Enumeration Algorithm for Lattice Model

Seungsang Oh
Korea University

International Workshop on Spatial Graphs 2016
Waseda University, August 5, 2016
Contents

1. State Matrix Recursion Algorithm
2. Monomer-Dimer Problem (best application)
3. Multiple Self-Avoiding Polygon Enumeration
4. Further Applications in Lattice Statistics
Contents

1 State Matrix Recursion Algorithm

2 Monomer-Dimer Problem (best application)

3 Multiple Self-Avoiding Polygon Enumeration

4 Further Applications in Lattice Statistics
State matrix recursion algorithm enumerates 2-dimensional lattice models such as

- Monomer-dimer coverings
- Multiple self-avoiding walks and polygons
- Independent vertex sets
- Quantum knot mosaics

These are famous problems in Combinatorics and Statistical Mechanics studied by topologists, combinatorialists and physicists alike.
State matrix recursion algorithm is divided into three stages:

- Stage 1. Conversion to appropriate mosaics
- Stage 2. State matrix recursion formula
- Stage 3. State matrix analyzing

During this talk, the algorithm will be briefly demonstrated by solving the Monomer-Dimer Problem.
1 State Matrix Recursion Algorithm

2 Monomer-Dimer Problem (best application)

3 Multiple Self-Avoiding Polygon Enumeration

4 Further Applications in Lattice Statistics
Monomer-dimer coverings

Monomer-dimer covering in $m \times n$ rectangle on the square lattice $\mathbb{Z}_{m \times n}$

Generating function

$$D_{m \times n}(z) = \sum k(t) z^t$$

where $k(t)$ is the number of monomer-dimer coverings with $t$ monomers.

- $D_{m \times n}(1)$ is the number of monomer-dimer coverings.
- $D_{m \times n}(0)$ is the number of pure dimer coverings (i.e., no monomers).
Breakthrough results

[Kasteleyn and Temperley-Fisher 1961]

Pure dimer problem for even $mn$

$$\prod_{j=1}^{m} \prod_{k=1}^{n} \sqrt{2 \cos \left( \frac{\pi j}{m+1} \right) + 2i \cos \left( \frac{\pi k}{n+1} \right)}$$

[Tzeng-Wu 2003]

Single boundary monomer problem for odd $mn$
(it has a fixed single monomer on the boundary)

$$\prod_{j=1}^{\frac{m-1}{2}} \prod_{k=1}^{\frac{n-1}{2}} \left[ 4 \cos^2 \left( \frac{\pi j}{m+1} \right) + 4 \cos^2 \left( \frac{\pi k}{n+1} \right) \right]$$

Question: How about if we allow many monomers? Generating function?
Monomer-Dimer Theorem

**Theorem**

\[ D_{m \times n}(z) = (1, 1)\text{-entry of } (A_m)^n \]

where \( A_m \) is a \( 2^m \times 2^m \) matrix defined by the recurrence relation

\[
A_k = \begin{bmatrix}
z A_{k-1} + \begin{bmatrix}
A_{k-2} & \mathbb{O}_{k-2} \\
\mathbb{O}_{k-2} & \mathbb{O}_{k-2}
\end{bmatrix}
A_{k-1} \\
A_{k-1} & \mathbb{O}_{k-1}
\end{bmatrix}
\]

starting with \( A_0 = \begin{bmatrix} 1 \end{bmatrix} \) and \( A_1 = \begin{bmatrix} z & 1 \\
1 & 0
\end{bmatrix} \) where \( \mathbb{O}_k \) is the \( 2^k \times 2^k \) zero-matrix.

Note that it is not a closed form solution, but a sparse recurrence algorithm.
### Exact enumeration

<table>
<thead>
<tr>
<th>( n )</th>
<th>( D_{n \times n}(1) )</th>
<th>( (D_{n \times n}(1))^{\frac{1}{n^2}} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1.000</td>
</tr>
<tr>
<td>2</td>
<td>7</td>
<td>1.627</td>
</tr>
<tr>
<td>3</td>
<td>131</td>
<td>1.719</td>
</tr>
<tr>
<td>4</td>
<td>10012</td>
<td>1.778</td>
</tr>
<tr>
<td>5</td>
<td>2810694</td>
<td>1.811</td>
</tr>
<tr>
<td>6</td>
<td>2989126727</td>
<td>1.833</td>
</tr>
<tr>
<td>7</td>
<td>11945257052321</td>
<td>1.849</td>
</tr>
<tr>
<td>8</td>
<td>179788343101980135</td>
<td>1.860</td>
</tr>
<tr>
<td>9</td>
<td>10185111919160666118608</td>
<td>1.869</td>
</tr>
<tr>
<td>10</td>
<td>2172138783673094193937750015</td>
<td>1.877</td>
</tr>
<tr>
<td>11</td>
<td>1743829823240164494694386437970640</td>
<td>1.882</td>
</tr>
<tr>
<td>12</td>
<td>5270137993816086266962874395450234534887</td>
<td>1.887</td>
</tr>
<tr>
<td>13</td>
<td>59956919824257750508655631107474672284499736089</td>
<td>1.891</td>
</tr>
</tbody>
</table>
Stage 1. Conversion to monomer-dimer mosaics

Adjacency Rule: Attaching edges of adjacent tiles have the same letter.

Boundary state requirement: All boundary edges are labeled with letter a.
Stage 2. State matrix recursion formula

State polynomial: Twelve suitably adjacent $3 \times 3$-mosaics associated with $b$-state $aba$, $t$-state $bab$ and the trivial $l$- and $r$-states $aaa$ to produce the associated state polynomial $1 + 5z^2 + 5z^4 + z^6$. 
State matrix $A_{m\times n}$ for the set of suitably adjacent $m \times n$-mosaics is a $2^m \times 2^m$ matrix $(a_{ij})$ where $a_{ij}$ is the state polynomial associated to $i$-th $b$-state, $j$-th $t$-state, and the trivial $l$- and $r$-states. (Trivial state condition is needed for the boundary state requirement)

We arrange $2^m$ states of length $m$ in the lexicographic order.

For example, $(3,6)$-entry of $A_{3\times3}$ is $a_{3,6} = 1 + 5z^2 + 5z^4 + z^6$. 

![Diagram showing state matrix and state polynomials]
Recursion strategy to find the state matrix $A_{m \times n}$.

1. Find the **starting state matrices** $A_1$ and $B_1$ for $1 \times 1$-mosaics.

2. Find the **bar state matrices** $A_k$ and $B_k$ for suitably adjacent $k \times 1$-mosaics (or bar mosaics) by attaching a mosaic tile recursively on the right side.

3. Find the **state matrix** $A_{m \times k}$ for suitably adjacent $m \times k$-mosaics by attaching a bar mosaic of length $m$ on the top side.
Summary

First, we get the recursive relation from the bar state matrix recursion lemma

\[ A_k = \begin{bmatrix} z A_{k-1} + B_{k-1} & A_{k-1} \\ A_{k-1} & \varnothing_{k-1} \end{bmatrix} \] and \[ B_k = \begin{bmatrix} A_{k-1} & \varnothing_{k-1} \\ \varnothing_{k-1} & \varnothing_{k-1} \end{bmatrix} \]

starting with \( A_0 = \begin{bmatrix} 1 \end{bmatrix} \) and \( B_0 = \begin{bmatrix} 0 \end{bmatrix} \).

Then, we have the state matrix from the state matrix multiplication lemma

\[ A_{m \times n} = (A_m)^n. \]
Stage 3. State matrix analyzing

Monomer-dimer generating function w.r.t. the number of monomers

\[ D_{m \times n}(z) = (1,1)\text{-entry of } A_{m \times n}. \]
Monomer-Dimer Theorem

**Theorem**

\[ D_{m \times n}(z) = (1, 1)\text{-entry of } (A_m)^n \]

where \( A_m \) is a \( 2^m \times 2^m \) matrix defined by the recurrence relation

\[
A_k = \begin{bmatrix}
z A_{k-1} + \begin{bmatrix}
A_{k-2} & \mathbb{O}_{k-2} \\
\mathbb{O}_{k-2} & \mathbb{O}_{k-2}
\end{bmatrix} & A_{k-1} \\
A_{k-1} & \mathbb{O}_{k-1}
\end{bmatrix}
\]

starting with \( A_0 = \begin{bmatrix} 1 \end{bmatrix} \) and \( A_1 = \begin{bmatrix} z & 1 \\ 1 & 0 \end{bmatrix} \) where \( \mathbb{O}_k \) is the \( 2^k \times 2^k \) zero-matrix.
Contents

1 State Matrix Recursion Algorithm

2 Monomer-Dimer Problem (best application)

3 Multiple Self-Avoiding Polygon Enumeration

4 Further Applications in Lattice Statistics
Self-avoiding polygons

Self-avoiding polygon (SAP) on the square lattice $\mathbb{Z}^2$

Finding $p_n$ is the central unsolved problem during last 70 years in Combinatorics and Statistical Mechanics.

There are many numerical datas, but few mathematically proved results.

$p_n =$ number of SAPs of length $n$
up to translations
**Breakthrough results**

[Hammersley 1957]
The limit $\mu = \lim (p_n)^{\frac{1}{n}}$ exists.

$$\mu = 2.638158530323 \pm 2 \times 10^{-12} \text{ : best estimate on } \mathbb{Z}^2 \text{ during 50 years.}$$

$\mu = \sqrt{2} + \sqrt{2}$ on the hexagonal lattice $\mathbb{H}^2$ (easier than on $\mathbb{Z}^2$).

Nobody expects that there will be a closed form of $p_n$. 
Multiple self-avoiding polygons

Multiple self-avoiding polygon (MSAP) in $\mathbb{Z}_{m \times n}$

$p_{m \times n} = \text{number of MSAPs in } \mathbb{Z}_{m \times n} \text{ (not up to translations)}$

Theorem

$p_{m \times n} = (1, 1)$-entry of $(A_m)^n - 1$

where the $2^m \times 2^m$ matrix $A_m$ is defined by

$A_{k+1} = \begin{bmatrix} A_k & B_k \\ B_k & A_k \end{bmatrix}$ and $B_{k+1} = \begin{bmatrix} B_k & A_k \\ A_k & 0 \end{bmatrix}$

starting with $A_0 = \begin{bmatrix} 1 \end{bmatrix}$ and $B_0 = \begin{bmatrix} 0 \end{bmatrix}$.
MSAPs in the 1-slab square lattice

Multiple self-avoiding polygons (links) in the 1-slab square lattice $\mathbb{Z}_{m\times n\times 2}$
(2 layers of the planes)
Conversion to 1-slab MSAP mosaics by using 65 mosaic tiles
MSAP enumeration in $\mathbb{Z}_{m \times n \times 2}$

**Theorem**

The number of MSAPs in the 1-slab square lattice $\mathbb{Z}_{m \times n \times 2}$ is

$$(1, 1)\text{-entry of } (A_m)^n - 1$$

where the $4^m \times 4^m$ matrix $A_m$ is defined by

$$A_{k+1} = \begin{bmatrix} A_k + D_k & B_k + C_k & B_k + C_k & A_k + D_k \\ B_k + C_k & A_k & A_k + D_k & C_k \\ B_k + C_k & A_k + D_k & A_k & B_k \\ A_k + D_k & C_k & B_k & A_k \end{bmatrix}, \quad B_{k+1} = \begin{bmatrix} B_k + C_k & A_k & A_k + D_k & C_k \\ A_k & \bigcirc_k & C_k & \bigcirc_k \\ A_k + D_k & C_k & B_k & A_k \\ C_k & \bigcirc_k & A_k & \bigcirc_k \end{bmatrix},$$

$$C_{k+1} = \begin{bmatrix} B_k + C_k & A_k + D_k & A_k & B_k \\ A_k + D_k & C_k & B_k & A_k \\ A_k & \bigcirc_k & \bigcirc_k & \bigcirc_k \\ B_k & \bigcirc_k & \bigcirc_k & \bigcirc_k \end{bmatrix} \quad \text{and} \quad D_{k+1} = \begin{bmatrix} A_k + D_k & C_k & B_k & A_k \\ C_k & \bigcirc_k & A_k & \bigcirc_k \\ B_k & A_k & \bigcirc_k & \bigcirc_k \\ A_k & \bigcirc_k & \bigcirc_k & \bigcirc_k \end{bmatrix},$$

starting with $A_0 = [1]$ and $B_0 = C_0 = D_0 = [0]$.

- The number of MSAPs in $\mathbb{Z}_{7 \times 60 \times 2}$ is $5.345706 \cdots \times 10^{261}$. 
Links in the 3-dimensional cubic lattice $\mathbb{Z}_{l \times m \times n}$
(not up to translations and ambient isotopies)
### Contents

1. State Matrix Recursion Algorithm
2. Monomer-Dimer Problem (best application)
3. Multiple Self-Avoiding Polygon Enumeration
4. Further Applications in Lattice Statistics
Different regular lattices

Hexagonal (honeycomb) lattice $\mathbb{H}_{m \times n}$ (MSAP model)
Different regular lattices

Triangular lattice $T_{m \times n}$ (Monomer-dimer model)
Different regular lattices

1-slab square lattice $\mathbb{Z}_{m \times n \times 2}$ (Multiple self-avoiding polygon (link) model)
Polymer model

Monomer-dimer-trimer-tetramer covering

tetramer

trimer
dimer

monomer
Polyomino model

Monomino-domino-tromino tiling
Independent vertex model

Independent vertex sets

- Independent vertex set (Hard Square Problem)
- Independent vertex set with 2-nb exclusion
- Independent vertex set with 3-nb exclusion
Quantum knot mosaic

Quantum knot mosaic

with 11 knot mosaic tiles as follows
Squared rectangle model

Tiling a rectangle by squares with various integer sizes
Tetris model

Tetris configuration by 7 tetrominoes
Thank you!