

On computing the total displacement number via weighted Motzkin paths

Andreas Bärtzchi^{*1}, Barbara Geissmann^{†1}, Daniel Graf^{‡1}, Tomas Hruz^{§1}, Paolo Penna^{¶1},
and Thomas Tschager^{||1}

¹ETH Zürich, Department of Computer Science

Abstract

Counting the number of permutations of a given total displacement is equivalent to counting weighted Motzkin paths of a given area (Guay-Paquet and Petersen [10]). The former combinatorial problem is still open. In this work we show that this connection allows to construct efficient algorithms for counting and for sampling such permutations. These algorithms provide a tool to better understand the original combinatorial problem. A by-product of our approach is a different way of counting based on certain “building sequences” for Motzkin paths, which may be of independent interest.

1 Introduction

Consider the set \mathcal{S}_n of all permutations over n elements $\{1, 2, \dots, n\}$. Diaconis and Graham [5] studied the *disarray* statistic of permutations, also called *total displacement* by Knuth [13, Problem 5.1.1.28], defined as follows. For any permutation π define its distance to the identity permutation as the sum of the displacements of all elements:

$$D(\pi) := \sum_{i=1}^n |i - \pi(i)| = 2 \sum_{\pi(i) > i} (\pi(i) - i).$$

Note that this distance is always *even*. The following natural question is still unresolved:

How many permutations at a given distance $2d$ from the identity permutation are there?

That is, one would like to know the following *total displacement* number:

$$D(n, d) := |\{\pi \in \mathcal{S}_n \mid D(\pi) = 2d\}|,$$

that is the number of permutations of total displacement equal to $2d$. So far, a closed formula for arbitrary n is only known for fixed d up to seven ($d \leq 7$) [10]. Entry A062869 [7] of the OEIS reports values of $D(n, d)$ for small n and d ($n \leq 30$).

*andreas.baertschi@inf.ethz.ch

†barbara.geissmann@inf.ethz.ch

‡daniel.graf@inf.ethz.ch

§tomas.hruz@inf.ethz.ch

¶paolo.penna@inf.ethz.ch

||thomas.tschager@inf.ethz.ch

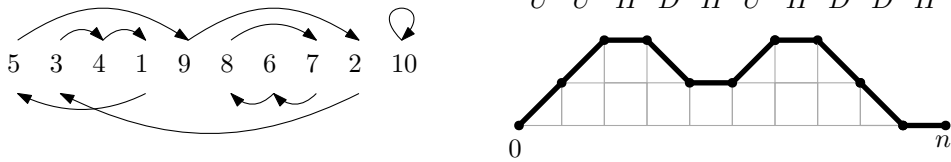


Figure 1: A permutation and its Motzkin path of width 10 and area 12.

Guay-Paquet and Petersen [10] made recently significant progress in this question by showing that these permutations are in correspondence to *Motzkin paths* whose *area* is exactly the distance d under consideration. Their result shows that, for any Motzkin path (see below) of area d , one can easily calculate the number of permutations that correspond to this specific path. Therefore the problem above translates into the problem of counting *weighted* Motzkin paths of a given area.

A Motzkin path consists of a sequence of U (Up-right), H (Horizontal-right), and D (Down-right) moves over the two-dimensional lattice starting at coordinate $(0, 0)$ and such that the path never goes below the $y = 0$ axis and ends on the $y = 0$ axis (see Figure 1 (right) for an example). For any such path, one can consider its *width* and its *area* defined as the number of moves and the area of the region between the $y = 0$ axis and the path. The permutations over n elements with total displacement $2d$ map into Motzkin paths of width n and area $A = d$.

For instance, the permutation in Figure 1 is mapped into a Motzkin path according to the following rule. The first element $\pi(1) = 5$ is mapped into a U because the element at position 1 goes to a higher position (right) and also the number coming into position 1 is higher than 1: $\pi(1) > 1 < \pi^{-1}(1)$. The fourth element is mapped into D because the opposite happens: $\pi(4) = 1 < 4 > 3 = \pi^{-1}(4)$. Finally, elements 3, 5, 7, 10 are mapped into H because neither of the previous cases apply.

Let h_i denote the maximum height of the path during move i (for U : after the move, for D : before the move, and anytime for H). Then the number $\omega(mz)$ of permutations that map to a certain Motzkin path mz is [10]

$$\omega(mz) = \prod_i^n \omega_i \text{ where } \omega_i = \begin{cases} h_i & \text{if } mz_i = U \text{ or } mz_i = D, \\ 2h_i + 1 & \text{if } mz_i = H. \end{cases} \quad (1)$$

We also refer to $\omega(mz)$ as the *weight* of mz . In the example in Figure 1 this gives $1 \cdot 2 \cdot 5 \cdot 2 \cdot 3 \cdot 2 \cdot 5 \cdot 2 \cdot 1 \cdot 1 = 1200$. Note how this formula separates over the moves of the Motzkin path. This independence is what we will exploit in this article.

Theorem 1 ([10]). *For any n and d , let $MZ(n, A)$ be the set of all Motzkin paths of width n and area $A = d$. Then it holds that*

$$D(n, d) = \sum_{mz \in MZ(n, A)} \omega(mz). \quad (2)$$

Corollary 1 (Appendix A). *Given a Motzkin path mz of length n , we can sample uniformly at random one of the $\omega(mz)$ many permutations mapping into mz in time $\mathcal{O}(n)$.*

Our contribution. In this work, we address counting and sampling of permutations from both a combinatorial and computational point of view. Specifically:

- On the computational side, we show that the total displacement number $D(n, d)$ can be computed efficiently, namely, in time $\mathcal{O}(n^4)$ and $\mathcal{O}(n^3)$ space.

- On the combinatorial side, we introduce sequences of certain *building blocks* which provide a different perspective on the problem structure. Moreover, this is a crucial part of a Markov chain sampling method which constitutes the third contribution of this paper.
- Finally, we consider the task of *sampling* permutations of a given total displacement with uniform distribution.

To compute the number of permutations efficiently, we look at the paths from left to right. Building on an operation introduced by Barucci et al. [1], we can provide an elegant dynamic programming formulation which achieves a running time of $\mathcal{O}(n^4)$ and needs space $\mathcal{O}(n^3)$. Consequently, we can compute the sequences A062869 [7] and A129181 [3] to much higher values of n and d than was possible before.

Considering the combinatorial aspects, we show that every Motzkin path comes from a sequence a describing its building blocks. We provide an explicit formula for the number $m(a)$ of paths that these building blocks can form. The weights in Equation (1) are preserved in the sense that the weight of a path depends only on its building sequence.

Since the exact formula seems to be currently out of reach, to achieve good estimates of $D(n, d)$ for very large n and d , we contribute sampling methods which can also be of independent interest. In particular, the dynamic programming algorithm provides a sampler with the same complexity as the algorithm itself. Further, we show that sampling sequences of building blocks with appropriate distribution automatically gives a sampler for the permutations. One application of the latter result is a Monte Carlo Markov chain (MCMC) method which gives an alternative approach to the dynamic programming. The computational experiments with the MCMC method show a promising convergence speed leading to a sampler with very high values of n and d . The experimental results support a hypothesis that the MCMC method is faster than the method based on dynamic programming and runs in $\mathcal{O}(n^3)$ time.

Related Work. Different metrics on permutations have been studied, for a survey see [4]. Sampling and counting of permutations of a fixed distance was studied for several metrics [12] but not for total displacement.

The number of Motzkin paths under various conditions were also studied in a more general frame of enumeration of lattice paths [11, 8]. Motzkin numbers play a role in many combinatorial problems as is illustrated for example in [6]. The total area under a set of generalized Motzkin paths, where the horizontal segments have a constant length k ($k \geq 0$) have been studied in [16] and [15]. Moreover, the author in [17] studies the moments of generalized Motzkin paths where the first moment describes the area under a Motzkin path. Heinz [3] describes a different algorithm for enumerating unweighted Motzkin paths with a given area, cf. Remark 4 in Section 2.1.

The use of Markov chains for sampling and counting combinatorial objects is a very active research area (see e.g. the book [2]), and some works exploit the connection between combinatorial structures and paths of a certain type to accomplish this task (see e.g. [9]).

Paper Organization. Section 2 describes the dynamic programming algorithm. Section 3 describes how weighted Motzkin paths can be counted via building block sequences. Section 4 provides a Markov chain sampling algorithm as well as its experimental evaluation.

2 Weighted Motzkin Paths using Dynamic Programming

Recall that we denote by $D(n, d)$ the number of permutations on n elements with total displacement $2d$ (OEIS A062869 [7]). Let $M(n, A)$ denote the number of Motzkin paths of width n and area A (OEIS A129181 [3]).

2.1 Dynamic Program for Counting Weighted Motzkin Paths

Theorem 2. *Computing $M(n, A)$ and $D(n, d)$ can be done in time $\mathcal{O}(n^4)$ and space $\mathcal{O}(n^3)$.*

Proof. The key ingredient is a construction by Barucci et al. [1] that produces every possible Motzkin path through a unique sequence of insertion steps.

Let us look at the last fall of a given Motzkin path, i.e., its suffix of Down-right moves. At one of the positions before or after any of these fall moves, we insert a new *peak* (a U and a D) or we insert a new *flat* (an H). Repeatedly inserting peaks and flats this way along the last fall will create our path. See Figure 2 for an example. This construction is complete and unique [1], meaning that every Motzkin path can be created through a unique sequence of such insertions.

This allows us to derive a dynamic programming formulation for counting $M(n, A)$. We add the last fall length l to our state and write it as $M(n, A, l)$. So how can we recursively express $M(n, A, l)$? We undo the last insertion step. If we inserted a flat last, then we were at $M(n-1, A-l, l')$ before the insertion, for some $l' \geq l$, because the last fall was at least as long before the insert. When inserting a peak, we might increase the last fall length by one, but not more. So $M(n-2, A-(2l-1), l')$ for all $l' \geq l-1$ are also possible predecessor states. Together with the base case $M(0, 0, 0) = 1$ this gives the recurrence

$$M(n, A, l) = \sum_{l' \geq l}^{n/2} M(n-1, A-l, l') + \sum_{l' \geq l-1}^{n/2} M(n-2, A-(2l-1), l'), \quad (3)$$

which allows for $\mathcal{O}(n^4)$ many states as $A \leq n^2$ and $l \leq n$. Hence we immediately get an $\mathcal{O}(n^5)$ time algorithm with $\mathcal{O}(n^4)$ space. We can shave off one factor of n in both time and space as follows: We first note, that we can compute the two sums in constant time if we precompute the prefix sums over the last variable l' . Let us denote these prefix sums as $SM(n, A, l) = \sum_{l'=0}^l M(n, A, l') = SM(n, A, l-1) + M(n, A, l)$. This allows us to compute every value of $M(n, A, l)$ in amortized constant time, so in time $\mathcal{O}(n^4)$ overall. Finally, our recurrence only relies on the last two values of n , so when computing $M(n, \cdot, \cdot)$ only the $\mathcal{O}(n^3)$ many values for $M(n-1, \cdot, \cdot)$ and $M(n-2, \cdot, \cdot)$

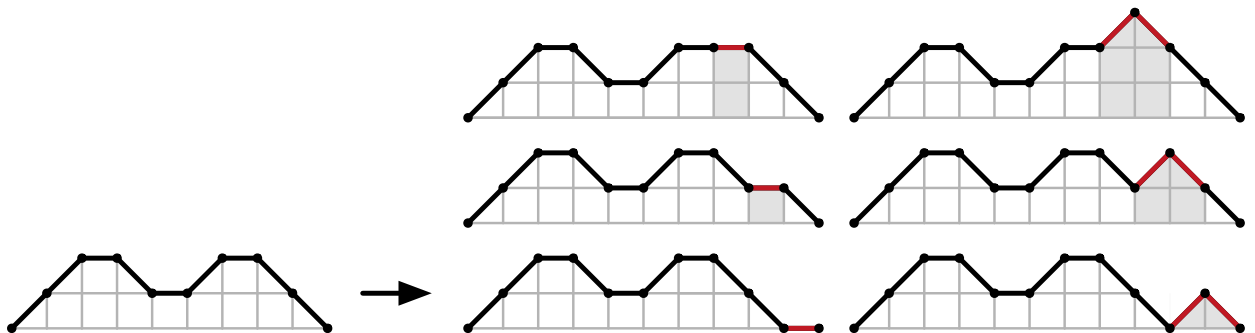


Figure 2: All six possible flat- and peak-extensions of the last fall of length 2.

need to be stored. The values $M(n, A)$ are then simply the marginals of $M(n, A, l)$ over all last fall lengths l .

We can extend this recurrence to the weighted case which by Corollary 1 gives rise to the total displacement count: We distribute the factors of the weight $\omega(mz)$ (Equation (1)) over the steps of the dynamic program. As l denotes the height of the last flat or peak that we add, we have factors $2l + 1$ or l^2 :

$$D(n, d, l) = (2l + 1) \sum_{l' \geq l}^{n/2} D(n - 1, d - l, l') + l^2 \sum_{l' \geq l-1}^{n/2} D(n - 2, d - (2l - 1), l'). \quad \square$$

Remark 3. *The bounds in Theorem 2 assumed that basic operations have unit-cost. The numbers involved can be exponential in n however. We can easily bound $M(n, A) \leq 3^n$ and $D(n, d) \leq n!$ showing that their bit-representations are at most of length $\mathcal{O}(n \log n)$. Our dynamic programs only use multiplication with small numbers of size $\mathcal{O}(\log n)$ and addition. So one can consider a refined analysis by multiplying both the time and space bounds of Theorem 2 by $\Theta(n \text{ polylog } n)$. Finally, as suggested by an anonymous reviewer, the space could be further improved by counting modulo small primes and using the Chinese Remainder Theorem.*

Remark 4. *For computing $M(n, A)$, the OEIS contains a dynamic program by Heinz [3]. It is stated as a Maple code snippet without any further comment or reference. It uses a different state and might have the same time complexity as ours. We believe that our extension to the weights of $D(n, d)$ can also be applied.*

2.2 Sampling from the Dynamic Program

Theorem 5. *After running the dynamic program from Theorem 2, we can sample (weighted) Motzkin paths in time $\mathcal{O}(n)$.*

Proof. Given access to a source of randomness and the filled table for M , we can randomly retrace the steps through the dynamic programming states to sample a Motzkin path from right to left. For the weighted paths according to $D(n, d)$ all the steps will be exactly the same. We first sample the last fall length by picking a random number $x \in_{\text{u.a.r.}} \{0, \dots, M(n, A) - 1\}$ and then finding the smallest l such that its prefix sum $SM(n, A, l)$ is larger than x . We continue with $x - SM(n, A, l - 1)$, the offset within the class of paths with last fall length l . For each step, we first decide whether we are in the flat-case or in the peak-case of the recurrence by comparing x to the left summand of (3). We then know whether the move before the last fall was an H or a U . We increment l' until we find the last fall length of the previous state. We adapt x and recurse until we end at $M(0, 0, 0)$ with $x = 0$. Note that the search for the initial l takes linear time. After that, every time we compare x to a value of M , we fix at least one move of the sampled Motzkin path, so sampling takes $\mathcal{O}(n)$ time overall. \square

Remark 6. *This sampling procedure requires the full table of the dynamic program to be stored. Hence the memory optimization from $\mathcal{O}(n^4)$ to $\mathcal{O}(n^3)$ in Theorem 2 can not be used simultaneously.*

Remark 7. *A C++ implementation of our counting and sampling approaches by Theorems 2 and 5 is available at <http://people.inf.ethz.ch/grafdan/motzkin/>. With our code, we can quickly compute for n up to 100 (and all d) the integer sequences A062869 [7] and A129181 [3] which were only known up to $n \leq 30$ and $n \leq 50$ before.*

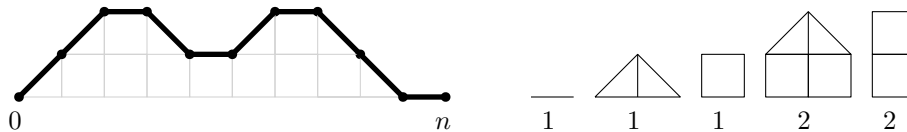


Figure 3: The Motzkin path on the left can be obtained from its building blocks.

3 Combinatorial Structure of Motzkin Paths

In this section, we look at the combinatorial structure of Motzkin paths: There is a natural decomposition of any Motzkin path into “building blocks”, already hinted at in the last section. For each height i of the Motzkin path we count the number of flats f_i and peaks p_i .

Definition 1 (building sequence). *For given positive integers n and A , a finite sequence of non-negative integers $a = (f_0, p_1, f_1, p_2, \dots, p_h, f_h)$ is a building sequence if all p -entries are non-zero, $p_1, p_2, \dots, p_h > 0$, and the following two conditions hold:*

$$(f_0 + f_1 + \dots + f_h) + 2(p_1 + p_2 + \dots + p_h) = n, \quad (4)$$

$$(0f_0 + 1f_1 + \dots + hf_h) + (1p_1 + 3p_2 + \dots + (2h-1)p_h) = A. \quad (5)$$

The set of all building sequences satisfying (4)-(5) is denoted as $S(n, A)$.

Such a sequence has a natural interpretation as a set of “building blocks” that generate a number of Motzkin paths of width n and area A (see Figure 3): We have f_i flats and p_i peaks of height i which can be split into pieces of width 1 and then rearranged into a Motzkin path.

Proposition 1. *For any Motzkin path mz of width n and area A there exists a unique building sequence $a^{(mz)} \in S(n, A)$ such that mz can be obtained by splitting and rearranging the blocks of this sequence.*

Theorem 1 gives a surjective mapping from permutations into Motzkin paths. It is easy to see that the number of permutations $\omega(mz)$ mapping into the same path mz , given by Equation (1), is uniquely determined by the building block sequence $a = a^{(mz)}$, since we have

$$\text{perm}(a) := \prod_{f_i} (2i+1)^{f_i} \prod_{p_i} i^{2p_i} = \omega(mz). \quad (6)$$

Hence $\omega(mz)$ is independent of the actual Motzkin path and only depends on its combinatorial structure. This raises the question of whether also the number of Motzkin paths which share a common building sequence a is solely determined by a . We answer this in the positive, deriving a formula for this number, denoted by $m(a)$. We proceed in a top-down fashion by looking at the number of peaks and flats in the highest level and how these can be rearranged. Once a level is fixed, we proceed recursively by arranging the blocks one level below.

Theorem 8. *For any building sequence $a = (f_0, p_1, f_1, \dots, p_h, f_h) \in S(n, A)$, the number of Motzkin paths of width n and area A that can be constructed out of the building sequence a is exactly*

$$m(a) = \binom{f_h + p_h - 1}{p_h - 1} \binom{p_h + f_{h-1}}{f_{h-1}} \binom{p_h + f_{h-1} + p_{h-1} - 1}{p_{h-1} - 1} \binom{p_{h-1} + f_{h-2}}{f_{h-2}} \dots \\ \dots \binom{p_3 + f_2 + p_2 - 1}{p_2 - 1} \binom{p_2 + f_1}{f_1} \binom{p_2 + f_1 + p_1 - 1}{p_1 - 1} \binom{p_1 + f_0}{f_0}. \quad (7)$$

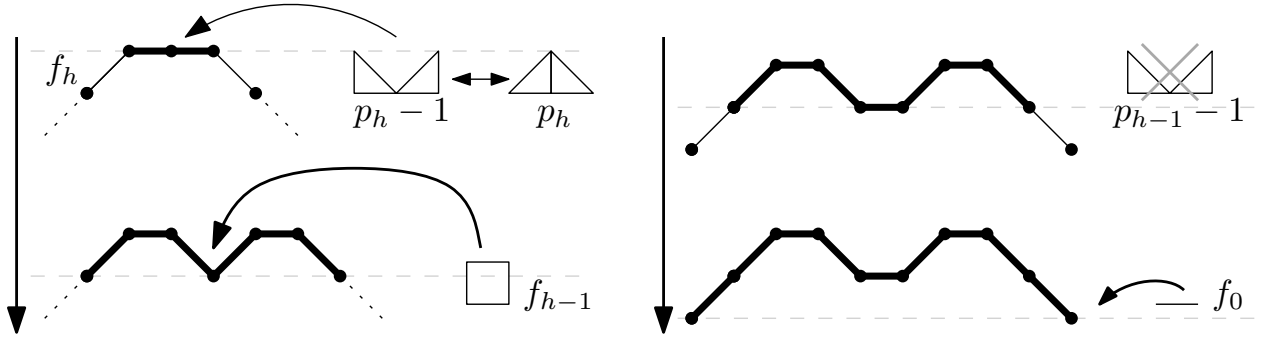


Figure 4: The top down construction of paths from the given sequence $(1, 1, 1, 2, 2)$. Note that $p_1 - 1 = 0$, and thus no DU valley is inserted at height 1.

Proof. We start with the highest flats of the sequence a . There are f_h of those flats. Two (or more) such flats can either lie directly next to each other, or they might be separated by a *Down-right* move followed at some point by an *Up-right* move. We call this setting a *DU* valley; we get such valleys by splitting peaks of height h and reassembling them the other way round, see Figure 4. A feasible Motzkin path has to have a *U* slope at the very left and a *D* slope at the very right of all height h pieces. The remaining $p_h - 1$ *DU* valleys can be freely placed around the f_h flats, that is we choose their places from $f_h + p_h - 1$ available positions. The number of ways to do this is

$$\binom{f_h + p_h - 1}{p_h - 1}. \quad (8)$$

Now we continue on the second highest level $h - 1$. Naturally, the number of times that our Motzkin path rises above level $h - 1$ is exactly the number p_h of peaks of height h . We can distribute our f_{h-1} flats of height $h - 1$ around those peaks, i.e. pick from $p_h + f_{h-1}$ many positions, hence we can choose from

$$\binom{p_h + f_{h-1}}{f_{h-1}} \quad (9)$$

many possibilities. After placing the flats, we will have to place new valleys down to the next lower level around the existing p_h peaks and f_{h-1} flats. As before, the leftmost up and down slopes are fixed, hence the number of ways to distribute $p_{h-1} - 1$ valleys is given by the third factor in Equation (7). Since the choices in different levels are independent, we can iterate this reasoning until we include flats of height 0. \square

We conclude with a corollary of Theorems 1 and 8:

Corollary 2. *There exists a surjective mapping from permutations over n elements into building sequences satisfying the following condition: For any building sequence $a \in S(n, A)$, the number of permutations π which are at distance $D(\pi) = 2d = 2A$ from the identity permutation and that are mapped into this building sequence a is precisely*

$$P(a) := m(a) \cdot \text{perm}(a), \quad (10)$$

where $m(a)$ is given by Equation (7) and $\text{perm}(a)$ by Equation (6). Therefore the total number of permutations at distance $2d = 2A$ from the identity permutation satisfies

$$D(n, d) \stackrel{(2)}{=} \sum_{mz \in MZ(n, A)} \omega(mz) = \sum_{a \in S(n, A)} P(a). \quad (11)$$

Example 1. *The building blocks in Figure 3 yield $\binom{3}{1}\binom{3}{1}\binom{3}{0}\binom{2}{1} = 18$ Motzkin paths, and each path corresponds to 1200 permutations. So, there are $1200 \cdot 18 = 21\,600$ permutations mapping into the building sequence $a = (1, 1, 1, 2, 2)$.*

Remark 9. *Theorem 8 and Corollary 2 allow for a dynamic program for counting and sampling weighted Motzkin paths, similar to Sections 2.1 and 2.2. Additionally, we can easily sample paths with a fixed number of highest peaks and flats, at the cost of an additional $O(n^3)$ -factor in the running time, see Appendix C.*

4 Sampling Weighted Motzkin Paths by Length and Area

In this section, we consider the task of selecting (sampling) permutations with uniform distribution over all permutations of a given total displacement. By Corollary 1 it is enough to sample Motzkin paths with the proper weights. We have already seen in Section 2.2 that we can sample such weighted Motzkin paths using dynamic programming at the cost of large memory consumption.

We will show in Section 4.2 an approach to sample weighted Motzkin paths based on the building sequences introduced in Section 3 that requires only $\mathcal{O}(n)$ memory. In general, observe that sampling permutations can be accomplished efficiently if we can sample building sequences with a probability proportional to $P(a) = m(a) \cdot \text{perm}(a)$ in polynomial time:

Theorem 10. *Every polynomial-time algorithm that samples sequences in $S(n, A)$ with probability $\pi(a) \propto P(a)$ can be turned into a polynomial-time algorithm for sampling permutations uniformly at random among the permutations over n elements and of total displacement $2d = 2A$.*

Proof. Given a sequence $a \in S(n, A)$, the sampler maps this sequence into a random Motzkin path, and then into a random permutation as follows:

1. Pick a Motzkin path mz uniformly at random among those that can be created with a , that is, with probability $\frac{1}{m(a)}$.
2. Pick a permutation u.a.r. among those that map into the Motzkin path mz , that is, with probability $\frac{1}{\text{perm}(a)}$.

Step 1 (sequences to Motzkin paths). The top-down construction used to prove Theorem 8 suggests also how to sample one of the $m(a)$ Motzkin paths for a given sequence a with uniform distribution. Namely, we pick the positions of the DU valleys at height h uniformly at random (Equation (8)), then we pick the positions of the f_{h-1} flats uniformly at random (Equation (9)), and repeat this to the lower level exactly as described in the top-down construction. Since a particular path corresponds to exactly one choice in each of these steps, by Equation (7) its probability is precisely $1/m(a)$.

Step 2 (paths to permutations). This is shown in Corollary 1 above. □

4.1 Preliminary definitions on Markov chains

In this section, we introduce some of the definitions on Markov chains used throughout this work (see e.g. [14]). A Markov chain over a finite state space S is specified by a transition matrix P , where $P(a, a')$ is the probability of moving from state a to state a' in one step. The t^{th} power of the transition matrix gives the probability of moving from one state to another state in t steps. The chain studied in this work is ergodic (see below for a proof), meaning that it has a unique *stationary distribution* π , that is, $\lim_{t \rightarrow \infty} P^t(a, a') = \pi(a')$ for any two states a and a' .

Reversible chains. We shall use the definition of a *reversible* Markov chain, also called *detailed balanced condition*: If the transition matrix P admits a vector π such that $\pi(a)P(a, a') = \pi(a')P(a', a)$ for all a and a' , then π is the *stationary distribution* of the chain with transitions P .

Total Variation Distance and Mixing Time. The *total variation* distance of two distributions μ and π is

$$d_{TV}(\mu, \pi) := \frac{1}{2} \cdot \sum_{a \in S} |\mu(a) - \pi(a)|.$$

The *mixing time* of an ergodic Markov chain with transition matrix P is defined as

$$t_{mix}(\epsilon) = \min \{t : d_{TV}(P^t(a_0, \cdot), \pi) \leq \epsilon \text{ for all } a_0 \in S\}.$$

It is common to also define the quantity $t_{mix} := t_{mix}(1/4)$, which is justified by the fact that, for any ϵ , $t_{mix}(\epsilon) \leq \lceil \log_2 1/\epsilon \rceil \cdot t_{mix}$. In our experiments, we shall evaluate the total variation distance for $\epsilon = 0.05$ to get better estimate.

4.2 A Markov Chain Sampler

Suppose we have a set of k possible local changes transforming any sequence a into another sequence a' such that all sequences can be obtained by applying a certain number of such operations. Then the following standard Metropolis chain samples sequences with the desired distribution:

1. With probability $\frac{1}{2}$ do nothing. Otherwise,
2. Select one of the k local operations u.a.r. If this operation cannot be applied to the current sequence a (the new sequence is unfeasible) do nothing; Otherwise, let a' be the sequence obtained from a by applying this operation;
3. Accept the operation transforming a to a' with probability

$$A(a, a') := \min \left\{ 1, \frac{P(a')}{P(a)} \right\} = \min \left\{ 1, \frac{m(a')}{m(a)} \cdot \frac{perm(a')}{perm(a)} \right\}, \quad (12)$$

and do nothing with remaining probability $1 - A(a, a')$.

Local operations over the sequences We define our Metropolis chain \mathcal{M}_{blocks} through four types of operations: Peak to Flat (PF), Flat to Valley (FV), Flat to Flat (FF), and Peak into Valley (PV). We formally define them as:

$$PF(i, j) := \begin{cases} p_i & \leftarrow p_i - 1 \\ f_{i-1} & \leftarrow f_{i-1} + 2 \\ f_j & \leftarrow f_j - 1 \\ f_{j+1} & \leftarrow f_{j+1} + 1 \end{cases}, \quad FV(i, j) := \begin{cases} f_i & \leftarrow f_i - 2 \\ p_i & \leftarrow p_i + 1 \\ f_j & \leftarrow f_j - 1 \\ f_{j+1} & \leftarrow f_{j+1} + 1 \end{cases}$$

$$FF(i, j) := \begin{cases} f_i & \leftarrow f_i - 1 \\ f_{i+1} & \leftarrow f_{i+1} + 1 \\ f_j & \leftarrow f_j - 1 \\ f_{j-1} & \leftarrow f_{j-1} + 1 \end{cases}, \quad PV(i, j) := \begin{cases} p_i & \leftarrow p_i - 1 \\ f_{i-1} & \leftarrow f_{i-1} + 2 \\ p_j & \leftarrow p_j - 1 \\ f_j & \leftarrow f_j + 2 \end{cases}$$

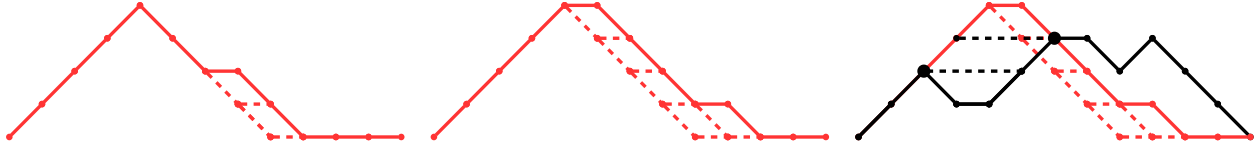


Figure 5: (left, middle) The canonic path for $n = 12$ and $A = 18$, resp. $A = 21$. (right) Building a canonic path (red) from a given path (black) with same area.

Note that each type of operation applies to two indices i and j , and we also implicitly consider the reversed operations which “undo” the changes. We now explain step 2 of the chain \mathcal{M}_{blocks} in more detail: The Markov chain \mathcal{M}_{blocks} picks two indices i and j at random, then picks one of the four operations above, and decides with probability $1/2$ whether to choose the operation or its reversed version. As for step 3, computing the transitional probability $A(a, a')$ can be done in constant time as only a few of the factors in Equations (6) and (7) change.

Theorem 11. *The Markov chain \mathcal{M}_{blocks} defined above is ergodic and its unique stationary distribution satisfies $\pi(a) \propto P(a)$ for every $a \in S(n, A)$.*

Proof. The proof consists of two steps. First, we have to show that the chain is ergodic, that is, it is aperiodic and connected (see e.g. [14]). Then we use the standard detail balance condition to obtain the stationary distribution.

4.2.1 Connectivity of \mathcal{M}_{blocks} .

To prove that the chain \mathcal{M}_{blocks} is connected (from every *building sequence* a we can reach every other building sequence a' in a maximum of $\mathcal{O}(A)$ operations) we argue in two steps. Intuitively, we show that we can transform any two *paths* into each other by some operations depicted in Figure 7. Then it can be seen that every operation in Figure 7 corresponds to a sequence of operations in the Markov chain \mathcal{M}_{blocks} , given in Figure 6. Formally:

Every path of width n and area $A \leq \lfloor n^2/4 \rfloor$ can be turned into any other path of the same area and width by using the operations in Figure 7. To prove this we consider a *canonic path* for a given width n and area A . The canonic path is the uniquely defined path $mz \in MZ(n, A)$ for which the following holds: For every i , after i steps (i.e. between $x = 0$ and $x = i$) mz has maximum area among all paths in $MZ(n, A)$. The possible forms of the canonic path are shown in Figure 5. Any given path with width n and area A can be transformed into the canonic path of the same area using the steps from Figure 7. We overlay the given path with the canonic path and proceed in steps to the right as is schematically shown in Figure 5 with the black path being the given path and the red path being the canonic path. There are three possibilities. Either the paths coincide, in which case we proceed to the right, or the given path differs proceeding with a D move or with an H move. In both cases the given path must intersect the canonic path on the falling part because otherwise the area cannot be the same. Now we use the operations in Figure 7 in horizontal sweeps from left to right to fill-in the missing area of the canonic path. At the end both paths must coincide because the areas are the same.

Each of these operations can be simulated by some operations on the sequences in Figure 6. This can be seen immediately because the four cases in Figure 7 correspond directly to one or two operations in Figure 6.

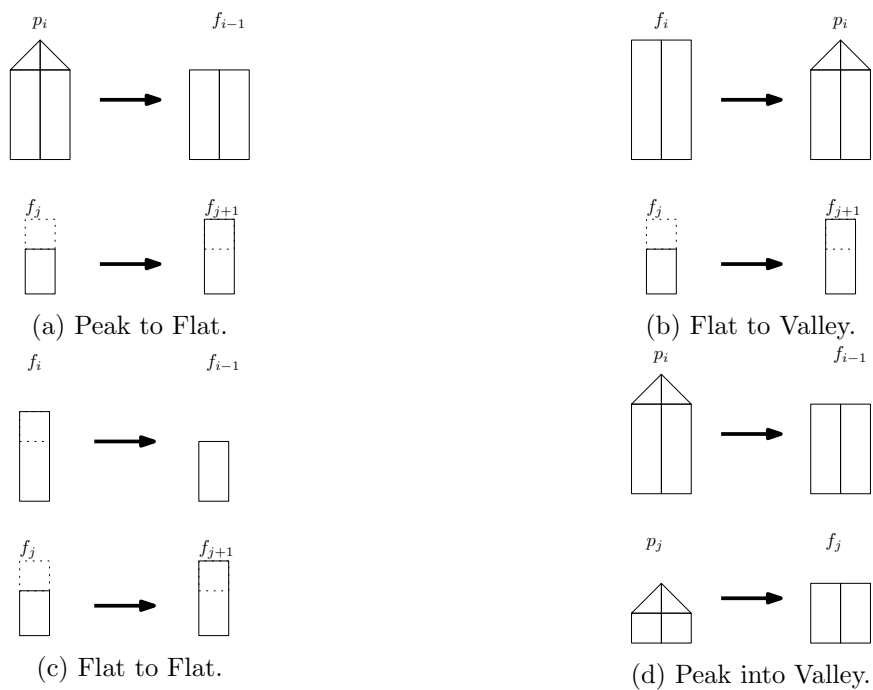


Figure 6: The basic operations over the sequences.

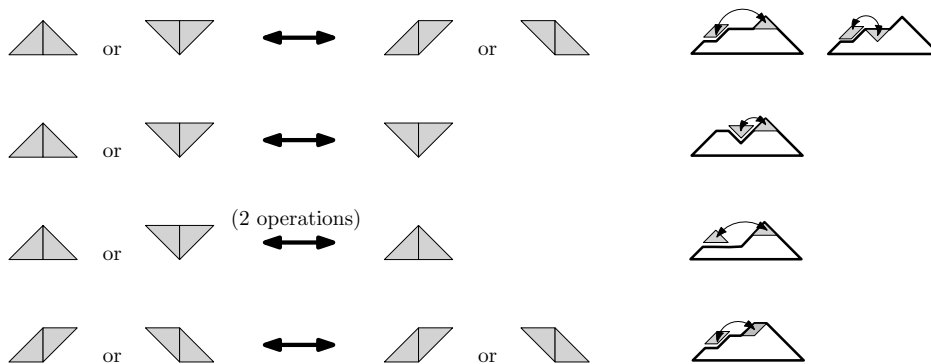


Figure 7: Changing *the path*. The shape on one side is transformed to the shape on the other side. In some cases we need two of the operations defined in Figure 6.

4.2.2 Stationary Distribution of \mathcal{M}_{blocks} .

It is well-known that the Metropolis chains with acceptance probability $A(a, a') = \min \left\{ 1, \frac{\pi(a')}{\pi(a)} \right\}$ have stationary distribution π since the detailed balance condition is obviously satisfied: given that the number of operations is k , we have

$$P(a, a') = \frac{A(a, a')}{2k} \quad \text{and} \quad P(a', a) = \frac{A(a', a)}{2k}$$

and the definition of $A(\cdot, \cdot)$ yields the detailed balance condition, that is, $\pi(a)P(a, a') = \pi(a')P(a', a)$. \square

4.2.3 Experimental Evaluation of \mathcal{M}_{blocks}

We are interested in the required number of steps until the distribution of \mathcal{M}_{blocks} is sufficiently close to its stationary distribution. We measure the distance between two distributions by the *total variation distance*. The *mixing time* of a Markov chain is the smallest time t such that the total variation distance between the stationary distribution and the distribution after t steps, starting from any state, is smaller than some small $\epsilon > 0$.

We study the mixing time of \mathcal{M}_{blocks} for a given area A and a given width n by running the following experiment. We estimate the distribution after a given number of steps by repeatedly running \mathcal{M}_{blocks} with an initial state a_0 defined as follows: The building block sequence consists of one peak of height h for every $h \leq \lfloor \sqrt{A} \rfloor$ and the remaining area and width is filled greedily with flats of maximal possible height. The total variation distance of the distribution of \mathcal{M}_{blocks} after some number of steps t from its stationary distribution π is

$$d_{TV}(P^t(a_0, \cdot), \pi) = \frac{1}{2} \cdot \sum_{a \in S(n, A)} |P^t(a_0, a) - \pi(a)|.$$

We estimate the mixing time for a given area A and a given width w by computing the total variation distance for increasing t until the total variation distance is below 0.05.

Figure 8 (left) illustrates the mixing time for width 8 and every area A with more than one possible building block sequence. The maximal mixing time (400 steps) is necessary for area 9. In fact, for every width smaller than 13, the mixing time is maximal for area $A_n^* = ((n-2)/2)^2$ if n is even and $A_n^* = ((n-1)/2)^2$ otherwise. This is due to our choice of the initial state of \mathcal{M}_{blocks} . We estimate the maximal mixing time for widths larger than 12 by computing the mixing time for A_n^* *only*, as the number of repeats necessary to estimate the distribution of \mathcal{M}_{blocks} after t steps depends on the number of possible building block sequences, which grows exponentially depending on n . Figure 8 (right) shows the maximal mixing time up to width 40. The plot suggests that the number of steps necessary to approximate the stationary distribution does not grow exponentially depending on the width n , the algorithm is probably faster than the sampler based on dynamic programming and the results suggest that the MCMC sampler achieves the mixing time $\mathcal{O}(n^3)$.

Conjecture 12. \mathcal{M}_{blocks} mixes in time $\mathcal{O}(n^3)$.

Remark 13. The implementation of \mathcal{M}_{blocks} is available at <http://people.inf.ethz.ch/grafdan/motzkin/>.

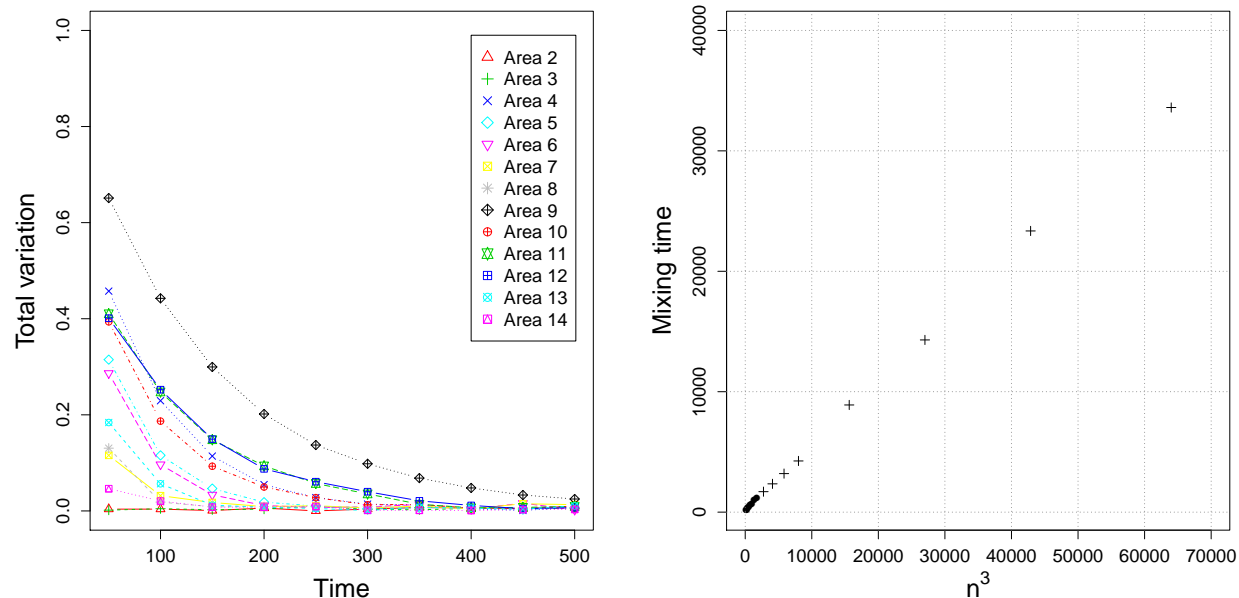


Figure 8: (left) Total variation for $n = 8$ and all $A \leq (n/2)^2$ with $|S(8, A)| > 1$. (right) Maximal mixing time for given widths (\bullet), mixing time for areas A_n^* ($+$).

References

- [1] Elena Barucci, Alberto Del Lungo, Elisa Pergola, and Renzo Pinzani. A construction for enumerating k -coloured motzkin paths. In *Computing and Combinatorics*, pages 254–263. Springer, 1995.
- [2] Russ Bubley. *Randomized Algorithms: Approximation, Generation and Counting*. Springer, 2011.
- [3] Emeric Deutsch and Alois P Heinz. A129181 motzkin paths by area, Online Encyclopedia of Integer Sequences. <http://oeis.org/A129181>, June 2012.
- [4] Michael Deza and Tayuan Huang. Metrics on permutations, a survey. *Journal of Combinatorics, Information and System Sciences*, 1998.
- [5] Persi Diaconis and Ronald L Graham. Spearman’s footrule as a measure of disarray. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 262–268, 1977.
- [6] Robert Donaghey and Louis W Shapiro. Motzkin numbers. *Journal of Combinatorial Theory, Series A*, 23(3):291–301, 1977.
- [7] Olivier Gérard, Mathieu Guay-Paquet, and Alois P Heinz. A062869 permutation with fixed total displacement, Online Encyclopedia of Integer Sequences. <https://oeis.org/A062869>, May 2014.
- [8] Ian P Goulden and David M Jackson. *Combinatorial enumeration*. Dover Publications, 2004.
- [9] Sam Greenberg, Amanda Pascoe, and Dana Randall. Sampling biased lattice configurations using exponential metrics. In *20th ACM-SIAM Symposium on Discrete Algorithms SODA ’09*, pages 76–85, 2009.

<i>width</i>	<i>area</i>	<i>runs</i>	<i>mixing time</i>
4 to 12	all relevant areas	10 000	200 to 1 200
14	36	100 000	1 700
16	49	100 000	2 350
18	64	100 000	3 200
20	81	100 000	4 250
25	144	100 000	8 900
30	196	100 000	14 300
35	289	400 000	23 350
40	361	500 000	33 600

Table 1: Experimental setup: The distribution after t steps was estimated using 10 000 to 500 000 runs of \mathcal{M}_{blocks} . The maximal mixing time was estimated by computing the mixing time for A_n^* for widths larger than 12.

- [10] Mathieu Guay-Paquet and Kyle Petersen. The generating function for total displacement. *The Electronic Journal of Combinatorics*, 21(3):P3–37, 2014.
- [11] Katherine Humphreys. A history and a survey of lattice path enumeration. *Journal of statistical planning and inference*, 140(8):2237–2254, 2010.
- [12] Ekhine Irurozki. *Sampling and learning distance-based probability models for permutation spaces*. PhD thesis, University of the Basque Country, Donostia - San Sebastián, July 2014.
- [13] Donald E. Knuth. The art of computer programming. *Sorting and searching*, 3:426–458, 1999.
- [14] David Asher Levin, Yuval Peres, and Elizabeth Lee Wilmer. *Markov chains and mixing times*. American Mathematical Soc., 2009.
- [15] Donatella Merlini. Generating functions for the area below some lattice paths. In *Discrete Random Walks, DRW'03*, pages 217–228, 2003.
- [16] E Pergola, R Pinzani, S Rinaldi, and RA Sulanke. A bijective approach to the area of generalized motzkin paths. *Advances in Applied Mathematics*, 28(3):580–591, 2002.
- [17] Robert A Sulanke. Moments of generalized motzkin paths. *Journal of Integer Sequences*, 3(00.1):1–14, 2000.

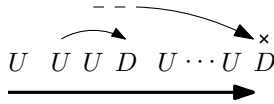
Appendix

A Postponed Proofs

Corollary 1 (Appendix A). *Given a Motzkin path mz of length n , we can sample uniformly at random one of the $\omega(mz)$ many permutations mapping into mz in time $\mathcal{O}(n)$.*

Proof. Given a Motzkin path mz , we can sample a permutation u.a.r. among all permutations that map into mz . For this, consider mz as a (feasible) sequence of letters U , D and H (denoting diagonally *Up-right* moves, diagonally *Down-right* moves and *Horizontal-right* moves). Then do the following:

1. Scan the sequence from left to right. When a new D is found, match it with any of the U on the left that are not yet matched to any D (choose such a U u.a.r.). This step constructs left-to-right edges from U to D :



2. Scan from right to left the sequence, matching a newly encountered U with any D on the right not yet matched in this step (the matchings of the previous ‘‘left-to-right’’ step do not count). This step is the symmetric of the previous one and it constructs right-to-left edges from D to U .
3. For any H we choose a ‘‘fixed point’’, a ‘‘left-to-right’’ or a ‘‘right-to-left’’ edge, meaning the following: The number of left-to-right edges crossing this H – corresponding to a flat of height i – is equal to the number i of right-to-left edges also crossing this H (this property is due to the ‘‘balanced’’ matchings of U and D). So there are $2i + 1$ options for H , where the $2i$ options correspond to breaking one of the i left-to-right edges, or one of the i right-to-left edges. The last option is to let the H map to a trivial cycle of the permutation. We choose one of these $2i + 1$ options u.a.r.

□

B Experiments

We estimate the distribution after a given number of steps by repeatedly running \mathcal{M}_{blocks} for an appropriate number of steps depending on the width (cf. Table 1). To compute the total variation distance of the distribution of \mathcal{M}_{blocks} after t steps from its stationary distribution π , i.e.

$$d_{TV}(P^t(a_0, \cdot), \pi) = \frac{1}{2} \cdot \sum_{a \in S(n, A)} \left| P^t(a_0, a) - \pi(a) \right|,$$

we do not need to know all building block sequences in $S(n, A)$. Let $A_t \subseteq S(n, A)$ be the set of all building block sequences reached by \mathcal{M}_{blocks} after t steps in at least one run. Then, the total variation distance after t steps is

$$d_{TV}(P^t(a_0, \cdot), \pi) = \frac{1}{2} \cdot \left((D(n, A) - \sum_{a \in A_t} (m(a) \cdot perm(a))) + \sum_{a \in A_t} \left| P^t(a_0, a) - \pi(a) \right| \right).$$

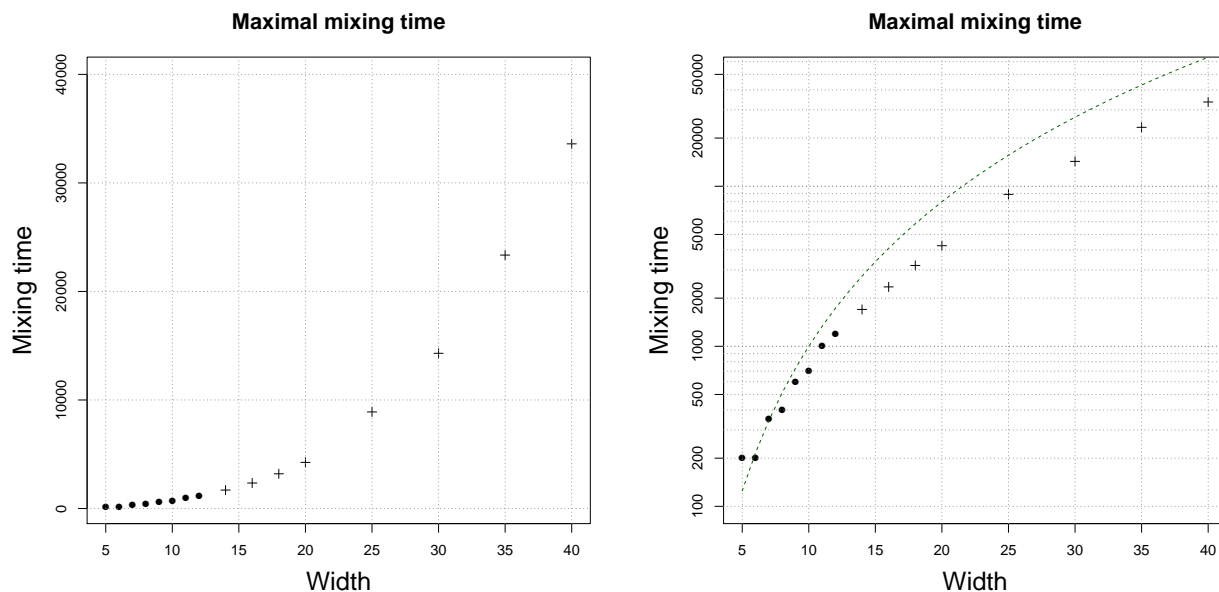


Figure 9: Plot of the maximal mixing time for a given width (filled circle) and mixing times for area A_n^* (cross) with a linear y-axis (left) and a logarithmic y-axis (right). The dashed line in the right plot shows $f(x) = x^3$.

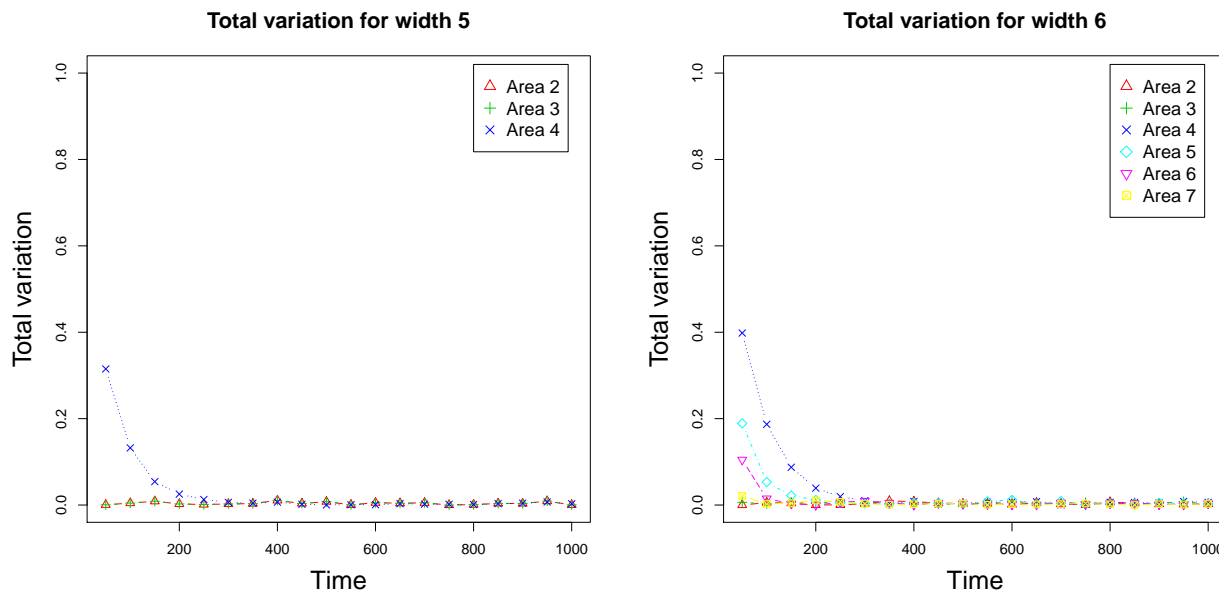


Figure 10: Total variation for width $n = 5$ (left), respectively $n = 6$ (right), and every area $A \leq (n/2)^2$ with $|S(n, A)| > 1$.

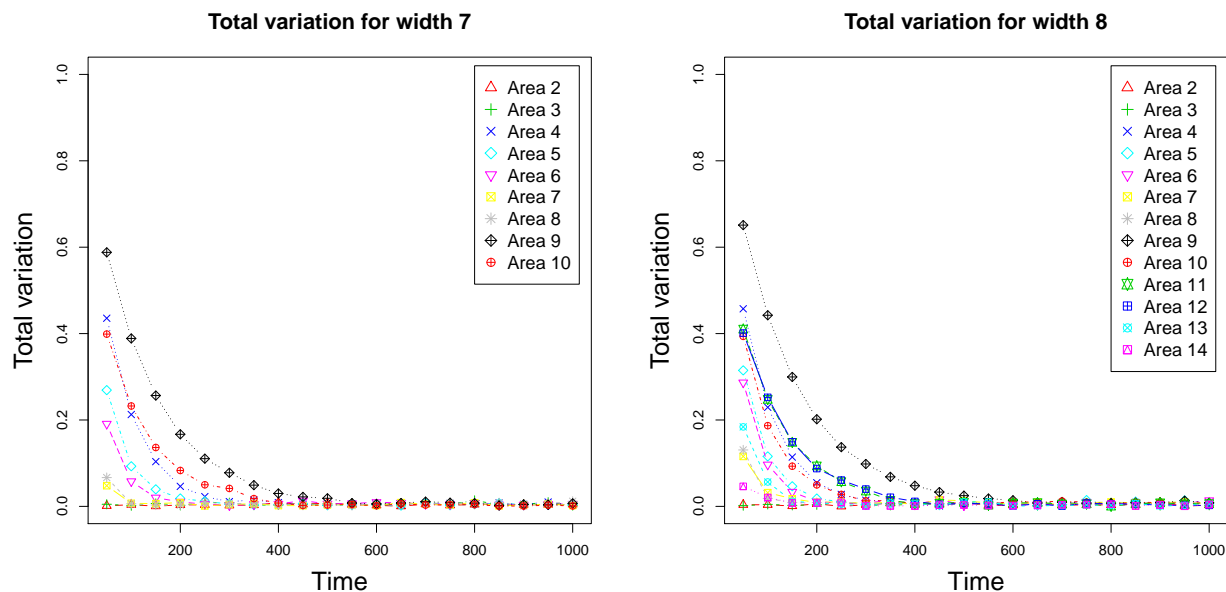


Figure 11: Total variation for width $n = 7$ (left), respectively $n = 8$ (right), and every area $A \leq (n/2)^2$ with $|S(n, A)| > 1$.

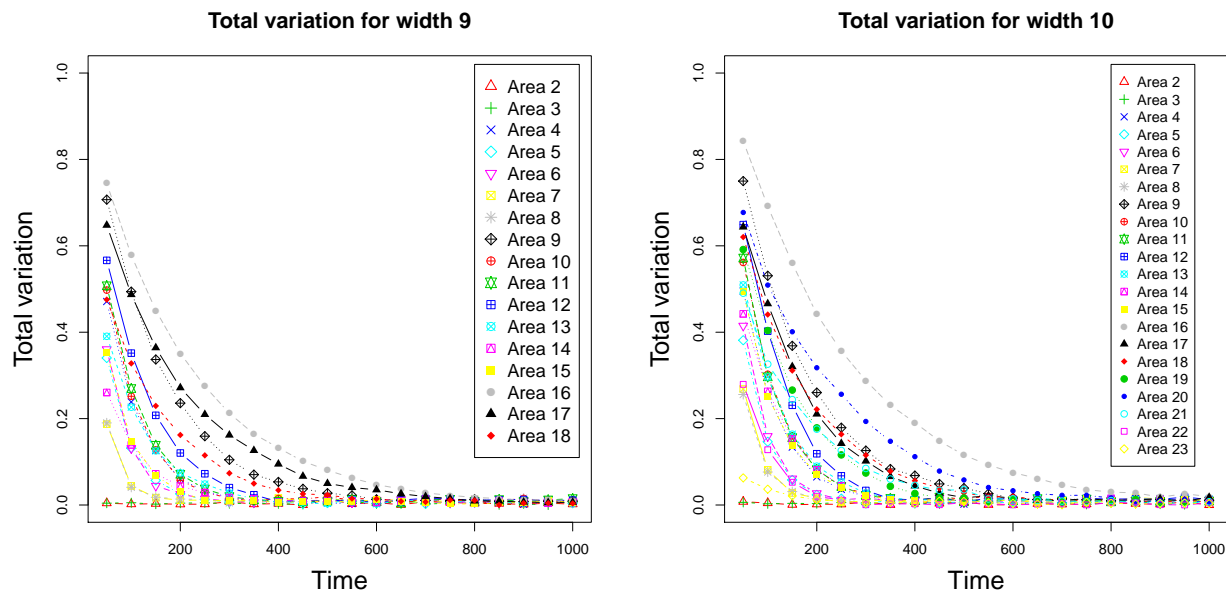


Figure 12: Total variation for width $n = 9$ (left), respectively $n = 10$ (right), and every area $A \leq (n/2)^2$ with $|S(n, A)| > 1$.

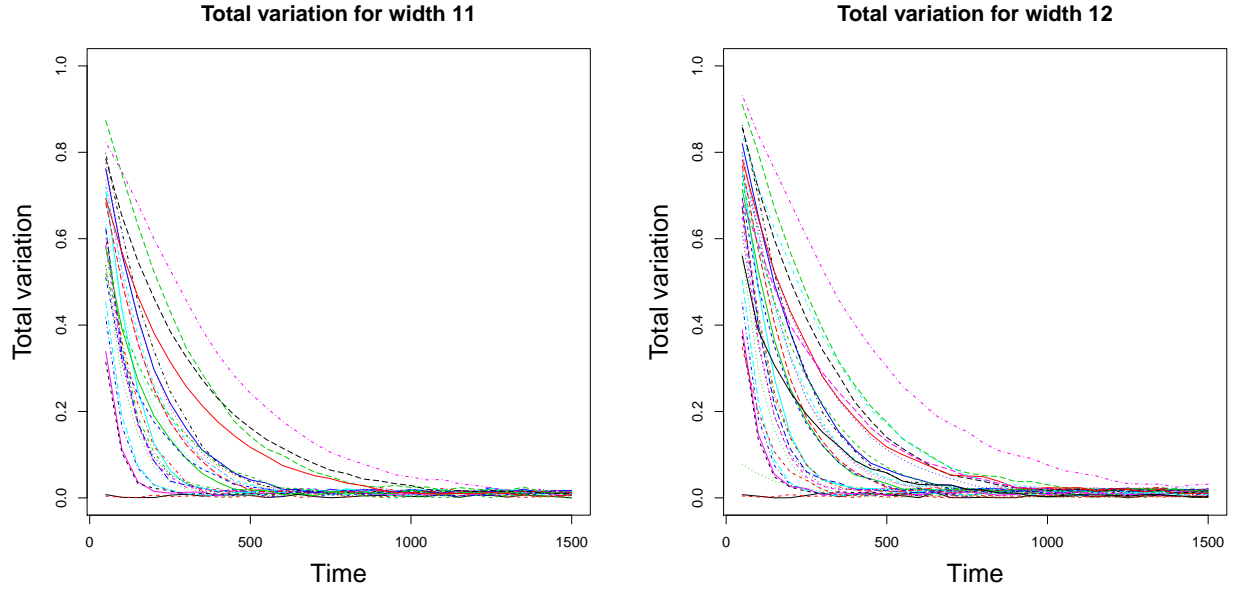


Figure 13: Total variation for width $n = 11$ (left), respectively $n = 12$ (right), and every area $A \leq (n/2)^2$ with $|S(n, A)| > 1$.

C Top-Down Dynamic Programming Approach

For the sake of clarity we first describe the dynamic programming procedure for counting *unweighted* Motzkin paths of any given width n and area A .¹ Specifically, we define the following subproblem:

$$M(n, A, h, p) = \text{the number of Motzkin paths of width } n, \\ \text{area } A, \text{ height } h \text{ and with } p_h = p \text{ and } f_h = 0.$$

Note that we forbid flats of height h and fix the number of peaks at height h . To count the number of paths of a given n , A and height h including flats at height h (so dropping the $f_h = 0$ condition) and not fixing the number of peaks at height h we can simply compute

$$M(n, A, h) := M(n, A, h + 1, 0).$$

The number of Motzkin paths of given width and area can be obtained by summing over all h :

$$M(n, A) := \sum_{h=0}^n M(n, A, h).$$

If we also sum over all possible areas, we get the classic Motzkin numbers

$$M(n) := \sum_{A=0}^{n^2} M(n, A).$$

We next show that M can be computed in polynomial time. We make use of Equation (7) in Theorem 8 for the number $m(a)$ of Motzkin paths that can be constructed out of building sequence a .

¹Though this task can be solved by an easier dynamic programming, our approach does extend to the weighted case, which is our main goal.

Theorem 14. *Table $M(n, A, h, p)$ can be computed for all the $\mathcal{O}(n^5)$ many possible parameters in $\mathcal{O}(n^7)$ total time.*

Proof. Recall that in M we forbid flats of height h and fix the number of peaks of height h . This allows the following recursive counting:

Base case. If any of the parameters is equal 0 then

$$M(n, A, h, p) := \begin{cases} 1 & \text{if } n = A = h = p = 0, \\ 0 & \text{else if } n \leq 0 \text{ or } A \leq 0 \text{ or } h \leq 0. \end{cases} \quad (13)$$

Recursion. Otherwise, when all parameters are strictly positive, we have

$$M(n, A, h, p) := \sum_{f=0}^n \binom{p+f}{f} \cdot \left(\sum_{p'=1}^{n/2} \binom{p+f+p'-1}{p'-1} \cdot M(n', A', h-1, p') \right), \quad (14)$$

where $n' = n - 2p - f$ and $A' = A - (2h - 1)p - (h - 1)f$.

In the recursive case, we enumerate all potential numbers of flats and peaks at height $h - 1$ so that we can look up the corresponding subproblems. These subproblems are then weighted by the number of possible interleavings $\binom{p+f}{f} \binom{p+f+p'-1}{p'-1}$ which we derived in Equations (8) and (9). We have thus shown that the table $M(n, A, h, p)$ above can be computed for all the $\mathcal{O}(n^5)$ many possible parameter values in time $\mathcal{O}(n^7)$ with the bottleneck being the two nested sums in Equation (14). \square

Let us now consider the problem of counting *weighted* Motzkin paths, that is, the function $D(n, d)$ in Equation (2). To this end, we extend the definition of M above so to count each path mz according to its weight $perm(a^{(mz)})$. The resulting table $D(n, d, h, p)$ can be computed recursively in a top-down fashion by incorporating into the recursion of $M(n, A, h, p)$ the two terms defining $perm(a^{(mz)})$ in Equation (6):

$$D(n, d, h, p) := \underbrace{h^{2p}}_{\text{peak options}} \cdot \sum_{f=0}^n \underbrace{(2h-1)^f}_{\text{flat options}} \cdot \binom{p+f}{f} \cdot \left(\sum_{p'=1}^{n/2} \binom{p+f+p'-1}{p'-1} \cdot D(n', d', h-1, p') \right), \quad (15)$$

where $n' = n - 2p - f$ and $d' = d - (2h - 1)p - (h - 1)f$. The base case is identical to the unweighted case Equation (13). Again, we can drop the condition that there are no flats on the last level h :

$$D(n, d, h) := D(n, d, h + 1, 0),$$

and count all weighted Motzkin paths of given area and width simply as

$$D(n, d) := \sum_{h=0}^n D(n, d, h).$$

Both the time for computing $D(n, d, h, p)$ and its overall space are asymptotically the same as those used for $M(n, A, h, p)$. We have thus proven the following:

Corollary 2. *Table $D(n, d, h, p)$ and its marginals $D(n, d, h)$ and $D(n, d)$ can be computed for all $\mathcal{O}(n^5)$ many possible parameter values in $\mathcal{O}(n^7)$ total time.*