Combinatorics, Probability and Computing (2010) 19, 463–480. © Cambridge University Press 2010
doi:10.1017/S0963548309990678

Counting Matchings and Tree-Like Walks in Regular Graphs

IAN M. WANLESS†

School of Mathematical Sciences, Monash University, Vic 3800, Australia
(e-mail: ian.wanless@sci.monash.edu.au)

Received 21 December 2007; revised 27 November 2009; first published online 10 February 2010

The number of closed tree-like walks in a graph is closely related to the moments of the roots of the matching polynomial for the graph. Thus, by counting these walks up to a given length it is possible to find approximations for the matching polynomial. This approach has been used in two separate problems involving asymptotic enumerations of 1-factorizations of regular graphs. Nevertheless, a systematic way to count the required walks had not previously been found.

In this paper we give an algorithm to count closed tree-like walks in a regular graph up to a given length. For small m, this provides expressions for the number of m-matchings in the graph in terms of the numbers of copies of certain small subgraphs that appear in the graph. The simplest of these expressions were already known, having been rediscovered by numerous authors using *ad hoc* methods. We offer the first general method for producing the expressions. We also find generating functions that isolate the contribution from the simplest kind of subgraph – namely a single cycle of arbitrary length.

1. Introduction

The aim of this work is to use generating functions to count a certain type of walk which arises in enumeration of Latin rectangles (Godsil and McKay [8]) and 1-factorizations of regular graphs (McLeod [13]). In doing so we will also establish relationships between numbers of matchings and counts of certain small subgraphs in regular graphs. Such relationships have been found before (*e.g.*, [1], [2], [6], [15]), but our method seems substantially more general than previous methods.

All graphs in this paper are simple and undirected. Throughout, G will be a $(\kappa+1)$-regular graph with $2n$ vertices. These values are chosen solely to make the formulae that we derive neater. It is important to stress that n is not an integer when G has an odd number of vertices, but that our theory applies without change in that case.

Suppose that W is a sequence of vertices $v_0v_1v_2\cdots v_l$ of G such that v_i is adjacent to v_{i-1} for each $i=1,2,\ldots,l$. We say that W is a *walk* of length l from v_0 to v_l in G and that

† Research supported by the Australian Research Council's Discovery Grant scheme.

https://doi.org/10.1017/S0963548309990678 Published online by Cambridge University Press

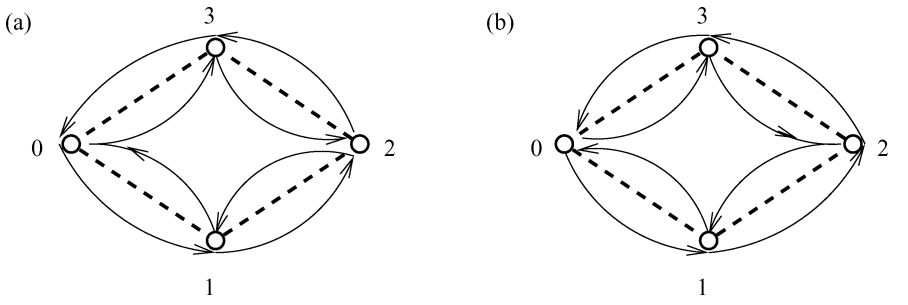


Figure 1. Two walks. In both cases the induced graph is a 4-cycle, 01230. Walk (a) is 012103230, which is closed and tree-like. Walk (b) is 012303210, which is closed but is not tree-like, since red(01230) is not a path.

W is *closed* if $v_0 = v_l$ and *trivial* if $l = 0$. We write that $\mathrm{len}(W) = l$. When $v_0, v_1, \ldots, v_l$ are all distinct, W is a *path*.

A walk $W = v_0 v_1 \cdots v_l$ is *reducible* if $v_{i-1} = v_{i+1}$ for some i, otherwise it is *irreducible*. In the reducible case W can be *reduced* by omitting v_i and v_{i+1} from the sequence, to obtain a shorter walk. After successive reductions are applied to W we must eventually reach an irreducible walk, $\mathrm{red}(W)$. Unsurprisingly, W uniquely determines $\mathrm{red}(W)$ (see Godsil [7]). If $\mathrm{red}(W)$ happens to be trivial we say W is *totally reducible*. Note that totally reducible walks are necessarily closed and have even length. We also stress that the trivial walk is itself totally reducible.

For the walk $W = v_0 v_1 v_2 \cdots v_l$ we define $W_i = v_0 v_1 \cdots v_i$ to be a *prefix* of W, consisting of the first $i+1$ vertices in the walk. If $\mathrm{red}(W_i)$ is a path for each i then W is called *tree-like*. A tree-like walk is closed if and only if it is totally reducible. Although we have defined a walk as a list of vertices, we define the graph induced by the walk to be the graph induced by the edges $\{v_0, v_1\}, \{v_1, v_2\}, \ldots, \{v_{l-1}, v_l\}$. Note that it is possible for a tree-like walk to induce a cycle, as shown in Figure 1.

The number of closed tree-like walks in G can be expressed, for small lengths, as a polynomial in n, κ and the number of certain subgraphs of the graph. The need to know this polynomial has arisen in two enumeration problems: [8] and [13]. In both cases the authors established the polynomials that they needed by 'experimental' means, namely analysing specific examples to get enough equations so that the unknown coefficients could be inferred. The purpose of this paper is to show how the polynomials can be derived directly using generating functions. Avoiding the 'experimental' approach is more satisfying, and may prove vital if the calculations in [8] or [13] are ever generalized or pushed to greater accuracy. It particular (see (6.4)), we can now derive infinite classes of coefficients – something which was not possible with the other method. Such progress may help in extremal problems involving matrix permanents [12, 16, 17], which directly apply the theory from [8] that we are improving.

Let u_r be the number of totally reducible walks of length $2r$ in G. McKay [11] showed that

$$u_r = 2n \sum_{j=0}^{r} \binom{2r}{j} \frac{2r-2j+1}{2r-j+1} \kappa^j \tag{1.1}$$

since G is $(\kappa+1)$-regular on $2n$ vertices. Let w_r be the number of closed tree-like walks of length $2r$ in G. Also, let $\varepsilon_r = u_r - w_r$ be the number of totally reducible walks in G

https://doi.org/10.1017/S0963548309990678 Published online by Cambridge University Press

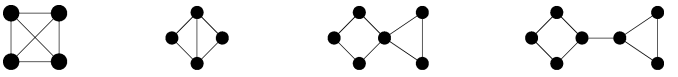


Figure 2. Examples of the contributor notation.

of length $2r$ which are not tree-like. The subgraph induced by such a walk necessarily contains a cycle. The primary goal of this paper is to investigate means of calculating ε_r (and hence, given (1.1), to also find w_r). One reason to be interested in finding w_r is that it determines a moment of the roots of the matching polynomial for G. This will allow us to generalize results from, *e.g.*, [1], [2], [6], [15]. For details, see Section 6.

2. Contributors

As mentioned in the Introduction, for each r, we can express ε_r as a polynomial in n, κ and the numbers of various small subgraphs of G. We call these subgraphs *contributors* (a formal definition will be given below). The coefficients in the polynomial were previously known to be rational numbers, but we will show in Section 7 that they must in fact be integers.

We now introduce some conventions for how we will refer to contributors. We will use a superscript $^\circ$ to indicate that a letter represents a contributor, and will use the same letter without the superscript to denote the number of copies that G has of the contributor.

We use C_i° to denote a cycle of length i, K_i° to denote a clique of order i and $K_{i,j}^\circ$ to denote the complete bipartite graph with colour classes of size i and j. We also use $\theta_{a,b,c,\ldots}^\circ$ to denote a graph with two vertices that are joined by internally disjoint paths of lengths $a, b, c, \ldots$ (where the length of a path is the number of edges in it). Finally, we use $C_{i\cdot j}^\circ$ to denote an i-cycle and a disjoint j-cycle which have had one vertex from each cycle identified, and C_{i-j}° to denote the graph formed by adding an edge between an i-cycle and a disjoint j-cycle. Examples of these notations have been drawn in Figure 2. In some cases we will encounter contributors that are not covered by any of these notations, in which case we will simply draw a picture of the subgraph.

In the case when G is bipartite it has no cycles smaller than C_4°, so $\varepsilon_r = 0$ for $r < 4$. Godsil and McKay [8] used a computer to calculate ε_r for $r \leqslant 6$ in this case, getting:

$$\begin{aligned} \varepsilon_r &= 0 \quad \text{for } r < 4, \\ \varepsilon_4 &= 8C_4, \\ \varepsilon_5 &= 80\kappa C_4, \\ \varepsilon_6 &= 528\kappa_4^{2C} + 12C_6 - 48\theta_{2,2,2}. \end{aligned} \tag{2.1}$$

In the general (that is, not necessarily bipartite) case, McLeod [13] found that:

$$\begin{aligned} \varepsilon_r &= 0 \quad \text{for } r < 3, \\ \varepsilon_3 &= 6C_3, \\ \varepsilon_4 &= 48\kappa C_3 + 8C_4, \\ \varepsilon_5 &= 270\kappa^2 C_3 + 80\kappa C_4 + 10C_5 - 40\theta_{2,2,1}. \end{aligned} \tag{2.2}$$

https://doi.org/10.1017/S0963548309990678 Published online by Cambridge University Press

Table 1. Number of contributors.

# edges	All	Bipartite
3	1	0
4	1	1
5	2	0
6	5	2
7	9	1
8	22	5
9	60	6
10	164	18
11	494	26
12	1624	89

We can now define contributors. Intuitively, a contributor is a possible subgraph of G which merits a separate variable in equations like (2.1) and (2.2). Formally, a *contributor* is a connected graph with minimum degree at least 2. We call contributors *unicyclic* or *polycyclic* depending on whether they consist of a single cycle or contain more than one cycle respectively. Only contributors with at most r edges can affect ε_r. In particular, for any given r there are only finitely many relevant contributors. As a technical point, we will see later (in Figure 5) that not all connected graphs with minimum degree 2 and at most r edges affect ε_r. However, we still consider these graphs to be contributors, even though their coefficient may be zero.

The number of contributors with r edges is given in Table 1 for $r \leqslant 12$. The same table also gives the number of bipartite contributors. It is clear that if G is known to be bipartite then many of the contributors cannot occur, and there will be significantly fewer variables to deal with.

It is worth commenting on why we have defined contributors the way we have. We count non-tree-like walks by considering the graph that they induce, but not all of these graphs are contributors. Any walk must induce a connected graph, but that graph may have vertices of degree one. Suppose G is a connected graph and H is the graph formed iteratively from G by removing a vertex of degree 1 until no such vertices remain. Our method is to define a formal notion of *extensions* of walks. We then count any non-tree-like walk W that induces G by considering it to be an extension of a walk that induces the contributor H. Of course, W may also be an extension of a walk that induces a contributor that is a proper subgraph of H. So to determine the coefficient of H in ε_r we apply a kind of inclusion–exclusion to ensure that W is counted exactly once overall.

3. Preliminary results

Throughout this paper, generating functions will use the variable x to keep track of the length of walks. Since the walks of greatest interest to us are totally reducible and hence consist of edges being walked along in the 'forward' and later the 'backward' direction, we use the power of x to count the 'forward' edges, *i.e.*, half the total length of the walk. Whenever we say that a function F is a generating function for a class of walks we mean

https://doi.org/10.1017/S0963548309990678 Published online by Cambridge University Press

that $[x^m]F$ is the number of walks of length $2m$ in the class. Here and henceforward, $[x^m]F$ denotes the coefficient of x^m when F is expanded as a power series in x. Similarly, we will use $[C]F$ to denote the coefficient of C (to the first power) in F, where C is the variable for a contributor.

We begin by proving some simple results which we will need in subsequent sections. The first involves a generating function for closed walks in an infinite, nearly regular tree. The tree is *rooted* in the sense that it has a distinguished vertex called the *root* and walks in the tree are *rooted* if they begin at the root. Given its natural context, our result has quite possibly been proved before. Certainly, a similar result was obtained by Quenell [14] (see also Chung and Yau [3]).

Lemma 3.1. *Let $\mathcal{T}$ be an infinite rooted tree in which the root has degree r and every other vertex has degree $\kappa+1$. The generating function for closed rooted walks in $\mathcal{T}$ is*

$$T_r = \frac{2\kappa}{2\kappa - r + r\sqrt{1-4x\kappa}}. \tag{3.1}$$

Proof. Any non-trivial closed rooted walk in $\mathcal{T}$ decomposes as follows:

(1) the walk moves from the root to a neighbour v,
(2) some closed walk which avoids the root is undertaken,
(3) the walk returns to the root from v,
(4) a closed rooted walk is undertaken.

As there are r choices for v and a unique trivial walk, it follows that

$$T_r - 1 = rxT_\kappa T_r. \tag{3.2}$$

Solving (3.2) in the case when $\kappa = r$ gives

$$T_\kappa = \frac{1-\sqrt{1-4x\kappa}}{2x\kappa}. \tag{3.3}$$

(We have considered the limit as $x \to 0$ to decide which root of the quadratic to take.) Substituting this value for T_κ into (3.2) then allows us to find T_r. □

We now want to consider two ways to extend a walk $W = w_0 w_1 \cdots w_\ell$. The first is to add *diversions*. We say that W' is formed by adding a diversion to W if $W' = w_0 \cdots w_i d w_{i+1} \cdots w_\ell$ where $w_i d$ is a totally reducible walk (of any length $\geqslant 0$), and if $i < \ell$ then no prefix of $w_i d$ reduces to $w_i w_{i+1}$.

In our next result we stress that saying a walk reduces to W does not imply that W cannot be further reduced.

Lemma 3.2. *Let $W = w_0 w_1 \cdots w_\ell$ be a walk of length ℓ in G. If W' reduces to W then there is a unique way to write $W' = w_0 d_0 w_1 d_1 w_2 d_2 \cdots w_\ell d_\ell$, where each $w_i d_i$ is a diversion. Hence the generating function for walks in G that can be reduced to W is $x^{\ell/2} T_\kappa^\ell \, T_{\kappa+1}$.*

Proof. If W' reduces to W then it can be written as $w_0 d_0 w_1 d_1 w_2 d_2 \cdots w_\ell d_\ell$ where each $w_i d_i$ is totally reducible. Suppose there is some i for which $w_i d_i$ is not a diversion, and

https://doi.org/10.1017/S0963548309990678 Published online by Cambridge University Press

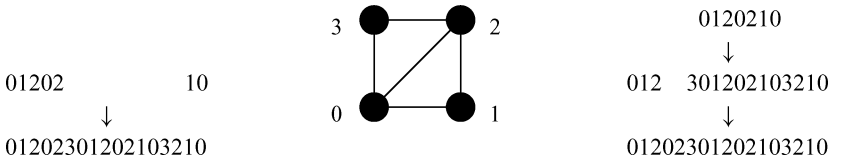


Figure 3. Non-uniqueness of extensions.

choose the smallest such i. Then $w_i d_i = w_i d'_i w_{i+1} u$ where $w_i d'_i$ and $w_{i+1} u w_{i+1}$ are totally reducible walks. Hence, we can replace our original choices of d_i and d_{i+1} by, respectively, d'_i and $u w_{i+1} d_{i+1}$. As d'_i is shorter than d_i, a finite number of such adjustments will lead to an expression for W' in terms of diversions, as required.

Suppose W' could be written in two different ways, for example $w_0 d_0 w_1 d_1 w_2 d_2 \cdots w_\ell d_\ell$ and $w_0 d'_0 w_1 d'_1 w_2 d'_2 \cdots w_\ell d'_\ell$, where $w_i d_i$ and $w_i d'_i$ are diversions for each i. Let $j = \min\{i : d_i \neq d'_i\}$. Without loss of generality, suppose $w_j d_j$ has length l_j and is shorter than $w_j d'_j$. Then the prefix of $w_j d'_j$ of length $l_j + 1$ is $w_j d_j w_{j+1}$, which reduces to $w_j w_{j+1}$, a contradiction.

The generating function for $w_i d_i$ is T_κ for $i < \ell$ and is $T_{\kappa+1}$ for $i = \ell$. Stitching together the $w_i d_i$ takes ℓ edges, which explains the factor of $x^{\ell/2}$. □

The second way of extending W is called adding a tail, and requires W to be closed. We say a walk W' has been formed by *adding a tail* of length t to W if $W' = u_1 u_2 \cdots u_t W u_t \cdots u_2 u_1$, where $u_1 u_2 \cdots u_t w_0$ is an irreducible walk of length t and neither $u_t w_0$ nor $w_0 u_t$ occurs anywhere in W. For convenience, we also consider W to be formed by adding diversions or adding a tail (of length 0) to itself.

We say that a walk W' is an *extension* of a walk W (or that W *extends* to W') if W' can be formed by adding diversions to a walk that itself has been formed by adding a tail to W. Note that if the tail has positive length this includes the possibility of diversions being added to the tail. If W is totally reducible then so is any extension of W.

Lemma 3.2 showed that if W' can be formed by adding diversions to W, then this can happen in only one way. There is no corresponding uniqueness result for extensions. In Figure 3 we show that the walk 01202301202103210 can be obtained in two quite distinct ways by extending the walk 0120210. One way is to simply add a diversion and the second is to add a tail of length 4, and then add a diversion.

We know from Lemma 3.2 that the only ambiguity derives from the choice of tail. Hence, when we speak about the number of different ways in which a walk W' can be extended to a walk W, we will mean the number of different tails that can be added to W' so as to produce a walk that can have diversions added to it to produce W.

The generating function in the following lemma might be said to be improper in that it counts some objects more than once.

Lemma 3.3. *Suppose W is a closed walk in G of length 2ℓ and let C° be the subgraph induced by W. Suppose the first vertex of W has degree δ in C°. The generating function*

https://doi.org/10.1017/S0963548309990678 Published online by Cambridge University Press

for walks in G that are extensions of W is

$$\psi(\ell,\delta) = x^\ell T_\kappa^{2\ell} T_{\kappa+1}\left(\frac{1-(\delta-1)xT_\kappa^2}{1-\kappa x T_\kappa^2}\right),$$

where each walk is counted the number of times that it can be formed by extending W.

Proof. By Lemma 3.2, the walks given by adding diversions to W are generated by $x^\ell T_\kappa^{2\ell} T_{\kappa+1}$. So suppose we add a tail $u_1u_2\cdots u_tw_0$ of length $t \geqslant 1$. We have $\kappa+1-\delta$ choices for u_t since it has to be adjacent to w_0 in G but not in C°. Then for each of $u_{t-1}, u_{t-2}, \ldots, u_1$ in turn we have κ choices, since the tail must be irreducible. We thus have $(\kappa+1-\delta)\kappa^{t-1}$ choices for the tail. Once we have chosen the tail, Lemma 3.2 tells us how many ways we can add diversions. Thus the generating function for extensions of W is

$$\begin{aligned}\psi(\ell,\delta) &= x^\ell T_\kappa^{2\ell} T_{\kappa+1} + \sum_{t\geqslant 1}(\kappa+1-\delta)\kappa^{t-1}x^{\ell+t}T_\kappa^{2\ell+2t}T_{\kappa+1}\\ &= x^\ell T_\kappa^{2\ell} T_{\kappa+1}\left(1+(\kappa+1-\delta)\sum_{t\geqslant 1}\kappa^{t-1}x^tT_\kappa^{2t}\right)\\ &= x^\ell T_\kappa^{2\ell} T_{\kappa+1}\left(1+(\kappa+1-\delta)\frac{xT_\kappa^2}{1-\kappa xT_\kappa^2}\right), \qquad (3.4)\end{aligned}$$

which simplifies to the required expression. □

4. Walks around a single cycle

Our overall aim is to develop techniques to count totally reducible walks that are not tree-like. In order to be non-tree-like a walk must 'wrap itself round a cycle' in a particular way. We now define some more terminology to help us formalize this notion.

We say that a walk W *envelops a cycle at its ith step* if either $\mathrm{red}(W_i) = v_1v_2\cdots v_cv_1$ or $\mathrm{red}(W_i) = u_1\cdots u_dv_1v_2\cdots v_cv_1$, where $v_1,\ldots,v_c$ are distinct vertices, $d \geqslant 1$ and $u_d \neq v_c$. We define the number of cycles enveloped by W to be the number of distinct i for which W envelops some cycle at its ith step. A walk is tree-like if and only if it envelops no cycles.

As an example, the totally reducible walk $W = 0123423051503243210$ envelops just one cycle, since $\mathrm{red}(W_5) = 012342$ envelops the cycle 2342. Note that W does not envelop cycles at either its 6th or 7th step: $\mathrm{red}(W_6) = 0123423$ does not envelop 3423 and $\mathrm{red}(W_7) = 01234230$ is closed, but not a cycle.

Our aim in this section is, for each c and r, to find $[C_c]\varepsilon_r$, the coefficient in ε_r of the number of c-cycles. In practice, $[C_c]\varepsilon_r$ is the number of non-tree-like walks of length $2r$ in an infinite $(\kappa+1)$-regular graph in which there is a unique cycle and that cycle has length c. These coefficients correctly count non-tree-like walks in G that envelop a single cycle, but walks that envelop several cycles get counted multiple times. The role of the polycyclic contributors in the next section is to rectify this overcount.

Theorem 4.1. *Suppose H is an infinite $(\kappa+1)$-regular graph containing a single cycle C°, that has length c. Let X_c be the generating function for totally reducible walks in H which*

https://doi.org/10.1017/S0963548309990678 Published online by Cambridge University Press

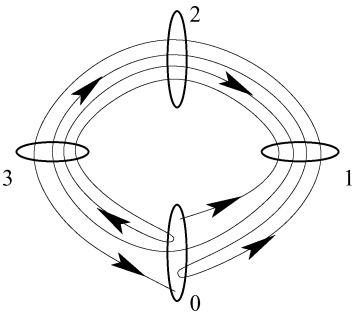


Figure 4. 'Double tour' around a 4-cycle 01230 starting at 0.

envelop C°. Then

$$X_c = \frac{2cx^c T_\kappa^{2c+1}}{(2-T_\kappa)\left(1+x^c T_\kappa^{2c}\right)}, \tag{4.1}$$

where T_κ is given by (3.3).

Proof. For a walk W in H that envelops C°, let i^* denote the lowest value of i such that W envelops C° at its ith step. Then red(W_{i^*}) must consist of the unique irreducible walk from the initial vertex of W to C°, followed by a lap of C°. Suppose W is an extension of another totally reducible non-tree-like walk W'. The properties of tails prevent this from happening by adding to W' any tail that includes an edge of the unique cycle C°. So, unlike the situation in Figure 3, there is no ambiguity involving the choice of tail. In other words, W can be formed *in a unique way* by extending W'.

Arbitrarily assign an orientation to C°. Fix a vertex v of C° and let u be the neighbour of v in the anticlockwise direction. By symmetry, the number of walks which envelop C° is $2c$ times the number of walks W for which red(W_{i^*}) finishes with uv. If red(W_{i^*}) finishes with uv then W's first tour of C° must go clockwise from v. Such choices of W are counted by $\psi(c,2)$, except that this function counts some walks which make an anticlockwise tour of C° as a diversion before making the required clockwise tour. Figure 4 shows the shortest example of the problem in the case $c=4$.

We can count the walks which make double tours in the same way that we counted the single tour walks. In doing so we will overcount some walks which make triple tours. Inevitably, we get an inclusion–exclusion formula,

$$\begin{aligned} X_c &= 2c(\psi(c,2) - \psi(2c,2) + \psi(3c,2) - \psi(4c,2) + \cdots) \\ &= 2cT_{\kappa+1}\left(\frac{1-xT_\kappa^2}{1-\kappa x T_\kappa^2}\right)\left(x^c T_\kappa^{2c} - x^{2c}T_\kappa^{4c} + x^{3c}T_\kappa^{6c} - \cdots\right) \\ &= 2cT_{\kappa+1}\left(\frac{1-xT_\kappa^2}{1-\kappa x T_\kappa^2}\right)\left(\frac{x^c T_\kappa^{2c}}{1+x^c T_\kappa^{2c}}\right). \end{aligned} \tag{4.2}$$

Now using (3.2) we find that $T_{\kappa+1}(1-xT_\kappa^2) = T_\kappa$. Substituting this into (4.2) gives (4.1). □

https://doi.org/10.1017/S0963548309990678 Published online by Cambridge University Press

As an example of Theorem 4.1 we find that X_4 has the following expansion, the first three terms of which were given in (2.1):

$$\begin{aligned} X_4 = {} & 8x^4 + 80\kappa x^5 + 528\kappa^2 x^6 + 2912\kappa^3 x^7 + (14560\kappa^4 - 8)x^8 + (68544\kappa^5 - 144\kappa)x^9 \\ & + (310080\kappa^6 - 1520\kappa^2)x^{10} + (1364352\kappa^7 - 12320\kappa^3)x^{11} \\ & + (5883768\kappa^8 - 85008\kappa^4 + 8)x^{12} + (24996400\kappa^9 - 526240\kappa^5 + 208\kappa)x^{13} + O(x^{14}). \end{aligned}$$

So far we have only examined the contributions from unicyclic contributors, but even they give an upper bound on ε_r, since all non-tree-like walks get counted at least once. Hence $\varepsilon_r \leqslant [x^r] \sum_{i \leqslant r} C_i X_i$ for arbitrary r. This bound will work best for sparse graphs and small r.

5. Polycyclic contributors

In this section we describe an algorithm which allows the contributions from polycyclic graphs to ε_r to be calculated for small values of r. The idea is to define a class of walks, which we call *essential walks*, such that every non-tree-like totally reducible walk can be obtained by extending at least one essential walk. We then use Lemma 3.3 to count the walks that are extensions of essential walks. This approach greatly reduces the number of walks that need to be considered and hence allows the computations to go further. Although the definition of an essential walk is a little technical, the idea is that they contain all the important steps of the walk (in the sense that they result in cycles being enveloped), without any redundancy (steps that can be omitted without reducing the number of cycles enveloped). Generating functions are then used to add all possible extensions. The formal definition is as follows. A walk $W = w_0 w_1 \cdots w_\ell$ is *essential* if the following three conditions are satisfied.

(a) W is totally reducible and non-tree-like.
(b) There exists i in the range $0 < i < \ell$ such that $w_0 = w_i = w_\ell$.
(c) If $w_{j-1} = w_{j+1}$ for j in the range $0 < j < \ell$, then W envelops a cycle at its jth step, and $\mathrm{len}(\mathrm{red}(W_{j-1})) + 1 = \mathrm{len}(\mathrm{red}(W_j)) = \mathrm{len}(\mathrm{red}(W_{j+1})) + 1$.

The definition of an essential walk has been designed to ensure that such walks do not have any 'redundancy'. For example, condition (c) ensures that there is no diversion that can be omitted from an essential walk E without decreasing the number of cycles enveloped. Similarly, if E can be formed by adding a tail of positive length to a walk E', then E' must envelop strictly fewer cycles than E. This is guaranteed by condition (b), given that there are no 'unnecessary' diversions. The condition on the lengths of the reduced walks in (c) is necessary to rule out the alternative hypothesis that $\mathrm{len}(\mathrm{red}(W_{j-1})) = \mathrm{len}(\mathrm{red}(W_j)) + 1 = \mathrm{len}(\mathrm{red}(W_{j+1}))$. Without this extra condition, for example, 012302020203210 would be essential. It is not desirable that the back-and-forth motion along the edge $\{0, 2\}$ in this example be allowed in an essential walk (it becomes possible when we extend the walk). A key property of essential walks is given in our next result.

https://doi.org/10.1017/S0963548309990678 Published online by Cambridge University Press

Theorem 5.1. *Let W be any totally reducible, non-tree-like walk. Then there exists an essential walk $\mathcal{E}(W)$ such that, for all essential walks E, the number of ways to extend E to form W equals the number of ways to extend E to form $\mathcal{E}(W)$.*

Proof. In this proof (a), (b) and (c) will refer to the conditions defining essential walks. Observe by taking $E = \mathcal{E}(W)$ that the conditions of the theorem require W to be an extension of $\mathcal{E}(W)$. Define $\mathcal{E}(W)$ to be the walk created by starting with W and performing these steps:

(i) while (c) fails, reduce the current walk at the first violation of (c); then
(ii) while (b) fails, remove the first and last edges of the walk.

Since W envelops at least one cycle and neither (i) nor (ii) changes the number of cycles enveloped, we must finish with a walk satisfying (a), (b) and (c). That is, $\mathcal{E}(W)$ is essential.

To show that $\mathcal{E}(W)$ has the other required property, we induct on the number of edges removed in steps (i) and (ii). The base case is that W is essential, in which case $\mathcal{E}(W) = W$ trivially works as required. So suppose W is not essential.

In the first instance suppose that $W = w_0 w_1 \cdots w_l$ fails (c) at a point j. Let E be any essential walk that extends to W and let R denote the edges of W that are removed to form E. (In the context of this proof, edges have an inherent position in W, and edges in different positions in the walk are different even if they represent the same edge in the underlying graph.) We distinguish four cases.

Case 1: $w_{j-1}w_j \in R$ and $w_j w_{j+1} \in R$. We can remove $w_{j-1}w_j$ and $w_j w_{j+1}$ from W and apply induction.

Case 2: $w_{j-1}w_j \notin R$ and $w_j w_{j+1} \in R$. Then $\exists i > j$ such that $w_i = w_j$ and $w_j w_{j+1} \cdots w_i$ is a totally reducible walk with all edges in R. If we choose the least such i then $w_{j+1} = w_{i-1}$ and we can replace R by $(R \setminus \{w_{i-1}w_i\}) \cup \{w_{j-1}w_j\}$ and this will not change E. We thereby reduce the problem to Case 1.

Case 3: $w_{j-1}w_j \in R$ and $w_j w_{j+1} \notin R$. This works by a similar argument to Case 2.

Case 4: $w_{j-1}w_j \notin R$ and $w_j w_{j+1} \notin R$. This case is impossible. Let E' be the walk obtained from W_{j-1} by removing edges in R. Then $\mathrm{red}(E')$ is a subwalk of $\mathrm{red}(W_{j-1})$. Hence E fails (c) for the same reason that W fails (c), contradicting the fact that E is essential.

Finally, we consider the possibility that W satisfies (c) but fails (b). In this case, the first edge and last edge of W are not present in E, since E is essential. Removing these two edges from W allows us to apply induction. □

We next outline how to calculate an integer, which we call the *score*, for each essential walk up to a given length. These scores are used to rectify any overcounting when finding the contributions from polycyclic graphs to ε_r. In other words, the score of an essential walk is chosen to compensate for the number of times it gets counted as an extension of a shorter walk.

To find the contribution $\texttt{contrib}(C^\circ)$ from a contributor C° with e edges to ε_i for $e \leqslant i \leqslant r$ we follow Algorithm A.

https://doi.org/10.1017/S0963548309990678 Published online by Cambridge University Press

Algorithm A.

```
1  contrib(C°) := 0;
2  for i from e to r
3      for each essential walk E of length 2i such that the edges of E induce C°
4          if the score for E has not been previously calculated then
5              thisscore := 1;
6              for each walk E′ such that E reduces to E′
7                  Subtract from thisscore the sum of the scores of all essential
                   walks E″ ≠ E such that E′ is formed by adding a tail to E″;
8              end for;
9              Store thisscore as the score for E;
10         else set thisscore to the known score for E;
11         end if;
12         Add thisscore · ψ(i, δ) to contrib(C°), where δ is the degree
           in C° of the initial vertex of E;
13     end for;
14 end for;
```

By way of example, the essential walk $E = 0120234143210$ is allocated a score of -1. It begins with a score of 1 in step 5, then has this reduced by two in step 7; one for the choice $E' = 01234143210$ (which loses its tail to become $E'' = 123414321$) and one for the choice $E' = 0120210$ (here $E'' = E'$). In both cases E'' has a score of 1, as would have been computed and stored earlier. Since E induces $\theta^{\circ}_{3,2,1}$ and begins at a vertex of degree $\delta = 2$, its score of -1 means that in step 12 we would subtract $\psi(i, 2)$ from $\texttt{contrib}(\theta^{\circ}_{3,2,1})$.

To calculate the score for an essential walk inducing a contributor C° we need to know the scores for various shorter essential walks. These walks might induce C° or a proper subgraph of C°. Thus we assume that by the time we run Algorithm A for C° we have previously run the algorithm for all contributors that are proper subgraphs of C°. This is easily managed by treating the contributors in order of increasing e. Under this assumption we can show the following.

Theorem 5.2. *At the conclusion of Algorithm A,* $[x^m]\texttt{contrib}(C^{\circ}) = [C]\varepsilon_m$, *for* $m \leqslant r$.

Proof. Every walk counted by $\texttt{contrib}(C^{\circ})$ is totally reducible and non-tree-like. Essential walks have both properties by definition, and extending them cannot destroy either property.

For the purpose of induction we assume the theorem is true for all proper subgraphs of C°. Then for $m \leqslant r$ we define

$$\varepsilon'_m = [x^m] \sum_{D^{\circ} \subseteq C^{\circ}} D \cdot \texttt{contrib}(D^{\circ}), \tag{5.1}$$

where the sum is over all contributors D° that are subgraphs of C°, including C° itself.

Any essential walk E of length $2i$ that induces C° is counted exactly once by ε'_i. We give E an initial score of 1 in step 5. Then in step 7 this score is adjusted for every way

https://doi.org/10.1017/S0963548309990678 Published online by Cambridge University Press

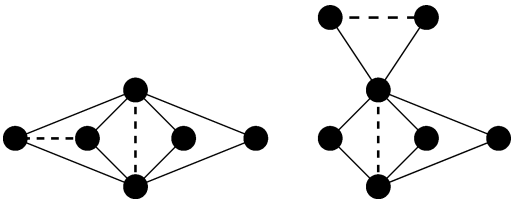


Figure 5. Two contributors with coefficient zero.

that E can be created by extending shorter essential walks. This accounts for all the times we have already counted E by the time we calculate the score for E. The score we then give E ensures that it is counted once overall by ε'_i, given that $[x^i]\psi(i,\delta) = 1$.

Next suppose that W is a totally reducible non-tree-like walk that induces C° but is not essential. Let $\mathcal{E}(W)$ be the essential walk guaranteed by Theorem 5.1. The properties of $\mathcal{E}(W)$ ensure that W is counted by $\varepsilon'_{\mathrm{len}(W)/2}$ the same number of times that $\mathcal{E}(W)$ is counted by $\varepsilon'_{\mathrm{len}(\mathcal{E}(W))/2}$, namely once (according to our preceding argument).

Hence, all walks that induce C° are counted correctly by (5.1). We conclude that $[C]\varepsilon_m = [C]\varepsilon'_m = [x^m]\texttt{contrib}(C^\circ)$ as required. □

Corollary 5.3. *The formula for ε_i derived by running Algorithm A on all contributors with at most i edges, counts each totally reducible non-tree-like walk of length $2i$ once.*

In particular, Algorithm A gives a constructive proof that ε_i can be expressed in terms of polynomials in n, κ and the variables counting occurrences of contributors.

We next make some further remarks regarding Algorithm A. Firstly, it should be noted that step 7 allows $E'' = E'$ (we also allow $E = E'$ in step 6). Secondly, Figure 3 shows why we allow the same walk E'' to be used more than once in step 7 for a given choice of E. By contrast, a walk E' should only ever be used once in step 6 for a given E, even if there are several sequences of reductions that lead from E to E'.

Since scores can be negative or positive, cancellation can result in a contributor with e edges not contributing at all to ε_e. The smallest examples have $e = 10$. For that size, Table 1 records that there are 164 contributors and only 162 of these appear in ε_{10} with a non-zero coefficient. The two which make no contribution to ε_{10} both have non-zero coefficient in ε_{11}. They are shown in Figure 5, if the dashed lines are interpreted as normal edges. Interestingly, if the two dashed lines in either picture are replaced by paths of length two then the resulting bipartite graphs have zero coefficient in ε_{12} and these are the smallest bipartite contributors which have a zero coefficient in ε_e, where e is their number of edges.

The scores for essential walks are stored in step 9. To reduce storage requirements we canonically label walks before storing their scores. A walk is canonically labelled if its vertices are numbered with non-negative integers and if the walk never visits a vertex i before it has visited vertices $0, 1, \ldots, i-1$. It is a simple matter to canonically relabel any walk in order to look up its score, or to store its score. In general, a number of different

https://doi.org/10.1017/S0963548309990678 Published online by Cambridge University Press

Table 2. Counts of essential walks categorized by scores.

Length	Score	Number of walks	Length	Score	Number of walks
6	1	1	16	−1	1144
8	1	1		0	1260
10	−1	10		1	1765
	1	1	18	−1	11108
12	−1	48		0	16105
	0	2		1	12302
	1	13	20	−1	106100
14	−1	181		0	204966
	0	91		1	100172
	1	218			

walks around C° will produce the same canonical walk. This is the reason for the check in step 4, which saves recomputation.

The (canonically labelled) essential walks of length up to 10, together with their scores, are

0120210	1			012303210	1
01202130310	−1	01202303210	−1	01202313210	−1
01203130210	−1	01203230210	−1	01230203210	−1
01230313210	−1	01230320210	−1	01231303210	−1
01231320210	−1	01234043210	1		

Table 2 gives the numbers of canonically labelled essential walks of lengths up to 20, categorized by their scores. The two walks of length 12 that score zero are 0120230203210 and 0123020320210.

We now give some results from the algorithm. As well as confirming the values given in (2.2), it found that:

$$\begin{aligned}\varepsilon_6 &= (1320\kappa^3 - 6)C_3 + 528\kappa^2 C_4 + 120\kappa C_5 + 12C_6 + 192K_4 \\ &\quad - (480\kappa + 12)\theta_{2,2,1} - 48(\theta_{3,2,1} + \theta_{2,2,2} + C_{3\cdot 3}),\end{aligned}$$

$$\begin{aligned}\varepsilon_7 &= (6006\kappa^4 - 84\kappa)C_3 + 2912\kappa^3 C_4 + 910\kappa^2 C_5 + 168\kappa C_6 + 14C_7 + 2688\kappa K_4 \\ &\quad + 84\theta_{2,2,2,1} - (3640\kappa^2 + 168\kappa + 28)\theta_{2,2,1} - (672\kappa + 14)\theta_{3,2,1} - 672\kappa(\theta_{2,2,2} + C_{3\cdot 3}) \\ &\quad - 56(\theta_{3,3,1} + \theta_{3,2,2} + \theta_{4,2,1} + C_{4\cdot 3} + C_{3-3}) + 168\,\text{} + 224\,\text{},\end{aligned} \tag{5.2}$$

$$\begin{aligned}\varepsilon_8 &= (26208\kappa^5 - 720\kappa^2)C_3 + (14560\kappa^4 - 8)C_4 + 5600\kappa^3 C_5 + 1440\kappa^2 C_6 + 224\kappa C_7 + 16C_8 \\ &\quad + (23040\kappa^2 + 720)K_4 + 96(\theta_{3,2,2,1} + \theta_{2,2,2,2}) + (1344\kappa + 192)\theta_{2,2,2,1} \\ &\quad - 64(\theta_{5,2,1} + \theta_{4,3,1} + \theta_{4,2,2} + \theta_{3,3,2}) - (896\kappa + 16)(\theta_{4,2,1} + \theta_{3,3,1}) \\ &\quad - (5760\kappa^2 + 224\kappa + 16)\theta_{3,2,1} - (5760\kappa^2 + 48)\theta_{2,2,2} \\ &\quad - (22400\kappa^3 + 1440\kappa^2 + 448\kappa - 128)\theta_{2,2,1} - 896\kappa(\theta_{3,2,2} + C_{4\cdot 3} + C_{3-3})\end{aligned}$$

https://doi.org/10.1017/S0963548309990678 Published online by Cambridge University Press

$$- 64(C_{5\cdot3} + C_{4\cdot4} + C_{4-3}) - 5760\kappa^2 C_{3\cdot3} + (2688\kappa + 96)\,\text{} + 3584\kappa\,\text{}$$
$$+ 192\Big(\text{} + \text{} + \text{} + \text{}\Big) + 256\Big(\text{} + \text{} + \text{} + \text{} + \text{}\Big)$$
$$- 64\,\text{} - 512\,\text{} - 768\,\text{}.$$

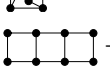
The algorithm easily furnished the next few values in the sequence too, but they become too cumbersome to present in this format. An indication of the number of variables involved is given by Table 1.

The important special case when G is bipartite involves fewer variables. In this case, in addition to the values given in (2.1), we find that:

$$\varepsilon_7 = 2912\kappa^3 C_4 + 168\kappa C_6 - 672\kappa\theta_{2,2,2} - 56\theta_{3,3,1},$$

$$\varepsilon_8 = (14560\kappa^4 - 8)C_4 + 1440\kappa^2 C_6 + 16C_8 + 96\theta_{2,2,2,2} - (896\kappa + 16)\theta_{3,3,1}$$
$$- (5760\kappa^2 + 48)\theta_{2,2,2} - 64(\theta_{4,2,2} + C_{4\cdot4}) + 256\,\text{},$$

$$\varepsilon_9 = (68544\kappa^5 - 144\kappa)C_4 + 9792\kappa^3 C_6 + 288\kappa C_8 - 1152K_{3,3} + 1728\kappa\theta_{2,2,2,2}$$
$$- (39168\kappa^3 + 864\kappa)\theta_{2,2,2} - (8568\kappa^2 + 288\kappa)\theta_{3,3,1} - 72(\theta_{5,3,1} + \theta_{3,3,3} + C_{4-4})$$
$$- 1152\kappa(\theta_{4,2,2} + C_{4\cdot4}) + 4608\kappa\,\text{} + 216\,\text{} + 288\,\text{},$$

$$\varepsilon_{10} = (310080\kappa^6 - 1520\kappa^2)C_4 + 58140\kappa^4 C_6 + 3040\kappa^2 C_8 + 20C_{10} - 23040\kappa K_{3,3}$$
$$+ 480\theta_{2,2,2,2,2} + 120(\theta_{4,2,2,2} + \theta_{3,3,3,1}) + (18240\kappa^2 + 960)\theta_{2,2,2,2}$$
$$- (1440\kappa + 20)\theta_{5,3,1} - (63840\kappa^3 + 3040\kappa^2 + 40)\theta_{3,3,1}$$
$$- (232560\kappa^4 + 9120\kappa^2 - 240)\theta_{2,2,2} - (12160\kappa^2 + 40)\theta_{4,2,2} - 1440\kappa(\theta_{3,3,3} + C_{4-4})$$
$$- 80\Big(\theta_{6,2,2} + \theta_{4,4,2} + C_{6\cdot4} + \text{}\Big) - 12160\kappa^2 C_{4\cdot4} + (48640\kappa^2 + 280)\,\text{}$$
$$+ (4320\kappa + 40)\,\text{} + 5760\kappa\,\text{} - 640\,\text{} - 960\,\text{}$$
$$+ 240\Big(\text{} + \text{} + \text{}\Big) + 320\Big(\text{} + \text{} + \text{} + \text{} + \text{}\Big).$$

6. The matching polynomial

Recall that G is a $(\kappa+1)$-regular graph on $2n$ vertices, but that $2n$ may be odd. For each i let m_i be the number of i-matchings in G (that is, the number of sets of i edges of G in which no two edges share a vertex). By convention, we agree that $m_0 = 1$, regardless of the structure of G. We define the *matching polynomial* of G by

$$\alpha(x) = \sum_{i\geqslant 0} (-1)^i m_i x^{2n-2i}. \tag{6.1}$$

Our terminology follows Diaconis and Gamburd [4] and Godsil [7] and is a specialization of the two variable matching polynomial of Beezer and Farrell [1]. From Heilmann and Lieb [9] we know that $\alpha(x)$ has $2n$ real roots $\lambda_1 \leqslant \lambda_2 \leqslant \cdots \leqslant \lambda_{2n}$ and that $|\lambda_i| < 2\sqrt{\kappa}$ for each i. Also, from Godsil [7] we find that w_r is the $2r$th moment of the roots of $\alpha(x)$.

https://doi.org/10.1017/S0963548309990678 Published online by Cambridge University Press

That is, the number of closed tree-like walks of length $2r$ in G is $w_r = \sum_i \lambda_i^{2r}$. The first few values of $w_r = u_r - \varepsilon_r$ are easily calculated from (1.1), (2.2) and (5.2).

Applying the relationship between the elementary symmetric functions and power symmetric functions (as given, for example, by Macdonald [10, p.23]) to the roots of the matching polynomial, we find that

$$m_r = (-1)^r [x^{2r}] \exp\left(\int_0^x \sum_{i \geqslant 1} w_i (-t)^{2i-1} \, dt \right) = [x^r] \exp\left(- \sum_{i \geqslant 1} \frac{1}{2i} (u_i - \varepsilon_i)(-x)^i \right). \tag{6.2}$$

This allows us to calculate the first few values of m_i, getting

$$\begin{aligned}
m_1 &= n(\kappa + 1), \\
m_2 &= \frac{1}{2} n(\kappa + 1)(n\kappa - 2\kappa + n - 1), \\
m_3 &= \frac{1}{6} n(\kappa + 1)(n^2\kappa^2 - 6n\kappa^2 + 10\kappa^2 + 2n^2\kappa - 9n\kappa + 8\kappa + n^2 - 3n + 2) - C_3, \\
m_4 &= \frac{1}{24} n(\kappa + 1)(n^3\kappa^3 - 12n^2\kappa^3 + 52n\kappa^3 - 84\kappa^3 + 3n^3\kappa^2 - 30n^2\kappa^2 + 96n\kappa^2 - 84\kappa^2 \\
&\quad + 3n^3\kappa - 24n^2\kappa + 55n\kappa - 36\kappa + n^3 - 6n^2 + 11n - 6) - (n\kappa + n - 6\kappa)C_3 + C_4, \\
m_5 &= \frac{1}{120} n(\kappa + 1)(n^4\kappa^4 - 20n^3\kappa^4 + 160n^2\kappa^4 - 620n\kappa^4 + 1008\kappa^4 + 4n^4\kappa^3 - 70n^3\kappa^3 \\
&\quad + 460n^2\kappa^3 - 1300n\kappa^3 + 1152\kappa^3 + 6n^4\kappa^2 - 90n^3\kappa^2 + 475n^2\kappa^2 - 980n\kappa^2 + 648\kappa^2 \\
&\quad + 4n^4\kappa - 50n^3\kappa + 210n^2\kappa - 350n\kappa + 192\kappa + n^4 - 10n^3 + 35n^2 - 50n + 24) \\
&\quad - \left(\frac{1}{2} n^2\kappa^2 - 7n\kappa^2 + 27\kappa^2 + n^2\kappa - \frac{15}{2} n\kappa + \frac{1}{2} n^2 - \frac{1}{2} n \right) C_3 + (n\kappa + n - 8\kappa)C_4 \\
&\quad - C_5 + 4\theta_{2,2,1}, \\
m_6 &= \frac{1}{720} n(\kappa + 1)(n^5\kappa^5 - 30n^4\kappa^5 + 380n^3\kappa^5 - 2580n^2\kappa^5 + 9568n\kappa^5 - 15840\kappa^5 \\
&\quad + 5n^5\kappa^4 - 135n^4\kappa^4 + 1480n^3\kappa^4 - 8160n^2\kappa^4 + 21860n\kappa^4 - 19800\kappa^4 \\
&\quad + 10n^5\kappa^3 - 240n^4\kappa^3 + 2245n^3\kappa^3 - 9930n^2\kappa^3 + 19560n\kappa^3 - 13200\kappa^3 \\
&\quad + 10n^5\kappa^2 - 210n^4\kappa^2 + 1655n^3\kappa^2 - 5925n^2\kappa^2 + 9460n\kappa^2 - 5280\kappa^2 \\
&\quad + 5n^5\kappa - 90n^4\kappa + 595n^3\kappa - 1800n^2\kappa + 2466n\kappa - 1200\kappa \\
&\quad + n^5 - 15n^4 + 85n^3 - 225n^2 + 274n - 120) + \frac{1}{2}(C_3)^2 \\
&\quad - \left(\frac{1}{6} n^3\kappa^3 - 4n^2\kappa^3 + \frac{104}{3} n\kappa^3 - 110\kappa^3 + \frac{1}{2} n^3\kappa^2 - \frac{17}{2} n^2\kappa^2 + 39n\kappa^2 \right. \\
&\quad \left. + \frac{1}{2} n^3\kappa - 5n^2\kappa + \frac{14}{3} n\kappa + \frac{1}{6} n^3 - \frac{1}{2} n^2 + \frac{1}{3} n + \frac{1}{2} \right) C_3 \\
&\quad + \left(\frac{1}{2} n^2\kappa^2 - 9n\kappa^2 + 44\kappa^2 + n^2\kappa - \frac{19}{2} n\kappa + \frac{1}{2} n^2 - \frac{1}{2} n \right) C_4 - (n\kappa + n - 10\kappa)C_5 + C_6 \\
&\quad + 16K_4 - 4C_{3\cdot 3} + (4n\kappa + 4n - 40\kappa - 1)\theta_{2,2,1} - 4\theta_{3,2,1} - 4\theta_{2,2,2}.
\end{aligned} \tag{6.3}$$

https://doi.org/10.1017/S0963548309990678 Published online by Cambridge University Press

It seems the first few formulae above will have been discovered many times, but there was previously no general algorithm for producing them. For example, the bipartite versions of our formulae for m_i have been found by Friedland, Krop and Markström [6] (for $i \leqslant 4$) and Wanless [15] (for $i \leqslant 5$). Farrell, Guo and Constantine [5] found formulae for m_1, m_2, m_3 in graphs that are not necessarily regular.

Beezer and Farrell [1] and [2] give a formula for m_4 equivalent to ours above and show in general that m_i can be written as a linear combination of counts of subgraphs with minimum degree at least 2, with coefficients that are functions of n and κ. The appearance of $(C_3)^2$ in our expression for m_6 does not contradict the linearity claimed by Beezer and Farrell. They do not insist on their subgraphs being connected, so their formula for m_6 would have a term for a subgraph consisting of two disjoint 3-cycles.

In [1] it is shown that $[C_c]m_c = (-1)^c$ for all $c \geqslant 3$. We next generalize this result by finding a generating function for all terms which depend linearly on the number of unicyclic contributors. For any generating function F and contributor C° we define $[C]^*F$ to be the terms in $[C]F$ that depend on n, κ, x only. Thus $[C]^*F$ extracts the terms that depend on C° and no other contributor, and that depend only linearly on the number of copies of C°.

Theorem 6.1. *For any $c \geqslant 3$,*

$$[C_c]^* \sum_{i\geqslant 0} m_i x^i = \log\left(1 + \left(\frac{-2x}{1+2\kappa x+\sqrt{1+4\kappa x}}\right)^c\right) \exp\left(-\sum_{i\geqslant 1} \frac{u_i}{2i}(-x)^i\right).$$

Proof. Recall from (1.1) that u_i is a function of n, κ only. Also, ε_i is a sum of terms each of which is a polynomial in n, κ times a contributor variable. Hence (6.2) implies that

$$[C_c]^* \sum_{i\geqslant 0} m_i x^i = \exp\left(-\sum_{i\geqslant 1} \frac{u_i}{2i}(-x)^i\right) \cdot [C_c]^* \sum_{i\geqslant 1} \frac{1}{2i}\varepsilon_i (-x)^i.$$

By Lemma 4.1,

$$[C_c]^* \sum_{i\geqslant 1} \frac{1}{2i}\varepsilon_i (-x)^i = \int_0^{-x} \frac{c x^{c-1} T_\kappa^{2c+1}\, dx}{(2-T_\kappa)(1+x^c T_\kappa^{2c})} = \log\left(1 + \left(\frac{-2x}{1+2\kappa x+\sqrt{1+4\kappa x}}\right)^c\right),$$

where the integration can be checked by differentiating the claimed answer and routine algebraic manipulation. The result follows. □

Applying Theorem 6.1, we find that, for any $c \geqslant 3$,

$$[C_c]^* \sum_{i\geqslant 0} m_i x^i = (-x)^c \Bigg[1 + (n\kappa + n - 2c\kappa)x + \Bigg(\frac{1}{2}n^2(\kappa+1)^2 - \frac{1}{2}n(\kappa+1)(4c\kappa+2\kappa+1) + c\kappa^2(2c+3)\Bigg)x^2 + O(x^3)\Bigg]. \quad (6.4)$$

Since there are no contributors with fewer than 3 edges, $[C_c]\sum_{i\geqslant 0} m_i x^i$ agrees with (6.4) to within the error term.

https://doi.org/10.1017/S0963548309990678 Published online by Cambridge University Press

7. The polynomial coefficients

In this final section we consider the form of the coefficients in expressions such as (5.2) and (6.3) for ε_r and m_r. We know such coefficients are polynomials. We wish to investigate the degree of these polynomials and to say what we can about their coefficients. In particular we show that the coefficients in ε_r are integers, when it was previously only known that they are rational [13]. Note that certain polynomials with non-integral coefficients map integers to integers. For example, consider $\binom{\kappa}{2}$. So it would be invalid to argue that our polynomials must have integer coefficients simply because they are counting something.

Throughout this section, if the degree of a polynomial evaluates as negative then we agree by convention that it must be the zero polynomial.

Beezer and Farrell [1] and [2] note that the graphs that appear in the formula for m_i have at most i edges and that those with exactly i edges have a coefficient that is independent of n and κ. In fact we can say much more.

Theorem 7.1. *Let r, e be positive integers and C° a contributor with e edges. Then $[C]\varepsilon_r$ is a polynomial in κ with integer coefficients and degree at most $r-e$. Let Γ° be any graph with minimum degree 2 and e edges. Then $[\Gamma]m_r$ is a polynomial in n and κ with rational coefficients and degree at most $r-e$ in either variable.*

Proof. Let $\mathbb{Z}[\kappa]$ denote the ring of polynomials in κ with integer coefficients.

Let i be an arbitrary non-negative integer. We first note by (3.3) that $[x^i]T_\kappa$ is κ^i times the ith Catalan number. In particular, $[x^i]T_\kappa$ is in $\mathbb{Z}[\kappa]$ and has non-negative coefficients and degree i. This last statement is also true for $[x^i]T_{\kappa+1}$, given that (3.2) implies that

$$T_{\kappa+1} = \frac{1}{1-x(\kappa+1)T_\kappa} = \sum_{j\geqslant 0}\big(x(\kappa+1)T_\kappa\big)^j.$$

Hence, by (3.4), $[x^i]\psi(\ell,\delta)$ is in $\mathbb{Z}[\kappa]$ and has degree $i-\ell$. By inspecting Algorithm A, we then see that $[C]\varepsilon_r$ is in $\mathbb{Z}[\kappa]$ and has degree at most $r-e$, as claimed. Note that, by induction on the length of the walk, the score for every essential walk is an integer.

We next consider m_r in the light of what we have just shown about ε_r. Given (6.2), it is immediate that the coefficients in m_r are rational, and it is clear from (6.3) that they are not always integers. Also, from (6.2),

$$m_r = [x^r]\sum_{j=0}^{r}\frac{1}{j!}\left(\sum_{i=1}^{r}\frac{\varepsilon_i-u_i}{2i}(-x)^i\right)^j = [x^r]\sum_{j=0}^{r}\frac{1}{j!}\left(n(\kappa+1)x+\sum_{i=2}^{r}\big(\varepsilon_i O(1)-nO(\kappa^i)\big)x^i\right)^j,$$

where we use $O(\cdot)$ notation to track the powers of n and κ. Now using our new knowledge of ε_i, we can see that $[\Gamma]m_r = O(1)\big(n(\kappa+1)\big)^{r-e} + O(k^{r-e}n^{r-e-1})$. □

Note that the examples given in Figure 5 show that the bounds on the degrees given in Theorem 7.1 are not always attained.

https://doi.org/10.1017/S0963548309990678 Published online by Cambridge University Press

References

[1] Beezer, R. A. and Farrell, E. J. (1995) The matching polynomial of a regular graph. *Discrete Math.* **137** 7–18.
[2] Beezer, R. A. and Farrell, E. J. (1998) Counting subgraphs of a regular graph. *J. Math. Sci.* (*Calcutta*) **9** 47–55.
[3] Chung, F. and Yau, S.-T. (1999) Coverings, heat kernels and spanning trees. *Electron. J. Combin.* **6** R12.
[4] Diaconis, P. and Gamburd, A. (2004) Random matrices, magic squares and matching polynomials. *Electron. J. Combin.* **11** R2.
[5] Farrell, E. J., Guo, J. M. and Constantine, G. M. (1991) On matching coefficients. *Discrete Math.* **89** 203–210.
[6] Friedland, S., Krop, E. and Markström, K. (2008) On the number of matchings in regular graphs. *Electron. J. Combin.* **15** R110.
[7] Godsil, C. D. (1981) Matchings and walks in graphs. *J. Graph Theory* **5** 285–297.
[8] Godsil, C. D. and McKay, B. D. (1990) Asymptotic enumeration of Latin rectangles. *J. Combin. Theory Ser. B* **48** 19–44.
[9] Heilmann, O. J. and Lieb, E. H. (1972) Theory of monomer–dimer systems. *Comm. Math. Phys.* **25** 190–232.
[10] Macdonald, I. G. (1995) *Symmetric Functions and Hall Polynomials*, 2nd edn, Oxford University Press.
[11] McKay, B. D. (1981) The expected eigenvalue distribution of a large regular graph. *Linear Algebra Appl.* **40** 203–216.
[12] McKay, B. D. and Wanless, I. M. (1998) Maximising the permanent of (0,1) matrices and the number of extensions of Latin rectangles. *Electron. J. Combin.* **5** R11.
[13] McLeod, J. C. Asymptotic enumeration of k-edge-coloured k-regular graphs. *SIAM J. Discrete Math.*, to appear.
[14] Quenell, G. (1994) Combinatorics of free product graphs. *Contemp. Math.*, **173** 257–281.
[15] Wanless, I. M. (1999) The Holens–Djoković conjecture on permanents fails. *Linear Algebra Appl.* **286** 273–285.
[16] Wanless, I. M. (1999) Maximising the permanent and complementary permanent of (0,1) matrices with constant line sum. *Discrete Math.* **205** 191–205.
[17] Wanless, I. M. (2003) A lower bound on the maximum permanent in Λ_n^k. *Linear Algebra Appl.* **373** 153–167.

https://doi.org/10.1017/S0963548309990678 Published online by Cambridge University Press