Local generating functions for the enumeration of connected embeddings in a lattice

1986 J. Phys. A: Math. Gen. 19 3425

(http://iopscience.iop.org/0305-4470/19/16/037)

J. Phys. A: Math. Gen. 19 (1986) 3425-3430. Printed in Great Britain

Local generating functions for the enumeration of connected embeddings in a lattice

M K Wilkinson

Department of Physics, King's College, Strand, London WC2R 2LS, UK

Received 22 January 1986

Abstract. A general formalism for the practical application of Sykes' partial generating function method is given. The formalism is suitable for the study of problems where the generating functions depend on detailed 'local' properties and is constructed in a manner which allows easy programming of the method.

1. Introduction

The enumeration of connected embeddings in a lattice is a problem of classic difficulty. The exponential increase in the totals with increasing number of sites or bonds makes direct enumeration a very time-consuming process. Recently Sykes (1986a) has described an indirect approach to the problem utilising the method of partial generating functions (Sykes *et al* 1965).

We consider in this paper the *practical* application of Sykes' method to enumeration problems in general and in particular to problems where detailed properties of the embeddings are recorded. Such properties are *local* in the sense that they depend on the immediate environment of each site; thus the generating functions of Sykes (1986a) become functions of an explicit site labelling. Problems of this type arise, for example, in the enumeration of lattice animals classified by valence distribution and the derivation of series expansions for the mean size in site percolation.

The aim of this paper is to reformulate the partial generating function method in such a way that its application in any particular case is reduced to following a simple general recipe. This new formalism is so constructed that implementation on a computer is a straightforward task.

A detailed account of the method of partial generating functions is given by Sykes (1986a, b) and a general familiarity with the method is assumed here. Briefly the method depends on writing the (restricted) generating functions for connected clusters, g^*, in terms of the more easily constructed partially restricted generating functions g. On a bipartite lattice having two equivalent sublattices A and B, each of N sites, $g_{m,n(A)}$ is defined as the (partially restricted) generating function for clusters whose 'connectedness' is not greater than that specified by m, a partition of the $n(A)$ A sites into connected sets. Each such partition is called a *class* (see § 2). If, for example, $m = \{3, 2, 2, 1\}$ and $n(A) = 8$, then out of eight A sites there are *at most* three connected together as one set, there are two sets each containing *at most* two sites connected together and there is one set which contains only one site. $g^*_{n(A)}$ denotes the enumerator

0305-4470/86/163425+06$02.50

of embeddings with *all* $n(A)$ A sites connected together. By definition

$$g^*_{n(A)} = \sum_{\{\varepsilon_i\}} C_{n(A)}(\alpha) \prod_i \alpha_i^{\varepsilon_i} \tag{1}$$

where $C_{n(A)}(\alpha)$ is the number of connected clusters with $n(A)$ A sites classified by some parameter set $\alpha = \{\alpha_i\}$. Note that in *all* problems one of the α_i is used to record the presence of B sites (i.e. $n(B) = \alpha_1$, say). There is in general a different $g^*_{n(A)}$ for *each* arrangement of $n(A)$ A sites. To recover information about the 'original' lattice one writes for C_n, the number of connected clusters per lattice site,

$$C_n = (1/2N) \sum C_{n(A),n(B)} \tag{2}$$

where the summation extends over all partitions of $n = n(A) + n(B)$ into two parts, and $C_{n(A),n(B)}$, the number of clusters with $n(A)$ A sites and $n(B)$ B sites, can be read off from (1). The power of the partial generating function method lies in exploiting the symmetry relation

$$C_{n(A),n(B)} = C_{n(B),n(A)}. \tag{3}$$

Thus if one can derive $g^*_{n(A)}$ for $n(A) = 1, 2, \ldots, n$ then information on clusters through $2n+1$ sites follows.

The application of Sykes' method is seen to be composed of two distinct parts. Firstly the exhaustive classification of A clusters with $n(A)$ A sites. The information retained in this classification is crucial to the range of applicability of the data. The most general classification would allow the complete reconstruction of all A clusters. The second part of the problem involves forming partially restricted generating functions $g_{m,n(A)}$ for each distinct A cluster and combining them in the manner specified by Sykes (1986a) to yield the g^* for that cluster. Use of (1), (2) and (3) then yields the desired configurational information.

The benefit of splitting the problem into two parts is the reduction in machine counting; to obtain the C_n in (2) say, for some n, one needs to count far fewer A clusters than the direct count C_n. To be set against this is the fact that a large amount of information about each A cluster is required. These data are, however, applicable to several problems and need only be counted once.

2. The labelled code

A detailed consideration of the method outlined in the introduction reveals that the information about A clusters required for the application of Sykes' method, in general, is essentially that contained in the adjacency matrix. It is, however, more convenient to arrange the information in a slightly different form. Since the method is concerned with *connected* clusters it is natural to classify each A cluster by its 'connectivity'. Consider the A cluster on the simple quadratic lattice shown in figure 1. Here b is connected (*within* the A cluster) to the A-site set $\{1, 2\}$, a is connected to the A-site set $\{2, 3, 4\}$, c and d are connected to the A-site set $\{4, 5\}$, e and f are connected to the A-site set $\{5\}$, etc. The number of B sites connected to *only* the A-site set S is conveniently called the *multiplicity* of *bridge* S. Thus bridge $\{1, 2\}$ has multiplicity 1, bridge $\{4, 5\}$ has multiplicity 2, bridge $\{5\}$ has multiplicity 2, and so on.

Clearly, the connectivity of an A cluster is completely specified by a list of $2^{n(A)} - 1$ integers that are indexed by the subsets of the set $\{1, 2, \ldots, n(A)\}$. For the above

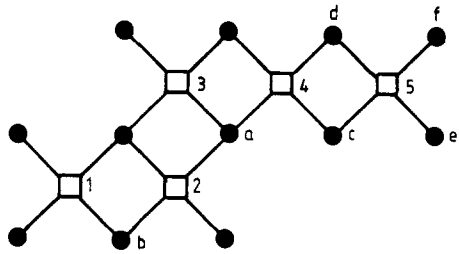

Figure 1. An A cluster on the simple quadratic lattice; □ represents A sites; ● represents B sites.

example the list is

{1}	{2}	{3}	{4}	{5}	{1, 2}	{1, 3}	...	{1, 2, 3, 4, 5}
2	1	1	0	2	1	0		0

This list of numbers is called the *labelled code.* Note that the A-site labelling is arbitrary. In certain enumeration problems not all of the detailed information contained in the

Table 1. Some SC and BCC A-cluster statistics.

	SC		BCC	
Number of A sites	Number of clusters	Number of labelled codes	Number of clusters	Number of labelled codes
1	1	1	1	1
2	9	2	13	3
3	113	5	237	9
4	1 647	25	4 995	61
5	26 121	117	114 219	407
6	437 400	823	2753 781	3823

Table 2. Details of BCC labelled codes.

Number of A sites	Count	Labelled code
1	1	8
2	3	444
2	6	662
2	4	771
3	3	4044040
3	24	4264020
3	24	4374010
3	42	6462020
3	72	6572010
3	28	7671010
3	12	4242022
3	24	4363011
3	8	5551111

labelled code is required and the code may be 'compressed' in various ways. If, for example, the labelling is removed altogether and all bridge multiplicities corresponding to bridges of the same cardinality are summed then the Ising codes of Sykes *et al* (1965) are recovered.

Using standard enumeration techniques (Martin 1974) lists of body-centred cubic (BCC) and simple cubic (SC) labelled codes have been produced through 6 A sites. Some information on the A clusters is given in table 1. It is estimated that to enumerate all labelled codes on the BCC lattice at order 7 would require some tens of hours of computer time. For illustration the first few BCC labelled codes are given in table 2 ($n(A)=1,2,3$). The bridge multiplicities are listed in standard dictionary order (i.e. $\{1\},\{2\},\ldots,\{n(A)\},\{1,2\},\ldots,\{1,2,\ldots,n(A)\}$). Note that the count of each labelled code is small. Such a list is therefore only a small compression of the list of all graphs, and the list is inevitably a long one.

3. The dummy matrix formulation

In this section a general formulation for the partial generating function method is described. Let ν be the set of labelled A sites, $\nu=\{1,2,\ldots,n(A)\}$. Let the ith subset of ν be λ_i (excluding the empty set, $\varnothing$). Denote by $P(\lambda)$ the set of all subsets of λ (excluding $\varnothing$), i.e. the power set of λ. To each element, σ, of $P(\lambda)$ there is an associated *dummy* variable $[\sigma]=\delta$, where δ takes the values 0 or 1. The set of dummy variables is called the *dummy power set* and is denoted by $P'(\lambda)$. The *unrestricted dummy enumerator*, $G_{n(A)}$, is defined by

$$G_{n(A)}(P'(\nu))=\prod_i (1+f(P'(\lambda_i))^{\lambda_i}) \tag{4}$$

where λ_i appearing as an exponent is understood to denote the multiplicity of the corresponding bridge. f is a function of sundry variables depending on which enumeration problem is under consideration. The particular form of f is the *substitution* for that problem. Note that f can carry a lot of information since the sites of the A cluster are explicitly labelled. f is constructed in a way which ensures that the partially restricted generating functions $g_{m,n(A)}$ are obtained by setting appropriate dummy variables to either 1 or 0 in (4). For example, setting $[1,2]=[3,4]=1$ and all other dummy variables to 0 produces the contribution to $g_{\{2,2\},4}$ from the disjoint pair $\{1,2\}$ and $\{3,4\}$. The reason for introducing the labelled code is now clear: each factor in the unrestricted dummy enumerator (4) corresponds uniquely to a bridge (λ_i).

In the fundamental inversion of Sykes (1986a) each unrestricted enumerator is associated with a particular *class*. When $n(A)=4$, for example, there are five classes (see table 3). To obtain g^*, the g are added with certain 'weights', the *inversion weights*, of which there is one for each class. The information that is in practice required for the calculation of g^* is seen to be a list of appropriate dummy variables along with the correct inversion weights. Each 'entry' in the list is a row of $2^{n(A)}-1$ zeros or ones, one for each element of the dummy power set. Such a list can be prepared by simple exhaustive generation of the partitions of ν. The array of dummy variables and inversion weights will be called the *dummy matrix*. This matrix is given to order 4 in table 3. At order 4 the breakdown into classes is also given.

In summary the partial generating function method may be implemented via the following equation:

$$g^*_{n(A)}=\sum{}^* \sum w_j G_{n(A)}(P'(\nu)). \tag{5}$$

Table 3. The dummy matrix.

$n(A)$	Inversion weight	Dummy substitution
1	1	1
2	1	111
2	−1	110
3	1	1111111
3	−1	1111000
3	−1	1110100
3	−1	1110010
3	2	1110000
4	1	111111111111111} ... $g_{\{4\},4}$
4	−1	111111010010000
4	−1	111110101001000 ... $g_{\{3,1\},4}$
4	−1	111101100100100
4	−1	111100011100010
4	−1	111110000100000 ... $g_{\{2,2\},4}$
4	−1	111100110000000
4	2	111110000000000
4	2	111101000000000
4	2	111100100000000 ... $g_{\{2,1,1\},4}$
4	2	111100010000000
4	2	111100001000000
4	2	111100000100000
4	−6	111100000000000} ... $g_{\{1,1,1,1\},4}$

The first summation is over all labelled codes, each term in this summation is weighted by the count of the labelled code. The second summation extends over all rows of the dummy matrix; w_j is the inversion weight of row j.

Note that in the evaluation of (5) it is sometimes advantageous to proceed indirectly. Specifically, when many of the factors appearing in (4) are the same, it is especially useful to define a set of *auxiliary polynomials* (see Sykes 1986a) corresponding to the dissimilar cases, of which there are, say, Δ. Then the labelled code can be compressed to a Δ-parameter 'supercode'. If Δ is small enough, it is feasible to store the supercodes in a hash table and to carry out the polynomial expansions after all A clusters have been processed. Since the expansions are time consuming there is a clear advantage in making the above simplification where possible. Indeed, this is exactly the procedure employed in applications to date (Sykes 1986a, b, c, d, Sykes and Wilkinson 1986).

We conclude this section with some simple examples. For the site animal problem considered by Sykes (1986a), $\alpha = \{x, b\}$, where x records the presence of a B site and b records the presence of a bond. The substitution is given by

$$f(P'(\lambda_i)) = b^{|\lambda_i|} x[\lambda_i]. \tag{6}$$

For the bond animal problem discussed by Sykes (1986b), the substitution is

$$f(P'(\lambda_i)) = x \sum_j b^{|\mu_j|}[\mu_j] \tag{7}$$

where μ_j is the jth subset of λ_i (excluding $\varnothing$). For the bond percolation problem, using the 'yield factor' method (Sykes 1986c), $\alpha = \{x, b, \Lambda\}$ and

$$f(P'(\lambda_i)) = x \sum_j b^{|\mu_j|} \Lambda^{|\lambda_i|-|\mu_j|}[\mu_j]. \tag{8}$$

Here, for $n(A) = 1, 2, \ldots, 6$, $\Delta = 1, 2, 3, 5, 7$ and 11 respectively, as follows from (8) by considering all possible different cases.

It is clear from the above examples that the formalism here given encompasses the cases already studied. These are simple cases in the sense that they do not contain local terms.

4. Conclusions

It is apparent that the enumeration of connected clusters on a lattice can be perceived on several levels. At the highest level, one obtains connected clusters by summing partially restricted generating functions, g, with the correct inversion weights. Then, however, each g is a sum over terms in a particular class of connected sites. Finally, at the lowest level, each term in any class is a product of factors; these factors are polynomials determined by the element of the class to which they correspond. They are indexed by elements of the power set, P, which also indicates the appropriate exponent in the labelled code.

A general recipe for the computer enumeration of connected clusters is then as follows: write down the substitution for the problem in hand and expand according to the dummy matrix, the exponents in the expansion being given by the labelled code. The above formalism can be applied to a number of 'local' problems including the enumeration of both bond and site animals classified by valence distribution and also the derivation of series expansions for the mean size in bipartite site percolation. In these local cases it is not possible to form a compressed code because of the explicit dependence of f on the A-cluster site labelling. It is hoped that applications will be made subsequently.

Acknowledgments

I am very grateful to Dr M F Sykes for helpful criticisms and advice concerning the above. I would also like to thank Dr J L Martin for modifying his cluster counting program to produce the labelled code data. I would like to thank Dr Sykes and Dr D S Gaunt for their critical reading of a draft of this paper. I am grateful to the SERC for financial support.

References

Martin J L 1974 *Phase Transitions and Critical Phenomena* vol 3, ed C Domb and M S Green (New York: Academic) p 97

Sykes M F 1986a *J. Phys. A: Math. Gen.* **19** 1007

—— 1986b *J. Phys. A: Math. Gen.* **19** 1027

—— 1986c *J. Phys. A: Math. Gen.* **19** 2425

—— 1986d *J. Phys. A: Math. Gen.* **19** 2431

Sykes M F, Essam J W and Gaunt D S 1965 *J. Math. Phys.* **6** 283

Sykes M F and Wilkinson M K 1986 *J. Phys. A: Math. Gen.* **19** 3415