Chapter 5

The swap-only and swap-and-reverse algorithms

This chapter (along with chapter 7) presents Markov chains for MCMC sampling of the model of random spatial permutations (chapter 2) within the MCMC-recipe framework of chapter 4. The algorithms are proved correct, then compared and contrasted. Computational results using the swap-and-reverse algorithm are given in chapter 11.

5.1 The swap-only algorithm

The swap-only algorithm for transitioning from π to π', within the context of the recipe in section 4.5, is as follows. One sweeps through sites $\mathbf{x}$ of the lattice in either sequential or uniform-random order. In either case[1], one obtains a lattice site $\mathbf{x}$. One then does a *Metropolis step* at site $\mathbf{x}$:

- Choose a site $\pi(\mathbf{y})$ from among the six nearest neighbors of $\pi(\mathbf{x})$.
- Propose to change π to the permutation π' which has $\pi'(\mathbf{z}) = \pi(\mathbf{z})$ for all $\mathbf{z} \neq \mathbf{x}, \mathbf{y}$ but $\pi'(\mathbf{x}) = \pi(\mathbf{y})$ and $\pi'(\mathbf{y}) = \pi(\mathbf{x})$. (See figure 5.1.)
- With probability $\min\{1, e^{-\Delta H}\}$ where $\Delta H = H(\pi') - H(\pi)$, accept the change. (If the change is rejected, set $\pi' = \pi$.)

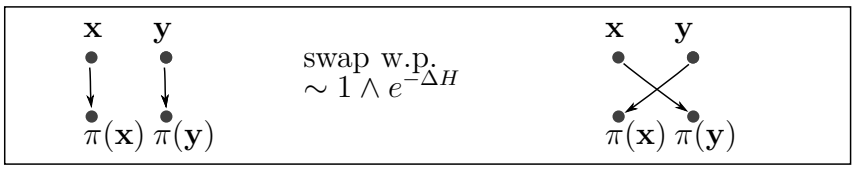

Figure 5.1. Metropolis moves for the swap-only algorithm.

Definition 5.1.1. A swap is *trivial* if $\mathbf{x} = \mathbf{y}$.

[1]For computational results presented in chapter 11, site selection was sequential. Experiments show that sequential site selection and random site selection produce indistinguishable results, for our model and for the near-critical temperature range we consider.

5.2 Explicit construction of the Markov matrices

For section 5.3 we will need an explicit construction of the Markov matrices corresponding to the swap-only algorithm as described in section 5.1.

The Markov perspective on the algorithm is that, given a probability distribution $P_k^{(\pi_0)}(\pi)$, the distribution for the subsequent permutation is

$$P_{k+1}^{(\pi_0)}(\pi') = \sum_{\pi \in \mathcal{S}_N} P_k^{(\pi_0)}(\pi) A_k(\pi, \pi')$$

or, in matrix/vector notation, $P_{k+1}^{(\pi_0)} = P_k^{(\pi_0)} A_k$. In this section we precisely describe the matrices A_k; in section 5.3 we show that $P_k^{(\pi_0)}$ approaches the Gibbs distribution P_{Gibbs} (equation (2.1.4)) as $k \to \infty$.

The matrices A_k are $N! \times N!$, with rows indexed by $\pi_1, \ldots, \pi_{N!}$ and columns indexed by $\pi'_1, \ldots, \pi'_{N!}$. Most of the entries of A_k are zero: Metropolis steps change only two permutation sites whereas most π, π' differ at more than two sites.

Definition 5.2.1. For $\pi, \pi' \in \mathcal{S}_N$, define

$$d(\pi, \pi') = \#\{i = 1, 2, \ldots, N : \pi(i) \neq \pi'(i)\}.$$

Remark. Note that $d(\pi, \pi') \neq 1$ since if two permutations agree on $N-1$ sites, they must agree on the remaining site.

Lemma 5.2.2. *The function $d(\pi, \pi')$ is a metric on $\mathcal{S}_N$.*

Proof. Symmetry is obvious, as is non-negativity. For positive definiteness, note that $d(\pi, \pi') = 0$ iff $\pi = \pi'$. For the triangle inequality, let $\pi, \pi', \pi'' \in \mathcal{S}_N$. Partition the set $\{1, 2, \ldots, N\}$ into the four disjoint sets

$$\begin{aligned}
A &= \{i = 1, 2, \ldots, N : \pi(i) = \pi'(i), \pi'(i) = \pi''(i)\}, \\
B &= \{i = 1, 2, \ldots, N : \pi(i) = \pi'(i), \pi'(i) \neq \pi''(i)\}, \\
C &= \{i = 1, 2, \ldots, N : \pi(i) \neq \pi'(i), \pi'(i) = \pi''(i)\}, \\
D &= \{i = 1, 2, \ldots, N : \pi(i) \neq \pi'(i), \pi'(i) \neq \pi''(i)\}.
\end{aligned}$$

Then $\pi = \pi''$ on all of A; $\pi \neq \pi''$ on all of B and C; and π, π'' may or may not agree on various elements of D:

A	B	C	D
$\pi = \pi'$	$\pi = \pi'$	$\pi \neq \pi'$	$\pi \neq \pi'$
$\pi' = \pi''$	$\pi' \neq \pi''$	$\pi' = \pi''$	$\pi' \neq \pi''$
$\pi = \pi''$	$\pi \neq \pi''$	$\pi \neq \pi''$	Varies

That is,

$$\begin{aligned} d(\pi, \pi') &= \#C + \#D, \\ d(\pi', \pi'') &= \#B + \#D, \\ \#B + \#C \leq d(\pi, \pi'') &\leq \#B + \#C + \#D. \end{aligned}$$

Then

$$d(\pi, \pi'') \leq \#B + \#C + \#D \leq \#B + \#C + 2\#D = d(\pi, \pi') + d(\pi', \pi'').$$

□

Definition 5.2.3. Lattice sites $\mathbf{x}$ and $\mathbf{y}$ are *nearest-neighbor* if $\|\mathbf{x} - \mathbf{y}\|_\Lambda = 1$.

Definition 5.2.4. For $\pi \in \mathcal{S}_N$ and $\mathbf{x} \in \Lambda$, define

$$R_{\mathbf{x}}(\pi) = \{\pi' \in \mathcal{S}_N : d(\pi, \pi') = 2 \text{ and } \|\pi(\mathbf{x}) - \pi(\mathbf{y})\|_\Lambda = 1\}$$

where the $\mathbf{x}$ and $\mathbf{y}$ are taken to be the two points at which π and π' differ. Then $R_{\mathbf{x}}(\pi)$ is the set of permutations π' reachable from π on a swap involving site $\mathbf{x}$. This is used for sequential site selection. Likewise, for use with random site selection, define

$$R(\pi) = \{\pi' \in \mathcal{S}_N : d(\pi, \pi') = 2 \text{ and } \|\pi(\mathbf{x}) - \pi(\mathbf{y})\|_\Lambda = 1\}$$

where the $\mathbf{x}$ and $\mathbf{y}$ are taken to be the two points at which π, π' differ. Then $R(\pi)$ is the set of permutations π' reachable from π on a swap. We also write

$$\pi' \circ\!\!-\!\!\circ \pi$$

if $\pi' \in R_{\mathbf{x}}(\pi)$ or $\pi' = \pi$ (for sequential site selection), or $\pi' \in R(\pi)$ or $\pi' = \pi$ (for random site selection).

The Metropolis steps are then described as follows. First consider sequential site selection. For each $\pi \in \mathcal{S}_N$,

$$A_{\mathbf{x}}(\pi, \pi') = \begin{cases} \frac{1}{6}\left(1 \wedge e^{-H(\pi')+H(\pi)}\right), & \pi' \in R_{\mathbf{x}}(\pi), \\ 1 - \displaystyle\sum_{\pi'' \in R(\pi)} \frac{1}{6}\left(1 \wedge e^{-H(\pi'')+H(\pi)}\right), & \pi = \pi'; \\ 0, & \text{otherwise.} \end{cases} \tag{5.2.5}$$

To justify the choice of prefactor 1/6, note that we choose one of six $\pi(\mathbf{y})$ at uniform random from the six nearest-neighbor lattice sites of $\pi(\mathbf{x})$. If the change is accepted, then we obtain π by swapping at $\mathbf{x}$ and $\mathbf{y}$; otherwise, we take $\pi' = \pi$.

Next we construct the Markov matrix for random site selection. For each $\pi \in \mathcal{S}_N$,

$$A(\pi, \pi') = \begin{cases} \frac{1}{3N}\left(1 \wedge e^{-H(\pi')+H(\pi)}\right), & \pi' \in R(\pi), \\ 1 - \sum_{\pi'' \in R(\pi)} \frac{1}{3N}\left(1 \wedge e^{-H(\pi'')+H(\pi)}\right), & \pi = \pi'; \\ 0, & \text{otherwise.} \end{cases} \tag{5.2.6}$$

To justify the choice of prefactor $1/3N$, note that there are N choices of lattice points $\mathbf{x}$. For each $\mathbf{x}$, there are 6 choices of $\pi(\mathbf{y})$ which are nearest neighbors to $\pi(\mathbf{x})$. This double-counts the $3N$ distinct choices of π' reachable from π in a single Metropolis step, since choosing $\mathbf{x}$ and then $\mathbf{y}$ results in the same Metropolis step as choosing $\mathbf{y}$ and then $\mathbf{x}$.

The use of the Markov matrices in practice is as follows. Number the lattice sites $\mathbf{x}_1, \ldots, \mathbf{x}_N$. When using sequential site selection, the kth Metropolis sweep (as described in section 4.5) begins with a permutation π distributed according to $P_k^{(\pi_0)}$. A Metropolis step is done at site $\mathbf{x}_1$, using transition matrix $A_{\mathbf{x}_1}$, followed by a Metropolis step at site $\mathbf{x}_2$, using transition matrix $A_{\mathbf{x}_2}$, and so on up to site $\mathbf{x}_N$. The sweep is then complete, and the distribution of π' is

$$P_{k+1}^{(\pi_0)}(\pi') = P_k^{(\pi_0)}(\pi)\, A_k, \qquad\qquad A_k = A_{\mathbf{x}_N} \cdots A_{\mathbf{x}_1}.$$

The chain is non-homogeneous, if we consider all the intermediate permutations after each Metropolis step. Yet, at the level of Metropolis sweeps, the chain is homogeneous since at each sweep we apply the composite transition matrix $A_{\mathbf{x}_N} \cdots A_{\mathbf{x}_1}$ to obtain π' from π.

When we use random site selection, the kth Metropolis sweep begins with a permutation π distributed according to $P_k^{(\pi_0)}$. We do N Metropolis steps, each using the transition matrix A, each selecting a site $\mathbf{x}$ at uniform random from among $\mathbf{x}_1, \ldots, \mathbf{x}_N$. The sweep is then complete, and the distribution of π' is

$$P_{k+1}^{(\pi_0)}(\pi') = P_k^{(\pi_0)}(\pi)\, A_k, \qquad\qquad A_k = A^N.$$

The chain is homogeneous, whether viewed at the level of Metropolis steps or Metropolis sweeps.

5.3 Correctness of the swap-only algorithm

It is clear that the swap-only algorithm produces a sequence of permutations, but with what distribution? From the Markov-chain theory in section 4.2, we know that if the chain is irreducible, aperiodic, and satisfies detailed balance, then the chain has the Gibbs distribution (equation (2.1.4)) as its unique invariant distribution. All the results in this section will apply for sequential or random site selection; in this section, we write A to refer to either A or $A_{\mathbf{x}}$.

Proposition 5.3.1 (Irreducibility)**.** *For all π, π', there is an n such that $A^n(\pi, \pi') > 0$. That is, any permutation is reachable from any other.*

Proof. Transpositions generate $\mathcal{S}_N$ [DF]. Thus, for all $\pi \in \mathcal{S}_N$, there exist transpositions $\sigma_1, \ldots, \sigma_m$ such that $\pi = \prod_{j=1}^{m} \sigma_j$. Thus, it suffices to show that given any permutation π and any two points $\mathbf{x}$ and $\mathbf{z}$, so $\pi : \mathbf{x} \mapsto \pi(\mathbf{x})$ and $\pi : \mathbf{z} \mapsto \pi(\mathbf{z})$, we can construct a sequence of swaps sending π to π' so that $\pi' : \mathbf{x} \mapsto \pi(\mathbf{z})$, $\pi' : \mathbf{z} \mapsto \pi(\mathbf{x})$, and $\pi'(\mathbf{y}) = \pi(\mathbf{y})$ for all $\mathbf{y} \neq \mathbf{x}, \mathbf{z}$. (If $\pi(\mathbf{x})$ and $\pi(\mathbf{z})$ are nearest-neighbor lattice sites, of course, then a single swap does the job.)

Define $G_{\mathbf{a},\mathbf{b}} : \mathcal{S}_N \to \mathcal{S}_N$ to be the swap operator for nearest-neighbor lattice sites $\pi(\mathbf{a})$ and $\pi(\mathbf{b})$, i.e. $\pi' = G_{\mathbf{a},\mathbf{b}}\pi$. Given $\mathbf{x}$ and $\mathbf{z}$, there is a (non-unique) sequence of lattice sites $\mathbf{y}_0, \mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n$ such that $\mathbf{y}_0 = \mathbf{x}$, $\mathbf{y}_n = \mathbf{z}$, and $\|\pi(\mathbf{y}_{i+1}) - \pi(\mathbf{y}_i)\|_\Lambda = 1$ for $i = 0, 1, \ldots, n-1$. (See figure 5.2.) We will construct a sequence of swaps along this nearest-neighbor path whose end result is to swap the permutation arrows starting at $\mathbf{x}$ and $\mathbf{z}$, leaving all other arrows unchanged. We first need a lemma about compositions of swaps.

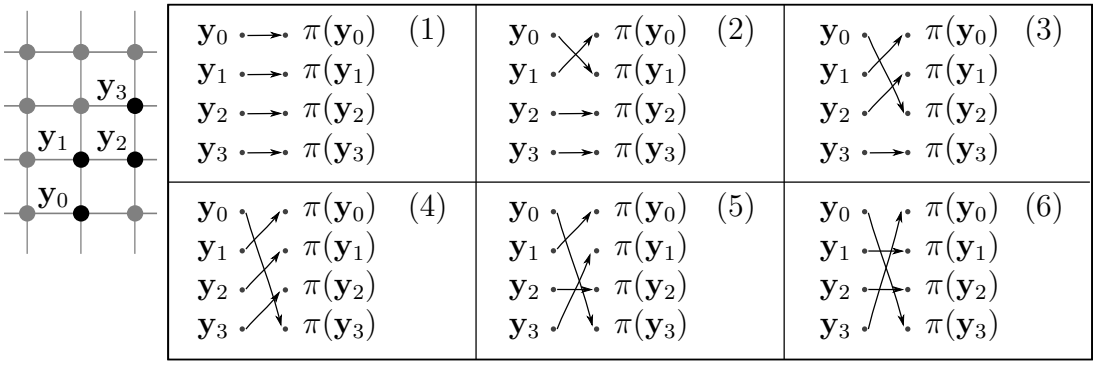

FIGURE 5.2. A sequence of (nearest-neighbor) swaps which results in a non-nearest-neighbor swap.

Notation 5.3.2. Given $\mathbf{x}_1, \ldots, \mathbf{x}_N$ and a permutation π, we may write π as an *image map* with the $\mathbf{x}_i$'s along the top row and their images along the bottom row:

$$\begin{pmatrix} \mathbf{x}_1 & \ldots & \mathbf{x}_N \\ \pi(\mathbf{x}_1) & \ldots & \pi(\mathbf{x}_N) \end{pmatrix}$$

Lemma 5.3.3. *The composition $G_{\mathbf{a},\mathbf{c}} \circ G_{\mathbf{a},\mathbf{b}}$ behaves as follows:*

$$\pi = \begin{pmatrix} \ldots & \mathbf{a} & \mathbf{b} & \mathbf{c} & \ldots \\ \ldots & \pi(\mathbf{a}) & \pi(\mathbf{b}) & \pi(\mathbf{c}) & \ldots \end{pmatrix} \mapsto \pi' = \begin{pmatrix} \ldots & \mathbf{a} & \mathbf{b} & \mathbf{c} & \ldots \\ \ldots & \pi(\mathbf{c}) & \pi(\mathbf{a}) & \pi(\mathbf{b}) & \ldots \end{pmatrix}$$

Proof. The first map, $G_{\mathbf{a},\mathbf{b}}$, does

$$\begin{pmatrix} \dots & \mathbf{a} & \mathbf{b} & \mathbf{c} & \dots \\ \dots & \pi(\mathbf{a}) & \pi(\mathbf{b}) & \pi(\mathbf{c}) & \dots \end{pmatrix} \mapsto \begin{pmatrix} \dots & \mathbf{a} & \mathbf{b} & \mathbf{c} & \dots \\ \dots & \pi(\mathbf{b}) & \pi(\mathbf{a}) & \pi(\mathbf{c}) & \dots \end{pmatrix};$$

$G_{\mathbf{a},\mathbf{c}}$ sends this to

$$\begin{pmatrix} \dots & \mathbf{a} & \mathbf{b} & \mathbf{c} & \dots \\ \dots & \pi(\mathbf{c}) & \pi(\mathbf{a}) & \pi(\mathbf{b}) & \dots \end{pmatrix}.$$

□

Corollary 5.3.4. *The composition $G_{\mathbf{y}_0,\mathbf{y}_n} \circ G_{\mathbf{y}_0,\mathbf{y}_{n-1}} \circ \ldots \circ G_{\mathbf{y}_0,\mathbf{y}_2} \circ G_{\mathbf{y}_0,\mathbf{y}_1}$, sending $\pi \mapsto \pi'$, performs the right cyclic shift on images of $\mathbf{y}_0, \ldots, \mathbf{y}_n$ given by*

$$\begin{pmatrix} \mathbf{y}_0 & \mathbf{y}_1 & \cdots & \mathbf{y}_{n-1} & \mathbf{y}_n \\ \pi(\mathbf{y}_0) & \pi(\mathbf{y}_1) & \dots & \pi(\mathbf{y}_{n-1}) & \pi(\mathbf{y}_n) \end{pmatrix} \mapsto \begin{pmatrix} \mathbf{y}_0 & \mathbf{y}_1 & \cdots & \mathbf{y}_{n-1} & \mathbf{y}_n \\ \pi(\mathbf{y}_n) & \pi(\mathbf{y}_0) & \dots & \pi(\mathbf{y}_{n-2}) & \pi(\mathbf{y}_{n-1}) \end{pmatrix}.$$

Likewise, $G_{\mathbf{y}_n,\mathbf{y}_1} \circ G_{\mathbf{y}_n,\mathbf{y}_2} \circ \ldots \circ G_{\mathbf{y}_n,\mathbf{y}_{n-2}} \circ G_{\mathbf{y}_n,\mathbf{y}_{n-1}}$ leaves the image of $\mathbf{y}_0$ unchanged and performs the left cyclic shift on images of $\mathbf{y}_1, \ldots, \mathbf{y}_n$ given by

$$\begin{pmatrix} \mathbf{y}_0 & \mathbf{y}_1 & \cdots & \mathbf{y}_{n-1} & \mathbf{y}_n \\ \pi(\mathbf{y}_0) & \pi(\mathbf{y}_1) & \dots & \pi(\mathbf{y}_{n-1}) & \pi(\mathbf{y}_n) \end{pmatrix} \mapsto \begin{pmatrix} \mathbf{y}_0 & \mathbf{y}_1 & \cdots & \mathbf{y}_{n-1} & \mathbf{y}_n \\ \pi(\mathbf{y}_0) & \pi(\mathbf{y}_2) & \dots & \pi(\mathbf{y}_n) & \pi(\mathbf{y}_1) \end{pmatrix}.$$

Proof. These follow from the lemma by induction on n. □

Composing these two maps, we find that

$$(G_{\mathbf{y}_n,\mathbf{y}_1} \circ G_{\mathbf{y}_n,\mathbf{y}_2} \circ \ldots \circ G_{\mathbf{y}_n,\mathbf{y}_{n-2}} \circ G_{\mathbf{y}_n,\mathbf{y}_{n-1}}) \circ (G_{\mathbf{y}_0,\mathbf{y}_n} \circ G_{\mathbf{y}_0,\mathbf{y}_{n-1}} \circ \ldots \circ G_{\mathbf{y}_0,\mathbf{y}_2} \circ G_{\mathbf{y}_0,\mathbf{y}_1})$$

swaps the images of $\mathbf{x} = \mathbf{y}_0$ and $\mathbf{z} = \mathbf{y}_n$ while leaving all other images unchanged, that is,

$$\begin{pmatrix} \mathbf{y}_0 & \mathbf{y}_1 & \cdots & \mathbf{y}_{n-1} & \mathbf{y}_n \\ \pi(\mathbf{y}_0) & \pi(\mathbf{y}_1) & \dots & \pi(\mathbf{y}_{n-1}) & \pi(\mathbf{y}_n) \end{pmatrix} \mapsto \begin{pmatrix} \mathbf{y}_0 & \mathbf{y}_1 & \cdots & \mathbf{y}_{n-1} & \mathbf{y}_n \\ \pi(\mathbf{y}_n) & \pi(\mathbf{y}_1) & \dots & \pi(\mathbf{y}_{n-1}) & \pi(\mathbf{y}_0) \end{pmatrix}.$$

This ends the proof of propostion 5.3.1. □

Remark. Below we will discuss winding cycles, and the empirical fact that the swap-only algorithm reaches them only rarely. The chain is irreducible but the non-zero transition probabilities can still be very small.

Proposition 5.3.5 (Aperiodicity)**.** *The swap-only algorithm's Markov chain is aperiodic.*

Proof. This follows from irreducibility, which says in particular that for every π, there is an integer m such that $A^m(\pi, \pi) > 0$. Then $A^n(\pi, \pi) > 0$ for all $n > m$, implying $p(\pi) = 1$. □

Proposition 5.3.6 (Detailed balance)**.** *For all* $\pi, \pi' \in \mathcal{S}_N$*,*

$$P_{\text{Gibbs}}(\pi)A(\pi, \pi') = P_{\text{Gibbs}}(\pi')A(\pi', \pi). \tag{5.3.7}$$

Remark. For the swap-only algorithm, this is a trivial result. We work through the details in order to foreshadow the non-trivial construction of proposition 7.7.5 for the worm algorithm.

Proof. The detailed-balance statement in terms of the Gibbs distribution (equation (2.1.4)) and the swap-only Metropolis transition matrices (equations (5.2.5) and (5.2.6)) is

$$\frac{e^{-H(\pi)}}{Z}\left(1 \wedge e^{-H(\pi')}e^{H(\pi)}\right) \stackrel{?}{=} \frac{e^{-H(\pi')}}{Z}\left(1 \wedge e^{-H(\pi)}e^{H(\pi')}\right).$$

The Z's cancel. The lemma below shows that $A(\pi, \pi') \neq 0$ iff $A(\pi', \pi) \neq 0$. If $A(\pi, \pi') = 0$, then detailed balance holds. If $A(\pi, \pi') \neq 0$, then there are two cases. If $H(\pi') \leq H(\pi)$, then

$$e^{-H(\pi)}\,(1) = e^{-H(\pi')}\left(e^{-H(\pi)}e^{H(\pi')}\right).$$

If $H(\pi') > H(\pi)$,

$$e^{-H(\pi)}\left(e^{-H(\pi')}e^{H(\pi)}\right) = e^{-H(\pi')}\,(1).$$

In all cases, detailed balance holds. □

Lemma 5.3.8. *For all* $\pi, \pi' \in \mathcal{S}_N$*,*

$$A(\pi, \pi') \neq 0 \iff A(\pi', \pi) \neq 0.$$

Proof. As a direct consequence of definition 5.2.4 of $R(\pi)$, $\pi' \in R(\pi)$ if and only if $\pi \in R(\pi')$. The same holds for $A_{\mathbf{x}}$ and $R_{\mathbf{x}}(\pi)$. □

This lemma completes the proof that the swap-only algorithm satisfies detailed balance and thus has the Gibbs distribution as its invariant distribution.

The following proposition is not a correctness result, but rather a sanity check. It shows that cycles may grow or shrink upon swaps.

Proposition 5.3.9. *If* $\mathbf{x}$ *and* $\mathbf{y}$ *are in disjoint cycles before a non-trivial swap at* $\mathbf{x}$ *and* $\mathbf{y}$*, then they are in the same cycle afterward and vice versa (see figure 5.3).*

FIGURE 5.3. Swaps merge disjoint cycles and split single cycles. The left-hand permutation can be reached from the right-hand permutation via a swap, and vice versa.

Proof. First suppose that $\mathbf{x}$ and $\mathbf{y}$ are in disjoint cycles. Let the respective cycle lengths be $\ell(\mathbf{x}) = a$ and $\ell(\mathbf{y}) = b$. Recall that $1 \leq a, b \leq N$. Those cycles are

$$\mathbf{x} \mapsto \pi(\mathbf{x}) \mapsto \pi^2(\mathbf{x}) \mapsto \ldots \mapsto \pi^{a-1}(\mathbf{x}) \mapsto \mathbf{x}$$

and

$$\mathbf{y} \mapsto \pi(\mathbf{y}) \mapsto \pi^2(\mathbf{y}) \mapsto \ldots \mapsto \pi^{b-1}(\mathbf{y}) \mapsto \mathbf{y}.$$

Since these are disjoint cycles, all elements listed are distinct lattice sites. After the swap, we have

$$\mathbf{y} \mapsto \pi(\mathbf{x}) \mapsto \pi^2(\mathbf{x}) \mapsto \ldots \mapsto \pi^{a-1}(\mathbf{x}) \mapsto \mathbf{x}$$

and

$$\mathbf{x} \mapsto \pi(\mathbf{y}) \mapsto \pi^2(\mathbf{y}) \mapsto \ldots \mapsto \pi^{b-1}(\mathbf{y}) \mapsto \mathbf{y}.$$

This is a single cycle of length $a+b$, starting with $\mathbf{y}$, including $\mathbf{x}$, and returning to $\mathbf{y}$.

Second, suppose that $\mathbf{x}$ and $\mathbf{y}$ are in the same cycle. Let $a = \ell_{\mathbf{x},\mathbf{y}}(\pi)$ and $b = \ell_{\mathbf{x},\mathbf{y}}(\pi)$ (see definition 3.2.2). (These numbers are both positive since the swap is non-trivial, i.e. $\mathbf{x} \neq \mathbf{y}$.) Then we have

$$\mathbf{x} \mapsto \pi(\mathbf{x}) \mapsto \pi^2(\mathbf{x}) \mapsto \pi^{a-1}(\mathbf{x}) \mapsto \mathbf{y} \mapsto \pi(\mathbf{y}) \mapsto \pi^2(\mathbf{y}) \mapsto \ldots \mapsto \pi^{b-1}(\mathbf{y}) \mapsto \mathbf{x}.$$

This is a single cycle of length $a + b$; all lattice sites listed are distinct. After the swap, we have

$$\mathbf{y} \mapsto \pi(\mathbf{x}) \mapsto \pi^2(\mathbf{x}) \mapsto \pi^{a-1}(\mathbf{x}) \mapsto \mathbf{y} \quad \text{and} \quad \mathbf{x} \mapsto \pi(\mathbf{y}) \mapsto \pi^2(\mathbf{y}) \mapsto \ldots \mapsto \pi^{b-1}(\mathbf{y}) \mapsto \mathbf{x}.$$

These are disjoint cycles of length a and b, respectively; the first contains $\mathbf{x}$ and the second contains $\mathbf{y}$. □

Remark 5.3.10. If $\mathbf{x} \circ\!\!-\!\!\circ \mathbf{y}$, i.e. the swap splits their common cycle, the old cycle lengths $\ell_{\mathbf{x}}(\pi)$ and $\ell_{\mathbf{y}}(\pi)$ are equal, and the new cycle lengths after the swap are

$$\ell_{\mathbf{x}}(\pi') = \ell_{\mathbf{y},\mathbf{x}}(\pi) \qquad \text{and} \qquad \ell_{\mathbf{y}}(\pi') = \ell_{\mathbf{x},\mathbf{y}}(\pi).$$

Otherwise, i.e. the swap merges the disjoint cycles, we have

$$\ell_{\mathbf{x}}(\pi') = \ell_{\mathbf{y}}(\pi') = \ell_{\mathbf{x}}(\pi) + \ell_{\mathbf{y}}(\pi).$$

5.4 Winding numbers and the swap-and-reverse algorithm

The propositions of section 5.3 showed that the swap-only algorithm is correct, asymptotically in the number of Metropolis sweeps — in particular, any permutation is reachable from any other with non-zero probability. However, in practice some of these transition probabilities can be quite small. In particular, we observe empirically that the swap-only algorithm always generates permutations with zero winding number. This is readily proved.

Proposition 5.4.1. *In the short-jump-length regime (as discussed in sections 2.3, 3.1, and 3.6), the swap step of the swap-only algorithm preserves winding number.*

Remark. This means that a single jump of length on the order of $L/2$ — which happens with non-zero but very small probability — is required for the SO algorithm to change a winding number.

Proof. The permutations before and after the swap have winding numbers

$$\mathbf{W}(\pi) = \begin{pmatrix} W_x(\pi) \\ W_y(\pi) \\ W_z(\pi) \end{pmatrix} = \sum_{i=1}^{N} \mathbf{d}_\Lambda(\pi(\mathbf{x}_i), \mathbf{x}_i), \quad \mathbf{W}(\pi') = \begin{pmatrix} W_x(\pi') \\ W_y(\pi') \\ W_z(\pi') \end{pmatrix} = \sum_{i=1}^{N} \mathbf{d}_\Lambda(\pi'(\mathbf{x}_i), \mathbf{x}_i)$$

where the difference vector $\mathbf{d}_\Lambda$ is as defined in equation (3.1.4). Since we work in the regime of short jumps, and since $\pi'(\mathbf{x}_i)$ is a nearest neighbor of $\pi(\mathbf{x}_i)$, the Euclidean charts overlap and the $\mathbf{x}_i$'s cancel when we subtract $\mathbf{W}(\pi)$ from $\mathbf{W}(\pi')$. Also, π agrees with π' except at the two swap points $\mathbf{x}$ and $\mathbf{y}$; we have $\pi'(\mathbf{x}) = \pi(\mathbf{y})$ and vice versa. Thus the change in winding number is computed entirely in the same chart and we have

$$\begin{aligned} \mathbf{W}' - \mathbf{W} &= \frac{1}{L} \sum_{i=1}^{N} \mathbf{d}_\Lambda(\pi'(\mathbf{x}_i), \pi(\mathbf{x}_i)) = \frac{1}{L} \left[\mathbf{d}_\Lambda(\pi'(\mathbf{x}), \pi(\mathbf{x})) + \mathbf{d}_\Lambda(\pi'(\mathbf{y}), \pi(\mathbf{y})) \right] \\ &= \frac{1}{L} \left[\pi'(\mathbf{x}) - \pi(\mathbf{x}) + \pi'(\mathbf{y}) - \pi(\mathbf{y}) \right] = \frac{1}{L} \left[\pi(\mathbf{y}) - \pi(\mathbf{x}) + \pi(\mathbf{x}) - \pi(\mathbf{y}) \right] = 0. \end{aligned}$$

□

An example is shown in figure 5.4 in dimension $d = 2$ with $N = 8$ points on a lattice of width $L = 4$. The sites affected by the permutation swap are $\mathbf{x} = \mathbf{x}_3$ and

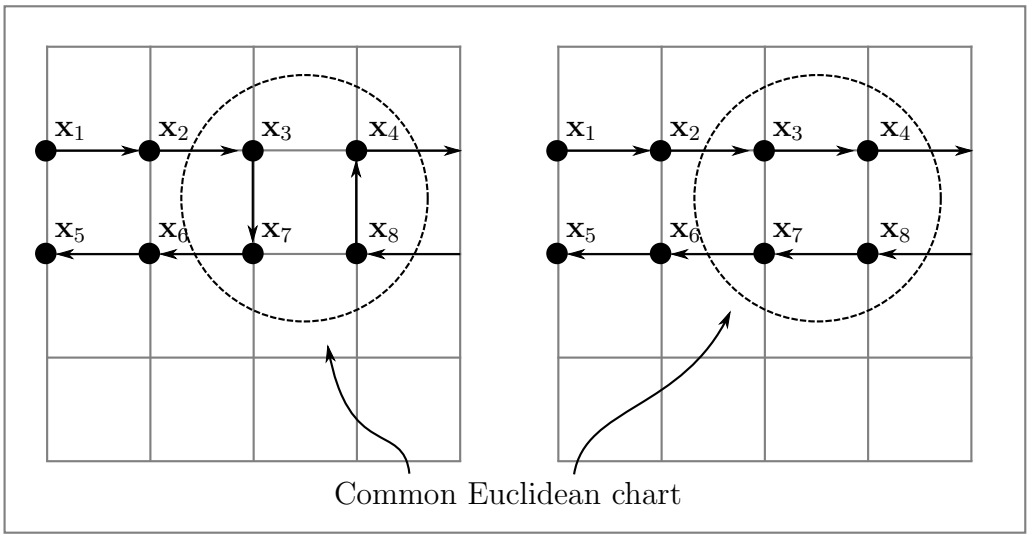

FIGURE 5.4. Example permutations π (left) and π' (right) illustrating winding-number conservation.

$\mathbf{y} = \mathbf{x}_7$. The change in winding numbers is

$$\begin{aligned}
\mathbf{W}(\pi') - \mathbf{W}(\pi) &= \frac{1}{4}\left[\mathbf{d}_\Lambda(\pi'(\mathbf{x}_3), \mathbf{x}_3) + \mathbf{d}_\Lambda(\pi'(\mathbf{x}_7), \mathbf{x}_7) - \mathbf{d}_\Lambda(\pi(\mathbf{x}_3), \mathbf{x}_3) - \mathbf{d}_\Lambda(\pi(\mathbf{x}_7), \mathbf{x}_7)\right] \\
&= \frac{1}{4}\left[\begin{pmatrix} 1 \\ 0 \end{pmatrix} + \begin{pmatrix} -1 \\ 0 \end{pmatrix}\right] - \frac{1}{4}\left[\begin{pmatrix} 0 \\ -1 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix}\right] = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \\
&= \frac{1}{4}\left[\mathbf{d}_\Lambda(\pi'(\mathbf{x}), \pi(\mathbf{x}))\right] + \frac{1}{4}\left[\mathbf{d}_\Lambda(\pi'(\mathbf{y}), \pi(\mathbf{x}))\right] \\
&= \frac{1}{4}\left[\begin{pmatrix} 1 \\ 1 \end{pmatrix} + \begin{pmatrix} -1 \\ -1 \end{pmatrix}\right].
\end{aligned}$$

A partial solution is explained intuitively by figure 5.5. Part 1 of the figure shows a permutation π with a long cycle on the torus which almost meets itself in the x direction. In part 2, after a Metropolis step sending π to π', one cycle winds by $+1$ and the other by -1. Metropolis steps create winding cycles only in opposite-direction pairs; the total $W_x(\pi)$ is still zero. Part 3 of the figure shows that if we reverse one cycle (which is a zero-energy move), $W_x(\pi)$ is now 2. In general, winding numbers of even parity can be generated. We are sampling from several, but not all, modes in a multimodal probability distribution on permutations which is indexed by the winding numbers W_x, W_y, and W_z.

The *swap-and-reverse* algorithm adds a second type of sweep to the swap-only algorithm. Namely: (1) In a swap-only sweep, for each lattice site one does a Metropolis step as above. (2) In a cycle-reversing sweep, for each cycle in the permutation, one

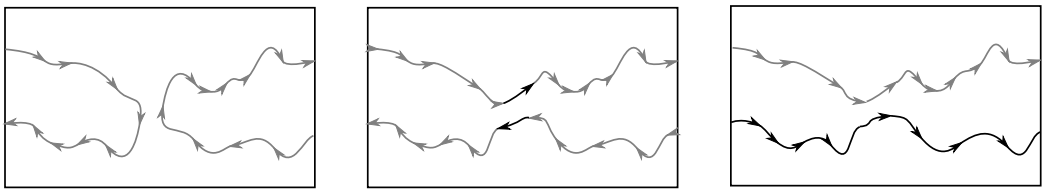

FIGURE 5.5. Conservation of winding number in the swap-only algorithm, and a partial solution provided by the swap-and-reverse algorithm.

reverses the direction of the cycle with probability 1/2. This permits winding numbers of even parity in each of the three axes. The correctness proof is unaffected, since cycle reversal is a zero-energy change. The time required to reach permutations with non-zero winding numbers, which the asymptotic correctness proof does not address, is reduced. The additional penalty in terms of CPU time consumed by cycle reversal is found to be negligible.

Other solutions exist to generate winding numbers of arbitrary parity: the band-update method of chapter 6 and the worm algorithm of chapter 7. As will be shown, they suffer from too-low acceptance rate and too-long stopping time, respectively. Therefore, the swap-and-reverse algorithm is our current best algorithm; it is used to generate all the results discussed in chapter 11. The order parameters f_S and f_W depend on winding phenomena, but the other three, $1/\xi$, f_I, and $f_{\max}$, do not; furthermore, results obtained in chapter 11 using each of the five order parameters are, for the most part, compatible. Yet, as we will see in chapter 11, f_S and f_W do not permit successful finite-size scaling.