Constructing shortest covering walks of neighbour-swap graphs generating linear extensions of posets by adjacent transpositions

Bellaard, Gijs

Award date:
2019

Constructing Shortest Covering Walks of Neighbour-swap Graphs

Generating linear extensions
of posets by adjacent transpositions

Gijs Bellaard (1014090)
Supervisor: Dr. ir. Tom Verhoeff

July 24, 2019

Abstract

This paper considers graphs, where the vertices are linear extensions of a partially ordered set and edges are adjacent transpositions on these extensions, aptly called neighbour-swap graphs. We give a way to partition neighbour-swap graphs into smaller ones, ways of telling when a graph is *not* a neighbour-swap graph, and share a new proof of the number of edges in specific kinds of neighbour-swap graphs. Just like permutations, linear extensions can be split by their parity into two categories: even or odd. The difference between the number of even and odd linear extensions, also called the surplus, turns out to be closely related to the existence of Hamiltonian paths and other shortest covering walks within neighbour-swap graphs. New ways of calculating this surplus are given together with some already known results in doing so. Everything that we could find on constructing shortest covering walks is also summarized and a "master" conjecture is stated which encompasses every result so far. This master conjecture is more general than what has been published so far.

Contents

1 Introduction

Suppose we have the set $\{1, 2, 3, 4\}$. Consider a permutation of this set, like 1234. Swapping two adjacent elements is called a neighbour-swap. Performing a neighbour-swap yields again a permutation. For example, we can swap the 1 and 2 to obtain 2134. An interesting question is if one can perform these neighbour-swaps (not necessarily starting with 1234) in such a way that every permutation is encountered exactly once. In this case the problem is quite simple and is solved using the well-known Steinhaus-Johnson-Trotter algorithm [1, 8, 9](Theorem 5.1). And, in fact, when the set has more than two elements you can start anywhere, because then there exists a cycle.

So to make things more interesting, let's consider a more general setting using multisets, like $\{1, 1, 2, 2\}$, instead of sets. We can still perform neighbour-swaps but some are useless: swapping the two ones in a permutation of the multiset leaves us with the same permutation. Is it now still possible to visit every permutation exactly once? This multiset-version is what Lehmer and Verhoeff write about in their respective articles [3, 10].

A convenient way to visualize the permutations and neighbour-swaps is to present them as a graph. Permutations become vertices and neighbour-swaps become edges. Such graphs are called, unsurprisingly, neighbour-swap graphs. This also allows us to restate our problem as finding a Hamiltonian path in the neighbour-swap graph. The neighbour-swap graph corresponding to $\{1, 1, 2, 2\}$ can be seen below.

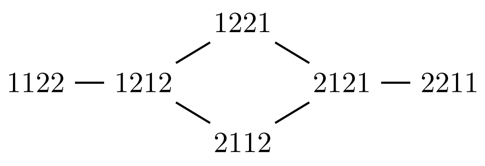


Figure 1: Neighbour-swap graph of the multiset $\{1, 1, 2, 2\}$.

From this graph it can be seen that, sadly, no Hamiltonian path exists. This means the multiset version is more interesting than the set version because it is no longer the case that Hamiltonian paths always exist. We can also loosen our requirement for a Hamiltonian path and just search for a shortest covering walk. Lehmer also suggests this in [3] and writes about "imperfect Hamiltonian paths" which Verhoeff calls Lehmer paths in [10]. What these Lehmer paths exactly are is explained in section 5.

There is an even more general way to create neighbour-swap graphs, viz. by using partially ordered sets (posets). Take for example the poset on $\{1, 2, 3, 4\}$ with the order that 1 precedes 2 and 3 precedes 4. Now the neighbour-swap graph of the linear extensions of this poset acts exactly like $\{1, 1, 2, 2\}$.

$$\begin{tikzcd}
 & & 1342 \arrow[dr, dash] & & \\
1234 \arrow[r, dash] & 1324 \arrow[ur, dash] \arrow[dr, dash] & & 3142 \arrow[r, dash] & 3412 \\
 & & 3124 \arrow[ur, dash] & &
\end{tikzcd}$$

Figure 2: Neighbour-swap graph where 1 precedes 2 and 3 precedes 4.

And for example combining the set $\{1, 2, 3, 4\}$ with the order that 1 and 2 must both precede 3 and 4 gives us the following neighbour-swap graph, which was impossible to acquire previously using just multisets or sets.

$$\begin{tikzcd}
1234 \arrow[r, dash] \arrow[d, dash] & 2134 \arrow[d, dash] \\
1243 \arrow[r, dash] & 2143
\end{tikzcd}$$

Figure 3: A neighbour-swap graph that can not be created using just sets or multisets.

1.1 Overview

Section 2 starts with the preliminaries which consists of concepts, notation, terminology and some visualization. It is suggested everyone should read it no matter how knowledgeable in the subject. Section 3 dives into properties of neighbour-swap graphs. This includes a new formula for the number of edges in a neighbour-swap graph of a multiset. Section 4 is about counting linear extensions and computing the surplus (which will be defined properly in the preliminaries). Most notably two new ways of computing the surplus are shared. Section 5 tackles the main question of finding shortest covering walks of neighbour-swap graphs, which includes Hamiltonian paths. Section 6 concludes the paper.

2 Preliminaries

In the introduction it is discussed that it is a fruitful idea to work with a poset with a relation of the form "this element should precede this element". Obviously the relation is transitive: when element a precedes b, and b precedes c, a should also precede c. The relation is also irreflexive because an element cannot precede itself. To conclude:

Definition 2.1. In our setting a poset P is defined as a *finite* set, also called the ground plane, combined with a relation $\prec$ which is irreflexive and transitive. When $a \prec b$ or $b \prec a$ we call a and b comparable. The cardinality of the ground plane is $|P|$ and when there can be no confusion it is denoted with n.

We can visualize posets as directed graphs in which the vertices represent the elements of the ground plane and the arrows the relation. To reduce clutter we may only show the transitive reduction. These graphs are called Hasse Diagrams. We can be assured a transitive reduction exists because we are working with finite sets. Furthermore the reduction is unique due to there being no cycles because of the irreflexivity.

Example 2.1. In the following figure, two posets with their Hasse diagram can be seen. Conventionally the arrows always point downwards.

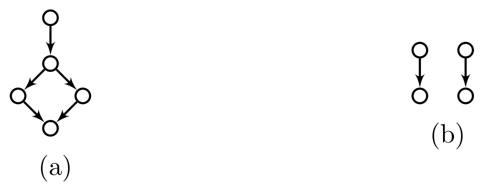


Figure 4: Hasse diagrams of two posets

Definition 2.2. A linear extensions is a total ordering of the poset. Let $\mathscr{L}(P)$ denote the set of all linear extensions of a poset P. We remark that linear extensions can also be considered as posets themselves. Since the linear extensions are a subset of the permutations of the ground plane we may also talk about even and odd linear extensions.

Now we can continue defining the neighbour-swap graphs.

Definition 2.3. A neighbour-swap is defined as a transposition of two adjacent elements within a sequence. When two sequences s_1, s_2 differ by one neighbour-swap we write $s_1 \sim s_2$. A neighbour-swap graph $\mathscr{G}(P)$ of a poset P is defined as the graph on $\mathscr{L}(P)$ with edges corresponding to neighbour-swaps.

Searching for a Hamiltonian path in neighbour-swap graphs would be ridiculous if they were not connected. Luckily they always are.

Definition 2.4. An inversion is two elements who differ in order between two sequences.

Theorem 2.1. *Every neighbour-swap graph is connected.*

Proof. Suppose we have two different linear extensions l_1 and l_2. Between these two there is a set of all inversions of which the cardinality is called the inversion number. There always exists an inversion which involves two adjacent elements in l_1, because if it did not, the two linear extensions would not differ. Neighbour-swapping these two elements in l_1 decreases the inversion number by exactly one. Repeating this process until the inversion number reaches zero means we have reached l_2 from l_1 using only neighbour-swaps. $\square$

Neighbour-swap graphs also have the following property:

Theorem 2.2. *Every neighbour-swap graph is bipartite where the two parts consist of the even and odd linear extensions.*

Proof. As previously stated, the linear extensions of a poset are a subset of the permutations of the ground plane and neighbour-swaps are, after all, nothing more than transpositions. This directly implies neighbour-swap graphs are bipartite because every edge is between an even and an odd permutation. $\square$

Definition 2.5. Define the surplus $\mathscr{D}(\mathscr{G}(P))$, or $\mathscr{D}(P)$, as the absolute difference between the cardinality of the two parts of the bipartition.

You might be wondering why this definition is useful. The following theorem will make clear it is of great interest.

Theorem 2.3.

1. *If $\mathscr{G}(P)$ has a Hamiltonian path then $\mathscr{D}(P) \leq 1$.*
2. *If $\mathscr{G}(P)$ has a Hamiltonian cycle then $\mathscr{D}(P) = 0$.*

Proof. Any path in $\mathscr{G}(P)$ alternates between the two parts of the bipartition. Therefore the difference between the number of vertices visited in one part and the other can not be greater than 1. The proof in case of a cycle goes analogously. $\square$

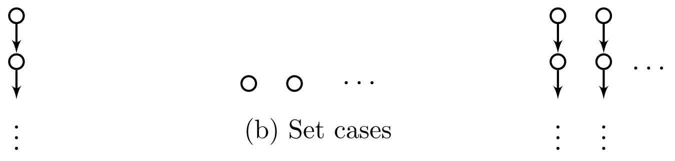
Corollary 2.3.1. *If $\mathscr{D}(P) > 1$ then $\mathscr{G}(P)$ has no Hamiltonian path. If $\mathscr{D}(P) > 0$ then $\mathscr{G}(P)$ has no Hamiltonian cycle.*

Neighbour-swap graphs made from posets generalize the graphs made from multisets and sets. It only seems natural to introduce notation to quickly create posets that reflect these cases.

Definition 2.6. For two disjoint posets P and Q we define $P \mid Q$ to be their parallel composition and PQ to be their series composition. Parallel composition is done by taking the union of the ground planes and relations. Series composition is done by taking the parallel composition together with that everything of P should precede Q. We let i^n denote a chain of n elements labeled with i. In some cases the labeling of the poset is of no importance and in these cases we will write $\circ$. Because we often need to construct chains with the same length as the number of elements of a poset P, we let $\circ^P$ represent this chain. This allows us to avoid using the symbol $\mid$ needlessly.

This lets us write the set cases as posets of the form $\circ \mid \circ \mid \cdots$, and the multiset cases as posets of the form $\circ^{k_1} \mid \circ^{k_2} \mid \cdots$. Furthermore, posets of the form $\circ^{k_1} \mid \circ^{k_2}$ are called binary.

Example 2.2. In the following figure the Hasse diagrams of the posets corresponding with chains, set cases and multiset cases can be seen.



(a) A chain (b) Set cases (c) Multiset cases

Figure 5: Hasse diagrams of chains, set cases and a multiset cases.

Another kind of poset frequently used in literature [2, 5, 6] is called a forest.

Definition 2.7. A forest is a poset such that for every element the subposet of all succeeding elements is a chain. Maximal elements are called roots and minimal elements leafs. A tree is a forest which has only one root (i.e. a greatest element). In this way a forest is just the parallel composition of trees.

Forests generalize the multiset and set cases and can also be created with just parallel and series composition, more specifically using just $\cdot \mid \cdot$ and $\cdot\circ$, where the dots represents some forest to be filled in.

Example 2.3. The following figure shows the Hasse diagram of a forest. Notice that this forest is made out of two trees. It can be, cumbersomely, written as $(\circ|(\circ|\circ)\circ\circ)\circ|\circ\circ$ where series composition has a higher precedence than parallel composition.

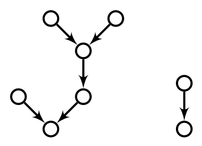


Figure 6: Hasse diagram of a forest.

Forests do not include all posets that can be made using just parallel and series composition. It is also true that not all posets can be made using these compositions. So let us introduce another class of posets.

Definition 2.8. A poset that can be made using just parallel and series composition is called a series-parallel poset, but I will refer to it as a bush.

To recap: neighbour-swap graphs of posets contain those of bushes, who contain those of forests, then multisets and then sets. Figure 7 visualizes this hierarchy. In the next section we will explore some properties of neighbour-swap graphs.

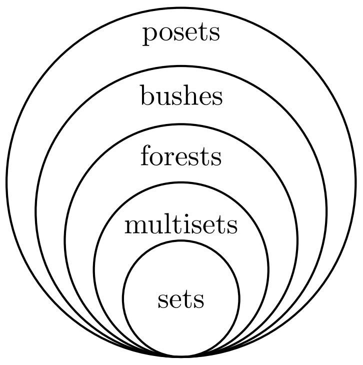


Figure 7: Euler diagram visualizing the hierarchy of neighbours-swap graphs.

3 Properties of neighbour-swap graphs

Before we start searching for Hamiltonian paths and the like we will explore some properties of neighbour-swap graphs. We start by looking at when two posets create the “same” neighbour-swap graph and discuss a natural way to partition neighbour-swap graphs into smaller neighbour-swap graphs. After this we will draw some conclusions when a neighbour-swap graph contains a certain “part”. The section concludes with some trivia.

3.1 Isomorphisms

Unsurprisingly when two posets are isomorphic, i.e. one is just a relabeling of the other, then the neighbour-swap graphs are also isomorphic in the same sense.

Theorem 3.1. *$\mathscr{G}(P) \cong \mathscr{G}(Q)$ if $P \cong Q$.*

Proof. Trivial. $\square$

Any poset P has a dual P^* in which the ordering is reversed. A little more surprising but still trivial is then the following.

Theorem 3.2. *$\mathscr{G}(P) \cong \mathscr{G}(P^*)$ for a poset P.*

Proof. There is a natural bijection between $\mathscr{L}(P)$ and $\mathscr{L}(P^*)$ by reversing the linear extensions. $\square$

If there is an element s in the poset P which is comparable with every other element it becomes unswappable and therefore uninteresting; it adds nothing to the neighbour-swap graph.

Theorem 3.3. *If P has an unswappable element s then $\mathscr{G}(P) \cong \mathscr{G}(P \setminus \{s\})$*

Proof. Notice that s is always within the same spot of every linear extension thus removing and placing s on that location gives the bijection. $\square$

Example 3.1. In the following figure examples of theorems 3.2 and 3.3 can be seen. Namely, 8a and 8b are each other’s dual and 8c is obtained when the unswappable element 3 is removed from 8b.

321 — 312

(a) $\mathscr{G}(3\,(1\,|\,2))$

123 — 213

(b) $\mathscr{G}((1\,|\,2)\,3)$

12 — 21

(c) $\mathscr{G}(1\,|\,2)$

Figure 8: Different posets which all create the same neighbour-swap graph

3.2 Partitioning

Select from a poset some elements. Within a linear extension of the poset is also an ordering of these select elements. This idea yields a natural partition of the linear extensions of a poset into parts defined by the ordering of the select few. This leads to the following definition and theorem.

Definition 3.1. For two poset P and Q, where Q is on a subset of the ground plane of P, define $P \leftharpoonup Q$ as their union. This definition allows us to write down P constrained under more orderings.

Theorem 3.4. *For any subposet $Q \subseteq P$ we can partition $\mathscr{L}(P)$ into $\{\mathscr{L}(P \leftharpoonup L) \mid L \in \mathscr{L}(Q)\}$*

Proof. Take an arbitrary linear extension of P. Within this linear extension the elements of Q have also been placed in an order which, obviously, corresponds with a linear extension of Q. □

We can also partition our neighbour-swap graphs into smaller neighbour-swap graphs in the same way.

Theorem 3.5. *For any subposet $Q \subseteq P$ we can partition $\mathscr{G}(P)$ into subgraphs $\{\mathscr{G}(P \leftharpoonup L) \mid L \in \mathscr{L}(Q)\}$ which are connected to each other isomorphic to $\mathscr{G}(Q)$.*

Proof. The theorem follows from the previous almost directly, the only thing left to prove would be why the subgraphs are connected to each other isomorphic to $\mathscr{G}(Q)$. This is seen by considering a neighbour-swap and splitting it into three different cases: both elements are in Q, one is in Q and the other in $P \setminus Q$, or both are in $P \setminus Q$. Only the first case results in a swap which jumps from subgraph to subgraph which naturally correspond with an edge of $\mathscr{G}(Q)$. □

Corollary 3.5.1. *Every neighbour-swap graph is isomorphic to a subgraph of a set case.*

Example 3.2. In the following figure $\mathscr{G}(1 \mid 2 \mid 34)$ is partitioned with the subposet $1 \mid 2 \mid 3$. The boundaries indicate the partition.

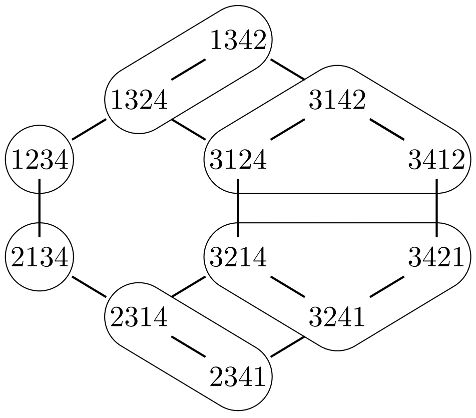


Figure 9: $\mathscr{G}(1 \mid 2 \mid 34)$ partitioned with $1 \mid 2 \mid 3$

3.3 Identification

Certain parts of a neighbour-swap graph reveal a lot of things about the corresponding poset that makes it up. This in turn enforces certain parts upon the neighbour-swap graph. This allows us to quickly identify when a graph is *not* (isomorphic to) a neighbour-swap graph. Sadly, I have been unable to find a way to quickly determine when a graph is (isomorphic to) a neighbour-swap graph.

Theorem 3.6. *If a neighbour-swap graph contains a vertex of degree 1 then the graph is isomorphic to a subgraph of a binary case.*

Proof. A vertex of degree 1 corresponds with a linear extension in which only one neighbour-swap can be made. This means no neighbour-swap can be performed on the elements left of the swap i.e. they are ordered. Similarly, all the elements on the right of the swap are also ordered. We may conclude that the poset can be divided into two chains on which maybe some more orderings are placed. $\square$

The following conjecture seems to be true but I have found no satisfactory proof so far.

Conjecture 3.1. *Every neighbour-swap graph has at at most two vertices of degree 1.*

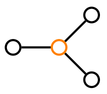
For example binary cases always have two vertices of degree 1. If we add "1 precedes 4" to the poset displayed in figure 2 we would get a neighbour-swap graph with one vertex of degree 1:

$$\begin{tikzcd}
 & & 1342 \arrow[dr, dash] & \\
1234 \arrow[r, dash] & 1324 \arrow[ur, dash] \arrow[dr, dash] & & 3142 \\
 & & 3124 \arrow[ur, dash] &
\end{tikzcd}$$

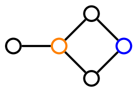
Figure 10: Neighbour-swap graph with one vertex of degree 1.

Vertices of degree 2 do not seem to reveal anything about the graph, but vertices of degree 3 do.

Theorem 3.7. *If a neighbour-swap graph contains a vertex, highlighted in orange, of degree 3:*



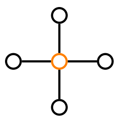
then the graph has at least:



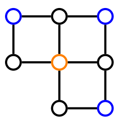
Proof. A vertex of degree 3 corresponds with a linear extension in which three neighbour-swaps can be made. In the worst case these neighbour-swaps all "interfere" which each other, i.e. their locations are all right next to each other. This means the linear extension is of the form ...*abcd*... where $a\&b, b\&c$ and $c\&d$ may be swapped. Aside from the obvious three linear extensions $bacd, acbd, abdc$ we can be certain another exists, namely: *badc*. This fourth linear extension is coloured blue in the figure. ☐

The same can be done for a vertex of degree 4.

Theorem 3.8. *If a neighbour-swap graph contains a vertex, highlighted in orange, of degree 4:*



then the graph has at least:



In general a vertex of degree d reveals as many vertices as the number of ways one can choose two or more adjacent pairs, all disjoint, from a sequence of length $n = d + 1$. Let R_n be this amount.

Theorem 3.9.

$$R_n = F_{n+1} - n, \text{ for } n > 0$$

where F_n is the Fibonacci sequence starting with $F_0 = 0, F_1 = 1$.

Proof. Suppose we have a sequence of length n. We split the problem into two cases whether or not the first pair from the left includes the first element of the sequence. When the first pair does not include the first element there are R_{n-1} ways of still choosing the pairs. If the first pair does include the first element then we must count the number of ways we can still chose *one* or more adjacent pairs, all disjoint, from a sequence of length $n - 2$. This boils down to $R_{n-2} + n - 3$. Combining everything we end up with:

$$R_n = R_{n-1} + R_{n-2} + n - 3, \text{ where } R_1 = R_2 = 0$$

Adding n to both sides together with some small rearranging we can get:

$$R_n + n = (R_{n-1} + n - 1) + (R_{n-2} + n - 2)$$

By defining $S_n = R_n + n$ we reveal the Fibonacci nature of the sequence:

$$S_n = S_{n-1} + S_{n-2}, \text{ where } S_1 = 1, S_2 = 2$$

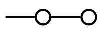
Therefore:

$$R_n = S_n - n = F_{n+1} - n$$

□

R_n is the sequence https://oeis.org/A001924.

Theorem 3.10. *If a neighbour-swap graph contains the following part:*



i.e. a "tail", then the graph is trivial i.e. it is a path.

Proof. We have a vertex of degree 1 so per Theorem 3.6 the corresponding poset consists of two chains with maybe some more orderings. Let us say that the vertex corresponds with the linear extension $\ldots a_2 a_1 b_1 b_2 \ldots$ where a and b are the labels of the two chains. Then the vertex with degree 2 corresponds with $\ldots a_2 b_1 a_1 b_2 \ldots$ and because its degree is 2 it must either be that $a_2 \prec b_1$ or $a_1 \prec b_2$. Without loss of generality let us assume that $a_2 \prec b_1$. The corresponding poset then takes the form of:

$$\begin{tikzcd}
 & & & b_1 \arrow[r] & b_2 \arrow[r] & \cdots \\
\cdots \arrow[r] & a_3 \arrow[r] & a_2 \arrow[ur] \arrow[dr] & & & \\
 & & & a_1 & &
\end{tikzcd}$$

of which the neighbour-swap graph is trivial. Any subgraph of a trivial graph (which is still a neighbour-swap graph) is still trivial thus our original graph is also trivial. □

3.4 Edges

Ruskey in [5] writes about counting the number of edges in neighbour-swap graphs. He deduces a recurrence relation and solves it in the case of multisets. He does not give a proof on how he solved the recurrence and did not notice his expression could be simplified. We give the simplified expression and a proof which does not require solving a recurrence relation.

Definition 3.2. A multinomial coefficient is defined as:

$$\binom{n}{k_1, k_2, \cdots} = \frac{n!}{k_1!k_2!\cdots}$$

Sometimes we will defy convention by not necessarily letting $\sum k_i = n$.

Theorem 3.11. *The number of edges in $\mathscr{G}(M)$, where $M = \circ^{k_1} \mid \circ^{k_2} \mid \cdots$ is a multiset case, equals:*

$$\frac{n^2 - n_2}{2}\binom{n-1}{k_1, k_2, \cdots}$$

where $n = k_1 + k_2 + \cdots, n_2 = k_1^2 + k_2^2 + \cdots$.

Proof. Let us consider the permutations that begin with an element of the chain corresponding with k_1 and one from k_2. Trivially there are

$$2\binom{(k_1-1) + (k_2-1) + k_3 + \cdots}{(k_1-1), (k_2-1), k_3, \cdots}$$

such permutations, which we can rewrite to

$$2k_1k_2\binom{n-2}{k_1, k_2, \cdots}.$$

Because the calculation goes similarly for any choice of two different elements we can deduce that the number of permutations in which there exists a neighbour-swap in the first two elements is equal to

$$\left(\sum_{i \neq j} 2k_ik_j\right)\binom{n-2}{k_1, k_2, \cdots}$$

Now by expanding $n^2 = (k_1 + k_2 + \cdots)^2$ we see that it is equivalent to

$$n_2 + \sum_{i \neq j} 2k_i k_j$$

which we can use to rewrite our previous equation again to

$$\left(n^2 - n_2\right) \binom{n-2}{k_1, k_2, \ldots}$$

These permutations correspond with all the edges in the graph which swap the first two elements so there are

$$\frac{n^2 - n_2}{2} \binom{n-2}{k_1, k_2, \ldots}$$

such edges. Our next observation is that counting all the edges which correspond with a swap of the first two elements is equivalent to counting the edges which correspond to a neighbour-swap in any choice of two adjacent elements. Therefore the total number of edges is simply our previous equation multiplied by $n - 1$:

$$\frac{n^2 - n_2}{2} \binom{n-1}{k_1, k_2, \ldots}$$

The observant reader notices that the proof only works when we are talking about a family of chains consisting of at least two chains. But it turns out our final formula still holds when there is only 1 chain; it properly evaluates to 0. □

3.5 Some more trivia

Theorem 3.12. *$\mathscr{G}\left(\left(\circ^Q \mid P\right) Q\right)$ is the more connected version of $\mathscr{G}\left(Q \mid P\right)$.*

Proof. By taking a linear extension of $\left(\circ^Q \mid P\right) Q$ and "filling in" the elements of Q into the "slots" of $\circ^Q$ gives us a linear extension of $Q \mid P$. An example will clarify this procedure. Lets take $P = 1^3$ and $Q = 2^3$:

$$\circ \circ 1 \circ 11222 \to 221211$$

This procedure is easily inverted and gives us a bijection between the linear extensions of both posets. Now a neighbour-swap in $\mathscr{G}\left(Q \mid P\right)$ will always correspond with a neighbour-swap in $\mathscr{G}\left(\left(\circ^Q \mid P\right) Q\right)$ but the reverse is not true. For example, a neighbour-swap on two elements of Q of which their corresponding "slots" are not adjacent makes it an impossible neighbour-swap in $\mathscr{G}\left(Q \mid P\right)$. □

4 Counting linear extensions and their surplus

Knowing the surplus of a poset is, as previously seen, of great interest because it can tell us directly when a Hamiltonian path/cycle is impossible in the neighbour-swap graph. Counting the number of linear extensions seems to be closely related as we will see. In this section Verhoeff's and Ruskey's proof for the surplus of the multiset cases is given along with Ruskey's more general proof for the surplus of forests.

4.1 Counting linear extensions

Take a linear extension of $P \mid Q$. Intuitively, "woven" within it lays a linear extension of Q. Because the two posets are parallel *any* linear extensions of Q could have been woven through it in exactly the same way. This leads to the following theorem.

Theorem 4.1. $|\mathscr{L}(P \mid Q)| = \left|\mathscr{L}\left(P \mid \circ^Q\right)\right| \cdot |\mathscr{L}(Q)|$

Proof. Using Theorem 3.4 we can partition $\mathscr{L}(P \mid Q)$ into $\{\mathscr{L}(P \mid L) \mid L \in \mathscr{L}(Q)\}$ which are all isomorphic to $\mathscr{L}\left(P \mid \circ^Q\right)$. □

Maybe even more intuitive and simple, a similar thing holds for series composition. A linear extension of PQ is simply a linear extension of P after which we glue one of Q. This leads to the following theorem.

Theorem 4.2. $|\mathscr{L}(PQ)| = |\mathscr{L}(P)| \cdot |\mathscr{L}(Q)|$

Proof. Using Theorem 3.4 we can partition $\mathscr{L}(PQ)$ into $\{\mathscr{L}(PL) \mid L \in \mathscr{L}(Q)\}$ which are all isomorphic to $\mathscr{L}(P)$ per Theorem 3.3. □

Trivially the following is true:

Theorem 4.3. *The number of linear extensions of a binary case is:*

$$\left|\mathscr{L}\left(\circ^{k_1} \mid \circ^{k_2}\right)\right| = \binom{n}{k_1}$$

Example 4.1. To illustrate the use of the previous three theorems let us quickly count the number of linear extension of the poset $\circ(\circ \mid \circ \mid \circ)\left(\circ \mid \circ^3\right)$.

$$\begin{aligned}
\left|\mathscr{L}\left(\circ(\circ \mid \circ \mid \circ)\left(\circ \mid \circ^3\right)\right)\right| &= |\mathscr{L}(\circ)| \cdot |\mathscr{L}(\circ \mid \circ \mid \circ)| \cdot \left|\mathscr{L}\left(\circ \mid \circ^3\right)\right| \\
&= 1 \cdot \left|\mathscr{L}\left(\circ \mid \circ^2\right)\right| \cdot |\mathscr{L}(\circ \mid \circ)| \cdot 4 \\
&= 1 \cdot 3 \cdot 2 \cdot 4 \\
&= 24
\end{aligned}$$

Using just the previous three theorems the following can be proven:

Theorem 4.4. *The number of linear extensions of a forest F is:*

$$|\mathscr{L}(F)| = \frac{n!}{d_1 d_2 \cdots}$$

where d_i is the number of preceding elements, including itself, of the i'th element.

Proof. We will proceed by induction on the number of elements of the forest. For the forest $\circ$ the formula holds and will be the base case. Let the induction hypothesis be that the formula holds for all forest with fewer elements. Because F is a forest it is either the parallel composition of two forests, or the series composition of a forest and $\circ$.

First, suppose $F = F_1 \mid F_2$. Using Theorem 4.1 twice:

$$|\mathscr{L}(F)| = |\mathscr{L}(F_1)| \cdot |\mathscr{L}(F_2)| \cdot \left|\mathscr{L}\left(\circ^{F_1} \mid \circ^{F_2}\right)\right|$$

Using the induction hypothesis and Theorem 4.3:

$$|\mathscr{L}(F)| = \frac{|F_1|!\,|F_2|!}{d_1 d_2 \cdots} \frac{(|F_1| + |F_2|)!}{|F_1|!\,|F_2|!}$$

Which properly reduces to:

$$|\mathscr{L}(F)| = \frac{n!}{d_1 d_2 \cdots}$$

Now suppose $F = F_1 \circ$. We will refer to this added element $\circ$ as the n'th element. Using Theorem 4.2 and the induction hypothesis:

$$|\mathscr{L}(F)| = |\mathscr{L}(F_1)| \cdot |\mathscr{L}(\circ)| = |\mathscr{L}(F_1)| = \frac{(n-1)!}{d_1 d_2 \cdots d_{n-1}}$$

But $d_n = n$ so:

$$|\mathscr{L}(F)| = \frac{n!}{d_1 d_2 \cdots}$$

□

This formula is also stated in [5, Equation 5] but Ruskey gives no proof. Because forests generalize multisets and sets the theorem has two corollaries:

Corollary 4.4.1. *The number of linear extensions of a multiset case is:*

$$\left|\mathscr{L}\left(\circ^{k_1} \mid \circ^{k_2} \mid \cdots\right)\right| = \binom{n}{k_1, k_2, \cdots}$$

Corollary 4.4.2. *The number of linear extensions of a set case is:*

$$|\mathscr{L}(\circ \mid \circ \mid \cdots)| = n!$$

4.2 Counting the surplus

Surprisingly, the same counting rules for the linear extensions hold for the surplus.

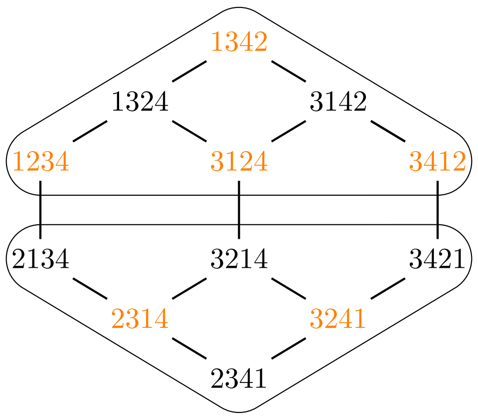
Theorem 4.5. $\mathscr{D}(P \mid Q) = \mathscr{D}\left(P \mid \circ^{Q}\right) \cdot \mathscr{D}(Q)$

Proof. Using Theorem 3.5 we can partition $\mathscr{G}(P \mid Q)$ into $\{\mathscr{G}(P \mid L) \mid L \in \mathscr{L}(Q)\}$ which are all isomorphic to $\mathscr{G}\left(P \mid \circ^{Q}\right)$. Take one of these parts; let us say $\mathscr{G}(P \mid L_1)$. It is either that the even linear extensions in it outweigh the odd or vice versa. Suppose the even outweigh the odd in this case. Now take another part $\mathscr{G}(P \mid L_2)$ for which $L_2 \sim L_1$. Now take any linear extension K of P. The vertex KL_1 in $\mathscr{G}(P \mid L_1)$ and KL_2 in $\mathscr{G}(P \mid L_2)$ have an edge between them; thus it must be that in $\mathscr{G}(P \mid L_2)$ the parity of every extension is opposite to that in $\mathscr{G}(P \mid L_1)$. Therefor, in $\mathscr{G}(P \mid L_2)$ *the odd outweigh the even.* We can deduce from this that the parts of the partition do this "swapping of parity" corresponding to $\mathscr{G}(Q)$. Therefor, the surplus of the isomorphic parts $\mathscr{G}\left(P \mid \circ^{Q}\right)$ mostly cancel each other out, except when $\mathscr{G}(Q)$ has surplus. Thus the surplus of the complete graph $\mathscr{G}(P \mid Q)$ is equal to $\mathscr{D}\left(P \mid \circ^{Q}\right) \cdot \mathscr{D}(Q)$. ☐

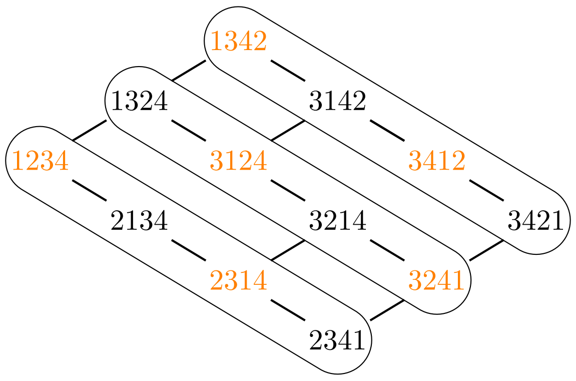
Theorem 4.6. $\mathscr{D}(PQ) = \mathscr{D}(P) \cdot \mathscr{D}(Q)$

Proof. Analogously to the previous theorem but now we use Theorem 3.5 to partition $\mathscr{G}(PQ)$ into $\{\mathscr{G}(PL) \mid L \in \mathscr{L}(Q)\}$ which are all isomorphic to $\mathscr{G}(P)$. ☐

Example 4.2. To illustrate Theorem 4.5 we give an example using the poset $R = 1 \mid 2 \mid 34$ of which the surplus can be manually checked to be 0. In figure 11a we see that, indeed, $\mathscr{D}(R) = \mathscr{D}\left(\circ^2 \mid 34\right) \cdot \mathscr{D}(1 \mid 2) = 2 \cdot 0 = 0$. In figure 11b another way to apply the theorem is seen and again $\mathscr{D}(R) = \mathscr{D}\left(1 \mid \circ^3\right) \cdot \mathscr{D}(2 \mid 34) = 0 \cdot 1 = 0$.



(a) Partitioned with 1 | 2



(b) Partitioned with 2 | 34

Figure 11: Two ways of applying Theorem 4.5 to 1 | 2 | 34.

Using the parallel rule and some knowledge of trivial neighbour-swap graphs we can prove the following:

Theorem 4.7. $\mathscr{D}(\circ^n \mid \circ^m) = 0$ *for odd* n, m*.*

Proof. For clarity we will abuse notation for a second and omit the chain and parallel notation.

$$\begin{aligned}\mathscr{D}(1, n-1, 1, m-1) &= \mathscr{D}(n, m) \cdot \mathscr{D}(1, n-1) \cdot \mathscr{D}(1, m-1) \\ &= \mathscr{D}(1, 1) \cdot \mathscr{D}(2, n-1, m-1) = 0\end{aligned}$$

Which follows from applying Theorem 4.5 on the l.h.s. in two different ways and the last step holds because $\mathscr{D}(1, 1)$ is trivially zero. The trick lays in knowing that $\mathscr{D}(1, k) = 1$ when k is even therefore $\mathscr{D}(n, m)$ must be zero. □

In [10, Theorem 4] Verhoeff proves the following theorem, which was first proven in [2, Theorem 1], in a “new” way. It relies on an involution on the set of linear extensions for which two things hold:

1. Elements, which are not mapped to themselves, map to a element of different parity.

2. All fixed points must be of the same parity.

Computing the surplus is then reduced to finding the fixed points of the involution. The first proof is a short version of Verhoeff’s proof and the second proof we came up with together during a meeting.

Theorem 4.8. *The surplus of a multiset case is:*

$$\mathscr{D}\left(\circ^{k_1} \mid \circ^{k_2} \mid \cdots\right) = \begin{cases} \binom{n \div 2}{k_1 \div 2, k_2 \div 2, \ldots} & \text{if at most one } k \text{ is odd} \\ 0 & \text{if at least two } k \text{ are odd} \end{cases}$$

where $\div$ *denotes division without remainder.*

Proof. Start by splitting a permutation into parts of adjacent pairs called lio (abbreviation for "left index odd") pairs:

$$e_1e_2e_3e_4\cdots \to e_1e_2 \quad e_3e_4 \quad \cdots$$

Which might have a trailing single element. A stutter permutation is a permutation of which all lio pairs consist of two equal elements. Because we are actually working with posets with "equal" we mean "from the same chain". Now, every permutation that does not stutter can be paired with another of the opposite parity by performing a neighbour-swap on the leftmost lio pair that makes it not stutter. It is also the case that all stutter permutations share the same parity. To see this realise that changing the position of a lio pair takes an even number of neighbour-swaps.

This means we have acquired our involution and we need only count the number of stutter permutations to count the surplus. We notice immediately that when two or more k's are odd there exist no stutter permutations and therefor the answer is 0. When at most one k is odd there do exist stutter permutations. To determine their number, coalesce each lio pair into a single element, and drop a possibly trailing single element (which always is the same symbol). We see this is equivalent to $\left|\mathscr{L}\left(\circ^{k_1 \div 2} \mid \circ^{k_2 \div 2} \mid \cdots\right)\right|$ which we already know how to calculate. □

Proof. An alternative proof would be to split a permutation into mirror pairs instead of lio pairs, i.e. the i'th element from the left is paired with the i'th element from the right. If all mirror pairs consist of two equal elements we, obviously, have a palindrome. Now, as before, a non-palindromic permutations can be paired with another of the opposite parity by swapping the innermost mirror pair that makes it not palindromic. And it is again true that all palindromic permutations are of the same parity, because changing the position of a mirror pair takes an even number of neighbour-swaps. We have again acquired our involution and we need only count the number of palindromes to obtain the surplus. □

Verhoeff's proof however is not novel; Ruskey proved it in exactly the same way in [4, Theorem 1]. Ruskey generalizes the result even further for forests in [5, Lemma 3]. Before we can state this theorem we need some more machinery.

Definition 4.1. A pairing in a forest F is a(n) (almost) perfect matching of its transitive reduced Hasse diagram, leaving at most one root unmatched. Such a pairing, if it exists, is unique and can be obtained by constructing it from the leaves towards the root. A collapsed forest $\bar{F}$ is obtained by contracting the elements which are paired and deleting the unmatched root if necessary.

Example 4.3. A forest, its pairing and the collapsed forest can be seen in the following figure. The pairing is marked in orange and the unmatched root in blue.

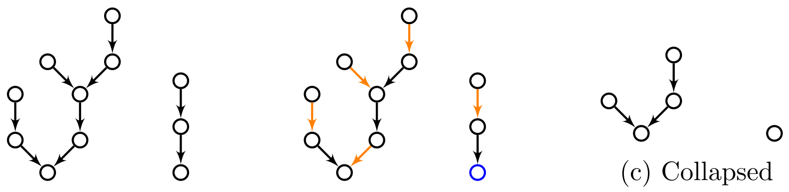

(a) Forest (b) Paired (c) Collapsed

Figure 12

The theorem, which I find extremely elegant, is as follows:

Theorem 4.9. *The surplus of a forest F is:*

$$\mathscr{D}(F) = \begin{cases} \left|\mathscr{L}\left(\bar{F}\right)\right| & \text{if } F \text{ has a pairing} \\ 0 & \text{otherwise} \end{cases}$$

Proof. We again split a linear extension into lio pairs. This time a stutter extension is a linear extension where no neighbour-swap can be performed within the lio pairs. We pair every linear extension that does not stutter with another of the different parity by performing a neighbour-swap on the leftmost lio pair that makes it not stutter. Because the poset is a forest all stutter extensions consist of the same lio pairs but in different orders. This corresponds with the uniqueness of the pairing of the forest. Thus all stutter extensions have the same parity and we have found our involution. We need only count the number of stutter extensions which is the same as counting the linear extensions of the collapsed forest. □

5 Constructing shortest covering walks

As seen in Theorem 2.3 knowing the surplus of a neighbour-swap graph can quickly tell us when a Hamiltonian path (cycle) is impossible. Shockingly the converse of Theorem 2.3.1 seems to be true i.e. a surplus less than or equal to 1 implies the existence of a Hamiltonian path. We start this section with some theorems related to constructing Hamiltonian paths and continue with shortest covering walks. We end this section with a discussion on the "master conjecture".

5.1 Hamiltonian

The Steinhaus-Johnson-Trotter algorithm can be generalized to posets. Ruskey also states this in [5, Lemma 2].

Theorem 5.1. *If $\mathscr{G}(P)$ has a Hamiltonian path then so does $\mathscr{G}(P \mid \circ)$.*

Corollary 5.1.1. *$\mathscr{G}(\circ \mid \circ \mid \cdots)$, i.e. a set case, has a Hamiltonian path.*

Which seems to be a specific case ($k = 1$) of the following new theorem which is visualized in figure 13a.

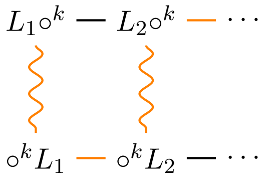
Theorem 5.2. *If $\mathscr{G}(P)$ and $\mathscr{G}\left(\circ^P \mid \circ^k\right)$ have Hamiltonian paths then so does $\mathscr{G}\left(P \mid \circ^k\right)$.*

Proof. We partition $\mathscr{G}\left(P \mid \circ^k\right)$ into $\{\mathscr{G}\left(L_1 \mid \circ^k\right), \mathscr{G}\left(L_2 \mid \circ^k\right), \dots\}$, where $\{L_1, L_2, \dots\}$ is a Hamiltonian path of P, which are all isomorphic to $\mathscr{G}\left(\circ^P \mid \circ^k\right)$ which have, by assumption, Hamiltonian paths. The Hamiltonian paths in these subgraphs must have their ends at $L_j\circ^k$ and $\circ^k L_j$ because these vertices have degree 1. By connecting these endpoints the right way we create a Hamiltonian path through $\mathscr{G}\left(P \mid \circ^k\right)$. □

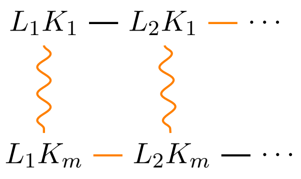
Analogously to how a Hamiltonian path can be found in the Cartesian product of two Hamiltonian graphs, we can prove the following new theorem:

Theorem 5.3. *If both $\mathscr{G}(P)$ and $\mathscr{G}(Q)$ have Hamiltonian paths then so does $\mathscr{G}(PQ)$.*

Proof. We partition $\mathscr{G}(PQ)$ into $\{\mathscr{G}(L_1Q), \mathscr{G}(L_2Q), \dots\}$, where $\{L_1, L_2, \dots\}$ is a Hamiltonian path of P. All these subgraphs are isomorphic to $\mathscr{G}(Q)$ which has a Hamiltonian path and we let K_1 and K_m denote the ends of this path. As can be seen in 13b the task of finding a Hamiltonian path in $\mathscr{G}(PQ)$ is then easily done. □



(a) Illustration of Theorem 5.2.



(b) Illustration of Theorem 5.3.

Figure 13: Illustration of Theorems 5.2 and 5.3. The Hamiltonian paths are given in orange.

Using our counting rules for the surplus and the previous theorems we can prove the following new result.

Theorem 5.4. *For a bush B if $\mathscr{D}(B) = 1$ then $\mathscr{G}(B)$ has a Hamiltonian path.*

Proof. We will proceed by induction on the number of elements of the bush. $\mathscr{G}(\circ)$ trivially has a Hamiltonian path and will be the base case. Let the induction hypothesis be that the statement is true for all bushes with fewer elements. Because B is a bush it is either the parallel or series composition of two bushes.

First, suppose $B = P \mid Q$ for some bushes P and Q. Without loss of generality we may assume $|Q| \leq |P|$. Then using Theorems 4.5 and 4.8:

$$\begin{aligned}\mathscr{D}(P \mid Q) = 1 &\Leftrightarrow \mathscr{D}(P) \cdot \mathscr{D}(Q) \cdot \mathscr{D}\left(\circ^P \mid \circ^Q\right) = 1\\ &\Leftrightarrow \mathscr{D}(P) = 1 \wedge \mathscr{D}(Q) = 1 \wedge \mathscr{D}\left(\circ^P \mid \circ^Q\right) = 1\\ &\Rightarrow (|Q| = 1 \wedge |P| = \text{even})\end{aligned}$$

So, actually, $B = P \mid \circ$ and $\mathscr{D}(P) = 1$. Also $|P| < |B|$ so using the induction hypothesis $\mathscr{G}(P)$ has a Hamiltonian path. Applying Theorem 5.1 gives us a Hamiltonian path in B.

Now suppose $B = PQ$ for some bushes P and Q. Using Theorem 4.6:

$$\begin{aligned}\mathscr{D}(PQ) = 1 &\Leftrightarrow \mathscr{D}(P) \cdot \mathscr{D}(Q) = 1\\ &\Leftrightarrow \mathscr{D}(P) = 1 \wedge \mathscr{D}(Q) = 1\end{aligned}$$

Again $|P|, |Q| < |B|$ so using the induction hypothesis $\mathscr{G}(P)$ and $\mathscr{G}(Q)$ have Hamiltonian paths. Applying Theorem 5.3 gives us a Hamiltonian path in B. □

Stachowiak proves in [7, Main Theorem] that:

Theorem 5.5. *If $\mathscr{G}(P)$ has a Hamiltonian path and $\mathscr{D}(P) = 0$ then so does $\mathscr{G}(P \mid Q)$ for every poset Q.*

Actually, Stachowiak states his theorem a little differently but is equivalent to this form. Assuming the neighbour-swap graph of two odd chains has a Hamiltonian path Stochawiak is able to easily prove the following:

Theorem 5.6. *Let M be a multiset case. If $\mathscr{D}(M) \leq 1$ then $\mathscr{G}(M)$ has a Hamiltonian path.*

Proof. From Theorem 4.8 it can be deduced that $\mathscr{D}(M) = 1$ only happens when $\mathscr{G}(M)$ is trivial i.e. it is a path. When $\mathscr{D}(M) = 0$ there must exist at least two chains of odd length, which can then be used as the basis for Theorem 5.5 to show $\mathscr{G}(M)$ has a Hamiltonian path. These two chains can be used as the basis because the surplus of two odd chains is indeed 0. $\square$

This theorem was conjectured by Ko and Ruskey in [2,4] but only proven for transpositions (not neighbour-swaps!) in [5, Theorem 1]. Stachowiak's work is a big step towards proving a conjecture by Ruskey [5, Question 1]:

Conjecture 5.1. *If $\mathscr{D}(P) = 0$ then $\mathscr{G}(P)$ has a Hamiltonian path for every poset P.*

Lehmer in [3] asks the question if, for *any* given multiset case M, one can always make $\mathscr{G}(M \mid N)$ Hamiltonian with some N. This is also answered by Stochawiak who starts [7, Lemma 4&6] by proving the following:

Theorem 5.7. *$\mathscr{G}(P \mid \circ \mid \circ)$ has a Hamiltonian path for every poset P.*

Which, of course, also follows from Stachowiak's main result (although circularly).

5.2 Lehmer

Searching only for Hamiltonian paths is quite restrictive; we miss out on a large number of neighbour-swap graphs. So let us broaden our view and instead search for shortest covering walks. Lehmer does the same in [3] and conjectures the following:

Conjecture 5.2. *$\mathscr{G}(M)$ has a Lehmer path with $\mathscr{D}(M) - 1$ single spurs for every multiset case M.*

Remark 5.2.1. When the surplus is ≤ 1 we mean there are no spurs in the Lehmer path i.e. a Hamiltonian path.

A Lehmer path, as Verhoeff calls them in [10], is like a Hamiltonian path but is allowed to be imperfect. The imperfections take the form of so-called spurs. Ruskey conjectures a similar thing in [5, Question 8] but calls a Lehmer cycle a cyclic comb.

Definition 5.1. An unvisited vertex u at distance 1 can be visited by sidestepping from a vertex v on the path to u and then immediately back again to v. That is, the same swap is done twice in a row. In the resulting path, v occurs twice, with only u in-between. Lehmer calls such a sidestep to an unvisited vertex at distance 1 a spur. 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.

Notice that Lehmer's conjecture is a more general version of Theorem 5.6 because when the surplus is ≤ 1 a Lehmer path reduces to a Hamiltonian path. Verhoeff managed to prove Lehmer's conjecture for the binary case in [10]. He does so by placing the tips of the spurs at the stutter permutations. Verhoeff mentioned to me during one of our meetings that Lehmer's conjecture is inductively a lot nicer because you always have something to work with. This is in contrast with the "Hamiltonian conjectures" which, during your inductive proof, might leave you with a neighbour-swap graph which has no Hamiltonian path.

In [3] Lehmer gives an heuristic algorithm for generating Lehmer paths in multiset cases which, for the few cases he applied it to, always gave paths that comply to his conjecture.

5.3 Master

All these conjectures and theorems can be generalized into the following new "master" conjecture:

Conjecture 5.3. *$\mathscr{G}(P)$ has a Lehmer path with $\mathscr{D}(P) - 1$ single spurs for every poset P.*

This generalizes Lehmer's conjecture 5.2 by restating it for posets instead of multisets, and it generalizes Conjecture 5.1 by Ruskey (in the case where $\mathscr{D}(P) > 0$).

6 Conclusion

There is still *a lot* left to do. The master conjecture seems to be true but I have no idea how to tackle it for posets in general. Posets that are made using parallel and series compositions seem to be the only ones that “play nicely” and therefor this whole paper circles around these compositions. This is also why I think that if someone is able to prove the master conjecture it probably will not involve these compositions; they are just not general enough. If I were to continue my research in this subject I would focus on finding what makes a graph a neighbour-swap graph. This would allow one to abstract away from the posets and focus purely on what really matters. Furthermore, Lehmer’s heuristic algorithm for generating Lehmer paths in multiset cases can be generalized to posets which, if it still works, might give insights into solving the master conjecture.

The main contribution of this paper is that it summarizes what is known about neighbour-swap graphs, and shortest covering walks of them, so far. Some new results are shared (Theorems 3.6, 3.7, 3.9, 3.11, 4.5, 4.6, 4.7, 5.2, 5.3, 5.4, and Conjecture 5.3) but the “big” theorems (4.8, 4.9, 5.5) are mostly the work of others. I hope that this paper clarifies the subject by making the important insights more explicit and formulating them in a uniform matter.

7 References

[1] S.M. Johnson. Generation of permutations by adjacent transposition. *Mathematics of Computation*, 17(83):282–285, 1963.

[2] C.W. Ko and F. Ruskey. Solution of some multi-dimensional lattice path parity difference recurrence relations. *Discrete Mathematics*, 71(1):47–56, 1988.

[3] D.H. Lehmer. Permutation by adjacent interchanges. *The American Mathematical Monthly*, 72(sup2):36–46, 1965.

[4] F. Ruskey. Solution of some lattice path parity difference recurrence relations using involutions. *Congressus Numerantium*, 59:257–266, 1987.

[5] F. Ruskey. Generating linear extensions of posets by transpositions. *Journal of Combinatorial Theory, Series B*, 54(1):77–101, 1992.

[6] G Stachowiak. Finding parity difference by involutions. *Discrete Mathematics*, 163(1):139–151, 1997.

[7] G. Stachowiak. Hamilton Paths in Graphs of Linear Extensions for Unions of Posets. *SIAM Journal on Discrete Mathematics*, 5(2):199–206, 2005.

[8] H. Steinhaus. *One Hundred Problems in Elementary Mathematics.* Dover Books on Mathematics Series. Dover Publications, 1979.

[9] H.F. Trotter. Algorithm 115: Perm. *Commun. ACM*, 5(8):434–435, August 1962.

[10] T. Verhoeff. The spurs of D.H. Lehmer: Hamiltonian paths in neighbor-swap graphs of permutations. *Designs, Codes, and Cryptography*, 81(1):295–310, 2017.