Computing the number of h-edge spanning forests in complete bipartite graphs

Rebecca J Stones

Discrete Mathematics and Theoretical Computer Science DMTCS vol. **16:1**, 2014, 313–326

Computing the number of h-edge spanning forests in complete bipartite graphs

Rebecca J. Stones*

School of Mathematical Sciences, Monash University, Australia
Department of Mathematics and Statistics, Dalhousie University, Canada

received 6th Oct. 2012, *revised 28th Feb. 2014*, *accepted 25th Apr. 2014*.

Let $f_{m,n,h}$ be the number of spanning forests with h edges in the complete bipartite graph $K_{m,n}$. Kirchhoff's Matrix Tree Theorem implies $f_{m,n,m+n-1} = m^{n-1}n^{m-1}$ when $m \geq 1$ and $n \geq 1$, since $f_{m,n,m+n-1}$ is the number of spanning trees in $K_{m,n}$. In this paper, we give an algorithm for computing $f_{m,n,h}$ for general m, n, h. We implement this algorithm and use it to compute all non-zero $f_{m,n,h}$ when $m \leq 50$ and $n \leq 50$ in under 2 days.

MSC 2010: 68R10, 05C85, 05C30, 05C38, 05C05

Keywords: complete bipartite graph, spanning forest

1 Introduction

A consequence of Kirchhoff's Matrix Tree Theorem is that the number of spanning trees in the complete bipartite graph $K_{m,n}$ is $m^{n-1}n^{m-1}$ when $m \geq 1$ and $n \geq 1$. In this paper, we consider a generalisation of this counting problem: the number $f_{m,n,h}$ of h-edge spanning forests in $K_{m,n}$. The number of spanning trees in $K_{m,n}$ is $f_{m,n,m+n-1}$ when $m \geq 1$ and $n \geq 1$, since spanning trees of $K_{m,n}$ have $m+n-1$ edges.

The numbers $f_{m,n,h}$ arise in relation to the Tutte polynomial of $K_{m,n}$. For a general graph $G = (V, E)$, the Tutte polynomial is

$$T_G(x, y) = \sum_{A \subseteq E} (x-1)^{k(A)-k(E)}(y-1)^{k(A)+|A|-|V|}$$

where $k(A)$ is the number of connected components in the graph (V, A). Importantly, for the Tutte polynomial, we take $0^0 = 1$. Thus, the number of spanning forests of G is given by

$$\begin{aligned} T_G(2,1) &= \sum_{A \subseteq E} 0^{k(A)+|A|-|V|} \\ &= \#\{A \subseteq E : k(A) + |A| = |V|\} \qquad (1) \\ &= \text{number of spanning forests of } G, \end{aligned}$$

*Email: rebecca.stones82@gmail.com. Stones supported by NSFC grant 61170301. Stones would also like to thank AARMS for their support.

since $k(A)+|A|=|V|$ if and only if A is a spanning forest.

In the present paper, we are interested in the case of when $G=K_{m,n}$, and will look for h-edge spanning forests. In this situation,

$$T_{K_{m,n}}(2,1)=\sum_{h\geq 0} f_{m,n,h}.$$

For general graphs $G=(V,E)$, Myrvold [10] gave an algorithm for computing the number of k-component spanning forests of G. Bjöklund et al. [4] described an algorithm that could compute $T_G(2,1)$ in time $2^{|V|}|V|^{O(1)}$ by using Kirchhoff's Matrix Tree Theorem for each of the $2^{|V|}$ subsets of V, then combining the results together with Inclusion-Exclusion. Other relevant work for the enumeration of spanning forests includes [9]. Porter [11] gave an algorithm for generating the spanning trees of $K_{m,n}$. Farr and McDiarmid [5] showed that computing the number of circuits in G is #P-complete[(i)].

Bounds for the number of spanning forests in graphs were given by Teranishi [13], from which we can deduce

$$T_{K_{m,n}}(2,1)\geq \sum_{k\geq 0}\left(\frac{\min(m,n)}{2}\right)^{m+n-k}\binom{m+n}{k}$$

for all $m\geq 0$ and $n\geq 0$. Jin and Liu [8] gave a simple formula for the number of rooted spanning forests in $K_{m,n}$ (see also [7] and [12]).

This paper instead heads in a different direction to previous work: we derive algorithms for enumerating h-edge spanning forests in $K_{m,n}$.

2 Basic results

To be clear, all forests in this paper will be labelled. Let $\mathbb{N}=\{0,1,2,\ldots\}$. The following lemma gives the boundary conditions on $f_{m,n,h}$.

Lemma 1 *Suppose* $m,n,h\in\mathbb{N}$. *Then*

- $f_{m,n,0}=1$,
- *if* $h\geq 1$, *then* $f_{0,n,h}=f_{m,0,h}=0$,
- *if* $m\geq 1$ *and* $n\geq 1$, *then* $f_{m,n,h}>0$ *if and only if* $0\leq h\leq m+n-1$,
- *if* $m\geq 1$ *and* $n\geq 1$, *then* $f_{m,n,m+n-1}=m^{n-1}n^{m-1}$.

Proof: If $h=0$, then there is exactly one subgraph of $K_{m,n}$ with no edges (and it is a spanning forest), so $f_{m,n,0}=1$. If $m=0$ or $n=0$, then $K_{m,n}$ has no edges, and thus we obtain the second bulleted item.

Now assume $m\geq 1$ and $n\geq 1$. Since $K_{m,n}$ is connected, it has a spanning tree, which must have exactly $m+n-1$ edges. By deleting edges from this spanning tree, we find h-edge forests in $K_{m,n}$ for all $0\leq h\leq m+n-1$. Hence $f_{m,n,h}>0$ when $0\leq h\leq m+n-1$.

Now assume A is an h-subset of the edges in $K_{m,n}$, where $m\geq 1$ and $n\geq 1$ and $h\geq m+n$. To be a spanning forest, we need $k(A)+h=m+n$ as per (1), but if $h\geq m+n$, then $k(A)\leq 0$, giving a contradiction.

(i) #P is the set of counting problems which ask for the number of "yes" instances for decision problems in NP. A #P problem is in #P-complete whenever any other problem in #P can be reduced to it by a polynomial-time counting reduction.

m	n	$h=0$	1	2	3	4	5	6	7	8	9	10	11
1	1	1	1										
1	2	1	2	1									
1	3	1	3	3	1								
1	4	1	4	6	4	1							
1	5	1	5	10	10	5	1						
1	6	1	6	15	20	15	6	1					
2	2	1	4	6	4								
2	3	1	6	15	20	12							
2	4	1	8	28	56	64	32						
2	5	1	10	45	120	200	192	80					
2	6	1	12	66	220	480	672	544	192				
3	3	1	9	36	84	117	81						
3	4	1	12	66	220	477	648	432					
3	5	1	15	105	455	1335	2673	3375	2025				
3	6	1	18	153	816	3015	7938	14499	16524	8748			
4	4	1	16	120	560	1784	3936	5632	4096				
4	5	1	20	190	1140	4785	14544	31520	44800	32000			
4	6	1	24	276	2024	10536	40704	117376	244224	331776	221184		
5	5	1	25	300	2300	12550	51030	155900	347500	515625	390625		
5	6	1	30	435	4060	27255	138606	544525	1641000	3645000	5400000	4050000	
6	6	1	36	630	7140	58680	369792	1834992	7210080	22083840	50388480	77262336	60466176

Fig. 1: Small non-zero values of $f_{m,n,h}$.

The fourth bulleted item is from Kirchhoff's Matrix Tree Theorem (already mentioned). □

The following lemma is an important (but basic) consequence of the symmetry of the problem.

Lemma 2 *For all $m, n, h \in \mathbb{N}$, we have $f_{m,n,h} = f_{n,m,h}$.*

In Figure 1 we list the non-zero values of $f_{m,n,h}$ when $0 \le m \le n \le 6$. These values were generated using a straightforward backtracking algorithm.

3 Combinatorial equivalence

In this section, we will describe some combinatorial objects that are equivalent to h-edge spanning forests of $K_{m,n}$.

Let M be an $m \times n$ $(0,1)$-matrix. We define a *cycle* in M to be a set $\{e_1, e_2, \ldots, e_{2t}\}$ of entries of M, such that:

- each e_i contains the symbol 1, and
- for $i \in \{1, 2, \ldots, t\}$, we have that (a) e_{2i-1} belongs to the same row as e_{2i} and (b) e_{2i} belongs to the same column as e_{2i+1} (where we take $e_{2t+1} = e_1$).

For example, the matrix

$$\begin{array}{|ccccc|}
\hline
0 & 1 & 0 & 1 & 1 \\
0 & 0 & 1 & 0 & 1 \\
1 & 0 & 0 & 0 & 1 \\
1 & 1 & 0 & 0 & 0 \\
\hline
\end{array}$$

has a cycle (consisting of the circled 1's), whereas the matrices

$$\boxed{\begin{matrix} 0 & 1 & 1 \\ 0 & 0 & 0 \end{matrix}} \quad \text{and} \quad \boxed{\begin{matrix} 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{matrix}}$$

do not have cycles.

Let $A_{m,n,h}$ be the set of $(0,1)$-matrices with exactly h elements equal to 1. Let $p_{m,n,h}$ be the probability that an element of $A_{m,n,h}$ chosen uniformly at random contains a cycle. An anonymous user of `math.stackexchange.com` asked(ii) for a formula for $p_{m,n,h}$. In fact, this was the original motivation for the author to study this problem.

Lemma 3 *For all $m, n, h \in \mathbb{N}$, we have $f_{m,n,h}$ is the number of $m \times n$ $(0,1)$-matrices with h 1's that do not contain a cycle, and hence*

$$f_{m,n,h} = (1 - p_{m,n,h})\binom{mn}{h}.$$

Proof: A matrix $M \in A_{m,n,h}$ can be interpreted as the biadjacency matrix(iii) of an h-edge subgraph $G = G(M)$ of $K_{m,n}$. A cycle in $M \in A_{m,n,h}$ corresponds to a cycle in G. Hence

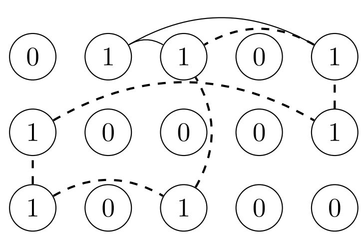
$$\begin{aligned} f_{m,n,h} &= (1 - p_{m,n,h})\,|A_{m,n,h}| \\ &= (1 - p_{m,n,h})\binom{mn}{h}. \end{aligned}$$

$\square$

A problem related to the $(0,1)$-matrix problem above comes from the study of $(0,1)$-matrices that do not have a submatrix which is the incidence matrix of any cycle of length at least 3; these are called "totally balanced matrices" (see e.g. [2]).

Another interpretation of matrices in $A_{m,n,h}$ is as induced subgraphs of $K_m \Box K_n$, where $\Box$ represents the Cartesian product of graphs. The graph $K_m \Box K_n$ is sometimes called the "rook's graph", since the edges represent the legal moves of a rook on an $m \times n$ chess board. The vertices in $K_m \Box K_n$ are $\{(i,j) : 1 \leq i \leq m \text{ and } 1 \leq j \leq n\}$ and there is an edge between distinct vertices (i,j) and (i',j') whenever $i = i'$ or $j = j'$. The graph $K_m \Box K_n$ is also the line graph of $K_{m,n}$.

There is a bijection between induced subgraphs $H = H(M)$ of $K_m \Box K_n$ and $(0,1)$-matrices $M = (m_{ij}) \in A_{m,n,h}$: we include the vertex (i,j) in H if and only if $m_{ij} = 1$. For example, if we ignore the vertices marked 0 in



(ii) Full URL: `http://math.stackexchange.com/q/24800`

(iii) The biadjacency matrix of a bipartite graph G on the vertex set $\{u_i\}_{1 \leq i \leq m} \cup \{v_j\}_{1 \leq j \leq n}$ is the $m \times n$ $(0,1)$-matrix $M = (m_{ij})$ with $m_{ij} = 1$ if and only if $u_i v_j$ is an edge in G.

we obtain an induced subgraph $H(M)$ of $K_3 \square K_5$, corresponding to the matrix

$$M = \boxed{\begin{array}{ccccc} 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \end{array}}$$

in $A_{3,5,6}$. We see that cycles in M map from cycles C_k in H for some $k \in \{4, 6, 8, \ldots\}$ (an example of a 6-cycle is highlighted). There are other cycles in H (e.g. C_3 in the above example; in fact, if M above had a row of 1's, then H would have a K_5 subgraph). However, *induced* cycles C_k in $H(M)$ of length $k \in \{4, 6, 8, \ldots\}$ are in one-to-one correspondence with cycles in M via the above bijection.

Lemma 4 *For all $m, n, h \in \mathbb{N}$, we have that $f_{m,n,h}$ is the number of h-vertex induced subgraphs of $K_m \square K_n$ that do not contain an induced C_k for any $k \in \{4, 6, 8, \ldots\}$.*

4 Simplifying the equation

4.1 A formula for $f_{m,n,h}$

We will look at two related ways of simplifying $f_{m,n,h}$. Let $K_{m,n}$ have the vertex bipartition $M \cup N$, where $M = \{u_1, u_2, \ldots, u_m\}$ and $N = \{v_1, v_2, \ldots, v_n\}$. Let $B_{m,n,h}$ be the set of h-edge spanning forests of $K_{m,n}$. Hence $f_{m,n,h} = |B_{m,n,h}|$. Define

$$W_{m,n,h} = \{G \in B_{m,n,h} : G \text{ has no isolated vertices}\}$$

and $w_{m,n,h} = |W_{m,n,h}|$. We observe the following:

- Given any spanning forest in $B_{m,n,h}$ we can delete the isolated vertices to obtain a spanning forest in $W_{i,j,h}$ for some $i \leq m$ and $j \leq n$.
- Conversely, given any spanning forest in $W_{i,j,h}$, we can add isolated vertices to it in $\binom{m}{m-i}\binom{n}{n-j}$ ways to obtain a spanning forest in $B_{m,n,h}$.

In both of the operations above, we need to relabel the vertices, but we preserve the order of the indices of the non-isolated vertices within M and N.

Hence

$$f_{m,n,h} = \sum_{i=0}^{m} \sum_{j=0}^{n} \binom{m}{i} \binom{n}{j} w_{i,j,h}. \tag{2}$$

This formula eliminates the need to further account for isolated vertices.

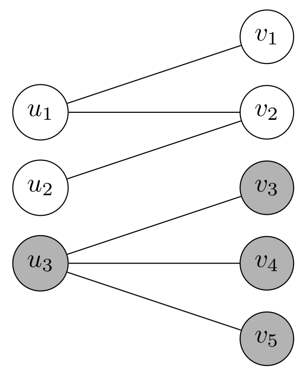
We can decompose any $G \in W_{m,n,h}$ into disjoint components, each of which belong to $W_{i,j,k}$ for some $i \leq m$ and $j \leq n$ and $k \leq h$. Hence any $G \in W_{m,n,h}$ decomposes into the following.

- A partition P of M where $x, y \in M$ belong to the same part if there is a path from x to y in G.
- A partition Q of N where $x', y' \in N$ belong to the same part if there is a path from x' to y' in G.
- A bijection $\alpha : P \to Q$ such that every edge ab in G has $a \in p$ and $b \in \alpha(p)$ for some $p \in P$. Hence we have $|P| = |Q|$.

- For each $p \in P$, a subgraph in $W_{|p|,|\alpha(p)|,|p|+|\alpha(p)|-1}$ induced by the vertices in $p \cup \alpha(p)$.

Regarding the final bulleted item, we observe that the subgraph induced by $p \cup \alpha(p)$ for some $p \in P$ must be a spanning tree on $K_{|p|,|\alpha(p)|}$. Any more edges would cause a cycle, while fewer edges would cause disjoint components (in which case G could be further decomposed).

We give an example of a graph G in $W_{3,5,6}$ below.



Here, the partitions P and Q are given by $P = \{\{u_1, u_2\}, \{u_3\}\}$ and $Q = \{\{v_1, v_2\}, \{v_3, v_4, v_5\}\}$ and we have the bijection α such that

$$\begin{aligned} \{u_1, u_2\} &\mapsto \{v_1, v_2\} \\ \{u_3\} &\mapsto \{v_3, v_4, v_5\}. \end{aligned}$$

We see that G decomposes into a graph in $W_{2,2,3}$ and a graph in $W_{1,3,3}$.

In general, for any $p \in P$, we must have exactly $|p| + |\alpha(p)| - 1$ edges in $p \cup \alpha(p)$. Summing this over all $p \in P$ gives $h = m + n - |P|$. We conclude that

$$|P| = m + n - h$$

(and thus $|Q| = m + n - h$).

We will now describe how to construct any graph in $W_{m,n,h}$ via its decomposition. Let $\mathcal{P}$ be the set of partitions of M and let $\mathcal{Q}$ be the set of partitions of N. Given (a) a partition $P \in \mathcal{P}$ of size $|P| = m+n-h$, (b) a partition $Q \in \mathcal{Q}$ and (c) a bijection $\alpha : P \to Q$, we can construct $\prod_{p \in P} w_{|p|,|\alpha(p)|,|p|+|\alpha(p)|-1}$ distinct graphs $G \in W_{m,n,h}$ by the following steps:

1. Start with G as the graph with vertex set $M \cup N$ and no edges.

2. For each $p \in P$ add one of the subgraphs in $W_{|p|,|\alpha(p)|,|p|+|\alpha(p)|-1}$ on the vertices $p \cup \alpha(p)$.

Since all graphs in $W_{m,n,h}$ can be constructed uniquely by the above steps we have the following theorem.

Theorem 1 *For $m, n, h \in \mathbb{N}$,*

$$f_{m,n,h} = \sum_{i=0}^{m} \sum_{j=0}^{n} \binom{m}{i} \binom{n}{j} w_{i,j,h}$$

where

$$w_{m,n,h} = \sum_{\substack{P \in \mathcal{P} \\ |P|=m+n-h}} \sum_{\substack{Q \in \mathcal{Q} \\ |Q|=m+n-h}} \sum_{\substack{\alpha: P \to Q \\ \alpha \text{ is a bijection}}} \prod_{p \in P} w_{|p|,|\alpha(p)|,|p|+|\alpha(p)|-1} \tag{3}$$

where

$$w_{|p|,|\alpha(p)|,|p|+|\alpha(p)|-1} = |p|^{|\alpha(p)|-1} |\alpha(p)|^{|p|-1}.$$

4.2 An improved formula for $f_{m,n,h}$

We will derive two further equations for $f_{m,n,h}$ (in Theorems 2 and 3), which are related to Theorem 1. They result in a slightly more complicated algorithm, but will allow us to compute $f_{m,n,h}$ faster than (2).

For all $m, n, h \in \mathbb{N}$, define

$$W'_{m,n,h} = \{G \in W_{m,n,h} : \text{the vertices in } N \text{ have degree } \geq 2\}.$$

Given any $G \in B_{m,n,h}$ we can delete the isolated vertices and the leaves (vertices of degree 1) in N to obtain a spanning forest in $W'_{i,j,h-k}$ for some $i \leq m$ and $j \leq n$ and $k \in \mathbb{N}$. Conversely, given any $G \in W'_{i,j,h-k}$, we can add:

- $m - i$ isolated vertices to M in $\binom{m}{m-i}$ ways, so as to increase $|M|$ to m, then
- $n - j$ isolated vertices to N in $\binom{n}{n-j}$ ways, so as to increase $|N|$ to n, then
- k edges to G in $\binom{n-j}{k} m^k$ ways, so as to increase the number of leaves in N by k,

thereby obtaining a spanning forest in $B_{m,n,h}$. Similar to the derivation of (2), we need to relabel the vertices so as to preserve the order of the indices of the non-isolated vertices within M and the non-isolated non-leaf vertices within N. Hence

$$f_{m,n,h} = \sum_{i=0}^{m} \sum_{j=0}^{n} \sum_{k=0}^{\min(n-j,h)} \binom{m}{i} \binom{n}{j, k, n-j-k} m^k w'_{i,j,h-k}.$$

All of the steps involved for finding the formula (3) for $w_{m,n,h}$ are still valid for $w'_{m,n,h}$. Hence (3) remains true if we replace w with w'. However, we can no longer make use of the simple formula for $w_{|p|,|\alpha(p)|,|p|+|\alpha(p)|-1}$. Nevertheless, we will be able to find a formula for $w'_{|p|,|\alpha(p)|,|p|+|\alpha(p)|-1}$ via Inclusion-Exclusion.

For $I \subseteq N$, define

$$A_I = \{G \in W_{m,n,m+n-1} : \text{vertices in } I \text{ are leaves}\}.$$

Note that spanning forests in A_I might also have leaves outside of I. Hence $|A_\emptyset| = |W_{m,n,m+n-1}| = m^{n-1} n^{m-1}$. By symmetry, if $I, J \subseteq N$ and $|I| = |J|$ then $|A_I| = |A_J|$. So we will assume $I = \{n-i+1, n-i+2, \ldots, n\}$. We can construct the graphs in A_I by adding i leaves to the graphs in $W_{m,n-|I|,m+n-|I|-1}$ (*i.e.*, the set of spanning trees of $K_{m,n-|I|}$), and these leaves can be added in m^i ways. Hence

$$\begin{aligned} |A_I| &= m^i \cdot \text{number of spanning trees of } K_{m,n-i} \\ &= m^i m^{n-i-1} (n-i)^{m-1} \\ &= m^{n-1} (n-i)^{m-1} \end{aligned}$$

when $m \geq 1$ and $n \geq 1$. Here we take $0^0 = 1$, since we want to account for $K_{1,0} \in W_{1,0,0}$. Hence, by Inclusion-Exclusion, we find

$$\begin{aligned} w'_{m,n,m+n-1} &= |A_\emptyset| - \Big| \bigcup_{\substack{I \subseteq N \\ I \neq \emptyset}} A_I \Big| \\ &= m^{n-1} n^{m-1} - \sum_{i=1}^{n} (-1)^{i-1} \binom{n}{i} m^{n-1} (n-i)^{m-1} \\ &= m^{n-1} \sum_{i=0}^{n} (-1)^i \binom{n}{i} (n-i)^{m-1} \end{aligned}$$

when $m \geq 1$ and $n \geq 1$. By definition, we also have $w'_{m,n,m+n-1} = 0$ when either $m = 0$ or $n = 0$ (and $w'_{m,n,m+n-1}$ is not defined when $m = 0$ and $n = 0$).

Before identifying formulae for $w_{m,n,h}$ and $w'_{m,n,h}$ in the next section, we make the following observation.

Lemma 5 *For all $m, n, h \in \mathbb{N}$, we have $n!$ divides $w'_{m,n,h}$.*

Proof: The symmetric group on N acts on $W'_{m,n,h}$ by permuting the vertices in N. Suppose a graph $G \in W'_{m,n,h}$ has a non-trivial stabiliser subgroup under this action. Then there exists a non-identity permutation α such that $\alpha G = G$, and distinct vertices $v, v' \in N$ for which $\alpha(v) = v'$. Since $G \in W'_{m,n,h}$, the degree of v is 2 or more, so assume v has distinct neighbours $a, b \in M$. Then, since $\alpha G = G$, we find that $v' = \alpha(v)$ has the neighbours a and b too. Thus, $\{v, v', a, b\}$ induces a 4-cycle, giving a contradiction. Hence, all stabiliser subgroups are trivial, and by the Orbit-Stabiliser Theorem, all orbits have size $n!$. Hence $n!$ divides $w'_{m,n,h}$. □

4.3 Formulae for $w_{m,n,h}$ and $w'_{m,n,h}$

Now we will simplify (3), noting that our simplifications remain valid when we replace w with w'.

For $a \geq 1$ and $t \geq 1$, let $S_{a,t}$ be the set of (number) partitions of a into t non-zero parts. We will interpret elements of $S_{a,t}$ as multisets; for example $\{4, 3, 3, 1, 1\} \in S_{12,5}$. We will require $S_{a,t} = \emptyset$ when $t < 0$, and $S_{0,0} = \{\emptyset\}$ and $S_{0,t} = \emptyset$ when $t > 0$. For $z \in S_{a,t}$, let $T[z]$ be the set of partitions of $\{1, 2, \ldots, a\}$ whose part sizes induce the number partition z. For $z \in S_{a,t}$, let $\hat{z}$ denote an arbitrary element of $T[z]$. If $P \in T[x]$ and $Q \in T[z]$, then any bijection $\alpha : P \to Q$ induces a bijection $\beta : x \to z$.

Hence

$$\begin{aligned}
w_{m,n,h} &= \sum_{x,z\in S_{m,m+n-h}} \sum_{P\in T[x]} \sum_{Q\in T[z]} \sum_{\substack{\alpha:P\to Q\\ \alpha \text{ is a bijection}}} \prod_{p\in P} w_{|p|,|\alpha(p)|,|p|+|\alpha(p)|-1} \\
&= \sum_{x,z\in S_{m,m+n-h}} \sum_{P\in T[x]} \sum_{Q\in T[z]} \sum_{\substack{\alpha:\hat{x}\to \hat{z}\\ \alpha \text{ is a bijection}}} \prod_{p\in \hat{x}} w_{|p|,|\alpha(p)|,|p|+|\alpha(p)|-1} \\
&= \sum_{x,z\in S_{m,m+n-h}} |T[x]|\cdot|T[z]| \sum_{\substack{\alpha:\hat{x}\to \hat{z}\\ \alpha \text{ is a bijection}}} \prod_{p\in \hat{x}} w_{|p|,|\alpha(p)|,|p|+|\alpha(p)|-1} \\
&= \sum_{x,z\in S_{m,m+n-h}} |T[x]|\cdot|T[z]| \sum_{\substack{\beta:x\to z\\ \beta \text{ is a bijection}}} \prod_{r\in x} w_{r,\beta(r),r+\beta(r)-1}.
\end{aligned}$$

If $z \in S_{a,t}$, then

$$|T[z]| = \frac{a!}{\prod_{i\geq 1} i!^{s_i(z)} s_i(z)!}$$

where $s_i(z)$ denotes the number of parts i in z (see *e.g.* [1, Theorem 13.2]). Hence we have the following theorems.

Theorem 2 *For $m, n, h \in \mathbb{N}$,*

$$f_{m,n,h} = \sum_{i=0}^{m}\sum_{j=0}^{n} \binom{m}{i}\binom{n}{j} w_{i,j,h}$$

where

$$w_{m,n,h} = \sum_{x,z\in S_{m,m+n-h}} \sum_{\substack{\beta:x\to z\\ \beta \text{ is a bijection}}} \frac{m!}{\prod_{i\geq 1} i!^{s_i(x)} s_i(x)!} \frac{n!}{\prod_{i\geq 1} i!^{s_i(z)} s_i(z)!} \prod_{r\in x} r^{\beta(r)-1}\beta(r)^{r-1}. \tag{4}$$

Theorem 3 *For $m, n, h \in \mathbb{N}$,*

$$f_{m,n,h} = \sum_{i=0}^{m}\sum_{j=0}^{n}\sum_{k=0}^{\min(n-j,h)} \binom{m}{i}\binom{n}{j, k, n-j-k} m^k w'_{i,j,h-k}.$$

where

$$w'_{m,n,h} = \sum_{x,z\in S_{m,m+n-h}} \sum_{\substack{\beta:x\to z\\ \beta \text{ is a bijection}}} \frac{m!}{\prod_{i\geq 1} i!^{s_i(x)} s_i(x)!} \frac{n!}{\prod_{i\geq 1} i!^{s_i(z)} s_i(z)!} \prod_{r\in x} w'_{r,\beta(r),r+\beta(r)-1}$$

where

$$w'_{m,n,m+n-1} = m^{n-1}\sum_{i=0}^{n}(-1)^i\binom{n}{i}(n-i)^{m-1}.$$

The main advantage of Theorems 2 and 3 over Theorem 1 is the summation is over number partitions (rather than set partitions).

5 Implementation

The author has implemented the two formulae for $f_{m,n,h}$ described in the preceding section, along with a simple backtracking algorithm. The results between all three implementations concur, giving confidence in the accuracy of the code. The C source code is available as supplementary material to this document. The GMP library was used for arbitrary precision arithmetic [6].

The formula involving $w'_{m,n,h}$ instead of $w_{m,n,h}$ was (unsurprisingly) much faster, largely because many values of $w'_{m,n,h}$ equal 0, such as when $m \leq n$, which can be used to drastically reduce the search tree.

5.1 Symmetry breaking

The run-time of the program was also improved through the use of symmetry breaking, which we will now describe. For $x \in S_{m,m+n-h}$ and $z \in S_{n,m+n-h}$, there are often many bijections $\beta : x \to z$ which map the same elements to the same elements (since both x and z are multisets). In these instances, a naïve algorithm would repeat the same computation unnecessarily. Let x and z be the multisets $x = \{x_1, x_2, \ldots, x_{m+n-h}\}$ and $z = \{z_1, z_2, \ldots, z_{m+n-h}\}$. We define the condition:

> *Symmetry breaking condition 1*: We say $\beta : x \to z$ is *half-canonical* if $\beta^{-1}(z_{i-1}) < \beta^{-1}(z_i)$ whenever $z_{i-1} = z_i$.

We need to add a multiplicative factor to adjust for the restriction to half-canonical bijections. Hence

$$w_{m,n,h} = \sum_{x,z \in S_{m,m+n-h}} \sum_{\substack{\beta : x \to z \\ \beta \text{ is a bijection} \\ \beta \text{ is half-canonical}}} \left(\prod_{i \geq 1} i!^{s_i(z)} \right) \frac{m!}{\prod_{i \geq 1} i!^{s_i(x)} s_i(x)!} \frac{n!}{\prod_{i \geq 1} i!^{s_i(z)} s_i(z)!} \prod_{r \in x} w_{r,\beta(r),r+\beta(r)-1}$$

and similarly with w' in place of w. Using this assumption, we reduce the number of bijections by a factor of $\prod_{i \geq 1} i!^{s_i(z)}$, which results in a substantial time saving.

Instead of the single symmetry breaking assumption, it is possible to utilise symmetry breaking using an additional condition:

> *Symmetry breaking condition 2*: We say $\beta : x \to z$ is *canonical* if it is half-canonical and $\beta(x_{i-1}) < \beta(x_i)$ whenever $x_i = x_{i-1}$.

Again, we find

$$w_{m,n,h} = \sum_{x,z \in S_{m,m+n-h}} \sum_{\substack{\beta : x \to z \\ \beta \text{ is a bijection} \\ \beta \text{ is canonical}}} \Gamma_{x,z,\beta} \frac{m!}{\prod_{i \geq 1} i!^{s_i(x)} s_i(x)!} \frac{n!}{\prod_{i \geq 1} i!^{s_i(z)} s_i(z)!} \prod_{r \in x} w_{r,\beta(r),r+\beta(r)-1},$$

and similarly with w' in place of w, where $\Gamma_{x,z,\beta}$ is the number of bijections between the multisets x and z which map the same elements to the same elements (as β). A formula for $\Gamma_{x,z,\beta}$ was given by *e.g.* [3], namely

$$\Gamma_{x,z,\beta} = \frac{\prod_{i \geq 1} s_i(x)! \prod_{i \geq 1} s_i(z)!}{\prod_{i \geq 1} \prod_{j \geq 1} s_{i,j}(x,\beta)!}$$

where

- $s_i(z)$ denotes the number of parts i in z (as before), and
- $s_{i,j}(x, \beta)$ is the number of elements (i, j) in the multiset $\{(r, \beta(r)) : r \in x\}$.

The author has implemented both of these symmetry breaking schemes in order to compare their performance (see Section 5.3).

5.2 Pseudo-code

Algorithm 1 gives a pseudo-code version of the C code used to implement the algorithm described by Theorem 3 using the half-canonical symmetry breaking condition. The partitions of m and n into k parts were computed whenever needed and stored in memory. Iterating through the half-canonical bijections was performed "on the fly" using a backtracking algorithm.

While Theorems 2 and 3 are valid for all $m, n, h \in \mathbb{N}$, we need to set $f_{m,n,0} = 1$ separately in the C code.

5.3 Performance

The C code was run on a 2×2.66 GHz processor (although the code itself is not parallelised). The following table gives the run-times (in seconds) for the two algorithms under the two symmetry breaking schemes when computing all non-zero $f_{m,n,h}$ with $m, n \leq 19$.

	half-canonical	canonical
Theorem 2	27.1	6.8
Theorem 3	4.1	4.5

The fastest version is Theorem 3 using half-canonical symmetry breaking. Under these conditions, the code had the following run-times to find all non-zero values of $f_{m,n,h}$ with $m, n \leq t$:

t	time (sec)
25	20.5
26	27.7
27	35.4
28	46.0
29	62.2
30	83.0

This table indicates the scalability of the program, which is not overwhelming for these values of t. The author also ran the program to compute $f_{m,n,h}$ with $m, n \leq t$ where $t = 50$, which took under 2 days (to be precise, it took 1 day 15 hours and 17 minutes). The largest number encountered was $f_{50,50,101}$, which has 167 digits, and is equal to the number of spanning trees of $K_{50,50}$, which is $50^{50-1} \cdot 50^{50-1} = 50^{98}$.

5.4 Complexity

In the worst case, $x = z = \{1, 2, 3, \ldots, t\}$, where $\min(m, n) = 1 + 2 + \cdots + t = \frac{1}{2}t(t + 1)$ (so $t = O(\sqrt{\min(m, n)})$, in which case the program must iterate through all $t!$ bijections from x to z. So, for a worst case analysis, we assume $m = n$ and $m + n - h = \lfloor \sqrt{m} \rfloor$.

Algorithm 1 Implementation of Theorem 3 using half-canonical symmetry breaking

```
f_{m,n,m+n-1} := 0; for all m, n, h
w'_{m,n,m+n-1} := 0; for all m, n, h
w'_{0,0,0} := 1;
for m = 0, 1, ..., MAX(m) do
    for n = 0, 1, ..., MAX(n) do
        if m > n and m > 0 and n > 0 then // Other values of w'_{m,n,h} are 0
            for i = 0, 1, ..., n do
                // Using the Inclusion-Exclusion formula
                w'_{m,n,m+n-1} := w'_{m,n,m+n-1} + (-1)^i binom(n,i) (n-i)^{m-1};
            end for
            for h = 0, 1, ..., m+n-2 do
                w'_{m,n,h} := 0;
                Compute all partitions of m into m+n-h parts and store in memory.
                Compute all partitions of n into m+n-h parts and store in memory.
                for partitions x of m into m+n-h parts do
                    for partitions z of n into m+n-h parts do
                        for half-canonical bijections β : x → z do // via backtracking algorithm
                            Γ := ∏_{i≥1} i!^{s_i(z)};
                            w'_{m,n,h} := w'_{m,n,h} + Γ · m!/(∏_{i≥1} i!^{s_i(x)} s_i(x)!) · n!/(∏_{i≥1} i!^{s_i(z)} s_i(z)!) · ∏_{r∈x} w'_{r,β(r),r+β(r)-1};
                        end for
                    end for
                end for
            end for
        end if
        f_{m,n,0} = 1;
        for h = 1, 2, ..., m+n-1 do
            for i = 0, 1, ..., m do
                for j = 0, 1, ..., n do
                    for k = 0, 1, ..., min(n-j, h) do
                        f_{m,n,h} := f_{m,n,h} + binom(m,i) binom(n; j,k,n-j-k) m^k w'_{i,j,h-k};
                    end for
                end for
            end for
        end for
        for h = 0, 1, ..., m+n-1 do
            Print f_{m,n,h};
        end for
    end for
end for
```

Hence, given $x, z \in S_{m,m+n-h}$, there can be $O(\lfloor\sqrt{m}\rfloor!)$ canonical bijections from x to z. Hardy and Ramanujan's asymptotic formula for the number of partitions of m, namely

$$\frac{1}{4m\sqrt{3}}e^{\pi\sqrt{2m/3}},$$

gives a crude asymptotic upper bound on $|S_{m,\lfloor\sqrt{m}\rfloor}|^2$, specifically

$$|S_{m,\lfloor\sqrt{m}\rfloor}|^2 = O\left(\frac{e^{\text{const}\cdot m}}{m^2}\right)$$

as the number of ways of choosing x and z. Hence, (4) has

$$O\left(e^{\text{const}\cdot\sqrt{m}}m^{\sqrt{m}/2+\text{const}}\right)$$

terms, by Stirling's Approximation. In contrast, a backtracking algorithm would need to generate and check around $\binom{m^2}{\lfloor\sqrt{m}\rfloor} \leq \frac{1}{\lfloor\sqrt{m}\rfloor!}m^{2\lfloor\sqrt{m}\rfloor}$ graphs, which, when $m = n$ and $h = 2m - \sqrt{m}$, is

$$O(e^{\sqrt{m}}m^{1.5\sqrt{m}+\text{const}})$$

iterations, by Stirling's Approximation. Of course, when implementing these algorithms, we use pruning whenever possible to reduce the search space, which makes a drastic difference not accounted for in these approximations.

5.4.1 When h is fixed

We conclude this paper with the observation that, when h is fixed, computing $f_{m,n,h}$ is asymptotically "easy". The underlying reason is that, for sufficiently large m or n, we must have isolated vertices in graphs in $B_{m,n,h}$. Thus, (2) only contains a finite number of non-zero terms.

Theorem 4 *For fixed h, computing $f_{m,n,h}$ can be performed in time $O\big(\log(mn)\big)$.*

Proof: If $i > h$ or $j > h$, then $w_{i,j,h} = 0$ (since any graph in $B_{i,j,h}$ must have an isolated vertex). Hence, (2) is equivalent to

$$f_{m,n,h} = \sum_{i=0}^{h}\sum_{j=0}^{h}\binom{m}{i}\binom{n}{j}w_{i,j,h}.$$

For fixed h, there is a finite number of terms in this sum. Thus, for fixed h, we could write a program in which:

- we store a list of the pairs (i, j) for which $w_{i,j,h}$ is non-zero, along with the value of $w_{i,j,h}$,
- we iterate through this list, computing $\binom{m}{i}\binom{n}{j}w_{i,j,h}$, and add it to a running total.

We can compute $\binom{m}{i} = \frac{1}{i!}m(m-1)\cdots(m-i+1)$ using $O(h)$ multiplications (since $i \leq h$), each of which takes time $O(\log m)$, and one division. Hence, $\binom{m}{i}$ can be computed in time $O(\log m)$ time (since h is fixed). Similarly $\binom{n}{j}$ can be computed in $O(\log n)$ time. We conclude that the whole summation can be performed in time $O\big(\log(mn)\big)$. $\square$

Acknowledgements

The author would like to thank Graham Farr for his feedback. The author would also like to thank Zur Luria for directing her to [3]. Thanks also to `mathoverflow.net` and `math.stackexchange.com` for providing a forum to discuss problems arising in this work.

References

[1] G. E. ANDREWS, *The theory of partitions*, Cambridge University Press, 1984.

[2] R. P. ANSTEE AND M. FARBER, *Characterizations of totally balanced matrices*, J. Algorithms, 5 (1984), pp. 215–230.

[3] A. BARVINOK, *Enumerating contingency tables via random permanents*, Combin. Probab. Comput., 17 (2008), pp. 1–19.

[4] A. BJÖKLUND, T. HUSFELDT, P. KASKI, AND M. KOIVISTO, *Computing the Tutte polynomial in vertex-exponential time*, in Proc. of the 49th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2008), 2008, pp. 677–686.

[5] G. FARR AND C. MCDIARMID, *The complexity of counting homeomorphs*, Theoretical Computer Science, 36 (1985), pp. 345–348.

[6] T. GRANLUND ET AL., *GNU multiple precision arithmetic library*. `http://gmplib.org/`.

[7] Y. JIN AND C. LIU, *The enumeration of labelled spanning trees of $K_{m,n}$*, Australas. J. Combin., 28 (2003), pp. 73–79.

[8] ——, *Enumeration for spanning forests of complete bipartite graphs*, Ars Combin., 70 (2004), pp. 135–138.

[9] C. J. LIU AND Y. CHOW, *Enumeration of forests in a graph*, Proc. Amer. Math. Soc., 83 (1981), pp. 659–662.

[10] W. MYRVOLD, *Counting k-component forests of a graph*, Networks, 22 (1992), pp. 647–652.

[11] T. PORTER, *Generating the list of spanning trees in $K_{s,t}$*, J. Combin. Math. Combin. Comput., 50 (2004), pp. 17–32.

[12] L. A. SZÉKELY, *Counting rooted spanning forests in complete multipartite graphs*, Ars Combin., 73 (2004).

[13] Y. TERANISHI, *The number of spanning forests of a graph*, Discrete Math., 290 (2005), pp. 259–267.