Department of Mathematics and Computer Science
Software Engineering and Technology

Proving Lehmer's Conjecture: Hamiltonian Paths in Neighbor-Swap Graphs

Master's Thesis

Max Opperman

Supervisor:
dr. ir. T. Verhoeff

Version 1.1

Eindhoven, December 2024

Abstract

This work addresses the problem of permutation generation by neighbor swaps. This problem is transformed into a graph problem by considering each permutation a node and each neighbor swap between two permutations an edge. The objective is to identify a Hamiltonian path within this “neighbor-swap graph”. For instances where a Hamiltonian path is not possible, D. H. Lehmer proposed a relaxation using single spurs. T. Verhoeff conjectured a solution where these spurs are so-called “stutter permutations”. We will prove that Verhoeff’s conjecture holds and that a Hamiltonian path exists on the non-stutter permutations for every neighbor-swap graph, and in most cases, even a Hamiltonian cycle. Therefore we will also prove Lehmer’s conjecture. We also developed a software project in Python that supports most of our findings.

Contents

List of Figures

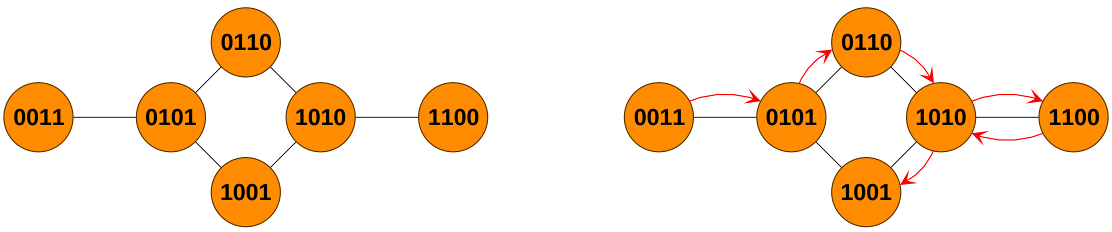
List of Tables

Chapter 1

Introduction

In 1965, D. H. Lehmer introduced a problem focused on generating all permutations of a multiset of elements [12]. His approach was limited to utilizing only adjacent transpositions of distinct elements. These adjacent transpositions are called neighbor swaps. One can construct a neighbor-swap graph by turning each permutation into a node and drawing edges for the neighbor-swaps. To generate all permutations by neighbor swaps is equivalent to finding a Hamiltonian path in such a graph. Not all neighbor-swap graphs admit a Hamiltonian path, as shown in Fig. 1.1a. A signature describes the number of occurrences of every element in a multiset.

Formally, a signature is a tuple $(k_0, k_1, \ldots, k_{K-1})$ of K colors. Every color $0 \leq i < K$ occurs $k_i \geq 0$ times in a multiset. This signature results in a multiset of size $n = \sum_{i=0}^{K-1} k_i$. The i^{th} color is represented by the number i. For example, multiset $[0, 0, 1, 2]$ has signature $(2, 1, 1)$. The first element of a permutation has index 0 and we say element 1_0 is of color 1 and occurs at index 0 for that permutation. Other digits or symbols can replace elements because it will result in isomorphic neighbor-swap graphs.



(a) The neighbor-swap graph of signature $(2, 2)$.

(b) An example of a Lehmer path in a neighbor-swap graph.

Figure 1.1: Neighbor-swap graph example with signature $(2, 2)$.

We have two options to relax the problem for the situations without Hamiltonian paths. We can either neglect the nodes that cannot be incorporated into the Hamiltonian path, or visit some nodes twice. Both solutions aim to minimize the number of deviations from a perfect Hamiltonian path. Lehmer proposed an approach that combines the solutions. He suggested extending a path to incorporate nodes outside the path using *spurs*. A spur is a node at distance 1 of a path in a graph. Therefore we can first neglect nodes and then later incorporate them. Fig. 1.1b shows a Hamiltonian path with a single spur. We call 1010 the spur base and 1100 the spur tip. A double spur occurs when the same node serves as the spur base of two distinct spur tips. However, we will only utilize single spurs. Lehmer defines an *imperfect* Hamiltonian path as follows: a path through a graph that traverses all nodes once, except the spur bases are visited twice. We will call this a *Lehmer* path (as is done by

Verhoeff [24]):

Definition 1. *A Lehmer path/cycle in a graph is a path/cycle, possibly with single spurs, that visits the spur bases twice and all other vertices once.*

We will prove Conjecture 1:

Conjecture 1. *A Lehmer path can be constructed for every neighbor-swap graph.*

To do this, we will first introduce related work in Chapter 2. This chapter also introduces the relevant terminology and definitions. We will also discuss two algorithms in more detail because they will be used in the proof of Conjecture 1. Those are the algorithms of Verhoeff [24] and a part of the proof of Stachowiak [19]. Then we will discuss how Verhoeff's work is extended to find Hamiltonian cycles in neighbor-swap graphs where the spur tips are left out in Chapter 3. In other words, Chapter 3 proves Conjecture 1. Continuing, we address the software that the author developed to aid the proving process in Chapter 4. With Chapter 5 we conclude the thesis by reflecting on our contributions and by discussing possible future work.

Chapter 2

Related Work

This chapter starts by giving a general outline of the related work regarding the context of permutation generation. Then two studies that utilize neighbor-swaps are discussed in more detail. We summarize their proofs because they will be used in later chapters.

2.1 Literature Study

Some of the early work on permutation generation by neighbor-swaps was done by three independent researchers: Steinhaus [21], Johnson [8], and Trotter [23]. They developed an algorithm to generate all permutations of n distinct elements using only neighbor swaps. This algorithm is called the Steinhaus-Johnson-Trotter algorithm. If the multiset size equals the number of colors ($n = K$), the neighbor-swap graph will result in a *permutahedron*. It is well-known in the literature that this graph structure admits a Hamiltonian cycle [6].

Ruskey [17] and Knuth [10] have published more extensive work on problems regarding permutation generation. These books are dedicated to the problem of Combinatorial Generation. These problems are important in areas like molecular biology, optimization, cryptography, etc. Combinatorial generation also includes subset permutations and Gray code problems (which will both be explained for completeness). Subset permutations are problems where we have permutations of length l for a multiset of size n with $l \leq n$. This means we leave out elements and thus change the signature for some permutations. Depending on l and the starting signature, it varies how often we change the signature of the permutations. While these algorithms might give some insights, we are restricted to only the case where we have permutations with length n for a multiset of size n.

Gray Codes [14] are permutations where only a single element is incremented or decremented by 1 to generate the next permutation. Consequently, the signature of the permutation changes between permutation changes. Gray Codes are important in error correction and preventing incorrect outputs from electromechanical switches.

A variation on this is the subset problem where successively generated permutations are obtained by changing exactly one element. This is called the *Minimal Change Property* [5]. The difference is that elements can only increment or decrement by 1 with Gray codes. A subset permutation problem can also admit the Minimal Change Property. Here elements can be changed to any other element in the multiset instead of only incrementing or decrementing.

However, there is a relation between Gray codes and neighbor-swap permutation generation. Using Gray code reduction [13], Merino et al. prove that several permutation generation problems are NP-complete. The problems include permutation generation using only: neighbor-swap, transposition, substring reversal, and substring rotation. The neighbor-swap permutation generation problem, referred to as *PermSwapGC*, is defined as follows:

PermSwapGC

Input: A list of m permutations of length n; $\tau_1, \dots, \tau_m \in \mathbb{P}_n$. Where $\mathbb{P}_n$ is the set of all permutations of length n.

Question: Is there a permutation $\pi \in \mathbb{P}_n$ such that $\tau_{\pi(i)}$ and $\tau_{\pi(i+1)}$ differ in an adjacent transposition for every $1 \leq i < m$?

PermSwapGC is proven to be NP-complete in [13]. The difference between this problem and the problem we are addressing is that PermSwapGC does not require that the input consists of *all* permutations. For our problem, *all* permutations must be in the input set. Therefore, PermSwapGC concerns permuting a list of permutations rather than a list of elements, as in our problem. Consequently, Merino et al.'s [13] proof does not establish the NP-completeness of our problem.

Lehmer [12] presented the problem of permutation generation by neighbor swaps. He used that the number of permutations can be computed from just its signature using a multinomial coefficient. So the number of permutations of signature $(k_0, \dots, k_{K-1})$ equals $M(k_0, \dots, k_{K-1})$. A multinomial coefficient can be computed as shown in (2.1):

$$\begin{aligned} M(k_0, \dots, k_{K-1}) &= \binom{n}{k_0\ k_1\ \dots\ k_{K-1}} \\ &= \frac{n!}{k_0!\ k_1!\ \dots\ k_{K-1}!} \end{aligned} \tag{2.1}$$

Moreover, Lehmer observed that neighbor-swap graphs are bipartite. An *inversion* of a permutation refers to a (not necessarily adjacent) pair occurring out of order. Two elements occur out of order when the left element is greater than the right element. So the number of inversions in 10021 equals three because $(1_0, 0_1)$, $(1_0, 0_2)$, and $(2_3, 1_4)$ are out of order. Parity is the property of an integer, whether it is even or odd. The parity of a permutation is the parity of the number of inversions of that permutation. A single neighbor swap changes the parity since it only changes the number of inversions by one. Therefore the neighbor-swap graph can be classified as a bipartite graph. The two parts are composed of the even and odd permutations. For bipartite graphs, it is known that there are no odd-length (sub)cycles [1, 2]. A Hamiltonian path is possible only when the difference between the number of even and odd nodes does not exceed one. This condition arises because the path must alternate between the even and odd permutations. The concept that the number of odd permutations does not exceed the number of even permutations has been well-established in the literature, with various proofs presented over time. Verhoeff also contributed a new proof of this result in [24].

Definition 2. *The defect of a neighbor-swap graph G is denoted by $d(G)$. This defect equals the difference between the graph's number of even and odd permutations.*

For a neighbor-swap graph G to admit a Hamiltonian cycle it is required that $d(G) = 0$. For a Hamiltonian path, the neighbor-swap graph must have $d(G) \leq 1$ [12]. Thus the number of spurs must be at least $d(G)-1$ for a neighbor-swap graph to admit a Lehmer path; otherwise, nodes remain unvisited.

Lastly, Lehmer presented an algorithm to find a Lehmer path in a neighbor-swap graph. This algorithm can be found in Appendix A.1. The author implemented the algorithm as part of the Preparation Phase for this Graduation Project. However, it failed to produce valid solutions for our problem. The algorithm fails for $n = K = 6$ or permutations of length $n \geq 8$ if the number of colors is less than the permutation length. For example signature $(5, 3)$ or $(3, 3, 2)$. Lehmer's algorithm takes a greedy approach by iteratively picking the node with the least number of connections and forming a path.

Ruskey [16] proved for the case of two colors that a Hamiltonian path exists if and only if both colors occur an odd number of times. Stachowiak [19] proved for the case of at least three colors that a Hamiltonian path exists if and only if at least two colors occur an odd number of times, and under certain conditions even a Hamiltonian cycle. We will discuss the proof of Stachowiak [19] further in §2.2. In short, he proved that a Hamiltonian path of even length in a neighbor-swap graph can be extended to a Hamiltonian cycle by adding one or more new colors. This means the Hamiltonian cycle is on a larger neighbor-swap graph. Stachowiak also showed that in binary odd-odd neighbor-swap graphs, there exists a Hamiltonian cycle if two vertices are removed; $1^{k_1}0^{k_0}$ and $0^{k_0}1^{k_1}$ [20]. However, we will focus on the Hamiltonian paths of the odd-odd case where these vertices are included.

To obtain a Hamiltonian path in an odd-odd neighbor-swap graph we will use the technique of Verhoeff [24]. He showed that for the case where $K = 2$, it is possible to generate a Lehmer path, proving the conjecture of [12] for two colors. Verhoeff presents the idea of stutter permutations (defined below). This uses the notion of *lio pairs*; adjacent pairs in a permutation whose left index is odd. Lio pairs look like this;

$$e_1e_2|e_3e_4|\dots|e_{2j-1}e_{2j}|\dots$$

and when the multiset has an odd length, there is a trailing element that does not belong to a lio pair.

Definition 3. *A permutation is a stutter permutation if and only if all lio pairs consist of two equal elements, i.e. when the element at index $2j$ equals the element at $2j-1$ for all relevant j.*

Stutter permutations take the form $aabb\dots yy$, with a possibly trailing z (similar to lio pairs). Verhoeff shows that the number of stutter permutations equals the defect of a neighbor-swap graph. This means that it is sweetly reasonable to choose the stutter permutations as the spur tips in Lehmer's conjecture. By applying this, Verhoeff proves that all graphs with $K = 2$ admit a Lehmer path. Moreover, Verhoeff presents an inductive proof that a neighbor-swap graph for $K \geq 3$ admits a disjoint cycle cover on non-stutter permutations. These theorems are discussed more detailed in § 2.3. Verhoeff also supervised several other projects that contribute to the context of Conjecture 1 [3, 18].

In 1976, Hu and Tien [7] published an algorithm that computes all permutations using neighbor swaps. However, it sometimes sees a whole block of elements of equal color as a single element. This means it uses different techniques to generate subsequent permutations than only neighbor swaps. Rivertz [15] also presents a similarly altered version of our problem. It works by creating permutations by adjacent transposition as before, however, when there are no neighbor-swaps that generate "new" permutations it differs from Lehmer's problem. Then only transpositions are allowed between two elements if all elements between the transposed elements are equal to the smallest of those two elements. These transpositions can also be seen as restricted rotations where all elements in between are equal. Furthermore, this means there is no notion of spurs such as in [12], although sometimes the rotations can be seen as spurs, for example, in the case of signature $(2,2)$. Here the algorithm of Rivertz yields the path $1010 \sim 1001 \sim 1100$. This can be seen as a spur since we can take the path $1010 \sim 1001 \sim 1010 \sim 1100$. Here 1010 is the spur base and 1001 is the spur tip.

Rivertz [15] poses the question of whether the total motion (defined below) of all transpositions is minimal for his algorithm. The algorithm is also compared to the algorithm of Eades [5] in this regard. Here, the motion is the sum of the differences in index between the two transposed elements. The author found that this is not true. For example in the case of the signature $(3,3)$. This should result in a minimal motion of 19, such as can be achieved using Verhoeff's [24] or Ruskey's [16] theorems but Rivertz's algorithm has a total motion of 23. This example is demonstrated in Fig. A.1.

Several other algorithms using rotations have been presented in the literature. Williams [25] presents an algorithm to generate all permutations of a multiset by prefix rotations. Ruskey & Williams [26] provide an algorithm to generate all permutations of binary signatures of length n by prefix rotations. While their methods are distinct from ours, the applications of these algorithms are within the same fields. These algorithms also contain some minimal change between subsequently generated permutations.

Various other algorithms address the problem of efficiently generating multiset permutations. Takaoka [22] presents an algorithm centralized around tree traversal. His algorithm takes $\mathcal{O}(1)$ time between permutations. This property was also achieved using recursive algorithms like [4] and loopless algorithms like [11] to generate permutations of a multiset.

2.2 Stachowiak

Stachowiak [19] addresses a more general setting by using posets instead of multisets. His proof implies that the neighbor-swap graphs of signatures containing two or more odd-occurring colors admit a Hamiltonian cycle — except for the $(\textit{even}, 1, 1)$ signature which only admits a Hamiltonian path. This proof is explained in more detail in the Preparation Phase report of this Graduation Project. We implemented Stachowiak's main theorem but ultimately did not use it in our final proof or code. This was because the theorem does not impose conditions on specific edges, which becomes necessary later in our proof. However, we still use the lemma that generates the Hamiltonian cycle in the neighbor-swap graph for $(\textit{odd}, 1, 1)$ signatures and Hamiltonian paths for the $(\textit{even}, 1, 1)$ signatures. For this we borrow Stachowiak's definitions of parallel edges and gluing them:

Definition 4. *In a neighbor-swap graph $G = (V, E)$, edges $a \sim b \in E$ and $c \sim d \in E$ are parallel if and only if cross edges $a \sim c \in E$ and $b \sim d \in E$ as well. That is, if both b and c differ by a single neighbor-swap from a, and both a and d differ by a single neighbor-swap from b.*

Definition 5. *To combine path (or cycle) p and cycle q into one path (or cycle if p is a cycle), we can glue a pair of parallel edges $a \sim b$ (from p) and $c \sim d$ (from q). This involves traversing p until a, then taking the cross edge $a \sim c$, traversing q from c to d (excluding $c \sim d$), and finally using the cross edge $b \sim d$ to return to p and complete the traversal. This construction ensures that all vertices of p and q are visited, but the parallel edges $a \sim b$ and $b \sim d$ are excluded from the resulting path (or cycle), while the cross edges $a \sim c$ and $b \sim d$ are included.*

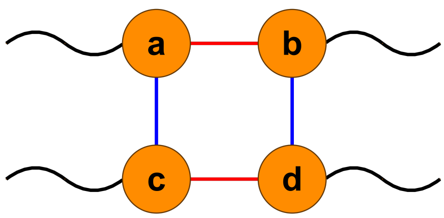


Figure 2.1: Gluing of edges visualized. The red edges $a \sim b$ and $c \sim d$ are parallel edges in two paths. The blue edges $a \sim c$ and $b \sim d$ are cross edges.

Fig. 2.1 shows how the gluing of edges can be applied to a path with edge $a \sim b$ to another path with edge $c \sim d$. The red edges show the parallel edges (the prior named edges). The blue edges show the *cross edges*. The gluing process replaces the parallel edges with the cross edges.

Lemma 1 is rephrased from Stachowiak's Lemma 2 [19]. The original lemma is more general because it addresses the poset setting and uses different terminology.

Lemma 1. *The neighbor-swap graph G of signature $(k_0, 1, 1)$ has the following properties:*

1. *If k_0 is odd, then G contains a Hamiltonian cycle.*

2. *If k_0 is even, then G contains two non-Hamiltonian cycles. Each of these cycles contains all but two vertices of G. This is visualized in Fig. 2.2a without the blue edge and with the orange dotted edge. The omitted vertices are either*

   (a) 120^{k_0} *and* 210^{k_0} *or*

   (b) $0^{k_0}12$ *and* $0^{k_0}21$.

This lemma is proven by using the following paths d_i for all $0 \le i \le k_0$:

$$d_i = \left[10^{k_0-i}20^i, 010^{k_0-i-1}20^i, \cdots, 0^{k_0-i}120^i, 0^{k_0-i}210^i, \cdots, 20^{k_0-i}10^i\right] \tag{2.2}$$

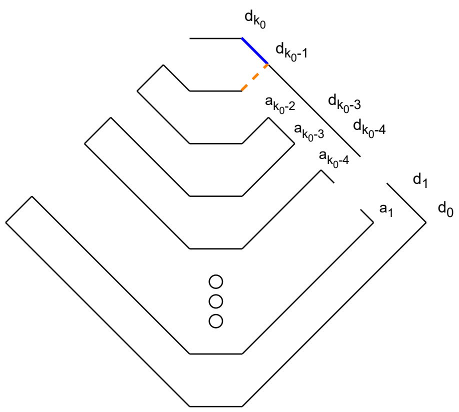
Note that path d_i has length $2 \cdot (k_0 + 1 - i)$. We pairwise connect the start of path d_j (with j even) with the start of path d_{j+1}. This is shown in the edges on the left side of Fig. 2.2a and Fig. 2.2b. Then the last edge of d_i is glued with the last edge of d_{i+1}, which is possible since the last two nodes of every d_i and d_{i+1} differ by a neighbor-swap, except when k_0 is even. This creates $\lfloor (k_0+1)/2 \rfloor$ disjoint cycles with parallel edges. The number of cycles is independent of the parity of k_0. We glue the parallel edges are between the last edge of every d_i and d_{i+1} for i is odd indicated with a_1 and a_2, ..., a_{k_0-2} and a_{k_0-1} in Fig. 2.2. In case k_0 is even, the gluing of edges creates an extra path d_{k_0} of length two; the edge $120^{k_0} \sim 210^{k_0}$ which is not included in the cycle. This concludes Case 2(a) of the lemma.

Case 2(b) is similar with paths

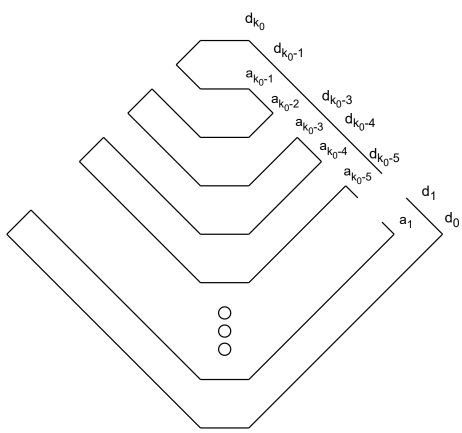
$$d_i' = \left[0^i20^{k_0-i}1, 0^i20^{k_0-i-1}10, \cdots, 0^i210^{k_0-i}, 0^i120^{k_0-i}, \cdots, 0^i10^{k_0-i}2\right] \tag{2.3}$$

for $0 \le i \le k_0$. So then the two nodes that are left out in the case that k_0 is even are the edge $0^{k_0}12 \sim 0^{k_0}21$. In practice, we can view the graph from Case 2(b) as the graph from Case 2(a) turned upside down.

When k_0 is even, the cycle on the first $k_0 - 1$ paths can be extended to a Hamiltonian path by starting from vertex 120^{k_0} with edge $120^{k_0} \sim 210^{k_0}$ and then 2010^{k_0-1}, followed by the Hamiltonian cycle.



(a) Path (with the blue edge) or non-Hamiltonian cycle (with the orange edge) in the neighbor-swap graph of signature $(\mathit{even}, 1, 1)$.

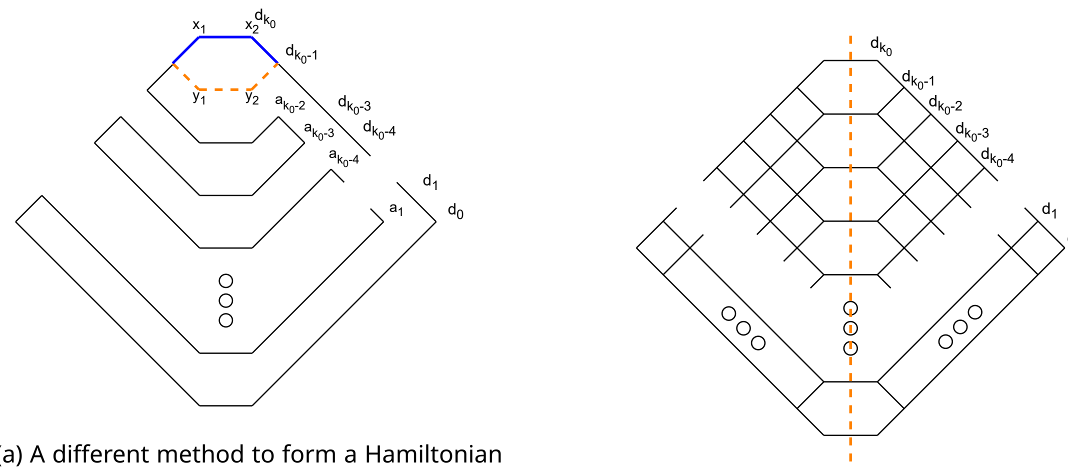


(b) Hamiltonian cycle in the neighbor-swap graph of signature $(\mathit{odd}, 1, 1)$.

Figure 2.2: Structural comparison between the main two cases of Lemma 1

The construction of these paths can be found in Fig. 2.2. Fig. 2.2a represents the $(\mathit{even}, 1, 1)$ signature. The cycle without the top two nodes contains the orange dotted edge and the blue edge is left out. However, the blue edge in Fig. 2.2a illustrates the connection to $120^{k_0} \sim 210^{k_0}$ or $0^{k_0}12 \sim 0^{k_0}21$ (representing Case 2(a) or Case 2(b) respectively). The orange dotted edge in Fig. 2.2a indicates the edge that must be removed when forming the Hamiltonian path. Fig. 2.2b shows the Hamiltonian cycle in the neighbor-swap graph with signature $(odd, 1, 1)$.

Moreover, what is not mentioned in Stachowiak's work is that there are more cycles possible in this case. Two of those cycles become apparent when looking at Fig. 2.3a. These cycles of $(even, 1, 1)$ signatures will be used in §3.7. The cycle that Stachowiak found omits the nodes $x_1 \sim x_2 = 120^{k_0} \sim 210^{k_0}$ in d_{k_0} and $x'_1 \sim x'_2 = 0^{k_0}12 \sim 0^{k_0}21$ in d'_{k_0}. However, we can also remove the edge $y_1 \sim y_2 = 0120^{k_0-1} \sim 0210^{k_0-1}$ in d_{k_0-1} or $y'_1 \sim y'_2 = 0^{k_0-1}210 \sim 0^{k_0-1}120$ in d'_{k_0-1}. In that case, it is also possible to form a cycle with the edge $x_1 \sim x_2$ when y_1 and y_2 are omitted, and similar for $x'_1 \sim x'_2$ with $y'_1 \sim y'_2$. This extends to more edges in the $(even, 1, 1)$ neighbor-swap graph. If we draw all edges in the graph it will become clear that the edges that can be left out to form a Hamiltonian cycle are the horizontal ones in Fig. 2.3a. The proof of this lies behind the fact that if we draw any line in a planar graph (like this one), a Hamiltonian cycle cannot cross this line an odd number of times. An example of such a line is shown in Fig. 2.3b; note that all the edges between the d_i and d'_i paths are drawn here. Say the Hamiltonian cycle has a starting region, then we must cross the line (to the other region) an even number of times to get back into the starting region (and close the cycle). However, the $(even, 1, 1)$ neighbor-swap graph contains an odd number of edges crossing the vertical line in Fig. 2.3b. This means we have to omit one of these pairs of adjacent nodes to form the cycle.



(a) A different method to form a Hamiltonian path (or cycle when omitting two nodes) in the neighbor-swap graph for $(even, 1, 1)$ signatures than in Case 2(a) and Case 2(b).

(b) Odd-sized cut in a neighbor-swap graph with an $(even, 1, 1)$ signature.

Figure 2.3: Different representations of the neighbor-swap graph for $(even, 1, 1)$ signatures.

Lemma 1 is used for $(even, 1, 1)$ and $(odd, 1, 1)$ signatures in our proof. The work of Stachowiak continues by providing an inductive proof to construct Hamiltonian cycles in neighbor-swap graphs; this is Lemma 11 in his work. The prerequisite for this is that the neighbor-swap graph of a subsignature contains an even number of vertices strictly greater than two and admits a Hamiltonian path. The inductive step in this algorithm is on the number of colors. Our proof uses an inductive step on the permutation length, i.e. adding a trailing element to a subcycle. Stachowiak lexicographically interleaves the colors. However, since we require a stronger alternative to Stachowiak's theorem, we will not utilize it in practice. Our theorem has a guarantee that certain edges are covered in the Hamiltonian cycle or path.

2.3 Verhoeff

Verhoeff [24] has two main results. One is proof that for the binary case (only two colors), a Hamiltonian path exists in the neighbor-swap graph of non-stutter permutations (Definition 3). The second result is that the neighbor-swap graphs of all multiset permutations admit a disjoint cycle cover on the non-stutter permutations. We will discuss both theorems here. The proof for the disjoint cycle cover was used as the basis for the rest of our work and will be described in more detail in Chapter 3. To conclude, Verhoeff also strengthened Conjecture 1. We will discuss this in the last part of this section.

2.3.1 Binary Case

The binary case occurs when we have two colors, signature (k_0, k_1). Verhoeff splits this into three cases: even-odd, even-even, and odd-odd.

We will briefly address each of these cases as they are comprehensively explained in Verhoeff's work. Formally, Verhoeff is proving the following theorem:

Theorem 1. *The neighbor-swap graph of binary (i.e., with two colors) non-stutter permutations admits a Hamiltonian path. If k_0 and k_1 are both even then there exists a Hamiltonian cycle. Moreover, we have:*

1. *The path for the odd-even case (k_0, k_1) can run between*

$$0^{k_0}1^{k_1} \overset{*}{\sim} 1^{k_1-1}0^{k_0}1 \tag{2.4}$$

   Note that this path cannot avoid the edge

$$0^{k_0}1^{k_1} \sim 0^{k_0-1}\underline{10}1^{k_1-1} \tag{2.5}$$

2. *The cycle for the even-even case (k_0, k_1) can include these two specific edges (the underlined elements in the middle are swapped):*

$$\begin{aligned} 0^{k_0-2}\underline{10}1^{k_1-1}0 &\sim 0^{k_0-1}1^{k_1}0 \\ 1^{k_1-2}\underline{01}0^{k_0-1}1 &\sim 1^{k_1-1}0^{k_0}1 \end{aligned} \tag{2.6}$$

This theorem is proven using strong induction on the size $n = k_0 + k_1$ in the multiset. The main idea behind the proof is to fix some trailing elements and then show that there are enough parallel edges between the graphs to connect the subgraphs into a Hamiltonian path or cycle.

2.3.1.1 Odd-even

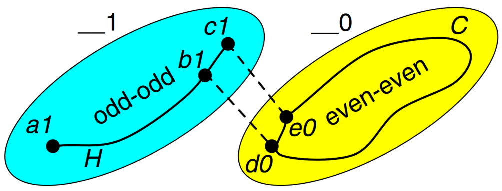


Figure 2.4: Binary odd-even case split on the trailing element. Figure from Verhoeff [24].

This case also covers the even-odd case as we can swap all 0's by 1's and 1's by 0's at the end. So let's take $k_0 \geq 1$ odd and $k_1 \geq 2$ even. By restricting the last element of the permutations, we get one part with trailing 0 and signature $(k_0 - 1, k_1)$. The other part has trailing 1 and signature $(k_0, k_1 - 1)$. These parts combined must contain all permutations of signature (k_0, k_1) because 0 and 1 are the only possible trailing elements. The part with trailing 0 is reduced to an even-even case with stutter permutations and the other part to an odd-odd case without stutter permutations (by definition). By the induction hypothesis, a path can now be constructed from $a = 0^{k_0}1^{k_1-1}$ to $c = 1^{k_1-1}0^{k_0}$ using edge $b \sim c$ with $b = 1^{k_1-2}010^{k_0-1}$. Note that $b1$ and $c1$ can be connected with the even-even case cycle (see §2.3.1.2) because the second to last element is 0 and the last element 1, see Fig. 2.4. Let's call $d = 1^{k_1-2}010^{k_0-2}1$ and $e = 1^{k_1-1}0^{k_0-1}1$. By gluing parallel edges $b1 \sim c1$ and $d0 \sim e0$, we create the cross edges $b1 \sim d0$ and $c1 \sim e0$. This obtains a Hamiltonian path in the neighbor-swap graph of an odd-even or even-odd signature.

2.3.1.2 Even-even

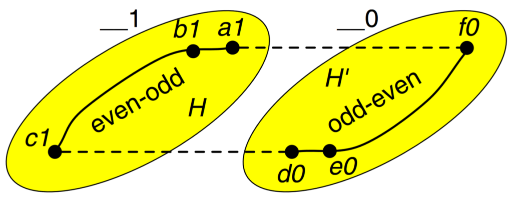


Figure 2.5: Binary even-even case split on the trailing element. Figure from Verhoeff [24].

We may assume $k_0 \geq 2$ and $k_1 \geq 2$ are even. Again we split on the trailing element, obtaining an even-odd and an odd-even case. By the induction hypothesis, we have a path in the even-odd part from $a1 = 1^{k_1-1}0^{k_0}1$ to $c1 = 0^{k_0-1}1^{k_1-1}01$. By interchanging the trailing element we obtain an edge from $a1$ to $f0 = 1^{k_1-1}0^{k_0-1}10$ and from $c1$ to $d0 = 0^{k_0-1}1^{k_1}0$. The induction hypothesis also provides the path from $d0$ to $f0$. Thus we obtain a cycle as in Fig. 2.5. Node $b1 = 1^{k_1-2}010^{k_0-1}1$ and $e0 = 0^{k_0-2}101^{k_1-1}0$ are parts of unavoidable edges in the induction hypothesis. Nodes $a1$ and $d0$ have degree two and are part of the Hamiltonian cycle. This implies that the Hamiltonian cycle must cover both edges connected to these nodes. So the even-even case results in a Hamiltonian cycle with guaranteed edges

$$0^{k_0-2}\underline{10}1^{k_1-1}0 \sim 0^{k_0-1}1^{k_1}0 \tag{2.7}$$

$$1^{k_1-2}\underline{01}0^{k_0-1}1 \sim 1^{k_1-1}0^{k_0}1. \tag{2.8}$$

2.3.1.3 Odd-odd

Since $k_0 = 1$ or $k_1 = 1$ gives us a linear graph, we take $k_0 \geq 3$ and $k_1 \geq 3$ are odd. For this case, we split the graph based on the trailing two elements: permutations with trailing 11, 01, 10, and 00. The 01 and 10 parts give us a cycle by the induction hypothesis as shown in §2.3.1.2. These parts visit the nodes $c01 = 1^{k_1-2}0^{k_0-1}101$ and $d1 = 0^{k_0-2}1^{k_1-1}010$ respectively. The 11 part contains a Hamiltonian path from $a11 = 0^{k_0}1^{k_1}$ to $b11 = 1^{k_1-2}0^{k_0}11$ by the induction hypothesis. Similarly, the 00 part contains a Hamiltonian path from $e00 = 0^{k_0-2}1^{k_1}00$ to $f00 = 1^{k_1}0^{k_0}$. So we can connect these paths and cycles as in Fig. 2.6.

The harder part is to incorporate the stutter permutations of the even-even part. The 01 and 10 cycles are isomorphic and parallel by swapping the trailing two elements. The even-even cycles are of even length because they are bipartite [1]. So we will copy Verhoeff's lemma:

Lemma 2. *Two parallel and isomorphic cycles of even length and parallel single spurs can be combined into one cycle. Furthermore, we can select a specific edge of a cycle without spurs to appear in the combined cycle.*

However, we will strengthen Verhoeff's lemma in Definition 6 for a zig-zag cycle. We use these zig-zag cycles to incorporate the stutter permutations as single spurs in parallel and isomorphic cycles with two distinct trailing elements.

Definition 6. *A zig-zag cycle in a neighbor-swap graph is a Hamiltonian cycle on two parallel and isomorphic subcycles potentially with single spurs. Both subcycles have even lengths, one has two trailing elements* xy*, and the other* yx*. The spurs can be specified as the stutter permutations in the subcycle; appended with trailing* xy *or* yx*. Each stutter permutation is connected to the spur base where the rightmost pair of distinct elements of the stutter permutation are swapped. A zig-zag cycle is created by alternating between the* xy *and* yx *subcycle after each edge except when the end node of the edge is a spur base. When arriving at a spur base, rather than directly going to the spur base in the other cycle, go to the spur tip, then cross to the spur tip in the other cycle, and finally to the spur base in that other cycle.*

This specification of where the stutter permutations/spurs are connected is later (in Chapter 3) needed for $K \geq 3$. Since if $K = 2$, there is no dependency on where each stutter permutation is connected to the cycle. Let's take as an example the subcycles of signature $(2,2,2)$ with trailing elements 01 and 10, so we try to generate a zig-zag cycle for signature $(3,3,2)$. Now we add the stutter permutation 001122 in the subcycle on non-stutter permutations of signature $(2,2,2)$. The stutter permutation is a spur tip (at distance 1 from the cycle) and can be connected to either 010122 or 001212. We will use the latter edge (i.e. $001122 \sim 001212$) since the swap occurs between the rightmost two distinct elements of the stutter permutation. This extends to the signatures with two colors: 001100 will be connected to 001010 and not to 010100.

However, we must still look into two more trailing elements as in Fig. 2.7. Parallel edges connect the top and bottom parts of the figure by swapping the last two elements.

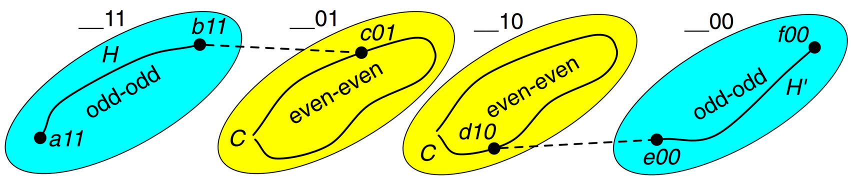


Figure 2.6: Binary odd-odd case split on the trailing two elements. Figure from Verhoeff [24].

Now we are left with three subcases; (k_0-1, k_1-3), (k_0-2, k_1-2), and (k_0-3, k_1-1). The even-even parts are yellow in Fig. 2.7 and show (k_0-1, k_1-3) & (k_0-3, k_1-1). Those parts must incorporate the stutter permutations similar to before using Lemma 2. Using the red dotted edges, two stutter permutations are shown to be incorporated in Fig. 2.7. The stutter permutations are outside the cycle since they are not in the even-even case of the induction hypothesis. Lastly, the four odd-odd parts are connected by creating a "square wave". This is possible since $p0101$ is parallel to both $p0110$ and $p1001$, and similarly for $p1010$ parallel to $p0110$ and $p1001$. Therefore we can construct a square wave path between four paths of even length with trailing; 0101, 0110, 1010, and 1001. The square wave path starts at $c01$, traverses the edges parallel to the even-even cycle, and ends at $d10$. Thus, the two paths for the permutations trailing in 11 and in 00 can be connected via the path from $c01$ to $d10$. The induction hypothesis provides the parallel edges:

$$u01 = 0^{k_0-2}1^{k_1-3}\underline{01}101 \ \sim\ 0^{k_0-2}1^{k_1-2}0101 = c'01$$
$$v10 = 1^{k_1-2}0^{k_0-3}\underline{10}010 \ \sim\ 1^{k_1-2}0^{k_0-2}1010 = d'10$$

This completes the proof for the odd-odd case and for Theorem 1.

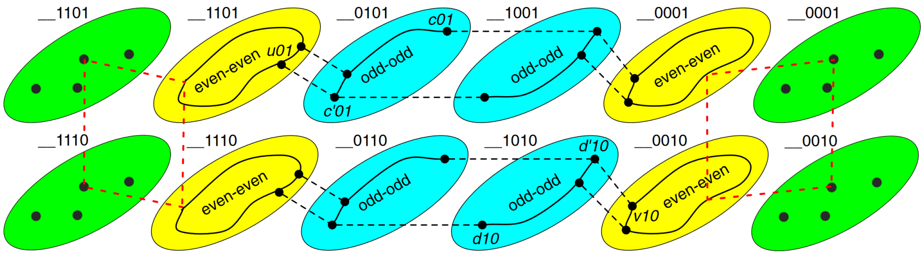


Figure 2.7: Binary odd-odd case split on the trailing four elements. Only the parts with trailing 01 and 10. Clarification of Fig. 2.6. The figure from Verhoeff [24] is extended with the green stutter permutation sections. The red dotted lines show those can be incorporated into the cycle.

2.3.2 Disjoint Cycle Cover

Verhoeff [24] also introduces the following theorem to simplify the proof of Conjecture 1.

Theorem 2. *When the signature consists of at least 3 colors and at most one is odd, the neighbor-swap graph of non-stutter permutations admits a disjoint cycle cover, i.e. a set of vertex-disjoint cycles that visit all permutations exactly once.*

The original proof of this theorem uses Stachowiak's theorem [19] for cases with two or more odd-occurring colors. However, we will show in Chapter 3 that the neighbor-swap graphs of these signatures can similarly be incorporated into the proof. Originally Verhoeff's proof uses a case distinction on the following signatures.

1. $(\mathit{even}, 2, 1)$
2. $(\mathit{even}, 1, 1)$
3. $(\mathit{odd}, 2, 1)$
4. All-even
5. One odd - rest even

We extended Verhoeff's proof to incorporate the other neighbor-swap graphs for signatures with two or more odd-occurring colors. The idea behind the proof remains the same: fix one or two trailing elements of the permutation of length n to obtain a signature with $n-1$ or $n-2$ elements. Using induction we can show that we can combine these subcycles with fixed trailing elements into a single Hamiltonian cycle.

2.3.3 Hamiltonian cycles in neighbor-swap graphs

In his work [24], Verhoeff suggested that Conjecture 1 could be strengthened through the definition of stutter permutations. To do so, he proposed the following conjecture.

Conjecture 2. *For every neighbor-swap graph, the subgraph consisting of its non-stutter permutations admits a Hamiltonian path. Furthermore, there even exists a Hamiltonian cycle on the non-stutter permutations, except when*

1. *the signature arity is zero or one, or*
2. *the signature is binary, and at least one of the k_i is odd, or*
3. *the signature is a permutation of $(2k, 1, 1)$.*

Additionally, he showed that the number of stutter permutations equals the defect of a neighbor-swap graph. Stutter permutations are at a distance 1 from a non-stutter permutation. Therefore, proving Conjecture 2 also shows that Conjecture 1 holds. Verhoeff proved this using the theorem:

"To prove Conjecture 1, it suffices to prove the existence of a Hamiltonian path (or cycle) on all non-stutter permutations."

The goal of this Master's Graduation Project is, therefore, to prove Conjecture 2, ultimately demonstrating that Conjecture 1 holds.

A part of Conjecture 2 is already proven to hold. We know that the signatures of arity zero or one do not contain any non-stutter permutations. Therefore Conjecture 2 must hold for that case. Verhoeff proved the binary case as we explained in §2.3.1. The $(\textit{even}, 1, 1)$ signatures have been proven to admit a Hamiltonian path by Stachowiak [19] and Verhoeff presented a different Hamiltonian path [24]. Therefore, the remainder of this Master's Graduation Project focuses on neighbor-swap graphs of signatures that admit a Hamiltonian cycle on the non-stutter permutations according to Conjecture 2.

Chapter 3

Lehmer's Conjecture for Three or More Colors

This section will present how the neighbor-swap graphs for signatures with three or more colors contain a Hamiltonian cycle on the non-stutter permutations. This excludes the $(\mathit{even}, 1, 1)$ signatures that only admit a Hamiltonian path as is demonstrated with Lemma 1. In other words, we will prove Conjecture 2.

While exploring the problem and ways to approach it, we found a case distinction that can be used to prove Conjecture 2. Therefore we distinguish between these signatures:

1. $(\mathit{odd}, 1, 1)$ — solved with Lemma 1
2. $(\mathit{even}, 2, 1)$
3. $(\mathit{odd}, 2, 1)$ (with the odd frequency ≥ 3)
4. $(\mathit{even}, 1, 1)$ (only a Hamiltonian path)
5. $(\mathit{odd}, \mathit{odd}, 1)$ (with both odd frequencies ≥ 3)
6. $(\mathit{even}, \mathit{odd}, 1)$ and $(\mathit{odd}, \mathit{even}, 1)$ (with the even frequency ≥ 4 and odd ≥ 3)
7. $(\mathit{even}, 1, 1, 1)$
8. $(\mathit{even}, 2, 1, 1)$
9. all-even
10. one odd - rest even
11. two odd - rest even
12. three or more odd - rest even

Using strong induction, we prove Conjecture 2. For each case, we will strengthen the theorem by stating some specific edges that will be present in the cycle (or path). The existence of these edges can then be exploited in the stronger induction hypothesis. For this, we assume the color frequencies in the signature are sorted in non-increasing order. If a subsignature is unsorted, we sort it before constructing the Hamiltonian cycle in the subgraph and then swap back the colors. The cases above appear because the neighbor-swap graphs of signatures $(\mathit{even}, \mathit{odd})$, $(\mathit{odd}, \mathit{odd})$, and $(\mathit{even}, 1, 1)$ only admit Hamiltonian paths and not cycles. Therefore we address cases 2 to 8 separately. The general technique to prove Conjecture 2 is similar to Verhoeff's proof for the binary case [24]. We fix some trailing element(s) and combine the Hamiltonian paths/cycles in the subgraphs into a Hamiltonian cycle on the non-stutter permutations. Thus we generate a disjoint cycle cover according to Theorem 2 and then combine the cycles to obtain a Hamiltonian cycle on the non-stutter permutations. We require some extra work for two odd - rest even signatures in §3.11 to incorporate some non-stutter permutations. The all-even signatures and the one odd - rest even signatures do not incorporate stutter permutations by design of Theorem 2. To construct the Hamiltonian cycles, we introduce the meta-graph of cycles (Definition 7). Using this we will prove that Conjecture 2 holds. The cycles are combined by gluing parallel edges according to Definition 5. Therefore we guarantee edges within a signature and use them as parallel edges in larger neighbor-swap graphs. A general overview of how the cases above depend on each other can be found in Appendix C.1.

3.1 Meta-graph of cycles

The idea behind the meta-graph of cycles follows from Theorem 2. Verhoeff [24] introduced a constructive way to generate a disjoint cycle cover in the neighbor-swap graphs of signatures with at most one odd-occurring element. We extended this by also forming such a disjoint cycle cover for signatures with two or more odd-occurring colors.

So we define the meta-graph of cycles:

Definition 7. *A meta-graph of cycles is a graph constructed as follows from a disjoint cycle cover of a neighbor-swap graph. Each cycle in the cycle cover is considered a meta-graph node (abbreviated as meta-node), and two such meta-nodes are connected when the two cycles contain a pair of cross edges. An edge in the meta-graph is called a meta-edge. We fix a certain number of trailing elements based on the number of odd-occurring colors:*

- *Two fixed elements for all-even signatures.*
- *One fixed element for one odd - rest even signatures.*
- *One fixed element for two odd - rest even signatures, except when the fixed element has an odd frequency, then we fix one more element.*
- *One fixed element for three or more odd - rest even signatures.*

We will discuss the connectivity of the meta-graph of cycles in each of the sections below. Note that the meta-graph of cycles is only created for the signatures:

- All-even
- One odd - rest even
- Two odd - rest even
- Three or more odd - rest even

Those cases admit a disjoint cycle cover that we will show connects into a Hamiltonian cycle. For this, we prove Lemma 4. However, for the two odd - rest even case, we will only use that the graph is connected. The proof for the existence of cross edges becomes easier this way. Furthermore, the meta-graph of cycles created by fixing one trailing element is fully connected. The reason for this is that we can fix any pair of distinct colors as the trailing two elements.

Lemma 3. *If the meta-graph of cycles is connected, then the underlying neighbor-swap graph admits a Hamiltonian cycle.*

Fig. 3.1 shows how a single cycle can be connected to multiple cycles in the disjoint cycle cover. This technique requires that a pair of cross edges between two subcycles is disjoint from a pair of cross edges to another subcycle. In other words, a permutation in a meta-node can only be connected to at most one permutation that is in another meta-node. Trivially this holds, because a cross edge can only have a trailing element that connects the path to one other subcycle, otherwise we would generate overlapping subcycles. For example, if one parallel edge has trailing xy and the other has trailing yx, then the pair of parallel edges can only connect the subcycles with trailing x and y and never with a trailing z. The two odd - rest even signatures use the technique of Fig. 3.1 to connect the meta-graph of cycles. Showing that a graph is connected can be done by showing that it admits a spanning tree. Moreover, we know that a fully connected graph admits a spanning tree.

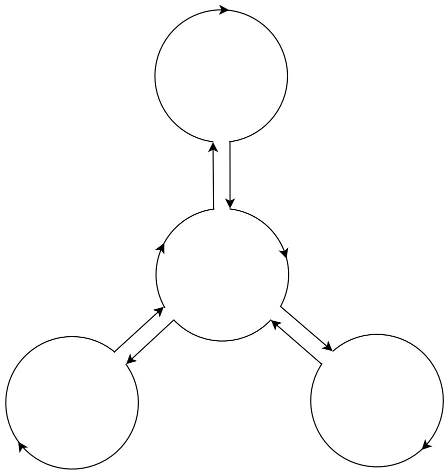


Figure 3.1: The meta-graph of cycles connected into a single cycle by connecting all cycles to a central cycle.

In the case of all-even, one odd - rest even, and three or more odd - rest even signatures we will use a slightly strengthened adaptation of Lemma 3. For these signatures we use that the meta-graph of cycles admits a Hamiltonian *path*. We want to obtain such a Hamiltonian path in the meta-graph of cycles as shown in Fig. 3.2. We can glue the cycles by their cross edges by listing the cycles in the order of the Hamiltonian path. The first cycle is cut on the parallel edge and attaches to the next cycle using the cross edge. We walk the first part of the cycle, use the cross edge to end up in the next cycle, and then walk back the rest of the cycle from the other cross edge to the last node. This idea of gluing cycles will connect the disjoint cycle cover into a single cycle, which then is a Hamiltonian cycle. Therefore Lemma 3 holds □.

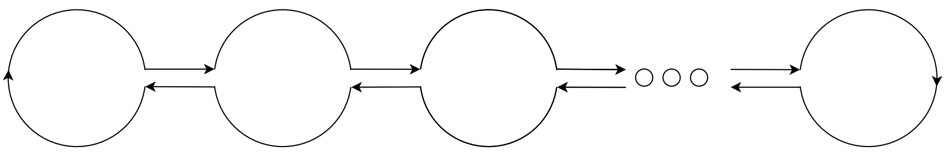


Figure 3.2: How a Hamiltonian path in the meta-graph of cycles connects the underlying neighbor-swap graph into a Hamiltonian cycle.

The method with the Hamiltonian path from Fig. 3.2 has the benefit that the first or last cycle can be left out and the disjoint cycle cover will still connect into a Hamiltonian cycle. We will use this in the proof of the one odd - rest even signatures.

We use Lemma 3 by proving that the meta-graph of cycles admits a Hamiltonian cycle in Lemma 4.

Lemma 4. *Each meta-graph of cycles contains a Hamiltonian cycle.*

The proof of this lemma in each of the sections below; we will split the proof into the cases of Definition 7. By showing that the meta-graph of cycles admits a Hamiltonian cycle, it is also a connected graph.

We extended Theorem 2 to incorporate the two odd - rest even and three or more odd - rest even signatures. This is because Stachowiak's theorem [19] does not guarantee the existence of any edges. For each of the signatures below, we provide a set of guaranteed edges. These edges are necessary to construct Hamiltonian cycles in neighbor-swap graphs that contain these signatures as subgraphs. We will say these subgraphs are represented by subsignatures according to Definition 8:

Definition 8. *A subsignature of a signature* $(k_0, k_1, \dots, k_{K-1})$ *is a tuple* $(m_0, m_1, \dots, m_{K-1})$ *where* $0 \le m_i \le k_i$ *for all* $0 \le i < K$*, and at least one* m_i *satisfies* $m_i < k_i$*.*

In other words, a subsignature $P = (m_0, m_1, \dots, m_{K-1})$ represents a subset of colors from the original multiset represented by $(k_0, k_1, \dots, k_{K-1})$ where each color i appears m_i times, with $m_i \le k_i$ and at least one $m_i < k_i$. This means that a subsignature corresponds to a multiset that is "contained within" the multiset represented by the original signature. We will also say that $S - P$ represents the signature of the remaining elements: $S - P = R = (k_0 - m_0, k_1 - m_1, \dots, k_{K-1} - m_{K-1})$.

We also define that we can 'translate' a neighbor-swap graph because we require it for some three or more odd - rest even signatures in §3.12:

Definition 9. *A translation of a neighbor-swap graph of signature* $(k_0, \dots, k_x, \dots, k_y, \dots, k_{K-1})$ *with* $k_x = k_y$ *is a neighbor-swap graph of signature* $(k_0, \dots, k_y, \dots, k_x, \dots, k_{K-1})$ *with the same color frequencies. In the resulting neighbor-swap graph, all permutations have the colors* x *and* y *interchanged, which guarantees a different set of edges than before.*

We will generate neighbor-swap graphs such that the colors are in non-increasing order of their frequencies. In case that two colors have the same frequency in a signature, they will be sorted on the color (i.e. the index of the color).

3.2 $(\textit{Even}, 2, 1)$

As is explained by Verhoeff [24], $(\textit{even}, 2, 1)$ is a special case of finding a Hamiltonian cycle on the non-stutter permutations. We give a summary of how Verhoeff generates the Hamiltonian cycle for the neighbor-swap graph of this signature in his work. The neighbor-swap graph is split into parts where the single trailing element is fixed. Assume the signature is $(\textit{even}, 2, 1)$. We have three cases based on the color of the trailing element:

- **Trailing** 0**:** This results in subsignature $(\textit{odd}, 2, 1)$ which contains a Hamiltonian cycle as shown in §3.3.2. However, we want to combine this with the Hamiltonian path of subsignature $(\textit{even}, 1, 1)$. So we will show in §3.3.1 that there exists a Hamiltonian path from $a0 = 120^{k_0-1}10$ to $b0 = 0210^{k_0-2}10$. The trailing element 0 is fixed and is not allowed to move to other positions of the permutations in this path.
- **Trailing** 1**:** This results in a subgraph with signature $(\textit{even}, 1, 1)$. We will show in §3.4 that there exists a Hamiltonian path from $c1 = 120^{k_0}1$ to $d1 = 0210^{k_0-1}1$. The trailing element 1 can not be moved to other positions of the permutations in this path.

- **Trailing** 2**:** This leads to binary subsignature $(\textit{even}, 2)$. This admits a Hamiltonian cycle on the non-stutter permutations by Theorem 1.

As we will show in the upcoming sections, the signatures $(\textit{odd}, 2, 1)$ with trailing 0 and $(\textit{even}, 1, 1)$ with trailing 1 will together form a Hamiltonian cycle. These cases have Hamiltonian paths from $a0 \sim b0$ and $c1 \sim d1$. These paths can be combined into a cycle:

$$\begin{aligned} a0 = 120^{k_0-1}\underline{10} &\sim 120^{k_0}1 = c1 \\ b0 = 0210^{k_0-2}\underline{10} &\sim 0210^{k_0-1}1 = d1 \end{aligned} \tag{3.1}$$

So we obtain a disjoint cycle cover in the neighbor-swap graph of non-stutter permutations of signature $(\textit{even}, 2, 1)$. An example of this case is visualized in Fig. 3.5 on the left. Appendix B Fig. B.2 shows the construction of the paths (3.1) for signature $(6, 2, 1)$.

3.2.1 Connecting the subcycles

What Verhoeff does not specify, is which edge can be used to combine this disjoint cycle cover into a Hamiltonian cycle on the non-stutter permutations. In [24], it is only mentioned that this can be done. Therefore we select an edge that combines the cycle with trailing 2 with the combined cycle with trailing 0 and 1:

$$10^{k_0}12 \sim \underline{01}0^{k_0-1}12 \tag{3.2}$$

$$10^{k_0}21 \sim \underline{01}0^{k_0-1}21 \tag{3.3}$$

This edge does not cause any trouble when generating neighbor-swap graphs for signatures of longer permutations. The left node in edge (3.2) has degree two in the $(\textit{even}, 2)$ subsignature Hamiltonian cycle. Trivially, any node with degree two that is part of a Hamiltonian cycle must have both its edges present in the Hamiltonian cycle. Moreover, edge (3.3) is generated in the $(\textit{even}, 1, 1)$ path. This subsignature guarantees the edge by (3.27).

3.3 $(\textbf{\textit{Odd}}, 2, 1)$

As explained in § 3.2 and by Verhoeff [24], we sometimes need a Hamiltonian path in the $(\textit{odd}, 2, 1)$ neighbor-swap graph to generate a cycle in the $(\textit{even}, 2, 1)$ signature neighbor-swap graphs. However, in general, we require a Hamiltonian cycle for the $(\textit{odd}, 2, 1)$ signature. Therefore this section shows how both of these options can be generated. Moreover, the Hamiltonian path and cycle use distinct cross edges that combine the subcycles where a set of trailing elements of the permutations are fixed.

3.3.1 Hamiltonian Path for $(\textbf{\textit{Odd}}, 2, 1)$

From §3.2, we know that a path in the $(\textit{odd}, 2, 1)$ signature must be between $a = 120^{k_0}1 \overset{*}{\sim} b = 0210^{k_0-1}1$. That path is required to combine this signature with an $(\textit{even}, 1, 1)$ signature path in the $(\textit{even}, 2, 1)$ signature. To do this, we split the graph based on the last two elements. We obtain the cases:

1. **Trailing** 00**:** This yields subsignature $(\textit{odd}, 2, 1)$ which contains a Hamiltonian path from $a' = 120^{k_0-2}1$ to $b' = 0210^{k_0-3}1$. The front-left of Fig. 3.3 shows this part.
2. **Trailing** 20 **and** 02**:** These are two isomorphic and parallel binary subgraphs that admit Hamiltonian cycles on the non-stutter permutations. However, subsignature $(\textit{even}, 2)$ contains stutter permutations. Using Lemma 2 the stutter permutations can be incorporated into a zig-zag cycle (Definition 6). The top-left of Fig. 3.3 shows this part.

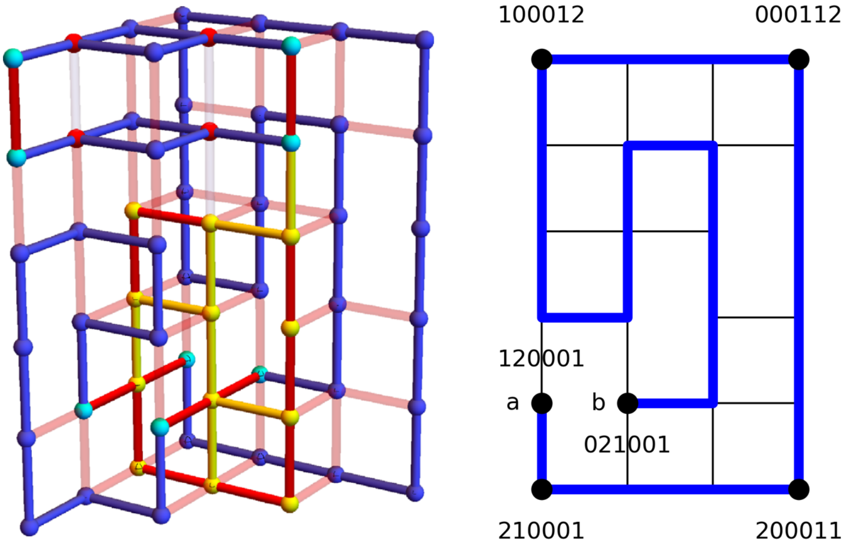


Figure 3.3: On the left the neighbor-swap graph of signature $(3, 2, 1)$ split according to trailing symbols. On the right $(3, 1, 1)$ with trailing 1 and $(3, 1)$ with trailing 12 merged with special vertices marked. Figure from Verhoeff [24].

3. **Trailing** 10**:** This $(\textit{even}, 1, 1)$ subsignature contains a Hamiltonian path between $c' = 120^{k_0-1} \overset{*}{\sim} d' = 0210^{k_0-1}$ to complete a cycle with a' and b' (addressed in §3.4). The bottom center of Fig. 3.3 shows this case.

4. **Trailing** 1 **and** 12**:** This is the part that creates the $a \sim b$ path. Fig. 3.3 (right) contains the path using solely the part with trailing 1 and 12. This can be extended using the ˥-shape described by Verhoeff [24] (Appendix B Fig. B.2a). The left of Fig. 3.3 shows the whole graph with the path from a to b in the back right.

The base case for this Hamiltonian path in the $(\textit{odd}, 2, 1)$ signature neighbor-swap graph is a $(1, 2, 1)$ signature. This is generated by an extended cycle of Lemma 1 such as in Fig. 2.2a. The path starts at 1201 and ends at 0211, which corresponds to the $a \overset{*}{\sim} b$ path for $k_0 = 1$. This Hamiltonian path can be found in Fig. B.1.

3.3.1.1 Combining the path

The four parts are combined into a Hamiltonian path between $a \overset{*}{\sim} b$. Similar to the $(\textit{even}, 2, 1)$ signature, the Hamiltonian paths with trailing 10 and trailing 00 can be combined into a Hamiltonian cycle. This is done by gluing the paths $a'00 \overset{*}{\sim} b'00$ and $c'10 \overset{*}{\sim} d'10$. This cycle will be combined by gluing parallel edges:

$$210^{k_0}1 \sim \underline{12}0^{k_0}1 \tag{3.4}$$
$$210^{k_0-1}10 \sim \underline{12}0^{k_0-1}10 \tag{3.5}$$

Edge (3.4) is generated in the path with trailing 1 and 12 that is shown in Fig. 3.3. This edge corresponds to node a to the node below. Edge (3.5) is generated by the $(\textit{even}, 1, 1)$ subsignature with trailing 10. This is the edge that is guaranteed by (3.29).

$$10^{k_0-1}102 \sim \underline{01}0^{k_0-2}102 \tag{3.6}$$
$$10^{k_0-1}012 \sim \underline{01}0^{k_0-2}012 \tag{3.7}$$

Edge (3.6) is generated in the zig-zag cycle. By Lemma 2, we can select a specific edge to occur in this zig-zag cycle. Moreover, the permutation $10^{k_0}1$ has only two neighbors in the $(\mathit{even}, 2)$ subcycle; $010^{k_0-1}1 \sim 10^{k_0}1 \sim 10^{k_0-1}10$. This means that we can select edge $010^{k_0-1}102 \sim 10^{k_0}102$ to be present in the zig-zag cycle. Edge (3.7) is generated in the newly generated path with trailing 1 and 12. This edge can be found in Fig. 3.3 in the top-left and taking the edge to the right.

3.3.2 Hamiltonian Cycle for $(\mathit{Odd}, 2, 1)$

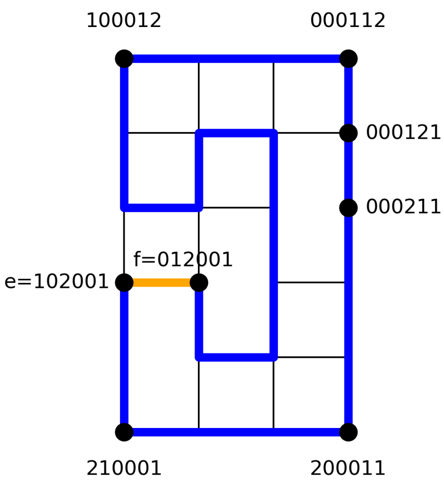
This proof works similarly to the Hamiltonian path in the neighbor-swap graph of the $(\mathit{odd}, 2, 1)$ signature. Here, we have a base case of $k_0 \geq 3$ because there does not exist a Hamiltonian cycle for the signature $(1, 2, 1)$. The generation of the Hamiltonian cycle is split into four parts again;

1. **Trailing 10:** A path in signature $(\mathit{even}, 1, 1)$ from $c'10 \overset{*}{\sim} d'10$ that we combine with $a'00 \overset{*}{\sim} b'00$ below to obtain a Hamiltonian cycle.

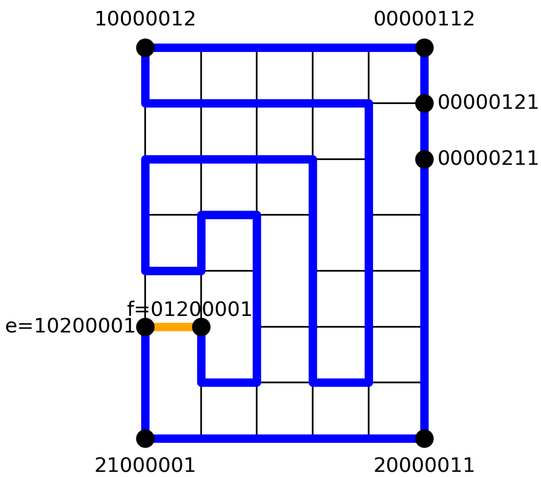
2. **Trailing 00:** A path from $a'00 \overset{*}{\sim} b'00$. This $(\mathit{odd}, 2, 1)$ subsignature is combined with the $c'10 \overset{*}{\sim} d'10$ above to obtain a Hamiltonian cycle.

3. **Trailing 02 and 20:** The isomorphic and parallel subcycles with trailing $02/20$. That can be combined into a cycle with its stutters incorporated using Lemma 2 into a zig-zag cycle (Definition 6).

4. **Trailing 1 and 12:** The cycle with trailing 1 and trailing 12. Instead of generating a path between $a = 120^{k_0}1$ and $b = 0210^{k_0-1}1$, we will generate a cycle "between" $e = 1020^{k_0-1}1$ and $f = 0120^{k_0-1}1$. This cycle is shown in Fig. 3.4a. This figure can be extended similarly to the Hamiltonian path in the neighbor-swap graph of the $(\mathit{odd}, 2, 1)$ signature with the ˥-shape (Fig. 3.4b).



(a) A cycle on $(3, 1, 1)$ with trailing 1 and $(3, 1)$ with trailing 12 merged, with special vertices marked.



(b) A cycle on $(5, 1, 2)$ with trailing 1 and $(5, 1)$ with trailing 12 merged, with special vertices marked.

Figure 3.4: One of the subcycles in the neighbor-swap graph of signature $(\mathit{odd}, 2, 1)$.

3.3.2.1 Combining the cycles

We combine the four parts into a Hamiltonian cycle. The Hamiltonian paths with trailing 10 and trailing 00 are combined into a Hamiltonian cycle on the neighbor-swap graph of the signature $(\textit{even}, 2, 1)$. This is done by gluing the paths $a'00 \overset{*}{\sim} b'00$ and $c'10 \overset{*}{\sim} d'10$. This is the same as in §3.3.1.1 for the Hamiltonian path. We glue the other subcycles with the parallel edges:

$$0210^{k_0-1}1 \sim 0\underline{12}0^{k_0-1}1 \tag{3.8}$$
$$0210^{k_0-2}10 \sim 0\underline{12}0^{k_0-2}10 \tag{3.9}$$

Edge (3.8) is an edge in the path between $e \overset{*}{\sim} f$. The edge is shown in Fig. 3.4 as the edge from f to the node below. Edge (3.9) is generated in a neighbor-swap graph of subsignature $(\textit{even}, 1, 1)$ in the $c' \overset{*}{\sim} d'$ path. The edge is guaranteed by (3.30).

We complete the cycle with cross edges between (3.10) and (3.11):

$$0^{k_0-1}1120 \sim 0^{k_0-2}\underline{10}120 \tag{3.10}$$
$$0^{k_0-1}1210 \sim 0^{k_0-2}\underline{10}210 \tag{3.11}$$

By Definition 6, the zig-zag cycle must cover edge (3.10). Edge (3.11) is generated in subsignature $(\textit{even}, 1, 1)$ and is guaranteed by (3.31).

So we have shown that we can generate a Hamiltonian cycle in the neighbor-swap graph of permutations with signature $(\textit{odd}, 2, 1)$.

3.3.2.2 Guaranteed edges

$$10^{k_0}12 \sim \underline{01}0^{k_0-1}12 \overset{*}{\sim} 0^210^{k_0-2}12 \sim 0^310^{k_0-3}12 \overset{*}{\sim} 0^{k_0-1}\underline{10}12 \sim 0^{k_0}112 \tag{3.12}$$

These edges are shown in the top row of Fig. 3.4. The edges could be split in the middle but we know they must be present in disjoint adjacent pairs.

$$0^{k_0-1}1120 \sim 0^{k_0-1}1\underline{21}0 \tag{3.13}$$

This edge is used as one of the cross edges between (3.10) and (3.11). Therefore the edge is guaranteed in an $(\textit{odd}, 2, 1)$ cycle.

$$120^{k_0}1 \sim \underline{21}0^{k_0}1 \tag{3.14}$$

This edge can be found in the bottom-left of Fig. 3.4 to the node above.

$$1020^{k_0-1}1 \sim 1\underline{20}0^{k_0-1}1 \tag{3.15}$$

This edge can be found in Fig. 3.4 from node e to the node below.

$$1020^{k_0-1}1 \sim \underline{01}20^{k_0-1}1 \tag{3.16}$$

This edge can be found in Fig. 3.4 from node e to node f.

$$0^{k_0}211 \sim 0^{k_0-1}\underline{20}11 \tag{3.17}$$

This edge can be found on the right of Fig. 3.4. Specifically, the edge from the node labeled $0^{k_0}211$ to the node below.

$$2110^{k_0} \sim \underline{12}10^{k_0} \tag{3.18}$$

Node 2110^{k_0} has degree two and the $(\textit{odd}, 2, 1)$ signature contains a Hamiltonian cycle. Therefore both the edges $2\underline{10}10^{k_0-1} \sim 2110^{k_0} \sim \underline{12}10^{k_0}$ must be part of this cycle.

$$20^{k_0-1}110 \sim \underline{02}0^{k_0-2}110 \tag{3.19}$$

This edge originates in the $(\textit{even}, 1, 1)$ signature with trailing 10. The edge is guaranteed as (3.28) in that neighbor-swap graph. It is not used as a cross edge in the $(\textit{odd}, 2, 1)$ signature so it remains intact.

$$0^{k_0}112 \sim 0^{k_0}1\underline{21} \tag{3.20}$$

This edge can be seen in Fig. 3.4 from the node at the top-right to the node below.

$$0^{k_0}121 \sim 0^{k_0}\underline{21}1 \tag{3.21}$$

This edge can be seen in Fig. 3.4 on the right. The nodes are labeled and the edge must clearly be present in our construction.

$$0^{k_0-1}1120 \sim 0^{k_0-1}11\underline{02} \tag{3.22}$$
$$110^{k_0}2 \sim 110^{k_0-1}\underline{20} \tag{3.23}$$

Edges (3.22) and (3.23) are generated in the zig-zag cycle with trailing $02/20$. They are not used as cross edges and must be present in the resulting neighbor-swap graph.

$$110^{k_0}2 \sim 1\underline{01}0^{k_0-1}2 \tag{3.24}$$

Edge (3.24) is generated in the zig-zag cycle with trailing $20/02$.

$$0^{k_0-1}1201 \sim 0^{k_0-1}\underline{21}01 \tag{3.25}$$

This corresponds to the edge in Fig. 3.4 on the second to the right column. This edge is the node to the left of $0^{k_0}211$ to the node below that. The technique of extending the path with the ˥-shape guarantees the edge in the Hamiltonian cycle of signature $(\textit{odd}, 2, 1)$.

$$1120^{k_0} \sim 11\underline{02}0^{k_0-1} \tag{3.26}$$

Edge (3.26) is guaranteed because the first node of the edge has degree two and is part of the Hamiltonian cycle.

3.4 $(\textit{Even}, 1, 1)$

As mentioned in §3.2, we require a path from $c = 120^{k_0}$ to $d = 0210^{k_0}$. For this, we rely on the explanation by Verhoeff [24]. Fig. 3.5 (right) shows how this path is constructed for signature $(4, 1, 1)$. If k_0 is incremented by two, two lines are added to the right and top. These can be added to the path as a "180° rotated L" . Appendix B Fig. B.2b shows how the ˥ is attached to Fig. 3.5 on the right. Note that the signature $(2, 1, 1)$ also can give us this path between $c \overset{*}{\sim} d$ by removing the ˥ from Fig. 3.5.

3.4.1 Guaranteed edges

$$10^{k_0}2 \sim \underline{01}0^{k_0-1}2 \tag{3.27}$$

This edge can be seen in Fig. 3.5 from the top-left node to the node on the right.

$$20^{k_0}1 \sim \underline{02}0^{k_0-1}1 \tag{3.28}$$

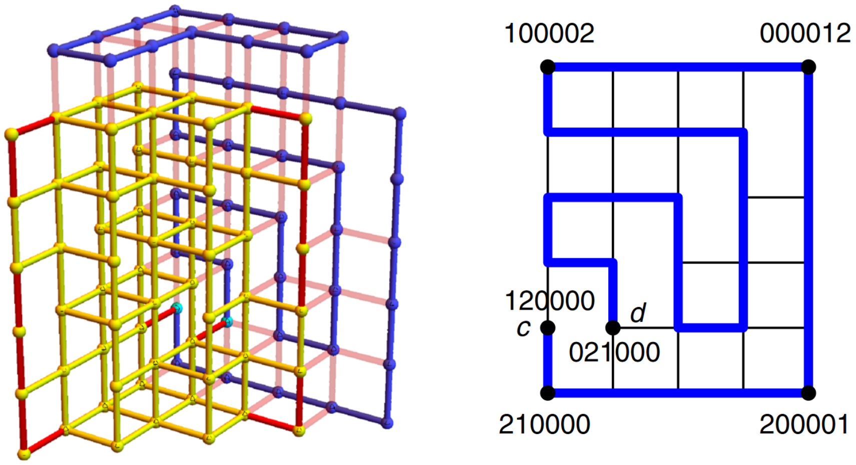


Figure 3.5: On the left the neighbor-swap graph of non-stutter permutations of signature $(4, 2, 1)$ split according to the trailing symbol. Signature $(4, 2)$ with trailing 2 at the top (blue), signature $(3, 1, 1)$ with trailing 1 at the back right (blue), signature $(3, 2, 1)$ with trailing 0 in the front left (red & yellow, see Fig. 3.3); on the right signature $(4, 1, 1)$ with trailing 1 or 12 with special vertices marked. Figure from Verhoeff [24].

This edge can be seen in Fig. 3.5 from the bottom-right node to the node above.

$$120^{k_0} \sim \underline{21}0^{k_0} \tag{3.29}$$

This edge can be seen in Fig. 3.5 from the node c to the node below.

$$0120^{k_0-1} \sim 0\underline{21}0^{k_0-1} \tag{3.30}$$

This edge can be seen in Fig. 3.5 from the node d to the node above.

$$0^{k_0}12 \sim 0^{k_0-1}\underline{10}2 \tag{3.31}$$

This edge can be seen in Fig. 3.5 in the top-right to the node on the left.

3.5 $(\textbf{\textit{Odd}}, \textbf{\textit{odd}}, 1)$

The $(odd, odd, 1)$ signature results in a unique case of generating a Hamiltonian cycle in a neighbor-swap graph. We fix a single trailing element and try to connect the subgraphs into a Hamiltonian cycle. However, subsignature (odd, odd) generates a neighbor-swap graph that only admits a Hamiltonian path and not a Hamiltonian cycle. If one or both of the values of *odd* are 1, we use Lemma 1 for $(odd, 1, 1)$ signatures. Thus we are proving that $(odd, odd, 1)$ signatures contain a Hamiltonian cycle when both $k_0 \geq 3$ and $k_1 \geq 3$. The subgraphs are;

1. **Trailing** 0**:** This $(even, odd, 1)$ subsignature admits a Hamiltonian cycle as shown in § 3.6. When the even color occurs twice, the signature becomes $(odd, 2, 1)$ from §3.3.2.
2. **Trailing** 1**:** This subsignature $(odd, even, 1)$ is isomorphic to $(even, odd, 1)$ signatures. Again, in §3.6 or §3.3.2 we show that the subgraph admits a Hamiltonian cycle.
3. **Trailing** 2**:** This (odd, odd) subsignature is split into three subcases:
    (a) **Trailing** 102 **and** 012**:** These two $(even, even)$ subgraphs are isomorphic and parallel. The $(even, even)$ subcycles are on the non-stutter permutations but Lemma 2 provides a single cycle including the stutter permutations. Specifically, we apply Definition 6 to obtain a zig-zag cycle.

(b) **Trailing** 002**:** This subgraph has signature (*odd*, *odd*). Below we will explain how we can combine this path with the cycle with trailing 0.

(c) **Trailing** 112**:** This (*odd*, *odd*) subgraph can be combined with the (*odd*, *even*, 1) subgraph with trailing 1. The technique is the same as the subgraph with trailing 002.

For the subgraph with trailing 002, we look at the construction of the (*even*, *odd*, 1) subsignature cycle with trailing 0. Similarly for the part with trailing 112 with the (*odd*, *even*, 1) subsignature cycle with trailing 1. We will explain how the graph with a trailing 0 is combined with the trailing 002 graph. These subgraphs have signatures (*even*, *odd*, 1) and (*odd*, *odd*) respectively. The proof is the same for the cycle with trailing 1 combined with the trailing 112 path.

We first address the base case with subsignature $(2, \mathit{odd}, 1)$ with trailing 0 and $(1, \mathit{odd})$ with trailing 002. This is generated as an (isomorphic) $(\mathit{odd}, 2, 1)$ cycle in Fig. 3.4. This subcycle covers a linear path between $01^{k_1}020 \overset{*}{\sim} 1^{k_1}0020$ that is guaranteed by (3.12). Now we will create a "side-step path" (Definition 10) between this linear path and the linear path of the $(1, \mathit{odd})$ subsignature with trailing 002:

$$\begin{array}{ccccccccccc}
01^{k_1}0\underline{02} & \sim & \underline{10}1^{k_1-1}002 & & 1^2 01^{k_1-2}0\underline{02} & \sim & 1^2\underline{10}1^{k_1-3}002 & & 1^{k_1-1}010\underline{02} & \sim & 1^{k_1-1}\underline{10}002 \\
\wr & & \wr & & \wr & & \wr & \overset{*}{\sim} & \wr & & \wr \\
01^{k_1}020 & & 101^{k_1-1}0\underline{20} & \sim & 1\underline{10}1^{k_1-2}020 & & 1^3 01^{k_1-3}0\underline{20} & & 1^{k_1-1}01020 & & 1^{k_1}00\underline{20}
\end{array} \tag{3.32}$$

The number of permutations in the path with trailing 002 is the same as the number of permutations with trailing 020. The binomial coefficients of the subsignatures are the same. Moreover, there are no stutter permutations so they are not considered in this count.

In the top row of (3.32), we have all the permutations of the Hamiltonian path of the $(1, \mathit{odd})$ neighbor-swap graph with the trailing 002. The bottom row of (3.32) contains the path between $01^{k_1}020 \overset{*}{\sim} 1^{k_1}0020$. Edge $1^{k_1-2}01^2020 \sim 1^{k_1-2}\underline{10}1020$ is not required to be direct but can also be a path. It can still be used to side-step to the path with the trailing 002 from the path with trailing 020. Therefore we can incorporate the $(1, \mathit{odd})$ signature with trailing 002 paths into a Hamiltonian cycle with $(2, \mathit{odd}, 1)$ trailing 020. We can do the same for $(\mathit{odd}, 1)$ (with trailing 112) and $(\mathit{odd}, 2, 1)$ (with trailing 121).

We can extend this idea to hold when the value of *even* is ≥ 4 with Lemma 5:

Lemma 5. *Consider a Hamiltonian cycle in the neighbor-swap graph with signature S. Let P be a subsignature of S and p a fixed sequence with signature P, where $1 \leq |p| < n$. Say p has trailing element y, i.e. $p = p'y$. Let the remaining colors in S that are not in P form the subsignature $R = S - P$.*
Now, we add a trailing element x to the Hamiltonian cycle of signature S, where $x \neq y$. If all permutations with trailing sequence px in this Hamiltonian cycle are organized as disjoint adjacent pairs, then it is possible to incorporate all nodes with signature R and trailing $p'xy$ into the Hamiltonian cycle.

Proof of Lemma 5: We consider one adjacent pair of nodes in the Hamiltonian cycle of the neighbor-swap graph with signature $S = (k_0, \ldots, k_x, \ldots, k_{K-1})$. Say the pair is $ap'y \sim bp'y$. We know that a and b are adjacent by a neighbor swap and both a and b have the signature R. We can generate an adjacent pair of nodes $ap'xy \sim bp'xy$. This pair of nodes can be incorporated into the Hamiltonian cycle on signature S where every permutation has a trailing x. Thus the Hamiltonian cycle is a subcycle on the signature $(k_0, \ldots, k_x + 1, \ldots, k_{K-1})$. This pair of nodes can be incorporated by forming the "side-step path" (Definition 10):

$$ap'yx \sim ap'\underline{xy} \sim \underline{b}p'xy \sim bp'\underline{yx}. \tag{3.33}$$

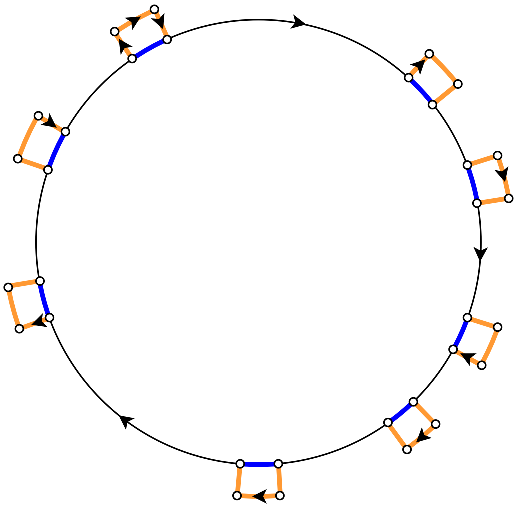


Figure 3.6: A visualization of Lemma 5. The blue edges are in the original signature and the orange edges form the side-step path from Definition 10. The blue edges are removed by applying Lemma 5.

Such a path can only be formed if there is an edge between a and b, not if there is an indirect path between the two nodes. The latter requires us to visit $ap'yx$ and $bp'yx$ twice, which does not result in a Hamiltonian cycle on the merged neighbor-swap graphs.

The number of nodes in the neighbor-swap graph of signature R with trailing elements $p'xy$ and the signature R with trailing elements $p'yx$ are equal. Therefore both the multinomial coefficients are $M(R)$, i.e. the number of permutations is the same. So we can conclude that if we have disjoint adjacent pairs, we can form paths similar to (3.33) without missing any node. This is visualized in Fig. 3.6. □

Lemma 5 clarifies that we need direct edges in the (*even*, *odd*, 1) and isomorphic (*odd*, *even*, 1) graphs. The order of the 020 edges is irrelevant because the 002 path contains the same permutations before the three trailing elements. We define a path that is generated by Lemma 5:

Definition 10. *A "side-step path" for signature S is a path that is constructed using Lemma 5. The edges that are part of the side-step path are those between*

$$ap'yx \sim ap'\underline{xy} \sim \underline{b}p'xy \sim bp'\underline{yx}. \tag{3.34}$$

where x and y are elements and a and b are permutations with signature R and $p'y$ has signature P. The signatures correspond the the definitions in Lemma 5.

$$\begin{array}{ccccccc} 1^{k_1}0^{k_0-2}0\underline{02} & \sim & 1^{k_1-1}\underline{01}0^{k_0-3}002 & & 0^{k_0-3}101^{k_1-1}0\underline{02} & \sim & 0^{k_0-3}\underline{01}1^{k_1-1}002 \\ \wr & & \wr & & \wr & & \wr \\ 1^{k_1}0^{k_0-2}020 & & 1^{k_1-1}010^{k_0-3}0\underline{20} & \overset{*}{\sim} & 0^{k_0-3}101^{k_1-1}020 & & 0^{k_0-2}1^{k_1}0\underline{20} \end{array} \tag{3.35}$$

A generalization of how the side-step path with trailing 002 is constructed is shown in (3.35). As will be explained in §3.6 for (*even*, *odd*, 1) signatures; the nodes that have a trailing 02 in the subsignature (thus 020 in the (*odd*, *odd*, 1) context) will remain to occur in pairs. Moreover, the permutations with trailing 002 in the neighbor-swap graph of the $(odd, odd, 1)$ signature occur in pairs in the Hamiltonian cycle. The same holds for the permutations with trailing 112. Therefore we can conclude that the neighbor-swap graphs of $(odd, odd, 1)$ signatures admit a disjoint cycle cover.

3.5.1 Connecting the subcycles

We will now glue the $(even, odd, 1)$, $(odd, even, 1)$, and combined $(even, even)$ subcycles. These are glued using parallel edges:

$$21^{k_1-1}0^{k_0-1}10 \sim \underline{12}1^{k_1-2}0^{k_0-1}10 \tag{3.36}$$
$$21^{k_1-1}0^{k_0}1 \sim \underline{12}1^{k_1-2}0^{k_0}1 \tag{3.37}$$

Edge (3.36) originates in the $(even, odd, 1)$ subsignature with trailing 0. If $k_0 = k_1 = 3$, we obtain subsignature $(2, 3, 1)$. The edge is guaranteed by (3.19). Otherwise, the edge is guaranteed by (3.67). Edge (3.37) exists by a similar argument. If $k_1 = 3$, the edge is guaranteed as (3.18). If $k_1 \geq 5$, the edge is (3.66).

Then we glue this combined cycle with trailing 0 and 1 with the zig-zag cycle with trailing $102/012$. For this, we use edges:

$$1^{k_1-1}0^{k_0}21 \sim 1^{k_1-2}\underline{01}0^{k_0-1}21 \tag{3.38}$$
$$1^{k_1-1}0^{k_0}12 \sim 1^{k_1-2}\underline{01}0^{k_0-1}12 \tag{3.39}$$

The existence of these nodes is fairly trivial as they originate from stutter permutations. Edge (3.38) originates in the subgraph with signature $(even, odd, 1)$ with trailing 1. In that subcycle, it is guaranteed by (3.70).

Edge (3.39) originates in the zig-zag cycle with trailing $102/012$. By the same argument, the edge must be present in that subgraph. Therefore we have generated a Hamiltonian cycle in the neighbor-swap graph of signature $(odd, odd, 1)$.

3.5.2 Guaranteed edges

$$0^{k_0}21^{k_1} \sim 0^{k_0-1}\underline{20}1^{k_1} \tag{3.40}$$
$$1^{k_1}20^{k_0} \sim 1^{k_1-1}\underline{21}0^{k_0} \tag{3.41}$$

These edges must be part of the Hamiltonian cycle in the $(odd, odd, 1)$ signature since the first nodes have degree two.

$$120^{k_0}1^{k_1-1} \sim 1\underline{02}0^{k_0-1}1^{k_1-1} \tag{3.42}$$
$$021^{k_1}0^{k_0-1} \sim 0\underline{12}1^{k_1-1}0^{k_0-1} \tag{3.43}$$

Edge (3.42) originates in an $(odd, 2, 1)$ signature with trailing 1 where it is guaranteed by (3.15). In graphs where $k_1 > 3$, we have that the edge remains intact in the $(even, odd, 1)$ signatures by (3.65). In this signature, we add a trailing even-occurring trailing element to obtain an $(odd, odd, 1)$ signature with this edge guaranteed. We can iteratively add a trailing 1 to obtain signature $(odd, even, 1)$ and $(odd, odd, 1)$. Edge (3.43) exists by a similar argument.

$$20^{k_0}1^{k_1} \sim \underline{02}0^{k_0-1}1^{k_1} \tag{3.44}$$
$$20^{k_0}1^{k_1} \sim 20^{k_0-1}\underline{10}1^{k_1-1} \tag{3.45}$$

Trivially both edges (3.44) and (3.45) are present in the Hamiltonian cycle on the $(odd, odd, 1)$ signature. This is because the node $20^{k_0}1^{k_1}$ has degree two.

$$21^{k_1}0^{k_0} \sim \underline{12}1^{k_1-1}0^{k_0} \tag{3.46}$$

Similarly, edge (3.46) is guaranteed because the first node has degree two.

$$0^{k_0}1^{k_1-1}21 \sim 0^{k_0}1^{k_1-2}\underline{21}1 \tag{3.47}$$

This edge is generated in the $(odd, even, 1)$ signature with a trailing 1. That signature guarantees the edge by (3.72). Or if the subsignature is $(odd, 2, 1)$ by (3.21).

$$0^{k_0}1^{k_1}2 \sim 0^{k_0-1}\underline{10}1^{k_1-1}2 \tag{3.48}$$

$$0^{k_0}1^{k_1}2 \sim 0^{k_0}1^{k_1-1}\underline{21} \tag{3.49}$$

Both these edges must be present in the $(odd, odd, 1)$ signature cycle because the first node of both edges has degree two.

$$1^{k_1-1}0^{k_0}21 \sim 1^{k_1-1}0^{k_0}\underline{12} \tag{3.50}$$

This edge is one of the cross edges between (3.38) and (3.39).

$$1^{k_1}0^{k_0}2 \sim 1^{k_1-1}\underline{01}0^{k_0-1}2 \tag{3.51}$$

This edge is also guaranteed because the first node has degree two in the Hamiltonian cycle.

3.6 $(\textit{Even}, \textit{odd}, 1)$

Similarly to the $(odd, odd, 1)$ signatures, the $(even, odd, 1)$ signatures have subgraphs that only admit a Hamiltonian path and not a Hamiltonian cycle. We will address this graph as $(even, odd, 1)$. Depending on the values of k_0 and k_1 we might use one different pair of cross edges as explained in this section. This occurs if k_0 odd and k_1 even satisfy $k_0 < k_1$ or if k_0 even and k_1 odd satisfy $k_0 < k_1$. This is to adhere to the guaranteed edges in two odd - rest even signatures.

If $k_1 = 2$, we have an $(odd, 2, 1)$ signature cycle (Fig. 3.4). If $k_1 = 1$, we have an $(even, 1, 1)$ signature for which we have a Hamiltonian path by Lemma 1. Thus we assume $k_0 \geq 4$ and odd $k_1 \geq 3$. We distinguish the following subgraphs:

1. **Trailing 0:** This results in an $(odd, odd, 1)$ signature of length $n-1$. This neighbor-swap graph admits a Hamiltonian cycle as explained in §3.5.
2. **Trailing 1:** This $(even, even, 1)$ signature is split into three subcases because we have to account for stutter permutations:
   - **Trailing 01:** A Hamiltonian cycle on the neighbor-swap graph of $(odd, even, 1)$ signatures with permutation length $n-2$. We fix the position of one even and one odd element to the suffix to get the signature $(even-1, odd-1, 1)$. This is a cycle on the non-stutter permutations by strong induction on n. The base case is an $(odd, 2, 1)$ subsignature from §3.3.2.
   - **Trailing 11:** A Hamiltonian cycle on the neighbor-swap graph of $(even, odd, 1)$ signatures with permutation length $n-2$.
     If $k_1 \geq 5$, this is a cycle on the non-stutter permutations by strong induction on n. If $k_1 = 3$, we get an $(even, 1, 1)$ signature. Here we use the Hamiltonian cycle from Lemma 1 Case 2(a) and thus leave out $x_1 11 \sim x_2 11 = 120^{k_0}11 \sim 210^{k_0}11$. These nodes are added between $120^{k_0-2}101 \sim \underline{21}0^{k_0-2}101$ in the cycle with trailing 01. This edge is guaranteed in the $(odd, 2, 1)$ neighbor-swap graph by (3.14).
   - **Trailing 21:** This results in a Hamiltonian cycle on the neighbor-swap graph of signature $(even, odd-1)$. However, we require that the stutter permutations are incorporated in this case so we will combine this into a zig-zag cycle with the trailing 12 part.

3. **Trailing 2:** This (*even*, *odd*) signature is split into two subcases:
   - **Trailing 12:** This results in a Hamiltonian cycle on the neighbor-swap graph of signature ($even, odd-1$). However, we require that the stutter permutations are incorporated in this case so we will combine this into a zig-zag cycle with the trailing 21 part.
   - **Trailing 02:** This is a binary (*odd*, *odd*) path. We will explain below how this part is integrated into the ($odd, odd, 1$) cycle with a trailing 0.

The (*odd*, *odd*) signature contains a Hamiltonian path by Theorem 1 and we add trailing elements 02 to every permutation. We combine this path with the ($odd, odd, 1$) subsignature with trailing 0. This is similar to how we did in §3.5 between the ($odd, even, 1$) subcycle with trailing 020 and the (*odd*, *odd*) path with trailing 002 using Lemma 5.

We have shown the ($odd, odd, 1$) subsignature cycle covers all nodes that end with 002 (and 112) as disjoint adjacent pairs for neighbor-swap graphs on permutations of length $n-1$ in §3.5. Furthermore, the (*even*, *even*) subcycles with trailing 102 and 012 in the ($odd, odd, 1$) context contain an even number of nodes. The number of nodes is

$$2 \cdot M(\textit{signature of the trailing } 102 \textit{ cycle}) = 2 \cdot M(\textit{signature of the trailing } 012 \textit{ cycle})$$

because the number of nodes in both the neighbor-swap graphs with trailing 102 and 012 is equal. This cycle is only cut on one position to form a single Hamiltonian cycle with the other subcycles for the ($odd, odd, 1$) subsignature. This is done with edge (3.39). This does not split an adjacent pair of nodes in the trailing 2 subgraph because it is a cycle of even length that is cut in a single position. If we would split it more than once, we would have to consider the distance between the cuts. However, cutting the cycle on a single position still gives a Hamiltonian path of even length within the Hamiltonian cycle.

Therefore we can conclude that all the nodes that end with 2 in an ($odd, odd, 1$) subsignature occur in disjoint adjacent pairs. In the ($even, odd, 1$) signature, we add a trailing 0 to all of these nodes to get the disjoint adjacent pairs with trailing 20. Therefore we have all the requirements of Lemma 5. So we conclude that we can incorporate all nodes with trailing 02 into the ($odd, odd, 1$) subcycle.

$$\begin{array}{ccccccc}
1^{k_1}0^{k_0-1}\underline{02} & \sim & 1^{k_1-1}\underline{01}0^{k_0-2}02 & & 0^{k_0-2}101^{k_1-1}\underline{02} & \sim & 0^{k_0-2}\underline{01}1^{k_1-1}02 \\
\wr & & \wr & & \wr & & \wr \\
1^{k_1}0^{k_0-1}20 & & 1^{k_1-1}010^{k_0-2}\underline{20} & \overset{*}{\sim} & 0^{k_0-2}101^{k_1-1}20 & & 0^{k_0-1}1^{k_1}\underline{20}
\end{array} \tag{3.52}$$

We show in (3.52) that this results in a path where all permutations that end with 02 are disjoint adjacent pairs. This is also required for the ($odd, odd, 1$) signatures in §3.5 to incorporate all the 002 nodes into an ($even, odd, 1$) subsignature with trailing 0. This edge can be found in (3.53), where both $k_0 \geq 3$ and $k_1 \geq 3$ are odd. Thus we have shown that the (*odd*, *odd*) subsignature can be combined with a Hamiltonian cycle on the ($odd, odd, 1$) subsignature into one cycle.

$$\begin{array}{ccccccc}
1^{k_1}0^{k_0-2}0\underline{02} & \sim & 1^{k_1-1}\underline{01}0^{k_0-3}002 & & 0^{k_0-3}101^{k_1-1}0\underline{02} & \sim & 0^{k_0-3}\underline{01}1^{k_1-1}002 \\
\wr & & \wr & & \wr & & \wr \\
1^{k_1}0^{k_0-2}\underline{02}0 & & 1^{k_1-1}010^{k_0-3}0\underline{20} & & 0^{k_0-3}101^{k_1-1}\underline{02}0 & & 0^{k_0-2}1^{k_1}0\underline{20} \\
\wr & & \wr & & \wr & & \wr \\
1^{k_1}0^{k_0-2}200 & & 1^{k_1-1}010^{k_0-3}\underline{20}0 & \overset{*}{\sim} & 0^{k_0-3}101^{k_1-1}200 & & 0^{k_0-2}1^{k_1}\underline{20}0
\end{array} \tag{3.53}$$

3.6.1 Connecting the subcycles

We glue subcycles into one Hamiltonian cycle. For this, we use the parallel edges below. k_0 remains the even color and k_1 the odd color.

$$120^{k_0-1}1^{k_1-1}01 \sim 1\underline{02}0^{k_0-2}1^{k_1-1}01 \tag{3.54}$$

$$120^{k_0-1}1^{k_1-1}10 \sim 1\underline{02}0^{k_0-2}1^{k_1-1}10. \tag{3.55}$$

Edge (3.54) has trailing 01, thus it is part of the $(odd, even, 1)$ subcycle. The edge is guaranteed by (3.65). Edge (3.55) also exists by (3.42) if $k_0 > k_1$ and (3.43) if $k_1 > k_0$.

$$0^{k_0-1}21^{k_1-2}101 \sim 0^{k_0-2}\underline{201}^{k_1-2}101 \tag{3.56}$$

$$0^{k_0-1}21^{k_1-2}011 \sim 0^{k_0-2}\underline{201}^{k_1-2}011 \tag{3.57}$$

Edge (3.56) has two different origins; both with signature $(even-1, odd-1, 1)$. This is either an $(odd, 2, 1)$ signature where it is edge (3.17). Otherwise, edge (3.56) is constructed in an $(even-1, odd-1, 1) = (odd, even, 1)$ cycle. There it is guaranteed as edge (3.62). The existence of edge (3.57) originates in subcycle $(even, odd-2, 1)$ cycle with trailing 11. If the subsignature is $(even, 1, 1)$ the edge is part of the d_1 edge (2.2) of Lemma 1. If $k_1 \geq 3$, it is guaranteed by (3.64).
For the last pair of parallel edges, we choose either of the two pairs below depending on the values of k_0 and k_1. We discuss in Appendix D.1 what happens if k_0 is odd and k_1 is even. If k_0 is even and k_1 odd with $k_0 > k_1$, the parallel edges (3.58) and (3.59) are used. If $k_0 < k_1$, the cycles are combined by gluing (3.60) and (3.61). This is because we have to guarantee edge (3.164) in the two odd - rest even signatures.

$$0^{k_0}1^{k_1-1}21 \sim 0^{k_0-1}\underline{101}^{k_1-2}21 \tag{3.58}$$

$$0^{k_0}1^{k_1-2}211 \sim 0^{k_0-1}\underline{101}^{k_1-3}211 \tag{3.59}$$

Edge (3.58) is generated in the zig-zag cycle with trailing $12/21$. By definition, the zig-zag cycle contains this edge. Edge (3.59) is generated in the zig-zag cycle of a smaller signature of the $(even, odd, 1)$ signature. There, it is part of the 12 part of the zig-zag cycle instead of the 21 part. If $k_1 = 3$, the edge is part of d_0 of (2.2) in Lemma 1.

$$1^{k_1-1}0^{k_0-1}021 \sim 1^{k_1-2}\underline{010}^{k_0-2}021 \tag{3.60}$$

$$1^{k_1-1}0^{k_0-1}201 \sim 1^{k_1-2}\underline{010}^{k_0-2}201 \tag{3.61}$$

In the zig-zag cycle with trailing $21/12$, edge (3.60) is guaranteed because $1^{k_1-1}0^{k_0}21$ is a stutter permutation in the $(even, even)$ context. So the zig-zag cycle contains the parallel edge. The existence of edge (3.61) follows from the $(odd, even, 1)$ subsignature cycle with trailing 01. In this graph node $1^{k_1-1}0^{k_0-1}2$ is a stutter permutation in the zig-zag cycle with trailing $02/20$.

Therefore we can combine all subcycles in signature $(even, odd, 1)$ into a single Hamiltonian cycle.

3.6.2 Guaranteed edges

In the edges of this section, we again have that k_0 is even and k_1 is odd. Thus we have signature $(even, odd, 1)$.

$$1^{k_1}20^{k_0} \sim 1^{k_1-1}\underline{210}^{k_0} \tag{3.62}$$

$$0^{k_0}21^{k_1} \sim 0^{k_0-1}\underline{201}^{k_1} \tag{3.63}$$

These edges are guaranteed because the first nodes of the edges have degree two and are part of the Hamiltonian cycle.

$$0^{k_0-1}21^{k_1}0 \sim 0^{k_0-2}\underline{201}^{k_1}0 \tag{3.64}$$

Edge (3.64) is guaranteed because it is guaranteed by (3.40) and is not used as a cross edge.

$$021^{k_1}0^{k_0-1} \sim 0\underline{121}^{k_1-1}0^{k_0-1} \tag{3.65}$$

This edge is generated in the $(odd, odd, 1)$ cycle with trailing 0. Within this cycle, it is guaranteed by edge (3.42) and (3.43). This edge is not used as a cross edge so it is guaranteed.

$$20^{k_0}1^{k_1} \sim \underline{02}0^{k_0-1}1^{k_1} \tag{3.66}$$

This edge is generated in the cycle with trailing 11. Within this $(even, odd, 1)$ signature, we fix the trailing 11 again until $k_1 = 1$. This gives the signature $(even, 1, 1)$. The edge is guaranteed as (3.28).

$$21^{k_1-1}0^{k_0}1 \sim \underline{12}1^{k_1-2}0^{k_0}1 \tag{3.67}$$

This edge follows directly from (3.66). The edge swaps the even and odd colors and adds the trailing 01 as is done above to switch from an $(odd, even, 1)$ to an $(even, odd, 1)$ signature.

$$1^{k_1}0^{k_0}2 \sim 1^{k_1-2}\underline{01}0^{k_0-1}2 \tag{3.68}$$

This edge is guaranteed because the first node has degree two. Therefore it must be one of the edges that is generated using Lemma 5 in the side-step path.

$$20^{k_0-1}1^{k_1}0 \sim \underline{02}0^{k_0-2}1^{k_1}0 \tag{3.69}$$

This edge is generated in the subcycle of signature $(odd, odd, 1)$ with trailing 0. It is guaranteed in this subsignature by (3.44). We obtain edge (3.69) by adding trailing element 0.

$$0^{k_0}1^{k_1}2 \sim 0^{k_0}1^{k_1-1}\underline{21} \tag{3.70}$$

$$1^{k_1-1}0^{k_0}12 \sim 1^{k_1-1}0^{k_0}\underline{21} \tag{3.71}$$

Edges (3.70) and (3.71) are generated in the zig-zag cycle with trailing $12/21$. They are not used as cross edges and thus they must be present in the resulting $(even, odd, 1)$ signature.

$$1^{k_1}0^{k_0}2 \sim 1^{k_1}0^{k_0-1}\underline{20} \tag{3.72}$$

Edge (3.72) is guaranteed because the first node of the edge has degree two and is part of the Hamiltonian cycle.

$$0^{k_0}1^{k_1}2 \sim 0^{k_0}\underline{10}1^{k_1-1}2 \tag{3.73}$$

Edge (3.73) is guaranteed because the first node of the edge has degree two and is part of the Hamiltonian cycle on this signature.

Conditionally guaranteed edges:
If $k_0 > k_1$:

$$0^{k_0}1^{k_1-1}21 \sim 0^{k_0}1^{k_1-2}\underline{21}1 \tag{3.74}$$

This edge is used as one of the cross edges between (3.58) and (3.59). Therefore it must be in the Hamiltonian cycle of the $(even, odd, 1)$ signature.

$$1^{k_1-1}0^{k_0}21 \sim 1^{k_1-2}\underline{01}0^{k_0-1}21 \tag{3.75}$$

Edge (3.75) is not used as a cross edge in this case and exists for the same reason as (3.60).

If $k_0 < k_1$:

$$1^{k_1-1}0^{k_0}21 \sim 1^{k_1-1}0^{k_0-1}\underline{20}1 \tag{3.76}$$

This edge is used as one of the cross edges between (3.60) and (3.61).

$$0^{k_0}1^{k_1-1}21 \sim 0^{k_0-1}\underline{10}1^{k_1-2}21 \tag{3.77}$$

Edge (3.77) is not used as a cross edge in this case and exists for the same reason as (3.58).

3.7 $(\textit{Even}, 1, 1, 1)$

The $(\textit{even}, 1, 1, 1)$ signature has three odd-occurring elements. We fix a single trailing element by the technique addressed in § 3.12 for three or more odd - rest even signatures. Some subgraphs are generated by signatures for which the neighbor-swap graph only admits a Hamiltonian path. These are the subsignatures $(\textit{even}, 1, 1)$. We split the graph into the following parts;

1. **Trailing** 0**:** This results in a neighbor-swap graph with signature $(\textit{odd}, 1, 1, 1)$. The base case is signature $(1, 1, 1, 1)$, where the neighbor-swap graph is a permutahedron. The Hamiltonian cycle of this subsignature is generated by three or more odd - rest even signatures (§3.12). We require edges $0120^{k_0-2}3 \sim 0\underline{21}0^{k_0-2}3$, $0130^{k_0-2}2 \sim 0\underline{31}0^{k_0-2}2$, and $0230^{k_0-2}1 \sim 0\underline{32}0^{k_0-2}1$. These are guaranteed by (3.172), (3.173), and (3.174) respectively. Those edges are used to connect two nodes from every $(\textit{even}, 1, 1)$ subsignature to transform the Hamiltonian path into a Hamiltonian cycle.

2. **Trailing** 1**:** This results in subsignature $(\textit{even}, 1, 1)$. We know that this graph does not admit a Hamiltonian cycle. We remove edge $y_1 \sim y_2 = 0230^{k_0-1} \sim 0320^{k_0-1}$ as in Lemma 1 Fig. 2.3a to obtain a Hamiltonian cycle. The two removed nodes are added as an edge between edge (3.174) in the $(\textit{odd}, 1, 1, 1)$ subcycle:

$$0230^{k_0-2}10 \sim 0230^{k_0-2}\underline{01} \sim 0\underline{32}0^{k_0-1}1 \sim 0320^{k_0-2}\underline{10}.$$

3. **Trailing** 2**:** This works similarly to the case above. Since we guaranteed edge $0130^{k_0-2}20 \sim 0\underline{31}0^{k_0-2}20$ in the $(\textit{odd}, 1, 1, 1)$ neighbor-swap graph by (3.173), we remove edge $0130^{k_0-1}2 \sim 0\underline{31}0^{k_0-1}2$ to obtain a Hamiltonian cycle. The nodes are connected in the $(\textit{odd}, 1, 1, 1)$ cycle with edges:

$$0130^{k_0-2}20 \sim 0130^{k_0-2}\underline{02} \sim 0\underline{31}0^{k_0}2 \sim 0310^{k_0-2}\underline{20}.$$

4. **Trailing** 3**:** Again the same strategy is applied. We use the Hamiltonian cycle from Fig. 2.3a without edge $y_1 \sim y_2$. This results in a Hamiltonian cycle in the $(\textit{even}, 1, 1)$ neighbor-swap graph. The two nodes are added in the $(\textit{odd}, 1, 1, 1)$ cycle between (3.172) as edge:

$$0120^{k_0-2}30 \sim 0120^{k_0-2}\underline{03} \sim 0\underline{21}0^{k_0-1}3 \sim 0210^{k_0-2}\underline{30}$$

3.7.1 Connecting the subcycles

The cycles are connected similarly to three or more odd - rest even signatures in § 3.12. We have three cycles with a trailing odd-occurring element and one with a trailing even-occurring element. So we combine the cycles with trailing elements in the order of Algorithm 3. We address the existence of each of these edges.

$$230^{k_0-1}10 \sim \underline{203}0^{k_0-2}10 \tag{3.78}$$
$$230^{k_0}1 \sim \underline{203}0^{k_0-1}1 \tag{3.79}$$

Edge (3.78) is generated by subsignature $(\textit{odd}, 1, 1, 1)$. In this three or more odd - rest even signature, the edge is guaranteed by (3.179). Edge (3.79) is generated in a Hamiltonian cycle of Lemma 1 where the nodes $y_1 \sim y_2$ were removed. So the edge is the one that connects d_{k_0} to d_{k_0-1} of (2.2).

$$0^{k_0}321 \sim 0^{k_0-1}\underline{302}1 \tag{3.80}$$
$$0^{k_0}312 \sim 0^{k_0-1}\underline{301}2 \tag{3.81}$$

Edge (3.80) is generated by subsignature $(\textit{even}, 1, 1)$, similar to edge (3.81). The edges are formed in Lemma 1 by edge d_0 in (2.2).

$$0^{k_0}132 \sim 0^{k_0-1}\underline{10}32 \tag{3.82}$$
$$0^{k_0}123 \sim 0^{k_0-1}\underline{10}23 \tag{3.83}$$

Again we can use the same argument as above for (3.82) and (3.83). The edges are generated in a neighbor-swap graph with signature $(\textit{even}, 1, 1)$. Thus both are formed in Lemma 1 by edge d_0 in (2.2).

So we can conclude that the neighbor-swap graph of signature $(\textit{even}, 1, 1, 1)$ contains a Hamiltonian cycle.

3.7.2 Guaranteed edges

$$xy0^{k_0}z \sim \underline{yx}0^{k_0}z \tag{3.84}$$

We can see that the cross edges between the subcycles in the $(\textit{even}, 1, 1, 1)$ signature are distinct from this edge. This edge is between permutations where the first two elements occur once. This is different from (3.79) because we now swap the first two elements instead of the second and third elements. Moreover, in the subcycle without the trailing z, the node $xy0^{k_0}$ has degree two. Because it is in a Hamiltonian cycle, it must be that edge (3.84) is in the neighbor-swap graph. This shows the existence of the three edges:

$$120^{k_0}3 \sim \underline{21}0^{k_0}3 \qquad 130^{k_0}2 \sim \underline{31}0^{k_0}2 \qquad 320^{k_0}3 \sim \underline{23}0^{k_0}1$$

$$x0^{k_0}yz \sim \underline{0x}0^{k_0-1}yz \tag{3.85}$$

Similar to the above, we show the existence of multiple edges using edge (3.85). Edges of this form are not used as cross edges because we have one element x that occurs once at the start. This is not the case for any cross edge. Therefore we can ways remove the trailing element z to obtain the signature $(\textit{even}, 1, 1)$. The d_0 path in (2.2) of Lemma 1 contains this exact edge. Therefore we can conclude that the following edges are in the $(\textit{even}, 1, 1, 1)$ signature:

$$\begin{array}{ccc} 10^{k_0}23 \sim \underline{01}0^{k_0-1}23 & 20^{k_0}13 \sim \underline{02}0^{k_0-1}13 & 30^{k_0}12 \sim \underline{03}0^{k_0-1}12 \\ 10^{k_0}32 \sim \underline{01}0^{k_0-1}32 & 20^{k_0}31 \sim \underline{02}0^{k_0-1}31 & 30^{k_0}21 \sim \underline{03}0^{k_0-1}21 \end{array}$$

$$10^{k_0}23 \sim 10^{k_0-1}\underline{20}3 \tag{3.86}$$

This edge is generated in the $(\textit{even}, 1, 1)$ subcycle where it glues the d_0 edge to the d_1 edge (2.2) of Lemma 1. The edge remains intact when removing edge $y_1 \sim y_2 = 0230^{k_0-1} \sim 0\underline{32}0^{k_0-1}$.

$$20^{k_0-1}310 \sim \underline{02}0^{k_0-1}310 \tag{3.87}$$

This edge is guaranteed because it is part of the $(\textit{odd}, 1, 1, 1)$ subcycle with trailing 0. It is guaranteed as edge (3.175) in that signature.

$$0^{k_0-1}1230 \sim 0^{k_0-1}1\underline{32}0 \tag{3.88}$$

This edge is generated in the $(\textit{odd}, 1, 1, 1)$ signature where it is guaranteed by (3.176).

$$0^{k_0}123 \sim 0^{k_0}1\underline{32} \tag{3.89}$$

Edge (3.89) is used as the cross edge between (3.82) and (3.83).

$$0^{k_0}321 \sim 0^{k_0}\underline{312} \tag{3.90}$$

Edge (3.90) is the cross edge between (3.80) and (3.81).

$$0^{k_0}xyz \sim 0^{k_0}\underline{yx}z \tag{3.91}$$

This edge is part of the subgraph with trailing z. Subsignature $(\textit{even}, 1, 1)$ guarantees this edge as part of the d_0 edge (2.2) in Lemma 1. It is also not used as a cross edge for the $(\textit{even}, 1, 1, 1)$ signature. Therefore this guarantees edges:

$$0^{k_0}123 \sim 0^{k_0}\underline{21}3 \qquad 0^{k_0}132 \sim 0^{k_0}\underline{31}2 \qquad 0^{k_0}231 \sim 0^{k_0}\underline{32}1$$

3.8 $(\textbf{\textit{Even}}, 2, 1, 1)$

For the neighbor-swap graphs of two odd - rest even signatures, we apply a technique explained in §3.11. In short, we fix a single trailing element, except when that element occurs an odd number of times, in which case we fix two elements. This is to account for the stutter permutations that are in the subgraphs. So when we fix the elements that occur once in this subgraph, we might end up with a neighbor-swap graph that only admits a Hamiltonian path; subsignature $(\textit{even}, 1, 1)$. This requires us to discuss the $(\textit{even}, 2, 1, 1)$ separately. Moreover, the signature $(2, 2, 1, 1)$ requires more attention because we obtain two subsignatures that generate neighbor-swap graphs that only admit a Hamiltonian path.

1. **Trailing** 0**:** This results in a neighbor-swap graph with signature $(\textit{odd}, 2, 1, 1)$, which is either a three or more odd - rest even signature or signature $(1, 2, 1, 1)$, i.e. an $(\textit{even}, 1, 1, 1)$ signature. For these subsignatures, we refer to § 3.12 and § 3.7 respectively. Both neighbor-swap graphs contain a Hamiltonian cycle.
2. **Trailing** 1**:** This results in subsignature $(\textit{even}, 1, 1, 1)$ which admits a Hamiltonian cycle by §3.7.
3. **Trailing** 2**:** This is split into three sub-cases to account for stutter permutations.
   (a) **Trailing** 02**:** This results in an $(\textit{odd}, 2, 0, 1)$ signature. This admits a Hamiltonian cycle as explained in §3.3.2. In the case of a $(1, 2, 0, 1)$ signature, we remove two nodes as explained below.
   (b) **Trailing** 12**:** This results in an $(\textit{even}, 1, 0, 1)$ signature by Lemma 1. This only admits a Hamiltonian path. However, we remove edge; $x_1 \sim x_2 = 130^{k_0}12 \sim 310^{k_0}12$. By Lemma 1 this results in a Hamiltonian cycle. The removed permutations are glued in the subcycle with a trailing 1. This is done between guaranteed edge (3.84):

   $$130^{k_0}21 \sim 130^{k_0}\underline{12} \sim \underline{31}0^{k_0}12 \sim 310^{k_0}\underline{21}$$

   (c) **Trailing** 32**:** This results in a Hamiltonian cycle on the neighbor-swap graph of signature $(\textit{even}, 2)$. However, we require that the stutter permutations are incorporated in this case so we will combine this into a zig-zag cycle with the trailing 23 part.
4. **Trailing** 3**:** This is split into three sub-cases.
   (a) **Trailing** 03**:** This results in an $(\textit{odd}, 2, 1)$ signature. This admits a Hamiltonian cycle as explained in §3.3.2. In the case of a $(1, 2, 1)$ signature, we remove two nodes as explained below.

(b) **Trailing** 13**:** This results in an $(\textit{even}, 1, 1)$ signature. This only admits a Hamiltonian path by Lemma 1. However, we remove two permutations; $x_1 \sim x_2 = 120^{k_0}13 \sim \underline{21}0^{k_0}13$. Those nodes are glued in the cycle with a trailing 1. Again, this results in a Hamiltonian cycle by Lemma 1. This is done between guaranteed edge (3.84):

$$120^{k_0}31 \sim 120^{k_0}\underline{13} \sim \underline{21}0^{k_0}13 \sim 210^{k_0}\underline{31}$$

(c) **Trailing** 23**:** This results in a Hamiltonian cycle on the neighbor-swap graph of signature $(\textit{even}, 2)$. However, we require that the stutter permutations are incorporated in this case so we will combine this into a zig-zag cycle with the trailing 32 part.

The neighbor-swap graphs of $(\textit{odd}, 2, 1)$ subsignatures with trailing 02 and trailing 03 do not admit Hamiltonian cycles if the starting signature is $(2, 2, 1, 1)$. Fixing the trailing 02 or 03 will result in subsignature $(1, 2, 1)$ which only admits a Hamiltonian path (Lemma 1). Therefore, we must remove two nodes for both of these paths. The paths are glued into the cycle with a trailing 0 between edges that are guaranteed by (3.84). The edges that are created are;

$$031120 \sim 0311\underline{02} \sim \underline{30}1102 \sim 3011\underline{20} \tag{3.92}$$

$$021130 \sim 0211\underline{03} \sim \underline{20}1103 \sim 2011\underline{30} \tag{3.93}$$

Now we have a cycle corresponding to Lemma 1. This neighbor-swap graph is also shown in Fig. B.1. The cycle with trailing 0 has four extra nodes while remaining a cycle.

3.8.1 Connecting the subcycles

We copy the technique for two odd - rest even signatures as described in §3.11 to combine the meta-graph of cycles by gluing parallel edges:

$$1020^{k_0-2}103 \sim \underline{01}20^{k_0-2}103 \tag{3.94}$$

$$1020^{k_0-2}013 \sim \underline{01}20^{k_0-2}013 \tag{3.95}$$

Edge (3.94) is generated in a cycle with trailing 03. In that $(\textit{odd}, 2, 1)$ corresponds to edge (3.16). Or if the signature is $(1, 2, 1)$ we see that the cycle without the node 0211 must contain edges $1021 \sim \underline{01}21 \sim 01\underline{12}$ (see Fig. B.1). Edge (3.95) is generated by signature $(\textit{even}, 1, 1)$ of Lemma 1 where it is part of the d_{k_0-1} path (2.2).

$$1030^{k_0-2}102 \sim \underline{01}20^{k_0-2}103 \tag{3.96}$$

$$1030^{k_0-2}012 \sim \underline{01}30^{k_0-2}012 \tag{3.97}$$

Edges (3.96) and (3.97) exist by the same arguments as (3.94) and (3.95).

$$0^{k_0}1123 \sim 0^{k_0-1}\underline{10}123 \tag{3.98}$$

$$0^{k_0}1213 \sim 0^{k_0-1}\underline{10}213 \tag{3.99}$$

Edge (3.98) is generated in the zig-zag cycle with trailing $23/32$ by Definition 6. Edge (3.99) is generated in the subgraph of Lemma 1 with trailing 13. It is in the d_0 path (2.2).

$$0^{k_0}1132 \sim 0^{k_0-1}\underline{10}132 \tag{3.100}$$

$$0^{k_0}1312 \sim 0^{k_0-1}\underline{10}312 \tag{3.101}$$

Edge (3.100) and (3.101) exist by the same arguments as (3.98) and (3.99).

Now we have a Hamiltonian cycle on the subcycles with trailing odd-occurring elements.

$$0^{k_0-1}11230 \sim 0^{k_0-1}1\underline{21}30 \tag{3.102}$$
$$0^{k_0-1}11203 \sim 0^{k_0-1}1\underline{21}03 \tag{3.103}$$

Edge (3.102) is generated in the $(\textit{odd}, 2, 1, 1)$ cycle where the edge is guaranteed by (3.189). If the subsignature is $(1, 2, 1, 1)$ the edge is guaranteed by (**??**). Edge (3.103) is generated by subsignature $(\textit{odd}, 2, 1)$ with trailing 03 which guarantees the edge by (3.20). The $(1, 2, 1)$ subsignature neighbor-swap graph also covers this edge as seen in Fig. B.1.

$$0^{k_0}1231 \sim 0^{k_0}\underline{21}31 \tag{3.104}$$
$$0^{k_0}1213 \sim 0^{k_0}\underline{21}13 \tag{3.105}$$

Edge (3.104) is generated in a neighbor-swap graph of signature $(\textit{even}, 1, 1, 1)$ where it is guaranteed by (3.91). Edge (3.105) exists because of the d_0 edge with trailing 13.

So we can conclude that the neighbor-swap graph of signature $(\textit{even}, 2, 1, 1)$ contains a Hamiltonian cycle.

3.8.2 Guaranteed edges

$$20^{k_0}113 \sim \underline{02}0^{k_0-1}113 \tag{3.106}$$
$$30^{k_0}112 \sim \underline{03}0^{k_0-1}112 \tag{3.107}$$

Both edges (3.106) and (3.107) exist by the same argument. They are both generated in the $(\textit{even}, 1, 1)$ signatures where they are part of the d_0 path (2.2) of Lemma 1.

$$30^{k_0}121 \sim \underline{03}0^{k_0-1}121 \tag{3.108}$$
$$20^{k_0}131 \sim \underline{02}0^{k_0-1}131 \tag{3.109}$$

Edges (3.108) and (3.109) have a trailing 1. Within the $(\textit{even}, 1, 1, 1)$ cycle, the edges are guaranteed by (3.85).

$$30^{k_0}211 \sim \underline{03}0^{k_0-1}211 \tag{3.110}$$
$$20^{k_0}311 \sim \underline{02}0^{k_0-1}311 \tag{3.111}$$

Edges (3.110) and (3.111) have a trailing 1. Within the $(\textit{even}, 1, 1, 1)$ cycle, the edges are guaranteed by (3.85) again.

$$3110^{k_0-1}20 \sim \underline{13}10^{k_0-1}20 \tag{3.112}$$
$$2110^{k_0-1}30 \sim \underline{12}10^{k_0-1}30 \tag{3.113}$$

Both edges (3.112) and (3.113) exist by similar arguments again. We fix the trailing 0 and see in subsignature $(\textit{odd}, 2, 1, 1)$ that the edges are guaranteed by (3.190) and (3.191).

$$0^{k_0}1123 \sim 0^{k_0}1\underline{21}3 \tag{3.114}$$
$$0^{k_0}1132 \sim 0^{k_0}1\underline{31}2 \tag{3.115}$$

Edge (3.114) is generated by the cross edges between (3.98) and (3.99). Edge (3.115) is one of the cross edges between (3.100) and (3.101).

$$0^{k_0}1123 \sim 0^{k_0}11\underline{32} \tag{3.116}$$
$$110^{k_0}23 \sim 110^{k_0}\underline{32} \tag{3.117}$$

Edges (3.116) and (3.117) are generated in the zig-zag cycle with trailing $23/32$.

$$0^{k_0}1231 \sim 0^{k_0}1\underline{32}1 \tag{3.118}$$

This edge originates in the $(\textit{even}, 1, 1, 1)$ signature where it is guaranteed by (3.89).

$$110^{k_0}32 \sim 1\underline{01}0^{k_0-1}32 \tag{3.119}$$

$$110^{k_0}23 \sim 1\underline{01}0^{k_0-1}23 \tag{3.120}$$

Both edges (3.119) and (3.120) are generated in the zig-zag cycle with trailing $32/23$. Moreover, it is not used as a cross edge in the $(\textit{even}, 2, 1, 1)$ signature. By definition of the zig-zag cycle, the edge must be present in the neighbor-swap graph.

3.9 All even

In the following sections, we generate a disjoint cycle cover on the neighbor-swap graphs of universal signatures and then glue it together using parallel edges. For the all-even signatures, this creation of the disjoint cycle cover is explained by Verhoeff [24]. We split the graph based on the trailing two elements x and y. We distinguish two cases based on whether x and y are of equal color.

If x and y are distinct, we obtain one subgraph for every combination of x and y. Thus $\binom{K}{2}$ subgraphs where two colors occur an odd number of times. When the signature is $(\textit{even}, 2, 2)$ this results in a subgraph that only admits a Hamiltonian path of subsignature $(\textit{even}, 1, 1)$. However, we generate two such paths; one with trailing xy and one with yx. These Hamiltonian paths are isomorphic and parallel. Therefore they can be combined into a Hamiltonian cycle on the subgraphs with trailing xy and yx. Without the trailing elements, these Hamiltonian paths are equal. They form a cycle by following the Hamiltonian path through the xy graph and then following the reversed path through the yx graph. This path ends at a node adjacent to the starting node in the xy Hamiltonian path, thus giving us a cycle. This is shown in Fig. 3.7.

Other subsignatures with two odd-occurring colors allow for a neighbor-swap graph where a Hamiltonian cycle can be constructed. These subsigantures are §3.3.2 ($(\textit{odd}, 2, 1)$), § 3.6 ($(\textit{even}, \textit{odd}, 1)$), §3.8 ($(\textit{even}, 2, 1, 1)$), and §3.11 (general two odd - rest even signatures). We cut this cycle to form a Hamiltonian path and then apply the same technique from Fig. 3.7. The position in which we cut the two odd - rest even subsignature cycle is such that edge (3.166) is the first edge of the path. We handle these signatures as a path because it reduces the number of meta-nodes in the meta-graph of cycles. This means there is less work left in the all-even meta-graphs to create the Hamiltonian cycle in the all-even neighbor-swap graph. Such a combined meta-node is indicated with xy/yx.

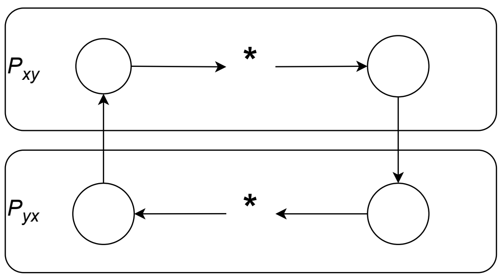


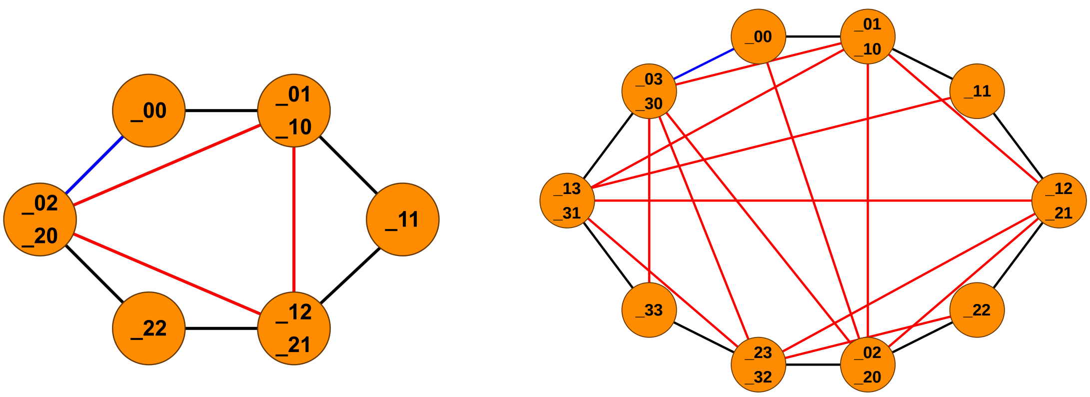
Figure 3.7: The generation of a cycle between two equal paths P; one with trailing xy and one with trailing yx. The * indicates the path that is the same for P_{xy} as for P_{yx}.

If $x = y$, then $k_x = k_y$ and we obtain a subgraph with K colors if $k_x > 2$ or a subgraph with $K-1$ colors if $k_x = 2$. The induction hypothesis gives us a Hamiltonian cycle on the non-stutter permutations for each subgraph. Combining these subgraphs and the subgraphs for $x \neq y$ gives us a disjoint cycle cover as was proven by Verhoeff with Theorem 2.

3.9.1 Connecting the meta-graph of cycles

We show that the meta-graph of cycles of the all-even signatures admits a Hamiltonian cycle. We construct this cycle using Algorithm 2 which the author developed to create this Hamiltonian cycle. The algorithm generates an order in the meta-graph of cycles based on three trailing elements. The meta-graph of cycles has two fixed trailing elements; however, by adding the third element we show how we can swap between the cycles. For example, the cycle with trailing 12 can be connected to the cycle with trailing 02 by using a pair of cross edges where one pair has trailing $\underline{01}2$ and the other $\underline{10}2$.

The meta-graphs of cycles for $K = 3$ and $K = 4$ are shown to admit a Hamiltonian cycle in Fig. 3.8a and Fig. 3.8b respectively. We can construct such a Hamiltonian cycle for all meta-graphs of cycles of all-even signatures with Algorithm 2 as explained in Appendix C.2.1.



(a) The meta-graph of a sorted all-even signature with 3 colors.

(b) The meta-graph of a sorted all-even signature with 4 colors.

Figure 3.8: Two examples of a Hamiltonian cycle on the meta-graph of cycles of all-even signatures. Both orders are generated by Algorithm 2.

Continuing, we will demonstrate the existence of a set of parallel edges that connects this ordering of the meta-graph of cycles of an all-even signature.

3.9.2 Cross edges in the meta-graph of cycles

We will now look more closely into how these cross edges can be guaranteed in the meta-graph of cycles. The cross edges have three fixed elements corresponding to Algorithm 2. The first element of these three fixed elements is part of the subsignature that generates the subcycle. The other two trailing elements determine whether the subsignature is all-even or two odd - rest even. We only combine an all-even subsignature with a two odd - rest even subsignature or vice-versa. We never combine all-even subsignatures with other all-even subsignatures. On the contrary, pairs of two odd - rest even subsignatures are connected in the meta-graph of cycles.

We specify the cross edges between cycles with all-even subsignatures and two odd -

rest even subsignatures respectively;

$$x^{k_x-1}0^{k_0}\dots(K-1)^{k_{K-1}}xyy \sim x^{k_x-2}\underline{0x}0^{k_0-1}\dots(K-1)^{k_{K-1}}xyy \tag{3.121}$$
$$x^{k_x-1}0^{k_0}\dots(K-1)^{k_{K-1}}yxy \sim x^{k_x-2}\underline{0x}0^{k_0-1}\dots(K-1)^{k_{K-1}}yxy \tag{3.122}$$

So we show that these edges are guaranteed in the neighbor-swap graph of all-even subsignatures. The dots $(\dots)$ in the edge represent all elements from 1^{k_1} up to and including $(K-2)^{k_{K-2}}$; with k_y-2 elements y but without any element x. These edges exists for all combinations of x and y. If k_x is k_0 or k_{K-1}, the $0^{k_0}\dots(K-1)^{k_{K-1}}$ part is replaced with $1^{k_1}\dots(K-1)^{k_{K-1}}$ or $0^{k_0}\dots(K-2)^{k_{K-2}}$ respectively. The same holds for y. Edge (3.121) is first generated in an $(\mathit{even}, \mathit{even}, 2)$ signature where we fix two trailing elements of k_2. The subcycle must contain edge (3.121) since any $(\mathit{even}, \mathit{even})$ contains edges $0^{k_0-2}101^{k_1-1}0 \sim 0^{k_0-2}\underline{01}1^{k_1-1}0 \sim 0^{k_0-1}1^{k_1-1}\underline{01}$. The middle node has degree two and the $(\mathit{even}, \mathit{even})$ signature contains a Hamiltonian cycle on the non-stutter permutations. The edge is guaranteed for other all-even signature subcycles by (3.128)

There are several options for the origin of edge (3.122). These are:

1. $(\mathit{even}, 1, 1)$
2. $(\mathit{odd}, 2, 1)$
3. $(\mathit{even}, \mathit{odd}, 1)$
4. $(\mathit{even}, 2, 1, 1)$
5. Two odd - rest even

All of these subsignatures contain two odd-occurring colors. We now address the subsignatures that generate the edges and explain why the cross edges are present.

3.9.2.1 Subsignature $(\boldsymbol{even}, 1, 1)$:

Edge (3.122) is either $10^{k_0}212 \sim \underline{010}^{k_0-1}212$, which is equal to (3.27) with fixed trailing 12. Or it is $20^{k_0}121 \sim \underline{020}^{k_0-1}121$ with fixed trailing 21, which is equal to (3.28).

3.9.2.2 Subsignature $(\boldsymbol{odd}, 2, 1)$:

The cross edge is;

$$0^{k_0-1}11202 \sim 0^{k_0-2}\underline{10}1202 \tag{3.123}$$

Edge (3.123) is guaranteed in the $(\mathit{odd}, 2, 1)$ cycle with (3.12).

3.9.2.3 Subsignature $(\boldsymbol{even}, \boldsymbol{odd}, 1)$:

The parallel edge is:

$$1^{k_1-1}0^{k_0}212 \sim 1^{k_1-2}\underline{010}^{k_0-1}212 \tag{3.124}$$

In the $(\mathit{even}, \mathit{odd}, 1)$ signature, this edge is guaranteed by (3.68).

3.9.2.4 Subsignature $(\boldsymbol{even}, 2, 1, 1)$:

In this signature, we have the parallel edges:

$$20^{k_0}11323 \sim \underline{020}^{k_0-1}11323 \tag{3.125}$$

This edge is present in the $(\mathit{even}, 2, 1, 1)$ signature because it is guaranteed as (3.107). Note that 2 and 3 can also be swapped which results in guaranteed edge (3.106).

3.9.2.5 Subsignature two odd - rest even:

These subsignatures are the most general and the cross edges are edge (3.122). We can directly refer to (3.166). This edge is precisely the edge that we want as the cross edge. If we add xy or yx to that edge we obtain an all-even signature.

We still need to show that the cross edges between two two odd - rest even signature subcycles are guaranteed. These are between cycles where Algorithm 2 outputs three distinct trailing elements. Assume we have trailing elements xyz. This means that k_y and k_z are odd in one subsignature. The other subsignature has odd-occurring colors k_x and k_z. So, we use the parallel edges:

$$z^{k_z-1}0^{k_0}\dots(K-1)^{k_{K-1}}x^{k_x-1}y^{k_y-1}xyz \sim z^{k_z-2}\underline{0z}0^{k_0-1}\dots(K-1)^{k_{K-1}}x^{k_x-1}y^{k_y-1}xyz \tag{3.126}$$

$$z^{k_z-1}0^{k_0}\dots(K-1)^{k_{K-1}}x^{k_x-1}y^{k_y-1}yxz \sim z^{k_z-2}\underline{0z}0^{k_0-1}\dots(K-1)^{k_{K-1}}x^{k_x-1}y^{k_y-1}yxz \tag{3.127}$$

The dots $(\dots)$ in edge (3.126) represent all elements from 1^{k_1} up to and including $(K-2)^{k_{K-2}}$; without any element x, y or z. These edges exists for all combinations of x, y, and z.

Edge (3.126) corresponds to the edge we see in (3.166) again. So we know that the edge is present in the Hamiltonian cycle with a two odd - rest even signature.

Edge (3.127) is similar. However, it is generated in a two odd - rest even signature where we added an even number of trailing elements y^{k_y}. This edge is guaranteed by (3.167).

So we conclude that a Hamiltonian cycle exists on the non-stutter permutations for all-even signatures.

3.9.3 Guaranteed edges

$$x^{k_x-1}0^{k_0}\dots(K-1)^{k_{K-1}}x \sim x^{k_x-2}\underline{0x}0^{k_0-1}\dots(K-1)^{k_{K-1}}x \tag{3.128}$$

The dots $(\dots)$ in edge (3.128) represent all elements from 1^{k_1} up to and including $(K-2)^{k_{K-2}}$; without any element x. The edge exists for all colors k_x. Note that neither of these nodes are stutter permutations. This edge is guaranteed because the signature where this edge is formed in a two odd - rest even signature with trailing $(K-1)x$. Within this subcycle, edge $x^{k_x-1}0^{k_0}\dots(K-1)^{k_{K-1}-1} \sim x^{k_x-1}\underline{0x}0^{k_0-1}\dots(K-1)^{k_{K-1}-1}$ has colors k_x and $k_{K-1}-1$ that are odd. That edge is guaranteed by (3.166). Note that we want only one trailing x for these nodes, this allows the edge to be used as a cross edge later. If we have $k_x \geq 3$ elements x The edge is parallel to (3.122) with trailing yx instead of trailing xy.

3.10 One odd - rest even

The one odd - rest even signatures are also addressed by Verhoeff [24]. The graph is split into subcycles by fixing the trailing element x and distinguishing two cases based on the parity of k_x. If k_x is odd, we obtain one subgraph of the all-even signature. By the induction hypothesis, this subgraph contains a Hamiltonian cycle since the length of the permutations with this fixed trailing element is $n-1$. Or if the signature was even-even-1 (with both even-occurring colors ≥ 4) then it is now a binary neighbor-swap graph. §3.2 addresses the case where at least one color occurs twice. By Theorem 1 we have a Hamiltonian cycle on the non-stutter permutations for this graph.

If k_x is even, we obtain $K-1$ distinct subgraphs. One for every color that occurs an even number of times. These subgraphs have two colors occurring an odd number of times and admit a Hamiltonian cycle, except when the resulting signature is $(\mathit{even}, 1, 1)$. If the resulting signature is $(\mathit{even}, 1, 1)$, the original signature was $(\mathit{even}, 2, 1)$ which we addressed in § 3.2. Similar to all-even signatures, we first connect the meta-graph of cycles into a Hamiltonian cycle and then we address the parallel edges that glue the subcycles together.

3.10.1 Connecting the meta-graph of cycles

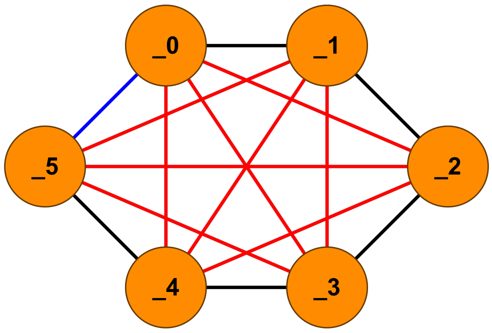


Figure 3.9: The meta-graph of signature one odd - rest even with 6 colors.

The meta-graph of cycles on both one odd - rest even and three or more odd - rest even signatures is trivial to admit a Hamiltonian cycle. The subcycles arise by fixing a single trailing element. This yields a meta-graph of cycles that is fully connected. Algorithm 3 generates one of the possible Hamiltonian cycles. The order begins with fixing the trailing element of color k_0, followed by k_1, continuing up to k_{K-1}. Algorithm 3 provides suffixes of length two where the rightmost element is the trailing element of the first cycle and the leftmost element of the second cycle. Providing these suffixes shows that we can form a Hamiltonian path in the meta-graph of cycles. An example of such a Hamiltonian cycle in the meta-graph of cycles with $K = 6$ is shown in Fig. 3.9.

3.10.2 Cross edges in the meta-graph of cycles

The subcycles of one odd - rest even signatures can have the following signatures in their subgraphs:

- $(\mathit{even}, \mathit{even})$
- $(\mathit{odd}, 2, 1)$
- $(\mathit{even}, \mathit{odd}, 1)$
- $(\mathit{even}, 2, 1, 1)$
- Two odd - rest even
- All-even.

We will provide a one odd - rest even signature for the first four signatures that has one of the signatures as its subgraph. For the last two signatures, we provide a more general technique. The one odd - rest even signature has one all-even subsignature and all others have subsignature two odd - rest even. We will present the cross edges that connect the meta-graph of cycles below.

3.10.2.1 Signature $(\mathit{even}, \mathit{even}, 1)$

We assume that both $k_0 \geq 4$ and $k_1 \geq 4$. This means that we have to show that the disjoint cycle cover on the signature $(\mathit{odd}, \mathit{even}, 1)$ with trailing 0, $(\mathit{even}, \mathit{odd}, 1)$ with trailing 1, and (*even*, *even*) with trailing 2 can be glued into one Hamiltonian cycle. To do this we use the parallel edges:

$$20^{k_0-1}1^{k_1-1}10 \sim \underline{020}^{k_0-2}1^{k_1-1}10 \tag{3.129}$$
$$20^{k_0-1}1^{k_1-1}01 \sim \underline{020}^{k_0-2}1^{k_1-1}01 \tag{3.130}$$

Edge (3.129) is guaranteed by (3.66). Edge (3.130) is generated in an (*even*, *odd*, 1) signature cycle. By edge (3.69), it is guaranteed in this subcycle.

$$1^{k_1-1}0^{k_0}21 \sim 1^{k_1-2}\underline{010}^{k_0-1}21 \tag{3.131}$$
$$1^{k_1-1}0^{k_0}12 \sim 1^{k_1-2}\underline{010}^{k_0-1}12 \tag{3.132}$$

Edge (3.131) is generated in subsignature (*even*, *odd*, 1) where the edge is guaranteed by (3.68). Edge (3.132) is generated by subsignature (*even*, *even*). Within that subcycle, the first node has degree two. Therefore the edge is guaranteed.

3.10.2.2 Signature (*odd*, 2, 2)

We show the existence of parallel edges:

$$1220^{k_0-1}01 \sim \underline{21}20^{k_0-1}01 \tag{3.133}$$
$$1220^{k_0-1}10 \sim \underline{21}20^{k_0-1}10 \tag{3.134}$$

Edge (3.133) is generated in an (*odd*, 1, 2) signature. There it is guaranteed by (3.18). Edge (3.134) is generated in an (*even*, 2, 2) subsignature that guarantees edge (3.128).

$$0^{k_0}1212 \sim 0^{k_0}\underline{21}12 \tag{3.135}$$
$$0^{k_0}1221 \sim 0^{k_0}\underline{21}21 \tag{3.136}$$

Edge (3.135) is generated in an (*odd*, 2, 1) signature. The edge is guaranteed by edge (3.21). Moreover, the same argument holds for edge (3.136). This edge is generated as an (*odd*, 1, 2) signature where the edge also reduces to (3.21).

3.10.2.3 Signature (*even*, *odd*, 2)

This signature has a two odd - rest even subsignature with trailing 0, an all-even subsignature with trailing 1, and subsignature (*even*, *odd*, 1) with trailing 2.

$$0^{k_0-1}221^{k_1-1}10 \sim 0^{k_0-2}\underline{202}1^{k_1-1}10 \tag{3.137}$$
$$0^{k_0-1}221^{k_1-1}01 \sim 0^{k_0-2}\underline{202}1^{k_1-1}01 \tag{3.138}$$

Edge (3.137) is guaranteed in a two odd - rest even subsignature by (3.166). Edge (3.138) is generated in an all-even subsignature. The edge reduces to guaranteed edge (3.128).

$$20^{k_0}1^{k_1-1}21 \sim \underline{020}^{k_0-1}1^{k_1-1}21 \tag{3.139}$$
$$20^{k_0}1^{k_1-1}12 \sim \underline{020}^{k_0-1}1^{k_1-1}12 \tag{3.140}$$

Edge (3.139) is generated in the all-even subsignature. Similar to before, it is guaranteed by (3.128). Edge (3.140) is generated in the signature (*even*, *odd*, 1). In this signature, the edge is guaranteed by edge (3.66).

3.10.2.4 Signature $(\mathit{even}, 2, 2, 1)$

Like before, we combine the four subcycles that we obtain by fixing one trailing element:

$$3220^{k_0-1}110 \sim \underline{23}20^{k_0-1}110 \tag{3.141}$$
$$3220^{k_0-1}101 \sim \underline{23}20^{k_0-1}101 \tag{3.142}$$

Edge (3.141) is generated in an $(\mathit{odd}, 2, 2, 1)$ signature. This two odd - rest even signature is guaranteed to cover edge (3.167). Edge (3.142) is generated in a signature that reduces to $(\mathit{even}, 2, 1, 1)$. In that signature, the edge is guaranteed by (3.112).

$$30^{k_0}1221 \sim \underline{03}0^{k_0-1}1221 \tag{3.143}$$
$$30^{k_0}1212 \sim \underline{03}0^{k_0-1}1212 \tag{3.144}$$

Edge (3.143) is generated in a subsignature that is reduced to $(\mathit{even}, 2, 1, 1)$. The edge is guaranteed by (3.110). Similarly, edge (3.144) is generated by a subsignature that reduces to $(\mathit{even}, 2, 1, 1)$. The edge is guaranteed by (3.108).

$$20^{k_0}1132 \sim \underline{02}0^{k_0-1}1132 \tag{3.145}$$
$$20^{k_0}1123 \sim \underline{02}0^{k_0-1}1123 \tag{3.146}$$

Edge (3.145) is generated by subsignature $(\mathit{even}, 2, 1, 1)$. In this signature, edge (3.106) is guaranteed. Edge (3.146) is generated in an $(\mathit{even}, 2, 2)$ signature. That subcycle guarantees edge (3.128) so the edge must be present.

3.10.2.5 Subsignature two odd - rest even

Let us first give a general way to represent the edge of a two odd - rest even signature subgraph. Assume we have k_x odd and all other colors even. Then the colors k_x and k_z are odd in this subgraph of edge (3.147) and k_x and k_y are odd in the subgraph of edge (3.148). We glue the parallel edges:

$$x^{k_x}0^{k_0}\dots(K-1)^{k_{K-1}}z^{k_z-1}y^{k_y-1}zy \sim x^{k_x-1}\underline{0x}0^{k_0-1}\dots(K-1)^{k_{K-1}}z^{k_z-1}y^{k_y-1}zy \tag{3.147}$$
$$x^{k_x}0^{k_0}\dots(K-1)^{k_{K-1}}z^{k_z-1}y^{k_y-1}yz \sim x^{k_x-1}\underline{0x}0^{k_0-1}\dots(K-1)^{k_{K-1}}z^{k_z-1}y^{k_y-1}yz \tag{3.148}$$

The dots $(\dots)$ in these edges represent all elements from 1^{k_1} up to and including $(K-2)^{k_{K-2}}$; without any element x, y or z. The edge exists for all combinations of colors k_x, k_y, and k_z. Edge (3.147) corresponds to edge (3.166) that is guaranteed in the Hamiltonian cycle of subsignature two odd - rest even. Edge (3.148) is guaranteed because of edge (3.167) in the other two odd - rest even subsignature. So we can conclude these edges exist and we can connect the subcycles with signature two odd - rest even.

3.10.2.6 Subsignature all-even

The edge in the all-even subsignature should be parallel to the edge we chose for the two odd - rest even signatures. Therefore, we will use edge (3.149) for the all-even subsignature cycle. The edge is parallel to edge (3.150) from a two odd - rest even subsignature. For these edges, we assume k_x is odd again.

$$z^{k_z-1}0^{k_0}\dots(K-1)^{k_{K-1}}x^{k_x-1}zx \sim z^{k_z-2}\underline{0z}0^{k_0-1}\dots(K-1)^{k_{K-1}}x^{k_x-1}zx \tag{3.149}$$
$$z^{k_z-1}0^{k_0}\dots(K-1)^{k_{K-1}}x^{k_x-1}xz \sim z^{k_z-2}\underline{0z}0^{k_0-1}\dots(K-1)^{k_{K-1}}x^{k_x-1}xz \tag{3.150}$$

The dots $(\dots)$ in these edges represent all elements from 1^{k_1} up to and including $(K-2)^{k_{K-2}}$; with k_y y's and without any elements x or z. The edge exists for all combinations of colors k_x,

k_y, and k_z. Edge (3.149) is present in an all-even signature by the guaranteed edge (3.128). Edge (3.150) is guaranteed in the two odd - rest even signature cycles similar to before. The edge reduces to the one guaranteed by (3.166).

Therefore we have shown that the neighbor-swap graph of one odd - rest even signatures contains a Hamiltonian cycle on the non-stutter permutations.

3.10.3 Guaranteed edges

For the edges below, we have k_x odd and all other colors are even. There are multiple of these edges, one for every color k_y.

$$0^{k_0}\dots(K-1)^{k_{K-1}}x^{k_x}y^{k_y} \sim 0^{k_0}\dots(K-1)^{k_{K-1}}x^{k_x-1}\underline{yx}y^{k_y-1} \tag{3.151}$$

The dots $(\dots)$ in edge (3.151) represent all elements from 1^{k_1} up to and including $(K-2)^{k_{K-2}}$; without any element x or y. Edge (3.151) is generated in a two odd - rest even signature with k_x and k_y-1 odd. Edge (3.163) is exactly equal to this edge. Moreover, if the subgraph has a special signature we have edge (3.22) and (3.23) for $(odd,2,1)$. If the signature is $(even,odd,1)$ we have edges (3.70) and (3.71). For the $(even,2,1,1)$ signature subcycle, the edge is guaranteed by (3.116) and (3.117).

3.11 Two odd - rest even

The two odd - rest even signatures for which we prove the Hamiltonian cycle in this section are those with $K \geq 3$ and where the signature is not $(even,1,1)$, $(odd,2,1)$, $(even,odd,1)$, or $(even,2,1,1)$. Recall that stutter permutations are only present for signatures with at most one odd-occurring color. This means they were not present in the previous Hamiltonian cycles on all-even and one odd - rest even signatures. The subgraphs contain stutter permutations, but when adding the trailing element(s) to obtain the two odd - rest even signature, they no longer are stutter permutations and must be incorporated. For this, we will use a technique that closely resembles that of Verhoeff for the binary signatures [24]. We make a split on the subgraphs of the two odd - rest even signatures by fixing the single trailing element, as shown in Fig. 3.10. Assume that the colors k_x and k_y are odd. When the fixed trailing element is x or y, we must fix another trailing element because the subcycles are on the non-stutter permutations of the one odd - rest even signatures. However, the stutter permutations of these signatures must be included in the two odd - rest even signature. Those two subcycles will be combined first to account for the stutter permutations. This is indicated with the orange edge in Fig. 3.10. After that, the blue edges will be used to attach each of the three odd - rest even subcycles with a trailing even-occurring element.

We split § 3.11.1 and § 3.11.2 into two parts. For both, we will separately explain how the two subcycles that have an odd-occurring trailing element are combined and how the connection to the other subcycles with even-occurring trailing elements is established.

3.11.1 Connecting the meta-graph of cycles

The subcycles in the meta-graph of cycles are obtained by fixing the single trailing element. However, instead of K subcycles with the one odd - rest even and three or more odd - rest even signatures, this results in $K-1$ subcycles because we combine the two subcycles where the trailing element is of an odd-occurring color. Therefore, we obtain a meta-graph of cycles that is shown in the top row of Fig. 3.10. This meta-graph is fully connected again. By obtaining the blue edges in the meta-graph of Fig. 3.10, we can glue the cycles in the star formation. Fig. 3.1 shows an example of this with four subcycles.

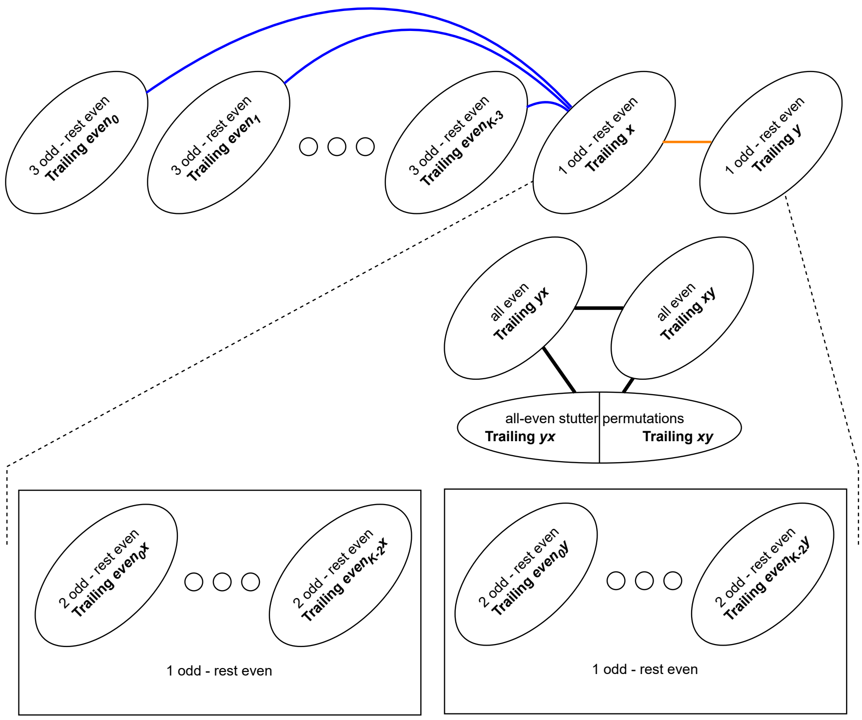


Figure 3.10: How to split a neighbor-swap graph of a signature with two colors odd-occurring colors k_x and k_y into cycles based on the trailing element.

The meta-graph of cycles admits a Hamiltonian cycle which can be generated with Algorithm 3. However, that would generate a Hamiltonian cycle on K cycles where we only have $K-1$ cycles. Therefore we have to remove either k_x or k_y from the signature and then apply Algorithm 3. That will generate a Hamiltonian cycle on the meta-graph of cycles thus proving Lemma 4 for the two odd - rest even signatures.

3.11.1.1 Connecting the cycles with odd-occurring trailing elements

To combine the two cycles with trailing x and y we have to fix more trailing elements. Fig. 3.11 shows a technique that can be used to generate a Hamiltonian cycle on two neighbor-swap graphs of one odd - rest even subsignatures with a trailing x or y. First, three cycles (indicated with orange) are created. These cycles are then combined using the blue edges. One of these cycles (the top one in Fig. 3.11) is a zig-zag cycle (Definition 6) on two all-even subsignatures. This cycle has the stutter permutations incorporated as can be seen in the figure. Then we still have the two cycles on one odd - rest even subsignatures with trailing odd-occurring element (in the bottom of Fig. 3.11). This is a Hamiltonian cycle by leaving out the all-even subcycle from the one odd - rest even signature. We use the technique from §3.10 to combine these subgraphs into a Hamiltonian cycle.

3.11.1.2 Adding the cycles with even-occurring trailing elements

There are $K-1$ subcycles in the one odd - rest even subsignatures. They range from the cycles with a trailing *even*$_0x$ to cycles with a trailing *even*$_{K-2}x$ as can be seen in Fig. 3.11.

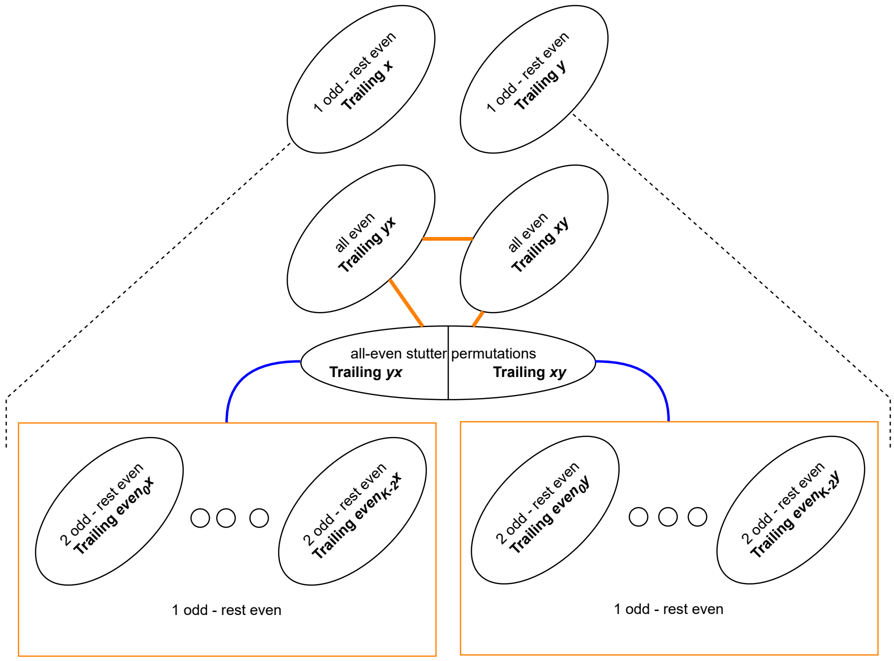


Figure 3.11: Combination of two cycles with signature one odd - rest even with a trailing odd element. Here, k_x and k_y are odd and they are the last two colors of the signature.

However, there are $K-2$ even-occurring colors in the signature. This is because the subcycle with trailing xx also results in a two odd - rest even subcycle. The same holds for y.

The meta-graph of cycles of two odd - rest even signatures is connected using the technique we explained above. We now continue to show what parallel edges are used to glue the subcycles into a Hamiltonian cycle.

3.11.2 Cross edges in the meta-graph of cycles

Again, we explain separately how the two subcycles that have an odd-occurring trailing element are combined. Then we address how the connection to the other subcycles with even-occurring trailing elements is established.

3.11.2.1 Connecting the cycles with odd-occurring trailing elements

We will first explain how the three cycles with a trailing odd-occurring color admit a Hamiltonian cycle. For this, we have to combine them using the blue edge in Fig. 3.11. Again, assume k_x and k_y are odd and the other colors are even. The signature is sorted such that the color frequencies are in decreasing order and lexicographically if the frequencies are equal. The parallel edges for the three subcycles are:

$$0^{k_0}\dots(K-1)^{k_{K-1}}xy \sim 0^{k_0}\dots(K-2)^{k_{K-2}-1}\underline{(K-1)(K-2)}(K-1)^{k_{K-1}-1}xy \tag{3.152}$$

$$0^{k_0}\dots(K-1)^{k_{K-1}-1}x(K-1)y \sim 0^{k_0}\dots(K-2)^{k_{K-2}-1}\underline{(K-1)(K-2)}(K-1)^{k_{K-1}-2}x(K-1)y \tag{3.153}$$

The dots $(\dots)$ in these edges represent all elements from 1^{k_1} up to and including $(K-2)^{k_{K-2}}$; with k_x-1 elements x and k_y-1 elements y. The elements here must be ordered corresponding to the signature.

Edge (3.152) is formed in the zig-zag cycle with trailing xy. This subcycle covers this edge by Definition 6.

Edge (3.153) is formed in the one odd - rest even signature cycle without the all-even subcycle. The subcycle has fixed trailing elements $(K-1)y$. This follows from the cross edges in one odd - rest even signatures which start with an odd-occurring color, whereas (3.153) does not. So we must show that edge

$$0^{k_0}\dots(K-1)^{k_{K-1}-1}x \;\sim\; 0^{k_0}\dots(K-2)^{k_{K-2}-1}\underline{(K-1)(K-2)}(K-1)^{k_{K-1}-2}x$$

is covered in the two odd - rest even subcycles. Edge (3.153) has several origins in signatures, $(\textit{odd},2,1)$, $(\textit{even},\textit{odd},1)$, $(\textit{even},2,1,1)$, and general two odd - rest even signatures.

Subsignature $(\textit{odd},2,1)$
In this signature, we are looking for edge:

$$0^{k_0-1}1120 \;\sim\; 0^{k_0-1}1\underline{21}0 \quad \text{with trailing } 21. \tag{3.154}$$

Edge (3.154) is guaranteed in the $(\textit{odd},2,1)$ signature by (3.13).

Subsignature $(\textit{even},\textit{odd},1)$
For this signature, we have either k_0 even and k_1 odd such as in §3.6 or we have k_0 odd and k_1 even from Appendix D.1. The prior means that we have trailing 20 and parallel edges:

$$0^{k_0-1}1^{k_1-1}21 \;\sim\; 0^{k_0-1}1^{k_1-2}\underline{21}1 \tag{3.155}$$
$$1^{k_1-1}0^{k_0-1}21 \;\sim\; 1^{k_1-1}0^{k_0-2}\underline{20}1 \tag{3.156}$$

Signature $(\textit{even},\textit{odd},1)$ guarantees edge (3.155) by (3.74) if $k_0 > k_1$ or (3.156) by (3.76) if $k_0 < k_1$.

Otherwise, we have trailing 21 with k_0 odd and k_1 even:

$$0^{k_0-1}1^{k_1-1}20 \;\sim\; 0^{k_0-1}1^{k_1-1}\underline{21}0 \tag{3.157}$$
$$1^{k_1}0^{k_0-1}20 \;\sim\; 1^{k_1}0^{k_0-2}\underline{20}0 \tag{3.158}$$

Signature $(\textit{odd},\textit{even},1)$ guarantees edge (3.157) by (D.5) if $k_0 > k_1$ or (3.158) by (D.7) if $k_0 < k_1$.

Subsignature $(\textit{even},2,1,1)$
The edge that we use is:

$$0^{k_0-1}1123 \;\sim\; 0^{k_0-1}1\underline{21}3 \quad \text{with trailing } 20. \tag{3.159}$$
$$0^{k_0-1}1132 \;\sim\; 0^{k_0-1}1\underline{31}2 \quad \text{with trailing } 30. \tag{3.160}$$

The edges are guaranteed in the Hamiltonian cycle of this signature by (3.114) and (3.115) respectively.

Subsignature two odd - rest even
We obtain this signature by fixing a trailing odd-occurring element x and then an even-occurring element $(K-1)$. Thus we have k_x becoming even and k_{K-1} becoming odd in the subgraph. Within this subgraph, the edge corresponds to the one between a stutter permutation and a permutation where the last distinct elements from the stutter part of the permutation are swapped. This is exactly the edge added in the subgraph on permutations of length $n-2$ using the zig-zag cycle (Definition 6). Moreover, this edge remained intact when gluing the subcycles in that two odd - rest even case by the guaranteed edge (3.163).

This concludes the explanation of how the three cycles with trailing x and y of Fig. 3.11 are glued with the two blue edges.

3.11.2.2 Adding the cycles with even-occurring trailing elements

We are left with showing that we can glue each of the subcycles with an even-occurring trailing element to the cycle with trailing x/y. This process is visualized by the blue edges in Fig. 3.10. We glue the parallel edges:

$$0^{k_0}\dots(K-1)^{k_{K-1}}z^{k_z-1}y^{k_y}xz \sim 0^{k_0}\dots(K-1)^{k_{K-1}}z^{k_z-2}\underline{yz}y^{k_y-1}xz \tag{3.161}$$

$$0^{k_0}\dots(K-1)^{k_{K-1}}z^{k_z-1}y^{k_y}zx \sim 0^{k_0}\dots(K-1)^{k_{K-1}}z^{k_z-2}\underline{yz}y^{k_y-1}zx \tag{3.162}$$

The dots $(\dots)$ in these edges represent all elements from 1^{k_1} up to and including $(K-2)^{k_{K-2}}$; with k_x-1 elements x and without any element y or z. The elements must be ordered corresponding to the signature.

Edge (3.161) is generated in the cycle with a three odd - rest even signature. The $(\mathit{odd},1,1)$ signature is never a subcycle here because adding the trailing z would only generate an $(\mathit{odd},2,1)$ or $(\mathit{even},1,1)$ signature.

Another three odd - rest even signature is $(\mathit{odd},\mathit{odd},1)$ signature. This only results in signatures where edge (3.161) is guaranteed by (3.44), (3.48), and (3.51).

The last special case of a three odd - rest even signature is $(\mathit{even},1,1,1)$ where the edge is guaranteed by (3.91).

For the three odd - rest even signatures, this edge is guaranteed by (3.205).

Edge (3.162) is generated in the subcycle where the trailing x with k_x odd is fixed. This results in subsignature one odd - rest even shown in one of the orange boxes of Fig. 3.11. By §3.10, this edge is guaranteed as edge (3.151).

Therefore we can conclude that two odd - rest even signature neighbor-swap graphs contain a Hamiltonian cycle.

3.11.3 Guaranteed edges

For all edges below, k_x and k_y are odd and all other colors occur an even number of times. The values of k_x and k_y can be treated interchangeably.

$$0^{k_0}\dots(K-1)^{k_{K-1}}x^{k_x}y^{k_y} \sim 0^{k_0}\dots(K-1)^{k_{K-1}}x^{k_x-1}\underline{yx}y^{k_y-1} \tag{3.163}$$

The dots $(\dots)$ in edge (3.163) represent all elements from 1^{k_1} up to and including $(K-2)^{k_{K-2}}$; without any element x or y. The elements must be ordered corresponding to the signature.

Edge (3.163) follows from the proof of the two odd - rest even case. Assume $k_y=1$, edge (3.163) is guaranteed by the definition of a zig-zag cycle (Definition 6) with trailing xy/yx. The subcycle is on signature all-even because k_x-1 and k_y-1 are even. This edge now has an even number of trailing elements y. This edge remains intact since it is not used as a cross edge in the two odd - rest even signatures. In the two odd - rest even signatures, we have a step where we add two trailing odd elements. This happens in the orange boxes of signature one odd - rest even of Fig. 3.11. Moreover, this edge is guaranteed in signature $(\mathit{odd},2,1)$ by (3.22), (3.23), and (3.26). Signature $(\mathit{odd},\mathit{even},1)$ also guarantees the edge by (3.70), (3.71), and (3.63). Signature $(\mathit{even},2,1,1)$ also guarantees this edge by (3.116) and (3.117).

$$(K-1)^{k_{K-1}}\dots0^{k_0}xy \sim (K-1)^{k_{K-1}}\dots1^{k_1-1}\underline{01}0^{k_0-1}xy \tag{3.164}$$

The dots $(\dots)$ in edge (3.164) represent all elements from $(K-2)^{k_{K-2}}$ up to and including 1^{k_1}; with k_x-1 elements x and k_y-1 elements y. The elements must be ordered in reverse corresponding to the signature.

We know that the edge must be in the zig-zag cycle by the same reasoning that was used for (3.152). This edge is sorted differently than (3.152), i.e. the even-occurring colors are sorted from least occurring to most occurring and reverse-lexicographically if the frequencies are equal. Signature $(\mathit{odd},2,1)$ guarantees this edge by (3.24). Signature $(\mathit{even},\mathit{odd},1)$

guarantees this edge by (3.75) or (3.77). Signature $(odd, even, 1)$ guarantees this edge by (D.6) or (D.8). Signature $(even, 2, 1, 1)$ also guarantees this edge by (3.119) and (3.120).

$$(K-1)^{k_{K-1}} \dots 0^{k_0} x y \sim (K-1)^{k_{K-1}} \dots 0^{k_0} \underline{y x} \tag{3.165}$$

The dots $(\dots)$ in edge (3.165) represent all elements from $(K-2)^{k_{K-2}}$ up to and including 1^{k_1}; with $k_x - 1$ elements x and $k_y - 1$ elements y. The elements must be ordered in reverse corresponding to the signature.

Edge (3.165) is generated in the zig-zag cycle with trailing xy/yx. By definition, this edge must be present in the two odd - rest even signature Hamiltonian cycle. Signature $(odd, 2, 1)$ guarantees the edge by (3.23). Signature $(even, odd, 1)$ or $(odd, even, 1)$ guarantees it by (3.71). Signature $(even, 2, 1, 1)$ guarantees the edge by (3.117).

$$x^{k_x} 0^{k_0} \dots (K-1)^{k_{K-1}} y \sim x^{k_x - 1} \underline{0x} 0^{k_0 - 1} \dots (K-1)^{k_{K-1}} y \tag{3.166}$$

The dots $(\dots)$ in edge (3.166) represent all elements from 1^{k_1} up to and including $(K-2)^{k_{K-2}}$; with $k_y - 1$ elements y and without any element x. The elements between the last x and last y are not ordered in any way specific as long as they are all together based on their color. Meaning all z's must be in one group as z^{k_z}.

For edge (3.166) we have several signatures as subcycles. Note that $K-1$ can be equal to y. Signature $(odd, 2, 1)$ guarantees this edge by (3.18) and the last edge in (3.12). Signature $(even, odd, 1)$ and $(odd, even, 1)$ guarantee the edge by (3.66), (3.67), and (3.68). Signature $(even, 2, 1, 1)$ guarantees the edge by (3.106) and (3.107). For other signatures, we can form this edge by starting with subsignature two odd - rest even cycle where k_x and $k_{K-1} - 1$ are odd. By adding the trailing elements $(K-1)y$, k_{K-1} becomes even and k_y becomes odd. This edge is not used in the two odd - rest even cycle as a cross edge so its existence is guaranteed.

$$x^{k_x} 0^{k_0} \dots y^{k_y} \dots (K-1)^{k_{K-1}} \sim x^{k_x - 1} \underline{0x} 0^{k_0 - 1} \dots y^{k_y} \dots (K-1)^{k_{K-1}} \tag{3.167}$$

The dots $(\dots)$ in edge (3.167) represent all elements from 1^{k_1} up to and including $(K-2)^{k_{K-2}}$; with somewhere in the middle a group of k_y elements y. There are no elements x in the middle part.

For (3.167) we use that edge (3.166) exists. From that edge, we can add two equal trailing elements in the three odd - rest even signature. These edges with equal trailing elements extend edge (3.166) into edge (3.167). Because the edge is not used in the three odd - rest even signatures, it remains intact in the two odd - rest even signatures.

3.12 Three or more odd - rest even

The meta-graph of cycles of three or more odd - rest even signatures is created by fixing a single trailing element. The subgraphs have at least two odd-occurring colors. This means we have no problems regarding stutter permutations for these signatures. The connectivity of the meta-graph of cycles is replicated from the one odd - rest even signatures. Therefore, we conclude that Lemma 4 holds. We are left with explaining the existence of parallel edges in the meta-graph of cycles.

Fixing a single trailing element in a three or more odd - rest even signature yields a new number of odd-occurring colors. The subsignature contains one more or one less odd-occurring color. If we fix the trailing element x and k_x is even, we obtain a signature with one more odd-occurring color. If k_x is odd that number is decremented by one. We simplify the proof of parallel edges splitting the three or more odd - rest even signatures into an "odd number of odd-occurring colors" signatures and an "even number of odd-occurring colors" signatures. This separation ensures that the parallel edges of an "odd number of odd-occurring colors" signatures do not interfere with the parallel edges of an "even number of odd-occurring colors" signatures, and vice versa. The

3.12.1 Cross edges in the meta-graph of cycles

Generally, the meta-nodes in the meta-graph of cycles are on signatures with two odd - rest even or three or more odd - rest even signatures. The two odd - rest even signatures will be discussed as a subsignature in the part of an "odd number of odd-occurring colors" signatures. Fixing a trailing element could also result in subsignatures:

- $(odd, 1, 1)$
- $(odd, 2, 1)$
- $(even, 1, 1, 1)$
- $(even, 2, 1, 1)$
- $(odd, odd, 1)$
- $(even, odd, 1)$.

These subsignatures require some special attention because they are not part of the general two odd - rest even or three or more odd - rest even signatures. Their neighbor-swap graphs admit Hamiltonian cycles that are discussed separately in the sections above. These subsignatures follow the same proof because the required edges are also guaranteed as shown in two odd - rest even signatures. However, we address subsignatures $(odd, 1, 1)$ and parts of the $(odd, 2, 1)$, $(even, 1, 1, 1)$, and $(even, 2, 1, 1)$ subsignatures here because they are required for §3.7 and §3.8. Appendix E discusses signatures where the cases above are subsignatures.

Generally, the idea behind the parallel edges in this signature is to use a different pair of cross edges for an "even number of odd-occurring colors" signatures and an "odd number of odd-occurring colors" signatures. Upon doing so, the cross edges from an "even number of odd-occurring colors" signatures remain intact in an "odd number of odd-occurring colors" signatures. Then if the number of odd-occurring elements increases again, we obtain an "even number of odd-occurring colors" signature where we can use the cross edge of the previous signature as our new parallel edge. The same technique is applied for an "odd number of odd-occurring colors" signatures.

3.12.1.1 Signature $(\boldsymbol{odd}, 1, 1, 1)$

This subgraph leads to the special signature $(odd, 1, 1)$ and $(even, 1, 1, 1)$. The $(odd, 1, 1, 1)$ signature has four subcycles. Three of those have subsignature $(odd, 1, 1)$ and one has subsignature $(even, 1, 1, 1)$. We connect the cycles in the order of Algorithm 3. If $k_0 = 1$, we refer to §3.12.1.3. For $k_0 \geq 3$, we glue the subcycles with parallel edges:

$$20^{k_0-1}310 \sim \underline{02}0^{k_0-2}310 \tag{3.168}$$
$$20^{k_0-1}301 \sim \underline{02}0^{k_0-2}301 \tag{3.169}$$

The edge (3.168) is generated in the $(even, 1, 1, 1)$ signature. We see the edge is guaranteed by (3.87). The parallel edge (3.169) is covered in the $(odd, 1, 1)$ subcycle. By Lemma 1, it exists as part of the d_1 path in (2.2).

The $(odd, 1, 1)$ cycles with trailing 1 to 2 and 2 to 3 are glued using the parallel edges:

$$0^{k_0}123 \sim 0^{k_0-1}\underline{10}23 \tag{3.170}$$
$$0^{k_0}132 \sim 0^{k_0-1}\underline{10}32 \tag{3.171}$$

The node $0^{k_0}12$ of (3.170) with trailing 3 has degree two and is in a Hamiltonian cycle. Therefore, edges $0^{k_0-1}201 \sim 0^{k_0-1}\underline{02}1 \sim 0^{k_0}\underline{12} \sim 0^{k_0-1}\underline{10}2$ must be part of the neighbor-swap graph. By the same argument, the edge (3.171) exists. These parallel edges are used with trailing elements $32/23$. The parallel edges with trailing 12 and 21 exist by a similar argument.

Guaranteed edges

$$0120^{k_0-1}3 \sim 0\underline{21}0^{k_0-1}3 \tag{3.172}$$
$$0130^{k_0-1}2 \sim 0\underline{31}0^{k_0-1}2 \tag{3.173}$$
$$0230^{k_0-1}1 \sim 0\underline{23}0^{k_0-1}1 \tag{3.174}$$

For $k_0 = 1$, we obtain a permutahedron from § 3.12.1.3. In these $(1,1,1,1)$ signatures, we used cross edges $0123 \sim 01\underline{32}$ and $1023 \sim 10\underline{32}$ to glue the cycles with trailing 2 and 3. In the subcycle, 012 has degree two. Therefore, edge (3.172) must be present in signature $(1,1,1,1)$. A similar argument holds for edges (3.173) and (3.174). If $k_0 \geq 3$, we see that the edges are not used as cross edges. So they are generated using Lemma 1. They are generated using the d_{k_0-1} path of (2.2).

$$20^{k_0}31 \sim \underline{02}0^{k_0-1}31 \tag{3.175}$$

Edge (3.175) is generated in the $(odd, 1, 1)$ signature subcycle with trailing 1. In that cycle, it is generated as part of the d_0 path in (2.2) of Lemma 1.

$$0^{k_0}123 \sim 0^{k_0}1\underline{32} \tag{3.176}$$

Edge (3.176) is guaranteed because it is one of the cross edges between (3.170) and (3.171).

$$0^{k_0-1}1230 \sim 0^{k_0-1}1\underline{32}0 \tag{3.177}$$

The edge (3.177) is guaranteed because subsignature $(even, 1, 1, 1)$ covers edge (3.89).

$$0^{k_0-1}1302 \sim 0^{k_0-1}\underline{31}02 \tag{3.178}$$

The edge is generated in the subcycle with trailing 2 and thus the $(odd, 1, 1)$ cycle of Lemma 1. So it is the middle part of the d_1 path (2.2).

$$230^{k_0}1 \sim 2\underline{03}0^{k_0-1}1 \tag{3.179}$$

The edge is generated in the subcycle with trailing 1. In the $(odd, 1, 1)$ cycle of Lemma 1, it is the glued edge between the d_{k_0} and d_{k_0-1} paths (2.2).

$$0^{k_0}132 \sim 0^{k_0}312 \tag{3.180}$$

Edge (3.180) is guaranteed in the subcycle with trailing 2 by the d_0 path (2.2) of Lemma 1.

3.12.1.2 Signature $(odd, 2, 1, 1)$

We can assume $k_0 \geq 3$ since $k_0 = 1$ gives an $(even, 1, 1, 1)$ signature. We will prove the cross edges between the subcycles in the order of Algorithm 3.

$$0^{k_0-1}12310 \sim 0^{k_0-1}1\underline{32}10 \tag{3.181}$$
$$0^{k_0-1}12301 \sim 0^{k_0-1}1\underline{32}01 \tag{3.182}$$

Edge (3.181) is generated by subsignature $(even, 2, 1, 1)$ where it is guaranteed by (3.118). Edge (3.182) is guaranteed by (3.177) in subsignature $(odd, 1, 1, 1)$.

$$0^{k_0}1321 \sim 0^{k_0}\underline{31}21 \tag{3.183}$$
$$0^{k_0}1312 \sim 0^{k_0}\underline{31}12 \tag{3.184}$$

Edge (3.183) is guaranteed in subsignature $(odd, 1, 1, 1)$ by (3.180). Edge (3.184) corresponds to edge (3.21) in subsignature $(odd, 2, 1)$.

$$110^{k_0}32 \sim 1\underline{01}0^{k_0-1}32 \tag{3.185}$$
$$110^{k_0}23 \sim 1\underline{01}0^{k_0-1}23 \tag{3.186}$$

Both edges (3.185) and (3.186) are formed in subsignature $(odd, 2, 1)$. They are guaranteed by edge (3.24).

Guaranteed edges

$$110^{k_0-1}230 \sim 110^{k_0-1}\underline{32}0 \tag{3.187}$$

This edge is guaranteed from the (*even*, $2, 1, 1$) signature by edge (3.117).

$$110^{k_0}23 \sim 110^{k_0}\underline{32} \tag{3.188}$$

Edge (3.188) is guaranteed because it is the cross edge between (3.185) and (3.186).

$$0^{k_0-1}1123 \sim 0^{k_0-1}1\underline{21}3 \tag{3.189}$$

Edge (3.189) is generated by the subsignature with trailing 3, the (*odd*, $2, 1$) signature where it is guaranteed by (3.20).

$$2110^{k_0}3 \sim 1210^{k_0}3 \tag{3.190}$$

$$3110^{k_0}2 \sim 1210^{k_0}2 \tag{3.191}$$

3.12.1.3 Signature all-ones (permutahedron)

It is possible to use the Steinhaus-Johnson-Trotter algorithm [21, 9, 23] for these signatures. However, the goal of our algorithm is to guarantee certain edges. Therefore we will form the cycles by the same inductive argument as before. We fix the trailing element and glue parallel edges in the subcycles. We can choose these parallel edges as:

$$01\ldots(K-4)(K-3)(K-2)(K-1) \sim 01\ldots\underline{(K-3)(K-4)}(K-2)(K-1) \tag{3.192}$$

We will take the base case where $n = K = 4$. The $(1, 1, 1)$ signature gives a single cycle in the form of a hexagon. Gluing the cycles with trailing 0 to 1, 1 to 2, and 2 to 3 gives only one option to combine the subcycles in that order. Every hexagon of signature $(1, 1, 1)$ has exactly the cycle:

$$21\mathbf{0} \sim \underline{12}\mathbf{0} \sim 1\underline{0\mathbf{2}} \sim \underline{01}\mathbf{2} \sim 0\underline{2\mathbf{1}} \sim \underline{20}\mathbf{1}$$

This cycle has two adjacent elements with each trailing element. These are the only options to connect the disjoint cycle cover of the $(1, 1, 1, 1)$ signature. One can also choose a different order to combine the meta-graph of cycles to obtain a different cycle. However, we follow the order of Algorithm 3. This gives the parallel edges:

$2310 \sim \underline{32}10$	$0321 \sim \underline{30}21$	$1023 \sim \underline{01}23$
$3201 \sim \underline{23}01$	$3012 \sim \underline{03}12$	$0132 \sim \underline{10}32$

The general form of the cross edge is:

$$01\ldots(K-2)(K-1)x \sim 01\ldots\underline{(K-1)(K-2)x} \tag{3.193}$$

$$01\ldots(K-2)(K-1) \sim 01\ldots\underline{(K-1)(K-2)} \tag{3.194}$$

To clarify; $k_x = 1$ in the signature of (3.193), i.e. it doesn't occur between elements 0 and $K-1$. For $n = K = 3$, the edge with trailing 0 guarantees (3.193). The edges with trailing 1 and 2 are special cases of (3.193):

$$01\ldots(K-3)(K-1)(K-2) \sim 01\ldots\underline{(K-1)(K-3)}(K-2) \tag{3.195}$$

$$01\ldots(K-3)(K-2)(K-1) \sim 01\ldots\underline{(K-2)(K-3)}(K-1) \tag{3.196}$$

These edges are in essence the same as (3.193). The edge $012 \sim 021$ is the edge (3.194) in the $(1, 1, 1)$ signature. The neighbor-swap graph for $n = K = 4$ also admits these edges. By

(3.193), we can add another trailing element to this signature and create the parallel edges (3.197) and (3.198):

$$01\ldots(K-2)(K-1)xy \sim 01\ldots\underline{(K-1)(K-2)}xy \tag{3.197}$$
$$01\ldots(K-2)(K-1)yx \sim 01\ldots\underline{(K-1)(K-2)}yx \tag{3.198}$$

These cross edges create the guaranteed edge (3.194) when $x = K-1$ and $y = K-2$. Moreover, this leaves intact edge (3.194) with a new trailing element x, therefore creating (3.193).

Therefore we have shown that the neighbor-swap graph of a signature with only colors that occur once contains a Hamiltonian cycle with the guaranteed edges below.

Guaranteed edges
The guaranteed edges are (3.193) and (3.194).

3.12.1.4 "Odd number of odd-occurring colors" signatures

We split this case into the three odd - rest even signatures and then explain the inductive step to signatures with an "odd number of odd-occurring colors" with five or more odd-occurring colors. We will distinguish two cases depending on the parity of the fixed trailing element. The parity of the trailing element can differ between the two subcycles that we combine.

A three odd - rest even signature has exactly three subgraphs with two odd - rest even signatures and a fixed trailing odd-occurring element. If $K \geq 4$, at least one two odd - rest even signature subcycle must be connected to a subsignature with four odd-occurring colors with a fixed trailing even-occurring element. If $K \geq 5$, the remaining cycles have four odd-occurring colors and a fixed even-occurring element. Two odd - rest even subsignatures cover the edge (3.164), even the special cases. Two edges conform to (3.164); one with trailing xy and one with xz. If we add an extra trailing odd-occurring element to both of the permutations of (3.164), we obtain the edges:

$$(K-1)^{k_{K-1}}\ldots 0^{k_0}xyz \sim (K-1)^{k_{K-1}}\ldots 1^{k_1-1}\underline{01}0^{k_0-1}xyz \tag{3.199}$$
$$(K-1)^{k_{K-1}}\ldots 0^{k_0}xzy \sim (K-1)^{k_{K-1}}\ldots 1^{k_1-1}\underline{01}0^{k_0-1}xzy \tag{3.200}$$

The dots $(\ldots)$ in these edges represent all elements from $(K-2)^{k_{K-2}}$ up to and including 1^{k_1}; with k_x-1 elements x, k_y-1 elements y, and k_z-1 elements z. All elements between $(K-1)^{k_{K-1}}$ and 0^{k_0} are ordered in reverse corresponding to the signature. We also require this for the remainder of this section. If we write $0^{k_0}\ldots(K-1)^{k_{K-1}}$ they are ordered in the same way as the signature.

Both these edges are present in a three odd - rest even signature with one fixed trailing element that occurs an odd number of times. This already solves the base case (odd, odd, odd).

If $K \geq 4$, we must connect at least one of the two odd - rest even signature subcycles to one of the subcycles on a signature with four odd-occurring colors. We say k_x, k_y, and k_z are odd and k_u and k_v are even. We pick the two trailing elements v and z. So we obtain the edges:

$$(K-1)^{k_{K-1}}\ldots u^{k_u}\ldots v^{k_v-2}\ldots 0^{k_0}vxyzv \sim (K-1)^{k_{K-1}}\ldots u^{k_u}\ldots v^{k_v-2}\ldots 0^{k_0}v\underline{yx}zv \tag{3.201}$$
$$(K-1)^{k_{K-1}}\ldots u^{k_u}\ldots v^{k_v-2}\ldots 0^{k_0}vxyvz \sim (K-1)^{k_{K-1}}\ldots u^{k_u}\ldots v^{k_v-2}\ldots 0^{k_0}v\underline{yx}vz \tag{3.202}$$

We can see that (3.201) is generated in a signature with four odd-occurring colors k_x, k_y, k_z, and k_v-1. Within this cycle, the edge has four trailing odd-occurring elements $vxyz$. So we can see that the edge (3.201) is guaranteed by (3.218).

The edge (3.202) contains two odd-occurring elements. Fig. 3.10 shows that this subcycle is generated in a three odd - rest even subsignature with an even-occurring trailing element v. This three odd - rest even signature has trailing elements vxy with k_v, k_x, and k_y odd. Therefore, we know that (3.202) must exist in that subgraph by (3.207).

Next, we discuss what happens when both trailing elements are of even parity. So u and v are the trailing two elements of the cycles we are connecting. We use the cross edges;

$$(K-1)^{k_{K-1}}\dots u^{k_u-2}\dots v^{k_v-2}\dots 0^{k_0}uvxyzuv \sim (K-1)^{k_{K-1}}\dots u^{k_u-2}\dots v^{k_v-2}\dots 0^{k_0}uv\underline{yx}zuv \quad (3.203)$$

$$(K-1)^{k_{K-1}}\dots u^{k_u-2}\dots v^{k_v-2}\dots 0^{k_0}uvxyzvu \sim (K-1)^{k_{K-1}}\dots u^{k_u-2}\dots v^{k_v-2}\dots 0^{k_0}uv\underline{yx}zvu \quad (3.204)$$

Both the edges (3.203) and (3.204) are generated in an "even number of odd-occurring colors" signatures. Both permutations have an even-occurring trailing element in these signatures. Edge (3.203) is guaranteed by (3.219) and edge (3.204) is guaranteed by (3.220).

The inductive argument for an "odd number of odd-occurring colors" signatures follows from these three odd - rest even signatures. If we increase the number of odd-occurring colors by one, we obtain an "even number of odd-occurring colors" signature. This uses a different cross edge as we show below. Adding another odd-occurring element, results in a signature with five odd-occurring colors so it is an "odd number of odd-occurring colors" signature. To glue that disjoint cycle cover we take the cross edges of the three odd - rest even signature and use them as our new parallel edges. This is a guaranteed edge by (3.207).

However, we must consider the signatures where two trailing elements have the same frequencies. Assume these equal trailing elements are $k_0 = k_1$. By design of Algorithm 3, we glue the cycles with trailing 0 and 1. This results in a subgraph where the required edge (3.164), (3.203), or (3.207) is not available because it is used as a cross edge in a two odd - rest even subsignature or a different cross edge was used in a three or more odd - rest even signature. In general, we tackle this problem by translating (Definition 9) one of the subsubsignatures before connecting those into a Hamiltonian cycle on the subsignature. The subsignature that we change is the one with a trailing element of k_0. This was generated as an $(k_1, k_0-1, \dots)$ subsignature. In this subsignature, we translate the subsubcycle with a trailing element of k_1 such that the elements of k_0 and k_1 are swapped. This is possible because $k_0 - 1 = k_1 - 1$ in the subsubcycle. Then we add a trailing element of k_1 and generate the subcycle with the technique for a two odd - rest even or an "even number of odd-occurring colors" signature. The subsubcycle with a trailing element of k_1 is connected to the subsubcycle with a trailing element of $k_0 - 1$ in this subsignature. The latter subcycle did not change so it can still be connected to the subcycle with trailing k_2. Consequently, we obtained a subcycle that has the required edge and which we use to glue the subcycles with trailing 0 and 1 (where the latter subcycle did not change). Examples of this procedure are given in Appendix E.2 and Appendix E.8. This technique remains valid since the parallel edges that we use to glue an "even number of odd-occurring colors" signature are never the same as those for an "odd number of odd-occurring colors" signature.

Guaranteed edges

With exactly three odd-occurring elements k_x, k_y, and k_z:

$$0^{k_0}\dots(K-1)^{k_{K-1}}x^{k_x}y^{k_y}z \sim 0^{k_0}\dots(K-1)^{k_{K-1}}x^{k_x-1}\underline{yx}y^{k_y-1}z \quad (3.205)$$

The existence of the edge (3.205) directly follows from (3.163). The edge (3.163) has the two odd colors k_x and k_y. We add a new element z to make k_z odd in the three odd - rest even signature cycle. This edge is not used as a cross edge in the three odd - rest even signature. Note that we could have $k_x = 1$ and $k_y = 1$. Moreover, signature $(\mathit{even}, 1, 1, 1)$ guarantees edge (3.91). Signature $(\mathit{odd}, \mathit{odd}, 1)$ covers edges (3.40), (3.41), (3.44), (3.46), (3.48), and (3.51).

$$x^{k_x}0^{k_0}\dots(K-1)^{k_{K-1}}yz \sim x^{k_x-1}\underline{0x}0^{k_0-1}\dots(K-1)^{k_{K-1}}yz \quad (3.206)$$

This edge is generated in the two odd - rest even signature where it is guaranteed by (3.166). It is not used as a cross edge in the three odd - rest even signature, thus we can guarantee it here. Moreover, signature $(odd, odd, 1)$ guarantees (3.48), and (3.51). Signature $(even, 1, 1, 1)$ also guarantees the edges by (3.85).

Now assume we have an "odd number of odd-occurring colors" signature with three or more odd-occurring colors. The odd-occurring colors are $k_w, k_x, k_y, k_z, \dots, k_C$ with $|D| \geq 3$ the number of odd-occurring colors. Let $C \dots zyxw$ indicate one occurrence of every odd-occurring color. If an odd color occurs more than once, we add the remaining even number of elements to the even part of the permutations below, i.e. the $0^{k_0} \dots (K-1)^{k_{K-1}}$ sequence. Assume the colors k_u and k_v are even.

$$(K-1)^{k_{K-1}} \dots 0^{k_0} C \dots zyxw \sim (K-1)^{k_{K-1}} \dots 0^{k_0} C \dots zy\underline{wx} \tag{3.207}$$

We defined above that we use the cross edge of an "odd number of odd-occurring colors" signatures where the node is generated as a stutter permutation and has trailing elements that occur an odd number of times. If $|D| = 3$, we obtain the first cross edge between (3.199) and (3.200). If we add one trailing odd-occurring element, we obtain the edge (3.218). If we add another odd-occurring trailing element, we obtain two such edges. Both with two new odd-occurring trailing elements s and t in orders st and ts. We use the edges of (3.218) with trailing s and t as the new parallel edges. This gives the cross edge (3.207) with trailing st/ts.

Moreover, the subsignature can be one that was addressed as a special case. The special signatures with two odd-occurring colors cover edge (3.164) as well. Signature $(odd, odd, 1)$ guarantees the edge by (3.50). Signature $(even, 1, 1, 1)$ guarantees edges (3.89) and (3.90).

$$w^{k_w} 0^{k_0} \dots (K-1)^{k_{K-1}} C \dots xyz \sim w^{k_w} 0^{k_0} \dots (K-1)^{k_{K-1}} C \dots \underline{yx}z \tag{3.208}$$

For edge (3.208) we require $|D| \geq 5$ because $|D| = 3$ obtains edge (3.206). We take the largest odd-occurring color k_w and put it at the start of the permutation. If we have five odd-occurring colors, we know the edge with trailing z is not used as a cross edge in an "odd number of odd-occurring colors" signatures. So we reduce it to an "even number of odd-occurring colors" signature where the edge is guaranteed by (3.217). Therefore, we can add an odd-occurring trailing element to guarantee this edge.

$$w^{k_w} 0^{k_0} \dots (K-1)^{k_{K-1}} vC \dots zyxv \sim w^{k_w} 0^{k_0} \dots (K-1)^{k_{K-1}} vC \dots z\underline{xy}v \tag{3.209}$$

Again, we require $|D| \geq 5$ because $|D| = 3$ has not generated guaranteed edge (3.217) yet. This edge has a trailing element v with k_v even. Because this edge is not used as a cross edge in an "odd number of odd-occurring colors" signatures, we can fix it as the trailing element. This results in an "even number of odd-occurring colors" signature where the edge is guaranteed by (3.217).

$$w^{k_w} 0^{k_0} \dots (K-1)^{k_{K-1}} Cv \dots zyxv \sim w^{k_w} 0^{k_0} \dots (K-1)^{k_{K-1}} Cv \dots z\underline{xy}v \tag{3.210}$$

Again, we require $|D| \geq 5$. Edge (3.210) is generated in a subsignature with one trailing even-occurring trailing element. This element is fixed since it is not used as a cross edge in an "odd number of odd-occurring colors" signature. The subsignature with an "even number of odd-occurring colors" guarantees the edge by (3.217).

3.12.1.5 "Even number of odd-occurring colors" signatures

Firstly, we address signatures with $K = 4$ with all colors having odd frequencies. This results in four subgraphs with a three odd - rest even signature. The subgraphs of special cases follow from § 3.12.1.4. For the signature with four odd-occurring colors, we glue parallel edges with k_w, k_x, k_y, and k_z odd:

$$x^{k_x}0^{k_0}\dots(K-1)^{k_{K-1}}yzw \sim x^{k_x-1}\underline{0x}0^{k_0-1}\dots(K-1)^{k_{K-1}}yzw \tag{3.211}$$

$$x^{k_x}0^{k_0}\dots(K-1)^{k_{K-1}}ywz \sim x^{k_x-1}\underline{0x}0^{k_0-1}\dots(K-1)^{k_{K-1}}ywz \tag{3.212}$$

The dots ($\dots$) in these edges represent all elements from 1^{k_1} up to and including $(K-2)^{k_{K-2}}$; with k_y-1 elements y, k_z-1 elements z, and k_w-1 elements w and without any x. All elements between 0^{k_0} and $(K-1)^{k_{K-1}}$ are ordered corresponding to the signature. We also require this for the remainder of this section. If we write $(K-1)^{k_{K-1}}\dots 0^{k_0}$ they are ordered in reverse corresponding to the signature.

In the three odd - rest even signatures, both of these edges are guaranteed by (3.208).

If $K \geq 5$, we can obtain permutations with trailing even-occurring colors and four odd-occurring colors. For this, we will assume k_w, k_x, k_y, and k_z are odd and k_u and k_v are even. We have to combine one of the cycles with an odd-occurring trailing element to a cycle with an even-occurring trailing element. We pick the cycles with trailing z and v:

$$x^{k_x}0^{k_0}\dots(K-1)^{k_{K-1}}vwyvz \sim x^{k_x}0^{k_0}\dots(K-1)^{k_{K-1}}v\underline{yw}vz \tag{3.213}$$

$$x^{k_x}0^{k_0}\dots(K-1)^{k_{K-1}}vwyzv \sim x^{k_x}0^{k_0}\dots(K-1)^{k_{K-1}}v\underline{yw}zv \tag{3.214}$$

The cycle with a trailing z is generated in an "odd number of odd-occurring colors" signature. This edge is guaranteed by (3.209) because it has an even-occurring trailing element v. Edge (3.214) is generated in an "odd number of odd-occurring colors" signature as well. This edge is guaranteed by (3.208).

Lastly, we prove the existence of parallel edges for cycles with two even-occurring, trailing elements. Those are;

$$x^{k_x}0^{k_0}\dots(K-1)^{k_{K-1}}uvzywvu \sim x^{k_x}0x0^{k_0-1}\dots(K-1)^{k_{K-1}}uvz\underline{wy}vu \tag{3.215}$$

$$x^{k_x}0^{k_0}\dots(K-1)^{k_{K-1}}uvzywuv \sim x^{k_x}0x0^{k_0-1}\dots(K-1)^{k_{K-1}}uvz\underline{wy}uv \tag{3.216}$$

Both edges (3.215) and (3.216) are generated in a subcycle with an "odd number of odd-occurring colors" signatures. These signatures guarantee the required edges with (3.209).

If the signature contains six or more odd-occurring colors and the number of odd-occurring colors is even, we must find the parallel edges. We can use the previously generated cross edges because they are guaranteed in an "odd number of odd-occurring colors" signatures by (3.208), (3.209), and (3.210). By adding one more trailing odd-occurring element we obtain an "even number of odd-occurring colors" signature with two distinct trailing elements. Similar to an "odd number of odd-occurring colors" signature, we can use this previous cross edge as the new parallel edge. Signatures, where the trailing elements have the same number of occurrences, require slightly more attention again. The subsubsignature glued slightly different parallel edges if the ordering of the colors was different. However, we can use the same technique as for an "odd number of odd-occurring colors" signature where the two trailing elements are of equal frequencies by translating one of the subsubsignature neighbor-swap graphs. An example of this is explained in Appendix E.6. Again, this technique persists for larger signatures since an "odd number of odd-occurring colors" signature uses different cross edges than an "even number of odd-occurring colors" signature.

Guaranteed edges

For the following signatures, we assume that we have an "even number of odd-occurring colors". The odd colors, in non-decreasing order, are $k_w, k_x, k_y, k_z, \dots, k_C$ with $|E| \geq 4$ the number of odd-occurring colors.

$$w^{k_w}0^{k_0}\dots(K-1)^{k_{K-1}}C\dots zyx \sim w^{k_w}0^{k_0}\dots(K-1)^{k_{K-1}}C\dots z\underline{xy} \tag{3.217}$$

To clarify; the "$C\dots zyx$" part of these permutations represents one element of each odd-occurring element except for w. In a signature with four odd-occurring elements, this edge is one of the cross edges between (3.211) and (3.212). If we add one odd-occurring trailing element, this results in an "odd number of odd-occurring colors" signature. That subgraph guarantees the edge by (3.208). If we add one more trailing element, we use this edge as the parallel edge to glue the subcycles with trailing x and y as explained above.

$$(K-1)^{k_{K-1}}\dots 0^{k_0}C\dots zyxw \sim (K-1)^{k_{K-1}}\dots 0^{k_0}C\dots z\underline{xy}w \tag{3.218}$$

This edge is generated in an "odd number of odd-occurring colors" signatures by edge (3.207). The edge is not used as a cross edge in an "even number of odd-occurring colors" signatures.

$$(K-1)^{k_{K-1}}\dots 0^{k_0}vC\dots zyxwv \sim (K-1)^{k_{K-1}}\dots 0^{k_0}vC\dots zy\underline{wx}v \tag{3.219}$$

The edge (3.219) is generated in a subcycle with an "odd number of odd-occurring colors" signature since k_v is even. We can fix the trailing element v because it is not used as a cross edge. This results in the edge (3.207).

$$(K-1)^{k_{K-1}}\dots 0^{k_0}Cv\dots zyxwv \sim (K-1)^{k_{K-1}}\dots 0^{k_0}Cv\dots zy\underline{wx}v \tag{3.220}$$

The edge (3.219) is generated in a subcycle with an "odd number of odd-occurring colors" signature since k_v is even and it is not used as a cross edge in an "even number of odd-occurring colors" signature. Therefore it is guaranteed by (3.207).

Therefore we can conclude that the neighbor-swap graphs of three or more odd - rest even signatures contain a Hamiltonian cycle.

Chapter 4

Software

To support the proof that all neighbor-swap graphs admit a Lehmer path the author developed a software project that is publicly available. Moreover, the author published the documentation.

The algorithms are programmed in Python to ease the understanding of the code and for its simplicity during development. Some general remarks are that we to use typing as much as possible. The signatures are typed as tuples to allow for caching of Hamiltonian cycles in subsignatures. For this reason they are not typed as lists, which are mutable and cannot be hashed; hashing is a requirement for caching.

In this section, we address the relevant parts of the software project. Firstly, we explain the implementation of the algorithms from the existing literature. We address the author's implementation of Stachowiak's work [19] and Verhoeff's binary algorithm [24]. Continuing, we discuss the implementation of the disjoint cycle cover. Then we explain how we combine the disjoint cycles. Lastly, we will describe how the stutter permutations are added to form a Lehmer path.

The software was used throughout the Master Graduation Project to develop the proof of Conjecture 1. The software does not support all cases from Chapter 3. The three or more odd - rest even signatures (§3.12) lack an implementation concerning the generation of cross edges. This will be discussed in more depth in §4.3.

4.1 Stachowiak's algorithm

Our code contains an implementation of Stachowiak's theorem [19] for Hamiltonian cycles in signatures with two or more odd-occurring colors. The theorem is defined for the poset setting. However, we only require the multiset setting so we can rephrase the theorem to:

If a neighbor-swap graph of signature S has an even number of permutations, with the number of permutations strictly greater than two, and contains a Hamiltonian path, then the neighbor-swap graph can be extended to a Hamiltonian cycle by introducing new colors while preserving the original color occurrences of S.

Part of the implementation is based on A. Smerdu's work [18]. The author altered this project because it has different typing for permutations. We use tuples of numbers instead of lists of strings. The implementation of Lemma 1 is used to obtain a Lehmer path in some cases as can be seen in Chapter 3. The other lemmas of Stachowiak are also implemented but are not used in the Hamiltonian cycles in neighbor-swap graphs. However, they are tested to contain Hamiltonian cycles and the software supports Stachowiak's theorem.

Stachowiak's theorem takes a Hamiltonian path in a neighbor-swap graph as its input. For this, we use a Hamiltonian path that is generated by Theorem 1 (see §4.2). It then takes an edge of this Hamiltonian path and interleaves the new color through the two adjacent permutations to obtain a subcycle. The color must be new to allow for neighbor swaps with the permutations of the Hamiltonian path. These subcycles of two adjacent permutations are ultimately combined into a single Hamiltonian cycle.

4.2 Verhoeff's algorithms

We resorted to Verhoeff's original implementation in Mathematica for the implementation of Theorem 1. The author converted this to Python. Our implementation generates slightly different paths than the original implementation for the "square tube" paths shown in Fig. 4.1. However, altering the order of these edges still results in a valid Hamiltonian path.

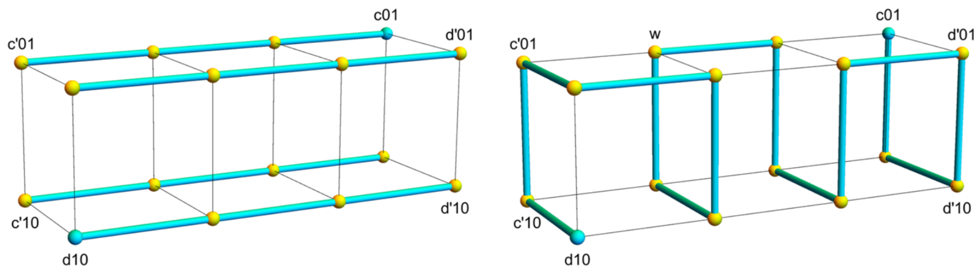


Figure 4.1: Four doubly parallel paths (*left*) combined into one "square tube" path (*right*). Figure from Verhoeff [24].

Continuing, the author implemented Theorem 2. Originally, this was done with Stachowiak's theorem for signatures with two or more odd-occurring colors. The implementation was based on foundations by Verhoeff [24]. This included recursive functions for $(\mathit{even}, 2, 1)$ signature Hamiltonian cycles and $(\mathit{odd}, 2, 1)$ signature Hamiltonian path. The new implementation uses an iterative generation of a part of the path in the neighbor-swap graph of signature $(\mathit{odd}, 2, 1)$ instead, similar for the Hamiltonian cycle. The Hamiltonian path of signature $(\mathit{even}, 1, 1)$ is implemented with two distinct methods from § 3.4 and Lemma 1, which result in the same path. We tested that the disjoint cycle cover contains all permutations of the input signature and whether the individual cycles are permutations that are adjacent by a neighbor swap. The disjoint cycle cover is typed as a list of unknown depth with a list of tuples that represent the permutations at the lowest level. The lists of unknown depth arise because the all-even signatures have all-even subsignatures. To remove this depth, the disjoint cycle cover in all-even signatures must be flattened into a single Hamiltonian cycle on the non-stutter permutations.

4.3 Combining the Disjoint Cycle Cover

To glue the disjoint cycle cover of Theorem 2, the order of the cycles is important as discussed before. Therefore the author developed Algorithm 2 and Algorithm 3. The disjoint cycle cover is generated such that the order matches the output of these algorithms. The general idea was to glue the disjoint cycles together such as in Fig. 3.2. This requires that we find parallel edges in the subcycles. The author wrote code that naively looks for these parallel edges. This is done by filtering them on their trailing elements, and comparing whether the edges are parallel. If two edges are parallel, they are never parallel to edges in other subcycles as stated in §3.1.

These edges were analyzed to find a general form of the parallel edges. However, Stachowiak's theorem does not guarantee edges as mentioned before. Therefore we opted to generate a disjoint cycle cover for two odd - rest even and three or more odd - rest even signatures as well. By generating the disjoint cycle cover and connecting it with the technique of Fig. 3.2, the Hamiltonian cycles have at most two places where the trailing element changes. This is beneficial when trying to connect the cycle for a larger signature. Stachowiak's algorithm has more swaps of trailing elements. With the subcycles of our solution, we keep the position of the last element fixed for a long time before moving on to the next cycle. This means that for some applications Stachowiak's solution might be more beneficial while if the last elements are known, our algorithm is better. Note that the technique we use to connect two odd - rest even signatures such as in Fig. 3.1 has more swaps as well. However, we always know what parallel edges are used as cross edges and thus which edges remain intact.

The author investigated whether the existence of parallel edges persists in larger signatures. By analyzing the edges, a general form of edges was found for every type of signature as explained in Chapter 3. This concluded the exploratory phase of the software development.

Continuing, an attempt was made to efficiently generate the Hamiltonian cycles on the non-stutter permutations. The author implemented this for all-even, one odd - rest even, and two odd - rest even signatures. The three or more odd - rest even signatures are not fully implemented yet. This has two reasons; firstly, the complex operations when generating the Hamiltonian cycles for three or more odd - rest even signatures where two trailing elements have equal frequencies. We refer to the operation where one of the subsubgraphs has to be translated (Definition 9) before it is glued to the other subsubcycles. Examples of this are shown in Appendix E.2, Appendix E.6, and Appendix E.8. The other parallel edges require some ordering in the odd-occurring elements. This is to obtain the previously used cross edge. An implementation of this sorting was not yet achieved by the author and is the incomplete part of the programming project.

For other signatures with at most two odd-occurring elements, the author implemented the generation of a Hamiltonian cycle on the non-stutter permutations. The cycles are examined to contain the correct permutations such that successive permutations differ by a single neighbor swap. Moreover, the Hamiltonian cycles are tested to contain the guaranteed edges. The three or more odd - rest even signatures are also tested for signatures where the parallel edges follow a simple structure.

The implementation of the three or more odd - rest even signatures breaks on signatures $(2,2,1,1,1)$ and $(3,3,3,2)$. The former does not result in a Hamiltonian cycle because the subsignature $(1,2,1,1,1)$ does not guarantee the correct cross edge since we do not translate subsubsignature $(1,1,1,1,1)$. Therefore the code fails when attempting to glue the subcycles with trailing 0 and 1. Signature $(3,3,3,2)$ glues an incorrect parallel edge that was not used as a cross edge for the subcycles with trailing 2 and 3. For these three or more odd - rest even signatures, we added a naive method to generate the Hamiltonian cycles to show that the neighbor-swap graphs contain Hamiltonian cycles on the non-stutter permutations. This implementation also provides valid outputs for larger signatures that contain subgraphs with these breaking signatures.

4.4 Lehmer Paths

The original goal of this work was to prove the existence of Lehmer paths. To do this we also implemented the code to incorporate the stutter permutations as single spurs in the neighbor-swap graphs. This code is straightforward because it loops over the list and adds a stutter permutation when a permutation is found that is adjacent to this stutter permutation. Moreover, we guarantee the minimum length of this Lehmer path by rotating the neighbor-swap graph to a permutation that is adjacent to a stutter permutation and starting the path with a stutter permutation. This guarantees that the path has length $M(S)+d(G)-1$, viz. the multinomial coefficient (2.1) plus the defect (Definition 2) minus one because we start at the stutter permutation. This means that we use the minimum number of edges to traverse a Lehmer path. This is also tested for the neighbor-swap graphs for which we can generate a Hamiltonian cycle on the non-stutter permutation.

Chapter 5

Conclusion

In Chapter 3 we have shown that the neighbor-swap graphs of non-stutter permutations admit a Hamiltonian cycle or path according to Conjecture 2. Therefore we have proven the conjecture and can now state Theorem 3:

Theorem 3. *For every neighbor-swap graph, the subgraph consisting of its non-stutter permutations admits a Hamiltonian path. Furthermore, there even exists a Hamiltonian cycle on the non-stutter permutations, except when*

1. *the signature arity is zero or one, or*
2. *the signature is binary, and at least one of the k_i is odd, or*
3. *the signature is a permutation of $(2k, 1, 1)$.*

By Verhoeff's theorem [24] (stated in §2.3.3), this also shows that Conjecture 1 holds. So we conclude that we have proven Theorem 4:

Theorem 4. *A Lehmer path can be constructed for every neighbor-swap graph.*

While we have proven Theorem 4, an implementation of our solution remains an open project. In Chapter 4, we already discussed that our software project was mainly used for exploration within the neighbor-swap graphs. However, it is still lacking in the implementation for three or more odd - rest even signatures. Moreover, the running time of an algorithm that generates Lehmer paths or cycles is still unknown. We believe that such an algorithm should be able to have a running time that is linear or worst case quadratic in the number of permutations. However, this heavily depends on the data types for cycles/paths and permutations. An algorithm that applies a technique that resembles our proof will result in a polynomial asymptotic running time.

The author found the proof of Theorem 3 based on Theorem 2. The general idea was to combine the disjoint cycle cover by connecting the subcycles using cross edges. The author started by choosing an order for this disjoint cycle cover. Following, an implementation was made to identify parallel edges to glue the disjoint cycle cover into a Hamiltonian cycle. The original proof of the disjoint cycle cover relies on Stachowiak's proof [19] for signatures with two or more odd-occurring colors. The author chose to adapt these signatures to follow a proof similar to that of the disjoint cycle cover because there are no guaranteed edges in Stachowiak's theorem. This adaptation required introducing additional base cases, as neighbor-swap graphs of some subsignatures only admit a Hamiltonian path. The proofs for these cases followed naturally with Lemma 5. Therefore the author has also introduced a new way of proving that multisets of signatures with two or more odd-occurring colors admit a Hamiltonian cycle (except for signature $(\mathit{even}, 1, 1)$). Moreover, this is also novel proof for the generation of all permutations of n (the length of the permutation) distinct elements using only neighbor swaps.

Consequently, the general signatures remained. An attempt was made to find a general pair of parallel edges. However, we add one or two fixed trailing elements but a pair of parallel edges requires one more fixed trailing element. The extra trailing element is not fixed in the subsignature because a general parallel edge implies it was used as a cross edge before. Thus a general pair of parallel edges was refuted. To guarantee different parallel edges the author chose a split into various types of signatures, depending on the number of colors with odd frequencies. The software project aided in finding the parallel edges for each signature type. The all-even and one odd - rest even signatures were fairly trivial but the signatures with two or more odd-occurring colors required more in-depth analysis. Ultimately, a solution was discovered where two odd - rest even signatures glue parallel edges that were established as cross edges in two odd - rest even subsignatures. Three or more odd - rest even signatures follow an inductive proof where the number of odd-occurring colors alternates to ensure parallel edges, relying on cross edges derived from smaller neighbor-swap graphs. The general methodology of our proof is straightforward, however, the complexity was found in details (i.e. proving the existence of the parallel edges). This complexity also becomes apparent by translations of certain subsubsignatures such as explained in §3.12.1.4 and §3.12.1.5.

We have presented one of the solutions for Theorem 4. However, this is not the only proof of this problem. An example that more solutions are possible can be found in three or more odd - rest even signatures. In §3.12 we presented the use of parallel edges that we used as cross edges in neighbor-swap graphs of previous three or more odd - rest even signature subcycles. We chose the edge that most closely resembled a stutter permutation although other edges could also have been a solution that correctly generated the three or more odd - rest even signature Hamiltonian cycles.

The changes that would enhance our proof of Theorem 3 involve simplifications. We distinguished twelve cases in Chapter 3. Reducing this to fewer cases would generalize the generation of Hamiltonian cycles in neighbor-swap graphs on non-stutter permutations. Additionally, there might be an alternative approach to Theorem 4 that does not rely on stutter permutations. Nevertheless, we have successfully established a proof for both Theorem 3 and Theorem 4. This result contributes to the field of Hamiltonian paths neighbor-swap graphs and combinatorial generation.

Bibliography

[1] Armen S. Asratian, Tristan M. J. Denley, and Roland Häggkvist. Bipartite Graphs and their Applications. *Bipartite Graphs and their Applications*, 7 1998. 4, 10

[2] Jørgen Bang-Jensen, Gregory Gutin Digraphs Theory, and Berlin Heidelberg NewYork London Paris Tokyo Hong Kong Barcelona Budapest. Springer-Verlag. Technical report, 2007. 4

[3] Gijs Bellaard. Constructing Shortest Covering Walks of Neighbour-swap Graphs. Technical report, 2019. 5

[4] James R Bitner, Gideon Ehrlich, and Edward M Reingold. Efficient Generation of the Binary Reflected Gray Code and Its Applications. 1976. 6

[5] Peter Eades and Brendan McKay. An algorithm for generating subsets of fixed size with a strong minimal change property. *Information Processing Letters*, 19(3):131–133, 1984. 3, 5

[6] Mahmoud El-Hashash and Heidi Burgiel. The Permutahedron π_n is Hamiltonian. Technical Report 1, 2009. 3

[7] T. C. Hu and B. N. Tien. Generating Permutations with Nondistinct Items. *The American Mathematical Monthly*, 83(8):629, 10 1976. 5

[8] Selmer M. Johnson. Iterative Solution of Linear Systems of Functional Equations. Technical Report 2, 1962. 3

[9] Selmer M. Johnson. Generation of Permutations by Adjacent Transposition. *Mathematics of Computation*, 17(83):282, 7 1963. 52

[10] Donald Ervin Knuth. *The art of computer programming - Generating all tuples and permutations*, volume 4 Fascicle 2. Addison-Wesley, Upper Saddle River, NJ ; London, 2005. 3

[11] James Korsh and Seymour Lipschutz. Generating Multiset Permutations in Constant Time. *Journal of Algorithms*, 25(2):321–335, 11 1997. 6

[12] Derrick H. Lehmer. Permutation by Adjacent Interchanges. Technical Report 2, 1965. 1, 4, 5, 67

[13] Arturo Merino, Namrata, and Aaron Williams. On the Hardness of Gray Code Problems for Combinatorial Objects. 1 2024. 3, 4

[14] Torsten Mütze. Combinatorial Gray codes-an updated survey. 2 2022. 3

[15] Hans Jakob Rivertz. Multiset permutation generation by transpositions. 9 2023. 5, 68

[16] Frank Ruskey. Generating Linear Extensions of Posets by Transposition. *Journal of Combinatorial Theory*, 54(B):77–101, 1992. 4, 5

[17] Frank Ruskey. Combinatorial Generation. *Preliminary working draft. University of Victoria, Victoria, BC, Canada*, 11:425–520, 2003. 3

[18] Ana Smerdu. Hamiltonian cycles in neighbor-swap graphs. Technical report, University of Ljubljana, Ljubljana, 2019. 5, 59

[19] Grzegorz Stachowiak. Hamilton Paths in Graphs of Linear Extensions for Unions of Posets. Technical report, 1992. 2, 4, 5, 6, 12, 13, 18, 59, 63

[20] Grzegorz Stachowiak. On a long cycle in the graph of all linear extensions of a poset consisting of two disjoint chains. *Discrete Mathematics*, 131(1-3):375–378, 8 1994. 5

[21] Hugo Steinhaus. *One Hundred Problems In Elementary Mathematics*. Dover, New York, 1979. 3, 52

[22] Tadao Takaoka. An O(1) Time Algorithm for Generating Multiset Permutations. Technical report, 1999. 6

[23] Hale F. Trotter. Algorithm 115: Perm. *Communications of the ACM*, 5(8):434–435, 8 1962. 3, 52

[24] Tom Verhoeff. The spurs of D. H. Lehmer: Hamiltonian paths in neighbor-swap graphs of permutations. *Designs, Codes, and Cryptography*, 84(1-2):295–310, 7 2017. vii, viii, 2, 4, 5, 9, 10, 11, 12, 13, 15, 16, 18, 19, 20, 23, 24, 37, 40, 44, 59, 60, 63

[25] Aaron Williams. Loopless Generation of Multiset Permutations using a Constant Number of Variables by Prefix Shifts. *SODA '09: Proceedings of the twentieth annual ACM-SIAM symposium on Discrete algorithms*, pages 987–996, 2009. 5

[26] Aaron Williams. The Coolest Way to Generate Combinations. Technical report, 2009. 5

Appendix A

Related work appendices

A.1 Lehmer's algorithm

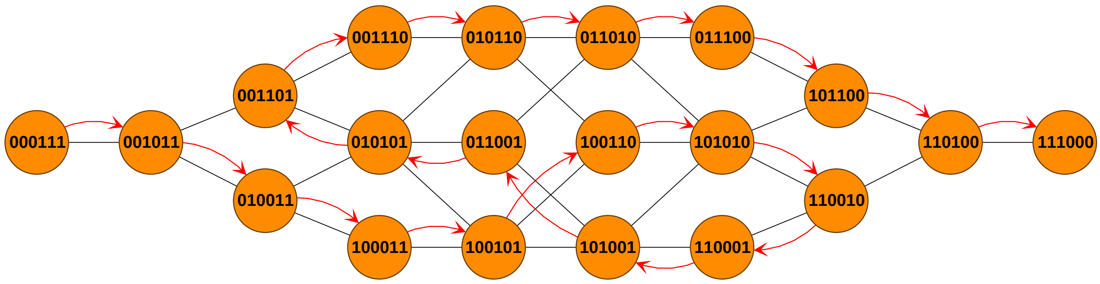
Below is a modern version of Lehmer's algorithm [12]. As stated in Chapter 2, the algorithm provides valid Lehmer paths for multiset permutations of up to a total length of 7. For $n \geq 8$, the solutions are invalid. Moreover, the algorithm works based on lexicographical values so it will fail for smaller signatures when they are unsorted.

Algorithm 1: Lehmer's Hamiltonian Path Procedure for Permutations by Adjacent Interchanges

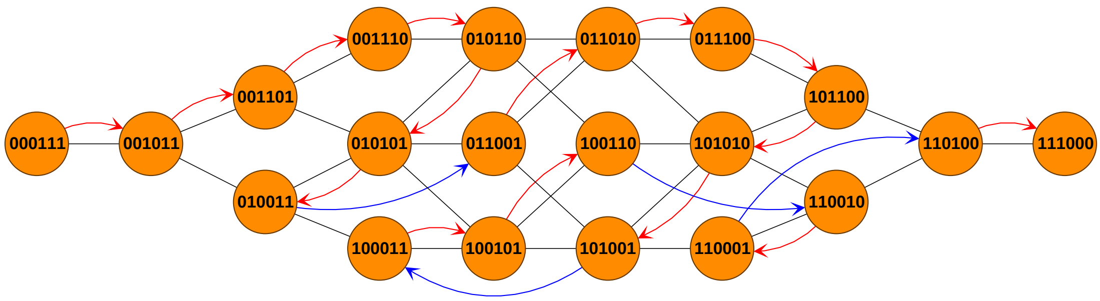
```
Input  : Neighbor-swap graph G
Output: Hamiltonian path, potentially with spurs
nTally ← 1;  // node tally
sTally ← 0;  // spur tally
path ← empty array;
B ← node with the least serial number;
while B has neighbors do
    N ← node connected to B of least multiplicity with the least serial number;
    if multiplicity(N) = 1 then
        path ← path + N + B;
        sTally ← sTally + 1;
        Disconnect B and N;
    else
        path ← path + N;
        Disconnect B from all nodes;  // reducing the multiplicity by 1 of
          each such node
        B ← N;
    end
    nTally ← nTally + 1;
end
Output sTally, nTally, and path;
```

A.2 Rivertz's Result Comparison

A solution of Rivertz's algorithm [15] to generate a Hamiltonian path in neighbor-swaps compared to Verhoeff's Theorem 1. With Fig. A.1 we show that Rivertz's algorithm does not provide optimal solutions.



(a) An optimal solution using Verhoeff's algorithm with a total motion of 19.



(b) The solution given by Rivertz's algorithm with a total motion of 23.

Figure A.1: Comparison of permutation generation in the binary signature $(3, 3)$

Appendix B

Special Paths in Neighbor-Swap Graphs

Fig. B.1 shows a Hamiltonian path in a neighbor-swap graph for the signature $(1, 2, 1)$. This neighbor-swap graph often requires special attention throughout the Master Thesis because it only admits a Hamiltonian path. However, we sometimes might end up with it when referring to an $(odd, 2, 1)$ subsignature.

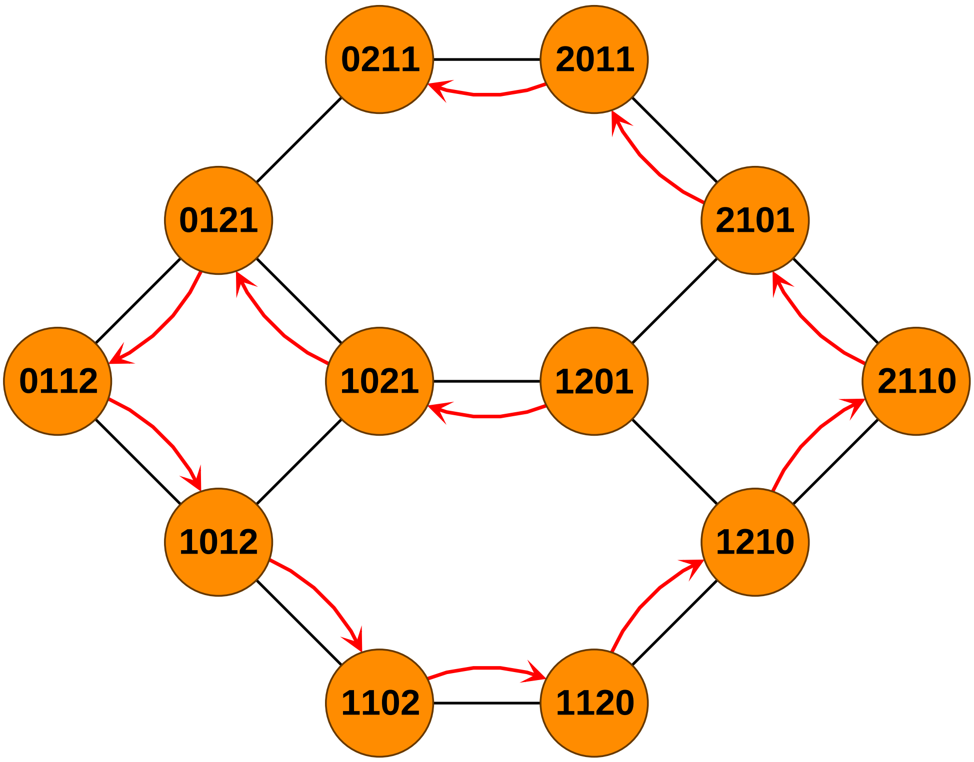
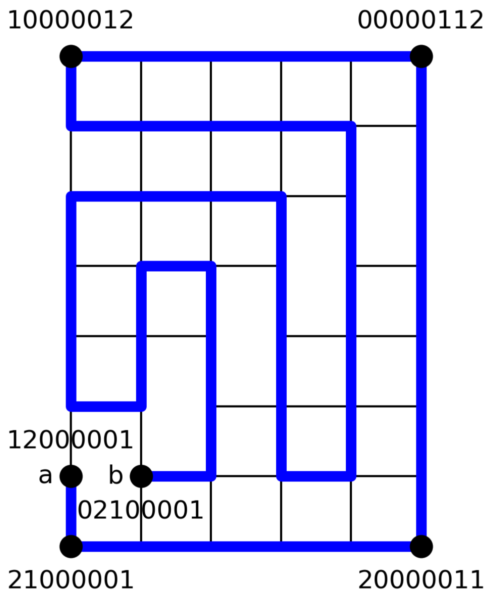
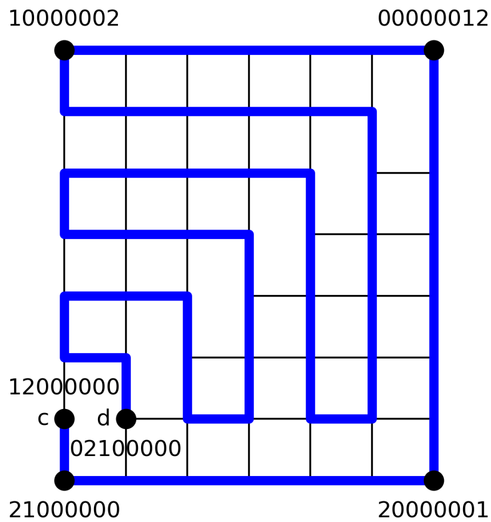


Figure B.1: A Hamiltonian path in the $(1, 2, 1)$ signature neighbor-swap graph.

Fig. B.2 shows how a “180° rotated L” or “⅂” can be attached to the Hamiltonian path or cycle in signature $(\mathit{odd}, 2, 1)$ of §3.3. This ⅂ also extends the Hamiltonian path in signature $(\mathit{even}, 1, 1)$ of §3.4.



(a) A subgraph of the neighbor-swap graph with signature $(5, 1, 1)$ with fixed trailing 1 and $(5, 1)$ with trailing 12 with special vertices marked.



(b) A subgraph of the neighbor-swap graph with signature $(6, 1, 1)$ with special vertices marked.

Figure B.2: The neighbor-swap graphs that generate a path between $a0 = 120^{k_0-1}\underline{10} \sim 120^{k_0}1 = c1$ and $b0 = 0210^{k_0-2}\underline{10} \sim 0210^{k_0-1}1 = d1$ for $k_0 = 6$.

Appendix C

Meta-graph of Cycles

C.1 Overview of signature dependencies

This section shows how the Hamiltonian paths and cycles depend on each other. Fig. C.1 shows all signatures between which we distinguish our cases in Chapter 3. The graphs below only filter out one of the general signatures and do not show the other cases are not related. Fig. C.2 shows the dependencies of all-even signatures. Fig. C.3 shows the dependencies of one odd - rest even signatures. Fig. C.4 shows the dependencies of two odd - rest even signatures. Fig. C.5 shows the dependencies of three or more odd - rest even signatures. From Fig. C.1 it is clearer why the special cases for some signatures appear. This is because:

1. Signature $(odd, 2, 1)$ has subsignature:
    - $(even, 1, 1)$
2. Signature $(even, 2, 1)$ has subsignature:
    - $(even, 1, 1)$
3. Signature $(odd, odd, 1)$ has subsignature:
    - (odd, odd)
4. Signature $(even, odd, 1)$ has subsignatures:
    - (odd, odd)
    - $(even, 1, 1)$
5. Signature $(even, 1, 1, 1)$ has subsignature:
    - $(even, 1, 1)$
6. Signature $(even, 2, 1, 1)$ has subsignature:
    - $(even, 1, 1)$.

The neighbor-swap graphs of these subsignatures only admit Hamiltonian paths and not Hamiltonian cycles. Therefore we require the case distinction from Fig. C.1. Note that the all-even signature also uses subsignature $(even, 1, 1)$ but we only require a path there as explained in §3.9.

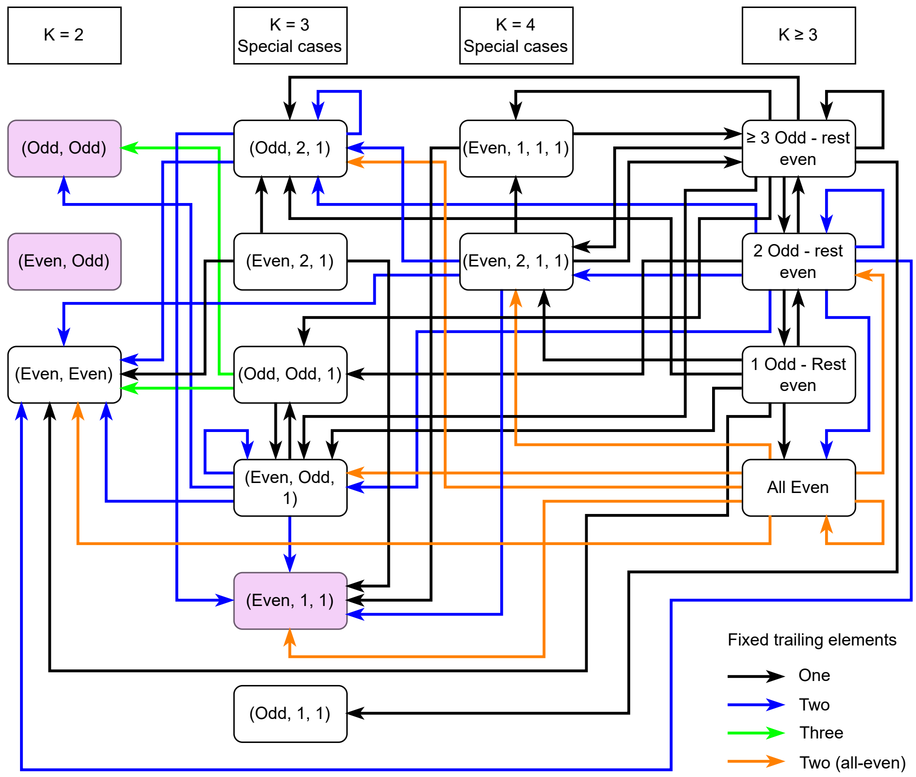


Figure C.1: Subgraphs of signatures. An arrow from a to b implies that b is a subgraph of a. Signatures that only admit a Hamiltonian path are indicated in purple.

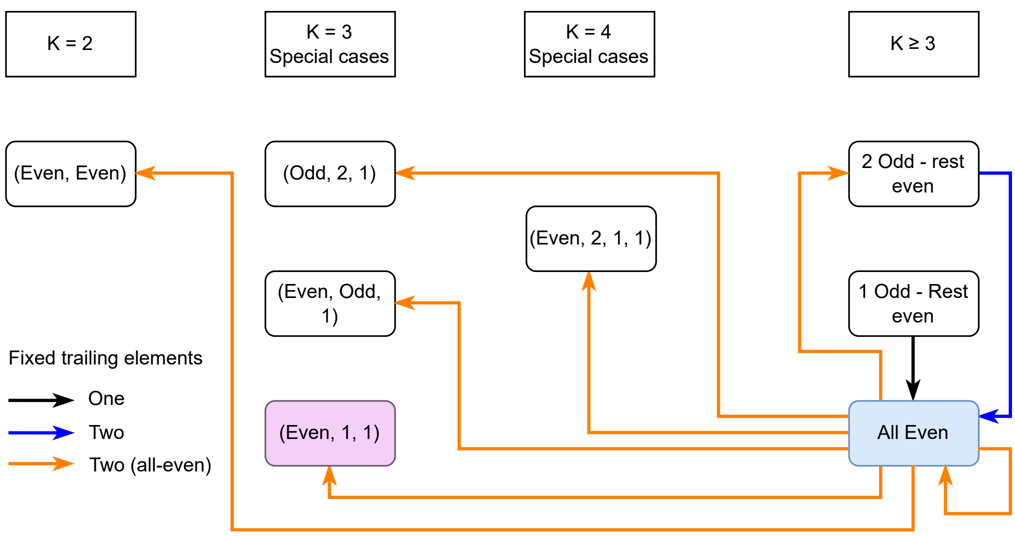


Figure C.2: Signatures that are subgraphs of all-even signatures. The all-even signature is indicated in blue.

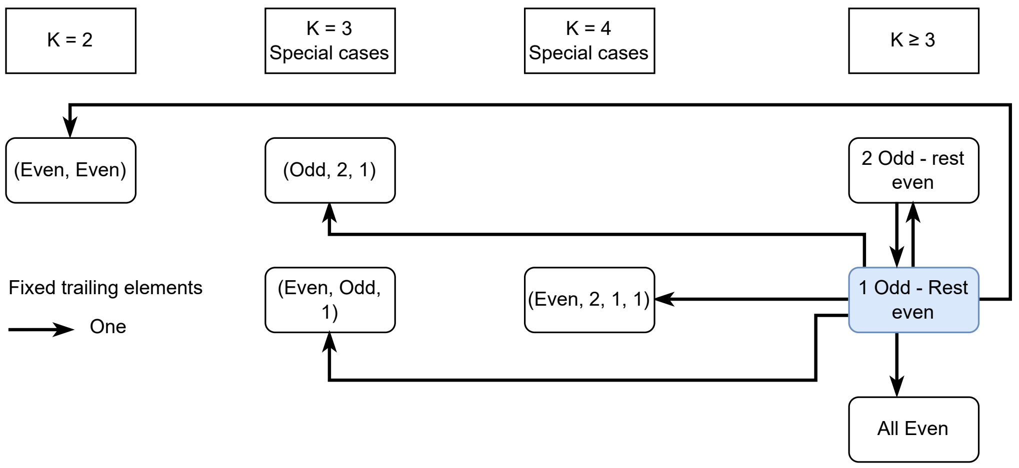


Figure C.3: Signatures that are subgraphs of one odd - rest even signatures. The one odd - rest even signature is indicated in blue.

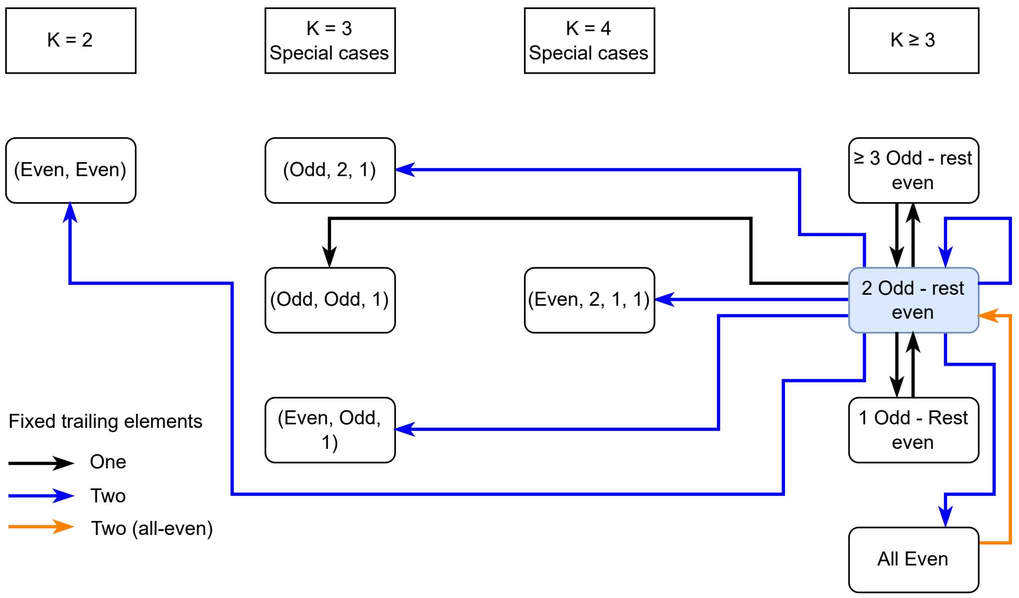


Figure C.4: Signatures that generate the subgraphs of two odd - rest even signatures. The two odd - rest even signature is indicated in blue.

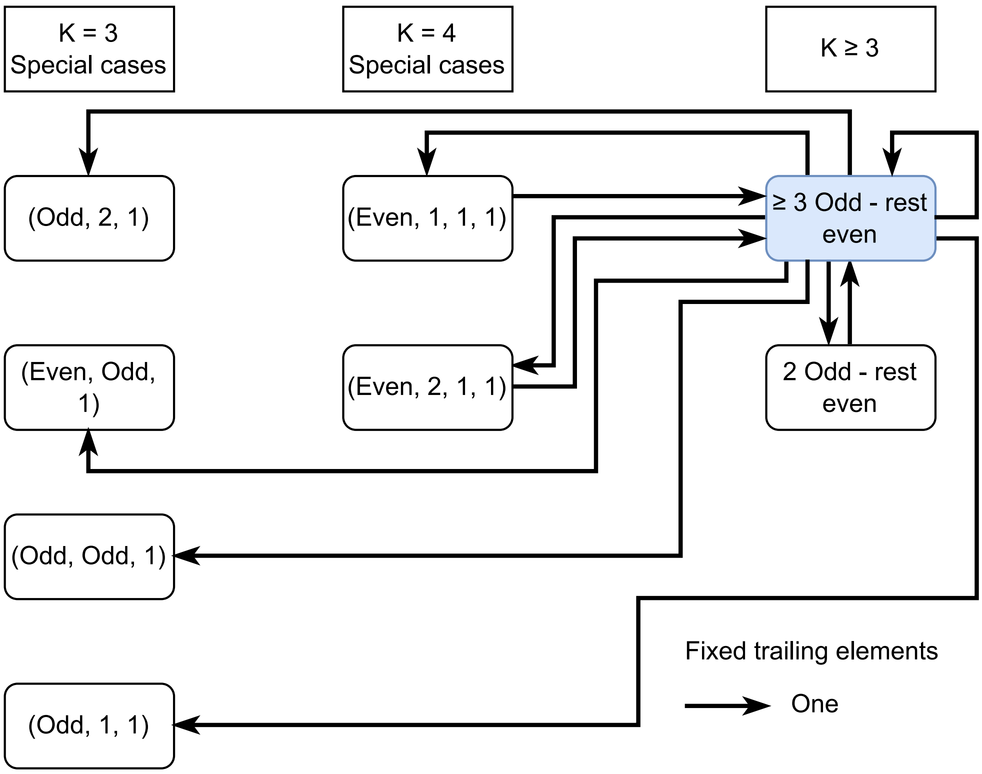


Figure C.5: Signatures that generate the subgraphs of three or more odd - rest even signatures. The three or more odd - rest even signature is indicated in blue.

C.2 Algorithms to connect the meta-graph of cycles

Here, we provide two algorithms that show that the meta-graphs of cycles admit a Hamiltonian cycle. These algorithms are used in the implementation of the author.

C.2.1 All-even

Algorithm 2: Hamiltonian cycle on the meta-graph of cycles for all-even signatures

Input : Signature (a signature with only even-occurring colors with length > 2)
Output: Tails (trailing elements that connect the cycles in the disjoint cycle cover)

1 `// The first three connections are always _100, _101, and _211`
2 Tails ← $[100, 101, 211]$;
3 **for** *i = 2; i < K - 1; i++* **do**
4   `// The two nodes that are at the start are always the same`
5   Tails ← Tails + $[i(i-1)i, 0ii]$;
6   **for** *j = 0; j < i-2; j++* **do**
7   `// The normal loop, starting at` $10i$ `and ending at` $(i-2)(i-3)i$
8   Tails ← Tails + $[(j+1)ji]$;
9   **end**
10   `// The last node of a sequence connects the one to the next`
11   Tails ← Tails + $[(i+1)(i-2)i]$;
12 **end**
13 `// The last row is going from K - 2 to 0`
14 Tails ← Tails + $[(K-1)(K-2)(K-1), (K-3)(K-1)(K-1)]$;
15 **for** *i = K - 3; i > 0 ; i–* **do**
16   Tails ← Tails + $[j(j+1)(K-1)]$;
17 **end**
18 `// Note that this tail is not used in case of a Hamiltonian path`
19 Tails ← Tails + $[0(K-1)0]$;
20 Return Tails;

Algorithm 2 has a running time of $\mathcal{O}(K^2)$ where K is the number of distinct colors in the signature. We have a loop that runs from 2 to $(K-1)$ and 0 to $i-2$ (depending on the outer loop). Asymptotically this reduces to $\mathcal{O}(K^2)$ because all operations inside the double loop take constant time. The second loop (on line 15) runs in $\mathcal{O}(K)$ time. Therefore the running time of the whole algorithm is $\mathcal{O}(K^2)$. This is trivial because we have also obtained $\binom{K+1}{2} = \frac{(K+1)K}{2}$ subcycles.

The smallest all-even signature is $(2, 2, 2)$ and its meta-graph is shown in Fig. 3.8a. The algorithm produces the same output if the signature keeps the same length (i.e., three colors). The meta-graph of cycles of these signatures also reduces to six subcycles with the same trailing elements. The output will be $[100, 101, 211, 212, 022, 020]$. The first three suffices are set at the initialization of the output array: *Tails*. The loop at line 3 is not executed since the $2 \not< 2 = K-1$. Line 14 adds 212 and 022, and line 19 adds 020.

For the proof of the algorithm, we will assume the signature is sorted corresponding to the order in which we generate the disjoint cycle cover. We will show that the algorithm is correct by induction on K, the number of colors. To ease the explanation; we do not use the all-even signature with $K = 3$ (which is shown to be correct above) since the loop on line 16 of Algorithm 2 is not executed. So with the base case $K = 4$ we will now prove that the output remains a Hamiltonian cycle on the meta-graph of cycles. The output for signature $K = 4$ is shown in Fig. 3.8b. The start remains identical; from 100 to 211. Then line 5 adds 212 and 022; line 8 adds 302. Then line 14 adds 323 and 133; the loop of line 16 adds 103; and line 19 adds

030. This connects the meta-graph of cycles with the black edges in Fig. 3.8b. An example can be found in Table C.1 for a signature of length six.

100					
101	211				
212	022	302			
323	033	103	413		
434	044	104	214	524	
545	355	235	125	015	050

Table C.1: Tails generated by Algorithm 2 for an all-even signature of length six. The first three tails $[100, 101, 211]$ are fixed and generated by line 2. The double vertical line indicates that the loop at line 6 starts. The double horizontal line indicates that the loop at line 3 ends. Below that, the first two tails are generated by line 14; after the horizontal double line by line 16; and after the last double line by line 19.

So we assume we have a Hamiltonian cycle on all an all-even signature of length $K-1$. This is a Hamiltonian cycle on all the subcycles with the trailing elements from 00 to $0(K-1)/(K-1)0$. Where $0(K-1)/(K-1)0$ denotes the combined meta-node with trailing $0(K-1)$ and $(K-1)0$. In between these subcycles, we do not have subcycles where K is in the two trailing elements. Thus we only add those when creating the output for length K. To add those cycles we will first alter the order of the cycles with suffixes containing $K-1$. The suffixes with $K-1$ were connected by the lines 14 to 19. Now they will be connected by lines 3 to 11. This practically flips the ordering for the tails containing $K-1$. The new order will be $(K-1)(K-2)(K-1)$ and $0(K-1)(K-1)$ (line 5). Then line 8 adds the meta-nodes with suffixes $10(K-1)$, $21(K-1)$, $\ldots$, $(K-3)(K-4)(K-1)$, and line 11 adds $K(K-3)(K-1)$. This is a Hamiltonian path on the meta-graph of cycles with all tails that contain trailing elements up to $K-1$.

This path is extended with the cycles containing trailing elements K using lines 14 to 19, similar to $(K-1)$ before. First line 14 adds $K(K-1)K$ and $(K-2)KK$. Then the loop on line 15 adds the cycles with tails $(K-3)(K-2)K$, $(K-4)(K-3)K$, $\ldots$, $01K$ and line 19 adds $0K0$; which closes the Hamiltonian cycle on the meta-graph of cycles for the all-even case.

C.2.2 One odd - rest even or three or more odd - rest even

Algorithm 3 is used to generate a Hamiltonian cycle in the meta-graph of cycles of one odd - rest even signatures and three or more odd - rest even signatures.

Algorithm 3: Hamiltonian cycle on the meta-graph of cycles for one odd - rest even and three or more odd - rest even signatures

Input : Signature (signature of the form one odd - rest even or three or more odd - rest even with $K \geq 3$)

Output: Tails (trailing elements that connect the cycles in the disjoint cycle cover)

1 Tails ← empty array;
2 **for** *i = 0; i < K - 1; i++* **do**
3   Tails ← Tails + $[(i+1)i]$
4 **end**
5 `// Note that this tail is not used in case of a Hamiltonian path`
6 Tails ← Tails + $[0(K))]$
7 Return Tails

So we start with a signature with three colors as our base case, e.g. $(3, 2, 2)$ or $(4, 4, 3)$. The algorithm will add suffixes 10 and 21 on line 3. Then on line 5, after the loop, the tail 02

is added. This is a Hamiltonian cycle on the meta-graph of cycles for a signature of length 3. By induction, we will show that we can generate a Hamiltonian cycle on the meta-graph of cycles for K, assuming we have a Hamiltonian cycle for $K-1$.

So the Hamiltonian cycle on the meta-graph of a one odd - rest even signature of length $K-1$ follows the path $10,\ 21,\ \ldots,\ 0(K-1)$. This can be extended to form a Hamiltonian cycle for K by replacing this last tail with $K(K-1)$ and adding $0K$. The inductive step in Algorithm 3 is applied by executing line 3 once more instead of line 5 for $K-1$ and then executing line 5 for $0K$ instead of $0(K-1)$. Therefore it correctly generates a Hamiltonian cycle on the meta-graph of cycles for one odd - rest even signatures and three or more odd - rest even signatures.

An example of a Hamiltonian path generated by Algorithm 3 can be seen in Fig. 3.9. This is for a signature of length 6. The black path is the generated ordering and combined with the last blue edge this is a Hamiltonian cycle on the meta-graph. The red edges indicate that more paths can be taken to achieve this.

The running time of the algorithm is $\mathcal{O}(K)$ since the loop on line 2 takes $\mathcal{O}(K)$ time. The operation inside the loop takes contact time. The line 6 also takes $\mathcal{O}(1)$. Therefore the total asymptotic running time of Algorithm 3 is $\mathcal{O}(K)$.

Appendix D

Alternative Hamiltonian cycles in two odd - rest even signatures

This chapter discusses a special case of two odd - rest even signatures. We discuss the $(odd, even, 1)$ signatures in Appendix D.1. This graph requires one distinct cross edge because of the guaranteed edges in two odd - rest even signatures.

D.1 Alternative $(odd, even, 1)$ Hamiltonian cycle

We list some parallel edges that are glued in the $(odd, even, 1)$ signature here. These are in practice the same as the parallel edges (3.58) and (3.59) or (3.61) and (3.60). However, we have k_0 odd and k_1 even. Thus the signature is $(odd, even, 1)$ instead of $(even, odd, 1)$ such as in §3.6.
If $k_0 - 1 < k_1$, we substitute the parallel edges between (3.58) and (3.59) for (D.1) and (D.2):

$$1^{k_1}0^{k_0-1}20 \sim 1^{k_1-1}\underline{010}^{k_0-2}20 \tag{D.1}$$

$$1^{k_1}0^{k_0-2}200 \sim 1^{k_1-1}\underline{010}^{k_0-3}200 \tag{D.2}$$

If $k_0 > k_1$, we substitute the parallel edges between (3.58) and (3.59) for (D.1) and (D.2):

$$0^{k_0-1}1^{k_1}20 \sim 0^{k_0-2}\underline{101}^{k_1-1}20 \tag{D.3}$$

$$0^{k_0-1}1^{k_1-1}210 \sim 0^{k_0-2}\underline{101}^{k_1-2}210 \tag{D.4}$$

The argument for the existence of these edges in the subcycles is similar to those of the original cross edges.

D.1.1 Guaranteed edges

In general, the same edges are guaranteed, except for the conditionally guaranteed edges (3.74), (3.75), (3.76), and (3.77). We guarantee new edges:
If $k_0 > k_1$:

$$0^{k_0-1}1^{k_1}20 \sim 0^{k_0-1}1^{k_1-1}\underline{21}0 \tag{D.5}$$

This edge is used as one of the cross edges between (D.3) and (D.4).

$$1^{k_1}0^{k_0-1}20 \sim 1^{k_1-1}\underline{010}^{k_0-2}20 \tag{D.6}$$

Edge (D.6) is not used as a cross edge in this case and exists for the same reason as (D.1).

If $k_0 < k_1$:

$$1^{k_1}0^{k_0-1}20 \sim 1^{k_1}0^{k_0-2}\underline{20}0 \tag{D.7}$$

This edge is used as one of the cross edges between (D.1) and (D.2).

$$0^{k_0-1}1^{k_1}20 \sim 0^{k_0-2}\underline{10}1^{k_1-1}20 \tag{D.8}$$

Edge (D.8) is not used as a cross edge in this case and exists for the same reason as (D.3).

Appendix E

Hamiltonian cycles for three or more odd - rest even signatures

We discuss some three or more odd - rest even signatures in more detail here. They have subsignatures that are special signatures in our case distinction, although they follow the same structure as other signatures. This appendix shows that the Hamiltonian cycles in these neighbor-swap graphs cover the guaranteed edges of §3.12.1.4 and §3.12.1.5.

E.1 Signature $(\textit{even}, 1, 1, 1, 1)$

This is a special case because four out of five subgraphs reduce to signature $(\textit{even}, 1, 1, 1)$. The other subsignature $(\textit{odd}, 1, 1, 1, 1)$ with the base case $(1, 1, 1, 1, 1)$. We connect the cycles in the order of Algorithm 3.

$$0^{k_0-1}23410 \sim 0^{k_0-1}2\underline{43}10 \tag{E.1}$$
$$0^{k_0-1}23401 \sim 0^{k_0-1}2\underline{43}01 \tag{E.2}$$

Edge (E.1) is generated in the $(\textit{odd}, 1, 1, 1, 1)$ subsignature. In subsignature $(1, 1, 1, 1)$, the edge is guaranteed by (3.193). In other $(\textit{odd}, 1, 1, 1, 1)$ signatures, the trailing $41/14$ nodes are not used as cross edges between the cycles by §3.12.1.4. So we continue to the subsignature $(\textit{odd}, 1, 1, 1)$ with trailing 4. In this signature, the edge is guaranteed by (3.176). Edge (E.2) is generated in the $(\textit{even}, 1, 1, 1)$ subsignature where it is guaranteed by (3.88).

$$30^{k_0}421 \sim \underline{03}0^{k_0-1}421 \tag{E.3}$$
$$30^{k_0}412 \sim \underline{03}0^{k_0-1}412 \tag{E.4}$$

Both edges (E.4) and (E.3) are generated in an $(\textit{even}, 1, 1, 1)$ signature where we edge (3.85) is guaranteed. By the same argument, we have the following edges:

$$10^{k_0}432 \sim \underline{01}0^{k_0-1}432 \tag{E.5}$$
$$10^{k_0}423 \sim \underline{01}0^{k_0-1}423 \tag{E.6}$$
$$10^{k_0}234 \sim \underline{01}0^{k_0-1}234 \tag{E.7}$$
$$10^{k_0}243 \sim \underline{01}0^{k_0-1}243 \tag{E.8}$$

E.1.1 Guaranteed edges

$$0^{k_0}1342 \sim 0^{k_0}1\underline{43}2 \tag{E.9}$$

Edge (E.9) is guaranteed in the $(\textit{even}, 1, 1, 1)$ signature with trailing 2 by (3.89).

$$10^{k_0}234 \sim 10^{k_0}2\underline{43} \tag{E.10}$$

This corresponds to one of the cross edges between (E.7) and (E.8).

$$0^{k_0-1}23410 \sim 0^{k_0-1}234\underline{01} \tag{E.11}$$
$$30^{k_0}421 \sim 30^{k_0}4\underline{12} \tag{E.12}$$
$$10^{k_0}432 \sim 10^{k_0}4\underline{23} \tag{E.13}$$

These edges correspond to cross edges between (E.1) and (E.2), (E.3) and (E.4), and (E.5) and (E.6) respectively.

E.2 Signature $(\textit{even}, 2, 1, 1, 1)$

The neighbor-swap graph of this signature has five subsignatures when we fix one trailing element. We present a way to connect the meta-graph of cycles in the order of Algorithm 3.

$$0^{k_0-1}123410 \sim 0^{k_0-1}12\underline{43}10 \tag{E.14}$$
$$0^{k_0-1}123401 \sim 0^{k_0-1}12\underline{43}01 \tag{E.15}$$

Edge (E.14) reduces to the signature $(\textit{odd}, 2, 1, 1, 1)$ where the edges with trailing $41/14$ are not used as cross edges. So we can reduce the edge to subsignature $(\textit{odd}, 1, 1, 1, 1)$. An all-ones signature guarantees the edge by (3.194). If $k_0 \geq 3$, the edge is guaranteed by (3.176) in subsignature $(\textit{odd}, 1, 1, 1)$.

Edge (E.15) has a trailing 0 and if $k_0 = 2$, this reduces to subsignature $(1, 2, 1, 1, 1)$. This $(\textit{even}, 1, 1, 1, 1)$ subsignature requires more work since we have to swap the elements of k_0 and k_1 after generating the Hamiltonian cycle for subsubsignature $(1, 1, 1, 1, 1)$. This results in one of the cross edges being $10234 \sim 102\underline{43}$ instead of $01234 \sim 012\underline{43}$. We use this subsubcycle when we generate the $(1, 2, 1, 1, 1)$ subcycle. We add a trailing 0 to this subsubgraph and combine the meta-graph of cycles in the order of Algorithm 3. We can use the same guaranteed edges to glue the subcycles by generating the Hamiltonian cycle in this order. The first subcycles are connected by gluing parallel edges:

$$023410 \sim 024310$$
$$023401 \sim 024301$$

The prior one of these edges was not affected by the change above because the subsubgraph guarantees edge (3.193) where both 0 and 1 lexicographically precede 2. The second cycle is not affected by the change explained above because it is generated in a different subsubcycle, similar to the other subsubcycles. Therefore we have created a subcycle in the $(2, 1, 1, 1, 1)$ neighbor-swap graph where edge $102340 \sim 102\underline{43}0$ is guaranteed. When we change the signature to $(1, 2, 1, 1, 1)$, this edge is the required (E.14). Other $(\textit{odd}, 2, 1, 1, 1)$ guarantee the parallel edge by (3.218).

$$0^{k_0}13421 \sim 0^{k_0}1\underline{43}21 \tag{E.16}$$
$$0^{k_0}13412 \sim 0^{k_0}1\underline{43}12 \tag{E.17}$$

Edge (E.16) is generated in subsignature $(\textit{even}, 1, 1, 1, 1)$ where it is guaranteed by (E.9). Edge (E.17) is generated in subsignature $(\textit{even}, 2, 1, 1)$ where it is guaranteed by (3.118).

$$110^{k_0}432 \sim 1\underline{01}0^{k_0-1}432 \tag{E.18}$$
$$110^{k_0}423 \sim 1\underline{01}0^{k_0-1}423 \tag{E.19}$$

Both edges (E.18) and (E.19) are guaranteed in subsignature $(even, 2, 1, 1)$ and guaranteed by (3.119).

$$110^{k_0}234 \sim \underline{10}10^{k_0-1}234 \tag{E.20}$$
$$110^{k_0}243 \sim \underline{10}10^{k_0-1}243 \tag{E.21}$$

Edges (E.20) and (E.21) also follow from the signature $(even, 2, 1, 1)$ by (3.120).

E.2.1 Guaranteed edges

$$10^{k_0}2341 \sim 10^{k_0}2\underline{43}1 \tag{E.22}$$

This edge is guaranteed by (E.10) in the subcycle with trailing 1.

$$30^{k_0}1142 \sim \underline{03}0^{k_0-1}1142 \tag{E.23}$$
$$40^{k_0}1132 \sim \underline{04}0^{k_0-1}1132 \tag{E.24}$$
$$20^{k_0}1143 \sim \underline{02}0^{k_0-1}1143 \tag{E.25}$$
$$20^{k_0}1134 \sim \underline{02}0^{k_0-1}1134 \tag{E.26}$$

All these edges are generated in $(even, 2, 1, 1)$ subsignature cycles. In that signature, edge (E.23), (E.25), and (E.26) are guaranteed by (3.106). Edge (E.24) is guaranteed by (3.107).

$$110^{k_0}234 \sim 110^{k_0}2\underline{43} \tag{E.27}$$

Edge (E.27) is guaranteed as the cross edge between (E.20) and (E.21).

E.3 Signature $(\boldsymbol{even}, \boldsymbol{odd}, 1, 1)$

This neighbor-swap graph has four subcycles where two cycles are of signature $(even, odd, 1)$ which is handled in §3.6. Assume $k_0 \geq 4$ and $k_1 \geq 3$. We connect the meta-graph of cycles by gluing parallel edges:

$$1^{k_1-1}0^{k_0-1}2310 \sim 1^{k_1-1}0^{k_0-1}\underline{32}10 \tag{E.28}$$
$$1^{k_1-1}0^{k_0-1}2301 \sim 1^{k_1-1}0^{k_0-1}\underline{32}01 \tag{E.29}$$

Edge (E.28) is generated in subsignature $(odd, odd, 1, 1)$ where the edge is guaranteed by (E.50). Edge (E.29) is generated in a two odd - rest even signature cycle. Both nodes in the edge have an even-occurring trailing element, so they reduce to an $(odd, even, 1, 1)$ signature where the edge is guaranteed by (E.35). However, the edge (E.29) can also reduce to subsignature $(even, 2, 1, 1)$ which guarantees (3.117).

$$1^{k_1-1}0^{k_0}321 \sim 1^{k_1-2}\underline{01}0^{k_0-1}321 \tag{E.30}$$
$$1^{k_1-1}0^{k_0}312 \sim 1^{k_1-2}\underline{01}0^{k_0-1}312 \tag{E.31}$$

Edge (E.30) is generated in an $(even, even, 1, 1)$ signature cycle. By edge (3.164) is guaranteed in this two odd - rest even signature. If the subsignature is $(even, 2, 1, 1)$, the edge is guaranteed by (3.119). Edge (E.31) is generated in an $(even, odd, 1)$ signature. This signature guarantees the edge by (3.75).

$$0^{k_0}1^{k_1}32 \sim 0^{k_0-1}\underline{10}1^{k_1-1}32 \tag{E.32}$$
$$0^{k_0}1^{k_1}23 \sim 0^{k_0-1}\underline{10}1^{k_1-1}23 \tag{E.33}$$

Both edges (E.32) and (E.33) have the signature $(even, odd, 1)$. This signature guarantees these edges by (3.73).

E.3.1 Guaranteed edges

$$20^{k_0}1^{k_1-1}31 \sim \underline{02}0^{k_0-1}1^{k_1-1}31 \tag{E.34}$$

This edge is generated in the $(\textit{even}, \textit{even}, 1, 1)$ signature with trailing 1. In the two odd - rest even signature it is guaranteed by the edge (3.166).

$$0^{k_0}1^{k_1-1}231 \sim 0^{k_0}1^{k_1-1}\underline{32}1 \tag{E.35}$$

$$1^{k_1-1}0^{k_0}231 \sim 1^{k_1-1}0^{k_0}\underline{32}1 \tag{E.36}$$

This edge is generated in the cycle of signature $(\textit{even}, \textit{even}, 1, 1)$ with trailing 1. By the two odd - rest even signatures, the edge reduces to (3.163). In the $(\textit{even}, 2, 1, 1)$ signature, this edge is guaranteed by (3.116). The same argument holds for (E.36).

$$1^{k_1}0^{k_0}32 \sim 1^{k_1}0^{k_0-1}\underline{30}2 \tag{E.37}$$

This edge is generated in the $(\textit{even}, \textit{odd}, 1)$ signature subcycle with trailing 2. That signature has the guaranteed edge (3.72).

$$1^{k_1}0^{k_0}23 \sim 1^{k_1-1}\underline{01}0^{k_0-1}23 \tag{E.38}$$

$$1^{k_1}0^{k_0}32 \sim 1^{k_1-1}\underline{01}0^{k_0-1}32 \tag{E.39}$$

Both edges (E.38) and (E.39) are generated in $(\textit{even}, \textit{odd}, 1)$ signatures because they are not used as crosse edges. The subgraph guarantees the edge by (3.68).

$$1^{k_1-1}0^{k_0}321 \sim 1^{k_1-1}0^{k_0}3\underline{12} \tag{E.40}$$

This edge is guaranteed because it is the cross edge between (E.30) and (E.31).

$$0^{k_0}1^{k_1}32 \sim 0^{k_0}1^{k_1}\underline{23} \tag{E.41}$$

This edge is guaranteed because it is the cross edge between (E.32) and (E.33).

E.4 Signature $(\textit{odd}, \textit{odd}, 1, 1)$

The cycle of this signature is generated from four subcycles. Two of these cycles have sub-signature $(\textit{odd}, \textit{odd}, 1)$ which is a special case, addressed in § 3.5. Both $k_0 \geq 3$ and $k_1 \geq 3$ because smaller frequencies result in the $(\textit{odd}, 1, 1, 1)$ or all-ones signatures. We connect the meta-graph of cycles according to the order of Algorithm 3 again:

$$20^{k_0-1}1^{k_1-1}310 \sim \underline{02}0^{k_0-2}1^{k_1-1}310 \tag{E.42}$$

$$20^{k_0-1}1^{k_1-1}301 \sim \underline{02}0^{k_0-2}1^{k_1-1}301 \tag{E.43}$$

Edge (E.42) is generated by a subgraph of signature $(\textit{even}, \textit{odd}, 1, 1)$. By (E.34), we must have that the edge exists. Similarly, edge (E.43) in the $(\textit{odd}, \textit{even}, 1, 1)$ signature subcycle exists by (E.34).

$$0^{k_0}1^{k_1-1}321 \sim 0^{k_0}1^{k_1-2}\underline{31}21 \tag{E.44}$$

$$0^{k_0}1^{k_1-1}312 \sim 0^{k_0}1^{k_1-2}\underline{31}12 \tag{E.45}$$

Edge (E.44) is generated in the $(\textit{odd}, \textit{even}, 1, 1)$ signature with the edge guaranteed as (E.37). Edge (E.45) is guaranteed in the $(\textit{odd}, \textit{odd}, 1)$ signature by (3.47).

$$0^{k_0}1^{k_1}23 \sim 0^{k_0-1}\underline{10}1^{k_1-1}23 \tag{E.46}$$

$$0^{k_0}1^{k_1}32 \sim 0^{k_0-1}\underline{10}1^{k_1-1}32 \tag{E.47}$$

Both edges (E.46) and (E.47) originate in subsignature $(\textit{odd}, \textit{odd}, 1)$. The edge is guaranteed by (3.48).

E.4.1 Guaranteed edges

$$0^{k_0-1}1^{k_1-1}2301 \sim 0^{k_0-1}1^{k_1-1}\underline{32}01 \tag{E.48}$$
$$1^{k_1-1}0^{k_0-1}2310 \sim 1^{k_1-1}0^{k_0-1}\underline{32}10 \tag{E.49}$$

This edge is guaranteed in the $(odd, even, 1, 1)$ signature cycle with trailing 1 by (E.35). Similarly, (E.49) is guaranteed by (E.36).

$$1^{k_1-1}0^{k_0}231 \sim 1^{k_1-1}0^{k_0}\underline{32}1 \tag{E.50}$$

Edge (E.50) is guaranteed in the $(odd, even, 1, 1)$ subsignature by (E.41).

$$1^{k_1-1}0^{k_0}213 \sim 1^{k_1-1}0^{k_0}\underline{12}3 \tag{E.51}$$
$$1^{k_1-1}0^{k_0}312 \sim 1^{k_1-1}0^{k_0}\underline{13}2 \tag{E.52}$$

Edge (E.51) is guaranteed by (3.50) in the $(odd, odd, 1)$ subsignature with trailing 3. Edge (E.52) also exists in the subcycle with trailing 2 by (3.50).

$$20^{k_0-1}1^{k_1-1}310 \sim 20^{k_0-1}1^{k_1-1}3\underline{01} \tag{E.53}$$
$$0^{k_0}1^{k_1-1}321 \sim 0^{k_0}1^{k_1-1}3\underline{12} \tag{E.54}$$
$$0^{k_0}1^{k_1}23 \sim 0^{k_0}1^{k_1}\underline{32} \tag{E.55}$$

These three edges are respectively the cross edges between (E.42) & (E.42), (E.44) & (E.45), and (3.171) & (3.170).

E.5 Signature $(\boldsymbol{even}, 1, 1, 1, 1, 1)$

We address this case because it is required for the $(even, 2, 1, 1, 1, 1)$ signature in Appendix E.6.

$$0^{k_0-1}234510 \sim 0^{k_0-1}23\underline{54}10 \tag{E.56}$$
$$0^{k_0-1}234501 \sim 0^{k_0-1}23\underline{54}01 \tag{E.57}$$

Edge (E.56) is either generated in an all-ones subsignature where it is guaranteed by (3.193). Or it is generated by an "even number of odd-occurring colors" signature where the edge is guaranteed by (3.218). Edge (E.57) is guaranteed by (3.219).

$$0^{k_0}34521 \sim 0^{k_0}3\underline{54}21 \tag{E.58}$$
$$0^{k_0}34512 \sim 0^{k_0}3\underline{54}12 \tag{E.59}$$
$$0^{k_0}14532 \sim 0^{k_0}1\underline{54}23 \tag{E.60}$$
$$0^{k_0}14523 \sim 0^{k_0}1\underline{54}23 \tag{E.61}$$
$$0^{k_0}12534 \sim 0^{k_0}1\underline{52}43 \tag{E.62}$$
$$0^{k_0}12543 \sim 0^{k_0}1\underline{52}34 \tag{E.63}$$
$$0^{k_0}12345 \sim 0^{k_0}1\underline{32}45 \tag{E.64}$$
$$0^{k_0}12345 \sim 0^{k_0}1\underline{32}45 \tag{E.65}$$

All edges above are guaranteed by (3.218).

E.6 Signature $(\boldsymbol{even}, 2, 1, 1, 1, 1)$

We will show that the generation of the Hamiltonian cycle in the neighbor-swap of this signature follows the same structure as other signatures with an "even number of odd-occurring colors". We rely on guaranteed edges of a special subsignature $(even, 2, 1, 1)$.

$$0^{k_0-1}1234510 \sim 0^{k_0-1}123\underline{54}10 \tag{E.66}$$
$$0^{k_0-1}1234501 \sim 0^{k_0-1}123\underline{54}01 \tag{E.67}$$

The edge (E.67) is formed in the $(odd, 2, 1, 1, 1, 1)$ cycle. If the original signature was $(2, 2, 1, 1, 1, 1)$, we require a technique similar to that explained in Appendix E.2. So we have to alter one of the subsubgraphs of subgraph $(1, 2, 1, 1, 1, 1)$. This subgraph is generated as an $(2, 1, 1, 1, 1, 1)$ signature. The subsubgraph that we have to alter is the one with trailing 0, where we have to swap all elements of k_0 and k_1. This results in edge $102345 \sim 1023\underline{54}$. Then we add the trailing 0 and generate the subgraph of signature $(2, 1, 1, 1, 1)$. For this we use the parallel edge $0234510 \sim 023\underline{54}10$ which remains guaranteed by (3.193). The edge is different than the required edge $1023450 \sim 1023\underline{54}0$. The other subsubcycles are not affected and the validity of their parallel edges remains. Ultimately, the subsignature is changed to $(1, 2, 1, 1, 1, 1)$ providing the required edge $0123451 \sim 0123\underline{54}1$ for (E.66). If $k_0 \geq 4$, the edge is guaranteed by (3.210). Edge (E.67) is guaranteed by (3.209).

$$10^{k_0}34521 \sim 10^{k_0}3\underline{54}21 \tag{E.68}$$
$$10^{k_0}34512 \sim 10^{k_0}3\underline{54}12 \tag{E.69}$$

Edge (E.68) is generated in the $(even, 1, 1, 1, 1, 1)$ signature cycle where the edge has a trailing 4 or 5. There are never cross edges between the cycles with trailing 4 and 2 nor between the cycles with trailing 5 and 2. Therefore, we can fix the trailing 2. This results in the subgraph with signature $(even, 1, 1, 1, 1)$ where the edge is guaranteed by (E.10). The edge (E.17) is generated in the $(even, 2, 1, 1, 1)$ signature where the edge is guaranteed by (E.22).

$$40^{k_0}11523 \sim \underline{04}0^{k_0-1}11523 \tag{E.70}$$
$$40^{k_0}11532 \sim \underline{04}0^{k_0-1}11532 \tag{E.71}$$

Both permutations are generated in the $(even, 2, 1, 1, 1)$ signature. This signature guarantees the edge with (E.23).

The edges with trailing $34/43$ and $45/54$ follow the same structure as (E.70) and (E.71). Therefore we say that there must be cross edges in the $(even, 2, 1, 1, 1)$ signature cycle by (E.24), (E.25), and (E.26). So we conclude that the $(even, 2, 1, 1, 1, 1)$ signature admits a Hamiltonian cycle.

E.7 Signature $(\boldsymbol{even}, \boldsymbol{odd}, 1, 1, 1)$

We now show that the $(even, odd, 1, 1, 1)$ signature also follows the same structure as other an "even number of odd-occurring colors" signatures.

$$0^{k_0-1}1^{k_1-1}23410 \sim 0^{k_0-1}1^{k_1-1}2\underline{43}10 \tag{E.72}$$
$$0^{k_0-1}1^{k_1-1}23401 \sim 0^{k_0-1}1^{k_1-1}2\underline{43}01 \tag{E.73}$$

Edge (E.72) and (E.73) originate in $(odd, odd, 1, 1, 1)$ and $(even, even, 1, 1, 1)$ signatures respectively where there are no cross edges between cycles with trailing $41/14$ and $04/40$. Thus we remove the second trailing element as well and obtain two $(odd, even, 1, 1, 1)$ signatures. This signature has the guaranteed edge by (E.80). If $k_1 = 3$, this subsignature will be $(odd, 2, 1, 1, 1)$, where the edge is guaranteed by (3.208)

$$30^{k_0}1^{k_1-1}421 \sim \underline{030}^{k_0-1}1^{k_1-1}421 \tag{E.74}$$

$$30^{k_0}1^{k_1-1}412 \sim \underline{030}^{k_0-1}1^{k_1-1}412 \tag{E.75}$$

Edge (E.74) originates in the signature $(even, even, 1, 1, 1)$. This three odd - rest even signature guarantees the edge by (3.206). Edge (E.75) is generated by the subgraph of signature $(even, odd, 1, 1)$. This signature guaranteed the edge with (E.34).

$$1^{k_1}0^{k_0}432 \sim 1^{k_1-1}\underline{010}^{k_0-1}432 \tag{E.76}$$

$$1^{k_1}0^{k_0}423 \sim 1^{k_1-1}\underline{010}^{k_0-1}423 \tag{E.77}$$

The edges (E.76) and (E.77) have the same origin in the $(even, odd, 1, 1)$ signature. This signature guarantees the edges by (E.39).

$$1^{k_1}0^{k_0}234 \sim 1^{k_1-1}\underline{010}^{k_0-1}234 \tag{E.78}$$

$$1^{k_1}0^{k_0}243 \sim 1^{k_1-1}\underline{010}^{k_0-1}243 \tag{E.79}$$

The edges (E.78) and (E.79) have the same origin in the $(even, odd, 1, 1)$ signature. This signature guarantees the edges by (E.38).

By combining the cycles using the parallel edges above, we have obtained a Hamiltonian cycle in the $(even, odd, 1, 1, 1)$ signature.

E.7.1 Guaranteed edges

$$1^{k_1}0^{k_0}234 \sim 1^{k_1}0^{k_0}2\underline{43} \tag{E.80}$$

The edge (E.80) is guaranteed because it is the cross edge between (E.78) and (E.79).

$$0^{k_0}1^{k_1}342 \sim 0^{k_0}1^{k_1}\underline{43}2 \tag{E.81}$$

This edge is generated as the guaranteed edge in the $(even, odd, 1, 1)$ signature by (E.55).

E.8 Signature $(odd, odd, 1, 1, 1)$

The $(odd, odd, 1, 1, 1)$ signature has five subcycles. Some of these subgraphs lead to special subsignature $(odd, odd, 1)$. We connect the cycles in the order of Algorithm 3.

$$1^{k_1-1}0^{k_0-1}23410 \sim 1^{k_1-1}0^{k_0-1}2\underline{43}10 \tag{E.82}$$

$$1^{k_1-1}0^{k_0-1}23401 \sim 1^{k_1-1}0^{k_0-1}2\underline{43}01 \tag{E.83}$$

Edge (E.82) is generated in subsignature $(even, odd, 1, 1, 1)$. If $k_0 \neq k_1$, the edge is guaranteed in the subgraph by (3.218). If $k_0 = k_1$, we have to translate one of the subsubgraphs to guarantee the edge. This is in the subgraph with trailing 0 which is generated by subsignature $(k_1, k_0 - 1, 1, 1, 1)$. The subsubsignature we change is the one with a trailing element of k_1. In this subsubgraph we swap the elements of k_0 and k_1 which both occur $k_0 - 1$ times. This guarantees edge $0^{k_0-1}1^{k_1-1}234 \sim 0^{k_0-1}1^{k_1-1}2\underline{43}$ to which we add a trailing element of 0. This does not affect the validity of the required parallel edge $0^{k_0-1}1^{k_1-2}23410 \sim 0^{k_0-1}1^{k_1-2}2\underline{34}10$ in that subcycle. The other subsubcycles are not changed; thus, we can form a Hamiltonian cycle in the neighbor-swap graph of this signature. In this neighbor-swap graph the edge (E.82) is guaranteed. Edge (E.83) is guaranteed by (3.218).

$$1^{k_1-1}0^{k_0}3421 \sim 1^{k_1-1}0^{k_0}\underline{43}21 \tag{E.84}$$
$$1^{k_1-1}0^{k_0}3412 \sim 1^{k_1-1}0^{k_0}\underline{43}12 \tag{E.85}$$

Edge (E.84) is generated in the $(odd, even, 1, 1, 1)$ subsignature where it guaranteed by (E.81). Edge (E.85) is guaranteed in the $(odd, odd, 1, 1)$ subsignature by (E.50).

$$1^{k_1-1}0^{k_0}1423 \sim 1^{k_1-1}0^{k_0}\underline{41}23 \tag{E.86}$$
$$1^{k_1-1}0^{k_0}1432 \sim 1^{k_1-1}0^{k_0}\underline{41}32 \tag{E.87}$$

Edge (E.86) has subsignature $(odd, odd, 1, 1)$. That signature guarantees the edge by (E.52). Similarly, (E.87) must exist by (E.52).

$$1^{k_1-1}0^{k_0}1243 \sim 1^{k_1-1}0^{k_0}\underline{21}43 \tag{E.88}$$
$$1^{k_1-1}0^{k_0}1234 \sim 1^{k_1-1}0^{k_0}\underline{21}34 \tag{E.89}$$

Both edges (E.88) and (E.89) exist by (E.51) in the $(odd, odd, 1, 1)$ subsignatures.

Therefore, the $(odd, odd, 1, 1, 1)$ signature neighbor-swap graph contains a Hamiltonian cycle.