Computation of matching polynomials and the number of 1-factors in polygraphs

Per Håkan Lundow

Research reports, No 12, 1996

Computation of matching polynomials and the number of 1-factors in polygraphs

Per Håkan Lundow

Abstract. The number of 1-factors in the 6-cube is 16 332 454 526 976. This was computed with the traditional permanent and also with a transfer matrix approach, associated with polygraphs. For polygraphs of the type $G \times P_m$ we present a method for compression of the transfer matrix. This compression gives a substantial reduction of the order of the transfer matrix by exploiting the automorphisms of the graph G. We compute and tabulate matching polynomials and the number of 1-factors in various polygraphs, such as the $4 \times 4 \times m$-grid. A Mathematica package for computing matching polynomials, permanents and generating transfer matrices etc is listed at the end of the paper.

1. Introduction

A simple graph is denoted $G = (V, E)$ where V is the set of vertices and E is the set of edges. A matching M is any set of independent edges in G, i.e., no pair of edges in M have a vertex in common. A k-matching is a matching on k edges and a perfect matching is a matching that covers all the vertices in G. The matching polynomial of a graph G on n vertices is defined as

$$\mu(G; x) = \sum_{k=0}^{\lfloor n/2 \rfloor} (-1)^k p(G, k) x^{n-2k}$$

where $p(G, k)$ denotes the number of k-matchings in G and we define $p(G, 0) = 1$. A 1-factor is a spanning 1-regular subgraph. The edges of a 1-factor then form a perfect matching and the number of 1-factors in a graph G is denoted $\Phi(G)$. In general it is a $\#P$-complete problem to compute $\mu(G; x)$ and also $\Phi(G)$, though there are families of graphs such as paths, cycles and complete graphs, for which these functions can be simply expressed. Apart from these instances, general expressions are scarce. It is well-known however, that $\Phi(G)$ can be computed in polynomial time for planar graphs. Computing the matching polynomial is still harder, becoming $\#P$-complete even for planar graphs. More information on these matters can be found in Godsil [5] and Lovász and Plummer [14]. For more on complexity classes, see Welsh [22].

In the next section we will state some of the applications of matching theory to physics and chemistry. This is followed by a quick introduction to the subject of computing the matching polynomial and the number of 1-factors in a graph. A family of graphs of interest in chemistry, polygraphs, is presented together with the transfer matrix method to compute their matching polynomials. We then present a

Many thanks go to my supervisor Roland Häggkvist for help and advice, and for introducing me to a branch of mathematics that really counts.

new result, a compression of the matrices, which allows us to make these matrices considerably smaller. The algorithms described are implemented in a Mathematica package, which is listed at the end of the paper. The use of the package routines are demonstrated and we give tables of matching polynomials and the number of 1-factors in some polygraphs along with some recurrence relations.

2. Applications of matching theory

There are several connections between matching theory and statistical physics and also chemistry. For example, adsorption of oxygen and hydrogen on a metallic surface can be modelled by a system of monomers-dimers. The question is whether adsorption undergoes a phase transition at some critical temperature. The surface is represented as a grid and it is exposed to a gas consisting of monomers and dimers. Dimers could here correspond to oxygen molecules which cover adjacent vertices on the grid. A set of dimers forms a matching on the grid and the state of the system is then represented by this matching. As partition function one takes the matching polynomial with non-negative coefficients. The paper by Heilmann and Lieb [7] contains a detailed study of this problem.

The Ising model is concerned with the phenomenon of spontaneous magnetization. If a magnetic material is placed in a hot environment it becomes unmagnetized, although below a certain critical temperature the material will regain a degree of its magnetism. We then have a phase transition at this critical temperature. The partition function of the Ising model can be expressed in terms of the matching polynomial of a graph with weighted edges, the weight of a matching here being the product of its edge-weights. Again we refer the reader to [7]. A nice introduction to the Ising model is given by Cipra [4].

In mathematical chemistry, molecules are viewed as graphs and chemists refer to 1-factors as Kekulé structures. It turns out that the stability of some families of molecules is closely related to the number of 1-factors in their graphs. Several types of polynomials, partition functions and invariants of interest in chemistry have been suggested, many of which are expressed in terms of the numbers $p(G,k)$. For example, the topological index

$$H(G) = \sum_{k=0}^{\lfloor n/2 \rfloor} p(G,k)$$

also known as the Hosoya index, has been used to model physicochemical properties such as the boiling point of hydrocarbons. The reader is referred to Hosoya [8], Rouvray [17] and Trinajstić [21].

A more general account of combinatorics in statistical physics and chemistry can be found in Chapter 37 and 38 of The Handbook of Combinatorics [6].

3. Computation of μ and Φ

3.1. Recursive methods. The following facts are useful when computing μ or Φ of a graph. We will just state them and refer the reader who requires proofs to [5].

$$\mu(G;x) = \mu(G-e;x) - \mu(G-u-v;x)$$
$$\Phi(G) = \Phi(G-e) + \Phi(G-u-v)$$

where $e = \{u, v\}$ is an edge of G. If G and H are disjoint graphs then

$$\mu(G \cup H; x) = \mu(G; x)\, \mu(H; x)$$
$$\Phi(G \cup H) = \Phi(G)\, \Phi(H)$$

Let P_n and C_n denote the path and cycle respectively on n vertices. The matching polynomials of these are

$$\mu(P_n; x) = \sum_{k=0}^{\lfloor n/2 \rfloor} (-1)^k \binom{n-k}{k} x^{n-2k}$$
$$\mu(C_n; x) = \sum_{k=0}^{\lfloor n/2 \rfloor} (-1)^k \frac{n}{n-k} \binom{n-k}{k} x^{n-2k}$$

The number of 1-factors in a path and a cycle is

$$\Phi(P_n) = \begin{cases} 1 & \text{if } n \text{ is even} \\ 0 & \text{otherwise} \end{cases} \qquad \Phi(C_n) = \begin{cases} 2 & \text{if } n \text{ is even} \\ 0 & \text{otherwise} \end{cases}$$

We can now give a simple recursive algorithm for computation of μ and Φ: If the maximum degree of the graph is at most 2, then the graph is a union of disjoint paths and cycles and we can compute the product of their respective μ or Φ. Otherwise, pick a pair of adjacent vertices of high degree, delete the vertices and the edge and make the appropriate recursive calls. Though recursive, the method works well for smaller graphs.

3.2. The permanent. For bipartite graphs, there is a simple non-recursive method to compute Φ. Let $G = (V \cup W, E)$ be a bipartite graph on $2n$ vertices with bipartition (V, W), where $V = \{ v_1, \dots, v_n \}$ and $W = \{ w_1, \dots, w_n \}$. The biadjacency matrix $B = (b_{i,j})$ is defined to have entries

$$b_{i,j} = \begin{cases} 1 & \text{if } \{v_i, w_j\} \in E \\ 0 & \text{otherwise} \end{cases}$$

The permanent of an $n \times n$-matrix B is defined as

$$\operatorname{per}(B) = \sum_{\pi} \prod_{i=1}^{n} b_{i,\pi(i)}$$

where the sum is taken over all permutations π of $\{ 1, \dots, n \}$. If B is the matrix defined above, then

$$\Phi(G) = \operatorname{per}(B).$$

Thus, counting the 1-factors in a bipartite graph is equivalent to evaluating the permanent of its biadjacency matrix. The permanent, looking deceptively similar to the determinant, shares few of its nice properties. Particularly the property $\det(AB) = \det(A)\det(B)$ does not hold for permanents. Also, whereas the determinant can be computed in $O(n^3)$ time, no polynomial-time algorithm is known for the permanent. In fact, it has been shown to be a $\#P$-hard problem, making computation of $\Phi(G)$ a $\#P$-complete problem for bipartite graphs as well. A detailed survey on the permanent is found in Minc [15] and a proof of the $\#P$-hardness result is sketched in [22]. Evaluation of the permanent, as formulated above, would require $n \cdot n!$ arithmetic operations. It was shown by Ryser [18] that

$$\operatorname{per}(B) = (-1)^n \sum_{S \subseteq [n]} (-1)^{|S|} \prod_{i=1}^{n} \sum_{j \in S} b_{i,j}$$

where $[n] = \{ 1, \dots, n \}$. This reduces the number of operations required to about $n^2\, 2^{n-1}$. Nijenhuis and Wilf [16] devised and implemented a method to reduce the number of operations by a factor n. Their main trick is to order the sets in the first sum in Gray-code order, i.e., so that consecutive sets differ in exactly one element. As it stands then, the permanent can be computed with about $n 2^{n-1}$ operations. Counting the 1-factors in the 6-cube (64 vertices) is thus quite feasible, but the 7-cube (128 vertices) would require immense computer resources with this approach.

There are inequalities for permanents of doubly stochastic matrices (having row and column sums equal to 1) that can be applied to regular bipartite graphs, see [14]. If the bipartite graph G above is k-regular then

$$n! \left(\frac{k}{n}\right)^n \leq \Phi(G) \leq (k!)^{n/k}$$

Applying this inequality to the 7-cube[1] we get the approximate bounds $3.9280 \cdot 10^{27} \leq \Phi(Q^7) \leq 7.0924 \cdot 10^{33}$.

3.3. Estimation of Φ. We finish this section by describing a simple probabilistic method for estimating $\Phi(G)$, proved in [14]. The adjacency matrix $A = (a_{i,j})$ of an oriented graph $\overrightarrow{G}$ on the vertices $\{ v_1, \dots, v_n \}$ has entries

$$a_{i,j} = \begin{cases} 1 & \text{if } (v_i, v_j) \in E \\ -1 & \text{if } (v_j, v_i) \in E \\ 0 & \text{otherwise} \end{cases}$$

Give the graph G an orientation by randomly orienting every edge with probability $1/2$ in either direction. It turns out that the expected value of $\det(A(\overrightarrow{G}))$ is $\Phi(G)$. This implies a probabilistic method to estimate $\Phi(G)$. Just compute

$$\frac{1}{p} \sum_{i=1}^{p} \det(A(\overrightarrow{G}))$$

where the sum is taken over p independently chosen orientations of G. When G is bipartite we can gain a factor 8 in running time. Give G a random orientation $\overrightarrow{G}$ by letting each non-zero entry of the biadjacency matrix B be positive or negative with equal probability. Observe that if G is bipartite then

$$A(\overrightarrow{G}) = \begin{pmatrix} 0 & B(\overrightarrow{G}) \\ -B(\overrightarrow{G})^T & 0 \end{pmatrix}$$

and the reader may verify that

$$\det(A(\overrightarrow{G})) = (\det(B(\overrightarrow{G}))^2$$

The major drawback with the method is that the number p which gives a small relative error with a large probability is not necessarily polynomially bounded in n.

[1]The n-cube is n-regular

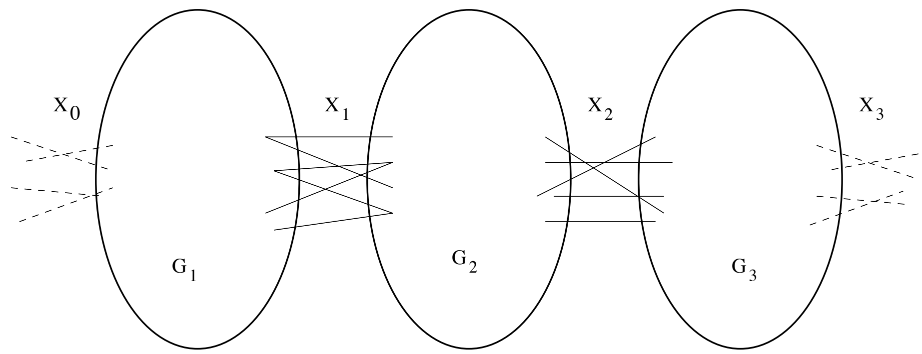

FIGURE 1. The structure of a polygraph

Only for a few families of graphs is this known to be the case. However, the very simplicity of the method makes it a first candidate for computing a rough estimate of $\Phi(G)$, or at least the number of digits of $\Phi(G)$. For example, a FORTRAN 77 implementation of the above, with $p = 10^7$ for the 7-cube resulted in the estimate $\Phi(Q^7) \approx 3.89 \cdot 10^{29}$.

Another approximation method has been presented by Jerrum and Sinclair [13]. Their algorithm is a fully-polynomial randomised approximation scheme. This means that it can in polynomial time, with a certain pre-chosen probability, approximate the permanent within a certain pre-chosen relative error. However, this algorithm lacks the simplicity of the one described above.

4. Polygraphs

So far we have not discussed how to take advantage of symmetries or recurring structures in a graph when computing matching polynomials. As an example, the reader may have in mind the $2 \times 2 \times m$-grid, $m \geq 1$, when reading this section. This is just the 2×2-grid, recurring m times, linked together by edges. Graphs of this kind belong to a family of graphs of interest in theoretical chemistry and are called polygraphs, see Figure 1. They were introduced by Babic et al. [2] who also gave a matrix method for computing matching polynomials of these graphs. A polygraph consists of a set of disjoint graphs $G_1, \dots, G_m$ and a set of binary relations $X_1, \dots, X_m$. Let $X_i \subseteq V(G_i) \times V(G_{i+1})$ for $i = 1, \dots, m-1$ and $X_m \subseteq V(G_m) \times V(G_1)$. For consistency we define X_0 to be identical to X_m. The polygraph Ω_m has vertices $V(G_1) \cup \cdots \cup V(G_m)$ and edges $E(G_1) \cup X_1 \cup \cdots \cup E(G_m) \cup X_m$. Let Γ_m be the graph Ω_m without the edges X_m. If $G_1 = \cdots = G_m = G$ and $X_1 = \cdots = X_m = X$ we denote Ω_m by ω_m and call it a rotagraph on (G, X). Likewise, we denote Γ_m by γ_m and call it a fasciagraph on (G, X). Let $M(X)$ be the set of all matchings in X. We index these matchings with numbers $1, 2, \dots, |M(X)|$ and adopt the convention of letting the first matching be the empty set. Let $W_i^{(k)}$ denote the ith element in $M(X_k)$. If $W \in M(X)$, let $D(W)$ and $R(W)$ be the domain and range respectively of W. Define $\mu(G - A - B; x) = 0$ and $\Phi(G - A - B) = 0$ if $A \cap B \neq \emptyset$, where $A, B \subseteq V(G)$. Define matrices $T_k = T_k(G_k, X_{k-1}, X_k)$, $k = 1, \dots, m$ with entries

$$T_k(i,j) = (-1)^{|W_j^{(k)}|}\, \mu(G - R(W_i^{(k-1)}) - D(W_j^{(k)});\, x) \qquad (4.1)$$

where the notation $T_k(i,j)$ refers to the entry in the ith row and jth column of the matrix T_k. Below we repeat some of the results in [2].

$$[T_1 \dots T_m](i,j) = (-1)^{|W_j^{(m)}|}\,\mu(\Gamma_m - R(W_i^{(m)}) - D(W_j^{(m)});\, x)$$
$$[T_1 \dots T_m](1,1) = \mu(\Gamma_m; x)$$
$$\operatorname{tr}(T_1 \dots T_m) = \mu(\Omega_m; x)$$

For rota- and fasciagraphs, we have that $T_1 = \cdots = T_m = T$ where

$$T(i,j) = (-1)^{|W_j|}\mu(G - R(W_i) - D(W_j);\, x) \tag{4.2}$$

We then have

$$T^m(i,j) = (-1)^{|W_j|}\,\mu(\Gamma_m - R(W_i^{(m)}) - D(W_j^{(m)});\, x)$$
$$T^m(1,1) = \mu(\gamma_m; x)$$
$$\operatorname{tr}(T^m) = \mu(\omega_m; x)$$

The results above can be simplified if we just want the number of 1-factors. In that case we replace the entries of the matrix (4.1) by

$$T_k(i,j) = \Phi(G - R(W_i^{(k-1)}) - D(W_j^{(k)})) \tag{4.3}$$

and obtain the following equations

$$[T_1 \dots T_m](i,j) = \Phi(\Gamma_m - R(W_i^{(m)}) - D(W_j^{(m)}))$$
$$[T_1 \dots T_m](1,1) = \Phi(\Gamma_m)$$
$$\operatorname{tr}(T_1 \dots T_m) = \Phi(\Omega_m)$$

For rota- and fasciagraphs we use the matrix

$$T(i,j) = \Phi(G - R(W_i) - D(W_j)) \tag{4.4}$$

and obtain

$$T^m(i,j) = \Phi(\gamma_m - R(W_i^{(m)}) - D(W_j^{(m)}))$$
$$T^m(1,1) = \Phi(\gamma_m)$$
$$\operatorname{tr}(T^m) = \Phi(\omega_m)$$

Now let T be defined by either Equation (4.2) or (4.4). From the matrix T, we can construct recurrence relations for the matching polynomial and the number of 1-factors in ω_m and γ_m. Denote the characteristic polynomial of the matrix T by

$$\Xi(T,\lambda) = \det(\lambda I - T) = \sum_{k=0}^{N} a_k \lambda^{N-k}$$

where $N = |M(X)|$ (which is also the order of T). Application of the Cayley-Hamilton theorem gives that $\Xi(T,T) = \mathbf{0}$, where the $\mathbf{0}$ represents a zero-matrix of order N. From this we derive the recursive formulae of order N

$$\sum_{k=0}^{N} a_k \operatorname{tr}(T^{m-k}) = 0$$

$$\sum_{k=0}^{N} a_k\, T^{m-k}(1,1) = 0$$

where $m \geq N$. Note that when we are determining $\mu(\omega_m; x)$ and $\mu(\gamma_m; x)$, the coefficients a_k will be polynomials in x.

5. Compression

Let T be the matrix defined by either Equation (4.2) or (4.4), depending on whether we want to compute the matching polynomial or the number of 1-factors.

Of course we wish the order of T to be as small as possible, to make matrix computations easy and the recurrence relations short. Unfortunately, though the method described in the previous section *does* take advantage of the recurring structure of the rota- and fasciagraphs, any symmetry in the graph G is *not* exploited. For example, if the edges in X are all independent, the matrix T has order $2^{|X|}$, no matter what graph G we use, empty or complete. In this section we will address this problem. In fact, in a special case we may reduce the order of the matrices by almost a factor the size of the automorphism group of G. First some notation though.

If G and H are graphs, then the Cartesian product $G \times H$ is defined as the graph having vertices $V(G) \times V(H)$ and where (v, w) is adjacent to (v', w') if and only if

$$v = v' \text{ and } \{w, w'\} \in E(H), \text{ or}$$
$$w = w' \text{ and } \{v, v'\} \in E(G)$$

For example, $P_m \times P_n$ is the $m \times n$-grid, $C_m \times P_n$ is a cylinder and $C_m \times C_n$ is a torus. Let $\mathrm{Aut}(G)$ denote the automorphism group of G and let $A \subseteq V(G)$ such that $\alpha(A) = A$ for all $\alpha \in \mathrm{Aut}(G)$. The case we are aiming for is the fasciagraph γ_m on (G, X) where we let $X = \{\, (v, v) : v \in A \,\}$. If $A = V(G)$ then $\gamma_m = G \times P_m$.

We will now classify the subsets of A into equivalence classes under the automorphism group according to the following; Let $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_r$ be the equivalence classes of subsets of A. That is to say, every $I \subseteq A$ belongs to some $\mathcal{A}_k$, and $I, J \in \mathcal{A}_k$ if and only if $J = \alpha(I)$ for some $\alpha \in \mathrm{Aut}(G)$. As a convention we let $\mathcal{A}_1 = \{\, \emptyset \,\}$. We can now define the compressed matrix C in terms of the matrix T. Since the edges in X are independent, no confusion will arise when we write $T(I, J)$ instead of $T(i, j)$ where $I = D(W_i)$ and $J = R(W_j)$.

Definition 5.1. The compressed transfer matrix C is the $r \times r$-matrix with entries

$$C(i, j) = \sum_{J \in \mathcal{A}_j} T(I, J) \quad \text{where } I \in \mathcal{A}_i \tag{5.1}$$

for $i, j = 1, \ldots, r$.

When calculating $C(i, j)$ we have to pick a set $I \in \mathcal{A}_i$. The following lemma says that it doesn't matter which set we pick, i.e., the matrix C is well-defined.

Lemma 5.2. *Let $I_1, I_2 \in \mathcal{A}_i$. Then*

$$\sum_{J \in \mathcal{A}_j} T(I_1, J) = \sum_{J \in \mathcal{A}_j} T(I_2, J)$$

for $i, j = 1, \ldots, r$

Proof. Since $I_1, I_2 \in \mathcal{A}_i$ we can assume that $I_2 = \alpha(I_1)$ for some permutation $\alpha \in \mathrm{Aut}(G)$. It suffices to show that the sets in $\{I_1 \cup J : J \in \mathcal{A}_j\}$ are isomorphic to the sets in $\{I_2 \cup J : J \in \mathcal{A}_j\}$ in some, possibly permuted, order. It follows by the definition of the set $\mathcal{A}_j$ that for all $\alpha \in \text{Aut}(G)$ and $J \in \mathcal{A}_j$ there is a $J' \in \mathcal{A}_j$ such that $J' = \alpha(J)$. Thus, for all $J \in \mathcal{A}_j$ there is a $J' \in \mathcal{A}_j$ such that

$$I_2 \cup J = \alpha(I_1) \cup \alpha(J') = \alpha(I_1 \cup J')$$

and the lemma follows. □

Theorem 5.3. *If $I \in \mathcal{A}_i$ then*

$$C^m(i,j) = \sum_{J \in \mathcal{A}_j} T^m(I,J)$$

Proof. By induction on m. The case $m = 1$ follows from the definition of the matrix C. Assume the theorem to be true for $m-1$ and show it for $m > 1$. We have

$$\sum_{J \in \mathcal{A}_j} T^m(I,J) = \sum_{J \in \mathcal{A}_j} \sum_{K \subseteq A} T^{m-1}(I,K)\, T(K,J) =$$

$$= \sum_{J \in \mathcal{A}_j} \sum_{k=1}^{r} \sum_{K \in \mathcal{A}_k} T^{m-1}(I,K)\, T(K,J) =$$

$$= \sum_{k=1}^{r} \sum_{K \in \mathcal{A}_k} T^{m-1}(I,K) \sum_{J \in \mathcal{A}_j} T(K,J)$$

By the lemma and the definition this is

$$\sum_{k=1}^{r} \sum_{K \in \mathcal{A}_k} T^{m-1}(I,K)\, C(k,j)$$

and the induction hypothesis allows us to write this as

$$\sum_{k=1}^{r} C^{m-1}(i,k)\, C(k,j) = C^m(i,j)$$

and by the principle of induction the theorem follows. □

Corollary 5.4. *If C is defined on the matrix T in Equation (4.2) then*

$$C^m(1,1) = \mu(\gamma_m; x)$$

Proof. Recall that $\mathcal{A}_1 = \{\emptyset\}$.

$$C^m(1,1) = \sum_{J \in \mathcal{A}_1} T^m(\emptyset, J) = T^m(\emptyset,\emptyset) = T^m(1,1) = \mu(\gamma_m; x)$$

□

Corollary 5.5. *If C is defined on the matrix T in Equation (4.4) then*

$$C^m(1,1) = \Phi(\gamma_m)$$

Comparing the orders of C and T, how much did we gain? The order of T is $N = 2^{|A|}$ since all edges in X are independent. If we denote by r the order of C, then r is (usually) slightly larger than $N/|\text{Aut}(G)|$ which is a lower bound on the number of equivalence classes. The exact number can be determined by employing Polya's Enumeration Theorem, see e.g. [6]:

$$r = \frac{1}{|\text{Aut}(G)|} \sum_{\pi \in \text{Aut}(G)} 2^{c(\pi,A)}$$

where $c(\pi, A)$ is the number of cycles in the permutation π that contain elements from A. Note that A was chosen so that every cycle in π either consists entirely of elements from A or entirely of elements from $V \setminus A$. In Broersma and Xueliang [3] a reduction of almost a factor 2 of the order of T was accomplished. They laid slightly less strong restrictions on the binary relation X (independent edges, though), but the graph G was restricted to having vertex-set $\{1, 2, \ldots, 2p\}$ and an automorphism $i \leftrightarrow p + i$, for $i = 1, \ldots, p$. The compression described here puts no restrictions on G, and works better the more automorphisms G has. Unfortunately we pay with information, since the trace of C no longer has the meaning it had for T.

6. Further reductions

We assume that we just want to count the 1-factors in γ_m. The order of the matrix C may then at least be halved to obtain a new, smaller, matrix $\hat{C}$. The simplest reduction stems from the fact that a graph on an odd number of vertices does not have a 1-factor. As before we let r denote the order of C. Renumber the families of sets that resulted from the classification procedure such that $\mathcal{A}_1, \ldots, \mathcal{A}_s$ contain the subsets of A of even size, and the remaining classes $\mathcal{A}_{s+1}, \ldots, \mathcal{A}_r$ contain the odd sets. If $|V(G)|$ is even then $C(i,j) = 0$ if $i \leq s$ and $j > s$, or, $i > s$ and $j \leq s$. If $|V(G)|$ is odd, then $C(i,j) = 0$ if $i, j \leq s$ or $i, j > s$. The matrix C will then look like

$$\begin{pmatrix} P & 0 \\ 0 & Q \end{pmatrix} \quad \text{for even } |V(G)|, \qquad \begin{pmatrix} 0 & R \\ S & 0 \end{pmatrix} \quad \text{for odd } |V(G)|. \tag{6.1}$$

Here P is an $s \times s$-matrix, Q an $(r-s) \times (r-s)$-matrix, R an $s \times (r-s)$-matrix and S an $(r-s) \times s$-matrix. Assume that $|V(G)|$ is even and define

$$\hat{C}(i,j) = C(i,j) \quad \text{for } i, j = 1, 2, \ldots, s \tag{6.2}$$

Note that $\hat{C}$ is the upper block P on the diagonal of C. The other blocks in C will not affect this matrix during matrix multiplication, since C is block triangular. We have then proved the following

Proposition 6.1.

$$\hat{C}^m(i,j) = C^m(i,j) \quad \textit{for } m \geq 1$$

We continue with the case when $|V(G)|$ is odd and define

$$\hat{C}(i,j) = C^2(i,j) \quad \text{for } i, j = 1, 2, \ldots, s \tag{6.3}$$

This means that $\hat{C}$ is the block product RS. Note that the upper left block in C^m will be a zero matrix when m is odd. A proposition similar to the one above follows.

Proposition 6.2.

$$\hat{C}^m(i,j) = C^{2m}(i,j) \quad \textit{for } m \geq 1$$

In both the odd and the even case we end up with an $s \times s$-matrix, where s is the number of even non-equivalent subsets of A. If $|A|$ is odd then $s = r/2$ and if $|A|$ is even then $s \approx r/2$. Roughly then, the order of $\hat{C}$ is half that of C.

The last case, finally, is when G is bipartite. Note that a bipartite graph on two sets of unequal size does not contain a 1-factor. Restrict G to be a bipartite graph on $2n$ vertices with bipartition (V, W) and let $|V| = |W| = n$. Again we renumber the classes, but this time such that for all $I \subseteq A$ we have that $I \in \mathcal{A}_1 \cup \cdots \cup \mathcal{A}_s$ if and only if $|I \cap V| = |I \cap W|$. Thus $I \in \mathcal{A}_{s+1} \cup \cdots \cup \mathcal{A}_r$ if and only if $|I \cap V| \neq |I \cap W|$. Then $C(i,j) = 0$ if $i \leq s$ and $j > s$, or, $i > s$ and $j \leq s$. The matrix C will then look like the matrix in Equation (6.1) (in the even case) and so we define

$$\hat{C}(i,j) = C(i,j) \quad \text{for } i,j = 1,2,\ldots,s \tag{6.4}$$

Correspondingly, Proposition 6.1 follows.

How much did this reduce the order of C? If we let $a_v = |A \cap V|$ and $a_w = |A \cap W|$, then the number of sets to classify is

$$a = \sum_{k=0}^{\min(a_v,a_w)} \binom{a_v}{k}\binom{a_w}{k}$$

The order of $\hat{C}$ is then approximately $\frac{ar}{N}$. For the special case when $A = V \cup W$, the above sum is

$$a = \sum_{k=0}^{n} \binom{n}{k}^2 = \binom{2n}{n} \sim \frac{2^{2n}}{\sqrt{\pi n}}$$

by Stirlings formula. Thus we can estimate the order of $\hat{C}$ to approximately $r/\sqrt{\pi n}$.

Henceforth, when we refer to $\hat{C}$ we mean that the appropriate reduction method has been applied. If G is bipartite as above, then we apply the reduction described for the bipartite case, and not merely the reduction in the even case.

7. Examples

In this section we apply the methods described above. At the same time we give a short demonstration of the functions in the Mathematica package in Section 9. It is assumed that the reader has properly installed the package on a computer. What the examples also should demonstrate is that the method of polygraphs is very general and unless we can use a compression technique it does not give us good, i.e. short, recursion formulae. It does, however, deliver the *specific* polynomials and numbers we desire, making tabulations of them fairly easy to carry through, even for rotagraphs, where this compression technique does not work.

Example 7.1. To compute the matching polynomial of a graph, we use the recursive method described in Section 3.1. The matching polynomial of the 4-cube is produced with the command

```
MatchingPolynomial[Hypercube[4],x]
```

where `x` is a variable. This returns the polynomial

$$272 - 3712\,x^2 + 11648\,x^4 - 14208\,x^6 + 8256\,x^8$$
$$-2496\,x^{10} + 400\,x^{12} - 32\,x^{14} + x^{16}$$

To obtain the number of 1-factors in the 4-cube, type

```
OneFactors[Hypercube[4]]
```

and we receive the constant term, 272, in the polynomial above. Since the 4-cube is bipartite the function computes the permanent of the biadjacency matrix. Had we entered a non-bipartite graph, the function would have used the recursive method of Section 3.1.

The permanent of a square matrix is computed with the Nijenhuis-Wilf method, see Section 3.2. This gives the permanent of the 10×10-matrix with zeroes on the diagonal and 1's off the diagonal

```
Permanent[1-IdentityMatrix[10]]
```

If we want to estimate the number of 1-factors in a fairly large graph, the probabilistic algorithm of Section 3.3 can be used. The command

```
OneFactorsEstimate[Hypercube[6],1000]
```

takes the average of 1000 determinants of oriented (bi-)adjacency matrices. The integer should be chosen with care, as large as possible to get a reliable result, modulo how long the user is prepared to wait. In this example, the graph is bipartite so the function will orient only the bi-adjacency matrix. A run returned the estimate $1.8051 \cdot 10^{13}$. Being a probabilistic algorithm though, the user will get different results at different runs.

The order of the compressed matrix C is the same as the number of non-equivalent 2-colourings of the set A. The number of non-equivalent 2-colourings of the corners on a 3×3-grid is

```
Classes[Automorphisms[GridGraph[3,3]],{1,3,7,9}]
```

i.e. 6. Observe that the set A *must* be preserved under the automorphism group of G.

Example 7.2. We compute the matching polynomial and the number of 1-factors in the fasciagraph $\gamma_m = P_2 \times P_2 \times P_m$ using the compression technique. The subsets of $A = V(P_2 \times P_2) = \{1, 2, 3, 4\}$ sorts into 6 classes under the automorphism group of $P_2 \times P_2$ and the compressed matrix C then has order 6. Type

```
g=GridGraph[2,2];
a=Range[V[g]];
t=CompressedTransferMatrix[g,a,x]
```

Here `g` is the graph, `a` is a set which is preserved under the automorphism group and `x` is the variable. The compressed matrix C, defined by Equation (5.1), is returned

$$\begin{pmatrix} 2-4x^2+x^4 & 8x-4x^3 & -4+4x^2 & 2x^2 & -4x & 1 \\ -2x+x^3 & 2-3x^2 & 2x & x & -1 & 0 \\ -1+x^2 & -2x & 1 & 0 & 0 & 0 \\ x^2 & -2x & 0 & 1 & 0 & 0 \\ x & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

If we type

```
c=RecursionCoefficients[t];
p[m_]:=p[m]=Sum[Expand[c[[i]]*p[m-i],{i,Length[c]}];
r=t[[1]];p[1]=r[[1]];
Do[r=Expand[r.t];p[i]=r[[1]],{i,2,Length[c]}]
```

then the recursive formula below is returned in the array `p`. The last line puts the first values of $\mu(\gamma_m; x)$ in an array, so that typing e.g. `p[7]` returns $\mu(\gamma_7; x)$. We obtain the recursive formula

$$\mu(\gamma_m;\,x) = (6 - 7x^2 + x^4)\,\mu(\gamma_{m-1};x)$$
$$+(-7 - 6x^2 + 6x^4 - x^6)\,\mu(\gamma_{m-2};x)$$
$$+(-8 + 26x^2 - 10x^4 + 2x^6)\,\mu(\gamma_{m-3};x)$$
$$+(9 - 6x^2 + 2x^4 - x^6)\,\mu(\gamma_{m-4};x)$$
$$+(2 + x^2 + x^4)\,\mu(\gamma_{m-5};x) -\, \mu(\gamma_{m-6};x)$$

If we want $\Phi(\gamma_m)$, observe that the graph $P_2 \times P_2 = (V \cup W, E)$ is bipartite with $|V| = |W| = 2$. So we only need to classify those subsets $I \subseteq A$ such that $|I \cap V| = |I \cap W|$. There are only 6 such sets and they sort into 3 classes. Thus, the matrix $\hat{C}$ has order 3. This time, leave out the symbol `x` in the function and type

```
g=GridGraph[2,2];
a=Range[V[g]];
t=CompressedTransferMatrix[g,a]
```

The matrix $\hat{C}$, defined by Equation (6.4), is returned

$$\begin{pmatrix} 2 & 4 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

To get a recursive formula for $\Phi(\gamma_m)$ we proceed as above and receive the following recursive formula

$$\Phi(\gamma_m) = 3\,\Phi(\gamma_{m-1}) + 3\,\Phi(\gamma_{m-2}) -\, \Phi(\gamma_{m-3})$$

We could of course solve this recursive relation to get an explicit formula for $\Phi(\gamma_m)$, but we leave this to the enthusiastic reader.

The recursive formulae above corresponds exactly to those obtained by Hosoya and Motoyama [10]. They also gave a recursive formula for $\Phi(\gamma_m)$ with $\gamma_m = P_2 \times P_3 \times P_m$. Typing the last command sequence with `g=GridGraph[2,3]` will return exactly the same recursive formula, namely

$$\Phi(\gamma_m) = 6\,\Phi(\gamma_{m-1}) + 21\,\Phi(\gamma_{m-2}) - 42\,\Phi(\gamma_{m-3})$$
$$-89\,\Phi(\gamma_{m-4}) + 68\,\Phi(\gamma_{m-5}) + 89\,\Phi(\gamma_{m-6}) - 42\,\Phi(\gamma_{m-7})$$
$$-21\,\Phi(\gamma_{m-8}) + 6\,\Phi(\gamma_{m-9}) +\, \Phi(\gamma_{m-10})$$

The authors of [10] estimated the order of the recursive formula for $\mu(\gamma_m;x)$ to be approximately 20. This method would return a compressed matrix of order 24 which suits fairly well to their estimate.

We finish this example with a word of warning. Suppose that we replace the graph used above, $P_2 \times P_2$, with an odd graph, such as P_3, and generate the matrix $\hat{C}$. Then $\hat{C}^m(1,1) = \Phi(P_3 \times P_{2m})$ (!). Also, the `RecursionCoefficients`-function returns the coefficients $\{5, -5, 1\}$, which should be interpreted as

$$\Phi(P_3 \times P_{2m}) = 5\,\Phi(P_3 \times_{2m-2}) - 5\,\Phi(P_3 \times P_{2m-4}) + \Phi(P_3 \times P_{2m-6})$$

Example 7.3. Let $G = P_2 \times P_2$ and $X = \{(1,1),(2,2),(3,3),(4,4)\}$. Then $\omega_m = P_2 \times P_2 \times C_m$. To compute $\mu(\omega_4;x) = \mu(Q^4;x)$ type

```
g=GridGraph[2,2];
b=Table[{i,i},{i,V[g]}];
t=TransferMatrix[g,b,b,x];
t4=Expand[MatrixPower[t,4]];
Sum[t4[[i,i]],{i,Length[t4]}]
```

Here `b` is the binary relation and `t` is the matrix defined by Equation (4.1). If we leave out the symbol `x`, the matrix defined by Equation (4.3) is returned. For larger matrices, the `MatrixPower`-function should be replaced by a more efficient function since we only want the entries on the diagonal. If we continue with the command

```
{i,j}=TransferMatrixPosition[b,b,{1,2},{3,4}];
t4[[i,j]]
```

then we receive the polynomial denoted by

$$(-1)^{|W_j^{(4)}|}\,\mu(P_2 \times P_2 \times P_4 - R(W_i^{(4)}) - D(W_j^{(4)});\, x)$$

where $R(W_i^{(4)}) = \{1,2\}$ and $D(W_j^{(4)}) = \{3,4\}$, see Section 4. This is the matching polynomial of the graph in Figure 2 where the vertices marked 1, 2, 15 and 16 are deleted.

We could of course obtain recursive formulae for $\Phi(\omega_m)$ and $\mu(\omega_m; x)$ with the `RecursionCoefficients`-function, but they would be unnecessarily long since they would both have order 16. In [10] a recursive formula for $\Phi(\omega_m)$ of order 8 was given, and the recursive formula for $\mu(\omega_m; x)$, was estimated to have order 10.

Example 7.4. In this example we scrutinize the 3-dimensional grids $\gamma_m = P_4 \times P_4 \times P_m$. Let us first view it as the fasciagraph on $P_4 \times P_4$ and $X = \{\,(1,1),\ldots,(16,16)\,\}$. The matrix T has order 65536, which would require an enormous amount of computer memory to store. However, T will be very sparse. Since 16 vertices overlap in X only $(3/4)^{16} \approx 1\%$ of the entries are non-zero and, if we only want 1-factors, fewer still are non-zero. The use of typical sparse matrix methods for computations of powers of T is of course a justified approach. Compression works well here, the automorphism group of $P_4 \times P_4$ has 8 elements and the order of C is 8548. This is still a trifle too big when we are storing polynomials in a computer. The matrix $\hat{C}$ on the other hand has order 1723, as computations have shown, and this is not too big to treat easily. Note that only the elements $\hat{C}^m(1,1)$ are desired, and so only vector-matrix multiplication needs to be performed. Tables of the numbers $\Phi(\gamma_m)$ are found in the Table 9. This approach does not bring us the matching polynomials of γ_m, but for smaller m we can use a rota-graph approach. Let $G = P_2 \times P_2 \times P_4$ and $X = \{(3,3),(4,2),(7,7),(8,6),(11,11),(12,10),(15,15),(16,14)\}$, see Figure 2. The rotagraph on (G, X) is the cubic grid $P_4 \times P_4 \times P_4$. The matrix T has order 256, which is fairly easily treated. The resulting polynomials are listed in Table 9.

Example 7.5. The n-cube, denoted Q^n, is the graph having the set of binary strings of length n as vertices. Two vertices are adjacent if their binary strings differ in exactly one position. Note that $Q^n = Q^{n-1} \times P_2$ and $Q^n = Q^{n-2} \times C_4$. We view Q^6 as the rotagraph $Q^4 \times C_4$ and proceed to compute $\Phi(Q^6)$. The compression technique does not work for rotagraphs, but the matrix T, for counting 1-factors, is extremely sparse and also symmetric, making matrix multiplication feasible. T has order $2^{16} = 65\,536$ but only $5\,494\,273$ non-zero elements, making storage of the matrix in a computer memory possible on a larger workstation. Recall that $\mathrm{tr}(T^4)$ is the desired number. This approach was implemented in Fortran 90 and

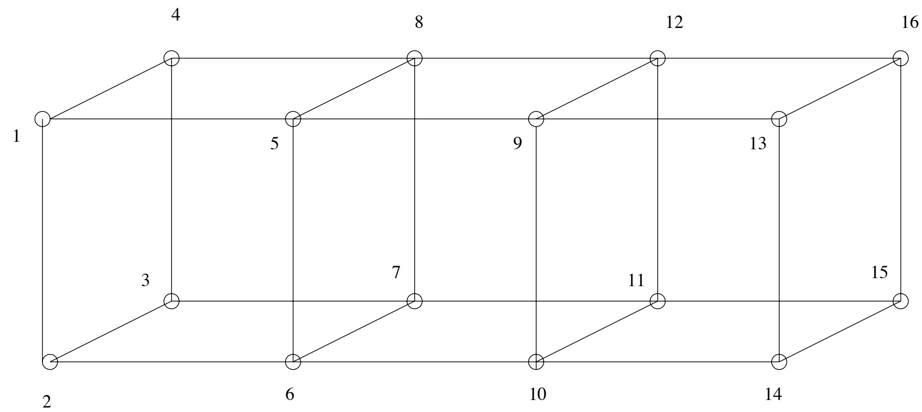

FIGURE 2. The $2 \times 2 \times 4$-grid

the computations were performed on an IBM RS/6000 25T and took about 33 hours. The resulting number, 16 332 454 526 976, was confirmed by computing the permanent of the biadjacency matrix. Using a Fortran 90 implementation of the permanent-function, this computation only took about 6 hours. The matching polynomial of Q^6 remains to be computed, though.

8. Tables

"This process of reduction to cipher is the highest effort man or woman is capable of making. It is the only effort worth making, and it is possible only through ever-increasing self-restraint..."

Gandhi, 1927.

The matching polynomials and the number of 1-factors has been extensively tabulated for various grids, cylinders and tori. General expressions exist for the number of 1-factors in graphs such as $P_m \times C_n$, $C_m \times C_n$, $P_2 \times P_3 \times P_m$ and, as mentioned before, $P_m \times P_n$. The papers by Hosoya et al. [8, 9, 10, 11, 12] contain plenty of tables and general expressions, to which we refer the reader. Fans of integer sequences might want to consult the book by Sloane and Plouffe [20], which also can be reached on the Internet as a searchable database. Below is listed tables of $p(G,k)$, $\Phi(G)$ and recurrence relations for some fasciagraphs on smaller cycles, grids and hypercubes. They were generated by running the package in Section 9 on a Power Macintosh 8100/80. In the tables of $p(G,k)$, integers being the number of 1-factors are printed in bold. To simplify the recurrence relations we let μ_m denote $\mu(\gamma_m; x)$ and Φ_m denote $\Phi(\gamma_m)$. Let also r denote the order of the compressed matrix C for matching polynomials and $\hat{r}$ the order of the compressed (and reduced) matrix $\hat{C}$ for 1-factors.

TABLE 1. Order of compressed matrices for some $G \times P_m$

G	r	$\hat{r}$	G	r	$\hat{r}$	G	r	$\hat{r}$
$P_2 \times P_3$	24	10	P_2	3	2	C_3	4	2
$P_2 \times P_4$	76	27	P_3	6	3	C_4	6	3
$P_2 \times P_5$	288	82	P_4	10	5	C_5	8	4
$P_2 \times P_6$	1072	268	P_5	20	10	C_6	13	6
$P_3 \times P_3$	102	51	P_6	36	14	C_7	18	9
$P_3 \times P_4$	1120	274	P_7	72	36	C_8	30	11
$P_4 \times P_4$	8548	1723	P_8	136	43	C_9	46	23
$C_3 \times C_3$	26	13	P_9	272	136	C_{10}	78	26
Q^3	22	9	P_{10}	528	142	C_{11}	126	63
Q^4	402	93	P_{11}	1056	528	C_{12}	224	62

TABLE 2. $\gamma_m = C_3 \times P_m$

	$p(\gamma_m, k)$								
k	$m=1$	$m=2$	$m=3$	$m=4$	$m=5$	$m=6$	$m=7$	$m=8$	$m=9$
0	1	1	1	1	1	1	1	1	1
1	3	9	15	21	27	33	39	45	51
2		18	69	156	279	438	633	864	1131
3		**4**	107	501	1399	3017	5571	9277	14351
4			36	672	3558	11613	29049	61374	115392
5				285	4338	25029	92109	259956	615348
6				**19**	2100	28557	175363	709740	2214051
7					276	15072	190575	1226919	5363931
8						2880	106824	1284651	8582760
9						**91**	25978	752716	8726408
10							1818	216951	5289783
11								23754	1730235
12								**436**	255239
13									11085

TABLE 3. $\gamma_m = C_4 \times P_m = Q^2 \times P_m = P_2 \times P_2 \times P_m$

	$p(\gamma_m, k)$							
k	$m=1$	$m=2$	$m=3$	$m=4$	$m=5$	$m=6$	$m=7$	$m=8$
0	1	1	1	1	1	1	1	1
1	4	12	20	28	36	44	52	60
2	**2**	42	142	306	534	826	1182	1602
3		44	440	1672	4248	8680	15480	25160
4		**9**	588	4863	19774	56333	129644	258907
5			288	7416	55200	235132	728840	1840836
6			**32**	5470	91200	637914	2810312	9294734
7				1620	84984	1112668	7465728	33741064
8				**121**	40553	1208714	13541312	88199495
9					8204	771436	16397296	164774936
10					**450**	261500	12752616	216370582
11						39080	5986432	194313364
12						**1681**	1532336	114468886
13							178272	41514628
14							**6272**	8380100
15								788536
16								**23409**

$$\Phi(C_3 \times P_{2m}) = 5\,\Phi_{2m-2} - \Phi_{2m-4}$$

$$\Phi(C_4 \times P_m) = 3\,\Phi_{m-1} + 3\,\Phi_{m-2} - \Phi_{m-3}$$

$$\Phi(C_5 \times P_{2m}) = 19\,\Phi_{2m-2} - 41\,\Phi_{2m-4} + 19\,\Phi_{2m-6} - \Phi_{2m-8}$$

$$\Phi(C_6 \times P_m) = 4\,\Phi_{m-1} + 16\,\Phi_{m-2} - 6\,\Phi_{m-3} - 16\,\Phi_{m-4} + 4\,\Phi_{m-5} + \Phi_{m-6}$$

$$\Phi(Q^3 \times P_m) = 24\,\Phi_{m-1} + 28\,\Phi_{m-2} - 479\,\Phi_{m-3} + 428\,\Phi_{m-4} + 428\,\Phi_{m-5} \\ -479\,\Phi_{m-6} + 28\,\Phi_{m-7} + 24\,\Phi_{m-8} - \Phi_{m-9}$$

$$\mu(C_3 \times P_m;\, x) = (-5\,x + x^3)\mu_{m-1} + (-5 + 3\,x^2 - x^4)\mu_{m-2} + (x + x^3)\mu_{m-3} - \mu_{m-4}$$

TABLE 4. $\gamma_m = C_5 \times P_m$

$p(\gamma_m, k)$							
k	$m=1$	$m=2$	$m=3$	$m=4$	$m=5$	$m=6$	$m=7$
0	1	1	1	1	1	1	1
1	5	15	25	35	45	55	65
2	5	75	240	505	870	1335	1900
3		145	1125	3910	9495	18880	33065
4		95	2710	17725	64660	173020	382305
5		**11**	3227	48193	286799	1081285	3103896
6			1645	77405	839930	4723695	18237825
7			240	69510	1612685	14550495	78786505
8				31060	1975730	31488555	251718625
9				5360	1465295	47151280	593631680
10				**176**	598928	47476226	1023782605
11					113015	30669915	1268978075
12					6625	11778955	1100004130
13						2360195	639919835
14						191480	234612615
15						**2911**	49020224
16							4885170
17							153830

TABLE 5. $\gamma_m = C_6 \times P_m$

$p(\gamma_m, k)$							
k	$m=1$	$m=2$	$m=3$	$m=4$	$m=5$	$m=6$	$m=7$
0	1	1	1	1	1	1	1
1	6	18	30	42	54	66	78
2	9	117	363	753	1287	1965	2787
3	**2**	336	2290	7562	17874	34954	60530
4		420	8139	46938	160887	414792	894189
5		192	16446	187530	987834	3472752	9527094
6		**20**	18141	487241	4241321	21158661	75753275
7			9870	813486	12846774	95402040	458907006
8			2148	843342	27359544	320645463	2143757547
9			**108**	509542	40372976	803176510	7768505882
10				160653	40170300	1489152993	21861085377
11				21438	25795320	2015817270	47616569682
12				**725**	9980480	1949485107	79675739431
13					2078160	1304474898	101182136226
14					188832	576346062	95821362789
15					**4480**	156728330	66035085642
16						23429940	32011697004
17						1566180	10405152504
18						**28561**	2112964124
19							239567604
20							12371220
21							**179928**

$$\mu(C_4 \times P_m; x) = (6 - 7x^2 + x^4)\,\mu_{m-1} + (-7 - 6x^2 + 6x^4 - x^6)\,\mu_{m-2}$$
$$+(-8 + 26x^2 - 10x^4 + 2x^6)\,\mu_{m-3} + (9 - 6x^2 + 2x^4 - x^6)\,\mu_{m-4}$$
$$+(2 + x^2 + x^4)\,\mu_{m-5} - \mu_{m-6}$$

TABLE 6. $\omega_m = C_m \times C_m$

$p(\omega_m, k)$						
k	$m=3$	$m=4$	$m=5$	$m=6$	$m=7$	$m=8$
0	1	1	1	1	1	1
1	18	32	50	72	98	128
2	99	400	1075	2340	4459	7744
3	180	2496	13050	45456	125146	294656
4	72	8256	98800	589158	2427852	7915232
5		14208	486550	5386752	34580280	159744256
6		11648	1578475	35826516	374911887	2516765440
7		3712	3346050	176198256	3166014068	31752956416
8		**272**	4504825	645204321	21136978051	326548985472
9			3648500	1758028568	112610506694	2772159201280
10			1614680	3538275120	481275179119	19602839305216
11			325600	5185123200	1653171079994	116203098803200
12			19600	5409088488	4559561602155	579928810356224
13				3885146784	10059761199922	2442898448804352
14				1829582496	17638871964378	8695281617537024
15				524514432	24344085192546	26145886979093504
16				81145872	26098044303940	66314671200371680
17				5415552	21350764013376	141482680513783808
18				**90176**	13017600602531	252884264307891200
19					5728859993332	376612501051478016
20					1740658301592	464055797382556160
21					342232281104	468926793714612224
22					39320640352	384331205955039232
23					2199577856	251998823731046400
24					39143552	129925722660333312
25						51533494605877248
26						15285388354162688
27						3264906532544512
28						476614180386816
29						44039669555200
30						2278206984192
31						52613062656
32						**311853312**

$$\begin{aligned}\mu(C_5 \times P_m;\, x) &= (15\,x - 9\,x^3 + x^5)\mu_{m-1} + (-19 + 19\,x^2 - 27\,x^4 + 10\,x^6 - x^8)\mu_{m-2} \\ &+ (34\,x - 85\,x^3 + 69\,x^5 - 19\,x^7 + 2\,x^9)\mu_{m-3} + (-41 + 95\,x^2 - 39\,x^4 - 9\,x^6 \\ &+ 6\,x^8 - x^{10})\mu_{m-4} + (2\,x - 65\,x^3 + 39\,x^5 - 11\,x^7 + 2\,x^9)\mu_{m-5} \\ &+ (-19 + 11\,x^2 - 7\,x^4 + 2\,x^6 - x^8)\mu_{m-6} + (3\,x + x^3 + x^5)\mu_{m-7} - \mu_{m-8}\end{aligned}$$

TABLE 7. $\gamma_m = P_3 \times P_3 \times P_m$

	$p(\gamma_m, k)$					
k	$m=1$	$m=2$	$m=3$	$m=4$	$m=5$	$m=6$
0	1	1	1	1	1	1
1	12	33	54	75	96	117
2	44	436	1260	2525	4231	6378
3	56	2984	16736	50552	113684	215393
4	18	11434	140322	672126	2085694	5054442
5		24766	778452	6277198	27731168	87622530
6		29180	2913096	42480118	276805102	1164755616
7		16984	7361472	211846420	2120333560	12163620462
8		3993	12381180	784200907	12634826746	101433879357
9		**229**	13428840	2154366513	59027097072	682916407521
10			8893248	4362041263	216913695094	3738673165242
11			3278784	6419477292	626708528128	16712392258753
12			568344	6716664818	1417900872204	61103060700766
13			31344	4835018662	2493032893120	182629834939538
14				2281569082	3367348279396	445089189580448
15				655842108	3437515277416	880370659944042
16				101934041	2593501127101	1403576812451606
17				6870327	1402515949328	1786799130667754
18				**117805**	520871037067	1793930275383832
19					124842772364	1397774304403158
20					17531745326	827727493314932
21					1217704320	362423901173076
22					28613174	113077255268116
23						23878571601956
24						3164202873629
25						233176559173
26						7654682266
27						**64647289**

m	$\Phi(\gamma_m)$
8	35669566217
10	19690797527709
12	10870506600976757
14	6001202979497804657
16	3313042830624031354513
18	1829008840116358153050197
20	1009728374600381843221483965
22	557433823481589253332775648233
24	307738670509229621147710358375321
26	169891178715542584369273129260748045
28	93790658670253542024618689133882565125
30	51778366130057389441239986148841747669217
32	28584927722109981792301610403923348017948449
34	15780685138381102545287108197623881881376915397
36	8711934690116480171969789787256390490181022415693

$$
\begin{aligned}
\mu(C_6 \times P_m;\, x) &= (-12 + 29\,x^2 - 11\,x^4 + x^6)\mu_{m-1} + (-32 + 12\,x^2 + 47\,x^4 - 49\,x^6 \\
&+13\,x^8 - x^{10})\mu_{m-2} + (71 - 568\,x^2 + 948\,x^4 - 714\,x^6 + 266\,x^8 - 46\,x^{10} + 3\,x^{12})\mu_{m-3} \\
&+(313 - 983\,x^2 + 1261\,x^4 - 1339\,x^6 + 848\,x^8 - 283\,x^{10} + 46\,x^{12} - 3\,x^{14})\mu_{m-4} \\
&+(40 + 924\,x^2 - 2103\,x^4 + 1956\,x^6 - 812\,x^8 + 97\,x^{10} + 34\,x^{12} - 11\,x^{14} + x^{16})\mu_{m-5} \\
&+(-601 + 2884\,x^2 - 4334\,x^4 + 3559\,x^6 - 1903\,x^8 + 823\,x^{10} - 241\,x^{12} + 40\,x^{14} \\
&-3\,x^{16})\mu_{m-6} + (-311 + 1132\,x^2 - 470\,x^4 + 161\,x^6 + 259\,x^8 - 351\,x^{10} + 153\,x^{12} \\
&-32\,x^{14} + 3\,x^{16})\mu_{m-7} + (368 - 892\,x^2 + 1743\,x^4 - 1764\,x^6 + 968\,x^8 - 265\,x^{10} \\
&+26\,x^{12} + 3\,x^{14} - x^{16})\mu_{m-8} + (251 - 529\,x^2 + 575\,x^4 - 205\,x^6 - 60\,x^8 + 59\,x^{10} \\
&-18\,x^{12} + 3\,x^{14})\mu_{m-9} + (-47 - 172\,x^4 + 130\,x^6 - 58\,x^8 + 14\,x^{10} - 3\,x^{12})\mu_{m-10} \\
&+(-40 + 28\,x^2 - 11\,x^4 + 9\,x^6 - x^8 + x^{10})\mu_{m-11} + (-5\,x^2 - x^4 - x^6)\mu_{m-12} + \mu_{m-13}
\end{aligned}
$$

TABLE 8. $\gamma_m = C_3 \times C_3 \times P_m$

$p(\gamma_m, k)$					
k	$m=1$	$m=2$	$m=3$	$m=4$	$m=5$
0	1	1	1	1	1
1	18	45	72	99	126
2	99	810	2241	4401	7290
3	180	7518	39678	116316	257106
4	72	38709	442575	2039814	6188463
5		110817	3254724	25088310	107856216
6		167448	16056147	223066398	1409411676
7		117900	53046918	1456699500	14108774220
8		29520	115246440	7029374175	109615427955
9		**1120**	158653112	25022727081	665714322238
10			129944880	65127684555	3168417127554
11			56958480	121909424148	11801137694058
12			10992408	159953324046	34221545160489
13			585792	141626935710	76569860426940
14				80001899586	130436645000040
15				26440161960	166051546684152
16				4418860545	154011257081100
17				278666595	100510188513840
18				**2861029**	43956690488688
19					11993327746128
20					1823418619560
21					126181749120
22					2535163200

m	$\Phi(\gamma_m)$
6	7537209013
8	19875272280736
10	52411725012875905
12	138211512392292291937
14	364468498187098321751584
16	961115930137025304064194421
18	2534495671871264129163903449317
20	6683552014192354263830206528781536
22	17624755892139792658340655302347504609
24	46477085776829046768207330138587650280353
26	122561669252399975974463854446912020628372448
28	323199325406575344085727204017210842952442869861
30	852287706102859795133057944653046179293760543914869

Table 9. $\gamma_m = P_4 \times P_4 \times P_m$ (see Example 7.4)

	$p(\gamma_m, k)$			
k	$m = 1$	$m = 2$	$m = 3$	$m = 4$
0	1	1	1	1
1	24	64	104	144
2	224	1816	4992	9768
3	1044	30208	146940	415368
4	2593	328214	2972395	12430848
5	3388	2456736	43888740	278659560
6	2150	13022504	490410658	4862322484
7	552	49492032	4243096376	67752463152
8	**36**	135062729	28849000711	767471193606
9		262610832	155554203920	7157834054584
10		357580896	668490123332	55469187090396
11		331384336	2293235516668	359485412847192
12		200032432	6270624556725	1956911884067608
13		73483328	13607937421412	8971759857716256
14		14707328	23264863112266	34682805390128328
15		1308928	31002090496224	113035590354067768
16		**32000**	31731778597928	310146213937970487
17			24460558393664	714514530994393464
18			13831123293040	1376672261486529068
19			5534768640848	2206488832067036760
20			1490639531680	2921624380278645192
21			250915666208	3168204916452408416
22			23455372800	2783182424023411992
23			980808000	1953962180835361272
24			**10885344**	1077824850339404286
25				457155298292389608
26				144991813332269700
27				33134934405040272
28				5183929033351776
29				515240510630328
30				28894756833940
31				736291240776
32				**5051532105**

m	$\Phi(\gamma_m)$
5	2132137503232
6	932814464901633
7	403325499406267520
8	175220727982196365632
9	75996591204223021534740
10	32983893365927357595999561
11	14312021563707748863632803328
12	6210770058902782255142931025577
13	2695087892546319780280018750890580
14	1169519565840471449485662353652345920

Table 10. $\gamma_m = Q^3 \times P_m = P_2 \times C_4 \times P_m$

	$p(\gamma_m, k)$					
k	$m=1$	$m=2$	$m=3$	$m=4$	$m=5$	$m=6$
0	1	1	1	1	1	1
1	12	32	52	72	92	112
2	42	400	1150	2300	3850	5800
3	44	2496	14188	43088	97188	184488
4	**9**	8256	107695	527266	1654929	4038692
5		14208	523432	4443544	20139672	64617928
6		11648	1644732	26500328	181068176	783276300
7		3712	3299496	113179016	1226328112	7358912824
8		**272**	4081767	346488571	6323647928	54373784286
9			2921316	753945400	24931287776	318825871424
10			1081110	1145095736	75067593808	1490728945948
11			166956	1178382344	171583684352	5565948414304
12			**6345**	784950482	294442950944	16568492919146
13				315397840	372993052000	39145995644824
14				68116132	340488015872	72864467884044
15				6389688	216497649376	105704983692720
16				**155969**	91345139744	117774571436597
17					23795425024	98825636211608
18					3410831424	60839882617160
19					218213376	26523443163360
20					**3794880**	7794028097326
21						1436289443104
22						148108980604
23						6963545496
24						**92524801**

m	$\Phi(\gamma_m)$
7	2254970505
8	54961579408
9	1339585632201
10	32649998822849
11	795784687676160
12	19395815427419969
13	472737980834179401
14	11522134787497383568
15	280831232750814806025
16	6844754271574955786881
17	166828527501840135007680
18	4066144157174109525965249
19	99104922608108341099841865
20	2415503559516749295949014032
21	58873538190527241983722166409
22	1434936200120917128502072225921
23	34973979171320665509608373120000
24	852427598504309296925221936759681

Table 11. $\gamma_m = Q^4 \times P_m = C_4 \times C_4 \times P_m$

	$p(\gamma_m, k)$			
k	$m = 1$	$m = 2$	$m = 3$	$m = 4$
0	1	1	1	1
1	32	80	128	176
2	400	2840	7568	14600
3	2496	59120	274560	759584
4	8256	803580	6848000	27822084
5	14208	7517264	124694656	763504368
6	11648	49715240	1718209088	16311133584
7	3712	235146480	18327675008	278274362192
8	**272**	795862790	153549653616	3858979023370
9		1910146160	1019460142080	44051088838656
10		3190117800	5389069021056	417676281992856
11		3594554960	22710637612800	3310348880868432
12		2605908220	76162736983680	22024174794317232
13		1129177840	202303330851072	123313091919432144
14		259084440	422310466869504	581630577946974072
15		25108944	685115567624704	2310324639457748096
16		**589185**	850667743539584	7715963153250311251
17			792016077516800	21604808702631926656
18			538003442426880	50504855552895180056
19			256874061012992	98016417871417039760
20			81810395008768	156788269717168962800
21			16087147553792	204849983435540593552
22			1725682248704	216149310892878810872
23			80406638592	181614258291882122496
24			**930336768**	119387717864796680906
25				60042777844937606416
26				22443085396359803280
27				5999543286903760304
28				1087639382471943076
29				123724794351752480
30				7805441127361896
31				217782023223920
32				**1545853411969**

m	$\Phi(\gamma_m)$
5	2551276235535120
6	4215052025641922305
7	6962828841161217269760
8	11502121570585083415761153
9	19000667589592082076903397136
10	31387734017421172152812450818177
11	51850272028503475884947837964718080
12	85652909298496678900816312674765830017
13	141492427606024240247420896433978493594896
14	233735283924117864284203606114476852290977281

9. A Mathematica package

Here is listed the source code of a Mathematica package that contains implementations of the various algorithms described in this paper. The package employs the Combinatorica package which is a part of the Standard Mathematica Packages. For more on the Combinatorica package, consult Adamchic et al. [1], which is delivered along with Mathematica or, preferably, Skiena [19]. The best book on Mathematica in general is Wolfram [23].

```
(* :Title: Matching
*)
(* :Author: P.H. Lundow
*)
(* :Summary:
This package contains various routines for computing matching
polynomials and the number of 1-factors (perfect matchings) in a
graph. Functions that returns (compressed) transfer matrices for
polygraphs are included and also a function that estimates the number
of 1-factors in a graph. Graphs should be simple and represented in
accordance with the Combinatorica package which is delivered with
Mathematica.
*)
(* :Discussion:
Send comments to
        P.H. Lundow
        Department of Mathematics
        Umea University
        S-901 87 Umea
        Sweden
        phl@abel.math.umu.se
*)
(* :Package Version: 1.0, (June 18, 1996).
*)
(* :Mathematica Version: 2.2
*)

BeginPackage["Matching`",{"DiscreteMath`Combinatorica`"}]

TransferMatrix::usage =
"TransferMatrix[ g, x, y, z ] returns the transfer matrix of the graph
g with the binary relations x and y, represented as pairs of vertices.
The argument z is an optional symbol and if it is included the matrix
will contain matching polynomials, otherwise it will contain only
integers for counting 1-factors."

TransferMatrixPosition::usage =
"TransferMatrixPosition[ x, y, a, b ] returns the position in the
transfer matrix that corresponds to deleting the set a from the first
graph and the set b from the last graph in the polygraph. The arguments
x and y are the binary relations such that the range of x is in the
first graph and the domain of y is in the last graph of the polygraph."

CompressedTransferMatrix::usage =
"CompressedTransferMatrix[ g, a, z, p ] returns the compressed
transfer matrix for the fasciagraph on g with the binary relation
$X=\{(v,v): v \in A\}$ where a is the set $A$. Also, $A$ must be
preserved under the automorphisms of g. The argument z is an optional
symbol and if it is included the matrix will contain matching
```

```
polynomials, otherwise it will contain only integers for counting
1-factors. If z is not included, the reduced matrix $\hat C$ is
returned. The argument p is an optional permutation group without
which the function generates the automorphism group."

MatchingPolynomial::usage =
"MatchingPolynomial[ g, x ] returns the matching polynomial of the
graph g in terms of the symbol x."

OneFactors::usage =
"OneFactors[ g ] returns the number of 1-factors in the graph g."

OneFactorsEstimate::usage =
"OneFactorsEstimate[ g, n ] returns an approximation of the number of
1-factors in the graph g by computing the average of n determinants
of randomly oriented (bi)adjacency matrices."

Permanent::usage =
"Permanent[ x ] returns the permanent of the square matrix x."

RecursionCoefficients::usage =
"RecursionCoefficients[ t ] returns the list of coeficients of the
recursion formula that the square matrix t defines. For example, if we
put c=RecursionCoefficients[t], then a[n_]:=a[n]=Sum[c[[i]]*a[n-i],{n}]
defines a recursion formula that remembers the previous values."

Classes::usage =
"Classes[ a, s ] returns the number of non-equivalent 2-colourings of
the vertex set s in a graph with automorphism group a. Observe that s
must be preserved under the automorphism group a."

Begin["`Private`"]

TransferMatrix[ g_Graph, x:{{_,_}...}, y:{{_,_}...}, z_:0 ] :=
Module[{a,t,u,v,xy,s},
       xy = Union[Map[First,y],Map[Last,x]];
       If[ Head[z] === Symbol,
              s = -1;
              t = entrytablemp[g,xy,z],
              s = 1;
              t = entrytable1f[g,xy]
       ];
       u = Map[ Map[Last, #]&, matchings[x]];
       v = Map[ Map[First,#]&, matchings[y]];
       Table[
              a = u[[i]];
              Map[
                     If[ Intersection[a,#]=={},
                          s^Length[#] t[[1+RankSubset[xy,Union[a,#]]]],
                          0
                     ]&,
                     v
              ],
              {i,Length[u]}
       ]
]

TransferMatrixPosition[ x:{{_,_}...}, y:{{_,_}...}, a_List, b_List ] :=
```

```
Module[{aa=Sort[a],bb=Sort[b],m,i,j},
	m = matchings[x];
	i = Position[
		m,
		First[
			Select[
				m,
				(Sort[Map[Last, #]]==aa)&,
				1
			]
		]
	][[1,1]];
	m = matchings[y];
	j = Position[
		m,
		First[
			Select[
				m,
				(Sort[Map[First,#]]==bb)&,
				1
			]
		]
	][[1,1]];
	{i,j}
]

matchings[ x:{{_,_}...} ] :=
(* Returns the subsets of the binary relation x that are matchings *)
Module[{a,t,r=Range[Length[x]]},
	a = Table[
		t = x[[NthSubset[i,r]]];
		If[ multq[Map[First,t]] || multq[Map[Last,t]],
			0,
			t
		],
		{i,0,2^Length[x]-1}
	];
	DeleteCases[ a, _Integer ]
]

multq[ x_List ] := Length[Union[x]] < Length[x]
(* Checks if x is a multiset *)

CompressedTransferMatrix::notpreserved =
"The set `1` is not preserved under the automorphism-group"

CompressedTransferMatrix[ g_Graph, x_List, z_:0, group_List:{} ] :=
Module[{a=Sort[x],group2,q=(group=={})},
	If[ q, group2 = Automorphisms[g] ];
	If[ MemberQ[ If[q,group2,group], y_ /; perm[a,y]!=a ],
		Message[CompressedTransferMatrix::notpreserved,x];
		Return[]
	];
	Which[
		Head[z] === Symbol,
			ctmmp[g,a,z,If[q,group2,group]],
		EvenQ[V[g]],
			ctm1fe[g,a,If[q,group2,group]],
```

```
                OddQ[V[g]],
                        ctm1fo[g,a,If[q,group2,group]]
        ]
]

ctmmp[ g_Graph, a_List, z_Symbol, group_List ] :=
(* Returns the compressed transfer matrix for matching polynomials *)
Module[{classes,x,t},
        classes = Map[
                Map[NthSubset[#,a]&,#]&,
                classify[a,Range[0,2^Length[a]-1],group]
        ];
        t = entrytablemp[g,a,z];
        Table[
                x = First[classes[[i]]];
                Map[
                        Times[
                                (-1)^Length[First[#]],
                                Apply[Plus,t[[ctmentries[x,#,a]]]]
                        ]&,
                        classes
                ],
                {i,Length[classes]}
        ]
]

ctm1fe[ g_Graph, a_List, group_List ] :=
(* Returns the compressed transfer matrix
   for 1-factors in even graphs *)
Module[{classes,x,y,t=TwoColoring[g]},
        If[ !MemberQ[t,0] && Count[t,1]==Count[t,2],
                x = Flatten[Position[t,1]];
                y = Flatten[Position[t,2]];
                classes = Select[
                        Range[0,2^Length[a]-1],
                        (Length[Intersection[NthSubset[#,a],x]]==
                         Length[Intersection[NthSubset[#,a],y]])&
                ],
                classes = Select[
                        Range[0,2^Length[a]-1],
                        EvenQ[Length[NthSubset[#,a]]]&
                ]
        ];
        classes = Map[
                Map[NthSubset[#,a]&,#]&,
                classify[a,classes,group]
        ];
        t = entrytable1f[g,a];
        Table[
                x = First[classes[[i]]];
                Map[ Apply[Plus,t[[ctmentries[x,#,a]]]]&, classes ],
                {i,Length[classes]}
        ]
]

ctm1fo[ g_Graph, a_List ,group_List ] :=
(* Returns the compressed transfer matrix
   for 1-factors in odd graphs *)
```

```
Module[{even,odd,x,t},
       even = Select[
              Range[0,2^Length[a]-1],
              EvenQ[Length[NthSubset[#,a]]]&
       ];
       odd  = Complement[ Range[0,2^Length[a]-1], even ];
       even = Map[Map[NthSubset[#,a]&,#]&,classify[a,even,group]];
       odd  = Map[Map[NthSubset[#,a]&,#]&,classify[a,odd, group]];
       t = entrytable1f[g,a];
       Dot[
              Table[
                     x = First[even[[i]]];
                     Map[Apply[Plus,t[[ ctmentries[x,#,a] ]]]&,odd],
                     {i,Length[even]}
              ],
              Table[
                     x = First[odd[[i]]];
                     Map[Apply[Plus,t[[ ctmentries[x,#,a] ]]]&,even],
                     {i,Length[odd]}
              ]
       ]

]

classify[ t_List, tnum_List, group_List ] :=
(* The numbers in tnum, referring to subsets of t, are
   classified under the permutation-group *)
Module[{a,b,c,class=1,score=Table[0,{Length[tnum]}]},
       score = Table[0,{Length[tnum]}];
       class = 1;
       While[ MemberQ[score, 0],
              a = NthSubset[tnum[[ Position[score,0][[1,1]] ]],t];
              b = Map[
                     RankSubset[t,#]&,
                     Union[Map[perm[a,#]&,group]]
              ];
              c = Flatten[Map[Position[tnum,#]&,b],1];
              score = MapAt[class&,score,c];
              class = class + 1
       ];
       Map[tnum[[ Flatten[Position[score,#]] ]]&,Range[Max[score]]]
]

perm[ a_List, p_List ] := Sort[ p[[ a ]] ]
(* Returns the set a permuted by the permutation p *)

ctmentries[ a_List, b_List, c_List ] :=
(* Returns the positions in the entry table *)
Sort[
       Map[
              (1+RankSubset[c,#])&,
              Map[
                     Union[a,#]&,
                     Select[b,(Intersection[a,#]=={})&]
              ]
       ]
]
```

```
entrytablemp[ g_Graph, a_List, z_Symbol ] :=
(* Returns the list of matching polynomials in the transfer matrix *)
Module[{v = Range[V[g]]},
       Table[
              MatchingPolynomial[
                     InduceSubgraph[
                            g,
                            Complement[v,NthSubset[i,a]]
                     ],
                     z
              ],
              {i,0,2^Length[a]-1}
       ]
]

entrytable1f[ g_Graph, a_List ] :=
(* Returns the list of number of 1-factors in the transfer matrix *)
Module[{v = Range[V[g]]},
       Table[
              OneFactors[
                     InduceSubgraph[
                            g,
                            Complement[v,NthSubset[i,a]]
                     ]
              ],
              {i,0,2^Length[a]-1}
       ]
]

MatchingPolynomial[ g_Graph, x_Symbol ] :=
Module[{a,b,c,d,e},
       d = degrees[g];
       If[Max[d]>2,
              a = Position[d,Max[d]][[1,1]];
              b = neighbors[g,a];
              c = b[[Position[d[[b]],Max[d[[b]]]][[1,1]]]];
              e = {a,c};
              c = Complement[Range[V[g]],e];
              MatchingPolynomial[DeleteEdge[g,e],x]-
              MatchingPolynomial[InduceSubgraph[g,c],x],
              b = Flatten[Position[d,1]];
              c = ConnectedComponents[g];
              Expand[
                     Product[
                            Which[
                                   Length[c[[i]]] == 1,
                                   x,
                                   Intersection[c[[i]],b] != {},
                                   matchpolypath[ Length[c[[i]]],x],
                                   True,
                                   matchpolycycle[Length[c[[i]]],x]
                            ],
                            {i,Length[c]}
                     ]
              ]
       ]
]
```

```
matchpolypath[ n_Integer, x_Symbol ] :=
(* Returns the matching polynomial of a path on n vertices *)
Sum[
       (-1)^k Binomial[n-k,k] x^(n-2k),
       {k,0,Floor[n/2]}
]

matchpolycycle[ n_Integer, x_Symbol ] :=
(* Returns the matching polynomial of a cycle on n vertices *)
Expand[
      n Sum[
            (-1)^k Binomial[n-k,k]/(n-k) x^(n-2k),
            {k,0,Floor[n/2]}
      ]
]

OneFactors[ g_Graph ]:=
Module[{t,r,c,x},
       If[ OddQ[V[g]], Return[0]];
       If[ Max[degrees[g]]<3 || MemberQ[t=TwoColoring[g],0],
              phi[g],
              If[ Count[t,1] == Count[t,2],
                     r = Flatten[Position[t,1]];
                     c = Flatten[Position[t,2]];
                     permanent[Edges[g][[r,c]]],
                     0
              ]
       ]
]

phi[ g_Graph ]:=
(* Returns the number of 1-factors in an even graph. *)
Module[{a,b,c,d,e,s},
       If[ V[g]==0, Return[1] ];
       d = degrees[g];
       If[Max[d]>2,
              a = Position[d,Max[d]][[1,1]];
              b = neighbors[g,a];
              c = b[[Position[d[[b]],Max[d[[b]]]][[1,1]]]];
              e = {a,c};
              c = Complement[Range[V[g]],e];
              phi[DeleteEdge[g,e]]+phi[InduceSubgraph[g,c]],
              c = ConnectedComponents[g];
              If[MemberQ[Map[OddQ[Length[#]]&,c],True],
                     0,
                     2^(Length[c]-Count[d,1]/2)
              ]
       ]
]

degrees[ Graph[g_,_] ] := Map[Apply[Plus,#]&,g]
(* Returns the list of degrees of the vertices in a graph *)

neighbors[ Graph[g_,_], v_Integer ] := Flatten[Position[g[[v]],1]]
(* Returns the set of neighbors to the vertex a in a graph *)

Permanent[ x_List ] :=
permanent[x] /; MatrixQ[x] && MatchQ[Dimensions[x],{m_,m_}]
```

```
permanent[ x_List ] :=
(* Returns the permanent of the square matrix x *)
Module[{n=Length[x], w},
	If[n == 1, Return[x[[1,1]]]];
	w = Map[(Last[#]-Apply[Plus,#]/2)&,Transpose[x]];
	Dot[
		Map[
			Apply[Times,
				w += Sign[#] x[[Abs[#]]]
			]&,
			gray[n-1]
		],
		altsigns[n-1]
	] 2 (-1)^n
]

gray[ n_Integer ] :=
(* Returns the bits to switch in the binary reflected Gray code *)
Join[Fold[Join[#1,{#2},-Reverse[#1]]&,{1},Range[2,n]],{-n}]

altsigns[ n_Integer ] := Nest[Join[#,#]&,{1,-1},n-1]
(* Returns a list of length 2^n containing alternating +-1's *)

OneFactorsEstimate[ g_Graph, n_Integer ] :=
Module[{a,e,r,c,t,z},
	If[OddQ[V[g]], Return[0.]];
	If[ !MemberQ[t=TwoColoring[g],0],
		If[ Count[t,1]==Count[t,2],
			r = Flatten[Position[t,1]];
			c = Flatten[Position[t,2]];
			z = Edges[g][[r,c]];
			e = Position[z,1];
			Sum[
				Det[MapAt[N[2Random[Integer]-1]&,z,e]]^2,
				{n}
			] / N[n],
			0.
		],
		z = Table[0.,{V[g]},{V[g]}];
		e = Select[Position[Edges[g],1],(First[#]<Last[#])&];
		Sum[
			a = MapAt[N[2Random[Integer]-1]&,z,e];
			Det[a-Transpose[a]],
			{n}
		] / N[n]
	]
]

RecursionCoefficients[ t_?MatrixQ ]:=
Module[{d,h},
	d = (-1)^(Length[t]+1) Expand[CharacteristicPolynomial[t,h]];
	Rest[Reverse[CoefficientList[Collect[d,h],h]]]
]

Classes::notprs =
"The set `1` is not preserved under the automorphism-group"
```

```
Classes[ a_List, s_List ]:=
(If[MemberQ[a,x_/;perm[s,x]!=s],Message[Classes::notprs,s];Return[]];
Apply[Plus,
       Map[2^Length[Select[ToCycles[#],(Intersection[#,s]!={})&]]&,a]
]/Length[a])

End[]

EndPackage[]
```

References

[1] V. Adamchic et al., *Guide to standard Mathematica packages, version 2.2 (technical report)*, Wolfram Research, 1993

[2] D. Babic et al., *The matching polynomial of a polygraph*, Discrete Appl. Math. **15** (1986) 11–24

[3] H.J. Broersma and Li Xueliang, *On 'The matching polynomial of a polygraph'*, Discrete Appl. Math. **46** (1993) 79–86

[4] B.A. Cipra, *An introduction to the Ising model*, Amer. Math. Monthly **94** 937–959

[5] C.D. Godsil, *Algebraic combinatorics*, Chapman and Hall, 1993

[6] R. Graham, M. Grötschel and L. Lovasz (editors), *The Handbook of Combinatorics*, Elsevier Science B. V., 1995

[7] O.J. Heilmann and E.H. Lieb, *Theory of monomer-dimer systems*, Commun. Math. Phys. **25** (1972) 190–232

[8] H. Hosoya, *On some counting polynomials in chemistry*, Discrete Appl. Math. **19** (1988) 239–257

[9] H. Hosoya, *Matching and symmetry of graphs*, Comp. and Maths. with Appls. **12B** (1986) 271–290

[10] H. Hosoya and A. Motoyama, *An effective algorithm for obtaining polynomials for dimer statistics. Application of operator technique on the topological index to two- and three-dimensional rectangular and torus lattices*, J. Math. Phys. **26** (1985) 157–167

[11] H. Hosoya et al., *Generalized expression of the perfect matching number for* $2 \times 3 \times n$ *lattices*, J. Math. Phys. **34** (1993) 1043–1051

[12] H. Hosoya et al., *Generalized expression for the number of perfect matchings of cylindrical* $m \times n$ *graphs*, J. Math. Phys. **32** (1991) 1885–1889

[13] M. Jerrum and A. Sinclair, *Approximating the permanent*, SIAM J. Comput. **18** (1989) 1149–1178

[14] L. Lovasz and M.D. Plummer, *Matching theory*, North-Holland, 1986

[15] H. Minc, *Permanents*, Addison-Wesley, 1978

[16] A. Nijenhuis and H. Wilf, *Combinatorial algorithms*, Academic Press, 1978

[17] D. H. Rouvray, *The modeling of chemical phenomena using topological indices*, J. Comput. Chem., **8**, (1987) 470–480

[18] H.J. Ryser, *Combinatorial mathematics*, The Carus mathematical monographs, 1963

[19] S.S. Skiena, *Implementing discrete mathematics: combinatorics and graph theory with Mathematica*, Addison-Wesley, 1990

[20] N.J.A. Sloane and S. Plouffe, *The encyclopedia of integer sequences*, Academic Press, 1995

[21] N. Trinajstić, *Chemical graph theory*, CRC Press, 1992

[22] D. J. A. Welsh, *Complexity: Knots, colourings and counting*, Cambridge university press, 1993

[23] S. Wolfram, *Mathematica: a system for doing mathematics by computer*, Addison-Wesley, 1991

E-mail address: `phl@abel.math.umu.se`