Efficient Enumeration of All Ladder Lotteries and Its Application

Katsuhisa Yamanaka[a], Shin-ichi Nakano[b], Yasuko Matsui[c], Ryuhei Uehara[d], Kento Nakada[e]

[a] *Graduate School of Information Systems, University of Electro-Communications, Chofugaoka 1-5-1, Chofu, Tokyo 182-8585, Japan.*
[b] *Department of Computer Science, Gunma University, Tenjin-cho 1-5-1, Kiryu, Gunma 376-8515, Japan.*
[c] *Department of Mathematical Sciences, Tokai University, Kitakaname 1117, Hiratsuka, Kanagawa, 259-1292, Japan.*
[d] *School of Information Science, Japan Advanced Institute of Science and Technology, Asahidai 1-1, Nomi, Ishikawa 923-1292, Japan.*
[e] *Faculty of Integrated Media, Wakkanai Hokusei Gakuen University, Wakabadai 1-2290-28, Wakkanai, Hokkaido 097-0013, Japan*

Abstract

A *ladder lottery*, known as the "Amidakuji" in Japan, is a common way to choose a permutation randomly. A ladder lottery L corresponding to a given permutation π is *optimal* if L has the minimum number of horizontal-lines among ladder lotteries corresponding to π. In this paper we show that for any two optimal ladder lotteries L_1 and L_2 of a permutation, there exists a sequence of local modifications which transforms L_1 into L_2. We also give an algorithm to enumerate all optimal ladder lotteries of a given permutation. By setting $\pi = (n, n-1, \ldots, 1)$, the algorithm enumerates all arrangements of n pseudolines efficiently. By implementing the algorithm we compute the number of arrangements of n pseudolines for each $n \leq 11$.

Key words: enumeration algorithm, family tree, ladder lottery, pseudoline arrangement

Email addresses: yamanaka@is.uec.ac.jp (Katsuhisa Yamanaka), nakano@cs.gunma-u.ac.jp (Shin-ichi Nakano), yasuko@ss.u-tokai.ac.jp (Yasuko Matsui), uehara@jaist.ac.jp (Ryuhei Uehara), nakada@wakhok.ac.jp (Kento Nakada)

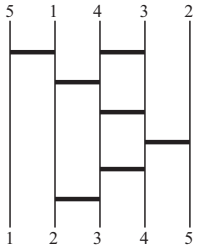

Figure 1: An optimal ladder lottery of the permutation (5,1,4,3,2).

1. Introduction

A *ladder lottery*, known as the "Amidakuji" in Japan, is a common way to choose a permutation randomly. Formally, a ladder lottery L of a permutation $\pi = (p_1, p_2, \ldots, p_n)$ is a network with n vertical lines (*lines* for short) and many horizontal lines (*bars* for short) each of which connects two consecutive vertical lines. The i-th line from the left is called *line* i. The top ends of lines correspond to π. The bottom ends of lines correspond to the identical permutation $(1, 2, \ldots, n)$. See Figure 1. Each number p_i in π starts the top end of line i, and goes down along the line, then whenever p_i comes to an end of a bar, p_i goes horizontally along the bar to the other end, then goes down again. Finally p_i reaches the bottom end of line p_i. We can regard a bar as a modification of the current permutation, and a sequence of such modifications in a ladder lottery always results in the identical permutation $(1, 2, \ldots, n)$.

A ladder lottery of a permutation $\pi = (p_1, p_2, \ldots, p_n)$ is *optimal* if it consists of the minimum number of bars among ladder lotteries of π. Let L be an optimal ladder lottery of π and m be the number of bars in L. Then we can observe that m is equal to the number of "inversions" of π, which is a pair (p_i, p_j) in π with $p_i > p_j$ and $i < j$. The ladder lottery in Figure 1 has seven bars, and permutation (5,1,4,3,2) has seven inversions: (5,1), (5,4), (5,3), (5,2), (4,3), (4,2) and (3,2), so the ladder lottery is optimal.

The ladder lotteries are strongly related to primitive sorting networks, which are deeply investigated by Knuth [4]. A comparator in a primitive sorting network replaces p_i and p_{i+1} by $\min(p_i, p_{i+1})$ and $\max(p_i, p_{i+1})$, while a bar in a ladder lottery always exchanges them.

In this paper we give an efficient algorithm to enumerate all optimal ladder lotteries of a given permutation π.

n	The number of combinatorial structures of arrangements of n pseudolines
1	1
2	1
3	2
4	8
5	62
6	908
7	24,698
8	1,232,944
9	112,018,190
10	18,410,581,880
11	5,449,192,389,984

Table 1: The number of arrangements of $n \leq 11$ pseudolines

One of our motivations lies in algebraic combinatorics on Coxeter groups [3]. Let W be a Coxeter group. Given $\pi \in W$, to compute the number of commutativity classes [11] of all reduced decompositions of π is one of important unsolved problems, even for an element of the Coxeter group of type A_{n-1}. By using our algorithm, one can compute the complete list of the commutativity classes of π in the symmetric group $\mathfrak{S}_n$ of degree n (the Coxeter group of type A_{n-1}). In this case, one can regard the set of commutativity classes of π as the set of optimal ladder lotteries of π.

By setting $\pi = (n, n-1, \ldots, 1)$, our algorithm can efficiently enumerate all arrangements of n pseudolines. A pseudoline is an x-monotone curve in the plane, and an arrangement is a set of pseudolines in which every pair of pseudolines intersects exactly once. Arrangements of pseudolines are one of important and appealing objects in the area of geometry and combinatorics.

By implementing our algorithm we compute the number of arrangements of n pseudolines for each $n \leq 11$ (Table 1). Only the numbers for $n \leq 10$ were known [4, 13] and this is the first report for $n = 11$.

In this paper we show a result similar to the Wagner's theorem. In 1936 Wagner [12] showed that, for any two maximal planar graphs G_1 and G_2 having the same number of vertices, there exists a sequence of local modifications, called diagonal flip, which transforms G_1 into G_2.

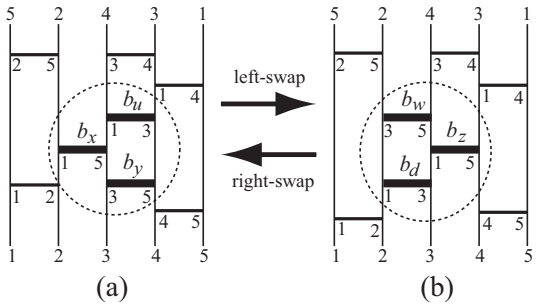

Figure 2: A local swap operation.

Let S_π be the set of all optimal ladder lotteries of a given permutation π. A *local swap operation*, which corresponds the notion of "braid relation" in the area of algebra, is a local modification of a ladder lottery as shown in Figure 2. Note that applying a local swap operation to $L_1 \in S_\pi$ results in other $L_2 \in S_\pi$, since the local swap operation preserves the local permutation. We show that, for any two optimal ladder lotteries L_1 and L_2 of a permutation, there exists a sequence of local swap operations which transforms L_1 into L_2.

We also give an algorithm to enumerate all optimal ladder lotteries of a given permutation. Our algorithm generates all ladder lotteries in $O(1)$ time for each in worst case. To the best of our knowledge this is the first such algorithm.

The idea of our enumeration algorithm is as follows. We first define a tree structure T_π, called *family tree* (see Figure 3), among ladder lotteries in S_π, in which each vertex of T_π corresponds to each ladder lottery in S_π and each edge of T_π corresponds to a relation between two ladder lotteries which can be transformed to the other by one local swap operation. Then we design an efficient algorithm to generate all child vertices of a given vertex in T_π. Applying the algorithm recursively from the root of T_π, we can generate all vertices in T_π, and also corresponding all ladder lotteries in S_π. Based on such tree structure and some other ideas a lot of efficient enumeration algorithms are designed [1, 7, 8, 9, 14].

The rest of the paper is organized as follows. Section 2 gives some definitions. Section 3 defines the tree structure among ladder lotteries in S_π. Section 4 gives an efficient algorithm to enumerate all ladder lotteries in S_π. Finally Section 5 is a conclusion.

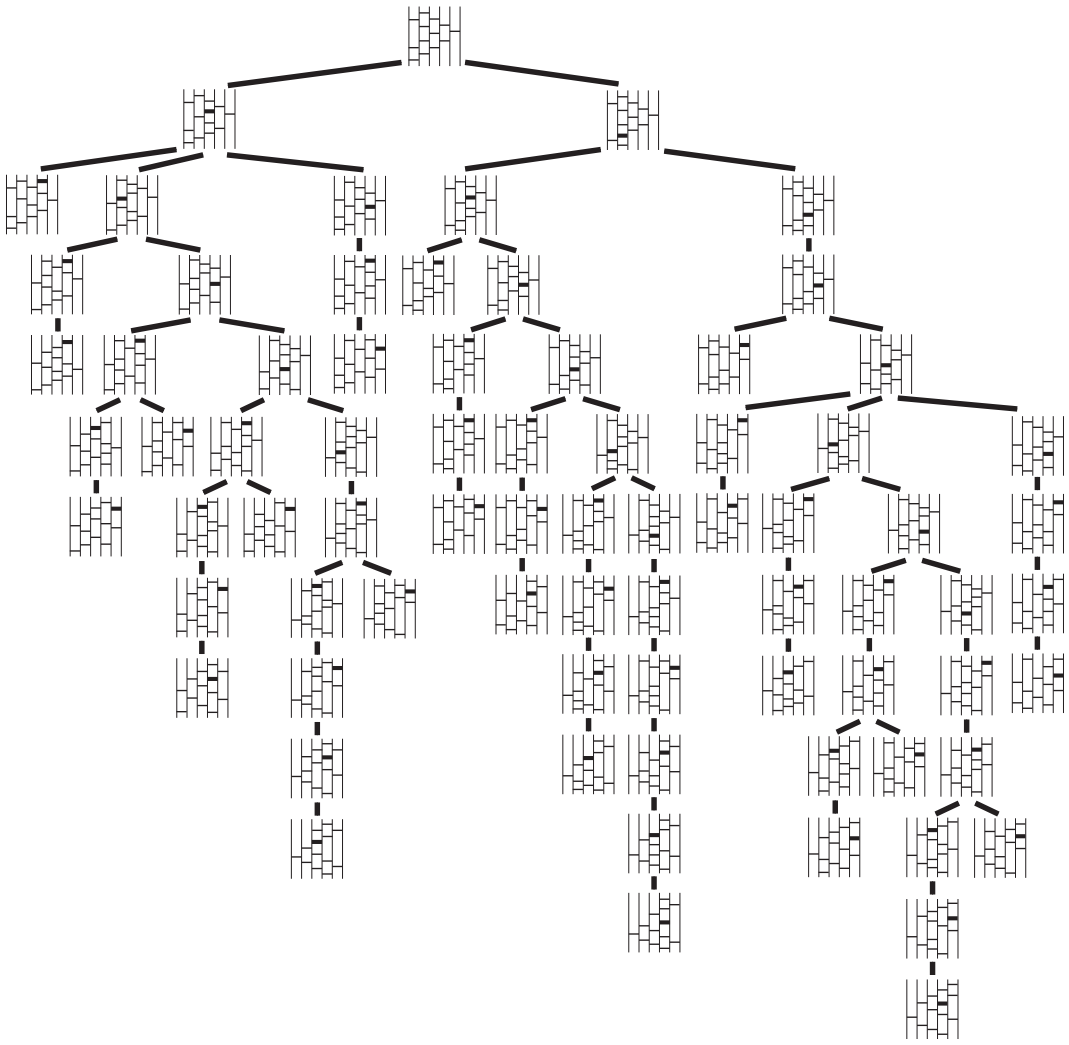

Figure 3: The family tree T_π, where $\pi = (5, 6, 3, 4, 2, 1)$.

2. Preliminary

A *ladder lottery* L of a permutation $\pi = (p_1, p_2, \ldots, p_n)$ is a network with n vertical lines (*lines* for short) and many horizontal lines (*bars* for short) each of which connects two consecutive vertical lines. The i-th line from the left is called *line* i. The top ends of the n lines correspond to π. The bottom ends of the n lines correspond to the identical permutation $(1, 2, \ldots, n)$. See Figure 1. Each number p_i in π starts the top end of line i, and goes down along the line, then whenever p_i comes to an end of a bar p_i goes to the other end and goes down again, then finally p_i reaches the bottom end of line p_i. This path is called *the route* of number p_i. We can regard a bar as a modification of the current permutation, and a sequence of such modifications in a ladder lottery always results in the identical permutation $(1, 2, \ldots, n)$.

Let $\pi = (p_1, p_2, \ldots, p_n)$ be a permutation. An *inversion* of π is a pair (p_i, p_j) with $p_i > p_j$ and $i < j$. Let m be the number of inversions of π. We can observe that any ladder lottery of π contains at least m bars, since each bar "cancels" at most one inversion of the "current" permutation (see, e.g., [5, 5.3.4 Figure 45]). If a ladder lottery L contains exactly m bars, then we say that L is *optimal*.

A *local swap operation* is a local modification of a ladder lottery as shown in Figure 2. Note that the dashed circle contains only three bars. Also note that applying this modification to an optimal ladder lottery of π results in other optimal ladder lottery of π, since the local swap operation preserves the local permutation. A local swap operation (a) to (b) in Figure 2 is called a *left swap operation to bar* b_u and we say b_u *is left swapped*. Note that in Figure 2 the left swap operation moves bar b_u from the (upper) right of the route of 5 to the (lower) left. We say *a bar* b_u *can be left swapped* if (i) there is no endpoint between the right end of b_u and the right end of a bar b_y locating below b_u, and (ii) there is exactly one right end of a bar b_x between the left end of b_u and the left end of b_y (see Figure 2(a)). Similarly, a local swap operation (b) to (a) in Figure 2 is called a *right swap operation to bar* b_d and we say b_d *is right swapped*. Note that the operation moves bar b_d from the (lower) left of the route of 5 to the (upper) right. We say *a bar* b_d *can be right swapped* if (i) there is no endpoint between the left end of b_d and the left end of a bar b_w locating above b_d, and (ii) there is exactly one left end of a bar b_z between the right end of b_d and the right end of b_w (see Figure 2(b)).

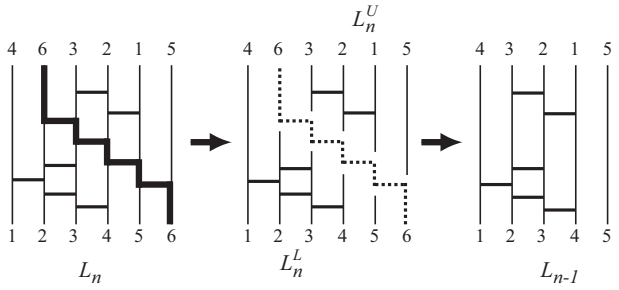

Figure 4: The removal of the n-path.

3. Family Tree

In this section we design a tree structure T_π among optimal ladder lotteries of a given permutation π. Each vertex of T_π corresponds to each optimal ladder lottery of π, and each edge corresponds to each relation between two ladder lotteries which can be transformed to the other by one local swap operation. This means that, for any two optimal ladder lotteries L_1 and L_2 of a permutation π, there exists a sequence of local swap operations which transforms L_1 into L_2.

Let S_π be the set of optimal ladder lotteries of a given permutation $\pi = (p_1, p_2, \ldots, p_n)$ and $L = L_n$ be a ladder lottery in S_π. Choose i to be $p_i = n$. Observe the route of number p_i. Since L_n is optimal, every bar in the route of n cancels exactly one inversion of the "current" permutation. Thus the route of n contains exactly $n - i$ bars. The intersection of this route and any line consists of at most one interval. We call such a route *monotone*. We note that this property does not necessarily hold for general p_i; for example, in Figure 4, the route of $p_4 = 2$ is not monotone while the ladder lottery is optimal.

The route of n partitions L_n into the upper part L_n^U and the lower part L_n^L. Removing the route of n from L_n then patching L_n^U and L_n^L, as shown in Figure 4, results in an optimal ladder lottery L_{n-1} of the permutation $(p_1, p_2, \ldots, p_{i-1}, p_{i+1}, \ldots, p_n)$ of $(1, 2, \ldots, n-1)$. Note that it is possible to be $p_n = n$, and in such case L_n^U is empty. We say L_n is *n-clean* if L_n^U has no bar. If L_n is n-clean then the $(n-1)$-path of L_{n-1} is also monotone and we can define L_{n-2} similarly, and we say L_{n-1} is $(n-1)$-clean if L_{n-1}^U has no bar. We repeat this process until some non-clean ladder lottery appears or the ladder lottery becomes empty. If L_k is k-clean for each $k = 1, 2, \ldots, n$, then L is called *the root ladder lottery* of π, denoted by R. Otherwise we say

clean level k if L is i-clean for each i with $n \geq i \geq k$ and L is not $(k-1)$-clean. Especially if L_n^U has a bar, then L has the clean level $n+1$, and the root R has the clean level 1.

The following lemma shows the root ladder lottery in S_π is unique.

Lemma 1. *The root ladder lottery R in S_π is unique.*

PROOF. Assume we have two root ladder lotteries $R, R' \in S_\pi$ and $R \neq R'$. Since $R \neq R'$, there exists i such that $R_i \neq R'_i$. We choose the minimum such i. If the route of i starts from the top end of line k in R_i, then it starts from the top end of line $l \neq k$ in R'_i. This means the corresponding permutations are different, which is a contradiction. □

Let L be an optimal ladder lottery having the clean level k. We can observe that the routes of numbers $n, n-1, \ldots, k$ form so called "brick structure," as follows. For $p_j \geq k$, let $(q_1, q_2, \ldots, q_{n_L})$ be the decreasing list of numbers each of which is larger than p_j and locating to the left of p_j in π. In L, the route of p_j first go left n_L times, along the bars sharing with the routes of $q_1, q_2, \ldots, q_{n_L}$, corresponding to the patches, then go right $p_j - j + n_L$ times. Note that on the right side of the route of p_j every bar is on the route of some number larger than p_j. Also L has at least one bar in the region below the route of number k and above the route of number $k-1$. See Figure 5. The region is called *the active region* of L. If L has clean level $k = n+1$, then we define the region above the route of n as the active region. Especially, we define the active region of R is empty for convenience (in the proof of Lemma 3).

Now we assign a *parent ladder lottery* in S_π for each ladder lottery L in $S_\pi - \{R\}$ as follows. We assume that L has the clean level k. Thus the active region of L has at least one bar. We say a bar b, where we assume b has ends on line l and $l+1$, is *upward visible from* $k-1$ if the lowest end of a bar on l above the route of $k-1$ is the end of b, and also the lowest end of a bar on $l+1$ above the route of $k-1$ is the end of b. Note that if b is upward visible from $k-1$ then no upward visible bar from $k-1$ has an end on line l nor $l+1$. Thus the number of the upward visible bars from $k-1$ is at most $\frac{n}{2}$. Among the upward visible bars from $k-1$, the rightmost bar is called *the active bar* of L. In Figure 5, bar b is the active bar. For any $L \in S_\pi - \{R\}$, applying a left swap operation to the active bar results in a ladder lottery, denoted by $P(L)$, in S_π. We say $P(L)$ is *the parent ladder lottery* of L, and

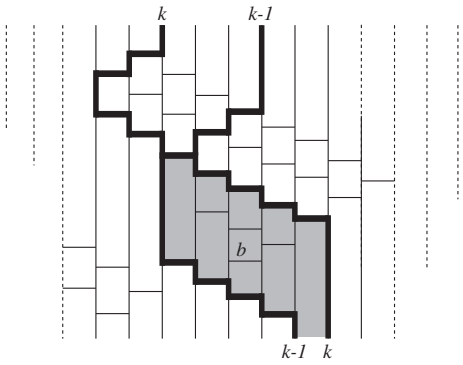

Figure 5: The region.

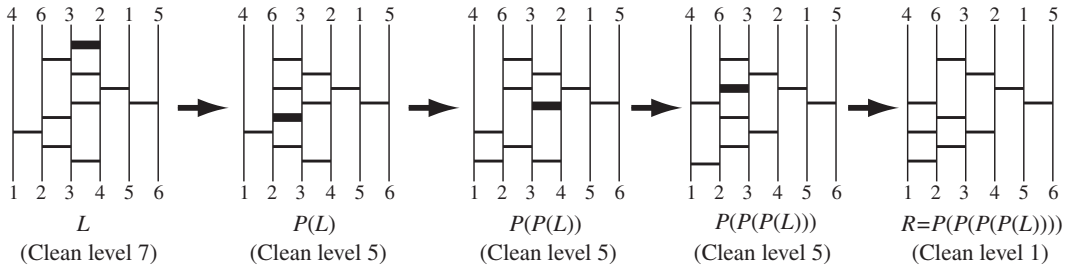

Figure 6: A sequence of optimal ladder lotteries L of (4,6,3,2,1,5).

L is a *child ladder lottery* of $P(L)$. Note that the parent ladder lottery of L is unique, while $P(L)$ may have many children. Also note that the clean level of $P(L)$ is smaller or equal to L, and $P(L)$ has less bars in the active region of L.

We have the following lemma.

Lemma 2. *For any $L \in S_\pi - \{R\}$, $P(L) \in S_\pi$ holds.*

Proof. Since the swap operation reserves the local permutation. □

Given a ladder lottery L in $S_\pi - \{R\}$, by repeatedly finding the parent ladder lottery of the derived ladder lottery, we can have the unique sequence $L, P(L), P(P(L)), \ldots$ of ladder lotteries in S_π, as shown below, which eventually ends up with the root R in S_π. See Figure 6. The active bars are depicted by thick lines.

We have the following lemma.

Lemma 3. *The sequence* $L, P(L), P(P(L)), \ldots$ *of* $L \in S_\pi - \{R\}$ *ends with* $R \in S_\pi$.

PROOF. For each $L \in S_\pi$ we define *the clean potential* C of L as $C(L) = (s, t)$, where s is the clean level of L and t is the number of bars in the active region of L. For $L_1, L_2 \in S_\pi$ with $C(L_1) = (s_1, t_1)$ and $C(L_2) = (s_2, t_2)$, we say L_1 *is cleaner than* L_2 if (1) $s_1 < s_2$ or (2) $s_1 = s_2$ and $t_1 < t_2$. For any $L \in S_\pi$ we can observe $P(L)$ is cleaner than L. R is the cleanest among S_π, since $C(R) = (1, 0)$. Thus for any $L \in S_\pi$ the sequence of clean potentials $C(L), C(P(L)), C(P(P(L))), \ldots$ always ends at $C(R)$. □

By merging all these sequences we can have a *family tree* of S_π, denoted by T_π, in which the root of T_π corresponds to R, the vertices of T_π correspond to the ladder lotteries in S_π and each edge corresponds to a relation between a ladder lottery in S_π and its parent. See Figure 3. The active bars are depicted by thick lines.

Thus we have the following theorem.

Theorem 4. *For any two ladder lotteries* L_1 *and* L_2 *in* S_π*, there exists a sequence of operations consisting of zero or more left swap operations followed by zero or more right swap operations which transforms* L_1 *into* L_2.

4. Enumerating All Optimal Ladder Lotteries

In this section we give an efficient algorithm to enumerate all ladder lotteries in S_π.

If we have an algorithm to enumerate all children of a given ladder lottery in S_π, then by recursively applying the algorithm starting at the root R of S_π, we can enumerate all ladder lotteries in S_π. Now we design such an algorithm.

We need some definitions. Let $L \neq R$ be an optimal ladder lottery of a given permutation $\pi = (p_1, p_2, \ldots, p_n)$. Assume L has the clean level k. So each bar locating on the right of the route of k is contained in some route of $x > k$, but in the active region (see Figure 5) there is at least one bar which is not contained in any route of $x \geq k - 1$. Each route of $x \geq k - 1$ goes left along bars, "turns," and goes right along bars. For each route of $x \geq k - 1$ if b is the first bar to go right after a bar to go left, then b is called *the turn bar* of x. Note that only if the route of x contains both at least one bar to left

and one bar to right, the route of x has the turn bar. Otherwise if the route of x contains only bars to left (or right) then the route of x is monotone in L and has no turn bar. Also note that the turn bar is defined only for the route of $x \geq k-1$ since otherwise it may have many "turns." In the next lemma we show that on the route of each $x = k-1, k, \ldots, n$, only the turn bar has a chance to be right swapped.

Lemma 5. *Assume L be a ladder lottery having the clean level k. Then on the route of $x \geq k-1$ only the turn bar has a chance to be right swapped.*

Proof. Since L has the clean level k, the route of each $x \geq k-1$ first goes left along bars, turns, and goes right along bars. As shown in Figure 2(b), a bar b_d can be right swapped only if the vertical segment between the left end of b_d and the left end of a bar b_w locating above b_d has no right end of other bars. Such condition is satisfied only at the turn bar of some route of $x \geq k-1$. □

Let $L[b]$ be the ladder lottery which is derived from L by applying a right swap operation to a bar b. Every child of L is $L[b]$ for some b, but not all $L[b]$s were children of L. $L[b]$ is a child of L only if b is the active bar of $L[b]$. Now we classify whether $L[b]$ is a child ladder lottery of L or not as follows. Remember the clean level of L is k. Let $R(x)$ be the region on the right side of the route of x, and $L(x)$ be the region on the left side of the route of x.

Type 1: b is a turn bar and can be right swapped.

Note that such a bar exists only on the routes of $k-1, k, \ldots, n$. Assume the local structure of L is as shown in Figure 7(a), the routes of x, y and z pass through there and b is the turn bar of the route of y. Since b is the turn bar of the route y, $y \geq k-1$ holds. Since L is optimal $x > y > z$ holds. Now $L[b]$ is not x-clean since b is not contained in the route of $j > x$, and b is contained in the routes of y ($y < x$) and z ($z < x$). Then the clean level of $L[b]$ is increased to $x+1$, and b is the only bar in the active region of $L[b]$, thus b is the active bar of $L[b]$. Thus $L[b]$ is a child of L. Note that b is "downward visible" from $x \geq k$.

Type 2: b can be right swapped but b is not a turn bar.

Such a bar exists only in $L(k) \cap L(k+1) \cap \cdots \cap L(n)$, and those are downward visible bars from some x, $1 \leq x \leq n$.

If the right swap operation moves b to $R(j)$ where $j \geq k$ for some j, crossing the route of j, then the clean level of $L[b]$ is $j+1$ and b is the only

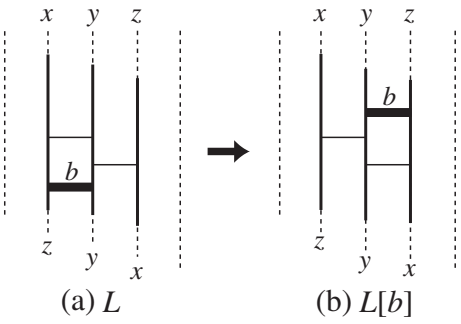

Figure 7: Illustration for Type 1.

bar in the new active region, so b is the active bar of $L[b]$. Thus $L[b]$ is a child of L.

If the right swap operation moves b to $R(k-1)$, crossing the route of $k-1$, then the clean level of $L[b]$ remains k, and b is appended to the active region. Assume the active bar of L has the right end on line s. If b in $L[b]$ has the right end on line $t \geq s-1$, then b is the active bar of $L[b]$, otherwise b is not. Now $L[b]$ is a child of L if and only if $t \geq s-1$.

Otherwise the right swap operation moves b to $R(x)$ where $x < k-1$ for some x, crossing the route of x. The clean level of $L[b]$ remains k, and b is not the active bar in $L[b]$. Thus $L[b]$ is not a child of L.

By the above analysis, we have the algorithm in Figure 8.

By maintaining (i) the clean level k (the clean level of $L[b]$ is always larger than or equal to the clean level of L) and (ii) the list of downward visible bars from x, for each $x = n, n-1, \ldots, k$ and (iii) the list of downward visible bars from $k-1$ each of which is the active bar in $L[b]$, we can enumerate all children of L in $O(1)$ time for each. We assume the bars appear in the list from left to right order in L. Also regarding L as a graph, we maintain its adjacency list representation with suitable data. We have the following lemma.

Lemma 6. *All children of L can be enumerated in $O(1)$ time for each.*

Proof. Let L be the current ladder lottery and k its clean level. Now we are going to compute each child $L[b]$ of L by applying a right swap operation to a bar b which is downward visible from the route of $x \geq k-1$.

Given (i)–(iii) of L, we can compute $L[b]$ and update (i)–(iii) for $L[b]$ in $O(1)$ time as follows.

Procedure find-all-children(L, k, a)
// L is the current ladder lottery, k is the clean level
// of L, and a is the active bar of L.
begin
01 Output L // Output the difference from the previous one.
02 **for** each $x \geq k$
03 **for** each downward visible bar b from x
04 **find-all-children**($L[b]$, $x+1$, b)
05 **for** each downward visible bar b from $k-1$ which can be the active bar in $L[b]$
06 **find-all-children**($L[b]$, k, b)
end

Algorithm find-all-ladder-lotteries($\pi = (p_1, p_2, \ldots, p_n)$)
begin
07 Compute R in S_π
08 **for** $x = 1$ **to** n
09 **if** the route of x has the turn bar b
10 **then find-all-children**($L[b]$, $x+1$, b)
end

Figure 8: Our algorithm.

Case 1: $L[b]$ is a Type 1 child of L.

Now b is the turn bar of the route of, say $y \geq k-1$ in L.

(i) The clean level of $L[b]$ is $x+1$. (ii) If b is downward visible from the route of some $z > x$ in $L[b]$, then we can find such z in $O(1)$ time, and also we can compute in $O(1)$ time the list of downward visible bars from z in $L[b]$ by appending b to the corresponding list in L. For each $w = x+1, x+2, \ldots, n$, except for the z above, the list of downward visible bars from w in $L[b]$ remains as in L. For each $w = 1, 2, \ldots, x-1$ we do not need the list for $L[b]$, since the clean level of $L[b]$ is $x+1$. (iii) The list of x in $L[b]$ is derived from the list of downward visible bars from x in L as follows. Replace the sublist of the bars up to b by at most two bars each of which is downward visible from x in $L[b]$ but not in L. Thus we can compute (iii) for $L[b]$ in $O(1)$ time. (Intuitively, each such bar is not downward visible in L since b hides it, but downward visible in $L[b]$ since b has moved to $R(x)$.)

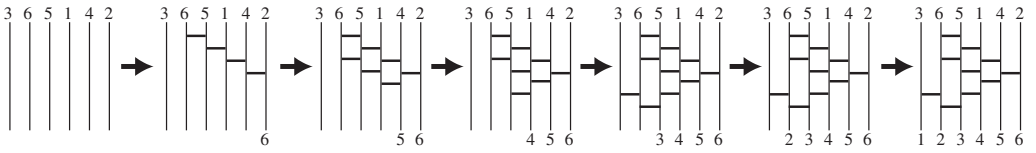

Figure 9: Computation of the root R in S_π, where $\pi = (3, 6, 5, 1, 4, 2)$.

Case 2: $L[b]$ is a Type 2 child of L.

Now b is not a turn bar.

If $x \geq k$ holds, then similar to Case 1, we can compute $L[b]$ and update (i)–(iii) in $O(1)$ time.

Now we assume that $x = k - 1$. (i) The clean level of $L[b]$ remains as k. (ii) Similar to Case 1. (iii) The list of $k - 1$ in $L[b]$ is derived from the list of downward visible bars from $k - 1$ in L as follows. Replace the bars up to b by at most two bars each of which is downward visible from $k - 1$ in $L[b]$ but not in L. Thus we can compute (iii) for $L[b]$ in $O(1)$ time.

Thus we can compute $L[b]$ from L in $O(1)$ time. By recording the modification of (i)–(iii) in a stack we can recover (i)–(iii) of L from $L[b]$ in $O(1)$ time. Thus we can compute all children of L in $O(1)$ time for each. □

From Lemma 6, we obtain the following theorem.

Theorem 7. *After generating and outputting the root ladder lottery R in S_π in $O(n^2)$ time, our enumeration algorithm runs in $O(|S_\pi|)$ time. The algorithm uses $O(n^2)$ space.*

Proof. We show that the root ladder lottery R in S_π can be generated in $O(n^2)$ time. We start with n vertical lines. Then we append the route of each $x = n, n - 1, \ldots, 1$. Each route goes left with some bars, turns, and goes right with some bars. When we append the route of i the part of route goes left is already completed, since at those bars the route of i crosses more larger numbers, so we only need to append the part goes right, consisting of monotone path. See Figure 9. Thus we can compute R in $O(n^2)$ time and space. Also the data structure of R can be computed in $O(n^2)$ time and space. □

By Theorem 7, our algorithm generates each ladder lottery in S_π in $O(1)$ time "on average." However it may have to return from the deep recursive

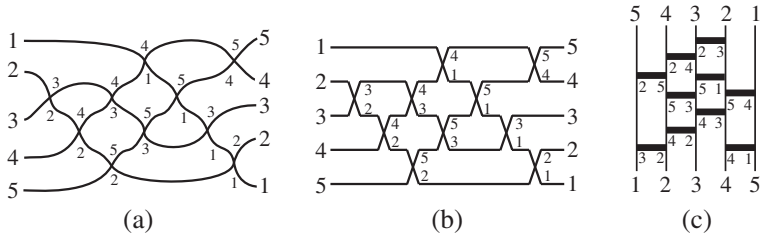

Figure 10: (a) An arrangement of 5 pseudolines, (b) its wiring diagram and (c) the corresponding optimal ladder lottery.

calls without outputting any ladder lottery in S_π after generating a ladder lottery corresponding to the rightmost leaf of a large subtree in a family tree. This takes much time. Therefore the next ladder lottery in S_π cannot be generated in $O(1)$ time in worst case.

This delay can be canceled by outputting the ladder lotteries in S_π in the "prepostorder" manner in which ladder lotteries in S_π are outputted in the preorder (and postorder) manner at the vertices of odd (and even, respectively) depth of a family tree. See [9] for further details of this method: in [9] the method was not explicitly named, and the name "prepostorder" is given by Knuth [6].

Now we have the following theorem.

Theorem 8. *After computing and outputting the root ladder lottery R in S_π in $O(n^2)$ time, the algorithm enumerates every ladder lottery in S_π in $O(1)$ time for each. The algorithm uses $O(n^2)$ space.*

5. Application to Pseudoline Arrangements

In this section we give a simple but efficient algorithm to enumerate all arrangements of n pseudolines by simplifying the algorithm in Section 4.

We give some definitions. A *pseudoline* is an x-monotone curve in the plane. An *arrangement* of pseudolines is a set of pseudolines in which every pair intersects exactly once. See Figure 10(a). An arrangement is *simple* if no three pseudolines share a common point. From now on, the term arrangement always denotes a simple arrangement of pseudolines.

A *wiring diagram*, introduced in [2], of an arrangement of n pseudolines is a network with n lines and $\binom{n}{2}$ intersections. See Figure 10(b). The left ends correspond to the reverse permutation $(n, n-1, \ldots, 1)$ in bottom to top

order. The right ends correspond to the identical permutation $(1, 2, \ldots, n)$ in bottom to top order. Each line i starts at the i-th left end from the top, then goes right, however at every intersection the line goes up or down to cross other line, then finally line i reaches the i-th right end from the bottom. Each such path corresponds to a pseudoline. Note that each path has exactly $n-1$ intersections.

The combinatorial structure of each pseudoline arrangement can be modeled as a wiring diagram, and each wiring diagram models the combinatorial structure of a set of pseudolines arrangements. Note that applying some perturbation to a pseudoline arrangement still results in the same corresponding wiring diagram. We say two pseudoline arrangements are *isomorphic* if there is a bijection between their faces of corresponding wiring diagrams preserving their neighbor relations.

By replacing intersections as bars a wiring diagram can be regarded as an optimal ladder lottery of a reverse permutation. For instance the wiring diagram in Figure 10(b) corresponds to the optimal ladder lottery in Figure 10(c). Then we can observe that there is a one-to-one correspondence between wiring diagrams and optimal ladder lotteries of a reverse permutation. Thus the combinatorial structure of a pseudoline arrangement uniquely corresponds to an optimal ladder lottery. In this section we consider to enumerate all optimal ladder lotteries of a reverse permutation.

Let L be a ladder lottery of the reverse permutation $(n, n-1, \ldots, 1)$. Assume L has clean level k. We can observe that the routes of numbers $n, n-1, \ldots, k-1$ form the following brick structure. The route of $x \geq k-1$ first goes left $n-x$ times, pass through the leftmost line, then go right $x-1$ times. If $x \geq k+1$ holds, we can observe that the route of $x-1$ go right along the route of x and immediately below the route of x (see Figure 11). Then only the turn bar b of the route of $x-1$ is downward visible from x and b can be always right swapped.

Therefore we need not to maintain the list of downward visible from x, for each $x \geq k+1$. Instead of the lists, we maintain only the list of the turn bars. By maintaining (i) the clean level k (ii) the list of the turn bars and (iii) the list of downward visible bars from k or $k-1$ (those are candidates to be right swapped, crossing the route of k or $k-1$), we can enumerate all children of L in $O(1)$ time for each.

We have the following theorem.

Theorem 9. *After $O(n^2)$ time preprocessing, we can enumerates all combi-*

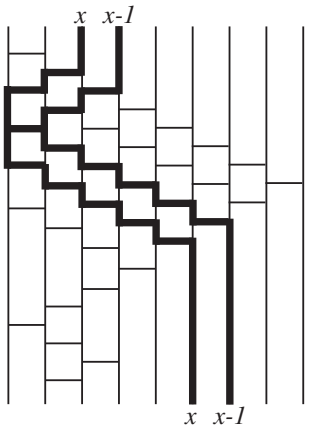

Figure 11: The two routes of x and $x-1$.

natorial structures of arrangements of n pseudolines in $O(1)$ *time for each.*

By implementing the algorithm we have computed the number of pseudoline arrangements for each $n \leq 11$, as shown in Table 1. The numbers for $n \leq 10$ match to the reports by Knuth [4] and Widom et al. [13] and the number for $n = 11$ is the first report. Our new result for $n = 11$ is added into A006245 in the database [10].

6. Conclusion

In this paper, we first showed that, for any two ladder lotteries L_1 and L_2 in S_π, there exists a sequence of operations consisting of zero or more left swap operations followed by zero or more right swap operations which transforms L_1 into L_2. We also gave an algorithm to enumerate all optimal ladder lotteries of a given permutation. Our algorithm uses $O(n^2)$ space and enumerate all ladder lotteries in S_π in $O(1)$ time for each in worst case. The algorithm can be applied to enumerate all combinatorial structures of pseudoline arrangements.

References

[1] D. Avis and K. Fukuda. Reverse search for enumeration. *Discrete Appl. Math.*, 65(1-3):21–46, 1996.

[2] J.E. Goodman. Proof of a conjecture of burr, grünbaum and sloane. *Discrete Math.*, 32:27–35, 1980.

[3] J.E. Humphreys. *Reflection groups and Coxeter groups*. Cambridge University Press, 1990.

[4] D.E. Knuth. *Axioms and hulls*. LNCS 606, Springer-Verlag, 1992.

[5] D.E. Knuth. *The art of computer programming*. 3, Addison-Wesley, 2nd edition, 1998.

[6] D.E. Knuth. *The art of computer programming*, volume 4, fascicle 4, generating all trees, history of combinatorial generation. Addison-Wesley, 2006.

[7] Z. Li and S. Nakano. Efficient generation of plane triangulations without repetitions. *Proc. The 28th International Colloquium on Automata, Languages and Programming, (ICALP 2001)*, LNCS 2076:433–443, 2001.

[8] S. Nakano. Efficient generation of triconnected plane triangulations. *Comput. Geom. Theory and Appl.*, 27(2):109–122, 2004.

[9] S. Nakano and T. Uno. Constant time generation of trees with specified diameter. *Proc. the 30th Workshop on Graph-Theoretic Concepts in Computer Science, (WG 2004)*, LNCS 3353:33–45, 2004.

[10] N.J.A. Sloane. *The On-Line Encyclopedia of Integer Sequences*. Published electronically at `http://www.research.att.com/~njas/sequences/`, 2009.

[11] J.R. Stembridge. On the fully commutative elements of coxeter groups. *Journal of Algebraic Combinatorics*, 5:353–385, 1996.

[12] K. Wagner. Bemerkungen zum vierfarbenproblem. *Jahresber. Deutsche Math.-Verein.*, 46:26–32, 1936.

[13] M. Widom, N. Destainville, R. Mosseri, and F. Bailly. Two-dimensional random tilings of large codimension. *Proc. the 6th International Conference on Quasicrystals*, 1998.

[14] K. Yamanaka and S. Nakano. Listing all plane graphs. *Journal of Graph Algorithms and Its Applications*, 13(1):5–18, 2009.