A 4/3-Approximation Algorithm for the Minimum 2-Edge Connected Multisubgraph Problem in the Half-Integral Case*

Sylvia Boyd † Joseph Cheriyan ‡ Robert Cummings ‡ Logan Grout §
Sharat Ibrahimpur ‡ Zoltán Szigeti ¶ Lu Wang ‡

November 6, 2020

Abstract

Given a connected undirected graph $\overline{G}$ on n vertices, and non-negative edge costs c, the 2ECM problem is that of finding a 2-edge connected spanning multisubgraph of $\overline{G}$ of minimum cost. The natural linear program (LP) for 2ECM, which coincides with the subtour LP for the Traveling Salesman Problem on the metric closure of $\overline{G}$, gives a lower bound on the optimal cost. For instances where this LP is optimized by a half-integral solution x, Carr and Ravi (1998) showed that the integrality gap is at most $\frac{4}{3}$: they show that the vector $\frac{4}{3}x$ dominates a convex combination of incidence vectors of 2-edge connected spanning multisubgraphs of $\overline{G}$.

We present a simpler proof of the result due to Carr and Ravi by applying an extension of Lovász's splitting-off theorem. Our proof naturally leads to a $\frac{4}{3}$-approximation algorithm for half-integral instances. Given a half-integral solution x to the LP for 2ECM, we give an $O(n^2)$-time algorithm to obtain a 2-edge connected spanning multisubgraph of $\overline{G}$ whose cost is at most $\frac{4}{3}c^T x$.

1 Introduction

The 2-edge connected multisubgraph (2ECM) problem is a fundamental problem in survivable network design where one wants to be resilient against a single edge failure. In this problem, we are given an undirected graph $\overline{G} = (\overline{V}, \overline{E})$ with non-negative edge costs c and we want to find a 2-edge connected spanning multisubgraph of $\overline{G}$ of minimum cost. A multigraph may contain multiple copies of edges but it cannot contain loops. Below we give an integer linear program for 2ECM. The variable x_e denotes the number of copies of edge e that are used in a feasible solution. For any $S \subseteq \overline{V}$, $\delta_{\overline{G}}(S) := \{e = uv \in \overline{E} : u \in S, v \notin S\}$ denotes the cut

*A preliminary version of this paper appeared in the Proceedings of APPROX 2020. `https://doi.org/10.4230/LIPIcs.APPROX/RANDOM.2020.61`

†`sboyd@uottawa.ca`. School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, Canada.

‡`{jcheriyan,sharat.ibrahimpur}@uwaterloo.ca`. Dept. of Combinatorics and Optimization, University of Waterloo, Waterloo, Canada.

§School of OR&IE, Cornell University, Ithaca, NY, USA.

¶`zoltan.szigeti@grenoble-inp.fr`. Univ. Grenoble Alpes, Grenoble INP, G-SCOP, 46 Ave. Félix Viallet, 38031 Grenoble, France.

induced by S; we use the shorthand $\delta(S)$ whenever the underlying graph is clear from the context. For any $F \subseteq \overline{E}$ and vector $x \in \mathbb{R}^{\overline{E}}$, we use $x(F)$ as a shorthand for $\sum_{e \in F} x_e$. Also, for any graph H with edge costs c, we may use $c(H)$ as a shorthand for $c(E(H))$.

$$
\text{(2ECM-IP)} \qquad
\begin{aligned}
\min \quad & \sum_{e \in \overline{E}} c_e x_e && && (1)\\
\text{subject to} \quad & x(\delta(S)) \geq 2 && \forall \emptyset \subsetneq S \subsetneq \overline{V}, && (2)\\
& x_e \geq 0 && \forall e \in \overline{E}, && (3)\\
& x_e \text{ integral} && \forall e \in \overline{E}. && (4)
\end{aligned}
$$

It is easy to see that an optimal solution for 2ECM never has to use more than two copies of an edge. As is discussed in [CR98], since we are allowed to use more than one copy of an edge, without loss of generality, we may assume that $\overline{G}$ is complete by performing the metric completion: for each $u, v \in \overline{V}$ we set the new cost of the edge uv to be the shortest path distance between u and v in $\overline{G}$. In what follows, we assume that $\overline{G}$ is a complete graph and that the cost function c is metric, i.e., $c \geq 0$ and for every $u, v, w \in \overline{V}$, we have $c_{uw} \leq c_{uv} + c_{vw}$.

The linear relaxation (2ECM-LP) for 2ECM is obtained by dropping the integrality constraints given by (4). By a result due to Goemans and Bertsimas [GB93] called the *parsimonious property*, adding the constraint $x(\delta(v)) = 2$ for each $v \in \overline{V}$ to (2ECM-LP) does not increase the optimal solution value; here, we require the assumption that the costs form a metric. So, the optimal value of (2ECM-LP) is the same as the optimal value for the well-known subtour elimination LP (Subtour-LP) for the Traveling Salesman Problem (TSP) defined below. Due to this connection, we often refer to an optimal solution for (2ECM-LP) as an optimal solution to (Subtour-LP), and vice versa. Another consequence of the parsimonious property is that for graphs with at least 3 vertices, the constraint $x_e \leq 1$ is implied by (Subtour-LP): for any $e = uv$, we have $2x_e = x(\delta(u)) + x(\delta(v)) - x(\delta(\{u, v\})) \leq 2$.

$$
\text{(Subtour-LP)} \qquad
\begin{aligned}
\min \quad & \sum_{e \in \overline{E}} c_e x_e && && (5)\\
\text{subject to} \quad & x(\delta(S)) \geq 2 && \forall \emptyset \subsetneq S \subsetneq \overline{V}, && (6)\\
& x(\delta(v)) = 2 && \forall v \in \overline{V}, && (7)\\
& x_e \geq 0 && \forall e \in \overline{E}. && (8)
\end{aligned}
$$

A long-standing open problem called the "four-thirds conjecture" states that the integrality gap of (Subtour-LP) is $\frac{4}{3}$. Besides the importance of 2ECM in the field of survivable network design, the connection between (2ECM-LP) and (Subtour-LP) has spurred interest in determining the integrality gap for (2ECM-LP) as a means to gaining useful lower bounds on the integrality gap for (Subtour-LP). The general version of metric TSP has resisted all attempts at proving an upper bound better than $\frac{3}{2}$ on the integrality gap, so a great deal of research has focused on obtaining improvements for important special cases. In [SWvZ14], the authors conjecture that the integrality gap for (Subtour-LP) is achieved on instances where an optimal (fractional) solution to (Subtour-LP) is half integral, i.e., $2x_e \in \mathbb{Z}_{\geq 0}$ for all $e \in \overline{E}$. We refer to such instances as *half integral instances*. More than two decades ago, Carr and Ravi [CR98] proved that the integrality gap of (2ECM-LP) is at most $\frac{4}{3}$ in the half-integral case. They show that $\frac{4}{3}x$ dominates a convex combination of incidence vectors of

2-edge connected spanning multisubgraphs of $\overline{G}$. This supports the four-thirds conjecture for TSP since the (integer) optimal value for 2ECM lower bounds the (integer) optimal value for TSP. However, the proof of Carr and Ravi [CR98] does not give a polynomial-time algorithm for 2ECM. Very recently, in [KKG20], Karlin, Klein, and Oveis Gharan gave a randomized approximation algorithm for half-integral instances of TSP whose (expected) approximation factor is $\frac{3}{2} - 0.00007$. This immediately implies a better than $\frac{3}{2}$-approximation algorithm, albeit randomized, for 2ECM as well.

We mention that the result of Carr and Ravi [CR98] does not apply to the strict variant of 2ECM (henceforth denoted by 2ECS) where we are allowed to pick at most one copy of an edge in $\overline{G}$, i.e., where we are considering subgraphs of $\overline{G}$ rather than multisubgraphs; similarly, our main result does not apply to 2ECS.

1.1 Our Work

Our main contribution, found in Section 3, is a deterministic approximation algorithm for 2ECM on half-integral instances that matches the existence result in [CR98].

Theorem 1. *Let x denote a half-integral solution to an instance $(\overline{G}, c)$ of* (Subtour-LP) *(and* (2ECM-LP)*). There is an $O(|V(\overline{G})|^2)$-time algorithm for computing a 2-edge connected spanning multisubgraph of $\overline{G}$ with cost at most $\frac{4}{3}c^T x$.*

For any $F \subseteq E$, let $\chi^F \in \{0,1\}^E$ denote the characteristic vector of F: $\chi^F_e = 1$ if and only if $e \in F$. Note that distinct multiedges in E correspond to distinct coordinates in χ^F.

The algorithm of Theorem 1 is facilitated by a simpler proof of the existence result in [CR98], which has insights that may be useful for generalizing their result in the future. This simpler proof is given in Section 2. Note that, as discussed in [CV04], if one could show that for every integer $k \geq 2$ and for any $2k$-regular $2k$-edge connected graph G that $\frac{4}{3k}\chi^{E(G)}$ is a convex combination of incidence vectors of 2-edge connected spanning subgraphs of G, then this would imply that the integrality gap of (2ECM-LP) is bounded above by $\frac{4}{3}$. Carr and Ravi [CR98] show that this is true for $k = 2$. Thus, any simplification of their proof provides hope for extending their results to other values of k.

Given a half-integral solution x to (Subtour-LP) for $\overline{G}$, let $G = (V, E)$ denote the multigraph induced by $2x$. Formally, the vertex-set is $V := \overline{V}$, and for each edge $e \in \overline{E}$, the edge-set E has $2x_e$ copies of the edge e. Note that if $|\overline{V}| \geq 3$, then $2x_e \in \{0, 1, 2\}$ for all $e \in \overline{E}$, and if $|\overline{V}| = 2$, then $2x_e = 4$ for the unique edge $e \in \overline{E}$. With a slight abuse of notation, we use the same cost function c to denote the edge costs in G, i.e., $c_f := c_e$ where $e \in \overline{E}$ corresponds to the edge $f \in E$. By (7) and (6), G is a 4-regular 4-edge connected multigraph. Theorem 1 follows from the following result applied to the graph G induced by $2x$.

Theorem 2. *Let $G = (V, E)$ be a 4-regular 4-edge connected multigraph on n vertices. Let $c : E \to \mathbb{R}$ be an arbitrary cost function on the edges of G (negative costs on the edges are allowed), and let e be an arbitrary edge in G. Then, in $O(n^2)$ time, we can find a 2-edge connected spanning subgraph H of $G - e$ satisfying:*

(i) $c(H) \leq \frac{2}{3}c(G - e)$; and

(ii) each multiedge of G appears at most once in H (multiedges may arise in H due to multiedges in G).

As mentioned before, Carr and Ravi [CR98] prove the existence of such a subgraph H by showing that for any 4-regular 4-edge connected multigraph G, there exists a finite collection $H_1, \ldots, H_k$ of 2-edge connected spanning subgraphs of G such that $\frac{2}{3}\chi^{E(G)\setminus\{e\}}$ lies in the convex hull of $\{\chi^{H_i}\}_i$. At a high level, their proof is inductive and splits into two cases based on whether G has a certain kind of a tight set (a cut of size 4). In the first case they construct two smaller instances of the problem by contracting each of the shores of the tight set, and in the second case they perform two distinct splitting-off operations at a designated vertex to obtain two smaller instances of the problem. In either case, the convex combinations from the two subinstances are merged to obtain a convex combination for G. The first case requires gluing since the subgraphs obtained from the two subinstances need to agree on a (tight) cut. Merging the convex combinations arising from the second case is rather straightforward as the two subinstances are more or less independent.

Our first insight in this work is that the case from Carr and Ravi's proof that requires the gluing step can be completely avoided, thereby unifying the analysis. This is discussed in Section 2. Our proof relies on an extension of Lovász's splitting-off theorem that is due to Bang-Jensen et al., [BJGJS99]. For further discussion on splitting-off theorems, see [Fra11, Chapter 8]. The challenge in efficiently finding a cheap subgraph H from the above convex combination construction is that each inductive step requires solving two subinstances of the problem, each with one fewer vertex, leading to an exponential-time algorithm. Having said that, an (expected) polynomial-time Las Vegas randomized algorithm can be easily designed that randomly recurses on one of the two subinstances and produces a 2-edge connected spanning subgraph whose expected cost is at most $\frac{2}{3}c(G-e)$. Our second insight, which is used in derandomizing the above procedure, is that it is easy to recognize which of the two subinstances leads to a "cheaper" solution, so we recurse only on the cheaper subinstance. Complementing this step, we lift the solution back to the original instance. This operation can lead to two different outcomes so the cost analysis must account for the worst outcome. There is a choice of defining the costs in the subinstance such that the cost of the lifted subgraph is the same irrespective of the outcome. Such a choice can lead to negative costs, but this is not a hindrance for our inductive step because Theorem 2 allows arbitrary real-valued edge costs. This generality of cost functions is crucial to our algorithm.

To obtain their results, Carr and Ravi [CR98] show that for any 4-regular 4-edge connected multigraph G, $\frac{2}{3}\chi^{E(G)}$ is dominated by a convex combination of incidence vectors of 2-edge connected spanning subgraphs of G. In Section 4 we consider a well-studied special case of 2ECS analogous to this problem. Let $G = (V, E)$ be a 3-regular 3-edge connected graph on n vertices. It is shown in [BL17] that $\frac{4}{5}\chi^{E(G)}$ can be written as a convex combination of incidence vectors of 2-edge connected spanning subgraphs of G, however, as was the case for the Carr and Ravi [CR98] result, the constructive proof given does not run in polynomial time. We give a randomized algorithm that produces a random 2-edge connected spanning subgraph H of G such that $\mathbb{E}\left[\chi^{E(H)}\right] = \frac{4}{5}\chi^{E(G)}$ and the expected running time is polynomial in n. Lastly, in Section 5 we show that the running time of the algorithm for Theorem 2 can be improved by using results from the area of dynamic graph algorithms.

1.2 Related Work

The 2ECM problem has been intensively studied in network design and several works have tried to bound the integrality gap α(2ECM) of (2ECM-LP). For the general case with metric costs, we have $\frac{6}{5} \leq \alpha(\mathsf{2ECM}) \leq \frac{3}{2}$, where the lower bound is from [ABEM06] and the upper bound follows from the polyhedral analysis of Wolsey [Wol80] and Shmoys and Williamson [SW90] (this analysis also gives a $\frac{3}{2}$-approximation algorithm). It is generally conjectured that $\alpha(\mathsf{2ECM}) = \frac{4}{3}$, however in [ABEM06], Alexander et al., study $\alpha(\mathsf{2ECM})$ and conjecture that $\alpha(\mathsf{2ECM}) = \frac{6}{5}$ based on their findings. As mentioned before, Carr and Ravi [CR98] show that the integrality gap of ($\mathsf{2ECM\text{-}LP}$) is at most $\frac{4}{3}$ in the half-integral case. In [BL17] Boyd and Legault consider a more restrictive collection of instances called half-triangle instances where the optimal LP solution is half-integral and the graph induced by the half-edges is a collection of disjoint triangles. They prove that $\alpha(\mathsf{2ECM}) = \frac{6}{5}$ in this setting. Half-triangle solutions are of interest as there is evidence that the integrality gap of ($\mathsf{2ECM\text{-}LP}$) is attained at such solutions (see [ABEM06]). When the costs come from a graphic metric (i.e., we want to find a minimum-size 2-edge connected spanning multisubgraph of a given unweighted graph), we have $\frac{8}{7} \leq \alpha(\mathsf{2ECM}) \leq \frac{4}{3}$ (see [BFS16, SV14]).

For the special case of 3-regular 3-edge connected graphs G, $\mathsf{2ECS}$ has also been extensively studied. In [BL17] it was shown that $\beta\chi^{E(G)}$ can be expressed as a convex combination of incidence vectors of 2-edge connected spanning subgraphs of G for $\beta = \frac{4}{5}$ and this was later improved to $\beta = \frac{7}{9}$ in [Leg17a]. Neither of these constructive proofs run in polynomial time, however in [HN18], it was shown that $\beta = \frac{7}{8}$ can be achieved in polynomial time. Note that the everywhere $\frac{2}{3}$ vector for G is feasible for ($\mathsf{Subtour\text{-}LP}$), thus for any edge costs c for which the everywhere $\frac{2}{3}$ vector is also optimal for ($\mathsf{Subtour\text{-}LP}$) (such as for the graphic metric), such an algorithm would provide a $\frac{3}{2}(\frac{7}{8}) = \frac{21}{16}$-approximation for $\mathsf{2ECS}$. For the graphic metric, there exists a $\frac{6}{5}$-approximation for 3-regular 3-edge connected graphs [BIT13], and for $\mathsf{2ECS}$ for general graphs, the best approximation factor known is 2 [Jai01].

2 A Simpler Proof of a Result of Carr and Ravi

In this section, we give a simplified proof of the following result from [CR98]. As mentioned before, avoiding the case involving the gluing operation is useful for our algorithm in Section 3. For notational convenience, for any subgraph K of some graph, we use χ^K to denote $\chi^{E(K)}$ whenever the underlying graph is clear from the context.

Theorem 3 (Statement 1 from [CR98]). *Let $G = (V, E)$ be a 4-regular 4-edge connected multigraph and $e = uv$ be an arbitrary edge in this graph. There exists a finite collection $\{H_1, \ldots, H_k\}$ of 2-edge connected spanning subgraphs of $G - e$ such that for some nonnegative $\mu_1, \ldots, \mu_k$ with $\sum_i \mu_i = 1$, we have $\frac{2}{3}\chi^{E\setminus\{e\}} = \sum_{i=1}^{k} \mu_i \chi^{H_i}$. Moreover, we may assume that none of the H_i's use more than one copy of an edge in E; H_i may have multiedges as long as they come from distinct edges in G.*

2.1 Operations involving splitting-off at a vertex

The following tools on the splitting-off operation will be useful. In keeping with standard terminology, we designate a vertex v (one of the endpoints of e in the theorem statement) at which the splitting-off operation is applied. For a multigraph $H = (V, E)$ and $x, y \in V$, let $\lambda_H(x, y)$ denote the size of a minimum (x, y)-cut in H, and let $\deg_H(x)$ denote the degree of x in H, that is $|\delta(\{v\})|$. Note that each multiedge is counted separately towards the degree of a vertex and the size of a cut.

Definition 4. *Given a multigraph G and two edges sv and vt that share an endpoint v, the graph $G_{s,t}$ obtained by* splitting off *the pair (sv, vt) at v is given by $G + st - sv - vt$.*

Definition 5. *Given a multigraph G and a vertex v of G of even degree, a* complete splitting *at v is a sequence of $\frac{1}{2}deg_G(v)$ splitting off operations that result in vertex v having degree zero in the resulting graph.*

Definition 6. *Let $k \geq 2$ be an integer and let $G = (V, E)$ be a multigraph such that for all $x, y \in V \setminus \{v\}$, $\lambda_G(x, y) \geq k$. Let sv and vt be two edges incident to v. We say that the pair (sv, vt) is* admissible *if for all $x, y \in V \setminus \{v\}$, $\lambda_{G_{s,t}}(x, y) \geq k$. For any edge $e \in \delta(v)$, we let A_e denote the set of edges $f \in \delta(v) \setminus \{e\}$ such that (e, f) is an admissible pair.*

The following result due to Bang-Jensen et al., [BJGJS99] shows that in our setting with a 4-regular 4-edge connected multigraph at least two distinct edges incident to v form an admissible pair with $e = uv$. Using this we can perform a complete splitting at v in two distinct ways.

Lemma 7 (Theorem 2.12 from [BJGJS99])**.** *Let $k \geq 2$ be an even integer. Let G be a multigraph and v a vertex of G such that for all $x, y \in V(G) \setminus \{v\}$, $\lambda_G(x, y) \geq k$ and $\deg_G(v)$ is even. Then, $|A_{uv}| \geq \frac{1}{2}\deg_G(v)$ for all $uv \in E(G)$.*

Lemma 8. *Let G be a 4-regular 4-edge connected multigraph and $e = vx$ be an edge incident to v. Then, (i) $|A_e| \geq 2$; and (ii) if (e, f) is an admissible pair for some $f = vy \in \delta(v) \setminus \{e\}$, then the remaining two edges in $\delta(v) \setminus \{e, f\}$ form an admissible pair in $G_{x,y}$.*

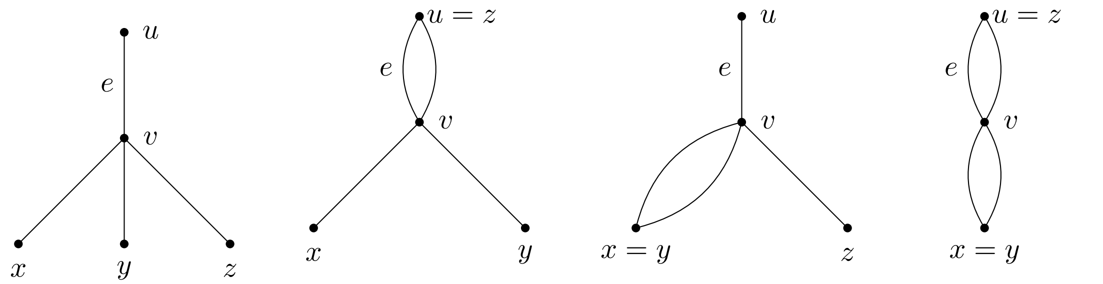
Proof. Conclusion (i) follows from Lemma 7 since G is 4-regular and 4-edge connected. For conclusion (ii), let $f \in \delta(v) \setminus \{e\}$ be such that (e, f) forms an admissible pair in G. Let $G_{x,y}$ denote the graph obtained by splitting off the pair $(e = vx, f = vy)$, i.e., $G_{x,y} = G - vx - vy + xy$. Observe that the hypothesis of Lemma 7 still holds for $G_{x,y}$ with $k = 4$ because (a) we performed a splitting off operation using an admissible pair of edges; and (b) $\deg_{G_{x,y}}(v) = 2$ is even. Let g denote one of the two remaining edges in $\delta(v) \setminus \{e, f\}$. By Lemma 7, the other unique edge $h \in \delta(v) \setminus \{e, f, g\}$ forms an admissible pair with g in $G_{x,y}$. □

Equipped with the above tools, we give a proof of Theorem 3.

Proof of Theorem 3. We prove this theorem via induction on the number n of vertices. The base case $n = 2$ corresponds to a pair of vertices having four parallel edges, call them e, f, g, h. Observe that $\frac{2}{3}\chi^{E\setminus\{e\}} = \frac{1}{3}\chi^{\{f,g\}} + \frac{1}{3}\chi^{\{f,h\}} + \frac{1}{3}\chi^{\{g,h\}}$, so the induction hypothesis is true for the base case.

For the induction step, suppose that $n \geq 3$ and the hypothesis holds for all 4-regular 4-edge connected multigraphs with at most $n - 1$ vertices and for all choices of the edge e. Consider a 4-regular 4-edge connected multigraph G on n vertices and an arbitrary edge $e = uv \in E(G)$. Besides e, let vx, vy, vz be the other three edges incident to v. With a relabeling of vertices, by Lemma 8, we may assume that (uv, vx) and (uv, vy) form an admissible pair in G (see Figure 1).

By the second conclusion of Lemma 8, (vy, vz) is an admissible pair in $G_{u,x}$, and (vx, vz) is an admissible pair in $G_{u,y}$. Consider the graph G_1 obtained by splitting off the pair (vy, vz) in $G_{u,x}$, i.e., $G_1 = G - v + \{ux, yz\}$; it is customary to drop the vertex v after all its edges



(a) v has four distinct neighbors $2 \leq |A_e| \leq 3$.

(b) v has two parallel edges with u $A_e = \{vx, vy\}$.

(c) v has two parallel edges with $x, x \neq u$ $A_e = \{vx, vy\}$.

(d) v has two parallel edges to each of $\{u, x\}$ $A_e = \{vx, vy\}$.

Figure 1: Four configurations of edges in $\delta(v) = \{uv, vx, vy, vz\}$ that can arise in our proof.

have been split off. Similarly, let G_2 be the graph obtained by splitting off the pair (vx, vz) in $G_{u,y}$, i.e., $G_2 = G - v + \{uy, xz\}$.

Since we only split off admissible pairs, both G_1 and G_2 are 4-regular 4-edge connected multigraphs on $n-1$ vertices. Recall that for any subgraph K of some graph, χ^K is a shorthand for $\chi^{E(K)}$ whenever the underlying graph is clear from the context. Applying the induction hypothesis to G_1 with the designated edge $e_1 = ux$ gives:

$$\frac{2}{3} \cdot \chi^{E(G_1)\setminus\{e_1\}} = \frac{2}{3} \cdot \chi^{(E\setminus\delta(v))\cup\{yz\}} = \sum_{i=1}^{k_1} \mu_i^1 \chi^{H_i^1} \ , \tag{ConvexComb-G_1}$$

where $\{\mu_i^1\}_i$ denote the coefficients in a convex combination, and $\{H_i^1\}_i$ are 2-edge connected spanning subgraphs of G_1 such that none of them use more than one copy of an edge in G_1. Repeating the same argument for G_2 with the designated edge $e_2 = uy$ gives:

$$\frac{2}{3} \cdot \chi^{E(G_2)\setminus\{e_2\}} = \frac{2}{3} \cdot \chi^{(E\setminus\delta(v))\cup\{xz\}} = \sum_{i=1}^{k_2} \mu_i^2 \chi^{H_i^2} \ , \tag{ConvexComb-G_2}$$

where $\{\mu_i^2\}_i$ denote the coefficients in the other convex combination arising from $\{H_i^2\}_i$. It remains to combine (ConvexComb-G_1) and (ConvexComb-G_2) to obtain such a representation for G with the designated edge e. We mimic the strategy from [CR98].

For each $i \in \{1, \ldots, k_1\}$, we lift H_i^1 to a spanning subgraph $\hat{H}_i^1$ of $G - e$. Define $\hat{H}_i^1$ as follows:

$$\hat{H}_i^1 := \begin{cases} H_i^1 - yz + vy + vz & \text{if } yz \in E(H_i^1), \\ H_i^1 + vy + vx & \text{if } yz \notin E(H_i^1). \end{cases} \tag{Lift-G_1}$$

Similarly, for each $i \in \{1, \ldots, k_2\}$, we define $\hat{H}_i^2$ as the following spanning subgraph of $G - e$:

$$\hat{H}_i^2 := \begin{cases} H_i^2 - xz + vx + vz & \text{if } xz \in E(H_i^2), \\ H_i^2 + vx + vy & \text{if } xz \notin E(H_i^2). \end{cases} \tag{Lift-G_2}$$

We finish the proof of Theorem 3 by arguing that the following convex combination meets all the requirements:

$$q := \frac{1}{2}\sum_{i=1}^{k_1} \mu_i^1 \chi^{\hat{H}_i^1} + \frac{1}{2}\sum_{i=1}^{k_2} \mu_i^2 \chi^{\hat{H}_i^2} \;. \tag{ConvexComb-G}$$

Most of our arguments are the same for G_1 and G_2, so we just mention them in the context of G_1. First of all, by the induction hypothesis and (Lift-G_1) it is clear that $e(= uv), yz, ux \notin E(\hat{H}_i^1)$, where yz and ux refer to the edges that originated from the splitting off operations applied at v. Next, we argue that $\hat{H}_i^1$ is a spanning subgraph of G that uses no more than one copy of any edge in G. By the induction hypothesis, none of the subgraphs H_i^1 use more than one copy of an edge in G_1, and H_i^1 spans $V(G) \setminus \{v\}$. By the way we lift H_i^1 to $\hat{H}_i^1$, it is clear that $\hat{H}_i^1$ uses no more than one copy of any multiedge in G, and that it is spanning. To see that $\hat{H}_i^1$ is 2-edge connected, observe that the two cases of lifting may be viewed as either (i) subdividing the edge yz by a vertex v when $yz \in E(H_i^1)$, or (ii) adding an edge yx and subdividing it by a vertex v when $yz \notin E(H_i^1)$. Clearly, these operations preserve 2-edge connectivity, hence, $\hat{H}_i^1$ is 2-edge connected.

It remains to argue that the vector q in the expression (ConvexComb-G) matches the vector $\frac{2}{3}\chi^{E(G)\setminus\{e\}}$. Since $\{\mu_i^1\}_i$ and $\{\mu_i^2\}_i$ denote coefficients in a convex combination, taking an unweighted average of these two combinations gives us another convex combination. Since none of the edges in $E(G) \setminus \delta(v)$ are modified in the lifting step, $q_f = \frac{2}{3}$ for any such edge f. Next, consider the edge vy. Observe that $\hat{H}_i^1$ always contains the edge vy, whereas $\hat{H}_i^2$ contains vy only when $xz \notin E(H_i^2)$ (this happens with weight $\frac{1}{3}$). Therefore, $q_{vy} = \frac{1}{2}\cdot 1 + \frac{1}{2}\cdot\frac{1}{3} = \frac{2}{3}$. The analysis for vx is symmetric. Lastly, consider the edge vz. It appears in $\hat{H}_i^1$ ($\hat{H}_i^2$) if and only if $yz \in E(H_i^1)$ (respectively, $xz \in E(H_i^2)$). Therefore, $q_{vz} = \frac{1}{2}\cdot\frac{2}{3} + \frac{1}{2}\cdot\frac{2}{3} = \frac{2}{3}$. This completes the proof of Theorem 3. □

3 Our Algorithm and the Proof of Theorem 2

In this section we give a proof of Theorem 2 and then use it to obtain a $\frac{4}{3}$-approximation algorithm for 2ECM on half-integral instances (Theorem 1). We apply the same splitting-off theorem of [BJGJS99] together with an induction scheme that is captured in Theorem 2. A key feature of this theorem is that we allow edges of negative cost, although the edge costs in any instance of 2ECM are non-negative.

Consider a 4-regular 4-edge connected multigraph $G = (V, E)$ on n vertices, and let $e = uv$ be an edge in G. Let $c : E \to \mathbb{R}$ be an arbitrary real-valued cost function. Our goal is to obtain a 2-edge connected spanning subgraph H of G whose cost is at most $\frac{2}{3}c(G - e)$ while ensuring that H uses no more than one copy of any multiedge in G.

As alluded to before, for the purposes of obtaining a cheap 2-edge connected subgraph, it suffices to only recurse on one of the two subinstances that arise in the proof of Theorem 3. This insight comes from working backwards from (ConvexComb-G). Since this convex combination for G is a simple average of the convex combinations from the two subinstances (see (ConvexComb-G_1) and (ConvexComb-G_2)), it is judicious to only recurse on the "cheaper" subinstance. Combining (ConvexComb-G_1) and (Lift-G_1), we get that the first subinstance gives rise to a convex combination for $\frac{2}{3}\chi^{E(G)\setminus\{e\}} + \frac{1}{3}(\chi^{\{vy\}} - \chi^{\{vx\}})$. On the other hand, the second subinstance gives rise to a convex combination for $\frac{2}{3}\chi^{E(G)\setminus\{e\}} + \frac{1}{3}(\chi^{\{vx\}} - \chi^{\{vy\}})$. Thus, we should recurse on G_1 if $c_{vx} \geq c_{vy}$, and G_2 otherwise. For the sake of argument, suppose that we are recursing on G_1. So far, we have ignored an important detail in the recursion: the splitting-off operation creates a new edge yz that was not originally present in G, so we need to assign it some cost to apply the algorithm recursively. Depending on how we choose the cost of yz, it might either be included or excluded from the subgraph obtained for the smaller instance, so to bound the cost of the lifted solution we must have a handle on both outcomes of the lift operation. Setting $c_{yz} := c_{vz} - c_{vx}$ balances the cost of both outcomes. Note that c_{yz} could possibly be negative, but this is permissible since the statement of Theorem 2 allows for arbitrary edge costs. We formalize the above ideas.

In the recursive step, we pick one end vertex v of e and apply a complete splitting off operation at v to obtain a 4-regular 4-edge connected graph on $n-1$ vertices; this can be implemented in $O(n)$ time. The running time of the algorithm is $O(n^2)$, since we apply the induction step $O(n)$ times. We remark that the running time of the algorithm in Theorem 2 can be improved to $O(n^{1+o(1)})$; we defer the details to Section 5.

Let $T = \{u, x, y, z\}$ be the four neighbors of v and let $e = uv$. Recall that A_e denotes the set of edges $f \in \delta(v) \setminus \{e\}$ such that (e, f) is an admissible pair (see Definition 6).

Lemma 9. *For $vx \in \delta(v) \setminus \{e\}$, we can check whether $vx \in A_e$ in $O(n)$ time.*

Proof. We may suppose that the elements of the set T of neighbors of v are all distinct. Otherwise, by Lemma 8, we know exactly which pairs are admissible, see Figure 1. Consider the graph $\hat{G} = (G_{u,x})_{y,z}$ obtained by splitting off the pairs (uv, vx) and (yv, vz) at v. Let G^* be the graph obtained from $\hat{G}$ by contracting ux to a single vertex s and contracting yz to a single vertex t. Then we apply a max s-t flow computation to check whether G^* has ≥ 4 edge-disjoint s-t paths; otherwise, G^* has an s-t cut $\delta(S)$ of size ≤ 3. In the latter case, it is clear that our trial splitting is not admissible.

In the former case, we claim that our trial splitting is admissible. Suppose that $\hat{G}$ is not 4-edge connected. Then there exists a non-empty, proper vertex set S in $\hat{G}$ such that $|T \cap S| \leq |T \setminus S|$ and $|\delta_{\hat{G}}(S)| < 4$. Clearly, $|S \cap T| \leq 2$, and if $|S \cap T| = 2$, then we have $|S \cap \{u, x\}| = 1$ and $|S \cap \{y, z\}| = 1$ (otherwise, S would give an (s,t)-cut of G^* of size ≤ 3). Since the size of the cut of S is the same in G and in $\hat{G}$, we have, by 4-edge connectivity of G, $4 > |\delta_{\hat{G}}(S)| = |\delta_G(S)| \geq 4$, a contradiction.

To see that the running time is linear, observe that G^* has $\leq 2n$ edges, an s-t flow of value ≥ 4 can be computed by finding 4 augmenting paths, and each augmenting path can be found in linear time. □

Proof of Theorem 2. First, consider the base case in the recursion when $n = 2$. The only such 4-regular 4-edge connected multigraph is given by four parallel edges between u and v, of which e is one. Picking the two cheapest edges from the remaining three edges gives the desired subgraph.

For the induction step, suppose that $n \geq 3$ and the induction hypothesis holds for all 4-regular 4-edge connected multigraphs with at most $n-1$ vertices and for all choices of edge e. Consider a 4-regular 4-edge connected multigraph G on n vertices and an edge $e = uv$ in G.

Our algorithm proceeds as follows. By Lemmas 8 and 9, we can find in $O(n)$-time two neighbors of v, say x and y, such that $vx, vy \in A_e$ and $c_{vx} \geq c_{vy}$. Next, we construct the graph $\hat{G} := (G_{u,x})_{y,z} = G - v + \{ux, yz\}$ and extend the cost function c to the new edge yz as $c_{yz} := c_{vz} - c_{vx}$ (note that the cost of ux is inconsequential and that c_{yz} may be negative or non-negative). We recursively find a 2-edge connected spanning subgraph $\hat{H}$ of $\hat{G}$ with cost at most $\frac{2}{3}c(\hat{G} - ux)$. Then, we lift $\hat{H}$ to obtain a spanning subgraph H of G:

$$H := \begin{cases} \hat{H} - yz + vy + vz & \text{if } yz \in E(\hat{H}), \\ \hat{H} + vy + vx & \text{if } yz \notin E(\hat{H}). \end{cases}$$

We analyze the cost of this subgraph. Regardless of the cases above, our choice of c_{yz} implies that $c(H) = c(\hat{H}) + c_{vy} + c_{vx}$. Therefore,

$$\begin{aligned} c(H) &\leq \frac{2}{3}c(\hat{G} - ux) + c_{vy} + c_{vx} \\ &= \frac{2}{3}(c(G - e) - c_{vx} - c_{vy} - c_{vz} + (c_{vz} - c_{vx})) + c_{vy} + c_{vx} \\ &= \frac{2}{3}c(G - e) + \frac{1}{3}(c_{vy} - c_{vx}) \leq \frac{2}{3}c(G - e) \ , \end{aligned}$$

where the last inequality follows from our choice of vx, vy to satisfy $c_{vx} \geq c_{vy}$.

It remains to argue that H is a 2-edge connected spanning subgraph of $G - e$ that uses no more than one copy of any multiedge in G. It is clear that the following hold: (a) $e \notin E(H)$; (b) H is a spanning subgraph of G; and (c) each multiedge of G appears at most once in H. Since $\hat{H}$ is 2-edge connected and adding and/or subdividing an edge preserves 2-edge connectivity, H is 2-edge connected. Overall, in $O(n^2)$-time we have constructed a 2-edge connected spanning subgraph H of $G - e$ whose cost is at most $\frac{2}{3}c(G - e)$, thereby proving Theorem 2. □

Using Theorem 2, we give a deterministic $\frac{4}{3}$-approximation algorithm for 2ECM on half-integral instances.

Proof of Theorem 1. Let x be a half-integral solution to (Subtour-LP) (and (2ECM-LP)) for an instance given by the complete n-vertex graph $\overline{G} = (\overline{V}, \overline{E})$ and a metric cost function c. Let $G = (V, E)$ denote the multigraph induced by $2x$ (see the paragraph preceding Theorem 2). By (7) and (6), G is a 4-regular 4-edge connected multigraph. With a slight abuse of notation, we use the same cost function for the edges of E: for any $f \in E$, $c_f := c_e$, where e denotes the edge in $\overline{E}$ that gave rise to f. We invoke Theorem 2 on G and some edge $e \in E$. This gives us a 2-edge connected spanning subgraph H of $G - e$ satisfying $c(H) \leq \frac{2}{3}c(G - e)$. Then $\overline{H} := H$ is a 2-edge connected spanning multisubgraph of $\overline{G}$; note that $\overline{H}$ uses at most two copies of any edge in $\overline{G}$. By the first conclusion of Theorem 2 and the non-negativity of c, $c(\overline{H}) = c(H) \leq \frac{2}{3}c(G - e) \leq \frac{2}{3}c(G) = \frac{4}{3}c^T x$, where the last equality follows by recalling that G is induced by $2x$. Besides invoking Theorem 2 we only perform trivial graph operations so the running time is $O(n^2)$. □

We conclude this section by presenting the following extension of Theorem 1 by applying the results of Carr and Vempala [CV02]: Given a half-integral solution x^* to an instance $(\overline{G}, c)$ of (Subtour-LP), there is a polynomial-time algorithm to find 2-edge connected spanning multisubgraphs $H_1, \ldots, H_k$ of $\overline{G}$ and a convex combination y of the incidence vectors of

$H_1, \ldots, H_k$ such that $y \leq \frac{4}{3}x^*$. The proof strategy of [CV02] relies on the analysis of the ellipsoid method in the rational model by Grötschel et al., [GLS88]; also see [LS11, Lemma 3.4], [CS16, Lemma 3.2], and the first part of [JMS03, Theorem 4.1]. The following proof outline is due to Chaitanya Swamy.

Let Z denote the set of all incidence vectors of 2-edge connected spanning multisubgraphs of $\overline{G} = (\overline{V}, \overline{E})$. As discussed previously, we may assume that for any $z \in Z$ we have $z_e \in \{0,1,2\}$ for all $e \in \overline{E}$, that is, $Z \subset \{0,1,2\}^{\overline{E}}$. Let n denote $|\overline{V}|$.

Consider the following linear program (P) that uses the variables $\lambda \in \mathbb{R}^Z$. Clearly, the number of variables is exponential in n. Note that z_e and x_e^* are constants, for all $z \in Z, e \in \overline{E}$. The dual linear program (D) of (P) uses the variables $w \in \mathbb{R}^{\overline{E}}$ and $q \in \mathbb{R}$.

$$
\begin{array}{llll}
\max & \displaystyle\sum_{z \in Z} \lambda_z, & & \text{(P)} \\
\text{subject to} & \displaystyle\sum_{z \in Z} \lambda_z z_e \leq \frac{4}{3}x_e^* & \forall\, e \in \overline{E}, & \\
& \displaystyle\sum_{z \in Z} \lambda_z \leq 1, & & \\
& \lambda_z \geq 0 & \forall z \in Z. &
\end{array}
$$

$$
\begin{array}{llll}
\min & \displaystyle\frac{4}{3}{x^*}^T w + q & & \text{(D)} \\
\text{subject to} & z^T w + q \geq 1 & \forall\, z \in Z, & \\
& w_e \geq 0 & \forall e \in \overline{E}, & \\
& q \geq 0. & &
\end{array}
$$

Note that (P) has a feasible solution $\lambda = 0$, and the optimal value of (P) is ≤ 1 (by one of the constraints). Moreover, every feasible solution w, q of (D) has objective value $\frac{4}{3}{x^*}^T w + q \geq 1$ (otherwise, by Theorem 1, there exists $z \in Z$ such that $z^T w + q \leq \frac{4}{3}{x^*}^T w + q < 1$; clearly, w, q violates one of the constraints of (D)). By linear programming duality, the optimal value of (P) is 1. Any feasible solution λ of (P) with objective value 1 (together with the associated vectors $\{z \in Z \ : \ \lambda_z > 0\}$) gives the required convex combination (of 2-edge connected spanning multigraphs of $\overline{G}$).

Let $(D(t))$ denote the linear program obtained from (D) by adding the constraint $\frac{4}{3}{x^*}^T w + q \leq t$. Our plan is to use the ellipsoid method with a so-called approximate separation oracle to solve $(D(t))$ for a particular value of t. Recall from Theorem 1 that we have a polynomial-time algorithm $\mathcal{A}$ that, given the inputs x^* and $w \in \mathbb{R}^{\overline{E}}, w \geq 0$, finds $z \in Z$ such that $z^T w \leq \frac{4}{3}{x^*}^T w$. The approximate separation oracle is based on $\mathcal{A}$. Given a candidate solution $\overline{w}, \overline{q}$ for $(D(t))$ (i.e., the center of the ellipsoid), we check whether $\overline{w} \geq 0$ and $\overline{q} \geq 0$ (if not, we find a violated constraint), and then we check the value of $\frac{4}{3}{x^*}^T \overline{w} + \overline{q}$. If this value is $> t$, then the additional constraint is violated. If $\frac{4}{3}{x^*}^T \overline{w} + \overline{q} < 1$, then we use $\mathcal{A}$ to find a violated constraint (of the form $z^T \overline{w} + \overline{q} \geq 1$ for some $z \in Z$). Otherwise, the approximate separation oracle reports that $\overline{w}, \overline{q}$ is feasible for $(D(t))$ (in fact, $\overline{w}, \overline{q}$ may be infeasible).

We pick a small rational number $\epsilon > 0$ (to be fixed below such that $1/\epsilon$ has encoding length polynomial in n, see [GLS88, Chapter 1.3]), and solve the (infeasible) linear program $(D(1-\epsilon))$ by the ellipsoid method using the approximate separation oracle; note that the encoding length of each constraint of $(D(1-\epsilon))$ is polynomial in n. The ellipsoid method runs for a polynomial (in n) number of iterations and deduces that $(D(1-\epsilon))$ is infeasible. Hence, the violated constraints of the form $z^T w + q \geq 1$ found during the run of the ellipsoid method, together with the constraints $w \geq 0, q \geq 0$, and the constraint $\frac{4}{3}{x^*}^T w + q \leq 1 - \epsilon$, yield an infeasible linear program (D').

Let (D'') be the linear program with objective function $\frac{4}{3}x^{*T}w + q$ and the violated constraints of the form $z^T w + q \geq 1$ found during the run of the ellipsoid method, together with the constraints $w \geq 0, q \geq 0$; note that (D'') has an optimal solution since it is feasible (by $\overline{w} = 0, \overline{q} = 1$) and it is not unbounded. The infeasibility of (D') implies that the optimal value τ'' of (D'') is $> 1 - \epsilon$. A key point is to choose ϵ to be sufficiently small (while ensuring the required bound on the encoding length of $1/\epsilon$) to ensure that τ'' is ≥ 1 (then, it must be 1, since $\overline{w} = 0, \overline{q} = 1$ has objective value 1). Observe that the objective function of (D'') and each of its constraints has encoding length $O(n^2)$. Hence, the denominator Δ of τ'' has encoding length polynomial in n. We compute an upper bound $\hat{\Delta}$ for Δ (such that the encoding length of $\hat{\Delta}$ is polynomial in n), and fix $\epsilon = 1/(2\hat{\Delta})$; then, $\tau'' > 1 - \epsilon$ implies that $\tau'' \geq 1$.

Thus, (D'') has a polynomial (in n) number of constraints, and these constraints suffice to certify that (D'') has the same optimal value as (D). The dual linear program (P'') of (D'') has a polynomial (in n) number of variables, and, by the duality theorem, (P'') has an optimal solution λ'' with objective value 1. Moreover, λ'' can be computed in time polynomial in n. Clearly, λ'' maps to a feasible solution of (P) with objective value 1, and gives the required convex combination (of 2-edge connected spanning multigraphs of $\overline{G}$).

Proposition 10. *Let $\overline{G} = (\overline{V}, \overline{E})$ be a complete graph on n vertices. Let $x^* \in \mathbb{R}^{\overline{E}}_{\geq 0}$ be a half-integral solution to* (Subtour-LP) *(or equivalently,* (2ECM-LP)*). Using the results of [CV02], in time polynomial in n, we can obtain 2-edge connected spanning multisubgraphs $H_1, \ldots, H_k$ of $\overline{G}$ and nonnegative real numbers λ_j, $j = 1, \ldots, k$, satisfying $\sum_{j=1}^{k} \lambda_j = 1$ and $\sum_{j=1}^{k} \lambda_j \chi^{H_j} \leq \frac{4}{3}x^*$.*

4 A randomized algorithm for 2-edge connected spanning subgraphs of 3-regular 3-edge connected graphs

Let $G = (V, E)$ be a 3-regular, 3-edge connected graph on n vertices. In this section we give a randomized algorithm (see Theorem 12) that produces a random 2-edge connected spanning subgraph H of G such that $\mathbb{E}\left[\chi^{E(H)}\right] = \frac{4}{5}\chi^{E(G)}$ and the expected running time is polynomial in n. Observe that for any edge-cost function $c : E \to \mathbb{R}$, we can use the above algorithm to obtain a random 2-edge connected spanning subgraph H satisfying $\mathbb{E}\left[c(H)\right] = \frac{4}{5}c(G)$.

First, we present an existence result due to Boyd and Legault [BL17], see Proposition 11. We include a proof for the sake of exposition and self-containedness. This particular proof appears in [Leg17b] and is needed for the design of our randomized algorithm.

Proposition 11 ([BL17],[Leg17b])**.** *Let $G = (V, E)$ be a 3-regular, 3-edge connected graph. There exists a finite collection of 2-edge connected spanning subgraphs $\{H_1, \ldots, H_k\}$ and nonnegative numbers $\mu_1, \ldots, \mu_k$ with $\sum_i \mu_i = 1$ that satisfy:*

$$\sum_{i=1}^{k} \mu_i \chi^{E(H_i)} = \frac{4}{5}\chi^{E(G)} .$$

Proof. Let $n := |V|$ denote the number of vertices in G. Note that n is even. We prove the theorem via induction on n. The induction hypothesis holds for the base case when $n = 2$:

suppose e_1, e_2, e_3 are the 3 parallel edges between $u, v \in V$, then

$$\frac{4}{5}\chi^{\{e_1,e_2,e_3\}} = \frac{1}{5}\chi^{\{e_1,e_2\}} + \frac{1}{5}\chi^{\{e_1,e_3\}} + \frac{1}{5}\chi^{\{e_2,e_3\}} + \frac{2}{5}\chi^{\{e_1,e_2,e_3\}} \ .$$

Suppose that the induction hypothesis holds for all such graphs on fewer than n vertices for some $n \geq 4$. Consider a 3-regular, 3-edge connected graph G on n vertices. We consider two cases depending on whether G has a nontrivial tight cut or not.

(a) **There exists $S \subsetneq V, |S| \notin \{1, n-1\}$ such that $|\delta(S)| = 3$.** Let $\delta(S) = \{e_1, e_2, e_3\}$. Consider the following two smaller 3-regular, 3-edge connected subgraphs: the graph $G_1 = (S \cup \{\overline{s}\}, E_1)$ is obtained by contracting $V \setminus S$ to a single vertex $\overline{s}$, and the graph $G_2 = ((V \setminus S) \cup \{s\}, E_2)$ is obtained by contracting S to a single vertex s; we discard all loops that arise due to contraction. Note that edges in $\delta(S)$ are in one-to-one correspondence with edges in $\delta(\overline{s})$ (in G_1) and $\delta(s)$ (in G_2), so we overload the notation and use $\delta(\overline{s})$ and $\delta(s)$ instead of $\delta_{G_1}(\overline{s})$ and $\delta_{G_2}(s)$, respectively. By induction, for each $j \in \{1, 2\}$ there exist 2-edge connected spanning subgraphs $\{H_1^j, \ldots, H_{k_j}^j\}$ and nonnegative numbers $\mu_1^j, \ldots, \mu_{k_j}^j$ with $\sum_i \mu_i^j = 1$ that satisfy:

$$\sum_{i=1}^{k_j} \mu_i^j \chi^{E(H_i^j)} = \frac{4}{5}\chi^{E(G_j)} \ .$$

A simple counting argument shows that for any $j \in \{1, 2\}$ and $F \subseteq \{e_1, e_2, e_3\}$, we have

$$q_F := \sum_{\substack{i=1,\ldots,k_j: \\ E(H_i^j) \cap \{e_1,e_2,e_3\} = F}} \mu_i^j = \begin{cases} \frac{1}{5} & \text{if } |F| = 2, \\ \frac{2}{5} & \text{if } |F| = 3, \\ 0 & \text{if } |F| = 0, 1 \ . \end{cases}$$

Consider the following gluing operation: if H_1' and H_2' denote 2-edge connected spanning subgraphs of G_1 and G_2, respectively, that agree on the cut $\delta(S)$, then $\mathsf{glue}(H_1', H_2') := (V(G), (E(H_1') \setminus \delta(\overline{s})) \cup (E(H_2') \setminus \delta(s)) \cup F)$, where F denotes the edges that are common to H_1' and H_2' on the cut $\delta(S)$. Since H_1' and H_2' are 2-edge connected, it follows that $\mathsf{glue}(H_1', H_2')$ will also be 2-edge connected. By the above, the convex combinations for G_1 and G_2 can be glued together consistently to obtain a convex combination for G.

(b) **For all $S \subsetneq V, |S| \notin \{1, n-1\}$, we have $|\delta(S)| \geq 4$.** For any $e = uv \in E(G)$, let $G^e := G - u - v + ab + pq$, where a, b (and v) are the neighbors of u, and p, q (and u) are the neighbors of v. In other words, G^e is obtained by deleting vertices u and v from G followed by including edges ab and pq. It is easy to see that G^e is a 3-regular graph on $n-2$ vertices. Furthermore, since G does not have any nontrivial tight cuts, G^e is clearly 3-edge connected. By induction, there exist 2-edge connected subgraphs $H_1^e, \ldots, H_{k_e}^e$ and nonnegative numbers $\mu_1^e, \ldots, \mu_{k_e}^e$ with $\sum_i \mu_i^e = 1$ that satisfy $\sum_{i=1}^{k_e} \mu_i^e \chi^{E(H_i^e)} = \frac{4}{5}\chi^{E(G^e)}$. Consider the following *lifting* operation that takes a 2-edge connected spanning subgraph H^e of G^e and lifts it to a 2-edge connected spanning subgraph of G: for any such H^e, $\mathsf{lift}_e(H^e) := (V(H^e) \cup \{u, v\},\ E(H^e) \setminus \{ab, pq\}) \cup \{au, bu, pv, qv\}$. In other words, the lifting operation can be seen as including the edge ab and pq, if they are not already present in

H^e, and then subdividing them at u and v, respectively. With this interpretation, it is straightforward to see that $\mathsf{lift}_e(H^e)$ is 2-edge connected. It follows that

$$\sum_{i=1}^{k_e} \mu_i^e \chi^{E(\mathsf{lift}_e(H_i^e))} = \frac{4}{5}\chi^{E(G)} + \frac{1}{5}\chi^{\{au,bu,pv,qv\}} - \frac{4}{5}\chi^{\{uv\}} .$$

Since every edge in G has 4 adjacent edges, the required convex combination for G can be obtained by averaging over the convex combinations for G^e for all $e \in E(G)$:

$$\sum_{e\in E(G)} \sum_{i=1}^{k_e} \frac{\mu_i^e}{|E(G)|} \chi^{E(\mathsf{lift}_e(H_i^e))} = \frac{4}{5}\chi^{E(G)} .$$

□

Las Vegas algorithm. We now describe an algorithm $\mathcal{A}$ that can be seen as a Las Vegas algorithm that efficiently implements the proof of Proposition 11. The algorithm $\mathcal{A}$ takes as input a 3-regular, 3-edge connected graph $G = (V, E)$ on n vertices and produces a random 2-edge connected spanning subgraph H of G such that $\mathbb{E}\left[\chi^{E(H)}\right] = \frac{4}{5}\chi^{E(G)}$. At a high level, the algorithm is recursive and follows the strategy in Proposition 11. The base of the recursion corresponds to $n = 2$, where the input G consists of three parallel edges e_1, e_2, e_3. The edge-set of the output follows the distribution given by: $\{e_1, e_2, e_3\}$ with probability $\frac{2}{5}$, and each of $\{e_1, e_2\}$, $\{e_1, e_3\}$, and $\{e_2, e_3\}$ with probability $\frac{1}{5}$. In the general case with $n > 2$ (note that n is even), the algorithm checks if there are any nontrivial tight cuts in G. (We run $O(n)$ max s-t flow computations with a fixed source vertex s and each of the other vertices as the sink t; for a chosen vertex t, if there exists a nontrivial tight s-t cut $\delta(S)$, then s has a neighbor in S and t has a neighbor in $(V \setminus S)$; using this fact, we can find such an s-t cut by running $O(1)$ max flow computations.) If there exists a cut $S \subsetneq V$, $1 < |S| \leq n/2$ with $|\delta(S)| = 3$, then we construct two smaller instances of the problem by contracting each of the two shores $V \setminus S$ and S, respectively; this is exactly the same construction that we used in the proof of Proposition 11. Let $G_1 = (S \cup \{\overline{s}\}, E_1)$ denote the 3-regular, 3-edge connected graph on $|S|+1$ vertices obtained by contracting $V \setminus S$ to a vertex $\overline{s}$, and let $G_2 = ((V \setminus S) \cup \{s\}, E_2)$ denote the 3-regular, 3-edge connected graph on $n-|S|+1$ vertices obtained by contracting S to a vertex s. We run $\mathcal{A}$ on G_2 to obtain a 2-edge connected spanning subgraph H_2 of G_2. Let $F \subseteq \delta(S)$ denote the edges in G corresponding to the edges in $E(H_2) \cap \delta(s)$. We repeatedly run $\mathcal{A}$ on G_1 until it gives a 2-edge connected spanning subgraph H_1 of G_1 such that H_1 agrees with F, i.e., F is the set of edges in G corresponding to the edges in $E(H_1) \cap \delta(\overline{s})$. The output of $\mathcal{A}$ on G is $\mathsf{glue}(H_1, H_2) = (V(G), (E(H_1) \setminus \delta(s)) \cup (E(H_2) \setminus \delta(\overline{s})) \cup F)$. On the other hand, if there are no nontrivial tight cuts in G we choose an edge $e = uv \in E$ uniformly at random and construct the graph $G^e := G - u - v + ab + pq$, where a, b (and v) are the neighbors of u, and p, q (and u) are the neighbors of v. Let H^e be the random 2-edge connected spanning subgraph of G^e obtained by running $\mathcal{A}$ on G^e. The output of $\mathcal{A}$ on G is $\mathsf{lift}_e(H^e) = (V(G), (E(H^e) \setminus \{ab, pq\}) \cup \{au, bu, pv, qv\})$. The following result shows the correctness and efficiency of $\mathcal{A}$.

Theorem 12. *Let $G = (V, E)$ be a 3-regular, 3-edge connected graph on n vertices. Let H denote the random 2-edge connected spanning subgraph obtained by running algorithm $\mathcal{A}$ on G. Then,*

1. $\mathbb{E}\left[\chi^{E(H)}\right] = \frac{4}{5}\chi^{E}$; *and*

2. *The expected running time $T(G)$ of the algorithm is polynomial in n.*

Proof. We prove the theorem via induction on the number n of vertices. The base case when $n = 2$ is trivial. Observe that $\mathbb{E}\left[\chi^{E(H)}\right] = \frac{4}{5}\chi^{E(G)}$. Suppose that for some even $n > 2$ the induction hypothesis (i.e., the theorem statement) holds for all instances of the problem with fewer than n vertices. The following claim will be useful. (Recall that q_F, for $F \subseteq E$, is defined in the proof of Proposition 11.)

Claim 13. *Let G and H be as defined in Theorem 12. Let $\delta(S) = \{e_1, e_2, e_3\}$ be a tight cut in G (possibly, $|S| \in \{1, |V| - 1\}$). If Theorem 12 holds for all such graphs on at most n vertices, then for any $F \subseteq \{e_1, e_2, e_3\}$, we have $\Pr[E(H) \cap \delta(S) = F] = q_F$. (Recall that $q_{\{e_1,e_2\}} = q_{\{e_1,e_3\}} = q_{\{e_2,e_3\}} = \frac{1}{5}$ and $q_{\{e_1,e_2,e_3\}} = \frac{2}{5}$.)*

For the general case with $n > 2$, recall that $\mathcal{A}$ checks whether G has a nontrivial tight cut or not.

(a) If there exists $S, 1 < |S| \le n/2$ with $|\delta(S)| = 3$, we obtain two smaller instances G_1 and G_2 by contracting $V \setminus S$ and S, respectively. Let $\delta(S) = \{e_1, e_2, e_3\}$. Running $\mathcal{A}$ on G_2 gives a random 2-edge connected spanning subgraph H_2 satisfying $\mathbb{E}\left[\chi^{E(H_2)}\right] = \frac{4}{5}\chi^{E(G_2)}$; the expected running time of this step is $T(G_2)$. Then, the algorithm stipulates that we repeatedly run $\mathcal{A}$ on G_1 until we obtain a 2-edge connected spanning subgraph H_1 such that H_1 and H_2 agree on the cut $\delta(S)$. Since each run of $\mathcal{A}$ on G_1 takes expected $T(G_1)$-time, we get $T(G) \le T(G_2) + \kappa\, T(G_1)$, where κ denotes the expected number of runs of $\mathcal{A}$ (on G_1) that are needed for H_1 to agree with H_2 (on the cut $\delta(S)$). For notational convenience, for each $i \in \{1, 2\}$ let F_i denote $E(H_i) \cap \delta(S)$. The following calculation shows that $\kappa = 4$.

$$\begin{aligned}
\kappa &= \sum_{\substack{F\subseteq\{e_1,e_2,e_3\},\\ |F|=2,3}} \Pr[F_2 = F] \cdot \mathbb{E}[\#\text{ runs to get } F_1 = F_2 | F_2 = F] \\
&= \sum_{\substack{F\subseteq\{e_1,e_2,e_3\},\\ |F|=2,3}} q_F \cdot \mathbb{E}[\#\text{ runs to get } F_1 = F] \qquad \text{(by independence)} \\
&= \sum_{\substack{F\subseteq\{e_1,e_2,e_3\},\\ |F|=2,3}} q_F \cdot \frac{1}{\Pr[F_1 = F]} \\
&= \sum_{\substack{F\subseteq\{e_1,e_2,e_3\},\\ |F|=2,3}} 1 \qquad = 4 \qquad \text{(by Claim 13, } q_F \text{ is independent of the graph).}
\end{aligned}$$

Overall, the expected running time in this case is at most $4 \cdot T(G_1) + T(G_2)$; we show that this bound is polynomial in n shortly. Next, we argue that $\mathbb{E}\left[\chi^{E(H)}\right] = \frac{4}{5}\chi^{E}$ holds. Since $\mathbb{E}\left[\chi^{E(H_2)}\right] = \frac{4}{5}\chi^{E(G_2)}$, it is easy to see that we get $\frac{4}{5}$, in expectation, on each edge in $(E(G_2) \setminus \delta(s)) \cup \delta(S)$. It remains to argue that the same guarantee holds for every edge in $(E(G_1) \setminus \delta(\overline{s}))$. For each $F \subseteq \{e_1, e_2, e_3\}, |F| \in \{2, 3\}$, let $y^F \in \mathbb{R}^{E(G_1)}_{\ge 0}$ denote the vector $\mathbb{E}\left[\chi^{E(H_1)}|F_1 = F\right]$, i.e., the expected characteristic vector of $E(H_1)$ conditioned on the event $\{F_1 = F\}$. By induction, Claim 13 holds for G_1 so we get:

$$\frac{4}{5}\chi^{E(G_1)} = \sum_{\substack{F\subseteq\{e_1,e_2,e_3\}:\\ |F|\in\{2,3\}}} q_F \cdot y^F \ .$$

If the output H_2 of running $\mathcal{A}$ on G_2 produces the pattern $F_2 = E(H_2) \cap \delta(S)$, then, conditioned on this event, the expected characteristic vector of the subgraph H_1 that is glued with H_2 is y^{F_2}. Since $\Pr\left[E(H_2) \cap \delta(S) = F\right] = q_F$, we get

$$\begin{aligned}\mathbb{E}\left[\chi^{E(\mathsf{glue}(H_1,H_2))\cap E(G_1)}|F_1 = F_2\right] &= \sum_{\substack{F\subseteq\{e_1,e_2,e_3\}:\\ |F|\in\{2,3\}}} \Pr\left[F_2 = F\right] \cdot \mathbb{E}\left[\chi^{E(H_1)}|F_1 = F\right] \\ &= \sum_{\substack{F\subseteq\{e_1,e_2,e_3\}:\\ |F|\in\{2,3\}}} q_F \cdot y^F = \frac{4}{5}\chi^{E(G_1)} \ .\end{aligned}$$

(b) If there are no nontrivial tight cuts in G, then the algorithm $\mathcal{A}$ chooses an edge $e \in E(G)$ uniformly at random, runs $\mathcal{A}$ on the smaller instance G^e (on $n-2$ vertices), and lifts the random output H^e to a random 2-edge connected spanning subgraph $H = \mathsf{lift}_e(H^e)$ of G. Following the proof of Proposition 11, $\mathbb{E}\left[\chi^{E(H)}\right] = \frac{4}{5}\chi^{E(G)}$. The expected running time $T(G)$ in this case is $\frac{1}{|E(G)|}\sum_{e\in E(G)} T(G^e)$.

We finish the proof by showing that $T(G)$ is bounded by a polynomial in n. With some abuse of notation, let $T(n)$ denote the maximum of $T(G)$ over all such graphs G on n vertices. We prove that $T(n)$ is polynomial in n. By the above case analysis,

$$T(n) \leq \max_{1<k\leq n/2, k \text{ odd}} \big(T(n-k+1) + 4\,T(k+1)\big) + O(n^2)$$

where the $O(n^2)$ term accounts for finding a nontrivial tight cut, if one exists, and for gluing/lifting the subgraphs obtained from smaller instances. Note that in the maximum shown above we require that k is odd. This is because $|\delta(S)|$ and $|S|$ have the same parity in a 3-regular graph. Clearly, the recursive inequality is satisfied by taking $T(n)$ to be $O(n^3)$, thus giving an (not necessarily tight) upper bound on the expected running time. □

5 Improving the running time in Theorem 2 via dynamic graph algorithms

Eppstein et al., [EGIN97] and Thorup [Tho01] have presented algorithms for maintaining the size k of a min-cut for a fully-dynamic graph in $\tilde{O}(\sqrt{n})$ time per edge insertion or deletion, for k upper-bounded by a poly-logarithmic function of n. (In this section, we use the term graph rather than multigraph. The $\tilde{O}()$ notation hides poly-logarithmic factors, i.e., $\tilde{O}(f(n)) = O(f(n)\log^j n)$ for a positive integer $j = O(1)$.) Recently, Jin and Sun [JS20] have reported substantial improvements from $\tilde{O}(\sqrt{n})$ time per operation to $n^{o(1)}$ time per operation for

$k = (\log n)^{o(1)}$; [JS20, Section 2, p. 11] lists four operations: edge insertion, edge deletion, insertion of an isolated vertex, deletion of an isolated vertex. Theorem 1.1 of [JS20] states: There is a deterministic fully dynamic k-edge connectivity algorithm on a graph of n vertices and m edges with $m^{1+o(1)}$ preprocessing time and $n^{o(1)}$ update and query time for any positive integer $k = (\log n)^{o(1)}$.

Recall that each iteration of the algorithm in Section 3 starts with a 4-regular 4-edge connected graph and applies one or two complete splittings at a chosen vertex; clearly, each graph $\hat{G}$ that occurs in an execution of the algorithm is Eulerian, hence, we can determine whether $\hat{G}$ is 4-edge connected or not via a 3-edge connectivity query. We improve the running time of the algorithm in Section 3 by $O(n)$ applications of the following three macro operations:

- given a vertex v and an edge uv incident to v, apply a (tentative) first complete splitting at v;
- query whether or not the resulting graph $\hat{G}$ is 3-edge connected;
- if $\hat{G}$ is not 3-edge connected, then "undo" the first complete splitting at v, and apply a different complete splitting at v (the resulting graph is guaranteed to be 4-edge connected, see Lemma 7).

Thus, the revised algorithm applies $O(n)$ macro operations, and each of these runs in time $n^{o(1)}$; the running time for all other steps is $O(n^{1+o(1)})$. Hence, the overall running time is $O(n^{1+o(1)})$, compared to the $O(n^2)$ running time of the algorithm in Section 3.

Proposition 14. *Using the results of [JS20], the algorithm for Theorem 2 can be implemented to run in time $O(n^{1+o(1)})$.*

We note that the results in the older papers [EGIN97, Tho01] also give improvements on the $O(n^2)$ running time of the algorithm in Section 3. The earlier paper [EGIN97] has the following setting. Vertex insertion/deletion is not allowed explicitly. Moreover,

- there is a fixed vertex set V (let n denote $|V|$);
- edges can be inserted or deleted;
- queries for (global) 3-edge connectivity can be made.

Theorem 5.4 of Eppstein et al., [EGIN97], states: The 4-edge connected components of an undirected graph can be maintained in time $O(n^{2/3})$ per update and $O(n^{2/3})$ per query.

In our algorithm in Section 3, in each iteration, we pick the designated edge e of the current graph, but we are free to apply complete splitting at either end vertex of e. Now, we will exploit this property. At the start of the algorithm, we fix a designated vertex v^* of the given 4-regular 4-edge connected graph $G = (V, E)$. Our plan is to avoid applying any splitting-off operation at v^*. Moreover, whenever we apply complete splitting at some vertex v, then instead of deleting v, we attach v to v^* by four parallel edges; thus the graph induced by V stays 4-edge connected and all vertices except v^* have degree four (in general, the graph induced by V need not be 4-regular; nevertheless, Lemma 7 applies to the graph).

Observe that in each iteration, we apply $O(1)$ edge insertions/deletions and $O(1)$ queries for 3-edge connectivity. The overall running time of the revised algorithm is $O(n^{5/3})$, since the time bound in [EGIN97] for each of these operations is $O(n^{2/3})$, and the running time for all other steps is $O(n^{5/3})$.

Proposition 15. *Using the results of [EGIN97], the algorithm for Theorem 2 can be implemented to run in time $O(n^{5/3})$.*

Acknowledgments

We thank several colleagues for their comments and valuable feedback, and we thank Chaitanya Swamy for discussions on [CV02].

References

[ABEM06] Anthony Alexander, Sylvia Boyd, and Paul Elliott-Magwood. On the integrality gap of the 2-edge connected subgraph problem. *Technical Report TR-2006-04, SITE, University of Ottawa, Ottawa, Canada*, 2006.

[BFS16] Sylvia Boyd, Yao Fu, and Yu Sun. A 5/4-approximation for subcubic 2EC using circulations and obliged edges. *Discrete Applied Mathematics*, 209:48–58, 2016.

[BIT13] Sylvia Boyd, Satoru Iwata, and Kenjiro Takazawa. Finding 2-factors closer to TSP tours in cubic graphs. *SIAM Journal on Discrete Mathematics*, 27(2):918–939, 2013.

[BJGJS99] Jørgen Bang-Jensen, Harold N. Gabow, Tibor Jordán, and Zoltán Szigeti. Edge-connectivity augmentation with partition constraints. *SIAM Journal on Discrete Mathematics*, 12(2):160–207, 1999.

[BL17] Sylvia Boyd and Philippe Legault. Toward a 6/5 bound for the minimum cost 2-edge connected spanning subgraph. *SIAM Journal on Discrete Mathematics*, 31(1):632–644, 2017.

[CR98] Robert D. Carr and R. Ravi. A new bound for the 2-edge connected subgraph problem. In Robert E. Bixby, E. Andrew Boyd, and Roger Z. Ríos-Mercado, editors, *Integer Programming and Combinatorial Optimization, 6th International IPCO Conference*, volume 1412 of *Lecture Notes in Computer Science*, pages 112–125. Springer, Berlin, 1998.

[CS16] Deeparnab Chakrabarty and Chaitanya Swamy. Facility location with client latencies: LP-based techniques for minimum-latency problems. *Math. Oper. Res.*, 41(3):865–883, 2016.

[CV02] Robert D. Carr and Santosh S. Vempala. Randomized metarounding. *Random Structures & Algorithms*, 20(3):343–352, 2002.

[CV04] Robert D. Carr and Santosh S. Vempala. On the Held-Karp relaxation for the asymmetric and symmetric traveling salesman problems. *Mathematical Programming*, 100(3):569–587, 2004.

[EGIN97] David Eppstein, Zvi Galil, Giuseppe F. Italiano, and Amnon Nissenzweig. Sparsification–a technique for speeding up dynamic graph algorithms. *J. ACM*, 44(5):669–696, September 1997.

[Fra11] András Frank. *Connections in Combinatorial Optimization*. Oxford Lecture Series in Mathematics and its Applications, Volume 38. Oxford University Press, 2011.

[GB93] Michel X. Goemans and Dimitris J. Bertsimas. Survivable networks, linear programming relaxations and the parsimonious property. *Mathematical Programming*, 60:145–166, 1993.

[GLS88] Martin Grötschel, László Lovász, and Alexander Schrijver. *Geometric Algorithms and Combinatorial Optimization*, volume 2 of *Algorithms and Combinatorics*. Springer, 1988.

[HN18] Arash Haddadan and Alantha Newman. Efficient constructions of convex combinations for 2-edge-connected subgraphs on fundamental classes. *CoRR*, abs/1811.09906, 2018.

[Jai01] Kamal Jain. A factor 2 approximation algorithm for the generalized Steiner network problem. *Combinatorica*, 21(1):39–60, 2001.

[JMS03] Kamal Jain, Mohammad Mahdian, and Mohammad R. Salavatipour. Packing Steiner trees. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '03, pages 266–274. ACM/SIAM, 2003.

[JS20] Wenyu Jin and Xiaorui Sun. Fully dynamic c-edge connectivity in subpolynomial time. *CoRR*, abs/2004.07650, 2020.

[KKG20] Anna R. Karlin, Nathan Klein, and Shayan Oveis Gharan. An improved approximation algorithm for TSP in the half integral case. In *Proccedings of the 52nd Annual ACM Symposium on Theory of Computing*, STOC 2020, pages 28–39. ACM, 2020.

[Leg17a] Philippe Legault. An upper bound of $7n/6$ for the minimum size 2EC on cubic 3-edge connected graphs. *CoRR*, abs/1706.01609, 2017.

[Leg17b] Philippe Legault. *Towards New Bounds for the 2-Edge Connected Spanning Subgraph Problem*. Masters Thesis. University of Ottawa, Ottawa-Carleton Institute for Computer Science, 2017.

[LS11] Ron Lavi and Chaitanya Swamy. Truthful and near-optimal mechanism design via linear programming. *J. ACM*, 58(6):25:1–25:24, 2011.

[SV14] András Sebő and Jens 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.

[SW90] David B. Shmoys and David P. Williamson. Analyzing the Held-Karp TSP bound: a monotonicity property with application. *Information Processing Letters*, 35(6):281–285, 1990.

[SWvZ14] Frans Schalekamp, David P. Williamson, and Anke van Zuylen. 2-Matchings, the Traveling Salesman Problem, and the subtour LP: A proof of the Boyd-Carr conjecture. *Mathematics of Operations Research*, 39(2):403–417, 2014.

[Tho01] Mikkel Thorup. Fully-dynamic min-cut. In *Proceedings of the Thirty-Third Annual ACM Symposium on Theory of Computing*, STOC '01, pages 224–230. ACM, 2001.

[Wol80] Laurence A. Wolsey. Heuristic analysis, linear programming and branch and bound. *Mathematical Programming Study*, 13:121–134, 1980.