Terminal Backup, 3D Matching, and Covering Cubic Graphs

Elliot Anshelevich* Adriana Karagiozova †

November 2006

Abstract

We define a problem called *Simplex Matching*, and show that it is solvable in polynomial time. While *Simplex Matching* is interesting in its own right as a nontrivial extension of non-bipartite min-cost matching, its main value lies in many (seemingly very different) problems that can be solved using our algorithm. For example, suppose that we are given a graph with terminal nodes, non-terminal nodes, and edge costs. Then, the TERMINAL BACKUP problem, which consists of finding the cheapest forest connecting every terminal to *at least one* other terminal, is reducible to Simplex Matching. Simplex Matching is also useful for various tasks that involve forming groups of at least 2 members, such as project assignment and variants of facility location.

In an instance of *Simplex Matching*, we are given a hypergraph H with edge costs, and edge size at most 3. We show how to find the min-cost perfect matching of H efficiently, if the edge costs obey a simple and realistic inequality that we call the *Simplex Condition.* The algorithm we provide is relatively simple to understand and implement, but difficult to prove correct. In the process of this proof we show some powerful new results about covering cubic graphs with simple combinatorial objects.

*Department of Computer Science, Rensselaer Polytechnic Institute, Troy, NY. Email: eanshel@cs.rpi.edu. Research supported by an NSF Postdoctoral Fellowship in Mathematical Sciences.

†Department of Computer Science, Princeton University, Princeton, NJ. Email: karagioz@cs.princeton.edu. Research supported by NSF ITR grant CCR-0205594, NSF CAREER award CCR-0237113, MSPA-MCS award 0528414, and Moses Charikar's Alfred P. Sloan Fellowship.

1 Introduction

Matching theory, as well as its extensions, is both extremely important and well-studied. Perhaps surprisingly, there still remain basic matching problems that can be solved efficiently, and yet are not solvable using existing matching algorithms and techniques [9, 12, 18, 19]. In this paper, we address one such problem that we call *Simplex Matching* and show how to solve it in polynomial time using an elegant covering argument. While *Simplex Matching* is interesting in its own right as a nontrivial extension of non-bipartite min-cost matching, its main value lies in many (seemingly very different) problems that can be solved using our algorithm. We now give two representative examples of such problems.

Terminal Backup. Consider the following network design scenario. As in the Steiner tree problem, we are given a graph consisting of terminal nodes, non-terminal nodes, and edges with costs c_e. The terminal nodes represent facilities that need to be connected for backup purposes. To do this, we must construct a network of edges so that every facility is connected to at least one other facility. In other words, we need to find a forest of minimum cost such that every component of this forest contains at least two terminals. The facilities connected together can backup their data and if any one facility failed there would be at least one other that contains its data. In [33], Xu et al. discuss applications of this problem beyond simply backing up data, and show how to solve it using Simplex Matching.

Project Assignment. Now consider a teacher with a list of projects for the students, who wants to break the students into groups of at least 2 (and at most k) and assign each group a project (several groups may do the same project as long as they don't work together). Also suppose that there is a function $u(s, p)$ that shows how much a student s likes project p. How should the teacher break the students into groups so that the sum of students' utilities is maximized? This question is a special case of facility location with lower bounds and can be reduced to Simplex Matching. Notice that if the groups were allowed to be of size 1, the optimum would simply assign each student to the project she likes best. If the groups had to be of size exactly 2, this is reducible to non-bipartite matching (notice that not all projects need to be assigned, otherwise this would be easily solvable by a flow argument). And if the group size had to be at least 3, this problem immediately becomes NP-Hard. The variant with group size at least 2 and arbitrary k, however, is reducible to Simplex Matching and so has nontrivial structure that can be exploited to form an efficient algorithm.

Simplex Matching The main focus of this paper lies in providing a polynomial-time algorithm for Simplex Matching, which is a generalization of min-cost non-bipartite matching. It is also a generalization of $\{K_2, K_3\}$-packing (see, e.g., [10, 19]). A lot of work has been devoted to packing of graphs with various subgraphs [3, 8, 10, 11, 18, 24] (for surveys, see [9, 12, 19]). In this context, the standard matching can be thought of as a packing with edges, i.e. a $\{K_2\}$-packing. The study of packing has a lot in common with our work as it deals with nontrivial extensions of matching, often using totally different methods. In [20], Hell and Kirkpatrick gave an elegant algorithm to find the perfect $\{K_2, K_3\}$-packing, and [25] classified some types of packings that can be found efficiently. Their results only held for unweighted graphs, however. Among other things, our algorithm for Simplex Matching gives a simple and intuitive way of finding the best (min-cost) perfect $\{K_2, K_3\}$-packing, even in the weighted case. The algorithm we provide is relatively simple to understand and implement but difficult to prove correct. In the process of this proof we show some powerful new results about covering cubic graphs with simple combinatorial objects.

We now define the Simplex Matching problem. In an instance of Simplex Matching, we are given a hypergraph H containing edges of sizes 2 and 3 with edge costs $c(e)$. Our goal is to find a perfect matching of H with minimum cost. Since H is a hypergraph, a perfect matching here (often referred to as a packing) is simply a collection S of edges in H such that every node of H appears in exactly one edge of S.

This problem is NP-Hard without additional constraints on the costs $c(e)$. To see this, notice that if H contains only edges of size 3 we have exactly 3D-Matching. However, in the applications that the authors are interested in, H satisfies the following extra condition that we call the *Simplex Condition*. It states that for every 3d edge (u_1, u_2, u_3), the corresponding 2d edges also exist (see Figure 1(a)), with the cost relation of

$$c(u_1, u_2) + c(u_1, u_3) + c(u_2, u_3) \le 2 \cdot c(u_1, u_2, u_3)$$

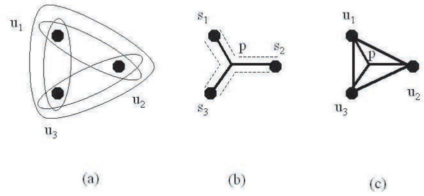

Figure 1: The Simplex Condition in Simplex Matching, Project Assignment, and Terminal Backup.

To understand why the Simplex Condition is natural, consider the Project Assignment problem. It easily reduces to Simplex Matching as follows. First notice that if the optimal solution forms groups larger than 3, then there is an equivalent solution with groups of only 2 and 3 students. Form a graph H with 2d edges (s_1, s_2) representing the smallest cost (negation of utility) of assigning students s_1 and s_2 to a project together, and 3d edges representing the same cost for triples of students. The Simplex Condition will hold in any such H. All possible 2d edges (s_1, s_2) exist and if an edge (s_1, s_2, s_3) corresponds to a project p then $c(s_1, s_2, s_3) = -u(s_1, p) - u(s_2, p) - u(s_3, p)$. Moreover, the inequality $c(s_i, s_j) \le -u(s_i, p) - u(s_j, p)$ holds for any pair of students (s_i, s_j) (Figure 1(b)). This gives us the inequality portion of the Simplex Condition.

The reduction from Terminal Backup to Simplex Matching is more complicated (see [33] for details). First, we can assume that all components of an optimal solution to Terminal Backup are either a path between two terminals, or a star with three terminals as the leaves and a Steiner node at the center. With this assumption we can form a graph H with a 2d edge for each such path and a 3d edge for each such star, and then solve the Terminal Backup problem using the covering version of Simplex Matching. Once again, any such graph H will satisfy the Simplex Condition. Consider an instance of Terminal Backup shown in Figure 1(c). There are three terminals u_1, u_2 and u_3, and a single Steiner node p. Let $c(u_i, u_j)$ be the cost of the cheapest path connecting terminal u_i to terminal u_j and $c(u_1, u_2, u_3)$ be the cost to connect all three terminals through p. Notice that if we only wanted to connect u_i to u_j, we could connect them through p, so we obtain that $c(u_1, u_2) + c(u_1, u_3) + c(u_2, u_3) \le 2\, c(u_1, u_2, u_3)$.

Our Results Our main contribution consists of providing a polynomial-time algorithm for Simplex Matching, which can be used to solve a variety of related problems. The algorithm is very simple conceptually. It starts with a perfect matching (packing) M, and at every step finds an M-alternating 2-factor,[1] such that augmenting M by this 2-factor creates a significantly cheaper perfect matching. It is not surprising that such an algorithm exists, since the min-cost perfect matching can be obtained from any perfect matching if we just augment it by the correct 2-factor. What is surprising here is that a desirable 2-factor can be found efficiently. Most of the paper is devoted to proving this.

Consider how a similar algorithm would behave if we wanted to find the min-cost perfect matching without any edges of size 3. Then any 2-factor is simply a collection of cycles, and could find an alternating cycle that decreases the matching cost sufficiently. In the case of Simplex Matching, however, the 2-factors can have very complex structure (see Figure 2) and finding a good M-alternating 2-factor may seem difficult.

To get around this problem, we show that there is no need to consider arbitrary 2-factors like the one in Figure 2, as there always exist good 2-factors with simple structure (containing at most

[1]Recall that a 2-factor is a subgraph with every node in this subgraph having exactly 2 incident edges.

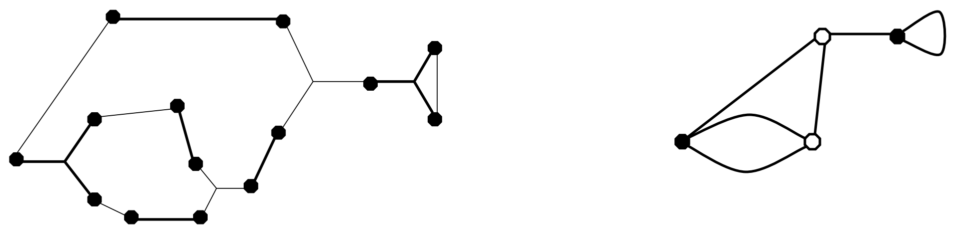

Figure 2: (Left) An M-alternating 2-factor. The bold edges are edges in M. 3d edges are drawn as a star with 3 leaves (i.e., the nodes in the middle of these stars are not real nodes). (Right) The dual of this 2-factor (see Section 3). Filled circles are nodes in M.

two 3d edges), even in the weighted case. The proof of this is complex and relies on our theorem about covering arbitrary cubic graphs with simple combinatorial objects we call *dual augmentors*. For discussion on the relationship between our results and other covering results, especially cycle covers [21, 22, 31, 34], see Section 4.

Related Work Terminal Backup is similar to many Steiner-tree variations [7, 14, 17]. However, all such variations are required to either connect particular pairs of terminals, connect terminals from a particular set, or connect at least k terminals in total. The problem of finding the cheapest forest with at least k terminals in *each* component has not been addressed before. In addition, all of the above variations are NP-Hard while Terminal Backup is solvable in polynomial-time for $k = 2$. For $k > 2$ it becomes NP-Hard, although there is a 2-approximation algorithm using [15].

Simplex Matching and especially Project Assignment are also very similar to variants of facility location. In fact, we can use Simplex Matching to solve instances of facility location where all open facilities have lower bounds of 2 and the facility costs obey the Simplex Condition (e.g., costs are all 0 or the cost of serving three clients is at least twice the cost of serving two clients). Although this is a very special case of facility location, it is the first result (to our knowledge) of a non-trivial facility location problem with lower bounds [1, 16, 23] that can be solved efficiently.

Matching theory is a very large field (see e.g. [26]), and there are many algorithms for weighted non-bipartite matching. A lot of work has also been done on exact *packings*, which are exact covers of a graph using more complicated combinatorial structures than just edges. For some results on packings, see e.g., [3, 8, 10, 11, 18, 24, 27], for surveys see [9, 12, 19]. Especially relevant to our work is packing by edges and triangles ($\{K_2, K_3\}$ packing), since choosing a 3d edge in Simplex Matching is similar to choosing a triangle for a packing. Hell and Kirkpatrick's algorithm for finding the perfect $\{K_2, K_3\}$ packing in unweighted graphs (in [20]) can easily be extended to solve the unweighted version of Simplex Matching. The weighted case is significantly more complicated, however, and cannot be solved by any simple extension of the unweighted algorithm. Since Simplex Matching is a generalization of $\{K_2, K_3\}$ packing, our algorithm can also be used to find the min-cost $\{K_2, K_3\}$ perfect packing. Another relevant line of research is packing by cycles of length *at least* three [4, 5].

Much of the literature on packing concerns itself with matching polyhedra. Unlike the standard perfect matching, which has a nice characterization as a linear program, many similar results for packing can be extremely complicated. While the weighted perfect matching problem lends itself to a primal-dual algorithm, this is not true with Simplex Matching, for which there is no nice linear program characterizing the solutions. See [9, 12, 19, 25] for polytope characterizations of other packing problems, although most of these are either for unweighted versions, or for packing problems very different from ours. An exception is Gyula Pap, who produces some results similar to ours in [28], although he uses completely different techniques and looks at this problem from quite a different perspective.

Finally, [33] is a companion paper to this one. In it the reader will find detailed applications of Simplex Matching, much discussion of the Terminal Backup problem, and an implementation of the Simplex Matching algorithm.

2 Algorithm for Weighted Simplex Matching

For the standard 2d matching we know that if we take a perfect matching M that is not the minimum-weight one, then there exists an alternating cycle that could be used to improve the current matching. We now show a similar condition for Simplex Matching.

Let M be a perfect matching of cost $\sum_{e\in M} c(e)$. For any set of edges S, define a potential function $\phi_M(S) = \sum_{e\in M\cap S} c(e) - \sum_{e\in S-M} c(e)$. If we augment M by S by replacing all edges in $M \cap S$ with the edges in $S - M$, the cost of the new set decreases by $\phi_M(S)$. Moreover, if S is an M-alternating 2-factor then this is still a perfect matching. (An M-alternating 2-factor is a set S such that every node in S has exactly two edges incident to it, exactly one of which is in M.) Let M^* be a minimum cost perfect matching. The components of the symmetric difference $M \oplus M^*$ are M-alternating 2-factors that augment M to M^*. Therefore, there always exists an M-alternating 2-factor S with $\phi_M(S) = \mathit{cost}(M) - \mathit{cost}(M^*)$.

If we could find the M-alternating 2-factor with maximum ϕ_M, we could simply augment by it and get the min-cost perfect matching. Instead, our algorithm will proceed by finding an M-alternating 2-factor S with high $\phi_M(S)$ at each step, and augmenting by it. Finding a 2-factor S with a high potential $\phi_M(S)$ seems difficult, since 2-factors for Simplex Matching can have a complex structure, as in Figure 2. We will show, however, that there is no need to consider arbitrary 2-factors like the one in Figure 2, because there always exists a good 2-factor with simple structure: it should contain at most two 3d edges. We call such 2-factors *augmentors*. More specifically, augmentors can be of the following types (see Figure 3):

Type-0: A Type-0 augmentor is an M-alternating cycle of 2d edges. This is the same as a 2d matching augmenting cycle.

Type-1: A Type-1 augmentor consists of two 3d edges (a_1, a_2, a_3) and (b_1, b_2, b_3) together with M-alternating paths of 2d edges connecting a_1 to b_1, a_2 to a_3 and b_2 to b_3. These paths must be disjoint and the entire augmentor must be M-alternating (so the 3d edges may or may not be in M).

Type-2: Same as Type-1, but the paths connect a_1 to b_1, a_2 to b_2 and a_3 to b_3.

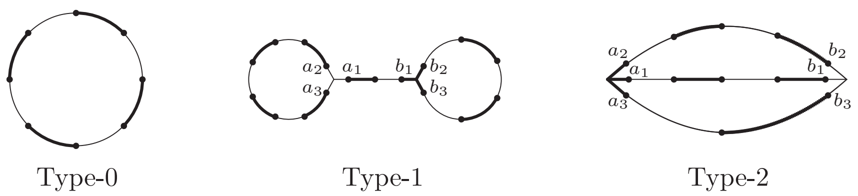

Figure 3: Simplex Matching Augmentors

The bulk of this paper is devoted to proving that an augmentor with high potential always exists. The following lemma allows us to claim that if this is true, then we can improve the current perfect matching in polynomial time. As with most of our results, we only provide a quick proof sketch here, with the full proof located in the Appendix.

Lemma 2.1 *Let A be an augmentor with maximum potential. We can find an M-alternating 2-factor S with $\phi_M(S) \geq \phi_M(A)$ in polynomial time (which may or may not be A itself).*

Proof Sketch: For every pair of of 3d edges, try removing them from the graph, and finding the min-cost 2d-edge matching. This matching together with the 3d edges should be cheaper than any augmentor containing this pair of 3d edges. We can make this process run much faster by considering only certain pairs of 3d edges. For other improvements, see Section 5, and [33]. □

Using the above lemma, we can state our algorithm for finding the minimum-cost weighted Simplex Matching:

```
Start with any perfect matching (possibly containing 3d edges).
Repeat until done
  Find a 2-factor better than any augmentor, and augment by it.
```

3 Dual Augmentors

We establish the running time of the above algorithm in Section 5, and now focus on its correctness and termination. To prove this, we need to show that for any perfect matching M that is not of minimum cost there exists an augmentor A with $\phi_M(A) > 0$. We will accomplish this by showing that every M-alternating 2-factor of positive potential contains an augmentor of positive potential.

For any S we form a dual graph $\overline{S}$ that is easier to deal with than S (see Figures 2 and 4). First, contract all 2d edges (u, v) of S such that u and v are not part of the same 3d edge. Then replace each 3d edge e with a node v_e. Form an edge between the new nodes if the 3d edges that produced them were adjacent. Note that this may result in parallel edges as well as self-loops (if both (u, v, w) and (v, w) were in S). The resulting graph $\overline{S}$ is a cubic (3-regular) graph. We will say that a node $v \in \overline{S}$ is in M if its corresponding 3d edge of S is in M.

Let S_{extra} be the multiset of 2d edges (u, v) such that a 3d edge (u, v, w) is in S, but $(u, v) \notin S$, as in Figure 4. Let $S' = S \cup S_{extra}$. We will associate a unique object from $\overline{S}$ that we call a *dual augmentor* with each augmentor A in S'. The dual augmentor is simply the subgraph of $\overline{S}$ corresponding to the dual of A, as shown in Figure 4. We explore exactly what this means below, but the reader can look at Lemma 3.1 or Figure 5 for a compact definition of a dual augmentor. Dual augmentors are simply one of the structures in Figure 5, with nodes of M having degree 1 or 3 (not 2).

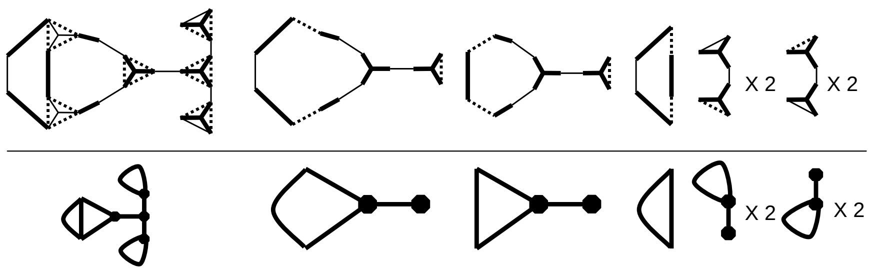

Figure 4: (Top) An M-alternating 2-factor S with edges of S_{extra} shown as dashed lines. To the right of it are some augmentors. (Bottom) The dual cubic graph $\overline{S}$, and the corresponding dual augmentors. The nodes of M as shown as black circles.

Type-0: A Type-0 augmentor A of S' is an M-alternating cycle of 2d edges, some of which are in S_{extra}, like the third augmentor shown in Figure 4. After contracting all 2d edges of S, A becomes a cycle of S_{extra} edges. In the dual graph $\overline{S}$, A corresponds to a cycle of nodes v_e where each e is the 3d edge that produced one of the above S_{extra} edges. We refer to these cycles as dual augmentors of Type-0. As A is M-alternating, it cannot be that any of the above v_e is in M, since then $e \in M$, which would mean that the endpoints of some S_{extra} edge in A are contained in two edges of M.

Type-1: A Type-1 augmentor A will produce a dual augmentor of one of three kinds, as shown in Figure 5. The M-alternating paths connecting a_1 to b_1, a_2 to a_3, and b_2 to b_3 behave exactly as the cycle augmentor in the previous case, namely they correspond to paths that do not contain vertices of M. This gives us a dual augmentor of Type-1c in Figure 5 that is a path together with the two cycles attached to it, with only nodes $v_{(a_1,a_2,a_3)}$ and $v_{(b_1,b_2,b_3)}$ possibly being in M.

Notice that if both (a_1, a_2, a_3) and (a_2, a_3) are in A, then the "cycle" incident to $v_{(a_1,a_2,a_3)}$ in the dual augmentor is just a self-loop. Consider the special case, however, when $(a_1, a_2, a_3) \in M$ and $(a_2, a_3) \in S_{extra}$, as in many augmentors of Figure 4. We cannot form a self-loop in the dual augmentor, because we only formed self-loops in $\overline{S}$ for edges of S, not S_{extra}. Because of this, we simply have no loop at all, and we associate to A a dual augmentor of Type-1a or Type-1b (depending if this special case occurs at both (a_1, a_2, a_3) and (b_1, b_2, b_3) or just one of them). Notice that only nodes that are in M can have degree 1 in a dual augmentor.

Type-2: By similar reasoning, we obtain a Type-2 dual augmentor shown in Figure 5, with only the degree 3 nodes possibly being in M.

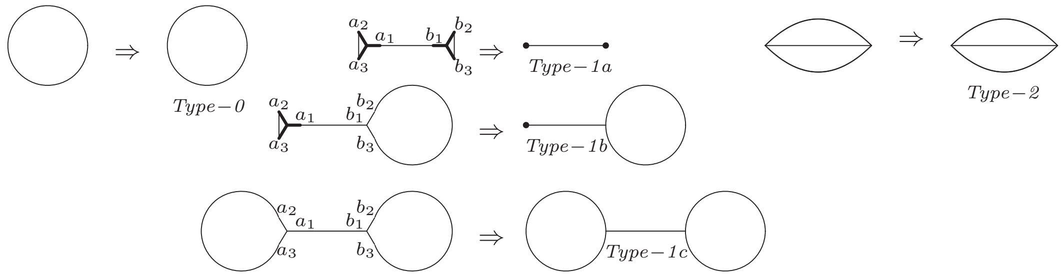

Figure 5: Transformations from augmentors to dual augmentors.

It is easy to see that the following lemma holds.

Lemma 3.1 *There is a one-to-one correspondence between augmentors in S' and dual augmentors in the cubic graph $\overline{S}$, and dual augmentors satisfy the following conditions (in fact, these conditions are an alternate definition of dual augmentors):*

1. *Degree 2 everywhere except at most 2 nodes.*
2. *All degree 1 nodes are in M.*
3. *All degree 2 nodes are not in M.*

In other words, dual augmentors are the structures in Figure 5, with the only nodes in M being the ones of degree 1 or 3. The reason for considering the dual graph $\overline{S}$ instead of S is that we now have a cubic (i.e., 3-regular) graph and as the lemma below will show, our goal now will be to cover this cubic graph with dual augmentors. While the same results can be proven directly for S' instead of $\overline{S}$, their statements become a lot more messy and complicated.

We now proceed to argue that there always exists an augmentor with high potential, for which we need the concept of *augmentor sum.* Let $\overline{\mathcal{A}}$ be the set of all possible dual augmentors contained in $\overline{S}$. We call a function $\alpha : \overline{\mathcal{A}} \to \mathbb{N}$ a *valid augmentor sum* of $\overline{S}$ iff $\exists x > 0$ such that for all edges e of $\overline{S}$, we have that $\sum_{A\in\overline{\mathcal{A}},A\ni e} \alpha(A) = x$. In other words, a valid augmentor sum is a cover of $\overline{S}$ with dual augmentors so that every edge is contained in exactly the same number of elements (which we call the *cover number*). Given that there is a one-to-one correspondence between augmentors in S' and dual augmentors in $\overline{S}$ we can also view α as a weight assignment on the augmentors in S'. Figure 4 shows a set of dual augmentors that form a valid augmentor sum by covering every edge of $\overline{S}$ twice.

It also shows the augmentors of S' they correspond to. The lemma below shows that if $\phi_M(S) > 0$, then the same must be true for at least one of the augmentors in that list. The idea behind it is that if all augmentors corresponding to the dual augmentors in α "add up" to S and S is improving, then so is some augmentor in the sum.

Lemma 3.2 *Given a perfect matching M and an M-alternating 2-factor S with $\phi_M(S) > 0$, there exists an augmentor A with $\phi_M(A) \geq \frac{\phi_M(S)}{|S|}$ if there exists a valid augmentor sum α of $\overline{S}$.*

Proof Sketch: If we had a cover of S by augmentors, such that every edge of S is contained in the same number of augmentors, then we immediately know that some augmentor must have positive potential. This follows because the total potential of the augmentors must equal to a multiple of $\phi_M(S)$. Unfortunately, we have such a cover of $\overline{S}$, not S. As shown in Figure 4, dual augmentors of $\overline{S}$ can correspond to augmentors that include edges in S_{extra}, not S. In fact, there are some 3d edges of S in Figure 4 that are not contained in any augmentors from the list, even though this list forms a valid augmentor sum of $\overline{S}$ (with cover number of 2). Notice, however, that the edges of S_{extra} corresponding to these 3d edges *are* included in the list of augmentors, which we are able to relate to the cost of the 3d edges using the Simplex Condition. □

Given a perfect matching M that is not min-cost, we know there exists an M-alternating 2-factor S with $\phi_M(S) > 0$, since every component of the symmetric difference of M with the min-cost perfect matching is such a 2-factor. By Lemma 3.2, it is enough to show that $\overline{S}$ has a valid augmentor sum to prove that our algorithm finds the min-cost perfect matching. The following theorem completes the correctness proof.

Theorem 3.3 *Any cubic multigraph $\overline{S}$ (possibly with self-loops) has a valid augmentor sum α with respect to any set of nodes M.*

4 Valid Augmentor Sums (proof of Theorem 3.3)

In this section we forget about our algorithm and Simplex Matching, and concentrate on proving Theorem 3.3. We assume that we are given an arbitrary cubic multigraph $\overline{S}$ that may contain self-loops, and some set M of nodes in $\overline{S}$. We show that there always exists a valid augmentor sum of $\overline{S}$ with respect to M. In other words, we show that we can always cover any cubic graph with objects from Figure 5 so that each edge appears in the same number of these objects.

To understand when such covers may exist, consider the special case when $M = \emptyset$. [31] states that we can cover any cubic 2-connected graph with cycles so that every edge is in the same number of cycles. Since $M = \emptyset$, all cycles are dual augmentors (they satisfy the conditions of Lemma 3.1), and so we always have a valid augmentor sum. There has been much work in finding cycle covers with small cover numbers [21, 34], and it is unknown if there always exists a cycle cover of a cubic 2-connected graph with cover number 2 (this is the Cycle Double Cover Conjecture). Since we are only proving an existence result, however, for our purposes the cover number does not need to be small.

The fact that M may not be empty complicates things. For example, while forming a cycle cover of a planar graph is easy, consider forming an augmentor sum of Figure 9, or of K_4 with $|M| = 1$. Lemma 3.1 puts degree constraints on nodes of M, which makes augmentor sums much more difficult to deal with than cycle covers. Our hope is that these results will lead to more covering results where nodes have general degree constraints.

We start our proof of Theorem 3.3 with the following easy lemma about augmentor sums.

Lemma 4.1 *Let $\mathcal{B}$ be a collection of edge subsets of $\overline{S}$, and $\beta : \mathcal{B} \rightarrow \mathbb{N}$ and x_β be such that for all edges e in $\overline{S}$, $\sum_{B \in \mathcal{B}, B \ni e} \beta(B) = x_\beta$. If every $B \in \mathcal{B}$ has a valid augmentor sum then $\overline{S}$ has a valid augmentor sum.*

We will also make use of the following cycle cover theorem, due to Paul Seymour. It gives a sufficient condition for the existence of a valid cycle sum, which we define in the same way as a valid augmentor sum.

Theorem 4.2 [31] *We are given a graph G with capacities $cap(e)$. Let $\mathcal{C}(G)$ be the collection of cycles in G and a* valid circuit sum *of G be a function $\beta : \mathcal{C}(G) \rightarrow \mathbb{Q}^+$ such that for every edge e of G, $\sum_{C \in \mathcal{C}(G), C \ni e} \beta(C) = cap(e)$. Then, a valid circuit sum exists if for every cut K, we have that $\forall e \in K$, $cap(e) \leq \sum_{e' \in K-e} cap(e')$.*

We now proceed to construct a valid augmentor sum of $\overline{S}$. We will prove Theorem 3.3 with a series of lemmas that show the existence of a valid augmentor sum of $\overline{S}$ in a different way depending on the structure of $\overline{S}$.

Lemma 4.3 *If $\overline{S}$ is 3-connected and $|M| \neq 1$, then there exists a valid augmentor sum of $\overline{S}$.*

Proof Sketch: Construct a new graph G by connecting an extra node s to all nodes of M. Set the capacities of edges (s, v) to 3, and all other capacities to 1. Since $\overline{S}$ is 3-connected, we can use Theorem 4.2 to form a cycle cover of G (see Figure 10). To form dual augmentors from these cycles, we need to break up all the cycles containing a node of M, since nodes of M cannot have degree 2 in a dual augmentor. We can do this since all such cycles contain at least two nodes of M. $\square$

Section 4.1 addresses the special (and tricky) case in the above Lemma when only a single node of $\overline{S}$ is in M. Hence, from this point on we can assume that $\overline{S}$ is not 3-connected. The following two lemmas provide us with augmentor sums for the cases when $\overline{S}$ is 1-connected or 2-connected.

Lemma 4.4 *Assume that all cubic multigraphs smaller than $\overline{S}$ have valid augmentor sums and that $\overline{S}$ contains a bridge. Then $\overline{S}$ has a valid augmentor sum.*

Proof: Let $e = (u, v)$ be a bridge in $\overline{S}$ (i.e., taking out edge e disconnects the graph). Let the other two edges incident to u be $e_1(u)$ and $e_2(u)$, and the other two edges incident to v be $e_1(v)$ and $e_2(v)$. Form two smaller cubic multigraphs S_1 and S_2 by removing e and contracting $e_2(u)$ and $e_2(v)$, as in Figure 6. During this contraction, we delete nodes u and v. Notice that in the case where $e_1(u) = e_2(u)$ (u has a self-loop), this process just forms a loop without any nodes, which is a dual augmentor of Type-0 and corresponds to a cycle with no 3d edges in S.

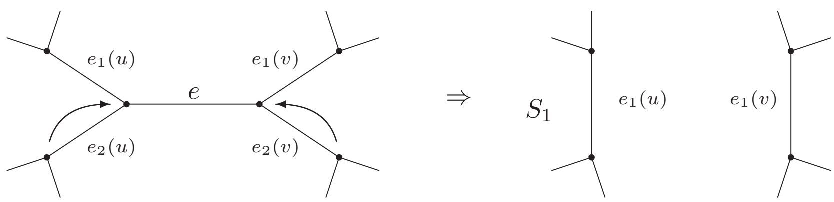

Figure 6: Breaking $\overline{S}$ with a bridge into two smaller cubic multigraphs

By the assumption made in the statement of the lemma, there exist valid augmentor sums α_1 and α_2 of S_1 and S_2, with corresponding cover values x_1 and x_2. If x is the least common multiple of x_1 and x_2, then $\frac{x}{x_1}\alpha_1$ and $\frac{x}{x_2}\alpha_1$ are valid augmentor sums of S_1 and S_2 with cover value x. This gives us a multiset of size x of dual augmentors in S_1 that contain $e_1(u)$ and another multiset of size x of dual augmentors in S_2 that contain $e_1(v)$. Pair up the dual augmentors of the S_1 multiset with the dual augmentors of the S_2 multiset and let (A_1, A_2) be one such pair. Consider the multigraph C resulting from adding e to A_1 and A_2, once again forming nodes u and v. If C has a valid augmentor sum regardless of what the types of A_1 and A_2 are, then using Lemma 4.1 we can deduce that the entire graph $\overline{S}$ has a valid augmentor sum.

Rather than providing a valid augmentor sum individually for all possible graphs C resulting from the pairing of the different types of dual augmentors, we give three simple rules that reduce most of such possible graphs to trivial cases. Each of these rules decomposes C into smaller graphs, such that if all these smaller graphs have valid augmentor sums, then by Lemma 4.1 C also has a valid augmentor sum. Figure 7 illustrates these rules (for a detailed description see the Appendix). We would apply Rule 1 if $v \in M$, Rule 2 if $v \notin M$, and one of the two variants of Rule 3 depending if u and v are in M.

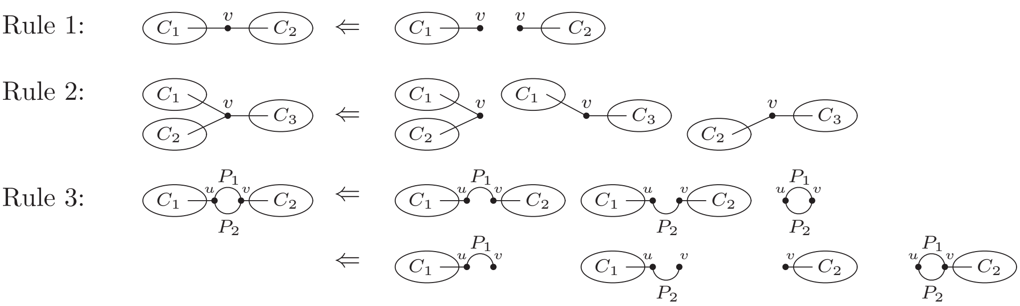

Figure 7: Rules for decomposing graphs into easier cases to prove the existence of augmentor sums

It is easy to check that no matter what two structures A_1 and A_2 from Figure 5 are stitched together by a bridge to form $C = A_1 \cup A_2 \cup e$, we can always decompose them into dual augmentors by repeated application of the above three rules. The only exception to this occurs in the example below.

Let A_1 be a dual augmentor of Type-1c, A_2 be a dual augmentor of Type-2 and $u \notin M$ lies on one of the cycles of A_1. Moreover, suppose that the node of degree 3 on the same cycle of A_1 is in M. Then we would apply the second version of Rule 3, as shown in Figure 8.

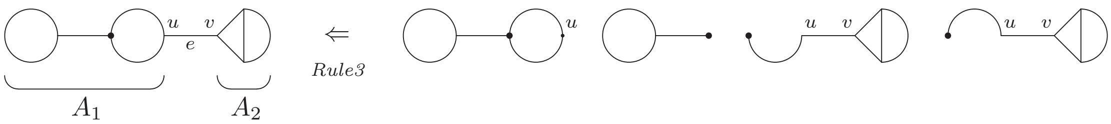

Figure 8: Sample reduction for Lemma 4.4

Graphs like the last one in Figure 8 and ones of similar structure are the only ones that can be formed by this process of repeatedly decomposing C, such that they are neither dual augmentors nor have a further decomposition that is possible using one of the three rules above. For these types of graphs we show a valid augmentor sum directly in Figure 9. This completes the proof of the lemma. □

The only case left now is if $\overline{S}$ is bridgeless, but not 3-connected, which is addressed below.

Lemma 4.5 *Assume that all cubic multigraphs smaller than $\overline{S}$ have a valid augmentor sum and that $\overline{S}$ contains two edges e_1 and e_2, the removal of which disconnects it. Then $\overline{S}$ has a valid augmentor sum.*

The proof of the above lemma is similar to Lemma 4.4: we decompose $\overline{S}$ into two smaller cubic graphs S_1 and S_2, and form augmentor sums of each. Then we show that after patching together a dual augmentor of S_1 with a dual augmentor of S_2, we get an object that itself has a valid augmentor sum. Note that the "patching" process here is different from Lemma 4.4. The full proof appears in the Appendix. Using the above lemmas, we are now able to prove that our algorithm works correctly.

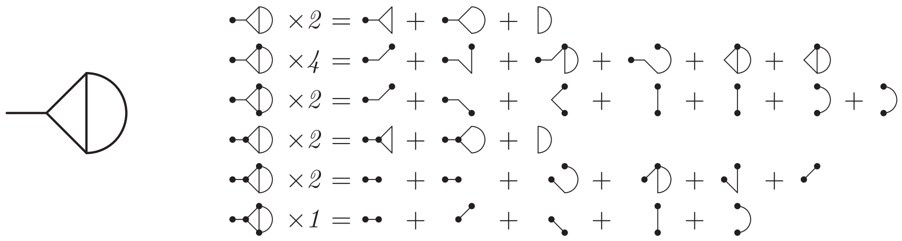

Figure 9: Proof that the last remaining case of Lemma 4.4 has a valid augmentor sum. Small circles indicate nodes of M, so this shows all possible cases.

Proof of Theorem 3.3. We prove this by induction on the number of nodes in $\overline{S}$. The smallest cubic multigraph consists of two nodes u, v and the entire graph is a dual augmentor of Type-2.

Now assume that all cubic multigraphs smaller than $\overline{S}$ have a valid augmentor sum. If $\overline{S}$ is not connected consider each component separately and use the inductive hypothesis. By Lemma 4.4, we have proved the inductive step for S containing a bridge, so we can assume S is 2-connected. If $\overline{S}$ is 3-connected, we can use Lemma 4.3, Lemma 4.6, or Lemma 4.7. Otherwise, we can use Lemma 4.5 together with the inductive hypothesis. This finishes the proof. □

4.1 Special Case: $|M| = 1$

This section is devoted to the special case where $|M| = 1$ and $\overline{S}$ is 3-connected. Lemma 4.6 states that if $\overline{S}$ contains three edges whose removal disconnects $\overline{S}$, then it has a valid augmentor sum due to arguments similar to Lemmas 4.4 and 4.5. See the Appendix for the proof.

Lemma 4.6 *Assume that all cubic multigraphs smaller than $\overline{S}$ have valid augmentor sums, and that there exists a cut of size 3 with more than a single node on each side. If $\overline{S}$ is 3-connected, and $|M| = 1$, then there there exists a valid augmentor sum of $\overline{S}$.*

The most difficult subcase of Theorem 3.3 is proven in the following main lemma of this section, which requires different techniques from most of our other proofs. We prove the cases when $\overline{S}$ is planar and non-planar separately. For the non-planar case, we show that there must be some subdivision of $K_{3,3}$ containing the node of M, and from this we form a valid augmentor sum using Theorem 4.2. In the planar case, we use powerful edge-coloring results (see Figure 15).

Lemma 4.7 *Assume that $\overline{S}$ does not have a cut of size 3 with more than a single node on each side. If $\overline{S}$ is 3-connected, and $|M| = 1$, then there exists a valid augmentor sum of $\overline{S}$.*

5 Running Time

Here we argue that our algorithm runs in polynomial time. We can find the initial perfect matching using the unweighted version of the Simplex Matching algorithm from [20]. If we are applying this to Terminal Backup or similar problems, then there always exists a perfect matching without 3d edges, so we can find it using traditional matching algorithms.

Theorem 5.1 *Our algorithm solves weighted* Simplex Matching *in polynomial time.*

Proof Sketch: Lemma 3.2, Theorem 3.3, and Lemma 2.1 tell us that we reduce the cost of our perfect matching by a large enough factor in every step, so that we only need a polynomial number of augmentations. For various optimizations of our algorithm, see the full proof in the Appendix. □

Acknowledgements

The authors would like to thank Moses Charikar, Sanjeev Khanna, and Avi Wigderson for interesting and productive discussions. We especially thank Paul Seymour for his great contribution to this paper, and Dahai Xu and Mung Chiang for formulating the *Terminal Backup* problem.

References

[1] Z. Abrams, A. Meyerson, K. Munagala, S. Plotkin. "On the Integrality Gap of Capacitated Facility Location."

[2] K. Appel, W. Haken. "Every planar map is four-colorable." In *Contemporary Mathematics,* 98 (1989).

[3] E. Arkin, R. Hassin, S. Rubinstein, M. Sviridenko. "Approximations for Maximum Transportation Problem with Permutable Supply Vector and Others Capacitated Star Packing Problems", *Algorithmica* v.39 (2004), pp.175–187.

[4] M. Bläser, B. Manthey. "Two Approximation Algorithms for 3-Cycle Covers." *Proceedings of the 5th International Workshop on Approximation Algorithms for Combinatorial Optimization*, p.40-50, 2002.

[5] M. Bläser, L. Ram, M. Sviridenko. "Improved approximation algorithms for metric maximum ATSP and maximum 3-cycle cover problems." *WADS 2005.*

[6] I. Cahit. "Spiral Chains: The Proofs of Tait's and Tutte's Three-Edge-Coloring Conjectures." arXiv preprint, math CO/0507127 v1, July 6, 2005.

[7] G. Calinescu, A. Zelikovsky. "The Polymatroid Steiner Problems." *J. Combinatorial Optimization* 9(3): 281-294 (2005).

[8] J. Chen, S. Lu, S. Sze, F. Zhang. "Improved algorithms for path, matching, and packing problems." To appear in *Proc. 18th ACM-SIAM Symposium on Discrete Algorithms (SODA 2007).*

[9] G. Cornuéjols. *Combinatorial Optimization: Packing and Covering.* CBMS-NSF Regional Conference Series in Applied Mathematics 74, SIAM 2001.

[10] G. Cornuéjols, D. Hartvigsen. "An extenstion of matching theory." *J. Combinatorial Theory, Series B* 40, (1986) pp. 285–296.

[11] G. Cornuéjols, D. Hartvigsen, W. Pulleyblank. "Packing subgraphs in a graph." *OR Letters* 1 (1982) 139–143.

[12] W. Cunningham. "Matching, matroids, and extensions." *Mathematical Programming* B91, (2002), pp. 515–542.

[13] U. Derigs. "A shortest augmenting path method for solving minimal perfect matching problems," *Networks* 11 (1981), pp. 379-390.

[14] G. Even, G. Kortsarz, W. Slany. "On network design problems: fixed cost flows and the covering steiner problem." *ACM Transactions on Algorithms (TALG)*, 1(1) pp: 74–101 (2005).

[15] M. Goemans, D. Williamson."A General Approximation Technique for Constrained Forest Problems." *SIAM Journal on Computing,* 24, pp. 296–317, 1995.

[16] S. Guha, A. Meyerson, K. Munagala. "Hierarchical placement and network design problems." *Proceedings of FOCS 2000.*

[17] A. Gupta, A. Srinivasan. “On the Covering Steiner Problem.” *Theory of Computing* Vol. 2, pp. 53–64.

[18] D. Hartvigsen, P. Hell, J. Szabó. “The k-piece packing problem”. *J. Graph Theory* 52 (2006) 267–293.

[19] P. Hell. “Packings in graphs.” *Electronic Notes in Discrete Mathematics,* volume 5 (2000).

[20] P. Hell, D. Kirkpatrick. “Packings by cliques and by finite families of graphs.” *Discrete Mathematics* 49 (1984), pp. 45–49.

[21] U. Jamshy, M. Tarsi.“Short cycle covers and the cycle double cover conjecture.” *J. Combinatorial Theory Series B.* 56 (1992), pp. 197-204.

[22] H. Kaplan, M. Lewenstein, N. Shafrir, M. Sviridenko. “Approximation algorithms for Asymmetric TSP by Decomposing Directed Regular Multigraphs.” *Journal of ACM* 52 (2005), pp. 602-626.

[23] D. Karger, M. Minkoff. “Building steiner trees with incomplete global knowledge.” *Proceedings of FOCS 2000.*

[24] A. Kelmans. “Optimal packing of induced stars in a graph.” In *Discrete Math.* 173 (1997) pp. 97–127.

[25] M. Loebl, S. Poljak. “Efficient Subgraph Packing.” *Journal of Combinatorial Theory, Series B* 59, pp. 106–121 (1993).

[26] L. Lovasz, M. Plummer. *Matching Theory.* Elsevier Science Ltd (1986).

[27] G. Pap. “Hypo-matchings in directed graphs.” *Graph Theory 2004*, Birkhuser Verlag, Basel, Switzerland (2006), pp. 325-335.

[28] G. Pap. “A TDI description of restricted 2-matching polytopes.” *IPCO 2004*, pp. 139–151.

[29] N. Robertson, P.D. Seymour. “Graph Minors. IX. Disjoint Crossed Paths.” In *Journal of Combinatorial Theory, Series B,* 49, 40-77 (1990).

[30] N. Robertson, D.P. Sanders, P.D. Seymour, R. Thomas. “The four-colour theorem.” In *Journal of Combinatorial Theory, Series B,* 70 (1997), 2-44.

[31] P.D. Seymour. “Sums of circuits.” In *Graph Theory and Related Topics*, J.A. Bondy and U.R.S. Murty Eds, Academic Press (1979) 341-355.

[32] P.G. Tait. “Note on a theorem in geometry of position.” In *Transactions of the Royal Society of Edinburgh,* 29 (1880), 657-660.

[33] D. Xu, E. Anshelevich, M. Chiang. “Simplex Cover: Modeling, Algorithms, and Networking Applications.” Submitted to International Conference on Distributed Computing Systems (ICDCS 2007), `http://www.princeton.edu/~dahaixu/pub/simplex/simplex.pdf`.

[34] C.Q. Zhang. *Integer flows and cycle covers of graphs.* Marcel Dekker, 1997.

Appendix

Lemma 2.1 *Let A be an augmentor with maximal potential. We can find an M-alternating 2-factor S with $\phi_M(S) \geq \phi_M(A)$ in polynomial time (which may or may not be A itself).*

Proof: Suppose A is of Type-0. Delete all 3d edges from H as well as all nodes that are matched in M using 3d edges, forming a graph H' with no 3d edges. Find a min-cost matching M^* of H'. $S = M^* \oplus M \mid_{H'}$ gives us an M-alternating 2-factor to augment M by. Since A is of Type-0 it cannot include any nodes incident to 3d edges of M, thus A is contained in H'. Augmenting by S results in the best possible matching in H', therefore $\phi_M(S) \geq \phi_M(A)$.

Now suppose A is of Type-1 or Type-2. Fix the two 3d edges e_1 and e_2 that it contains (there are only $|E|^2$ possibilities). We will find the best M-alternating 2-factor with only e_1 and e_2 as the 3d edges. Form a new graph H' as above, except leave the nodes incident to $e_1 = (u_1, u_2, u_3)$ and $e_2 = (v_1, v_2, v_3)$ in H'. If $e_1 \notin M$, form three new nodes u'_1, u'_2, u'_3 with edges (u_i, u'_i) and do likewise for e_2. There are only six unmatched nodes in H': either u_1, u_2, u_3 if $e_1 \in M$, or u'_1, u'_2, u'_3 if $e_1 \notin M$, and similarly for e_2. Now find a min-cost matching M^* of H'. $M^* \oplus M \mid_{H'}$ is an M-alternating 2-factor except for the nodes that were unmatched in H' using M, which have degree 1. Add e_1 and e_2 to $M^* \oplus M \mid_{H'}$ (if $e_1 \notin M$, then adding it replaces the edges (u_i, u'_i)), forming an M-alternating 2-factor S. As before, A is a possible way to augment M in H' and since S is the best such way we have $\phi_M(S) \geq \phi_M(A)$.

We now choose the best one of the resulting $|E|^2 + 1$ M-alternating 2-factors.

There are many ways to make the above algorithm run faster. First, notice that we do not need to consider pairs of 3d edges (e_1, e_2) if $e_1 \notin M$ is adjacent to a 3d edge of M that is not e_2. For other improvements, see Section 5, and [33]. □

Lemma 3.2 *Given a perfect matching M and an M-alternating 2-factor S with $\phi_M(S) > 0$, there exists an augmentor A with $\phi_M(A) \geq \frac{\phi_M(S)}{|S|}$ if there exists a valid augmentor sum α of $\overline{S}$.*

Proof: If we had a cover of S by augmentors, such that every edge of S is contained in the same number of augmentors, then we immediately know that some augmentor must have positive potential. This is simply because the total potential of the augmentors must equal a multiple of $\phi_M(S)$. Unfortunately, we have such a cover of $\overline{S}$, not S. As shown in Figure 4, dual augmentors of $\overline{S}$ can correspond to augmentors that include edges in S_{extra}, not S. In fact, there are some 3d edges of S in Figure 4 that are not contained in any augmentors from the list, even though this list forms a valid augmentor sum of $\overline{S}$ (with cover number of 2). Notice, however, that the edges of S_{extra} corresponding to these 3d edges *are* included in the list of augmentors, which we are able to relate to the cost of the 3d edges using the Simplex Condition.

Let x be the cover number of $\alpha : \overline{\mathcal{A}} \to \mathbb{N}$ and let $\mathcal{A}$ be the set of augmentors in S'. Since there is a one-to-one correspondence between $\overline{\mathcal{A}}$ and $\mathcal{A}$ we will consider α as an integer weight assignment to augmentors in S'. First we will compute how many times α covers an edge in $S \subseteq S'$.

1. If $e = (u, v) \in S$ then $\sum_{A \in \mathcal{A}, A \ni e} \alpha(A) = x$.
   Proof: A 2d edge in S' corresponds to some edge in $\overline{S}$, which is covered exactly x times.

2. If $e = (u, v, w) \in S$ such that some $(u, v) \in S$ then $\sum_{A \in \mathcal{A}, A \ni e} \alpha(A) = x$.
   Proof: This implies that the node v_e corresponding to e in $\overline{S}$ has a self-loop. By construction, all dual augmentors containing v_e must also contain the self-loop, which is covered x times, so e is covered exactly x times.

3. If $e = (u, v, w) \in M \cap S$ such that $e_1 = (u, v), e_2 = (v, w), e_3 = (u, w) \notin S$ then for all $i = 1, 2, 3$, $\sum_{A \in \mathcal{A}, A \ni e} \alpha(A) - 2 \sum_{A \in \mathcal{A}, A \ni e_i} \alpha(A) = x$.
   Proof: Consider the middle rightmost 3d edge of Figure 4. It appears in several augmentors,

with different corresponding edges from S_{extra}. What the above statement implies is that each of these edges in S_{extra} are covered the same number of times.
Let v_e be the node in $\overline{S}$ corresponding to e. Each edge incident to v_e appears in dual augmentors exactly x times and each dual augmentor either contains all three edges or exactly one of them, since the degree of a node in M is 1 or 3. Let a be the number of dual augmentors in α containing all 3 edges incident to v_e. Then all remaining dual augmentors that contain v_e must be organized as a set of size $x-a$ of triples such that each dual augmentor in the triple contains a different edge incident to v_e. Every dual augmentor like this corresponds to an augmentor of S' that contains the edge e and also contains exactly one of e_1, e_2 or e_3 in S_{extra}. The a augmentors and the $x-a$ triples of augmentors all contain e in S', and so edge e is contained in $a+3(x-a)$ augmentors, and each e_i is contained in $x-a$ augmentors, producing the desired result.

4. If $e=(u,v,w) \in S-M$ such that $e_1=(u,v), e_2=(v,w), e_3=(u,w) \notin S$ for all $i=1,2,3$, $\sum_{A\in\mathcal{A},A\ni e}\alpha(A)+2\sum_{A\in\mathcal{A},A\ni e_i}\alpha(A)=x$.
Proof: Let v_e be the node in $\overline{S}$ corresponding to e. Each edge incident to v_e appears in dual augmentors exactly x times and each dual augmentor either contains all three edges or exactly two of them, since the degree of v_e in a dual augmentor can only be 2 or 3. Let a be the number of dual augmentors in α containing all 3 edges incident to v_e. These correspond exactly to the augmentors containing e in S'. The rest of the dual augmentors that cover edges adjacent to v_e must be organized as a set of size $(x-a)/2$ of triples such that each dual augmentor in the triple contains two different edges incident to v_e, so that the entire triple covers each edge twice. A corresponding triple of augmentors in S' contains each of e_1, e_2 and e_3 once. Therefore, edge e is contained in a augmentors, and each edge e_i is contained in $(x-a)/2$, producing the result.

Using these covering results we now bound $x\ \phi_M(S)=x\sum_{e\in S}\phi_M(e)$.
If $e=(u,v)$, or $e=(u,v,w)\in S$ with some $(u,v)\in S$, we have $x\ \phi_M(e)=\sum_{A\in\mathcal{A},A\ni e}\alpha(A)\phi_M(e)$.
If $e=(u,v,w)\in M\cap S$ such that $e_1=(u,v), e_2=(v,w), e_3=(u,w)\notin S$, then $\phi_M(e)=c(e)$ and $\phi_M(e_i)=-c(e_i)$, so

$$x\ \phi_M(e)=\sum_{A\in\mathcal{A},A\ni e}\alpha(A)\phi_M(e)-2\sum_{A\in\mathcal{A},A\ni e_i}\alpha(A)\phi_M(e)\leq \tag{1}$$

$$\sum_{A\in\mathcal{A},A\ni e}\alpha(A)\phi_M(e)-\sum_{A\in\mathcal{A},A\ni e_i}\alpha(A)(c(e_1)+c(e_2)+c(e_3))= \tag{2}$$

$$\sum_{A\in\mathcal{A},A\ni e}\alpha(A)\phi_M(e)+\sum_{A\in\mathcal{A},A\ni e_i}\alpha(A)(\phi_M(e_1)+\phi_M(e_2)+\phi_M(e_3))= \tag{3}$$

$$\sum_{A\in\mathcal{A},A\ni e}\alpha(A)\phi_M(e)+\sum_{A\in\mathcal{A},A\ni e_1}\alpha(A)\phi_M(e_1)+\sum_{A\in\mathcal{A},A\ni e_2}\alpha(A)\phi_M(e_2)+\sum_{A\in\mathcal{A},A\ni e_3}\alpha(A)\phi_M(e_3) \tag{4}$$

The inequality holds because of the Simplex Condition on e and the last part because $\sum_{A\in\mathcal{A},A\ni e_i}\alpha(A)$ is the same for all $i=1,2,3$.

Similarly, if $e=(u,v,w)\in S-M$ such that $e_1=(u,v), e_2=(v,w), e_3=(u,w)\notin S$, then $\phi_M(e)=-c(e)$ and $\phi_M(e_i)=-c(e_i)$, so

$$x\ \phi_M(e)=\sum_{A\in\mathcal{A},A\ni e}\alpha(A)\phi_M(e)+2\sum_{A\in\mathcal{A},A\ni e_i}\alpha(A)\phi_M(e)\leq \tag{5}$$

$$\sum_{A\in\mathcal{A},A\ni e}\alpha(A)\phi_M(e)-\sum_{A\in\mathcal{A},A\ni e_i}\alpha(A)(c(e_1)+c(e_2)+c(e_3))= \tag{6}$$

$$\sum_{A\in\mathcal{A},A\ni e}\alpha(A)\phi_M(e)+\sum_{A\in\mathcal{A},A\ni e_1}\alpha(A)\phi_M(e_1)+\sum_{A\in\mathcal{A},A\ni e_2}\alpha(A)\phi_M(e_2)+\sum_{A\in\mathcal{A},A\ni e_3}\alpha(A)\phi_M(e_3) \tag{7}$$

Therefore we have that $x\ \phi_M(S)\leq\sum_{A\in\mathcal{A}}\sum_{e\in A}\alpha(A)\phi_M(e)=\sum_{A\in\mathcal{A}}\alpha(A)\phi_M(A)$. If for all $A\in\mathcal{A}$, $\phi_M(A)<\frac{\phi_M(S)}{|S|}$, then by the above inequality $x\ \phi_M(S)<\frac{\phi_M(S)}{|S|}\sum_{A\in\mathcal{A}}\alpha(A)$. Now consider the dual augmentors corresponding to these augmentors A. Every edge in $\overline{S}$ is covered exactly x times, hence $\sum_{A\in\mathcal{A}} \alpha(A) \leq x|E(\overline{S})|$. By definition of $\overline{S}$ we know that $|E(\overline{S})| \leq |S|$. Overall, this implies that $x\,\phi_M(S) < x\,\phi_M(S)$, giving us a contradiction, as desired. □

Lemma 4.1 *Let $\mathcal{B}$ be a collection of edge subsets of $\overline{S}$, and $\beta : \mathcal{B} \to \mathbb{N}$ and x_β be such that for all edges e in $\overline{S}$, $\sum_{B\in\mathcal{B},B\ni e} \beta(B) = x_\beta$. If every $B \in \mathcal{B}$ has a valid augmentor sum then $\overline{S}$ has a valid augmentor sum.*

Proof: Let α_B be the augmentor sum for $B \in \mathcal{B}$, and let x_B be the cover number of α_B. Let x be the least common multiple of all x_B's and $\alpha'_B = \frac{x}{x_B}\alpha_B$, so α'_B is a valid augmentor sum of B with cover number x. α'_B is defined only for dual augmentors in B but we can extend it to the set of all dual augmentors $\overline{\mathcal{A}}$ in $\overline{S}$ by setting $\alpha'_B(A) = 0$ for any dual augmentor A of $\overline{S}$ not contained in B.

Define $\alpha(A) = \sum_{B\in\mathcal{B}} \beta(B)\alpha'_B(A)$. Then for all edges e in $\overline{S}$

$$\sum_{A\in\mathcal{A},A\ni e} \alpha(A) = \sum_{A\in\mathcal{A},A\ni e}\sum_{B\in\mathcal{B}} \beta(B)\alpha'_B(A) = \tag{8}$$

$$= \sum_{B\in\mathcal{B}} \beta(B) \sum_{A\in\mathcal{A},A\ni e} \alpha'_B(A) = \tag{9}$$

$$= \sum_{B\in\mathcal{B},B\ni e} \beta(B)\, x = x_\beta x \tag{10}$$

Since every edge of $\overline{S}$ appears in exactly $x_\beta x$ dual augmentors, α is a valid augmentor sum of $\overline{S}$. □

Lemma 4.3 *If $\overline{S}$ is 3-connected and $|M| \neq 1$, then there exists a valid augmentor sum of $\overline{S}$.*

Proof: Construct a new graph G by adding an extra node s to $\overline{S}$, together with an edge (s, v) for all $v \in M$ (see Figure 10). Associate a capacity $cap(e)$ to all edges e of G. If e is one of the new edges (s, v), set $cap(e) = 3$, otherwise set $cap(e) = 1$. By Theorem 4.2, we know that there exists a valid circuit sum of G iff for all cuts K of G and all edges $e \in K$, $cap(e) \leq \sum_{e'\in K-e} cap(e')$. We now show that this holds true for G.

Any cut K in G contains at least three edges since G is 3-connected. If $e \in K$ is such that $cap(e) = 1$ then the above inequality is trivially satisfied (this also finishes the case when $|M| = 0$). Otherwise, if $e = (s, v)$ and K is the cut $(s, G - s)$ then there is another edge of capacity 3 in K since $|M| \geq 2$. Finally, if $e = (s, v)$ and K is not $(s, G - s)$, then $K - e$ is also a cut in $\overline{S}$ and it contains at least three edges of capacity 1 since $\overline{S}$ is 3-connected.

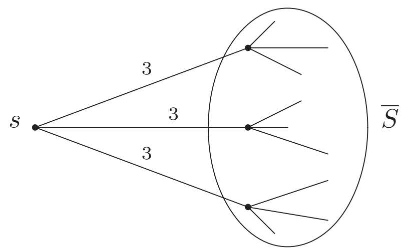

Figure 10: Capacitated Graph G in Lemma 4.3

We will partition each cycle C with $\beta(C) > 0$ into dual augmentors as follows. Every cycle that does not contain any nodes in M is contained in $\overline{S}$ and is trivially a Type-0 dual augmentor. No cycle passes through two edges with capacity 1 incident to a node $v \in M$, since all other cycles passing through v would not be able to fill (s, v) to its capacity. Therefore, every cycle entering a node in M must proceed to s. Removing all (s, v) edges from these cycles gives us a collection of Type-1a dual augmentors in $\overline{S}$. The valid circuit sum β can be viewed as a fractional weight assignment on those

Type-1a and Type-0 dual augmentors such that every edge in $\overline{S}$ is covered exactly once. We can now multiply β by a large enough constant to form a valid augmentor sum. □

Lemma 4.4 *Assume that all cubic multigraphs smaller than $\overline{S}$ have valid augmentor sums and that $\overline{S}$ contains a bridge. Then $\overline{S}$ has a valid augmentor sum.*

Proof: The proof can be found in the main body of the paper. Here we provide detailed description of the three decomposition rules shown in Figure 7.

Rule 1: Suppose C contains a vertex v of degree 2, the removal of which disconnects C into components C_1 and C_2. If $C - C_1$ and $C - C_2$ have valid augmentor sums, then by Lemma 4.1 C also has a valid augmentor sum, as the edge sets of $C - C_1$ and $C - C_2$ cover every edge of C exactly once. We can also apply a similar rule if v is of degree 3. This rule is made for the case where $v \in M$, since v has degree 1 in resulting graphs.
Rule 2: Suppose C contains a vertex v of degree 3, the removal of which disconnects C into components C_1, C_2 and C_3. As above, if $C - C_1$, $C - C_2$ and $C - C_3$ have valid augmentor sums then so does C, as these sets cover every edge of C twice. This rule should be used if $v \notin M$ since v has degree 2 in resulting graphs.
Rule 3: Suppose C contains two vertices u and v of degree 3, the removal of which partitions C into components C_1, C_2, P_1 and P_2 as in Figure 7, where P_1 and P_2 are paths. As above, if $C - P_1$, $C - P_2$ and $C - C_1 - C_2$ have valid augmentor sums then so does C, since those graphs cover every edge of C twice. Similarly, if $C - P_1 - C_2$, $C - P_2 - C_2$, $C - C_1 - P_1 - P_2$ and $C - C_1$ have valid augmentor sums then so does C.

We would apply the first variant of Rule 3 when either $u, v \in M$ or $u, v \notin M$, since it is then that the cycle $C - C_1 - C_2$ has a valid augmentor sum (if $u, v \in M$ then it is a sum of 2 dual augmentors). We would apply the second variant when $u \notin M$ and $v \in M$, since then v always has degree 1 or 3, as needed by Lemma 3.1. □

Lemma 4.5 *Assume that all 2-connected cubic multigraphs smaller than $\overline{S}$ have a valid augmentor sum and that $\overline{S}$ contains two edges e_1 and e_2, the removal of which disconnects it. Then $\overline{S}$ has a valid augmentor sum.*

Proof: We can assume that $\overline{S}$ is bridgeless. Let $e_1 = (u_1, v_1)$ and $e_2 = (u_2, v_2)$, so that u_1, u_2 are in the same component of $\overline{S} - e_1 - e_2$. We proceed as in the proof of Lemma 4.4 to form two smaller cubic multigraphs S_1 and S_2 by removing e_1 and e_2, and forming two new edges (u_1, u_2) and (v_1, v_2), as in Figure 11. Notice that the four nodes u_1, u_2, v_1 and v_2 must be distinct, since if $u_1 = u_2$, then the third edge incident to u_1 would be a bridge.

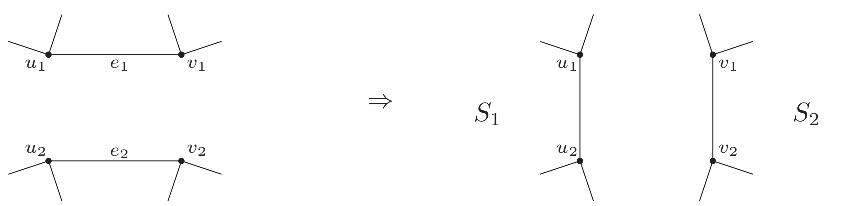

Figure 11: Breaking a 2-connected bridgeless $\overline{S}$ into two smaller cubic multigraphs

Similarly to Lemma 4.4, there exist valid augmentor sums for S_1 and S_2 with cover value x. This means that there exists a multiset of size x of dual augmentors in S_1 that contain (u_1, u_2) and another multiset of size x of dual augmentors in S_2 that contain (v_1, v_2). Pair up the dual augmentors of the S_1 multiset with the dual augmentors in the S_2 multiset and let (A_1, A_2) be one such pair. Consider the cubic multigraph C resulting from removing $(u_1, u_2), (v_1, v_2)$ and adding e_1 and e_2 to A_1 and A_2.

We will now show that C has a valid augmentor sum, regardless of the types of A_1 and A_2, which finishes the proof using Lemma 4.1.

Figure 12 shows what each dual augmentor might look like once the edge (u_1, u_2) (similarly (v_1, v_2)) is removed. The graph C is simply the joining of two of these objects along the two "connector" vertices (marked with empty nodes).

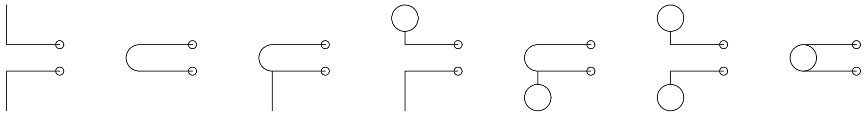

Figure 12: Dual augmentors with an edge removed. From left to right these are: Type-1a, Type-0, two versions of Type-1b (depending which edge is removed), two versions of Type-1c, and Type-2.

It is easy to see that all possible ways to join these objects to form C have already been proven to have a valid augmentor sum in Lemma 4.4, with the exception of the graph formed when the two dual augmentors A_1 and A_2 are of Type-2. Figure 13 shows the valid augmentor sum for this graph, which completes the proof of the lemma. □

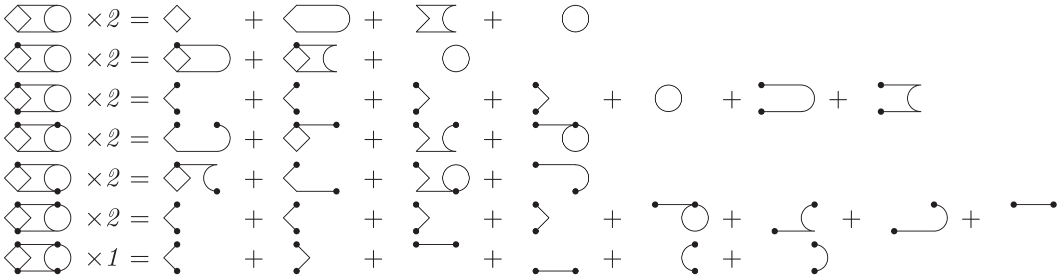

Figure 13: Proof that the last remaining case of Lemma 4.5 has a valid augmentor sum. Small circles indicate nodes of M, so this shows all possible cases.

Lemma 4.6 *Assume that all cubic multigraphs smaller than $\overline{S}$ have valid augmentor sums, and that there exists a cut of size 3 with more than a single node on each side. If $\overline{S}$ is 3-connected, and $|M| = 1$, then there there exists a valid augmentor sum of $\overline{S}$.*

Proof: This proof is similar to the proofs of Lemmas 4.4 and 4.5. Let the cut of size 3 be $\{e_1, e_2, e_3\}$, and let $M = \{r\}$. Form two new 3-connected cubic multigraphs S_1, S_2 by contracting each side of the cut into a single node, with $r \in S_1$. S_1 is a smaller set with a single node of M, so by our assumption, there exists a valid augmentor sum α_1 of S_1. Let v be the node of S_2 representing the contracted side of the cut. Here we have a choice: do we say that v is in M or not? Call the first set S_2, and the second set S_2'. In the first case, S_2 is a set with a single node of M, which must have some valid augmentor sum α_2 with cover number y. In the second case, S_2' would have no nodes in M, so by Lemma 4.3, we also have a valid augmentor sum α_2', with cover number y'.

We can say something more specific about α_2 and α_2'. Since S_2' is a 3-connected set with no nodes in M, by the proof of Lemma 4.3 we can assume that α_2' is a cycle cover (i.e., the only dual augmentors appearing in it with positive weight are cycles). In particular, all dual augmentors containing v in α_2' contain exactly two edges adjacent to v. On the other hand, S_2 is a 3-connected set with a single node v in M. By the proof of Lemma 4.7, we can inductively show that all dual augmentors containing v in α_2 contain all three edges adjacent to it.

Now let w be the node of S_1 representing the contracted side of the cut. Let a be the number of times w appears in a dual augmentor of α_1 containing all 3 edges of w, and b be the number of times w appears in such a dual augmentor of α_1 containing two edges of w. Since $w \notin M$, these are the only options, and so the cover number of α_1 is $a + 2b/3$.

The idea is that we are going to attach together α_2 with the a dual augmentors above, and α'_2 with the other b dual augmentors. To do this, let x be the least common multiple of y and $3y'/2$, and form new augmentor sums $x\alpha_1$, $\frac{xa}{y}\alpha_2$, and $\frac{2xb}{3y'}\alpha'_2$. This means that we now have xb dual augmentors containing exactly two edges of w in S_1, and xb dual augmentors containing exactly two edges of v in S'_2, the latter coming from $\frac{2xb}{3y'}\alpha'_2$. Just as in Lemmas 4.4 and 4.5, we can pair up these dual augmentors into pairs (A_1, A_2). Furthermore, since dual augmentors covering exactly two edges of a node must appear in triples (so that all edges are covered the same number of times), we can make sure that in every pair (A_1, A_2), both A_1 and A_2 use the same two edges from the set $\{e_1, e_2, e_3\}$. We can then form a multigraph C in $\overline{S}$ by patching A_1 and A_2 together, i.e., C consists of edges in $\overline{S}$ corresponding to either A_1 or A_2. Since all dual augmentors in α'_2 are cycles, C must be a dual augmentor, since patching a dual augmentor in this manner together with a cycle results in a dual augmentor again.

We now consider the xa dual augmentors containing exactly three edges of w in S_1, and xa dual augmentors containing exactly three edges of v in S_2, the latter coming from $\frac{xa}{y}\alpha_2$. We pair them up in the same way, and patch them together to form subgraphs $C = A_1 \cup A_2$ for each pair (A_1, A_2). All structures C that can be formed in this way are dual augmentors.

By taking the above subgraphs C, together with the dual augmentors of α_1, α_2, and α'_2 that do not intersect v or w, we form a valid augmentor sum for $\overline{S}$. □

Lemma 4.7 *Assume that $\overline{S}$ does not have a cut of size 3 with more than a single node on each side. If $\overline{S}$ is 3-connected, and $|M| = 1$, then there exists a valid augmentor sum o $\overline{S}$.*

Proof: Let $M = \{r\}$, the three nodes adjacent to r be v_1, v_2, v_3, and denote the edge (r, v_i) by e_i. First, we address the non-planar case, the proof of which is due largely to Paul Seymour.

Non-planar. Assume that $\overline{S}$ is not planar. We want to show that there exists a subdivision of $K_{3,3}$ in $\overline{S}$ with r as one of the degree-3 nodes in this $K_{3,3}$. From Theorem 2.4 in [29], it is easy to derive that this must hold. Specifically, delete r and its adjacent edges from $\overline{S}$ to form a new graph S'. Using the notation of [29], let $\{v_1, v_2, v_3\}$ be the set of "special" nodes $\overline{\Omega}$, and let the permutation Ω be (v_1, v_2, v_3). The society (S', Ω) is 3-connected, since if there were a separation of size 2, then there would be an edge cut (since S' is at most cubic, and v_1, v_2, v_3 have degree 2 in S') of size 2 separating r from some node in $\overline{S}$, which contradicts $\overline{S}$ being 3-connected. We also need to show that S' is not "rural", which means that it can be drawn in the plane without edge intersections, and so that the nodes of Ω are on the outside face. S' is not rural, since if it were, we could attach r again and make $\overline{S}$ become planar. S' cannot have a cross because $|\overline{\Omega}| = 3$. Therefore, by Theorem 2.4 in [29], S' must have a tripod. A tripod combined with r and e_1, e_2, e_3 gives us exactly a subdivision of $K_{3,3}$.

Let K be this $K_{3,3}$ instance, and let $\sigma(K)$ be the subdivision of it found above. By "a subdivision" we mean that $\sigma(K)$ is obtained from K by adding nodes in the middle of edges. As pictured in Figure 14, let v'_1, v'_2, v'_3 be the nodes adjacent to r in K, with v'_i appearing at the end of a path in $\sigma(K)$ starting at v_i, and let u_1, u_2 be the nodes in K that are at distance 2 from r. For any u, v, define $P(u, v)$ to be the unique path in $\sigma(K)$ that does not go through any degree 3 node except at its endpoints. Now we want to show that there exists such a $\sigma(K)$ with the length of $P(r, v'_i)$ equal to 1 for all i. Take a $\sigma(K)$ as above so that the lengths of $P(r, v'_i)$ are minimal. Suppose that there exists some path P from s to t, disjoint from $\sigma(K)$ except at endpoints, with $s \in P(r, v'_1)$. If $t \in P(v'_1, u_1)$, then we can form a new $K_{3,3}$ instance by replacing $P(t, v'_1)$ with P, as in Figure 14(Middle). This shortens the length of $P(r, v'_1)$, since s becomes the "new" v'_1, giving a contradiction. If $t \in P(u_1, v'_2)$, we can similarly re-design K by replacing $P(u_1, v'_1)$ with P, and making t the "new" u_1, as in Figure 14(Right). This

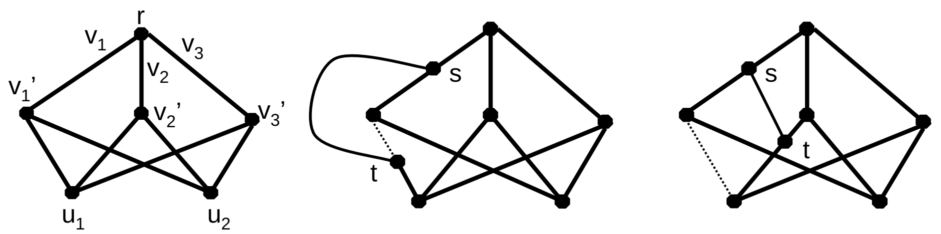

Figure 14: Proof of Lemma 4.7 (Left) A subdivision of $K_{3,3}$ (Middle) An s-t path with $t \in P(u_1, v_1')$ (Right) An s-t path with $t \in P(u_1, v_2')$

also shortens $P(r, v_1')$, and all other cases can be reduced to these, so we can assume that all paths from nodes in $P(r, v_i')$ must pass through v_1', v_2', v_3' to reach nodes outside of $\cup_i P(r, v_i')$. However, if $v_i \neq v_i'$ for all i, then the cut consisting of the closest edges to v_i' on each $P(r, v_i')$ disconnects the graph with more than one node on each side, which we assumed is impossible. This gives us that $v_i = v_i'$, as desired.

Take this $\sigma(K)$, which is really the union (although not a valid sum) of two dual augmentors containing r: one with degree 3 at u_1, and one with degree 3 at u_2; both with degree 3 at r. We will now form a valid augmentor sum α for $\overline{S}$ with the cover number of $2x$. Set the α value of each of these dual augmentors to some value x. To complete this augmentor sum, we need to cover all edges not in $\sigma(K)$ by $2x$ dual augmentors, and all edges in $\sigma(K)$ not adjacent to r by x. We do not need to cover the edges adjacent to r anymore, since we already covered them with $2x$, so we may as well remove them. This gives us three nodes of degree 2. Suppose v is such a node, with incident edges (v, w_1) and (v, w_2). v is not in M, so any dual augmentor must contain both (v, w_1) and (v, w_2). Without loss of generality, we can replace such nodes v and both incident edges with a single edge (w_1, w_2). This results in a cubic graph with edges that we need to cover either x or $2x$ times. By Theorem 4.2, we know we can cover this graph with cycles in the desired manner if it is 3-connected. If it were not, this would mean that there is some cut in this graph consisting of only two edges f_1, f_2. This graph was constructed by removing r, and then getting rid of the remaining degree 2 nodes, so this means that removing two edges f_1, f_2 from $\overline{S}$ together with r disconnects $\overline{S}$. Since r is degree 3, this implies that removing three edges from $\overline{S}$ disconnects it. By our assumption, the only way this is possible is if one side of the cut consists of a single node v that is adjacent to f_1, f_2, and r. However, we contracted all nodes adjacent to r, so $\{f_1, f_2\}$ could not be a cut in the resulting graph. Therefore there exists a circuit sum of this graph, giving us a valid augmentor sum together with the the augmentors in $\sigma(K)$.

Planar. We now address the planar case. Tait [32] showed that the Four Color Theorem [2, 30] is equivalent to the following statement: "Every 2-connected cubic planar graph is edge-3-colorable," and more recently [6] proved this statement without relying on the Four Color Theorem. Take such a coloring of $\overline{S}$, where each color just forms a perfect matching. Call these matchings M_1, M_2, M_3 and form symmetric differences $M_1 \oplus M_2, M_2 \oplus M_3$, and $M_3 \oplus M_1$. Each of these is a set of disjoint cycles, with every edge being in exactly two of these. Consider what $M_1 \oplus M_2$ looks like with respect to r, and let C_1 be the cycle of it containing r, and C_2 be the cycle of it containing the node adjacent to r attached by an edge e of M_3. Form a dual augmentor by taking $C_1 \cup C_2 \cup \{e\}$ (note that C_1 may equal C_2). Figure 15 shows what this object can look like. To check that this is a dual augmentor, notice that the only nodes of degree 3 are the endpoints of e, since C_1 and C_2 are disjoint. The only node of M is r, which has degree 3, as desired.

Now, take all the cycles of $M_1 \oplus M_2, M_2 \oplus M_3, M_3 \oplus M_1$, but replace C_1, C_2 by $C_1 \cup C_2 \cup \{e\}$ (and similarly for $M_2 \oplus M_3$ and $M_3 \oplus M_1$). If we take all of these dual augmentors and cycles, we get a dual augmentor cover that covers the edges next to r three times and all the other edges twice.

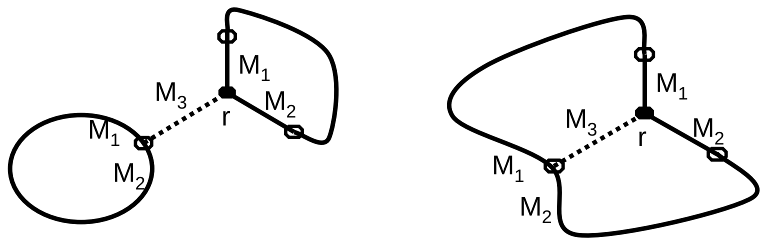

Figure 15: The planar case in Lemma 4.7. (Left) $C_1 \neq C_2$ (Right) $C_1 = C_2$

Remove r and its adjacent edges. As in the non-planar case, we can use Theorem 4.2 to show that we can cover the resulting graph with cycles so that every edge appears in exactly the same number x of cycles. Together this gives a valid augmentor sum, since we can multiply the cover above by x, combine it with the cycle cover, and end up with a valid augmentor sum of size $3x$. $\square$

Theorem 5.1 *Our algorithm solves weighted* Simplex Matching *in polynomial time.*

Proof: Let OPT be the cost of the best perfect matching M^*, and let M be some perfect matching. By Lemma 3.2 and Theorem 3.3 we know that there exists an augmentor A with $\phi_M(A) \geq \frac{\phi_M(M^* \oplus M)}{n}$, where n is the number of nodes in G. By Lemma 2.1 we can efficiently find an M-alternating 2-factor S such that $\phi_M(S) \geq \phi_M(A)$. Since $\phi_M(M^* \oplus M) = \mathit{cost}(M) - OPT$, then every time we augment in our algorithm, we decrease the cost by at least $(\mathit{cost}(M) - OPT)/n$.

Therefore, in the above algorithm, if we start with a matching that is D more expensive than OPT, then we will decrease D by at least a factor of $\frac{n-1}{n}$ at every step. So at the k'th step we have a solution of cost at most

$$D \cdot (\frac{n-1}{n})^k.$$

Therefore, it will take

$$\frac{\log D}{\log(n/n-1)}$$

steps until we find a perfect matching with $\mathit{cost}(M) - OPT < 1$. Since we have integer weights, this matching must be optimal.

$$\log \frac{n}{n-1} = \log n - \log(n-1) > 1/n,$$

so we require at most $n \log D$ steps. In general, D can be as large as $n \cdot OPT$, but for most of our applications, we can find an initial matching which is a close approximation to OPT, so $D < OPT$.

Done in a naive manner, each step consists of running a min-cost weighted matching algorithm for every pair of 3d edges. This could take as long as $O(n^3 m^2)$, where m is the number of 3d edges. However, there are some easy ways to make this run faster. We should take advantage of the fact that the min-cost matchings that we are calculating are extremely related. If we use an "augmenting path" algorithm for calculating min-cost matchings [13], then each calculation only takes $O(n^2)$ time, giving us a time of $O(n^3 + n^2 m^2)$. We can reduce this further in the geometric setting. Finally, notice that we do not need to consider all pairs of 3d edges. A lot of these pairs can be eliminated in advance, significantly reducing the running time. This is especially true when applying this algorithm to the Terminal Backup problem, or to any problem involving covering instead of exact matching. For more details, see [33].

In the case where the edge costs are not integer, the running time will depend on how these costs are represented. By running the augmentation algorithm until the improvement is at most ε, we can obtain an algorithm that finds a solution that costs at most $OPT + \varepsilon$ in time polynomial in $\log D$, $\log(1/\varepsilon)$ and H. $\square$