote the sum of the costs of the cycle edges, and P the cost of the paths. Note that the cost of the F2M solution is $\frac{1}{2}C + P$. The cost of the G2M is equal to the cost of all edges in the support graph $(P + C)$ plus the cost of the perfect matching. Because G' is cubic and 2-edge-connected, we can invoke Lemma 3.1 to show that the perfect matching has cost at most a third the cost of the edges in G', or at most $\frac{1}{3}P - \frac{1}{3}C$. Hence the cost of the G2M is at most

$$P + C + \frac{1}{3}P - \frac{1}{3}C = \frac{4}{3}P + \frac{2}{3}C = \frac{4}{3}\left(P + \frac{1}{2}C\right),$$

or at most $\frac{4}{3}$ the cost of the F2M solution, as claimed. ■

We now modify the proof of the theorem above so that the result extends to the case in which the F2M has cut edges.

THEOREM 3.2. (BOYD AND CARR [4]) *There exists a G2M of cost at most $\frac{4}{3}$ times the cost of an F2M.*

Proof. As described above, it is sufficient to focus on a single fractional component of the F2M, and we let G be the support graph of this component.

We once again create a new graph G' from G, so that we can later define a matching problem in G'. The matching will again show us how to create a G2M in G. We extend the previous construction to deal with the case when the support graph has cut paths. We introduce a gadget in G' for each cut path in G, which replaces the cut path and its two endpoints. The other paths in G are again replaced by single edges in G' of cost equal to the cost of the path. Each cycle edge in G is also in G' with cost equal to the negative of its cost in G.

To introduce the cut-path gadget, we begin by using an idea of Boyd and Carr [4]; namely, that we only need to consider three *patterns* to get an almost feasible graphical 2-matching on the cut path, when

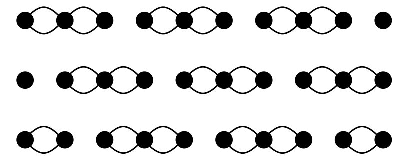

Figure 2: Illustrations of patterns for $\ell = 9$.

we allow ourselves to increase the cost by a third compared to the F2M. Suppose the cut path has ℓ edges and $\ell + 1$ nodes, and let $k = \lfloor \ell/3 \rfloor$. We can remove every third edge, double the remaining edges to obtain groups of nodes that are 2-edge-connected, where we get k groups of three nodes that are G2M components, plus one group of $\ell - 3k \in \{0, 1, 2\}$ nodes. Alternatively, we could remove every third edge, starting from the first edge and double the remaining edges, in which case the first group has one node, the next k or $k-1$ groups have three nodes and the last group again has one or two nodes. The final pattern removes every third edge, starting from the second edge, so that the first group has two nodes, the next k or $k - 1$ groups have three nodes, and, again, the last group has one or two nodes. Figure 2 illustrates the three patterns for $\ell = 9$.

To get a G2M that contains a certain pattern, we will ensure that if a group has size less than three, the G2M will include the two cycle edges incident to the first node (if the group is at the start of the pattern) or last node (if the group is at the end of the pattern).

We remark that for $\ell \geq 2$ there is exactly one pattern that starts with a group of size one, two and three, and hence two patterns need the G2M to include two cycle edges incident to the first node of the cut path. On the other hand, there is also exactly one pattern that ends with a group of size one, two and three (the length of the cut path determines which of the three patterns ends with a group of size three: it is the second pattern if $\ell \pmod 3 = 0$, the third pattern if $\ell \pmod 3 = 1$ and the first pattern if $\ell \pmod 3 = 2$), and hence there are also two patterns that need the G2M to include the two cycle edges incident to the last node of the cut path. If $\ell = 1$, there is one pattern that starts and ends with a group of size one, the other two patterns both start and end with a group of size two.

We are now ready to define the cut-path gadget. We replace each endpoint of the cut path in G by a path of length two in G'; each of these new edges will have cost 0. Each node on the path will be connected to a *pattern edge* corresponding to one of the three patterns. If $\ell \geq 2$, the middle node is connected to the pattern edge corresponding to the

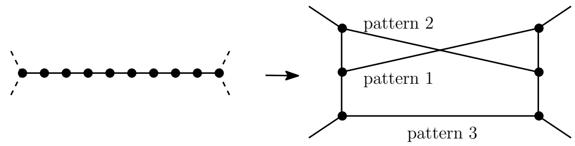

Figure 3: Pattern gadget for $\ell = 9$.

pattern which does not need two cycle edges incident to the endpoint of the cut path (i.e. the pattern for which the group containing the endpoint has size three). We set the cost of a pattern edge to the cost of the edges in the corresponding pattern. See Figure 3 for an illustration of the gadget when $\ell = 9$. If $\ell = 1$, the patterns associated with the illustration in Figure 3 are chosen so that pattern 3 corresponds to not adding the edge of the cut path, and patterns 1 and 2 correspond to doubling the edge.

If we replace each cut path in G by a cut-path gadget in G', once again G' will be a cubic graph. It is not hard to check that G' is also 2-edge-connected because we have replaced the cut path in G with three pattern edges crossing the cut in G'.

We argue that there is a minimum-cost perfect matching that uses exactly one edge from each cut-path gadget. Note that the fact that we replace only the cut paths in G by a cut gadget in G' means that a perfect matching in G' contains an odd number of pattern edges in a gadget. If it contains three pattern edges, then we could find a matching of no greater cost by choosing only one pattern edge, namely the pattern edge that is not incident to the middle node for the either one of its endpoints. Note that we can add two edges of cost 0 that connect the four nodes incident to the other two pattern edges, to again have a perfect matching without increasing the cost.

Now we show how to obtain a G2M in G from the minimum-cost perfect matching in G'. In the G2M we include all edges from G that are in paths which are not cut paths, the cycle edges in G which are *not* chosen by the perfect matching, duplicates of edges in paths in G that are chosen by the perfect matching, and the edges in a pattern if the corresponding pattern edge is in the perfect matching.

We argue that this set of edges is a G2M in G. Note that if the perfect matching contains only the pattern edge incident to the middle node, then the two cycle edges that are adjacent to the gadget are also in the matching. Hence the corresponding endpoint in G of the cut path has no cycle edges incident to it in the G2M, but since the pattern edge is incident to the middle node, the corresponding pattern ensures that the node has degree two and is in a connected component of size three. If the perfect matching contains the pattern edge incident to a node other than the middle node, then neither of the two cycle edges that are adjacent to the gadget in G' are in the perfect matching. Hence the corresponding endpoint of the cut path in G has both of these cycle edges incident to it in the G2M, and zero or two edges from the pattern corresponding to the chosen pattern edge. Hence the node has degree two or four and it is in a connected component of size at least three.

As before, because G' is cubic and 2-edge-connected, we can apply Lemma 3.1 to bound the cost of the perfect matching in G'. Let P_1 be the cost of the paths in G that are not cut paths, and P_2 the cost of the cut paths in G, so that the cost of the F2M is $P_1 + P_2 + \frac{1}{2}C$. Note that the cost of the three pattern edges in the gadget corresponding to a cut path sums up to four times the cost of the cut path. Thus the total cost of the edges in G' is $P_1 + 4P_2 - C$. By Lemma 3.1, the cost of the perfect matching in G' is at most $\frac{1}{3}P_1 + \frac{4}{3}P_2 - \frac{1}{3}C$. The cost of the G2M corresponding to the minimum-cost perfect matching is therefore at most

$$P_1 + \frac{1}{3}P_1 + \frac{4}{3}P_2 + C - \frac{1}{3}C = \frac{4}{3}P + \frac{2}{3}C = \frac{4}{3}\left(P + \frac{1}{2}C\right)$$

as claimed. ■

We can extend the ideas above to obtain a better G2M if no cut paths exist. The basic idea is that we replace every path by a cut-path gadget, and show that the solution $x(e) = \frac{1}{9}$ if e is a pattern edge and $x(e) = \frac{4}{9}$ if e is a cycle edge is feasible for the matching polytope (M).

THEOREM 3.3. *If an F2M has no cut edge, then there exists a G2M of cost at most $\frac{10}{9}$ times the cost of the F2M.*

4 A Polyhedral Proof of the Boyd-Carr Conjecture

We will generalize the result in Theorem 3.3 and show that the ratio between the cost of the optimal 2-matching and the subtour LP is at most $\frac{10}{9}$. In the combinatorial proofs of the previous section, we heavily used the fact that F2Ms have a nice simple structure, and, unfortunately, this does not hold for the subtour LP solution. We therefore turn to a polyhedral rather than a combinatorial proof. We derive a polyhedral description for graphical 2-matchings, and we then use this description to construct a feasible (fractional) G2M solution from any solution to the subtour LP of cost not more than $\frac{10}{9}$ times the value of the subtour LP. The manner in which the feasible G2M solution is defined based on a solution to $(SUBT)$ is a generalization of the proof of Theorem 3.3.

We start by giving a polyhedral description of a generalization of 2-matching, where the node set consists of "mandatory nodes" (V_{man}) and "optional nodes" (V_{opt}). The former need to have degree 2 in the solution, whereas the latter can have degree 0 or 2. We will refer to this problem as the 2-Matching with Optional Nodes Problem (2MO).

Theorem 4.1. *Let $G = (V_{\text{man}} \cup V_{\text{opt}}, E)$ be a 2MO instance. The convex hull of integer 2MO solutions is given by the following polytope:*

$$\sum_{e \in \delta(i)} y(e) = 2, \qquad \forall i \in V_{\text{man}}, \tag{4.9}$$

$$\sum_{e \in \delta(i)} y(e) \le 2, \qquad \forall i \in V_{\text{opt}}, \tag{4.10}$$

$$\sum_{e \in \delta(S) \setminus F} y(e) + \sum_{e \in F} (1 - y(e)) \ge 1, \quad \forall S \subseteq V, \; F \subseteq \delta(S),\ F \text{ matching},\ |F| \text{ odd}, \tag{4.11}$$

$$0 \le y(e) \le 1, \qquad \forall e \in E. \tag{4.12}$$

The proof of Theorem 4.1 is similar to the proof of the polyhedral description of the 2-matching polytope (Theorem 30.8) in Schrijver [19].

Recall the definition of a graphical 2-matching (G2M): (i) each node has degree either 2 or 4, (ii) each edge has 0, 1, or 2 copies, and (iii) each component has size at least three. We will (for the moment) relax the second condition so that each edge has at most 3 copies.

Lemma 4.1. *We can reduce a G2M instance $G = (V, E)$ to a 2MO instance $G' = (V', E')$ as follows: Let $V'_{\text{man}} = \{i_m : i \in V\}, V'_{\text{opt}} = \{i_o : i \in V\}, V' = V'_{\text{man}} \cup V'_{\text{opt}}, E' = \{(i_m, j_m) : (i,j) \in E\} \cup \{(i_m, j_o) : (i,j) \in E\}$. We add an edge $\{i,j\}$ to the (relaxed) G2M solution for each edge (i_m, j_m), (i_o, j_m) and (i_m, j_o) that is in the associated 2MO solution.*

If the edges have nonnegative costs, we may assume without loss of generality that each edge appears at most twice in an optimal G2M solution: if any edge appears three times, we can remove two copies of it without affecting the parity of its endpoints, and the cost cannot increase.

The following lemma shows how to map a solution x of the subtour LP to a solution y to the 2MO polytope corresponding to a G2M. The mapping is based on some insights gleaned from the proof of Theorem 3.3; we omit this discussion due to space constraints.

Lemma 4.2. *Given a graph $G = (V, E)$, let x be a feasible solution to the subtour LP for G. Then the following solution is a feasible solution to the 2MO instance $G' = (V', E')$ associated with the graphical 2-matching instance given by G for $\alpha = \frac{1}{9}$: $y(i_m, j_m) = (1 - \alpha)x(i,j)$, $y(i_m, j_o) = \alpha x(i,j)$, $y(i_o, j_m) = \alpha x(i,j)$ for all $(i,j) \in E$.*

Note that the cost of the constructed G2M solution is exactly $\frac{10}{9}$ times the cost of the solution of the subtour LP. Thus our result follows immediately from the lemma.

Corollary 4.1. *There exists a G2M of cost at most $\frac{10}{9}$ times the value of the subtour LP.*

Proof of Lemma 4.2: We need to show that y satisfies the constraints (4.9)-(4.12) on G', where G' is defined as in Lemma 4.1. Constraints (4.9), (4.10) and (4.12) are obviously met, and we only need to show that constraints (4.11) are met. To this end, fix $S \subseteq V'$, $F \subseteq \delta(S)$ where F is a matching and $|F|$ is odd. We define $z(e') = y(e')$ if $e' \in \delta(S) \setminus F$ and $z(e') = 1 - y(e')$ if $e' \in F$. For simplicity, for any set of edges $X \subseteq E'$, we define $z(X) = \sum_{e' \in X} z(e')$. Then we need to show that $z(\delta(S)) \ge 1$.

First, suppose S does not contain any node i_m for any $i \in V$. For any $j_o \in S$, we have that $z(\delta(S) \cap \delta(j_o)) = z(\{(i_m, j_o) : i \in V\})$. Since $|F| \ge 1$, there exists some $j_o \in S$ such that F contains some edge incident to j_o, say (i'_m, j_o). Then, $z(\{(i_m, j_o) : i \in V\}) = 1 - \alpha x(i', j) + \sum_{i \in V : i \neq i'} \alpha x(i,j) = \alpha x(\delta(j)) + 1 - 2\alpha x(i', j)$. Now, note that $x(\delta(j)) = 2$ and $x(i', j) \le 1$, hence $z(\delta(S) \cap \delta(j_o)) \ge 1$.

By symmetry, it remains to consider the case when both S and $V' \setminus S$ contain a node i_m for some $i \in V$.

We consider an edge $e = (i,j) \in G$ such that at least one of the three edges $(i_o, j_m), (j_m, i_m), (i_m, j_o)$ crosses the cut S in G'. Note that there are $2^3 - 1 = 7$ possible choices for the edges that cross the cut. We discern five different types of edges in G for which at least one of the three corresponding edges crosses the cut (type II and type V each cover 2 of the possible choices):

(I) The edge (i_m, j_m) crosses the cut.
(II) The edges (i_o, j_m) and (j_m, i_m) or the edges (j_m, i_m) and (i_m, j_o) cross the cut.
(III) The edges $(i_o, j_m), (j_m, i_m)$ and (i_m, j_o) cross the cut.
(IV) The edges $(i_o, j_m), (i_m, j_o)$ cross the cut.
(V) The edge (i_o, j_m) or the edge (i_m, j_o) crosses the cut.

Figure 4 illustrates the five types.

We use the notation i_* to denote either i_m or i_o, and we will say an edge $e' = (i_*, j_*) \in G'$ is in a gadget of type I, II, ..., V, if the edge $(i, j) \in G$ is an edge of that type.

We now consider three different cases, depending on the set F.

CLAIM 4.1. *If F contains an edge in a gadget of type IV or V, then $z(\delta(S)) \geq 1$.*

For the remaining cases, we associate a cut R in the graph G with the cut S in G': let $R = \{i \in V : i_m \in S\}$. Note that $R, V \backslash R$ are not empty. Note that if e is of type I, II, or III, then the edge (i_m, j_m) crosses the cut, and hence, the edge e crosses the cut R in G.

In the remainder of this proof, we will write $z(\delta(S)) = y(\delta(S)) + |F| - 2y(F)$, and we will give a lower bound on $y(\delta(S))$ to show that $z(\delta(S)) \geq 1$. In order to give a lower bound on $y(\delta(S))$, we need to use the fact that x satisfies degree constraints for each node, and that $x(\delta(R)) \geq 2$. It will therefore be convenient to relate the contribution to $y(\delta(S))$ of the three edges (i_o, j_m), (j_m, i_m), and (i_m, j_o) to the edge $(i, j) \in G$, if $(i, j) \in \delta(R)$, but also to the nodes i and j for certain types of nodes $i, j \in V$.

In particular, we say a node $i \in V$ is a *lonely node* if $|\{i_m, i_o\} \cap S| = 1$. We let L be the set of lonely nodes. We assign each lonely node i an amount of $\alpha x(i, j)$, for each edge (i, j) of type I, II, ..., V. Note that for each lonely node i, the paths $\{(i_o, j_m), (j_m, i_m)\}$ cross the cut for all $j \in V$, and hence, each lonely node gets assigned $\alpha \sum_j x(i, j)$, which by the degree constraints is equal to 2α.

(I) For an edge (i, j) of type I, the total contribution of the three edges $(i_o, j_m), (j_m, i_m), (i_m, j_o)$ to $y(\delta(S))$ is $(1 - \alpha)x(i, j)$. Note that both i and j are lonely nodes. We assign $(1 - 3\alpha)x(i, j)$ to the edge (i, j), and $\alpha x(i, j)$ each to nodes i and j.
(II) For an edge (i, j) of type II, the total contribution of the three edges $(i_o, j_m), (j_m, i_m), (i_m, j_o)$ to $y(\delta(S))$ is $x(i, j)$. Note that only one of i, j is a lonely node, and we therefore assign $(1-\alpha)x(i, j)$ to the edge (i, j), and $\alpha x(i, j)$ to the lonely node among i, j.
(III) For an edge (i, j) of type III, the total contribution of the three edges $(i_o, j_m), (j_m, i_m), (i_m, j_o)$ to $y(\delta(S))$ is $(1 + \alpha)x(i, j)$, and neither i nor j is a lonely node. We therefore assign $(1 + \alpha)x(i, j)$ to the edge (i, j).
(IV) For an edge (i, j) of type IV, the total contribution of the three edges $(i_o, j_m), (j_m, i_m), (i_m, j_o)$ to $y(\delta(S))$ is $2\alpha x(i, j)$. Since $(i, j) \notin \delta(R)$ and both i and j are lonely nodes, we assign 0 to (i, j) and $\alpha x(i, j)$ each to i and j.
(V) For an edge (i, j) of type V, the total contribution of the three edges $(i_o, j_m), (j_m, i_m), (i_m, j_o)$ to $y(\delta(S))$ is $\alpha x(i, j)$. Since $(i, j) \notin \delta(R)$ and only one of i and j is a lonely node, we can assign 0 to (i, j) and $\alpha x(i, j)$ to the lonely node.

By the argument above, we have assigned 2α to each lonely node. We now show how this fact, combined with the fact that $x(\delta(R)) \geq 2$ and the assignment of values to the edges in $\delta(R)$, allows us to conclude that $z(\delta(S)) \geq 1$.

CLAIM 4.2. *If $|F| = 1$, then $z(\delta(S)) \geq 1$.*

CLAIM 4.3. *If $|F| \geq 3$, then $z(\delta(S)) \geq 1$.*

Proof. By Claim 4.1, we may assume that all edges in F are contained in a gadget of type I, II or III, and hence, that the corresponding edges in $e \in G$ are in $\delta(R)$. Let E_1, E_2, E_3 be the edges in $\delta(R)$ of type I, II and III, respectively, for which the gadget contains one or more edges in F.

Note that a lonely node i can be incident to at most one edge in $E_1 \cup E_2 \cup E_3$: Only the edges $(i, j) \in E_1 \cup E_2$ can be incident to a lonely node i, and in the first case, (i_m, j_m) must be in F, and in the second case, either (i_m, j_o) or (i_m, j_m) is in F, since these are the only edges that cross the cut for these types. Now, since F is a matching, it can have at most one edge incident to i_m and hence i can be incident to at most one edge in $E_1 \cup E_2 \cup E_3$.

We therefore have that

$$\begin{aligned} y(\delta(S)) \quad &\geq \quad (1 - 3\alpha)x(E_1) + 4\alpha|E_1| + (1 - \alpha)x(E_2) \\ &\quad +2\alpha|E_2| + (1 + \alpha)x(E_3). \end{aligned}$$

On the other hand, since F is a matching, only the gadgets for edges of type III can contain two edges in F. Hence, $|F| = |E_1| + |E_2| + (1 + \beta)|E_3|$, where β is the fraction of edges in E_3 for which two edges in the corresponding gadget are contained in F.

Also, $y(F) \leq (1 - \alpha)\,(x(E_1) + x(E_2) + x(E_3))$, since $y((i_*, j_*)) \leq (1 - \alpha)x(i, j)$, and, if two edges

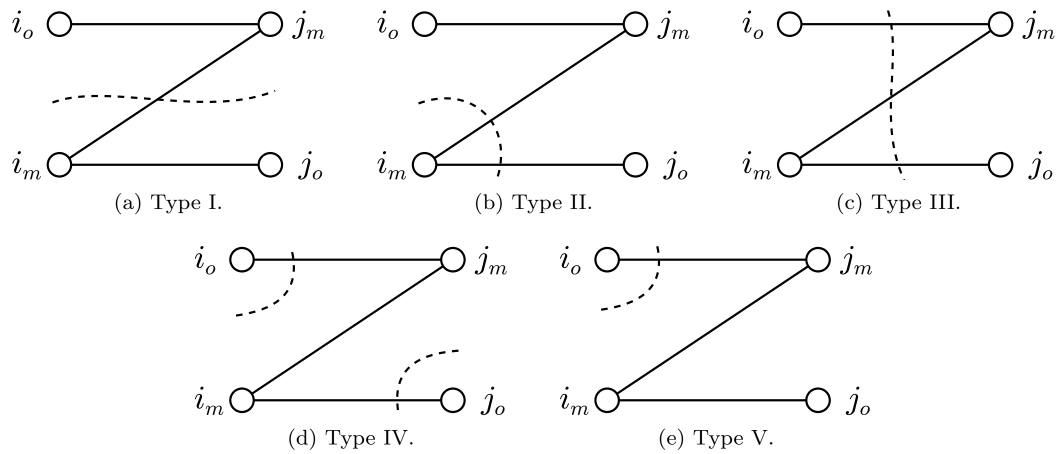

Figure 4: Illustrations of the five types of cuts of the edges in the reduction. The y-value on the top and bottom edge is $\alpha x(i,j)$ and the y-value on the middle edge is $(1-\alpha)x(i,j)$.

in the gadget for $e \in E_3$ are contained in F, then these edges both have y-value $\alpha x(e)$, and since $\alpha = \frac{1}{9}$, $2\alpha x(e) \le (1-\alpha)x(e)$.

Hence, we get that

$$\begin{aligned} z(\delta(S)) &= y(\delta(S)) + |F| - 2y(F) \\ &\geq (1+4\alpha)|E_1| + (-1-\alpha)x(E_1) \\ &\quad +(1+2\alpha)|E_2| + (-1+\alpha)x(E_2) \\ &\quad +|E_3| + (-1+3\alpha)x(E_3) + \beta|E_3| \\ &\geq 3\alpha(|E_1|+|E_2|+|E_3|) + \beta|E_3| \geq 3\alpha|F|, \end{aligned}$$

where the penultimate inequality follows from the fact that $x(E_k) \le |E_k|$ and $\alpha = \frac{1}{9}$, and the last inequality from the fact that $\alpha = \frac{1}{9}$. Hence, $z(\delta(S)) \ge 1$. ■

By the three claims we have for any $S \subseteq V', F \subseteq \delta(S)$, where F is a matching of odd size, that $\sum_{e' \in \delta(S)\setminus F} y(e') + \sum_{e' \in F}(1-y(e')) = z(\delta(S)) \ge 1$, and hence y is a feasible solution to the 2MO instance G'. ■

5 Conjectures and Conclusions

> *I conjecture that there is no [polynomial-time] algorithm for the traveling salesman problem. My reasons are the same as for any mathematical conjecture: (1) It is a legitimate mathematical possibility, and (2) I do not know.*
>
> — *Edmonds [10]*

We conclude our paper with a conjecture. We do so in the spirit of Jack Edmonds, quoted above; we do not know whether the conjecture is true or not, but we think that even a proof that this conjecture is false would be interesting. Our conjecture says that the integrality gap (or worst-case ratio) of the subtour LP is obtained for specific kinds of vertices of the subtour polytope; namely, ones in which the subtour LP solution has no subtour constraint as part of the dual basis, or, restated a different way, for costs c such that an optimal subtour LP solution for c is the same as an optimal fractional 2-matching for c. Let us call such costs c *fractional 2-matching costs* for the subtour LP. Note that for such solutions of the subtour LP, the fractional 2-matching will have no cut edge.

CONJECTURE 2. *The integrality gap for the subtour LP is attained for a fractional 2-matching cost for the subtour LP.*

We could make a similar conjecture for the ratio of the cost of the optimal 2-matching to the subtour LP, but by Corollary 4.1 and the example in Figure 1, we already know that the conjecture is true. However, its truth does not shed any light on the conjecture above.

In a companion paper, Qian et al. [18] show that if an analogous conjecture for edge costs $c(i,j) \in \{1,2\}$ is true, then the integrality gap for 1,2-TSP is at most $\frac{7}{6}$. They conjecture that the integrality gap for the 1,2-TSP is at most $\frac{10}{9}$; it is known that it can be no smaller than $\frac{10}{9}$. It would be nice to show that if the analogous conjecture is true then the integrality gap for 1,2-TSP is at most $\frac{10}{9}$.

Interestingly, we appear to know almost nothing about the consequences of Conjecture 2. Even for this very restricted set of cost functions, we do not know a better upper bound on the integrality gap of the subtour LP other than the bound of $\frac{3}{2}$. Note that the lower bound of $\frac{4}{3}$ is attained for a fractional 2-matching cost. It would be very interesting to prove that for such costs the integrality gap is indeed $\frac{4}{3}$. Boyd and Carr [3] have shown this for some fractional 2-matching costs in which all the cycles of the fractional 2-matching have size 3; this result also follows from the technique of Theorem 3.1, since the resulting graphical 2-matching is Eulerian if all cycles have size 3 and the fractional 2-matching has a single component (the graphical 2-matching may not be connected if there are cycles of size 5).

Acknowledgements We thank Sylvia Boyd for useful and encouraging discussions; we also thank her for giving us pointers on her various results. Gyula Pap made some useful suggestions regarding the polyhedral formulation of graphical 2-matchings.

References

[1] M. L. Balinski. Integer programming: Methods, uses, computation. *Management Science*, 12:253–313, 1965.

[2] G. Benoit and S. Boyd. Finding the exact integrality gap for small traveling salesman problems. *Mathematics of Operations Research*, 33:921–931, 2008.

[3] S. Boyd and R. Carr. Finding low cost TSP and 2-matching solutions using certain half-integer subtour vertices. To appear in *Discrete Optimization*. See `http://dx.doi.org/10.1016/j.disopt.2011.05.002`. Prior version available at `http://www.site.uottawa.ca/~sylvia/recentpapers/halftri.pdf`. Accessed June 27, 2011.

[4] S. Boyd and R. Carr. A new bound for the ratio between the 2-matching problem and its linear programming relaxation. *Mathematical Programming*, 86:499–514, 1999.

[5] S. Boyd and P. Elliott-Magwood. Structure of the extreme points of the subtour elimination polytope of the STSP. In S. Iwata, editor, *Combinatorial Optimization and Discrete Algorithms*, volume B23 of *RIMS Kôkyûroku Bessatsu*, pages 33–47. Research Institute for Mathematical Sciences, Kyoto University, Kyoto, Japan, 2010.

[6] S. Boyd, R. Sitters, S. van der Ster, and L. Stougie. TSP on cubic and subcubic graphs. In O. Günlük and G. J. Woeginger, editors, *Integer Programming and Combinatorial Optimization, 15th International Conference, IPCO 2011*, number 6655 in Lecture Notes in Computer Science, pages 65–77. Springer, Berlin, Germany, 2011.

[7] 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.

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

[9] J. Edmonds. Maximum matching and a polyhedron with (0,1) vertices. *J. Res. Nat. Bur. Standards Sect. B*, 69B:125–130, 1965.

[10] J. Edmonds. Optimum branchings. *Journal of Research of the National Bureau of Standards B*, 71B:233–240, 1967.

[11] M. X. Goemans. Worst-case comparison of valid inequalities for the TSP. *Mathematical Programming*, 69:335–349, 1995.

[12] M. X. Goemans and D. J. Bertsimas. Survivable networks, linear programming relaxations, and the parsimonious property. *Mathematical Programming*, 60:145–166, 1990.

[13] D. S. Johnson and L. A. McGeoch. Experimental analysis of heuristics for the STSP. In G. Gutin and A. P. Punnen, editors, *The Traveling Salesman Problem and Its Variants*, pages 369–444. Kluwer Academic Publishers, Dordrect, The Netherlands, 2002.

[14] T. Mömke and O. Svensson. Approximating graphic TSP by matchings. In *Proceedings of the 52th Annual Symposium on Foundations of Computer Science*, 2011. To appear.

[15] M. Mucha. Improved analysis for graphic TSP approximation via matchings. Appears at `http://arxiv.org/abs/1108.1130`, 2011.

[16] D. Naddef and W. R. Pulleyblank. Matchings in regular graphs. *Discrete Mathematics*, 34:283–291, 1981.

[17] S. Oveis Gharan, A. Saberi, and M. Singh. A randomized rounding approach to the traveling salesman problem. In *Proceedings of the 52th Annual Symposium on Foundations of Computer Science*, 2011. To appear.

[18] J. Qian, F. Schalekamp, D. P. Williamson, and A. van Zuylen. On the integrality gap of the subtour LP for the 1,2-TSP. Manuscript. Available at `http://arxiv.org/abs/1107.1630`, 2011.

[19] A. Schrijver. *Combinatorial Optimization - Polyhedra and Efficiency*. Springer, 2003.

[20] D. B. Shmoys and D. P. Williamson. Analyzing the Held-Karp TSP bound: A monotonicity property with application. *Information Processing Letters*, 35:281–285, 1990.

[21] D. P. Williamson. Analysis of the Held-Karp heuristic for the traveling salesman problem. Master's thesis, MIT, Cambridge, MA, June 1990. Also appears as Tech Report MIT/LCS/TR-479.

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