A $\frac{9}{7}$-approximation algorithm for Graphic TSP in cubic bipartite graphs

Jeremy A. Karp *, R. Ravi
Tepper School of Business, Carnegie Mellon University, United States

A R T I C L E   I N F O

Article history:
Received 11 May 2015
Received in revised form 23 October 2015
Accepted 27 October 2015
Available online 4 December 2015

Keywords:
Approximation algorithms
Traveling salesman problem
Barnette’s conjecture
Combinatorial optimization

A B S T R A C T

We prove new results for approximating the Graphic TSP. Specifically, we provide a polynomial-time $\frac{9}{7}$-approximation algorithm for cubic bipartite graphs and a $(\frac{9}{7} + \frac{1}{2k-2})$-approximation algorithm for $k$-regular bipartite graphs, both of which are improved approximation factors compared to previous results. Our approach involves finding a cycle cover with relatively few cycles, which we are able to do by leveraging the fact that all cycles in bipartite graphs are of even length along with our knowledge of the structure of cubic graphs.

© 2015 Elsevier B.V. All rights reserved.

1. Introduction

1.1. Motivation and related work

The traveling salesman problem (TSP) is one of most well known problems in combinatorial optimization, famous for being hard to solve precisely. In this problem, given a complete undirected graph $G = (V,E)$ with vertex set $V$ and edge set $E$, with non-negative edge costs $c \in \mathbb{R}^{|E|}$, $c \neq 0$, the objective is to find a Hamiltonian cycle in $G$ of minimum cost. In its most general form, TSP cannot be approximated in polynomial time unless $P = NP$. In order to successfully find approximate solutions for TSP, it is common to require that instances of the problem have costs that satisfy the triangle inequality ($c_{ij} + c_{jk} \geq c_{ik}$ $\forall i, j, k \in V$). This is the Metric TSP. The Graphic TSP is a special case of the Metric TSP, where instances are restricted to those where $\forall i, j \in E$, the cost of edge $(i,j)$ in the complete graph $G$ are the lengths of the shortest paths between nodes $i$ and $j$ in an unweighted, undirected graph, on the same vertex set.

One value related to the ability to approximate TSP is the integrality gap, which is the worst-case ratio between the optimal solution for a TSP instance and the solution to a linear programming relaxation called the subtour relaxation [6]. A long-standing conjecture (see, e.g., [10]) for Metric TSP is that the integrality gap is $\frac{4}{3}$. One source of motivation for studying the Graphic TSP is that the family of graphs with two vertices connected by three paths of length $k$ has an integrality gap that approaches $\frac{4}{3}$. This family of graphs demonstrates that The Graphic TSP captures much of the complexity of the more general Metric TSP.

For several decades, the Graphic TSP did not have any approximation algorithms that achieved a better approximation than Christofides’ classic $\frac{1}{2}$-approximation algorithm for Metric TSP [4], further motivating the study of this problem. However, a wave of recent papers [8,1,3,9,12,5,14] have provided significant improvements in approximating the Graphic...
TSP. Currently, the best known approximation algorithm for the Graphic TSP is due to Sebő and Vygen [14], with an approximation factor of $\frac{2}{7}$.

Algorithms with even smaller approximation factors have also been found for the Graphic TSP instances generated by specific subclasses of graphs. In particular, algorithms for the Graphic TSP in cubic graphs (where all nodes have degree 3) have drawn significant interest as this appears to be the simplest class of graphs that has many of the same challenges as the general case. Currently, the best approximation algorithm for the Graphic TSP in cubic graphs is due to Correa, Larré, and Soto [5], whose algorithm achieves an approximation factor of $\left(\frac{4}{7} - \frac{1}{6125}\right)$ for 2-edge-connected cubic graphs. Similarly, a $\frac{2}{7}$-approximation was recently obtained for instances of sub-quartic graphs [13]. In a recent preprint, van Zuylen [15] extends the work in [5,12] to obtain a $\frac{2}{7}$-approximation algorithm for cubic bipartite graphs and a $\left(\frac{4}{7} - \frac{1}{875}\right)$-approximation algorithm for 2-connected cubic graphs.

Progress in approximating the Graphic TSP in cubic graphs also relates to traditional graph theory, as Barnette’s conjecture [2] states that all bipartite, planar, 3-connected, cubic graphs are Hamiltonian. This conjecture suggests that instances of Graph TSP on Barnette graphs could be easier to approximate, and conversely, approximation algorithms for the Graphic TSP in Barnette graphs may lead to the resolution of this conjecture. Indeed, Correa, Larré, and Soto [5] provided a $\left(\frac{4}{7} - \frac{1}{18}\right)$-approximation algorithm for Barnette graphs. Along these lines, Aggarwal, Garg, and Gupta [1] were able to obtain a $\frac{4}{7}$-approximation algorithm for 3-edge-connected cubic graphs before any $\frac{3}{7}$-approximation algorithms were known for all cubic graphs. In this paper, we examined graphs that are cubic and bipartite, another class of graphs that includes all Barnette graphs. An improved approximation for this class of graphs is the primary theoretical contribution of this paper.

**Theorem 1.1.** Given a cubic bipartite connected graph $G$ with $n$ vertices, there is a polynomial time algorithm that computes a spanning Eulerian multigraph $H$ in $G$ with at most $\frac{9}{7}n$ edges.

**Corollary 1.2.** Given a $k$-regular bipartite connected graph $G$ with $n$ vertices where $k \geq 4$, there is a polynomial time algorithm that computes a spanning Eulerian multigraph $H$ in $G$ with at most $\left(\frac{2}{7} + \frac{1}{7(k-2)}\right)n - 2$ edges.

This extension complements results [16,7] which provide guarantees for $k$-regular graphs in the asymptotic regime. Corollary 1.2 improves on these guarantees for small values of $k$. Note that even for $k = 4$ Corollary 1.2 yields a solution with fewer then $\frac{4}{7}n$ edges.

### 1.2. Overview

In this paper, we will present an algorithm to solve the Graphic TSP, which guarantees a solution with at most $\frac{9}{7}n$ edges in cubic bipartite graphs. The best possible solution to the Graphic TSP is a Hamiltonian cycle, which has exactly $n$ edges, so this algorithm has an approximation factor of $\frac{9}{7}$.

A corollary of Petersen’s theorem is that every cubic bipartite graph contains three edge-disjoint perfect matchings. The union of any 2 of these matchings forms a 2-factor. The following proposition demonstrates the close relationship between 2-factors and Graphic TSP tours in connected graphs.

**Proposition 1.3.** Any 2-factor with $k$ cycles in a connected graph can be extended into a spanning Eulerian multigraph with the addition of exactly $2(k - 1)$ edges. This multigraph contains exactly $n + 2(k - 1)$ edges in total.

Proposition 1.3 can be implemented algorithmically by compressing each cycle in the 2-factor into a single node and then finding a spanning tree in this compressed graph. We then add two copies of the edges from this spanning tree to the 2-factor. We present an algorithm, BIGCYCLE, which begins by finding a 2-factor with at most $\frac{n}{7}$ cycles. Then, it applies Proposition 1.3 to generate a spanning Eulerian subgraph from this 2-factor containing at most $n + 2 \times \left(\frac{9}{7} - 1\right) = \frac{9}{7}n - 2$ edges.

BIGCYCLE first shrinks 4-cycles in the graph, then it generates a 2-factor in the condensed graph. If the resulting 2-factor has no 4- or 6-cycles, then we can expand the 4-cycles and this will be our solution. If the 2-factor has a 4-cycle, then we either contract it or show that this cycle will be extended into a longer cycle when we expand the graph. If the 2-factor does have a 6-cycle, then the algorithm contracts either this 6-cycle or a larger subgraph that includes this 6-cycle. We are able to iterate this process until we find a 2-factor in the compressed graph that can be shown to expand into a 2-factor in the original graph with relatively few cycles. Theorem 3.12 in Section 3.5 proves that this 2-factor has at most $\frac{9}{7}$ cycles.

## 2. A $\frac{9}{7}$-approximation algorithm for Graphic TSP in cubic bipartite graphs

### 2.1. Overview

In a graph with no 4-cycles (squares), all 2-factors will have an average cycle length of at least 6, so all 2-factors will have at most $\frac{9}{7}$ cycles, which results in a $\frac{9}{7}$-approximation after applying Proposition 1.3. In order to improve our approximation guarantee, we need to target 6-cycles, as well as 4-cycles. The algorithm we present finds a square-free 2-factor in which every 6-cycle can be put in correspondence with a distinct cycle of size 8 or larger. Then, we can find a 2-factor in which
every large cycle and its corresponding 6-cycles have average cycle length of at least 7 via an amortized analysis over the compressing iterations (Lemma 3.11 in Section 3.4). We then show that this is enough to conclude that the 2-factor contains at most \( \frac{n}{7} \) cycles (Theorem 3.12 in Section 3.5). This is the primary contribution of this paper.

A method used throughout this paper is to systematically replace certain subgraphs containing 4-cycles and 6-cycles with other subgraphs. We will refer to these replacement subgraphs as "gadgets". To keep track of portions of the graph that have not been altered by these gadgets, we define the term "organic" as follows.

**Definition 2.1.** A subgraph is organic if it consists entirely of nodes and edges contained in the original graph. For a single edge to be organic, both its end-nodes must be organic.

We also give a formal definition of the term "gadget":

**Definition 2.2.** A gadget is a subgraph that is inserted into the graph by the BIGCYCLE algorithm in place of a different subgraph. Examples of gadgets are shown in Section 2.2. Gadgets are used to replace other subgraphs containing 4- or 6-cycles.

In Section 2.2, we introduce the gadgets used in the BIGCYCLE algorithm. The BIGCYCLE algorithm is defined in Section 2.3. The BIGCYCLE algorithm progressively compresses larger subgraphs by replacing them with gadgets, and computes a carefully chosen 2-factor in the final compressed graph; Then it unwinds these contractions in expansion operations that expand the 2-factor into the whole original graph. In Section 3 we examine expansions that can introduce 6-cycles into our 2-factor and show that while expanding the graph can create a small number of new 6-cycles in our 2-factor, we are able to account for them, ensuring that the bounds described in the previous paragraph must hold.

### 2.2. Gadgets

In this section, we present the subgraphs that will be replaced with gadgets by the algorithm. In Section 2.3 and Appendices A–H we go on to show precisely how these gadgets are used by the algorithm. In total, there are 3 gadgets to replace 4-cycles, 6 gadgets to replace 6-cycles, and 1 gadget to replace parallel edges that are introduced by gadget replacement operations. We will give these configurations the names \( S_1, S_2, S_3, H_1, H_2, H_3, H_4, H_5, H_6, \) and \( D_1 \). The gadget that replaces a configuration \( X \) will be called \( X' \).

First, we introduce the gadget we use to replace squares whose outgoing edges are incident on four distinct vertices (see Figs. 1 and 2).

Next, we introduce the \( S_2 \), used to replace squares with two exiting edges connected to a common vertex (see Figs. 3 and 4).

We also introduce the \( S_3 \), which is used to replace squares whose outgoing edges are incident on only two vertices (see Fig. 5).

When we describe the BIGCYCLE algorithm in Section 2.3, the algorithm’s instructions include recognizing 4-cycles that pass through \( S_3' \)'s. For clarity, in order for a cycle to pass through or include an \( S_3 \) it must contain the edge that connects the nodes labeled \( A \) and \( B \) in Fig. 6.
The first gadget used to replace 6-cycles is two super-vertices which replace a simple 6-cycle, $H_1$ (see Figs. 7 and 8).

The next 5 gadgets are special cases of 6-cycles. Note that every $H_2$ contains a $H_1$, all $H_2$s contain a $H_2$, and $H_4$s, $H_5$s, and $H_6$s are special cases of $H_3$s.

The motivation to use these additional gadgets comes out of necessity, to prevent large numbers of 6-cycles from being introduced into the 2-factor during the expansion phase of the algorithm. For example, Figs. 25 and 26 in Section 3 document an expansion that turns an $x + y + 4$-cycle in the cycle cover passing through a gadget which replaced a $H_1$ into two cycles of lengths $x + 3$ and $y + 5$. In Section 2.3 we specify that the algorithm will condense $H_2$s before $H_1$s. This ensures that $y$, the length of a path, is at least 3, meaning that the $y + 5$-cycle is not a 6-cycle. The motivation for introducing the remaining 6-cycle gadgets is similar (see Figs. 9–18).
Definition 2.3. A subgraph, specifically one of the above configurations, is annihilated if it is replaced with a gadget but all of the edges exiting the configuration lead to nodes within the configuration. Note that replacing the configuration removes it completely from the graph.

As an example, if the edges labeled 1 and 2 in the $H_6$ are the same edge, then replacing this $H_6$ annihilates it. The last gadget, shown in Figs. 19 and 20 replaces parallel edges:
The original input graph is simple, but certain gadget replacements can potentially introduce parallel edges when performed by the BIGCYCLE algorithm described in Section 2.3. We accommodate this by using the $D'_1$ to eliminate these parallel edges.

In the next subsection, we present a detailed description of the algorithm.

2.3. The algorithm

Listing 1 presents pseudocode for the BIGCYCLE algorithm. The remainder of this section explains the details of the algorithm, broken up into three subroutines, and presents motivation for the operations performed by the algorithm. The COMPRESS, EXPAND, and DOUBLETREE subroutines called by BIGCYCLE are described in the following three subsections.
We start the algorithm by receiving a connected cubic bipartite graph. Call this graph $G_0$. If $G_0$ is a $K_{3,3}$ then we compute a 2-factor in this graph, which will be a Hamiltonian cycle, and return this cycle as our solution. Otherwise, we search for replaceable 4-cycles that are not contained in $K_{3,3}$ and replace them with their corresponding gadgets until we are returned a graph with no replaceable squares except possibly inside of $K_{3,3}$. We replace $S_2$ subgraphs first, followed by $S_3$s and $S_1$s so as to replace the most specialized subgraphs first. Note that if an $S_2$ or $S_3$ is present but contains the central edge of an $S'_3$, a subset of this configuration might be a replaceable $S_3$ or $S_1$ which would get replaced instead. If these replacements introduce a $D_1$ then we replace the $D_1$. If replacing a $D_1$ introduces an additional $D_1$, then we perform $D_1$ replacements until the compressed graph is simple. The only way in which the graph might not be simple is if there is a connected component of two nodes connected by three parallel edges. Let $i$ be the number of square and $D_1$ compressions made, and let $G_i$ be the compressed graph at the end of this process. Next, construct a 2-factor, $F_{i}$, in $G_i$. When we construct 2-factors throughout the algorithm, we do so by decomposing each connected component of the graph into 3 edge-disjoint perfect matchings and taking the union of the two perfect matchings containing the most $S'_3$ gadgets that replaced organic $S_3$s. These two perfect matchings form a 2-factor with limited potential to introduce organic 6-cycles of the type shown in Fig. 35 (Section 3.2), If $F_i$ contains no organic 6-cycles, then we advance to the next phase of the algorithm, described in the next subsection. In this case, $k = i$.

If $F_i$ does contain an organic 6-cycle, $C$, then we check if the current compressed graph $G_i$ contains organic subgraphs that can be replaced by gadgets in the following order (ordered from most specialized to most general): $H_6$, $H_5$, $H_4$, $H_3$, $H_2$, $H_1$. We choose the first organic configuration on the list (the most specialized configuration) we can find in $G_i$ and replace this configuration with the corresponding gadget, outputting graph $G_{i+1}$ to reflect this change. The order of choosing subgraphs to replace is useful in accounting for the average length of the cycles in the final 2-factor, as shown in the proof of Lemma 3.10. We then search for $D_1$s and replaceable 4-cycles that are not contained in $K_{3,3}$ and replace them with their corresponding gadgets until we have removed all $D_1$s and replaceable 4-cycles generated as a consequence of replacing a subgraph with one of our gadgets, obtaining a new compressed graph $G_{j}$, where $j = i + 1$ is the number of $D_1$s and replaceable 4-cycles compressed. $D_1$s are always given highest priority, meaning whenever the graph has a $D_1$, the next replacement is of a $D_1$. We construct a new 2-factor $F_j$ and repeat the process in this paragraph until we have a 2-factor $F_k$ with no organic 6-cycles, in a condensed graph $G_k$, where $k$ is the total number of gadget replacement operations performed during this phase of the algorithm. This process is performed by the COMPRESS subroutine in Listing 1.

### 2.3.2. Expanding a 2-factor in $G_k$ into a 2-factor in $G_0$

We will describe the process of expanding $F_k$ and $G_k$ so that we get back to the original graph $G_0$ with a desirable 2-factor $F_0$ in more detail.

We will reverse the process described in the previous subsection by replacing our gadgets in compressed graph $G_m$ (for $1 \leq m \leq k$) with the original configuration from the earlier graph $G_{m-1}$ in the reverse order of that in which we replaced the configurations. In other words, the gadgets we inserted last are those which we first replace with their original configuration. We call this process “expanding” because each one of these operations adds vertices and edges to the graph. After we have made each replacement to expand the graph, $F_m$ is no longer a 2-factor in $G_{m-1}$ because the new nodes added by the most recent expansion step are not covered by $F_m$. However, we can add edges to $F_m$ so that it becomes a 2-factor, $F_{m-1}$ in the graph after this expansion step. It may not be immediately clear that this is always possible. In fact, one of the bigger challenges in developing this algorithm was choosing a set of gadgets where this property holds. Figs. 21 and 22 show an example of how this process works. The exact sets of edges added for all expansions are listed in Appendixes A–H.

At each expansion, we are able to extend $F_{m-1}$ into a set of edges $F_{m-1}$, which will be a 2-factor in the expanded graph, $G_{m-1}$. Fig. 21.

![Fig. 21](image-url) A pair of super-vertices in $G_m$. The bold edges are included in 2-factor $F_m$. The dashed bold edges represent a path, included in $F_m$. **Algorithm 1**

```
Algorithm 1 BIGCYCLE(G)
Input: An undirected, unweighted, cubic, bipartite graph, $G = (V, E)$
F_{compressed} ← COMPRESS(G)
F ← EXPAND(F_{compressed})
TSP ← DOUBLETREE(G, F)
Return TSP
```

### 2.3.1. Finding a "good" 2-factor in the condensed graph $G_k$

**Definition 2.4.** Replaceable 4-cycles are 4-cycles that do not contain an edge that is the central edge of an $S'_3$.

It may not be immediately clear that this is always possible. In fact, one of the bigger challenges in developing this algorithm was choosing a set of gadgets where this property holds. Figs. 21 and 22 show an example of how this process works. The exact sets of edges added for all expansions are listed in Appendixes A–H.

At each expansion, we are able to extend $F_{m-1}$ into a set of edges $F_{m-1}$, which will be a 2-factor in the expanded graph, $G_{m-1}$. Fig. 21.
In order to optimize the performance of BIGCYCLE, we must impose one extra operation in this phase of the algorithm. After each expansion of a $H_1$ that introduces an organic 6-cycle, $C_1$, into the 2-factor, we will perform a local search to see if $C_1$ and the nearby edges of the newly expanded 2-factor $F_{m-1}$ contained in the surrounding portion of the graph are organic and in the position shown in Fig. 23. If they are, then we update $F_{m-1}$ so that it covers this region with one fewer cycle, as shown in Fig. 24. In addition to being an effective heuristic to reduce the number of cycles in our final 2-factor, this operation allows us to improve our approximation factor by eliminating an otherwise troubling corner case (see Remark 1, Fig. 28, and Lemma 3.10). Note that this operation will maintain or increase the number of $S_3$'s covered by the 2-factor as none of the edges removed from the 2-factor can be $S_3$'s but the edges added might be $S_4$'s.

At this point, we can repeat the process of replacing gadgets with their original configurations and adding edges to the 2-factor until we have expanded the graph back to the original input $G_0$ and have a 2-factor, $F_0$, in this graph. This process is performed by the EXPAND subroutine in Listing 1.

### 2.3.3. Obtaining a good final solution by adding edges to $F_0$

We now have a 2-factor $F_0$, which contains at most $x$ cycles. We compress each cycle into a single node and compute a spanning tree in this compressed graph. This spanning tree has $x - 1$ edges. Then, we add two copies of the edges in this spanning tree to our 2-factor $F_0$ to obtain a solution with $n + 2(x - 1)$ edges. In Section 3 we prove that $F_0$ has at most $\frac{n}{2}$ cycles, so this gives us a solution of at most $\frac{9}{7}n - 2$ edges. This process is performed by the DOUBLETREE subroutine in Listing 1.

**Proposition 2.5.** Given a simple cubic bipartite graph as input, all of the gadget replacement operations performed by the BIGCYCLE algorithm output a cubic bipartite graph.
Fig. 25. A cycle in $F_i$, a 2-factor over the condensed graph $G_i$, $SV_1$ and $SV_2$ are two super-vertices which replaced a standard hexagon. The dashed lines represent paths of length 3.

Fig. 26. The cycle from Fig. 25, after expanding $SV_1$ and $SV_2$.

Proof. We see that the output graph is cubic and bipartite by inspection of the gadget replacement operations. For each of the configurations and the gadgets we replace them with, we can place all vertices neighboring the configuration into two groups based on the bipartition of the input graph. We observe that all added nodes that are part of the gadgets we defined can be consistently placed into this bipartition. Furthermore, all gadget edges cross the bipartition. The output graph will be cubic because all nodes that are part of gadgets are degree 3, and the degrees of nodes outside of the configurations that get replaced do not change. ☐

3. Accounting for 6-cycles

In the proof of our approximation guarantee, the limitation on producing a lower approximation factor comes from the possibility that some proportion of our final 2-factor’s cycles will be of length 6. Most operations the algorithm performs while expanding the 2-factor from the condensed to the original graph result in cycles of length 8 or larger, so in this section we will look at all operations that create organic 6-cycles in detail. To account for 6-cycles, we show that every organic 6-cycle can be put in correspondence with some long cycle of length 8 or longer. Then, Lemma 3.11 demonstrates that the average cycle length of any long cycle and its corresponding set of 6-cycles is sufficiently long to ensure that our final cycle cover has relatively few cycles, even if some of them are 6-cycles. Note that immediately after expanding a $D'_i$, it is possible that our 2-factor has a cycle of two parallel edges. However, these cycles are not organic and we account for them by tracking when short organic cycles are introduced as the graph and our 2-factor expand.

Figs. 25 and 26, taking the dashed lines to be paths of lengths $x$ and $y$, demonstrate how a $(y + 7)$-cycle can turn into a 6-cycle and a $(y + 5)$-cycle after an expansion if $x = 3$.

In Sections 3.1–3.5 we will carefully analyze this and several other cases that form the bottleneck in our analysis, which occur when expanding our graph back to its original state. In Section 3.7 we will account for organic 6-cycles by creating a correspondence from every 6-cycle in the final 2-factor $F_0$ to some larger cycle in $F_0$. This way, if we can show that every large cycle of length $l \geq 8$ in $F_0$ is affiliated with at most $f(l)$ 6-cycles, then the average cycle length is at least $\min_{l\geq 8} \frac{6f(l)+l}{f(l)+1}$. Once we have placed a lower-bound on the average cycle length in this manner (Lemma 3.11), we can easily determine our approximation factor (Theorems 1.1 and 3.12).

We now state some definitions about “protected edges”, organic paths which help us formalize the correspondence between 6-cycles and larger cycles in our 2-factor. We use this term because protected edges cannot be separated from others in the 2-factor during subsequent expansion operations.
Definition 3.1. Protected edges are sets of edges that are marked by the BIGCYCLE algorithm during its EXPAND phase. Protected edges are always organic paths that are part of the 2-factor maintained by BIGCYCLE. The edges labeled “P” in Fig. 26 are an example of a set of protected edges.

The next two definitions provide a framework for using protected edges to establish a correspondence between 6-cycles and larger cycles.

Definition 3.2. For a given 6-cycle, $C_i$ in the final 2-factor, $F_0$, consider the value $i$ such that $C$ is a cycle of $F_i$ but not of $F_{i+1}$. Then, it was the expansion operation from $G_{i+1}$ to $G_i$ which “finalized” this cycle. Then, the protected edges identified during this “finalizing” operation are defined to be the protected edges corresponding to $C$.

Definition 3.3. For any cycle, $C_i$, of length at least 8, in the 2-factor $F$ we will define a set of 6-cycles, $S_{C_j}$ which correspond to $C_i$. For each 6-cycle, $C_i$, in $F$, we say $C$ is an element of $S_{C_j}$ if $C$’s protected edges are in $C_i$ or if $C$’s protected edges are in another 6-cycle $C’$ whose protected edges are in $C_i$.

3.1. Expanding $h_1$ gadgets

Appendix B in the Appendix documents, in detail for all cases, the process of winding the 2-factor through a $H_1$ after the algorithm has expanded a $H'_i$, the $H'_i$’s replacement gadget. An examination of this Appendix confirms that the example shown in Figs. 25 and 26 is the only type of operation involving the $H'_i$ gadget that can introduce an organic 6-cycle into the 2-factor during an expansion. This example is found in Figs. 65 and 66 in Appendix B. There are also expansions with a similar outcome which involve the $H'_2$ and $H'_3$ gadgets, respectively. The analysis to account for these expansions is very similar to the analysis of this case. Furthermore, these “bad” $H'_2$ and $H'_3$ expansions introduce larger sets of protected edges than the $H_1$ expansion in Figs. 25 and 26, so this $H_1$ expansion is what limits the approximation factor we obtain in our analysis. Note that it is possible that the path from $A_2$ to $B_2$ in the 2-factor that is shown to have length $y + 2$ in Fig. 26 could instead be a single $S_j$ edge. If this were to occur, the accounting for the organic 6-cycle that was introduced is the same, the same edges labeled “P” remain protected, and when the $S_j$ gets expanded we are in the state shown in Figs. 25 and 26.

Remark 1. If an expansion operation of the type shown in Figs. 25 and 26 were to occur, then it is not possible for the nodes corresponding to $A_1$ and $B_1$, in these figures to be the super-vertices of a $H'_i$ whose expansion places the protected edges in Fig. 26 into an organic 6-cycle. It is also not possible for this expansion to introduce a new organic 6-cycle and for this cycle’s protected edges as well as the previous set of protected edges to be contained in an organic 8-cycle. Both of these possibilities require $A_1$ and $B_1$ to be the vertices of an $H'_i$ and for the expansion of this $H'_i$ to introduce an organic 6-cycle, so we will consider the ways in which this can happen and show that these expansions do not lead to the two possibilities mentioned in the previous two sentences.

Appendix B.2 examines the expansions of $H'_i$’s where the gadget is covered by a single cycle. There are only two expansions in this section that introduce organic 6-cycles, and they are isomorphic to each other. If $A_1$ and $B_1$ were the super-vertices of a $H'_i$ whose expansion introduces an organic 6-cycle into the 2-factor, we must consider two cases. The path that goes through nodes $A_2$ and $B_2$ could end up either in the 6-cycle or the longer cycle after the expansion. In the first case, the graph $G_{i-1}$ would have contained a $H_2$ and would have been compressed differently at the time when $A_1$ and $B_1$ would have been created during a compression (see Fig. 27). Then, at this stage, the algorithm would have replaced a $H_1$ for some $i \geq 2$ at this time, not a $H_1$, which would have been necessary to create $A_1$ and $B_1$ as specified. In the second case, upon expanding, the graph $G_{i-1}$ and 2-factor $F_{i-1}$ would be in the configuration shown in Fig. 23, prompting a local improvement so that this expansion no longer introduces an organic 6-cycle and places its protected edges in an organic 8-cycle containing other protected edges. Fig. 27 shows the first case where $A_2$ and $B_2$ are in the 6-cycle after the expansion, and Fig. 28 shows the second case where $A_2$ and $B_2$ are in the longer cycle after the expansion.

3.2. Expanding $h_2$ gadgets

Appendix C in the Appendix documents, in detail for all cases, the process of winding the 2-factor through a $H_2$ after the algorithm has expanded a $H'_i$, the $H'_i$’s replacement gadget. An examination of this Appendix confirms that the example shown in Figs. 29 and 30 is the only type of operation involving the $H_2$ configuration that can introduce an organic 6-cycle into the 2-factor during an expansion. This example is found in Figs. 79 and 80 in Appendix C. Note that if $x = 1$ in Fig. 30 then this would be a 4-cycle, but by the construction of the BIGCYCLE algorithm this would mean that one of its edges is a $S'_j$ and so it is not an organic 4-cycle. Then if this cycle is organic it must have length at least 6.

This expansion introduces a set of five protected edges, rather than three for the $H_1$, when the graph is expanded. Furthermore, as we continue to expand the graph the organic cycle that contains these protected edges will be length 8 or greater. For the protected edges to become part of a 6-cycle from expanding $H_1$ or $H_2$ gadgets, they must be in between two corresponding super-vertices located exactly five edges apart on one of the cycles of the 2-factor. The path between $A_2$ and $B_2$ of protected edges, shown in Fig. 30, is five edges long and organic, so any super-vertices in this cycle are separated on this side of the cycle by more than five edges. Therefore, if these protected edges will end up in an organic cycle of length at least 8 in the final 2-factor.
Fig. 27. The cycles from Fig. 26, after expanding $A_1$ and $B_1$, if these nodes had been a $H_1$’s gadget whose expansion introduced a 6-cycle, where the path through $A_2$ and $B_2$ is in a 6-cycle after the expansion. We can see in this figure that nodes $A_2$ and $B_2$ are part of an organic $H_2$. Then, $A_1$ and $B_1$ cannot be a $H_1$’s gadget in this configuration, otherwise the BIGCYCLE algorithm would have performed different operations, compressing this $H_2$ or some other $H_i$, for some $i \geq 2$ instead of the $H_1$ that $A_1$ and $B_1$ replaced.

Fig. 28. The cycles from Fig. 26, after expanding $A_1$ and $B_1$, if these nodes had been a $H_1$’s gadget whose expansion introduced a 6-cycle, where the path through $A_2$ and $B_2$ is in the longer cycle after the expansion. We can see in this figure the cycle containing path $y$ corresponds to 6-cycle $C_1$ in the configuration shown in Fig. 23. Then if the edges in the 2-factor shown in this figure are organic, the algorithm would have updated the 2-factor to the configuration shown in Fig. 24, making path $y$ part of a 10-cycle. Then, because of this local correction, this expansion step would not have introduced an organic 6-cycle into the 2-factor and an organic 8-cycle containing two sets of protected edges.

Fig. 29. A cycle in $F$, a 2-factor over the condensed graph $G$, $SV_1$ and $SV_2$ are two super-vertices which replaced a $H_2$. 
Fig. 30. The cycle from Fig. 29, after expanding $SV_1$ and $SV_2$.

Fig. 31. Two cycles of lengths $x + 1$ and $y + 1$ pass through a $H'_3$.

Fig. 32. The cycles from the previous figure, after expanding the gadget, are now cycles of lengths $x + 5$ and $y + 7$, respectively. If $x = 1$ then this expansion introduces a 6-cycle.

Proposition 3.4. A cycle $C$ of length $\alpha$ in the final 2-factor produced by BIGCYCLE has at most $\frac{\alpha}{5}$ affiliated 6-cycles that were formed during the expansion of a $H_2$ gadget.

Proof. Suppose for the sake of contradiction that this cycle has $\beta > \frac{\alpha}{5}$ affiliated 6-cycles that were formed during the expansion of a $H_2$ gadget. Each of these 6-cycles was formed during a distinct expansion operation (no expansion operation introduces more than one organic 6-cycle), so the protected edges for each of these cycles are disjoint. Each of these 6-cycles has 5 protected edges, so $5\beta > \alpha$ of the edges in $C$ are protected edges affiliated with 6-cycles that were formed during the expansion of a $H_2$ gadget. This is a contradiction because $C$ has fewer than 5$\beta$ total edges. \qed

This “bad” expansion, then, is very similar to the “bad” expansion of $H_1$ gadgets. In fact, the main difference is that expanding these $H_2$ gadgets is less costly because such an expansion protects more edges than the corresponding $H_1$ gadget expansion. Consequently, in our worst-case analysis, we will tend to discuss the $H_1$ gadget expansion as this will be sufficient to analyze worst-case performance of the algorithm.

3.3. Expanding $H_3$ gadgets

Appendix D in the Appendix documents, in detail for all cases, the process of winding the 2-factor through a $H_3$ after the algorithm has expanded a $H'_3$, the $H_3$’s replacement gadget. An examination of this Appendix confirms that there are two operations involving the $H_3$ configuration that can introduce an organic 6-cycle into the 2-factor during an expansion. The first operation is shown in Figs. 31 and 32, and the second is shown in Figs. 33 and 34. The first example is found in Figs. 99–100 and Figs. 101–102 and the second example is found in Figs. 115–116, all in Appendix D.

Very much like the previous section, upon introducing an organic 6-cycle this first expansion introduces a long cycle with a path of 5 protected edges which will be used to account for the 6-cycle. Note that for this event to occur, compressing the $H_3$ that is now being expanded to introduce a 6-cycle must have introduced a parallel edge.

The second case where expanding a $H'_3$ can split off a 6-cycle are substantially different from those we examined for $H'_1$, $H'_2$s, or the first case for $H'_3$s. This is because $H_3$s are replaced by super-edges rather than from super-vertices. Figs. 33 and 34 demonstrate this operation:
Fig. 33. A cycle in $F_i$, a 2-factor over the condensed graph $G_i$. The edges $(S, B_1)$ and $(S, B_2)$ are two of the three super-edges that replaced a $H_3$. The dashed line is a path of length $x$, where $x \geq 4$ and even.

Fig. 34. The cycle from Fig. 33, after expanding the two super-edges.

Fig. 35. A $S_3'$ in $G_i$ which is not covered by the 2-factor $F_i$ (left), and the $S_3$ in $G_{i-1}$ that replaced the $S_3'$ after expansion (right).

In these figures, we see that this expansion requires two of the $H_3'$ super-edges to be directly neighboring each other in a cycle of the 2-factor. When this expansion is performed, the two edges are split off and form a 6-cycle while the larger cycle they came from increases in length by four. These four new edges are protected.

**Proposition 3.5.** A cycle, $C$, of length $\alpha$ has at most $\frac{\alpha}{4}$ affiliated 6-cycles that were formed during the expansion of a $H_3$ gadget.

**Proof.** Suppose for the sake of contradiction that this cycle has $\beta > \frac{\alpha}{4}$ affiliated 6-cycles that were formed during the expansion of a $H_3$ gadget. Each of these 6-cycles was formed during a distinct expansion operation (no expansion operation introduces more than one organic 6-cycle), so the protected edges for each of these cycles are disjoint. Each of these 6-cycles has at least 4 protected edges, so $4\beta > \alpha$ of the edges in $C$ are protected edges affiliated with 6-cycles that were formed during the expansion of a $H_3'$. This is a contradiction because $C$ has fewer than $4\beta$ total edges.

3.4. Expanding gadgets that replaced squares

Expanding $S_3'$s that replaced organic $S_3$s, of the type shown in Fig. 35, can introduce organic 6-cycles into the 2-factor. This example is found in Figs. 51 and 52 in Appendix A. However, we limit the number of these $S_3'$s whose edges are not in the 2-factor by computing three disjoint perfect matchings and taking as our 2-factor the union of the two perfect matchings that contain the most edges in $S_3'$s that replaced organic $S_3$s. Then, at most $\frac{1}{4}$ of the $S_3'$s that replaced organic $S_3$s will have edges not included in the 2-factor. The only $S_3'$s of this type that are not in the most compressed graph are those that were parallel edges removed by $D_1$ compressions. When we expand a $D_1$, if the central edge is not used in the 2-factor then both parallel edges are part of the expanded 2-factor. If the central edge is used in the 2-factor then one of the two parallel edges
Fig. 36. A $S'_3$ in $G_i$ that is covered by the 2-factor $F_i$ (left), and the $S_3$ in $G_{i+1}$ that replaced the $S'_3$ after expansion (right). The 5 bold edges labeled “P” will be the 6-cycle in Fig. 35’s protected edges.

Fig. 37. The 2-factor does not pass through the $D'_1$.

are part of the 2-factor. In Appendix H, where $D'_1$ expansions are described in detail, it states that edges of $S'_3$’s that replaced organic $S_3$’s are given priority over other edges in choosing which parallel edge to include in the 2-factor after expanding a $D'_1$. Consequently, at most $\frac{1}{2}$ of the $S'_3$’s which replaced organic $S_3$’s that are removed by $D_1$ replacements will have central edges not included in the 2-factor. Then, we can put the organic 6-cycles introduced by expanding each $S'_3$ of this type in one-to-one correspondence with a larger cycle containing the protected edges of the same type shown in Fig. 36. These edges are labeled “P” in Fig. 36.

3.5. Expanding $D_1$ gadgets

Appendix H in the Appendix documents, in detail for the two possible cases, the process of winding the 2-factor through a $D_1$ after the algorithm has expanded a $D'_1$, the $D_1$’s replacement gadget. These expansions do not produce organic 6-cycles, but they can introduce cycles consisting of two parallel edges, as shown in Figs. 37 and 38. This example is found in Figs. 171 and 172 in Appendix H.

Recall that the COMPRESS algorithm compresses $D_1$’s immediately if present. Also, the original graph is simple, so any $D'_1$ in the compressed graph is the result of replacing a configuration that is not a $D_1$, possibly followed by a series of $D_1$ compressions. Since the original graph is simple, we know that any 2-cycle revealed by after expanding a $D'_1$ (shown in Figs. 37–38) is not organic. Then this expansion operation cannot introduce an organic cycle into the 2-factor. Similarly, we argue in Appendix H (Figs. 169–170) that the other $D_1$ expansion operation does not introduce organic cycles. Then expanding $D'_1$’s will not introduce any organic cycles and therefore cannot introduce organic 6-cycles into the 2-factor.

3.6. Expanding all other gadgets

In the analysis that follows, we have identified all expansions with the potential to introduce organic 6-cycles cycles into $F_i$ that were not present in $F_{i+1}$. Confirming this fact requires checking all possible ways a 2-factor can pass through each gadget to ensure that all expansions that can introduce organic 6-cycles are properly analyzed. Diagrams documenting every one of these other expansions are shown in Appendices A–G, and a careful examination of these sections confirms that all other expansion operations are not capable of introducing an organic 6-cycle into the 2-factor. These other operations
result either in converting one cycle into one larger cycle, converting one cycle into two large (length of at least 8) cycles, or converting two cycles into one or two large cycles.

3.7. Analyzing the worst case

We call the edges identified in Definition 3.1 “protected” because they form organic paths in our 2-factor, so these paths will remain part of the 2-factor regardless of the future expansion operations performed. In Definition 3.2, we established a correspondence between protected edges and 6-cycles in our 2-factor. This relation allows us to place every 6-cycle in correspondence with some large cycle in our 2-factor, as stated in Definition 3.3. We will use this correspondence to analyze the performance of BIGCYCLE.

In Lemma 3.7 we prove that if a 6-cycle’s protected edges are in another 6-cycle, then the second cycle’s protected edges must be in a cycle of length at least 10. This is helpful to us, as the average length of these three cycles is at least $\frac{22}{3}$ which is greater than 7. In Lemma 3.10, we prove that all 8-cycles have at most one corresponding 6-cycle, so any 8-cycle and its corresponding 6-cycle have an average length of 7. Lemma 3.11 generalizes the previous two Lemmas to show that any long cycle and its corresponding 6-cycles have an average length of at least 7.

Proposition 3.6. If a 6-cycle, $C_1$, has its corresponding protected edges in another 6-cycle, $C_2$, then $C_2$ has 5 corresponding protected edges, and these protected edges are all in a cycle of length at least 8. Furthermore, $C_1$ and $C_2$ must have been introduced into the 2-factor during the expansion of a $H_1'$ and $H_2'$, respectively.

Proof. In Section 3, we show all ways an organic 6-cycle can appear in the final 2-factor. The appendices validate this claim, as all possible expansions are examined in detail, and all expansions not included in Section 3 do not produce organic 6-cycles. Immediately following each of these three special expansion operations, all newly identified protected edges are in cycles of length at least 8.

Then suppose, for the sake of contradiction, that in the final 2-factor, $C_1$ has a protected edge in another 6-cycle, $C_2$, but the lemma does not hold. The operation that brought $C_1$ into the 2-factor $F_{i-1}$ in graph $G_{i-1}$ is one that replaced some gadget in a condensed graph, $G_s$, with a $H_1$, $H_2$, $H_3$, or a small 2-cut as depicted in Figs. 25, 29, 31, 33 and 35. We now claim that such a situation cannot occur if the expansion operation replaced the gadget with a $H_2$, $H_3$, or a small 2-cut.

Consider the $H_2$ case first, as shown in Figs. 29–30. In this case, $C_1$ is the left cycle in Fig. 30 and has five protected edges (the path from $A_2$ to $B_2$). Furthermore, after this expansion the path from $A_2$ to $B_2$ in Fig. 30 is entirely organic, meaning none of the nodes or edges (including $A_2$ and $B_2$) are part of gadgets. The only way $C_1$’s protected edges could end up in a 6-cycle in the final 2-factor is if another special expansion operation split $C_2$ into two cycles. For this to happen, however, either two super-vertices which together form a gadget must be a distance of exactly 5 edges away from each other in a cycle or two super-edges must be directly next to each other in a cycle of the 2-factor. However, in this case, $C_1$’s protected edges form an organic path of length 5, which prevents this from occurring, as the two closest possible super-vertices would be $A_1$ and $B_1$, which are separated by 7 edges and the two closest possible super-edges would be $(A_2, B_1)$ and $(A_1, B_2)$, which are separated by 5 edges.

Similarly, consider if the expansion operation that first introduced $C_1$ into the 2-factor replaced a gadget with a $H_3$ or a small 2-cut of the type shown in Fig. 36. The same reasoning as in the previous paragraph applies here, too, as $C_1$ would have at least six protected edges appearing consecutively in a $H_3$ or small 2-cut in these cases. Any corresponding super-vertices or super-edges in this cycle are separated by too many organic edges and nodes to split off some of the protected edges into a new 6-cycle.

The only remaining possibility is that $C_1$ was introduced through an expansion operation that replaced a gadget with a $H_1$, as shown in Fig. 26. In this case, the only way these protected edges could be split off into an organic 6-cycle is if nodes $A_1$ and $B_1$ are a gadget. If this gadget was one that replaced a $H_1$, then we see by Remark 1 that the expansion of this gadget cannot
introduce an organic 6-cycle into the 2-factor. Then, $A_1$ and $B_1$ must have been an $H'_1$. Thus, $C_2$ was necessarily introduced into the 2-factor through the expansion of an $H'_2$, as depicted in Figs. 29–30. Then, $C_2$ has 5 corresponding protected edges of its own. We conclude that these protected edges will be in a cycle of length at least 8 because we demonstrated earlier in this proof that if protected edges are identified from expanding an $H'_2$, these protected edges cannot be part of a 6-cycle in the final 2-factor. The final 2-factor does not contain any odd cycles (due to bipartiteness of the original graph) and contains no 4-cycles so we conclude that $C_2$’s protected edges are part of a cycle of length at least 8, proving the lemma. □

Lemma 3.7. If a 6-cycle, $C_1$, has its corresponding protected edges in another 6-cycle, $C_2$, then $C_2$ has 5 corresponding protected edges. These protected edges are all in a cycle which either contains no other protected edges or is of length at least 10. Furthermore, $C_1$ and $C_2$ must have been introduced into the 2-factor during the expansion of an $H'_1$ and $H'_2$, respectively.

Proof. By Proposition 3.6, we know that $C_2$ has 5 protected edges in a third cycle, $C_3$, of length at least 8. Suppose then, for the sake of contradiction, that this lemma is violated. Then, $C_3$ contains $C_1$’s 5 protected edges, some set of other protected edges, and has length less than 10. By Proposition 3.6 and the fact that the graph is bipartite, we conclude that $C_1$ must have length 8. Protected edges from the same expansion cannot get separated from each other, as there are no super-vertices or super-edges along a path of protected edges, and protected edges come in sets of 3, 4, and 5 edges, depending on the expansion operation. $C_3$ contains the five protected edges from $C_2$ as well as another set of protected edges, so this additional set of protected edges must be a set of three edges. This is only possible if the additional set of protected edges were introduced by expanding an $H'_1$, under the circumstances depicted in Figs. 25–26. Such an expansion would require the nodes corresponding to $A_2$ and $B_2$ in Fig. 30 to be a gadget. However, by Proposition 3.6, $A_2$ and $B_2$ must be organic, as they are part of the $H'_1$ whose expansion introduced $C_1$ into the 2-factor and $C_1$’s protected edges into $C_3$. Since $A_2$ and $B_2$ are organic, they cannot be an $H'_1$, proving that $C_3$ contains no protected edges other than $C_2$’s protected edges. This contradicts our assumption, proving the lemma. □

Before we prove a lower bound on average cycle length, we need an additional lemma regarding 8-cycles in the final 2-factor. The following propositions are needed for the upcoming proof of Lemma 3.10.

Proposition 3.8. Let $P_1$ and $P_2$ be the sets of protected edges corresponding to two 6-cycles, $C_1$ and $C_2$, respectively. Then $P_1 \cap P_2 = \emptyset$ and $V(P_1) \cap V(P_2) = \emptyset$.

Proof. Suppose $P_1 \cap P_2 \neq \emptyset$, for the sake of contradiction. Then there is some edge $e$ such that $e \in P_1$ and $e \in P_2$. All four expansion operations (described in detail in Sections 3.1–3.4) that introduce organic 6-cycles and identify protected edges are such that there is some integer $i$ where $e \notin E_{i+1}$ but $e \in E_i$. Then, the expansion operation from $G_{i+1}$ to $G_i$ will introduce both $C_1$ and $C_2$ to 2-factor $F_1$ as organic 6-cycles. This is not possible, as the expansion operations which introduce organic 6-cycles all introduce exactly one 6-cycle into the 2-factor, proving the first claim. Suppose $V(P_1) \cap V(P_2) \neq \emptyset$, for the sake of contradiction. Then there is some node $v$ such that $v \in V(P_1)$ and $v \in V(P_2)$. All four expansion operations (described in detail in Sections 3.1–3.4) that introduce organic 6-cycles and identify nodes that are endpoints of protected edges are such that there is some integer $i$ where $v \notin V_{i+1}$ but $v \in V_i$. Then, the expansion operation from $G_{i+1}$ to $G_i$ will introduce both $C_1$ and $C_2$ to 2-factor $F_1$ as organic 6-cycles. This is not possible, as the expansion operations which introduce organic 6-cycles all introduce exactly one 6-cycle into the 2-factor, proving the lemma. □

Proposition 3.9. Suppose there is a cycle $C$ in a preliminary 2-factor $F_1$ which is the union of four edge disjoint paths: $P_1, X, P_2, Y$. Furthermore, suppose that $P_1$ and $P_2$ are organic, $P_1$ shares its endpoints with endpoints of $X$ and $Y$, and $P_2$ shares its endpoints with the remaining endpoints of $X$ and $Y$. Then the final 2-factor $F$ does not contain a cycle $C'$ which is the union of $P_1, e_1, P_2, e_2$, where $e_1$ and $e_2$ are single edges separating $P_1$ and $P_2$ in $C'$, unless both $X$ and $Y$ are single edges (see Figs. 39 and 40).

Proof. Suppose, for the sake of contradiction, that $F$ contained such a cycle $C'$ where $C \neq C'$. The algorithm does not perform any expansion operations that combine two organic paths in separate cycles into a single cycle connected by two single edges. Then, none of the final $i$ expansions the algorithm performs can separate $P_1$ and $P_2$ into two different cycles. $|C| > |C'|$, so, for $F$ to contain $C'$, one of these final $i$ expansions must have shortened one of the paths connecting $P_1$ to $P_2$ to just a single edge while keeping both $P_1$ and $P_2$ in the same segment. Without loss of generality we assume that $X$ has
length at least 2. The algorithm does not perform any expansion operations that result in a new cycle containing both $P_1$ and $P_2$ such that the endpoints of $P_1$ and $P_2$ that were previously incident on $X$ are now connected by a single edge. Then there is no sequence of expansion operations that could result in $C$ being contained in $F$, contradicting our assumption. 

**Lemma 3.10.** Every cycle $C$ of length 8 in the final 2-factor $F$ has at most one corresponding 6-cycle.

**Proof.** Suppose, for the sake of contradiction, that there exists an 8-cycle $C$ in $F$ which has $k$ corresponding 6-cycles, $C_1, C_2, \ldots, C_k$, where $k \geq 2$. By Definition 6, for $1 \leq i \leq k$, $C_i$ has protected edges, which are located either in $C$ or in $C_j$ for some $j \neq i, 1 \leq j \leq k$.

First, let us consider when there are values $i, j$ such that $C_i$’s protected edges are in $C_j$. $C$ is of length 8, so by **Lemma 3.7**, it must be the case that $k = 2$ and without loss of generality, $i = 1$ and $j = 2$. Additionally, we know due to **Proposition 3.6** that $C_1$ was introduced through the expansion of a $H'_2$ and $C_2$ was introduced through the expansion of a $H'_2$.

If the expansion of the $H'_2$ that introduced $C_2$ into a preliminary 2-factor $F$, also introduced $C$ into $F$, then $C$ could not be an 8-cycle, which is a contradiction. To see this, see Fig. 30, which describes this class of expansion operation. If $C$ were an 8-cycle, then the dashed path connecting nodes $A_1$ and $B_1$ in this figure is of length 1, but this would mean that the algorithm compressed a $H_2$ at a time when there was a square present in graph, which is also a contradiction unless this edge between $A_1$ and $B_1$ is an $S'_1$. In this case this is still a contradiction because the cycle $C$ is in the final 2-factor and therefore must be organic. Then, the remaining possibility is that immediately following the expansion of the $H'_2$ that introduced $C_2$ into $F$, the cycle’s protected edges were in a non-organic cycle $C'$, of the form shown on the right side of Fig. 30. A later expansion operation must introduce $C$, resulting in the nodes corresponding to $A_2$ and $B_2$ in Fig. 30 being connected by a path of length 3. This expansion cannot be one where a square is expanded because none of these operations shorten the length of the cycles involved. Then any other potential expansion would contradict the algorithm, as a path of length 3 from $A_2$ to $B_2$ would form a square, and the algorithm would have previously contracted a $H'_1$ for some $i \geq 2$ when a square with no $S'_1$s is present in the graph. We know that none of this square’s edges are $S'_1$s because the edge between $A_2$ and $B_2$ is in an organic $H_2$ and the other edges are part of $C$, an organic cycle.

We have now ruled out the possibility of cycle $C_i$ having protected edges in a cycle other than $C$. By **Proposition 3.8**, we know that the protected edges of two cycles $C_i$ and $C_j$ are disjoint. Then, we can easily see that $k \leq 3$. Each 6-cycle has at least 3 protected edges in $C$, and $C$ is an 8-cycle, so if $k \geq 3$, then either $C$ would need to have more than 8 edges or the 6-cycles would need to share protected edges, which is not possible.

We must now also show that we obtain a contradiction when $k = 2$. By **Proposition 3.8**, we know that any two sets of protected edges do not share any vertices. Then if $C$ contains two sets of protected edges, they must be separated by at least one edge on each side of the cycle. Then, $8 = |C| \geq |P_1| + |P_2| + 2$, where $P_1$ and $P_2$ are the protected edges of $C_1$ and $C_2$, respectively. $P_i \geq 3$, and if either $P_1$ or $P_2$ is at least 4 then $|P_1| + |P_2| + 2 \geq 9$, so the only possibility we need to consider is when $|P_1| = |P_2| = 3$. This would require that both 6-cycles are introduced from expanding $H'_1$s. All other possibilities would result in $C$ containing more than 8 protected edges, which is not possible. We now demonstrate that this case results in a contradiction.

We now know that both 6-cycles are introduced through the expansion of two $H'_1$s, as all other cases result in a contradiction. The specific expansion of this type that can introduce organic 6-cycles is described in detail in Section 3.1 and shown in Figs. 25–26. If $P_1$ and $P_2$ are ever contained in different cycles of a preliminary 2-factor $F$, then some expansion operation will eventually bring these two sets of protected edges into the same cycle. However, observe that the algorithm does not perform any expansion operations that combine two organic paths in separate cycles into a single cycle, where the two paths share an endnode or are separated by a single edge on both sides. Then $P_1$ and $P_2$ will necessarily be separated by at least two edges on one side and at least one edge on the other side. By **Proposition 3.9**, no future expansions could result in $P_1$ and $P_2$ being contained in a single 8-cycle. This would contradict the assumptions that $C$ is an 8-cycle and contains two sets of protected edges introduced through the expansion of two $H'_1$s.

The other possibility is that the expansion of the second $H'_1$ introduces $P_2$ directly into a cycle that contains $P_1$. By **Remark 1**, we know that the nodes corresponding to $A_1$ and $B_1$ shown Fig. 26 cannot be super-vertices of an $H'_1$ whose expansion introduces an organic 6-cycle whose protected edges are in an 8-cycle that contains another set of protected edges. If $A_1$ or $B_1$ or either edge $(A_2, B_1)$ or $(B_2, A_1)$ are non-organic and get expanded before the expansion that introduces the 6-cycle into $F$, then these expansions will replace nodes $A_1$ and $B_1$ with new nodes that neighbor $B_2$ and $A_2$ in the 2-factor. By the same argument from **Remark 1**, these nodes cannot be super-vertices of an $H'_1$ whose expansion introduces $P_2$ into an organic 8-cycle containing $P_1$. If this were to happen then the local improvement step of BIGCYCLE would intervene
and turn the two 6-cycles and one 8-cycle into two 10-cycles, and neither \( P_1 \) or \( P_2 \) would be required to account for 6-cycles. Then, there must be at least 7 edges between any two super-vertices of a \( H'_1 \) whose expansion introduces an organic 6-cycle into \( F_2 \) whose protected edges, \( P_2 \), are in the same 8-cycle as \( P_1 \). Therefore, after the expansion introducing the second 6-cycle, the cycle containing both sets of protected edges in \( F_2 \) will have at least 10 edges, the 7 edges in between the two super-vertices that replaced the second \( H_1 \) and the second 6-cycle’s set of 3 protected edges. By Proposition 3.9, no future expansions can result in both \( P_1 \) and \( P_2 \) being contained in a single 8-cycle. This contradicts the assumptions that \( C \) is an 8-cycle and contains two sets of protected edges introduced through the expansion of two \( H'_1 \)’s.

We have proved that all cases that could result in the existence of an 8-cycle in the final 2-factor \( F \) that has two or more corresponding 6-cycles leads to a contradiction, proving the lemma.  

We are now prepared to prove the next lemma, regarding average cycle length of a large cycle and its set of corresponding 6-cycles:

**Lemma 3.11.** For any cycle \( C \) in the final 2-factors \( F \) of length \( l \) such that \( l \geq 8 \) and its set of corresponding 6-cycles, the average length of this set of cycles is at least 7.

**Proof.** First, consider the simple case where \( C \) has no corresponding 6-cycles. The set of cycles we are considering in this case is just a single cycle of length \( l \geq 8 \). By Proposition 3.8, this cycle has at least 7 edges. Therefore, the only cycle in the set of cycles has length at least 7.

Now, consider the case when all of the corresponding 6-cycles have their protected edges contained in the large cycle \( C \). If \( l \geq 7 \), then the average cycle length among cycle \( C \) and its corresponding 6-cycles is at least

\[
\left\lfloor \frac{l}{2} \right\rfloor \times 6 + l \geq 7.
\]

If \( l = 8 \) then by Lemma 3.10, \( C \) has at most one corresponding 6-cycle, so the average length of \( C \) and its corresponding 6-cycle is also 7. We must consider the case when at least one corresponding 6-cycle, \( C_1 \), has its protected edges in another 6-cycle, \( C_2 \). If \( l = 8 \) and \( C \) contains \( C_2 \)’s protected edges then \( C \) has at least two corresponding 6-cycles, \( C_1 \) and \( C_2 \), contradicting Lemma 3.10.

Next, consider if \( l = 10 \) and \( C \) contains \( C_2 \)’s 5 protected edges in the final 2-factor. \( C \) cannot contain another set of 5 protected edges, due to Proposition 3.8, because this would require these protected edges to share a node with \( C_2 \)’s protected edges. Then, in addition to \( C_1 \) and \( C_2 \), \( C \) can have at most one additional corresponding 6-cycle, otherwise \( C \) would contain more than 10 protected edges. In this case, \( C \) has at most 3 corresponding 6-cycles, so the average length of \( C \) and its corresponding 6-cycles is at most \( \frac{10 + 3 \times 6}{6} = 7 \).

The only remaining case is when \( l \geq 12 \) and at least one corresponding 6-cycle, \( C_1 \), has its protected edges contained in another 6-cycle, \( C_2 \). By Lemma 3.7, each 6-cycle has at least 3 protected edges in \( C \) or its protected edges are in another 6-cycle whose 5 protected edges are in \( C \). So, if a corresponding 6-cycle’s protected edges are not in \( C \), then there is another 6-cycle corresponding to \( C \) for which these two 6-cycles contribute 5 protected edges to \( C \). Then, each 6-cycle on average contributes at least \( \frac{7}{2} \) protected edges to \( C \), so there are at most \( \frac{2l}{3} \) 6-cycles corresponding to \( C \). Then, the average cycle length among cycle \( C \) and its corresponding 6-cycles is at least

\[
\left\lfloor \frac{2l}{3} \right\rfloor \times 6 + l \geq 7 \quad \text{(because } l \geq 12 \text{).}
\]

In all possible cases, \( C \) and its corresponding 6-cycles have average length of at least 7.  

3.8. Main theorems

**Theorem 3.12.** Given a cubic bipartite graph \( G \) with \( n > 6 \) vertices, there is an algorithm with running time \( O(n^{5/2}) \) that computes a 2-factor with at most \( \frac{n}{5} \) cycles.

**Proof.** It follows directly from Lemma 3.11 that the average cycle length in the 2-factor produced by BIGCYCLE is at least 7. Then, it must be the case that BIGCYCLE produces a 2-factor with at most \( \frac{n}{7} \) cycles.

The BIGCYCLE algorithm performs \( O(n) \) contractions and expansions. In between each contraction, the algorithm will search for other subgraphs to contract and will compute a 2-factor in the current graph. Classical algorithms can compute 2-factors with \( O(n^{5/2}) \) operations in the worse case [11], so the contraction phase of the algorithm runs in \( O(n^{5/2}) \) time. The expansion phase of the algorithm takes \( O(n) \) time in the worst case, since there are at most \( O(n) \) expansions and each one is performed in constant time. Then, BIGCYCLE finds a 2-factor with at most \( \frac{n}{7} \) cycles in \( O(n^{5/2}) \).  

We can now restate our main theorem from Section 1.1:

**Theorem 1.1.** Given a cubic bipartite connected graph \( G \) with \( n \) vertices, there is a polynomial time algorithm that computes a spanning Eulerian multigraph \( H \) in \( G \) with at most \( \frac{n}{7} \) edges.
Proof. Theorem 3.12 proves that the COMPRESS and EXPAND phases of BIGCYCLE produce a 2-factor with at most $\frac{8}{7}$ cycles for the required class of graphs. Proposition 1.3 demonstrates that the DOUBLETREE phase in BIGCYCLE successfully extends the 2-factor into a spanning Eulerian multigraph with at most $\frac{5}{7}n - 2$ edges.

From Theorem 3.12 that we can compute the required 2-factor in $O(n^{\frac{5}{2}})$ time. Once we have done this, we can compute the doubled spanning tree in $O(n)$ time as well, as the graph has $O(n)$ edges. The total running time of the algorithm is $O(n^{\frac{5}{2}})$ in the worst case.

4. An extension to $k$-regular bipartite graphs

In this section, we demonstrate how the BIGCYCLE algorithm can be used as a subroutine to produce an improved approximation algorithm for $k$-regular bipartite graphs. The main idea in this algorithm is that $k$-regular bipartite graphs contain cubic subgraphs on which we can run BIGCYCLE to obtain solutions to the cubic subgraphs, which will also be
Fig. 45. A cycle of length $x + y + 4$ that passes through a gadget that replaced a square.

Fig. 46. The cycle from the previous figure, after expanding the gadget, is now of length $x + y + 6$.

solutions to the original $k$-regular bipartite graphs. If the cubic subgraph we find is composed entirely of connected components of size 8 and larger, then we will get a solution with at most $\frac{9}{2}n - 2$ edges. However, if some of the components are of size 6 (in a cubic bipartite graph these will be $K_{3,3}$s), then the 2-factor we compute may have between $\frac{n}{k}$ and $\frac{k}{2}$ cycles, which gives us a solution of size $x$ where $\frac{9}{2}n - 2 \leq x \leq \frac{k}{2}n - 2$. Algorithm 2 provides the pseudo-code for selecting cubic subgraph from a $k$-regular bipartite graph containing a small number of $K_{3,3}$s. In the analysis that follows, we will bound the number of $K_{3,3}$s in the cubic subgraph computed by Algorithm 2, allowing us to prove a specific approximation factor. The subroutine FindPerfectMatching, used in Algorithm 2, takes a graph as input and returns a perfect matching in the graph. The subroutine CountK33, also used in Algorithm 2, takes a graph as input and returns the number of connected components of the graph that are $K_{3,3}$s.

Algorithm 2: An algorithm to find a cubic subgraph from a $k$-regular bipartite graph: CUBIC

Input: A connected, undirected, unweighted, $k$-regular, bipartite graph, $G_0 = (V, E)$

For $i = 1$ to $k$:
- $M_i \leftarrow \text{FindPerfectMatching}(G_{i-1})$
- $G_i \leftarrow G_{i-1} \setminus M_i$
- If $i = 3$:
  - $G_{\text{cubic}} \leftarrow (V, M_1 \cup M_2 \cup M_3)$
- If $i > 3$:
  - $G_{\text{temp}} \leftarrow (V, M_1 \cup M_2 \cup M_i)$
  - If CountK33($G_{\text{temp}}$) < CountK33($G_{\text{cubic}}$):
    - $G_{\text{cubic}} \leftarrow G_{\text{temp}}$

End Loop

Return $G_{\text{cubic}}$

Lemma 4.1. For any $k$-regular bipartite graph where $k \geq 4$, $G$, at most $\frac{n}{6(k-2)} K_{3,3}$s are contained in CUBIC($G$), the cubic bipartite graph output by Algorithm 2.

Proof. Consider an arbitrary $k$-regular bipartite graph $G$. Note that the nodes of any 6-cycle in $(V, M_1 \cup M_2)$ will form a $K_{3,3}$ in $(V, M_1 \cup M_2 \cup M_i)$ for at most 1 value of $i$ because the $k$ matchings $M_1, \ldots, M_k$ are edge-disjoint. There are at most $\frac{n}{6}$ 6-cycles in $M_1 \cup M_2$. In the worst case, the nodes of each of these 6-cycles can form a $K_{3,3}$ in $(V, M_1 \cup M_2 \cup M_i)$ for exactly one value of $i$ where $3 \leq i \leq k$. Then, by the pigeonhole principle, there is some value $i$ where $3 \leq i \leq k$ where $(V, M_1 \cup M_2 \cup M_i)$ contains at most $\frac{n}{k-2} = \frac{n}{6(k-2)} K_{3,3}$. Algorithm 2 finds $M_i$ where $(V, M_1 \cup M_2 \cup M_i)$ contains the fewest $K_{3,3}$s, so the algorithm will necessarily output three edge-disjoint matchings with at most $\frac{n}{6(k-2)} K_{3,3}$, proving the lemma. \(\Box\)
Theorem 4.2. Given a $k$-regular bipartite $G$ with $n$ vertices where $k \geq 4$, there is a polynomial time algorithm that computes a spanning Eulerian multigraph $H$ in $G$ with at most \((\frac{k}{7} + \frac{1}{21(k-2)})n - 2\) edges.

Proof. First, we will find a cubic subgraph of $G$, $G_{cubic}$, by running Algorithm 2 on $G$. In each component of $G_{cubic}$ that is a $K_{3,3}$ we will find a 6-cycle covering these nodes. This can be done in constant time by taking any walk through this component that does not visit a node twice as long as this is possible, then returning to the first node. In every other connected component, run the Contract and Expand phases of the BIGCYCLE algorithm, which will find a 2-factor over this component containing at most $\frac{n_i}{7}$ cycles, where $n_i$ is the number of nodes in the connected component. The upper bound of $\frac{n_i}{7}$ cycles is proven by Theorem 3.12. By Lemma 4.1, there are $x_1$ nodes in $K_{3,3}$s within $G_{cubic}$, where $x_1 \leq \frac{n}{k-2}$. These nodes are covered by $\frac{3x_1}{6}$ cycles. Then, there $x_2$ nodes in the remaining components and $x_1 + x_2 = n$. These $x_2$ nodes are covered by at most $\frac{x_2}{7}$ cycles. Then, the overall 2-factor of $G$ has at most $\frac{x_1}{6} + \frac{x_2}{7}$ cycles. The following calculations compute an upper
**Fig. 51.** The super-edge is not included in the 2-factor.

**Fig. 52.** A cycle of length 6 passes through the square after the super-edge in the previous figure is expanded. If this 6-cycle is organic but at least one of the two edges of the square that are not in the cycle are non-organic then we exchange the two vertical edges in the central square for the two horizontal edges. This exchange helps limit the number of organic 6-cycles in the 2-factor. The impact of these 6-cycles on the algorithm’s result is analyzed in Sections 3.4 and 3.7.

**Fig. 53.** A cycle of length $x + 1$ that passes through a gadget that replaced a square.

Bound on these cycles in terms of $n$:

\[
\frac{x_1}{6} + \frac{x_2}{7} = \frac{7x_1 + 6x_2}{42} \\
= \frac{6n + x_1}{42} \\
= \frac{n}{7} + \frac{x_1}{42} \\
\leq \frac{n}{7} + \frac{n}{42(k - 2)}.
\]
By Proposition 1.3, this 2-factor can be extended into a spanning Eulerian multigraph in $G$ with at most $n + 2(\frac{n}{7} + \frac{n}{42(k-2)}) - 1 = (\frac{n}{7} + \frac{1}{2(k-2)})n - 2$ edges, proving the theorem. □

**Acknowledgments**

We would like to thank Satoru Iwata and Alantha Newman for useful discussions during this project.
Fig. 58. The cycles from the previous figure, after expanding the gadget, now form a single cycle of length \( x + y + 8 \).

Fig. 59. Two cycles of lengths \( x + 2 \) and \( y + 2 \) that pass through a \( H'_1 \).

Fig. 60. The cycles from the previous figure, after expanding the gadget, now form two cycles of lengths \( x + 4 \) and \( y + 4 \), respectively.

Appendix A. Squares

The purpose of the appendices is to demonstrate in detail how the algorithm BIGCYCLE “winds” a 2-factor through the gadgets (4-cycles, \( H_1 \)'s, \( H_2 \)'s, \( H_3 \)'s, \( H_4 \)'s, \( H_5 \)'s, and \( H_6 \)'s) as it expands the condensed graph back to its original state.

A.1. \( S_1 \) expansions: Gadget is covered by three edges of a single cycle

If a gadget that replaced a square is covered by two disjoint cycles of \( F_i \), then the internal edge of the gadget must be included from \( F_i \). Then, \( F_i \) must include either edge 1 or 3 and either edge 2 or 4. However, while there are four orientations to consider, they are all symmetric to each other, so there is only one case to consider. In this case, we start with a cycle of length \( x + 3 \) in \( F_i \) and are returned a single cycle of length \( x + 5 \) in \( F_{i-1} \). \( x + 3 \geq 6 \), so \( x + 5 \geq 8 \), meaning this class of expansions cannot introduce an organic 6-cycle into the 2-factor (see Figs. 41 and 42).

A.2. \( S_1 \) expansions: Gadget is covered by two cycles

If a gadget that replaced a square is covered by two disjoint cycles of \( F_i \), then the internal edge of the gadget must be excluded from \( F_i \). Then, there is only one possible orientation in which \( F_i \) could cover the nodes of this gadget. In this case, we start with cycles of lengths \( x + 2 \) and \( y + 2 \) in \( F_i \) and are returned a single cycle of length \( x + y + 6 \) in \( F_{i-1} \). If \( x + y \geq 2 \) then the expansion returns a cycle of length at least 8. If \( x + y < 2 \) then either \( x = 0 \) or \( y = 0 \) meaning that either edges 1
**Fig. 61.** A cycle of length $x + y + 4$ that passes through a $H'_1$.

**Fig. 62.** The cycle from the previous figure, after expanding the gadget, is now of length $x + y + 8$.

**Fig. 63.** A cycle of length $x + y + 4$ that passes through a $H'_1$.

**Fig. 64.** The cycle from the previous figure, after expanding the gadget, is now two cycles of lengths $x + 3$ and $y + 5$, respectively. This expansion can produce an organic 6-cycle if $x = 3$ and the cycle is organic. The impact of these 6-cycles on the algorithm’s result is analyzed in Sections 3.1 and 3.7.

and 3 or 2 and 4 (or both) are parallel edges. This will only occur if at least one of each pair of parallel edges is an $S'_3$. Then this expansion may produce a 6-cycle, but this 6-cycle is not organic as it includes an $S'_3$ (see Figs. 43 and 44).
A cycle of length $x + y + 4$ that passes through a $H'_1$.

The cycle from the previous figure, after expanding the gadget, is now two cycles, of lengths $y + 3$ and $x + 5$, respectively. This expansion can produce an organic 6-cycle if $y = 3$ and the cycle is organic. The impact of these 6-cycles on the algorithm’s result is analyzed in Sections 3.1 and 3.7.

A cycle of length $x + y + 4$ that passes through a $H'_1$.

The cycle from the previous figure, after expanding the gadget, is now of length $x + y + 8$.

A.3. $S_1$ expansions: Gadget is covered by four edges of a single cycle

If a gadget that replaced a square is covered by four edges of a single cycle of $F_i$, then there are two orientations in which $F_i$ could pass through these edges. However, these two cases are symmetric, so there is only one case to consider. In this case, we start with a cycle of length $x + y + 4$ in $F_i$ and are returned a single cycle of length $x + y + 6$ in $F_{i-1}$. $x \neq 0$ because if edges
1 and 2 are not the same edge then the graph would not be bipartite. Edges 1 and 2 cannot be the same edge otherwise \( G_{i-1} \) contains a \( D_1 \) which would have been compressed before this \( S_1 \). Therefore, \( x \geq 1 \). The same argument shows that \( y \geq 1 \).

Then \( x + y + 6 \geq 8 \), meaning this class of expansions cannot introduce an organic 6-cycle into the 2-factor (see Figs. 45 and 46).

**A.4. \( S_2 \) expansions: A super-vertex replaces a square**

A super-vertex that replaced a square is necessarily covered by \( F_i \). Then, there are three ways we can select the edge in each of the super-vertices to exclude from \( F_i \). However, the cases where edges 2 and 3 are excluded from \( F_i \) are symmetric so
we will examine only the second of these cases. Then, there are only two case to consider. In all cases, we start with a cycle of length $x + 2$ in $F_i$ and are returned a single cycle of length $x + 6$ in $F_{i-1}$. If $x = 0$ because edges the two edges in the 2-factor that exit the $S'_2$ are parallel then at least one of these two edges is an $S'_3$ and the 6-cycle introduced by this expansion will not be organic. Otherwise, $x + 2 \geq 4$, so $x + 6 > 8$, meaning this class of expansions cannot introduce an organic 6-cycle into the 2-factor (see Figs. 47–50).

A.5. $S_3$ expansions: Super-edge replaces a square

See Figs. 51–54.

Appendix B. $H_1$s

If a replacement operation annihilates (see Definition 2.3) one of the configurations then we have freedom to choose a favorable way to cover the configuration with a cycle when we expand its annihilated gadget. If a replacement operation annihilates an $H_1$ then this configuration was part of a $K_{3,3}$. We know the gadget edges are organic, but the numbered edges
Fig. 77. Two cycles of lengths $x + 2$ and $y + 2$ pass through a $H'_2$.

Fig. 78. The cycles from the previous figure, after expanding the gadget, are now a single cycle of length $x + y + 10$.

Fig. 79. A cycle of length $x + y + 4$ passes through a $H'_3$.

Fig. 80. The cycle from the previous figure, after expanding the gadget, is now two cycles, of lengths $x + 3$ and $y + 7$, respectively. This expansion can produce an organic 6-cycle if $x = 3$ and the cycle is organic. The impact of these 6-cycles on the algorithm’s result is analyzed in Sections 3.2 and 3.7.

can potentially be $S'_3$’s. When we expand this $K_{3,3}$ we cover it with a non-organic 6-cycle that contains at least half of the $S'_3$ edges in the $K_{3,3}$.

B.1. Gadget is covered by two cycles

If a $H'_1$ is covered by two disjoint cycle in $F_i$, then, there are three ways we can select the edge in each of the super-vertices to exclude from $F_i$. However, without loss of generality, we can fix the edge of the first super-vertex to exclude from $F_i$. Then, there are only three cases to consider. In each of these cases, we start with cycles of lengths $x + 2$ and $y + 2$ in $F_i$ and are returned either a single cycle of length $x + y + 8$ or two cycles of lengths $x + 4$ and $y + 4$ in $F_{i-1}$. If the first expansion introduces an organic cycle then $x + 2 \geq 6$ and $y + 2 \geq 6$, so $x + y + 8 > 8$, meaning the first case cannot introduce an
If a $H_i^j$ is covered by a single cycle in $F_i$, then, there are three ways we can select the edge in each of the super-vertices to exclude from $F_i$. However, without loss of generality, we can fix the edge of the first super-vertex to exclude from $F_i$. organic 6-cycle into the 2-factor. Similarly, we see in the second case that if the $(x + 4)$-cycle after expansion is organic then $x + 2 \geq 6$ and so $x + 4 \geq 8$. Similarly, if the $(y + 4)$-cycle is organic then and $y + 4 \geq 8$ (see Figs. 55–60).

**B.2. Gadget is covered by one cycle**

...
Fig. 85. A cycle of length $x + y + 4$ passes through a $H'_2$.

Fig. 86. The cycle from the previous figure, after expanding the gadget, is now two cycles, of lengths $x + 5$ and $y + 5$, respectively. Note that $x$ and $y$ cannot be 1, otherwise this $H_2$ would be part of a $H_3$. This is not possible, otherwise the $H_3$ would have been contracted instead of this $H_2$. It is also possible that edges 1 and 5 and/or edges 2 and 6 could be the same edge. This would mean there is a square in this configuration, so these edges would be $S'_i$'s. If edges 1 and 5 were the same $S'_i$ edge then $x = 0$ and this cycle is a non-organic 4-cycle. Similarly, if edges 2 and 6 are the same $S'_i$ edge then the other cycle is a non-organic 4-cycle.

Fig. 87. A cycle of length $x + y + 4$ passes through a $H'_2$.

Then, in each of these three configurations, we examine the two orientations in which the cycle can pass through the two super-vertices. In all 6 cases, we start with a cycle of lengths $x + y + 4$ in $F_i$ and are returned either a single cycle of length $x + y + 8$, two cycles of lengths $x + 3$ and $y + 5$, or two cycles of lengths $x + 5$ and $y + 3$ in $F_{i-1}$. $x + y + 4 \geq 6$, so $x + y + 8 > 8$, meaning the first case cannot introduce an organic 6-cycle into the 2-factor. In the later two cases the $x + 3$ or $y + 3$ cycle can be an organic 6-cycle, but this is the expansion examined in detail in Sections 3.1 and 3.7. In these later two cases, the short cycle could be a 4-cycle, but such a cycle would not be organic (see Figs. 61–64 and 67–72).

Appendix C. $H_2$'s

If a replacement operation annihilates (see Definition 2.3) an $H_2$ configuration then we have freedom to choose a favorable way to cover the configuration with a cycle when we expand its annihilated gadget. We know the gadget edges are organic,
Fig. 88. The cycle from the previous figure, after expanding the gadget, is now two cycles, of lengths $x + 5$ and $y + 5$, respectively. Note that $x$ and $y$ cannot be 1, otherwise this $H_2$ would be part of a $H_3$. This is not possible, otherwise the $H_2$ would have been contracted instead of this $H_2$. It is also possible that edges 1 and 6 and/or edges 2 and 4 could be the same edge. This would mean there is a square in this configuration, so these edges would be $S_j$'s. If edges 1 and 6 were the same $S_j$ edge then $x = 0$ and this cycle is a non-organic 4-cycle. Similarly, if edges 2 and 4 are the same $S_j$ edge then the other cycle is a non-organic 4-cycle.

Fig. 89. A cycle of length $x + y + 4$ passes through a $H_2'$. 

Fig. 90. The cycle from the previous figure, after expanding the gadget, is now a cycle of length $x + y + 10$.

Fig. 91. None of the super-edges in a $H_j'$ are included in the 2-factor.

but the numbered edges will be $S_j$'s (otherwise a 4-cycle would have been contracted). We also know that the numbered edges are not parallel edges. For any possible arrangement of the numbered edges there is an 8-cycle or two 4-cycles that cover the gadget and includes at least half of the $S_j$'s. To see this, consider Appendix C.2 and see that this section illustrates ways to cover the $H_2$ using any two of the $S_j$ edges. For example, Fig. 83 shows how we could cover the gadget if edges 1
and 5 and edges 2 and 4 were each a single $S'_3$ edge. In all expansions where paths labeled $x$ and $y$ connect numbered edges the $H_2$ is covered by a single cycle or two cycles. These cycles will be non-organic when we expand annihilated $H_2$s so these expansions do not introduce organic 6-cycles.

C.1. Gadget is covered by two cycles

If a $H'_2$ is covered by a two cycles in $F_i$, then, there are three ways we can select the edge in each of the super-vertices to exclude from $F_i$. However, without loss of generality, we can fix the edge of the first super-vertex to exclude from $F_i$. Then, we only have to consider three cases. In each of these cases, we start with two cycles of lengths $x + 2$ and $y + 2$ in $F_i$ and are returned a single cycle of length $x + y + 10$ in $F_{i-1}$. If the $(x + y + 10)$-cycle in $F_{i-1}$ is organic then $x + 2 \geq 6$ and $y + 2 \geq 6$, so $x + y + 10 \geq 8$. This means that none of these expansions can introduce an organic 6-cycle into the 2-factor (see Figs. 73–78).

C.2. Gadget is covered by one cycle

If a $H'_2$ is covered by a single cycle in $F_i$, then, there are three ways we can select the edge in each of the super-vertices to exclude from $F_i$. However, without loss of generality, we can fix the edge of the first super-vertex to exclude from $F_i$. Then, in each of these three configurations, we examine the two orientations in which the cycle can pass through the two super-vertices. In all 6 cases, we start with a cycle of lengths $x + y + 4$ in $F_i$ and are returned a single cycle of length $x + y + 10$, two cycles of lengths $x + 5$ and $y + 5$, or two cycles of lengths $x + 3$ and $y + 7$ in $F_{i-1}$. $x + y + 4 \geq 6$, so $x + y + 10 > 8$, meaning the first case cannot introduce an organic 6-cycle into the 2-factor. In the second case, neither the $x + 5$ and $y + 5$ cycles can be organic 6-cycles, otherwise the $H_2$ the gadget replaced would have been part of an organic $H_3$, which would
have been contracted instead of the $H_2$. In the third case, the $x + 3$ cycle can be an organic 6-cycle (or a non-organic 4-cycle containing a $S'_3$), but this is the expansion examined in detail in Sections 3.2 and 3.7 (see Figs. 81, 82 and 84–90).

**Appendix D. $H_3$s**

If a replacement operation annihilates (see Definition 2.3) an $H_3$ configuration then we have freedom to choose a favorable way to cover the configuration with a cycle when we expand its annihilated gadget. We know the gadget edges are organic, but the numbered edges will be $S'_3$s (otherwise a 4-cycle would have been contracted). We also know that the numbered edges are not parallel edges. For any arrangement of the numbered edges there is a non-organic 10-cycle or two non-organic 6-cycles that cover the gadget and includes at least half of the $S'_3$s. To see this, consider Appendix D.3.2 and see that this section illustrates ways to cover the $H_3$ using any two of the $S'_3$ edges. For example, Fig. 105 shows how we could cover the gadget if edges 1 and 4 and edges 3 and 5 were each a single $S'_3$ edge. All expansions in this section cover the configuration.
Fig. 99. Two cycles of lengths $x+1$ and $y+1$ pass through a $H_3'$.

Fig. 100. The cycles from the previous figure, after expanding the gadget, are now cycles of lengths $x+5$ and $y+7$, respectively. If $x = 1$ then this expansion introduces a 6-cycle. The impact of this potential expansion is analyzed in detail in Sections 3.3 and 3.7.

Fig. 101. Two cycles of lengths $x+1$ and $y+1$ pass through a $H_3'$.

Fig. 102. The cycles from the previous figure, after expanding the gadget, are now cycles of lengths $x+5$ and $y+7$, respectively. If $x = 1$ then this expansion introduces a 6-cycle. The impact of this potential expansion is analyzed in detail in Sections 3.3 and 3.7.

in a single cycle or two cycles. These cycles will be non-organic so when we expand annihilated $H_3$s these expansions do not introduce organic 6-cycles.

**D.1. Zero super-edges are covered by 2-factor**

If none of the super-edges of a $H_3'$ are covered by $F_i$, then expanding this gadget returns a cycle of length 10 (see Figs. 91 and 92).

**D.2. One super-edge is covered by 2-factor**

If exactly one edge of a $H_3$ gadget is covered by a cycle of $F_i$, then, to ensure we examine every case, we examine all three ways we can select the super-edge to include in $F_i$. In all three cases, we start with a cycle of length $x + 1$ in $F_i$ and are
**Fig. 103.** Two cycles of lengths $x + 1$ and $y + 1$ pass through a $H'_3$.

**Fig. 104.** The cycles from the previous figure, after expanding the gadget, are now a single cycle of length $x + y + 12$.

**Fig. 105.** A cycle of length $x + y + 2$ passes through a $H'_3$.

**Fig. 106.** The cycle from the previous figure, after expanding the gadget, is now a cycle of length $x + y + 12$.

**Fig. 107.** A cycle of length $x + y + 2$ passes through a $H'_3$. 
Fig. 108. The cycle from the previous figure, after expanding the gadget, is now a cycle of length $x + y + 12$.

Fig. 109. A cycle of length $x + y + 2$ passes through a $H'_3$.

Fig. 110. The cycle from the previous figure, after expanding the gadget, is now a cycle of length $x + y + 12$.

returned either a single cycle of length $x + 11$ in $F_{i-1}$, $x + 1 \geq 2$, so $x + 11 \geq 8$, meaning the resulting cycle in $F_{i-1}$ cannot be a cycle of length 6 (see Figs. 93–98).

D.3. Two super-edges are covered by 2-factor

D.3.1. Two cycles pass through gadget

If two edges of a $H'_3$ are covered by two disjoint cycles in $F_i$, then, to ensure we examine every case, we examine all three ways we can select the super-edge to exclude from $F_i$. In all three cases, we start with cycles of lengths $x + 1$ and $y + 1$ in $F_i$ and are returned either a single cycle of length $x + y + 12$ or two cycles of lengths $x + 5$ and $y + 7$ in $F_{i-1}$. The first case is not problematic, but if $x = 1$, which could happen as part of a $D_1$, then the case where one cycles is length $x + 5$ and the other is length $y + 7$ introduces a 6-cycle. We analyze this case in detail in Sections 3.3 and 3.7 (see Figs. 103 and 104).

D.3.2. One cycle passes through gadget

If two edges of a $H'_3$ are covered by a single cycle in $F_i$, then, to ensure we examine every case, we examine all three ways we can select the super-edge to exclude from $F_i$ and then within each of these arrangements, we examine both orientations in which the two super-edges in the same cycle can be connected. In all six cases, we start with a cycle of length $x + y + 2$ in $F_i$ and are returned a single cycle of length $x + y + 12$ in $F_{i-1}$, except for one exception, which is analyzed in detail in
Sections 3.3 and 3.7 (see Figs. 106–110), $x$ and $y$ are non-negative, so $x + y + 12 > 8$, meaning the resulting cycle in $F_{i-1}$ cannot be a cycle of length 6, except for in the special case we have identified (see Figs. 111–114).

D.4. Three super-edges are covered by 2-factor

D.4.1. Three cycles pass through gadget

If a $H'_3$ is covered by three cycles, one passing through each edge of the gadget then there is only a single case to consider. When the gadget is expanded in this case, the three cycles of lengths $x + 1$, $y + 1$, and $z + 1$ are merged into a single cycle.
Fig. 115. A cycle of length $x + y + 2$ passes through a $H'_3$.

Fig. 116. The cycle from the previous figure, after expanding the gadget, is now two cycles of lengths $x + 6$ and $y + 6$, respectively. Both $x$ and $y$ cannot have length 0, otherwise this $H_3$ would be part of a $H_4$. This expansion can produce an organic 6-cycle if either $x$ or $y$ is a 0-length path and the cycle is organic. The impact of these 6-cycles on the algorithm’s result is analyzed in Sections 3.3 and 3.7.

Fig. 117. Three cycles of lengths $x + 1, y + 1$, and $z + 1$ pass through a $H'_3$.

Fig. 118. The cycles from the previous figure, after expanding the gadget, are now a single cycle of length $x + y + z + 13$.

Fig. 119. Two cycles of lengths $x + y + 2$ and $z + 1$ pass through a $H'_3$. 
Fig. 120. The cycles from the previous figure, after expanding the gadget, are now a single cycle of length $x + y + z + 13$.

Fig. 121. Two cycles of lengths $x + y + 2$ and $z + 1$ pass through a $H'_3$.

Fig. 122. The cycles from the previous figure, after expanding the gadget, are now a single cycle of length $x + y + z + 13$.

Fig. 123. Two cycles of lengths $x + y + 2$ and $z + 1$ pass through a $H'_3$.

of length greater than $x + y + z + 13 > 8$. This means that the expansion documented in this section cannot introduce an organic 6-cycle (see Figs. 117 and 118).
D.4.2. Two cycles pass through gadget

If a $H'_3$ is covered by two disjoint cycles in $F_i$, then two of the super-edges must be part of the same cycle, with the third super-edge in a different cycle. To ensure we examine every case, we examine all three ways we can select the super-edge to appear in a separate cycle and then within each of these arrangements, we examine both orientations in which the two super-edges in the same cycle can be connected (see Figs. 119–123). In all cases we start with cycles of lengths $x + y + 2$ and $z + 1$ in $F_i$ and are returned a single cycle of length $x + y + z + 13$ in $F_{i-1}$. $x$, $y$, and $z$ are non-negative, so $x + y + z + 13 > 8$, meaning the resulting cycle in $F_{i-1}$ cannot be a cycle of length 6 (see Figs. 124–130).
D.4.3. One cycle passes through gadget

If a $H'_3$ is covered by a single cycle in $F_i$, then without loss of generality (see Figs. 131 and 132), we can assume the cycle passes through the $(4, 3)$ super-edge first (see Figs. 133 and 134). To ensure we examine every case, we examine all four sides of the two remaining super-edges the 2-factor could enter after exiting edge 4 (see Figs. 135–138). Then, after exiting the other side of this second super-edge, we consider the two orientations in which the 2-factor could enter the third super-
Fig. 132. The cycle from the previous figure, after expanding the gadget, is now a cycle of length $x + y + z + 13$.

Fig. 133. A cycle of length $x + y + z + 3$ passes through a $H_3'$.

Fig. 134. The cycle from the previous figure, after expanding the gadget, is now a cycle of length $x + y + z + 13$.

edge. In total, this gives us eight cases to consider. In all cases we start with a cycle of lengths $x + y + z + 3$ in $F_i$ and are returned a single cycle of length $x + y + z + 13$ in $F_{i-1}$, $x$, $y$, and $z$ and non-negative, so $x + y + z + 13 > 8$, meaning the resulting cycle in $F_{i-1}$ cannot be a cycle of length 6 (see Figs. 139–146).

Appendix E. $H_4$s

If a replacement operation annihilates (see Definition 2.3) an $H_4$ configuration then we have freedom to choose a favorable way to cover the configuration with a cycle when we expand its annihilated gadget. We know the gadget edges are organic, but the numbered edges may be $S_j$'s. We also know that the numbered edges are not parallel edges. For any arrangement of the numbered edges there is a non-organic 12-cycle that covers the gadget and includes at least half of the $S_j$'s. To see this, consider Appendix E.2 and see that this section illustrates ways to cover the $H_4$ using any two of the $S_j$ edges. For example, Fig. 149 shows how we could cover the gadget if edges 1 and 2 was a single $S_j$ edge. All expansions in this section cover the configuration in a single cycle. These cycles will be non-organic and greater than 6 edges so when we expand annihilated $H_4$s these expansions do not introduce organic 6-cycles.

E.1. Gadget is covered by two cycles

If a $H_4'$ is covered by two disjoint cycles in $F_i$, then the internal edge cannot be part of the 2-factor. This leaves only the single possibility depicted in Figs. 147–148, which takes two cycles of lengths $x + 2$ and $y + 2$ in $F_i$ and returns a single cycle of length $x + y + 14$ in $F_{i-1}$ after the expansion. $x + 2$ and $y + 2$ are both at least 6, so $x + y + 14 > 8$, meaning the resulting cycle in $F_{i-1}$ cannot be a cycle of length 6.
Fig. 135. A cycle of length $x + y + z + 3$ passes through a $H'_1$.

Fig. 136. The cycle from the previous figure, after expanding the gadget, is now a cycle of length $x + y + z + 13$.

Fig. 137. A cycle of length $x + y + z + 3$ passes through a $H'_1$.

Fig. 138. The cycle from the previous figure, after expanding the gadget, is now a cycle of length $x + y + z + 13$.

Fig. 139. A cycle of length $x + y + z + 3$ passes through a $H'_1$. 

Fig. 140. The cycle from the previous figure, after expanding the gadget, is now a cycle of length $x + y + z + 13$.

Fig. 141. A cycle of length $x + y + z + 3$ passes through a $H'_4$.

Fig. 142. The cycle from the previous figure, after expanding the gadget, is now a cycle of length $x + y + z + 13$.

E.2. Gadget is covered by one cycle

If a $H'_4$ is covered by a single cycle in $F_i$, then we consider when the internal edge is part of $F_i$ and those when the internal edge is not included. First consider the cases when the internal edge is included in $F_i$. Then $F_i$ passes through either edge 1 or 3 and either edge 2 or 4. Each of these possibilities takes a cycle of length $x + 3$ in $F_i$ and returns a cycle of length $x + 13$ in $F_{i-1}$. $x + 3 \geq 6$, so $x + 13 > 8$, meaning the resulting cycle in $F_{i-1}$ cannot be a cycle of length 6. The case when exiting edges 1 and 4 are included in $F_i$ is symmetric to the case when exiting edges 2 and 3, so only the first of these cases is included in this appendix. Then, there are three unique cases to consider where the internal edge is included in $F_i$. We consider these cases in the first six figures of this subsection.

Next, consider when the internal edge is not included in $F_i$. Then all four exiting edges of the gadget must be used. There are two cases to consider where $F_i$ can pass through these four edges, because after exiting edge 1, the cycle can re-enter the gadget at either edge 2 or 4. Each of these possibilities takes a cycle of length $x + y + 4$ in $F_i$ and returns a cycle of length $x + y + 14$ in $F_{i-1}$. $x + y + 4 \geq 6$, so $x + y + 14 > 8$, meaning the resulting cycle in $F_{i-1}$ cannot be a cycle of length 6. We consider both cases in the last four figures of this subsection (see Figs. 150–158).
Fig. 143. A cycle of length $x + y + z + 3$ passes through a $H'_5$.

Fig. 144. The cycle from the previous figure, after expanding the gadget, is now a cycle of length $x + y + z + 13$.

Fig. 145. A cycle of length $x + y + z + 3$ passes through a $H'_5$.

Fig. 146. The cycle from the previous figure, after expanding the gadget, is now a cycle of length $x + y + z + 13$.

Appendix F. $H_5$s

Two edges of each $H'_5$ is covered by a cycle in $F_i$. Then, there are three cases to consider, when each of the gadget’s edges are excluded from $F_i$. In each of these cases, we start with a cycle of length $x + 2$ in $F_i$ and are returned a cycle of length $x + 14. x + 2 \geq 6$, so $x + 14 > 8$, meaning these expansion operations cannot introduce an organic 6-cycle into the 2-factor (see Figs. 159–164).
Fig. 147. Two cycles of lengths $x + 2$ and $y + 2$ pass through a $H'_4$. 

Fig. 148. The cycles from the previous figure, after expanding the gadget, are now a single cycle of length $x + y + 14$. 

Fig. 149. A cycle of length $x + 3$ passes through a $H'_4$. 

Fig. 150. The cycle from the previous figure, after expanding the gadget, is now a cycle of length $x + 13$. 

Appendix G. $H_6$s

If a replacement operation annihilates (see Definition 2.3) an $H_6$ configuration then edges 1 and 2 are a single edge. We know the gadget edges are organic, but the numbered edge may be an $S'_2$. We cover the $H_6$ with the non-organic 14-cycle shown in Fig. 168.

There are two cases to consider for a $H'_6$, when the edge is included in $F_i$ and when it is not. In both cases, expanding the gadget cannot introduce an organic 6-cycle to $F_{i-1}$. We consider both cases in this section (see Figs. 165–167).

Appendix H. $D_1$s

There are two cases to consider for a $D'_1$, when the internal edge is included in $F_i$ and when it is not. The second case introduces a short cycle of two parallel edges. We discuss this case in Section 3.5.
**Fig. 151.** A cycle of length $x + 3$ passes through a $H'_4$. 

**Fig. 152.** The cycle from the previous figure, after expanding the gadget, is now a cycle of length $x + 13$. 

**Fig. 153.** A cycle of length $x + 3$ passes through a $H'_4$. 

**Fig. 154.** The cycle from the previous figure, after expanding the gadget, is now a cycle of length $x + 13$. 

**Fig. 155.** A cycle of length $x + 4$ passes through a $H'_4$. 
Fig. 156. The cycle from the previous figure, after expanding the gadget, is now a cycle of length $x + y + 14$.

Fig. 157. A cycle of length $x + 4$ passes through a $H_j$.

Fig. 158. The cycle from the previous figure, after expanding the gadget, is now a cycle of length $x + y + 14$.

Fig. 159. A cycle of length $x + 2$ passes through a $H_j$.

Fig. 160. The cycle from the previous figure, after expanding the gadget, is now a cycle of length $x + 14$. 
Fig. 161. A cycle of length $x + 2$ passes through a $H'_5$.

Fig. 162. The cycle from the previous figure, after expanding the gadget, is now a cycle of length $x + 14$.

Fig. 163. A cycle of length $x + 2$ passes through a $H'_5$.

Fig. 164. The cycle from the previous figure, after expanding the gadget, is now a cycle of length $x + 14$.

Fig. 165. A $H'_6$, which is not included in the 2-factor.
Fig. 166. The cycle from the previous figure, after expanding the gadget, is now a cycle of length 14.

Fig. 167. A cycle of length $x + 1$ passes through a $H'_k$.

Fig. 168. The cycle from the previous figure, after expanding the gadget, is now a cycle of length $x + 15$.

Fig. 169. A cycle of length $x$ passes through a $D'_1$. 
Fig. 170. The cycle from the previous figure, after expanding the gadget, is now a cycle of length \( x + 2 \). When choosing between the two parallel edges, we select an edge that is an \( S'_3 \) that replaced an organic \( S_3 \) if at least one of the parallel edges is of this type. The original graph is simple, so at least one of nodes \( A_1, A_2, \) or the two parallel edges must be non-organic. If neither edge is an \( S'_3 \) then we select one of the two parallel edges such that the expanded cycle is non-organic.

Fig. 171. The 2-factor does not pass through the center of this \( D'_1 \).

Fig. 172. The cycle from the previous figure is unchanged, and a new cycle of only two parallel edges is introduced. This case is analyzed in Section 3.5.
References