Random Bichromatic Matchings

Nayantara Bhatnagar⋆, ‡, Dana Randall⋆, Vijay V. Vazirani†, and Eric Vigoda‡

College of Computing, Georgia Institute of Technology, Atlanta, GA 30332.
{nand,randall,vazirani,vigoda}@cc.gatech.edu.

Abstract. Given a graph with edges colored Red and Blue, we wish to sample and approximately count the number of perfect matchings with exactly k Red edges. We study a Markov chain on the space of all matchings of a graph that favors matchings with k Red edges. We show that it is rapidly mixing using non-traditional canonical paths that can backtrack, based on an algorithm for a simple combinatorial problem. We show that this chain can be used to sample dimer configurations on a 2-dimensional toroidal region with k Red edges.

1 Introduction

Counting the number of matchings in a graph is a well-studied problem in combinatorics and computer science. Counting the number of perfect matchings in a bipartite graph is equivalent to computing the permanent of a matrix with $0, 1$ entries. This problem is also of interest in statistical physics in the context of understanding the thermodynamic properties of a dimer system [3, 4]. Motivated by this application, Kastelyn showed that for planar graphs the number of perfect matchings can be computed exactly [9]. Recently Jerrum, Sinclair and Vigoda [6] gave an *fpras* (fully polynomial approximation scheme) approximating the number of perfect matchings in any bipartite graph, which is based on an *fpaus* (fully polynomial almost uniform sampler) for generating random perfect matchings.

A natural generalization of the matching problem is when the edges of the graph are colored Red or Blue:

Problem: *Given a graph $G(V, E)$, a partition $E = R \cup B$, and $k \leq \frac{|V|}{2}$, count the number of perfect matchings in G with exactly k edges in R.*

The decision version of this problem is to find a matching with exactly k Red edges. These problems have been studied in combinatorial optimization [12] as well as statistical physics [2]. There are several open questions regarding both the decision and the counting versions of this problem. For the decision version of this problem, known as *exact matchings*, Mulmuley, Vazirani and Vazirani [11] give a randomized algorithm for general graphs. A deterministic algorithm is known only when the graph is complete or complete bipartite [8, 14].

⋆ Supported in part by NSF grants CCR-0515105 and DMS-0505505. †Supported in part by NSF grants 0311541, 0220343 and CCR-051586. ‡Supported in part by NSF grant CCR-0455666.

A special case of the counting problem, of interest in statistical physics, is where G is the $\sqrt{n} \times \sqrt{n}$ 2-dimensional lattice and the horizontal edges are RED, while the vertical edges are BLUE. We wish to count the number of dimer coverings with exactly k horizontal edges, as well as solve the sampling problem. Fisher [2] gave a closed form solution for the limiting distribution (as the size of the lattice tends to infinity) of configurations in terms of the activities λ and μ of horizontal and vertical dimers, where the weight of a configuration with k horizontal edges and k' vertical edges is given by $\lambda^k \mu^{k'}$. To our knowledge, ours is the first work to address the sampling/counting problem for general graphs.

We make progress on this problem for general graphs and solve the problem in some natural special cases. Our results for general graphs are best viewed in terms of the partition function for matchings. Throughout, let $\mathcal{M}$ denote the set of all matchings of an input graph G, and $\mathcal{P}$ denote the set of perfect matchings. The standard partition function on matchings

$$Z(\lambda) = \sum_{M \in \mathcal{M}} \lambda^{|M|}$$

can be approximated for all λ by the algorithm of Jerrum and Sinclair [5]. We show that we can approximate a modified partition function which puts most weight on *(k, ℓ)-matchings*, i.e. matchings of size ℓ with exactly k RED edges.

Theorem 1. *For any $G(V, E)$ with a partition of the edges $E = R \cup B$, activities $\lambda, \mu \leq 1$, any $\ell \leq |V|/2$ and $k \leq \ell$, there is an fpras for estimating the following partition function over weighted matchings:*

$$Z_{k,\ell}(\lambda, \mu) = \sum_{M \in \mathcal{M}} \lambda^{||M \cap R| - k|} \mu^{||M| - \ell|}. \qquad (1)$$

An n-vertex graph is dense if it has minimum degree $d_{min} > n/2$. A bipartite graph with each partition of size n is dense if it has $d_{min} > n/4$.

Theorem 2. *For any dense graph $G(V, E)$, activity $\lambda \leq 1$, and $k \leq |V|/2$, there is an fpras for estimating the following partition function:*

$$\widehat{Z}_k(\lambda) = \sum_{P \in \mathcal{P}} \lambda^{||P \cap R| - k|}. \qquad (2)$$

We approximate the partition functions within a factor $(1 \pm \varepsilon)$ w.p. $\geq 1 - \delta$. The running time in each case is polynomial in $1/\lambda, 1/\mu, 1/\varepsilon, \log(1/\delta)$ and the size of the graph.

We demonstrate the significance of these results on the 2-dimensional torus $\mathbb{Z}_{m_1} \times \mathbb{Z}_{m_2}$ for even m_1, m_2, taking the horizontal edges to be RED and the vertical edges to be BLUE. In particular, we present a polynomial time algorithm for approximately sampling and counting the set of perfect matchings (or *dimer coverings*) with exactly k RED edges. We note that there are algorithms to exactly count the number of perfect matchings on the 2-d torus [9] which can be extended to bichromatic matchings. However, our proof can be extended to the *monomer-dimer* model in which we approximately sample and count (k, ℓ)-matchings of the 2-d torus, giving the first solution to this problem.

Theorem 3. *Given any torus $\mathbb{Z}_{m_1} \times \mathbb{Z}_{m_2}$ with m_1 and m_2 even, any non-negative integer $k \leq m_1 m_2/2$ and any $\ell \geq k$, there is an fpaus for generating a random (k,ℓ)-matching of the torus and an fpras for estimating the number of such matchings that run in time polynomial in m_1 and m_2.*

Theorem 1 uses a Markov chain defined on the set of all matchings of the graph which puts most weight on (k,ℓ)-matchings. We use the *canonical paths technique* to bound the convergence rate of the Markov chain. Here, these paths are non-trivial to define, in contrast to the usual matching problem where the analysis of the path congestion was the harder task.

The combinatorial fact that enables us to define our paths is as follows. Consider a graph with edges colored RED and BLUE. For any k and for all perfect matchings P, P' with exactly k RED edges, there is a polynomial length path between P and P' along almost perfect matchings, with successive matchings differing by only a few edges, such that each contains close to k RED edges. We can reduce the problem of finding such a path to a combinatorial problem about moving two points along a two-dimensional landscape in a co-ordinated manner so that the sum of their heights stays constant. The canonical path from P to P' defined in [5] starts at the matching P and alternately deletes an edge of P' and adds an edge of P' along an alternating cycle. An interesting aspect of our canonical paths is that they may backtrack along portions of the alternating cycle, for instance we might delete edges of P' that were previously added.

Our second technical contribution is proving combinatorial inequalities that allow us to approximate the number of (k,ℓ)-matchings on the torus. Kenyon, Randall and Sinclair [10] showed that the number of near perfect matchings in the d-dimensional torus is polynomially related to the number of perfect matchings, thereby yielding polynomial time algorithms for approximately counting and uniformly sampling perfect matchings. In this paper, we generalize their result to show that, on the 2-d torus, this relationship holds even when we restrict to sets of matchings with exactly k RED edges. Our result builds on ideas of Temperley [13] and Burton and Pemantle [1] for constructing augmenting paths where every horizontal and vertical segment has even length.

2 Approximately Counting Bichromatic Matchings

We outline the proof of Theorem 1 in this section; similar ideas are used to prove Theorem 2. By a standard reduction, approximating the partition function $Z_{k,\ell}(\lambda,\mu)$ can be reduced to approximate sampling [7], so we concentrate on the sampling problem and defer the details of the fpras to the full version.

To solve the sampling problem we define a Markov chain on the set of matchings $\mathcal{M}$ which puts most weight on (k,ℓ)-matchings. The same chain was used by Jerrum and Sinclair [5], with the transition probabilities defined so that the stationary distribution was uniform over all matchings.

The Markov Chain $\mathcal{T}$: The state space is $\mathcal{M}$, the set of all matchings of G. Let $\ell \leq |V|/2$, $0 \leq k \leq \ell$ and $0 < \lambda, \mu \leq 1$. Define the weight of a matching M, as $w(M) = \lambda^{|k-|M\cap R||}\mu^{|\ell-|M||}$. The transitions $M_t \rightarrow M_{t+1}$ of $\mathcal{T}$ are defined as follows.

From a matching M_t, choose a random edge $e = (u, v) \in E$.
1) If $e \in M_t$ set $M' = M_t \setminus \{e\}$.
2) If $M \in \mathcal{N}(u, v)$, (i.e. u, v are unmatched), set $M' = M_t \cup \{e\}$.
3) If for $z \neq v$, $M_t \in \mathcal{N}(u, z)$ and $(w, v) \in M_t$, set $M' = (M_t \cup \{e\}) \setminus (w, v)$. Set $M_{t+1} = M'$ with probability $\frac{1}{2}\min(1, w(M')/w(M))$, else set $M_{t+1} = M_t$.

It is straightforward to verify that the Markov chain is connected, aperiodic and reversible and has stationary distribution proportional to $w(M)$.

Intuition for the Canonical Paths

In the canonical path method for bounding the mixing time of a Markov chain, for each pair of matchings I, F, we define a path from I to F along transitions of the chain. We need to bound the *congestion* of these paths through every transition to show that the Markov chain converges quickly.

The approach of Jerrum and Sinclair [5] to obtain this bound is to focus on a specific transition T. For each pair (I, F) whose path uses the transition T, we define an "encoding" E, which is also a matching; T and E let us recover (I, F), so E can be viewed as an injective map. Then the number of (I, F) pairs whose path uses T is at most the number of matchings, which is $|\Omega|$. This is sufficient to bound the congestion for unweighted matchings. For weighted matchings, we also need to show that $w(I)w(F) \leq w(T)w(E)poly(n)$. The encoding is defined as $E = (I \cup F) \setminus (M \cup M')$ where $T = M \to M'$, so E can be viewed as the complementary matching of T with respect to (I, F).

Suppose that $\ell = |V|/2$ so that we favor perfect matchings. If I and F are perfect matchings with k RED edges, they have maximum weight. The weight of transitions and encodings along the canonical path from I to F must be comparable to the weight of I and F. Hence, both T and E need to contain close to k RED edges, and simultaneously be close to a perfect matching (i.e., have only a constant number of unmatched vertices or "holes").

Consider the perfect matchings I, F, and suppose $I \oplus F$ (the symmetric difference of I and F) consists of a single alternating cycle. The transitions of the chain allow us to easily "unwind" this alternating cycle: remove one of the edges of I on the cycle, then perform a series of shifts (moves of type 3), and then add the final edge of the cycle of F.

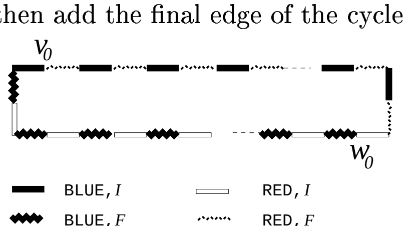

Fig. 1. An alternating cycle in $I \oplus F$

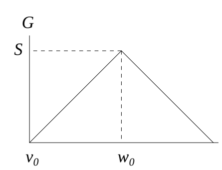

Fig. 2. Landscape for cycle

To see the difficulty, suppose, as in Figure 1, this cycle alternates RED on I and BLUE on F on one half of the cycle, and BLUE on I and RED on F on the other. Then, no matter where we start the unwinding there will be some intermediate matching with far more (or far less) RED edges than the intended

k. Notice that in this example there are *two* vertices v_0, w_0 so that if we unwind from these two points *simultaneously* then we can ensure that the number of RED edges differs from k by at most a constant. It turns out that we can always choose two such positions to begin the unwinding of the cycle. To define the unwinding, it is helpful to look at the alternating cycle together with a function representing the number of RED edges gained along the cycle.

However, the protocol for unwinding is not straightforward and we may need need to backtrack (switch edges back from F to I) from one position to continue unwinding at the other. Hence, it is not obvious whether our paths can always make progress. Additionally, the picture is more complicated when $I \oplus F$ consists of multiple cycles and paths with varying lengths and numbers of RED edges. We focus on formalizing the problem of unwinding a single alternating cycle and defer the general case to the full version.

Paired Mountain Climbing

Consider the case that $I \oplus F$ is a single alternating cycle and I and F both contain exactly k RED edges. We would like to transform the cycle from I to F so that all the intermediate matchings have close to k RED edges.

For every other vertex v on the alternating cycle, assign $-1, 0$ or $+1$ to denote the change in the number of RED edges. Thus, for $e = (u, v) \in I, e' = (v, w) \in F$, $f(v) = 1_{e' \in R} - 1_{e \in R}$, where 1 is the indicator function. Fix a start vertex on the cycle, say v_0, and a direction for unwinding the cycle. For every vertex $v_{2\ell+1}$ on the cycle, let $G(v_{2\ell+1}) = \sum_{i=0}^{\ell} f(v_{2i+1})$, where $v_0 \to v_1 \to \cdots \to v_\ell$ is the alternating path from v_0 to v_ℓ. The function G defines a "landscape", as shown in Figure 2.

It can be shown that if $|I \cap R| = |F \cap R| = k$, then there always exists a vertex v_0 so that $G(v_0) = 0$, $G(v_\ell) \geq 0$ for all ℓ, and 0 again at the last vertex. We choose a companion start vertex for v_0 which is a (global) maximum, denote this vertex as w_0. Let $S = G(v_0) + G(w_0)$. We break the alternating cycle into a pair of alternating paths, $P = \{v_0, v_1, \ldots, v_n\}$ and $Q = \{w_0, \ldots, w_m\}$, where v_n is the vertex before w_0 and w_m is the vertex before v_0.

We now start unwinding the cycle at the vertices v_0 and w_0. If unwinding from one of the positions adds a RED edge, then from the other position we need to remove a RED edge by moving forward or backwards as necessary. Thus, if at some intermediate step we are at vertices v_i and w_j, we need that $(G(v_i) - G(v_0)) + (G(w_j) - G(w_0)) = 0$, i.e. $G(v_i) + G(w_j) = S$. The mountain climbing problem is to determine a (short) trajectory from (v_0, w_0) to (v_n, w_m) so that at each intermediate step (v_i, w_j) we have $G(v_i) + G(w_j) = S$. We may need to move backwards on one path in order to move forward on the other path, and this corresponds to rewinding the cycle.

We defer the details of the canonical paths for general I, F to the full version and focus instead on the algorithm for the mountain climbing problem which has all the ideas necessary to solve the problem in general.

The Algorithm for Mountain Climbing

A *landscape* is a function $P : [n] \to \mathbb{Z}_{\geq 0}$ such that for $1 \leq i \leq n-1$, $|P(i+1) - P(i)| = 1$ (see Figure 3). For $n, m > 1$, given landscapes $P : [n] \to \mathbb{Z}_{\geq 0}$ and $Q =: [m] \to \mathbb{Z}_{\geq 0}$, we say P and Q are *S-matched* if there is an integer S s.t.

i) $P(1) + Q(m) = P(n) + Q(1) = S$
ii) $P(1) = \min_i\{P(i)\}$, $P(n) = \max_i\{P(i)\}$, $Q(1) = \max_j\{Q(j)\}$, $Q(m) = \min_j\{Q(j)\}$.

A *traversal* of S-matched landscapes P, Q is a sequence $(i_1, j_1), \cdots, (i_\ell, j_\ell)$, s.t.

i) $i_1 = 1$, $j_1 = 1$, $i_\ell = n$, $j_\ell = m$
ii) For $1 \leq k \leq \ell - 1$, $|i_{k+1} - i_k| = 1$, $|j_{k+1} - j_k| = 1$ and $P(i_k) + Q(j_k) = S$.

Lemma 1. *Let P and Q be S-matched landscapes on $[n]$ and $[m]$ respectively. Then, there exists a traversal of P and Q of length at most $O(nm)$ and it can be found in time $O(nm)$.*

Proof. The proof is by induction on $n + m$. Let $S = min + max$, where $f_1 = g_m = min$ and $f_n = g_1 = max$. Assume that the $min < max$, otherwise, the problem is trivial. Also, we use "$(1, n, 1, m)$" as shorthand for the problem of determining a traversal for P, Q. We start by showing the inductive step and conclude with the base cases.

Case I: P has a maximum or minimum at i where $1 < i < n$.

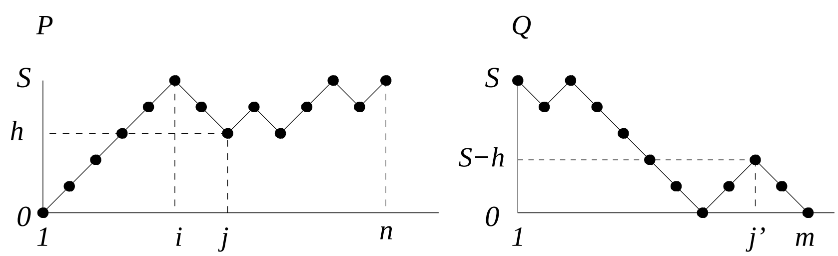

Fig. 3. Case Ia.

Case Ia.: Suppose that the first such point i is a maximum (Figure 3). Let h be the lowest value taken by P from i to n. Let j be the first point between i and n such that $P(j) = h$. Since both i and n are maxima of P, $i < j < n$. Let j' be the first point going from m to 1 (the direction here is important) such that $Q(j') = S - P(j)$. Note that it may be that $j' = 1$, but since m is a minimum of Q, $j' < m$. To find a traversal of P, Q, it is enough to concatenate the traversals for the following subproblems, in the given order: $(1, i, 1, m)$, (i, j, m, j'), (j, n, j', m). The functions on the shorter intervals take their values from P and Q. It can be verified that in each case, we obtain a problem of finding a traversal for smaller S-matched landscapes.

Case Ib.: The first such point i is a minimum (Figure 4). Let h be the maximum value taken by P from 1 to i. Let j be the first point between 1 and

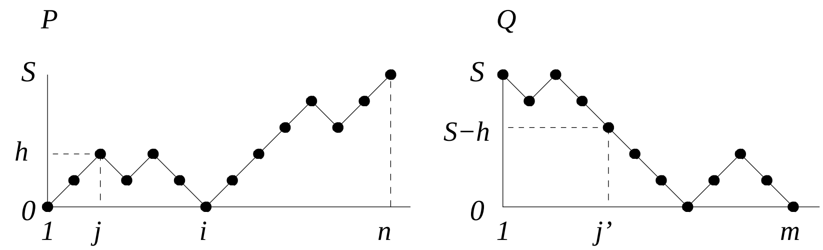

Fig. 4. Case Ib.

i such that $P(j) = h$. Since both 1 and i are minima of P, $1 < j < i$. Let j' be the first point after 1 where $Q(j') = S - h$. Since j is not a minimum of P, $j' > 1$. In this case, concatenate the traversals for the following subproblems in the given order to obtain a traversal of P, Q: $(1, j, 1, j'),\ (j, i, j', 1),\ (i, n, 1, m)$.
Case II: Q has a maximum or minimum at i where $1 < i < m$. This case follows by symmetry from Case I.
Case III: The last case is when there is a unique maximum and minimum on P and Q. We concatenate the traversals for the subproblems $(1, 2, 1, 2)$ and $(2, n, 2, m)$, both of which are smaller problems than $(1, n, 1, m)$. It can be verified that in both cases we are reduced to the problem of finding a traversal for S-matched landscapes. Note that to show this, it is crucial to use the fact that P and Q have a unique minimum and maximum.

For the base case, let $n = 2$. Then, $m = 2$ since we may assume the paths have unique maximum and minimum, otherwise we go by induction. Since the paths are S-matched, the only possibility (upto a reversal of direction) is that P is a landscape going 'up', and Q is a landscape going 'down'. The traversal is the obvious one.

Finally, we show by induction that there is a traversal of P, Q of length at most $O(nm)$ and it can be found in time $O(nm)$. If $n = 2$, the traversal is obvious and is of length $O(m)$. If $n, m > 2$, in each of the three cases above, the traversal restricted to P is obtained by traversing edge-disjoint 'sublandscapes'. Hence, the length of the traversal is at most $O(nm)$ by induction. The proof above gives an $O(nm)$ algorithm. □

Our solution to the mountain climbing problem allows us to define the canonical paths for matchings I, F. The canonical paths are defined so that every pair of successive matchings along the path is a transition of the Markov chain and the size of an intermediate matching lies between the sizes of I and F and consists of $[I_R - 5, F_R + 1]$ Red edges, where $I_R = |I \cap R|, F_R = |F \cap R|$ and $I_R \leq F_R$. Essentially, we think of the concatenation of all the paths and cycles of $I \oplus F$ as one long landscape, and apply Lemma 1 without ever unwinding more than constantly many cycles or paths at any time. By the previous argument, the paths are at most of polynomial length. With standard machinery it is now straightforward to show that the Markov chain mixes in polynomial time. The details can be found in the full version. This completes the outline of the proof of Theorem 1.

3 Bichromatic Matchings on the Torus

Let G_{m_1,m_2} be the torus $\mathbb{Z}_{m_1} \times \mathbb{Z}_{m_2}$ with horizontal edges colored RED and vertical edges BLUE. It is known that the total number of near perfect matchings is polynomially related to the number of perfect matchings [10]. We first generalize this result to relate near-perfect matchings restricted to k RED edges and perfect matchings on close to k RED edges. Our goal will be to show that the number of matchings with k red edges does not vary much as we change the size of the matching. We will show this by defining maps from one set of matchings to the other that are invertible with a small amount of additional information. This will allow us to define an *fpras* for counting the number of (k, ℓ)-matchings of the torus. Let $\mathcal{N}_k^i$ be the set of $(k, m_1m_2/2 - i)$-matchings of G_{m_1,m_2}. Let $\mathcal{P}_k = \mathcal{N}_k^0$. Let $\mathcal{N}_k(u, v)$ be the set of $(k, m_1m_2/2-1)$ matchings with holes at u and v.

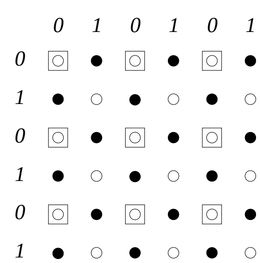

Fig. 5. The sublattice V_{00}

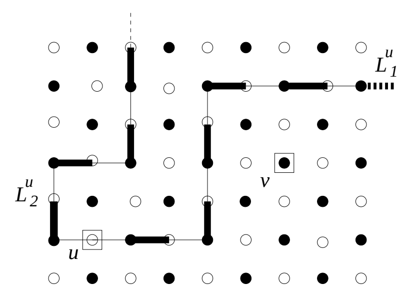

Fig. 6. Alternating paths L_1^u, L_2^u

Let m_1, m_2 be even. Let V_0 (white vertices) and V_1 (black vertices) be the even and odd sublattices of G_{m_1,m_2}. Further refining these sets, let V_{00} be the set of vertices both of whose co-ordinates are even: the sets V_{01}, V_{10} and V_{11} are defined analogously (Figure 5) . Note that if u, v are the holes of any near-perfect matching of G_{m_1,m_2}, then one of them is white while the other is black. Also, if m_1, m_2 are even, the number of RED edges in any perfect matching of G_{m_1,m_2} is even.

Theorem 4. *Let $m_1, m_2 \in \mathbb{Z}$ be even and $N = m_1m_2/2$. For $0 \le i \le N-1$ and $1 \le k \le N-1-i$, there is a map $f_i : \mathcal{N}_k^{i+1} \to \mathcal{N}_k^i \cup \mathcal{N}_{k+1}^i$ such that no matching of $\mathcal{N}_k^i \cup \mathcal{N}_{k+1}^i$ is mapped to by more than $O(N^3)$ matchings of $\mathcal{N}_k^{i+1}$.*

Proof. We first prove the theorem for $i = 0$. Let $N \in \mathcal{N}_k(u, v)$ and assume wlog that $u = (u_1, u_2)$ is in V_{00}. Define an alternating path L_1^u in N as follows: start at $z_0 = u$, and follow the unmatched RED edge to the vertex $z_1 = (u_1, u_2+1)$. Now, iteratively, if at an odd vertex z_{2i-1}, follow the unique matched edge to z_{2i} (see figure 6). ¿From each even vertex z_{2i} along the path, take the unmatched edge in the *same direction* as the edge (z_{2i-1}, z_{2i}), so each segment (in the horizontal or vertical direction) of the path after the first step has even length. Continue in this way until reaching v or revisiting a vertex, thus forming a cycle. The vertex set is finite, so one of these will occur. Define an alternating path L_2^u similarly, except, start with the edge from u to $(u_1, u_2 - 1)$. Note that every black vertex on these paths is in V_{01}, while the white vertices along the horizontal segments are in V_{00}, and those on the vertical segments are in V_{11}. Finally, define the paths K_1^u, K_2^u similarly, so that the first edges are to the vertices $(u_1 \pm 1, u_2)$. In this case, the black vertices on the path are in V_{10}, the white vertices on the vertical segments are in V_{00} and those on the horizontal segments are in V_{11}.

Let $v = (v_1, v_2) \in V_{01}$. We use these four paths to define an alternating path from u to v where the number of RED unmatched edges on the path is one more than the number of RED matched edges. Inverting along this path gives a perfect matching in $\mathcal{P}_{k+1}$. Given a perfect matching obtained in this way, we will be able to recover the near perfect matching with polynomial amount of information. We define the alternating path from u to v, by considering these cases.

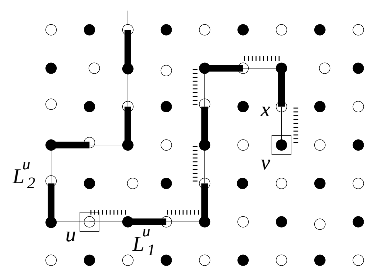

Fig. 7. L_1^u meets v

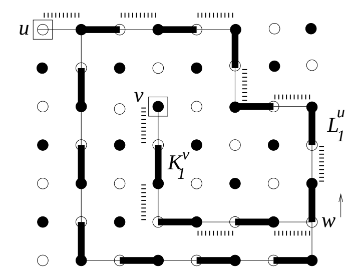

Fig. 8. The path K_1^v meets C_1

1. If one of the paths L_1^u or L_2^u reaches the vertex v before it cycles (Figure 7), then this is the alternating path. Say the path L_1^u reaches v. By construction, the number of unmatched RED edges along L_1^u is exactly one more than the number of matched RED edges, hence inverting along the path gives $P \in \mathcal{P}_{k+1}$. To invert the map, given $P \in \mathcal{P}_{k+1}$, start at v, if v is matched by a BLUE (RED) edge to the vertex x, the next unmatched edge along the path is taken to be the other BLUE (RED) edge incident with x. Continue in this way until u is reached.

2. If both paths L_1^u, L_2^u cycle without reaching v, we consider the following cases based on whether these cycles are contractible.

(a) At least one of the paths, say L_1^u , ends with a contractible cycle C_1 on the surface of the torus. It is easy to show that the interior of C_1 contains an odd number of vertices, and the number of black vertices exceeds the white vertices by 1. Hence, the interior of C_1 must contain an odd number of unmatched vertices and the unmatched vertex in the interior cannot be white: in particular, it cannot be u itself. So, v must lie in the interior of C_1. Consider the path K_1^v (see Figure 8). Since K_1^v cannot cycle in the interior of C_1 or end at an unmatched vertex, it must meet C_1. By construction, the white vertices of K_1^v are in V_{11}, hence the path meets the cycle on a vertical segment, say at the vertex w. The alternating path from u to v is defined by taking the subpath of L_1^u from u to w, and the subpath of K_1^v from v to w. The number of unmatched RED edges in the alternating path is one more than the number of matched RED edges, and so inverting along the path gives $P \in \mathcal{P}_{k+1}$ as required. Moreover, given P, and the edge incident to w in the original matching, the alternating path can be reconstructed.

(b) Both paths L_1^u, L_2^u end in non-contractible cycles on the surface of the torus. There are two possibilities, and we give a sketch of the arguments.

i) The cycles C_1 and C_2 are disjoint. This implies that the paths L_1^u and L_2^u are disjoint except at u. When a torus is cut along an incontractible simple cycle, we are left with a cylinder. If we cut along the cycles C_1 and C_2, we are left with 2 cylinders, one of which contains u and the paths L_1^u, L_2^u. The other cylinder can be shown to have an even number of vertices. Since the union of the two paths L_1^u, L_2^u is odd, the cylinder containing the paths has an odd number of vertices, and hence contains the vertex v. As before, K_1^v must hit one of the paths L_1^u or L_2^u since it cannot cycle on the cylinder.

ii) The cycles C_1, C_2 are not disjoint. In this case it can be shown that there exists a contractible cycle on the surface of the torus which can be cut out by starting at u along L_1^u, and ending at u along L_2^u (some edges may be used twice, once from above and once from below). As before, the interior contains an odd number of vertices which must be matched with each other, and hence must contain the vertex v. Since the cycle containing v is contractible, the path K_1^v must hit one of L_1^u, L_2^u.

In each case K_1^v hits the path from u on a vertical segment at a white vertex in V_{11} Given a matching in $\mathcal{P}_{k+1}, u, v$ and the vertex at which the paths from u and v meet, we can invert the map as described before.

In the case that $v \in V_{10}$, the same arguments can be made, except that we consider the paths $K_1^u, K_2^u, L_1^v, L_2^v$ instead of $L_1^u, L_2^u, K_1^v, K_2^v$ respectively. The difference is that the alternating paths constructed have one unmatched BLUE edge more than the number of matched BLUE edges along the path, so inverting edges with non-edges along the alternating path from u to v gives a matching in $\mathcal{P}_k$. This completes the proof for $i = 0$.

In the case that $i \neq 0$, suppose that $N \in \mathcal{N}_k^{i+1}$. Let u be the lexicographically first unmatched white vertex of N, and assume that $u \in V_{00}$. If one of L_1^u, L_2^u meets a black unmatched vertex v, then switching edges along the path from u to v gives a matching in $\mathcal{N}_{k+1}^i$. If not, then both L_1^u, L_2^u cycle.

Suppose L_1^u ends in a contractible cycle C_1. The interior of C_1 contains an odd number of vertices, including the vertices possibly on a segment of L_1^u starting at u. Hence, the interior contains an odd number of unmatched vertices. Since black vertices outnumber white vertices by one, the number of black unmatched vertices outnumber white unmatched vertices by one. In particular, the interior contains at least one black unmatched vertex, call it v.

Consider the paths K_1^v, K_2^v. If either one reaches a white unmatched vertex in the interior (including u), then switching edges along that path gives a matching in $\mathcal{N}_k^i$. Otherwise, if either one hits L_1^u, say at a vertex w, then we can switch edges along L_1^u from u to w, and then along K_1^v from v to w to obtain a matching either in $\mathcal{N}_{k+1}^i$ or $\mathcal{N}_k^i$ depending on whether v is in V_{01} or V_{10}. If the paths K_1^u, K_2^v do not hit a white unmatched vertex or L_1^u, they must cycle in a contractible cycle in the interior. Consider one of the paths, say K_1^v. Repeat the same argument as before, except now we consider white vertices u' in the interior of the cycle, and consider the paths $L_1^{u'}, L_2^{u'}$. Depending on whether u' is in V_{00} or V_{11} and the sublattice of v, alternating along the paths as before gives a matching either in $\mathcal{N}_k^i$ or $\mathcal{N}_{k+1}^i$. We can repeat this argument until we obtain an alternating path between a black and a white vertex, or, the interior of some cycle created by a vertex contains only one unmatched vertex. Since the single unmatched vertex cannot be the same as the vertex from which the cycle was created, this case can be solved in the same manner as the case when $i = 0$.

The remaining case, when L_1^u, L_2^u end in incontractible cycles, is similar to Case 2 above. □

Corollary 1. *Let m_1, m_2 be even, $N = m_1 m_2/2$. There is an algorithm to estimate the partition function $\widehat{Z}_k$ given in Equation (2) for every $\lambda \leq 1$ and k to within $(1 \pm \varepsilon)$ w.p. $\geq 1 - \delta$ in time polynomial in $N, \lambda, 1/\varepsilon$ and $\log(1/\delta)$.*

We can use similar arguments to relate the number of perfect matchings with k or $k+2$ RED edges.

Theorem 5. *Let m, n be even, $N = mn/2$. For every $0 \leq k \leq N-2$ even, $|\mathcal{P}_{k+2}|/p(N) \leq |\mathcal{P}_k| \leq p(N)|\mathcal{P}_{k+2}|$, where p is a polynomial.*

Proof. It suffices to show the upper bound for all k since the lower bound follows by switching the colors.

We construct a map from $\mathcal{P}_k$ to $\mathcal{P}_{k+2}$ as follows. Let $P \in \mathcal{P}_k$. Delete any vertical edge (u, v). Since $k \leq N-2$, there must be such an edge. Consider the paths L_1^u, L_2^u in $P \setminus (u, v)$. By parity, neither can reach v, and hence they must cycle on the surface of the torus. Since u is adjacent to v on the torus, neither path can end in a contractible cycle containing v in the interior. Hence both L_1^u, L_2^u end in incontractible cycles. By the arguments of Case 2 of the previous theorem, the path L_1^v must hit one of the paths L_1^u, L_2^u at a white vertex $w \in V_{11}$, i.e., on a vertical segment. Then, switching along the alternating path from u to v through w as before, we gain two RED edges, giving a matching in $\mathcal{P}_{k+2}$. The mapping is invertible given the vertex w and the vertices u, v, hence $|\mathcal{P}_k| \leq O(N^3)|\mathcal{P}_{k+2}|$. □

Using this Theorem and the estimator given by Corollary 1, we obtain an estimator for the set of perfect matchings of the torus with exactly k RED edges. The proof follows from standard arguments.

Theorem 6. *There is an algorithm to estimate $|\mathcal{P}_k|$ to within $1 \pm \varepsilon$ for every $0 < \varepsilon < 1$ with probability $\geq 1 - \delta$ in time polynomial in $N, 1/\varepsilon$ and $\log(1/\delta)$.*

These results can be generalized to approximating the size of the set of (k, ℓ)-matchings for any ℓ. By Theorem 1, we can approximate the partition function $\hat{Z}_{k,\ell}$ given in Equation (1) for every $\lambda, \mu \leq 1$ and $0 \leq k \leq \ell \leq n$. This estimator, together with the relations among sets of restricted matchings of arbitrary size (stated below) and the theorem of Kenyon, Randall and Sinclair [10] that the sizes of the sets $\mathcal{N}^i$ and $\mathcal{N}^{i+1}$ are polynomially related, gives an approximate counter for sets of restricted matchings of any size.

Theorem 7. *Let m_1, m_2 be even, $N = m_1 m_2/2$. For every $1 \leq i \leq N-1$, and $0 \leq k \leq N-i-1$, then for some polynomial p, $|\mathcal{N}_{k+1}^i|/p(N) \leq |\mathcal{N}_k^i| \leq p(N)|\mathcal{N}_{k+1}^i|$.*

The proof follows by constructing alternating paths as in Theorem 4.

Corollary 2. *There is an algorithm to estimate $|\mathcal{N}_k^i|$ to within $1 \pm \varepsilon$ for every $0 < \varepsilon < 1$ with probability $\geq 1-\delta$ in time polynomial in $N, 1/\varepsilon$ and $\log(1/\delta)$.*

Acknowledgements: Thanks to a referee for suggesting a simplification of the statement of Lemma 1.

References

1. R. Burton and R. Pemantle. Local characteristics, entropy and limit theorems for spanning trees and domino tilings via transfer-impedances. *Annals of Probability* **21(3)**(1993), pp. 1329-1371.
2. M.E. Fisher. Statistical Mechanics of Dimers on a Plane Lattice. *Phys. Rev.* **124** (1961), pp.1664-1672.
3. R.H. Fowler and G.S. Rushbrooke. Statistical theory of perfect solutions. *Transactions of the Faraday Society* **33** (1937), pp. 1272-1294.
4. O.J. Heilmann and E.H.Leib. Theory of monomer-dimer systems. *Communications in Mathematical Physics* **25** (1972), pp 190-232.
5. M. Jerrum and A. Sinclair. Approximating the Permanent. *SIAM Journal of Computing* **18** (1989), pp. 1149-1178.
6. M. Jerrum, A. Sinclair, and E. Vigoda. A Polynomial-Time Approximation Algorithm for the permanent of a matrix with non-negative entries. *Journal of the ACM* **51(4)** 2004, pp. 671-697.
7. M. Jerrum, L. Valiant, and V. Vazirani. Random generation of combinatorial structures from a uniform distribution. *Theoretical Computer Science* **43** 1986, pp.169-188.
8. A.V. Karzanov. Maximum Matching of given weight in complete and complete bipartite graphs. *Kibernetica: 7-11, English translation in CYBNAW* **23(1)** (1987) pp. 8-13.
9. P. W. Kasteleyn. The statistics of dimers on a lattice,I. The number of dimer arrangements on a quadratic lattice. *Physica* **27** (1961) pp. 1209-1225.
10. C. Kenyon, D. Randall, and A. Sinclair. Approximating the number of monomer-dimer coverings of a lattice. *Journal of Statistical Physics* **83** (1996) pp. 637-659.
11. K. Mulmuley, U.V. Vazirani, and V. V. Vazirani. Matching is as easy as matrix inversion. *Combinatorica* **7(1)** (1987), pp.105-113.
12. C. Papadimitriou. Polytopes and Complexity. *Progress in Combinatorial Optimization* (W.R. Pulleyblank ed., Academic Press Canada, Don Mills, 1984) pp. 295-305.
13. H. N. V. Temperley. In Combinatorics, Proceedings of the British Combinatorial Conference 1973, London Mathematical Society Lecture Notes Series No.13. 1974, pp. 202-204.
14. T. Yi, K. G. Murty, C. Spera. Matchings in colored bipartite networks. *Discrete Applied Mathematics* **121(1-3)** (2002), pp. 261-277.