Ranks of Trees and Grid Graphs

Jean H. Bevis, Gayla S. Domke, and Valerie A. Miller
Department of Mathematics and Computer Science
Georgia State University
Atlanta, GA 30303-3083 U.S.A.

ABSTRACT. There has been a great deal of interest in relating the rank of the adjacency matrix of a graph to other fundamental numbers associated with a graph. We present two results which may be helpful in guiding further development in this area. First, we give a linear time algorithm for finding the rank of any tree which is twice its edge independence number. Secondly, we give a formula for the rank of any grid graph consisting of mn vertices arranged in a rectangular grid of m rows and n columns on a planar, cylindrical, or toroidal surface.

Introduction

There has been a great deal of interest in relating the rank of the adjacency matrix of a graph, called the rank of the graph for short, to other fundamental numbers associated with a graph. Ranks are easy to compute compared to fundamental numbers whose computation are NP-complete problems. Ranks of graphs also have direct application in electrical networks [9]. Cyriel van Nuffelen [12, 13] conjectured in 1976 that the chromatic number of a graph was bounded above by the rank of its adjacency matrix. This conjecture was further supported by a computer program called Graffiti [6], but N. Alon and P.D. Seymour gave a counter example in 1989 [1]. On the positive side, Cyriel van Nuffelen has shown that the diameter of a graph is bounded above by its rank [14], and also the clique number, radius, and domination number of a graph are bounded above by its rank [15] while the independence number of a graph is bounded above by the rank of the complement of the graph. More recently, S.T. Hedetniemi, D.P. Jacobs, and R. Laskar [8] have shown that the upper open redundance number of a graph is bounded above by its rank.

There seems to be a need for more examples and tools for further development of this area. Mark Ellingham [5] has investigated graphs with no

JCMCC 18 (1995), pp. 109-119

isolated vertices in which no two vertices have identical neighborhoods and no vertex can be added without an increase in the rank of the resulting graph. In this paper we give easy methods for determining the rank of two large classes of graphs.

It is well known that a complete graph on n vertices has rank n, a graph consisting of a single path on n vertices has rank n if is even and rank $n-1$ if n is odd, and a complete bipartite graph has rank 2. Also, a graph consisting of a single cycle on n vertices has rank n if 4 does not divide n otherwise it has rank $n-2$. To these we add results for the ranks of trees and grid graphs. In section 2 we present an algorithm for associating indices with the vertices of a tree from which its rank can be computed. Such an algorithm seems to be necessary since the vertices of a tree (or the lines of its adjacency matrix) can interact in different ways. This algorithm may be considered as an extension of the algorithm given by Mitchell, Hedetniemi, and Goodman [11] for finding a maximal matching in a tree. A grid graph consists of mn vertices arranged in a rectangular pattern of m rows and n columns with edges joining vertices which are horizontally or vertically adjacent in this pattern. In section 3 we give a formula for the rank of a grid graph where the pattern is imposed on a planar, cylindrical, or toroidal surface. Such a formula is possible, in contrast to the result for trees, because of the uniform nature of grid graphs.

Ranks of Trees

A set of edges is said to *independent* if no two edges in the set have a vertex in common. The *edge independence number* of a graph G, denoted by $\beta_1(G)$, is the size of the largest independent set of edges in G. If S is an independent set of edges with $|S| = \beta_1(G)$, then S is called a β_1-*set* or *maximum matching* of G. $|S|$ denotes the cardinality of a set S. Furthermore, if all vertices of G are covered by edges of S, then S is called a perfect matching of G.

Theorem 1. *For any tree T,* rank$(T) = 2\beta_1(T)$.

Proof: Since $A = adj(T)$ is symmetric, rank(T) is the size of the largest principal submatrix of A with a nonzero determinant. But principal submatrices of A are adjacency matrices of subgraphs of T. Thus rank(T) is the number of vertices in the largest subgraph of T with a nonsingular adjacency matrix. Let H be a subgraph of T. According to [7, Theorem 3] the det$(adj(H))$ may be expanded as the sum of determinants of the adjacency matrices of linear subgraphs of H where a linear subgraph is a spanning subgraph whose components are edges or cycles. If H has no such subgraph, det$(ad(H)) = 0$ is the empty sum. Since a subgraph of T has no cycles, a linear subgraph of H is just a perfect matching of H.

Furthermore, by [7] the determinant of the adjacency matrix of such a linear subgraph of H is $(-1)^e$ where e is the number of edges in the linear subgraph. Now if H has rank(T) vertices and $\det(ad(H)) \neq 0$, then all of the rank(T)/2 edges in the perfect matching of H are independent in T so that rank(T) $\leq 2\beta_1(T)$. Conversely, if H is a subgraph of T induced by $\beta_1(T)$ independent edges, then H has exactly one linear subgraph and the $\det(ad(H))$ can be expanded as the sum of one nonzero term. Thus $\det(H) \neq 0$ and rank(T) $\geq$ size of adj(H) $= 2\beta_1(T)$.

Mitchell, Hedetniemi, and Goodman [11] give a linear time algorithm for finding the edge independence number of a tree, and thus by Theorem 1 their algorithm can also be used to find the rank of the tree. Bevis and Hall [2] give a linear time algorithm for finding a specialized depth-first-search ordering of the vertices of a tree such that its adjacency matrix has a factorization in integer matrices. A modification of the initial step of this algorithm, presented here as Algorithm 1, will also produce the rank of a tree. Algorithm 1 is also linear since it can be implemented with one pass of a depth first search of the tree. The basic idea of this algorithm is to find the number s of vertices in a tree T which must be eliminated before the resulting subgraph will have a perfect matching.

Algorithm 1: Given a tree T with n vertices, choose a vertex t to be called the root:

1. Set $s = 0$, and label all leaves $u \neq t$ of the tree as "odd". If $n = 1$, then label t as "odd".

2. Repeat the following until all vertices are labeled. When all children of u have been labeled and u has k odd children

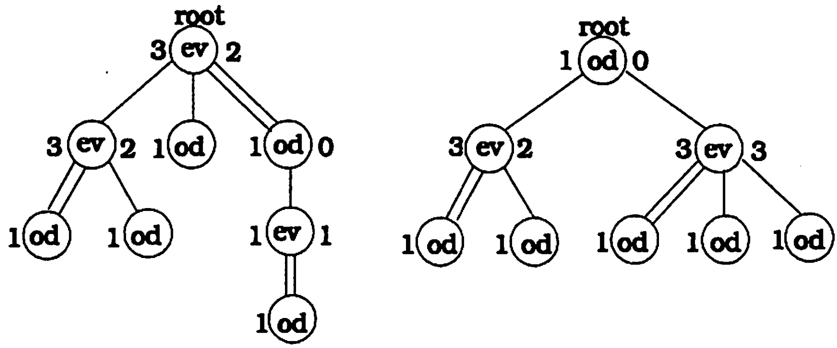
   (2.1) if $k = 0$, then label u as "odd", else

   (2.2) if $k > 0$, then label u as "even" and set $s = s + (k - 1)$.

3. If the root t is labeled odd, then set $s = s + 1$. Set rank(T) $= n - s$.

In (2.1) the odd label indicates that we do not wish to match u with any of its children and will hopefully match it with its parent. In (2.2) u has k odd children which wish to be matched with u. However, in the spirit of Caro and Schoenheim [3], who show that a tree has a perfect matching if and only if at each vertex there is exactly one branch with an odd number of vertices, we can only match u with one of them. This leaves $k-1$ vertices which will not be included in the matching.

It is not possible to find the rank of a tree from only knowledge about the number of odd branches at each vertex. For example, in Figure 1 we show two trees with the same numbers of odd branches at their vertices, namely



(A) (B)

Figure 1

1, 1, 1, 1, 1, 1, 3, 3 but these trees have different ranks. At each vertex the number of branches with an odd number of vertices is shown as a label at the left of the vertex. The edges marked with double lines give a maximal set of disjoint edges and thus determine the ranks of the trees. The tree in Figure 1(A) has rank 6 while the tree in Figure 1(B) has rank 4. The label in the center of each vertex gives the odd "od" or even "ev" labeling resulting from Algorithm 1. The number of odd children is shown at the right of each vertex (except for leaves which have no children). From the result of Caro and Schoenheim, a vertex with more than one odd branch could cause the rank of a tree to be decreased. A tree must have even rank, so the decrease would be two. In the tree of Figure 1(A) there are two vertices which cause a decrease in rank, but these vertices *interact* to cause a total decrease of two. In the tree of Figure 1(B) the two vertices with more than one odd branch act independently, each causing a decrease of two, resulting in a total decrease of four. Roughly speaking, the recursive nature of the labeling process of Algorithm 1 keeps track of the relevant interaction, and thus allows a computation of the rank of a tree.

Algorithm 1 may be considered as an extension of the MHG-algorithm of Mitchell, Hedetniemi, and Goodman [11] which uses a similar labeling process. Consider the graph of Figure 2 where for the moment the region indicated by S is empty. That is we have *leaves* or *endvertices* $u_1, u_2, \ldots u_k$ incident with the vertex v. The MHG-algorithm would label $u_1, u_2, \ldots u_k$ as independent vertices and label v as a dependent vertex. One of the edges,

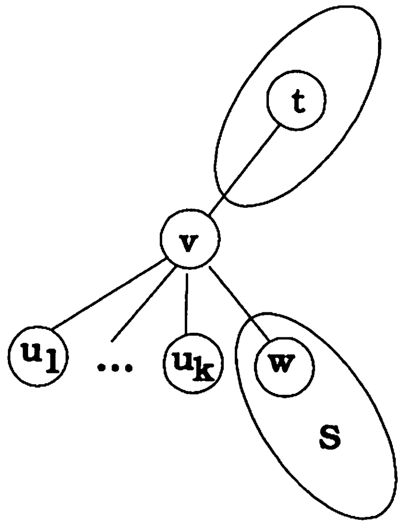


Figure 2

say (u_1, v), would be selected for a β_1-set of the tree. The MHG-algorithm would then select the remaining edges for the β_1-set from the tree with vertices $v, u_1, u_2, \ldots, u_k$ removed so that $u_2, \ldots, u_k$ are $k-1$ vertices not incident with edges of the β_1-set, and as in the proof of Theorem 1, do not contribute to the rank of the tree. Similarly, Algorithm 1 establishes v as an even vertex with k odd children, and in step 2.2 recognizes that $k-1$ of these odd children can not contribute to the rank of the tree. Thus in step 2 of Algorithm 1 we can define indices which are associated with the labeling and give the number of children of even vertices which can not contribute to the rank. That is, for each even vertex u we define the index k_u to be one less than the number of odd children of u, and for each odd vertex u define $k_u = 0$ with the specification that if the root t is labeled odd then $k_t = 1$. With this notation the last part of step 3 of Algorithm 1 can then be stated as "Set rank$(T) = n - \Sigma_{u \in V} k_u$" where V is the vertex set of T. The observation that the MHG-algorithm and Algorithm 1 give the same treatment to the vertices $v, u_1, u_2, \ldots, u_k$ of Figure 2 provides the inductive step in an argument establishing Theorem 2.

Theorem 2. *Algorithm 1 gives the rank of a tree, that is for a tree* $T = (V, E)$ *and indices* k_u, *for* $u \in V$, *defined as above,* rank$(T) = |V| - \sigma_{u \in V} k_u$.

Algorithm 1 and the MHG-algorithm may differ in the order in which they consider vertices. While the MHG-algorithm always works from endvertices inward, Algorithm 1 allows the specification of a root vertex and works from descendants towards the root. These differences can be significant. Consider the graph of Figure 2 with t chosen as the root. Then Algorithm 1 considers the vertices of S, which are all descendants of v, before v is considered while the MHG-algorithm considers v before w is treated. In fact the edge (v, w) could belong to a β_1-set of the tree, but would never be considered so by the MHG-algorithm. Also if the root t has degree 1, then the MHG-algorithm considers it in the first round, while it would be the last vertex considered by Algorithm 1. These differences allow us to establish additional results such as those given in Corollary 1.

Corollary 1. *Suppose that the odd/even labeling of Algorithm 1 has been applied to a tree $T = (V, E)$ with the vertex t as the chosen root, and that a new tree T' is formed by adding a new edge from a vertex $c \in V$, to a new vertex w. If c is labeled even then rank(T') = rank(T), and if $c = t$ is labeled odd then rank(T') = rank$(T) + 2$.*

Proof: Apply Algorithm 1 to T' with t chosen as the root of T'. Let k_u and k'_u denoted the indices obtained in T and T' respectively. Now $k'_w = 0$ since w is labeled odd and is not the root of T'. If c is labeled even then the labeling of common vertices of T and T' is the same so that $k'_c = k_c + 1$, and $k'_u = k_u$ for all $u \neq c$ in V. Hence, when c is even, $\text{rank}(T') = |V| + 1 - \Sigma_{u \in V'} k'_u = |V| - \Sigma_{u \in V} k_u = \text{rank}(T)$. Now if $c = t$ is labeled odd in T, then $k_t = 1$, but t has one odd child w in T', so that t is labeled even in T' and $k'_t = 0$. The labeling on all other vertices is unchanged so that $k'_u = k_u$ for all $u \neq t$ in V. Thus $\text{rank}(T') = |V'| - \Sigma_{u \in V'} k'_u = |V| + 2 - \Sigma_{u \in V} k_u = 2 + \text{rank}T$.

If the new edge of Corollary 1 is added at an odd vertex $c \neq t$, then the rank may or may not increase by two. For example consider the cases where T is a path of length two or three from t to c. Corollary 2 now follows by induction on the number of vertices with Corollary 1 supplying the inductive step. Finally Corollary 3 may be interpreted as saying that Algorithm 1 can be used to check the criteria given by Caro and Schonheim [3].

Corollary 2. *For any positive even integer r and any integer n with $r \leq n$, there is a tree of rank r with n vertices.*

Corollary 3. *Suppose that the n vertices of a tree T have been labeled odd or even by Algorithm 1. Then T has rank n, or equivalently, T has a perfect matching, if and only if the chosen root is labeled even and each even vertex has exactly one odd child.*

Ranks of Grid Graphs

The Cartesian product $G \oplus H$ of graphs G and H, has vertex set $V(G \oplus H) = V(G) \times V(H)$ consisting of ordered pairs of vertices of G and H, and the edge set $E(G \oplus H) = \{\{(a,b),(c,b)\} : \{a,c\} \in E(G), b \in V(H)\} \cup \{\{(a,b),(a,d)\} : a \in V(G), \{b,d\} \in E(H)\}$. In this section we obtain the ranks of graphs which are Cartesian products of paths and cycles. We refer to these as grid graphs since the Cartesian product of two paths gives a rectangular grid on a planar surface, while a product of a path with a cycle or of two cycles gives rectangular grids on a cylindrical and torodial surface, respectively.

The adjacency matrices of Cartesian products of graphs may be expressed in terms of tensor or Kronecker products of matrices. The tensor product of an m-by-n matrix $A = [a_{ij}]$ with a p-by-q matrix B, is a mp-by-nq matrix denoted by $A \otimes B$ which may be partitioned as

$$A \otimes B = \begin{pmatrix} a_{11}B & \dots & a_{1n}B \\ \dots & \dots & \dots \\ a_{m1}B & \dots & a_{mn}B \end{pmatrix}.$$

If G and H are graphs with m and n vertices respectively, it is not difficult to see that $\mathrm{adj}(G \oplus H) = (\mathrm{adj}(G) \otimes I_n) + (I_m \otimes \mathrm{adj}(H))$ where I_n denotes the n-by-n identity matrix. The tensor product of matrices satisfies several elementary properties which may be found in [10]. We summarize these as

Lemma 1. *The tensor product is associative, distributes over matrix addition, and for conformal matrices satisfies the following relationships with ordinary matrix multiplication* $(P \otimes Q)(S \otimes T) = (PS) \otimes (QT)$ *and* $(P \otimes Q)^{-1} = P^{-1} \otimes Q^{-1}$.

For a graph G, let $\eta(G)$ denote the multiplicity of 0 as an eigenvalue of $\mathrm{adj}(G)$. Since adjacency matrices are symmetric, they are similar to a diagonal matrix, and since similarity transformations preserve rank, the rank of a graph G with n vertices can be obtained from $\mathrm{rank}(G) = n - \eta(G)$. From this equality we obtain the following lemma where $|S|$ denotes the cardinality of a set S. It is also clear from this lemma that $\mathrm{rank}(G \oplus H) = \mathrm{rank}(H \oplus G)$ for any two graphs G and H.

Lemma 2. *Let G and H be graphs with m and n vertices respectively. If their eigenvalues are λ_i, $i = 1, 2, \dots, m$ and μ_j, $j = 1, 2, \dots, n$ respectively, then the eigenvalues of $adj(G \oplus H)$ are $\lambda_i + \mu_j$, $1 \le i \le m$, $1 \le j \le n$. Furthermore, $rank(G \oplus H) = mn - |\{(i,j) : \lambda_i + \mu_j = 0.\ 1 \le i \le m, 1 \le j \le n\}|$.*

Proof: Let $D_G = \mathrm{diag}(\lambda_1, \dots, \lambda_m)$ and $D_H = \mathrm{diag}(\mu_1, \dots, \mu_n)$ denote diagonal matrices with the specified diagonal entries. Now there are nonsingular matrices P and Q such that $P\mathrm{adj}(G)P^1 = D_G$ and $Q\mathrm{adj}(H)Q^1 = D_H$. Hence $(P \otimes Q)[\mathrm{adj}(G \oplus H)](P \otimes Q)^{-1} = (P \otimes Q)[\mathrm{adj}(G) \otimes I_n + I_m \otimes \mathrm{adj}(H)](P^{-1} \otimes Q^{-1}) = (D_G \otimes I_n) + (I_m \otimes D_H)$. The result now follows since this is a diagonal matrix with diagonal blocks $\lambda_i I_n + D_H$, $1 \le i \le m$.

Let P_n and C_n denote a path and cycle respectively with n vertices. The eigenvalues of $\text{adj}(P_n)$ are $2\cos\left(\frac{\pi i}{n+1}\right)$, $i = 1, 2, \ldots, n$, and the eigenvalues of $\text{adj}(C_n)$ are $2\cos\left(\frac{2\pi i}{n}\right)$, $i = 1, 2, \ldots, n$ [4, pp. 53 and 73]. Since the cosine function is monotone on the interval $(0, \pi)$, the eigenvalues of a path P_n are distinct. However, a cycle C_n has repeated eigenvalues. Thus it is more convenient to list the eigenvalues of C_n as: 2, -2 if n is even, and each of the values $2\cos\left(\frac{2\pi i}{n}\right)$ with multiplicity 2 for $i = 1, 2, \ldots, \lfloor (n-1)/2 \rfloor$ where $\lfloor x \rfloor$ denotes the largest integer not greater than x. We may now use Lemma 2 to obtain the ranks of some grid graphs.

Theorem 3. *Let P_n denote a path with n vertices and let C_n denote a cycle with n vertices where a cycle is assumed to have at least three vertices. Then*

$$rank(P_m \oplus P_n) = mn + 1 - gcd(m+1, n+1).$$

If $g = gcd(m+1, n)$, then

$$rank(P_m \oplus C_n) = \begin{cases} mn + 1 - g & \text{when } n \text{ is odd} \\ mn + 2 - g & \text{when } g \text{ is even but } n/g \text{ is odd} \\ mn + 2 - 2g & \text{when } n/g \text{ is even.} \end{cases}$$

If $g = gcd(m, n)$ then

$$rank(C_m \oplus C_n) = \begin{cases} mn & \text{when } m \text{ and } n \text{ are both odd} \\ mn + 1 - 2g & \text{when } m \text{ and } n \text{ have opposite parity} \\ mn + 2 - 2g & \text{when } m \text{ and } n \text{ are both even} \end{cases}$$

Proof: Let $\lambda_i = 2\cos\left(\frac{\pi i}{m+1}\right)$ and $\mu_j = 2\cos\left(\frac{\pi j}{n+1}\right)$, $i = 1, 2, \ldots, m$, $j = 1, 2, \ldots, n$ denote the eigenvalues of P_m and P_n respectively. Now $\lambda_i + \mu_j = 0 \Leftrightarrow \cos\left(\frac{\pi i}{m+1}\right) = \cos\frac{\pi(n+1-j)}{n+1} \Leftrightarrow \frac{i}{m+1} = \frac{n+1-j}{n+1}$ since the cosine is monotone on $(0, \pi)$. Let $g = gcd(m+1, n+1)$, $p = (m+1)/g$, and $q = (n+1)/g$. Thus $\lambda_i + \mu_j = 0 \Leftrightarrow i = p(n+1-j)/q$. Now p and q are relatively prime, q divides $n+1$, and i must be an integer so j must be a multiple of q. The only possible values for j satisfying $1 \le j \le n$ are then $j = lq, 2q, \ldots, (g-1)q$. For such values $q \le j \le (g-l)q$ implies $\frac{p}{q}[n+l-(g-l)q] \le i = \frac{p}{q}[n+1-j] \le \frac{p}{q}[n+l-q]$ so that $1 \le p = \frac{p}{q}[n+l-(n+1)+q] \le i \le \frac{m+1}{n+1}(n+1) - \frac{p}{q}q = m+1-p \le m$. We have established that there are exactly $g-1$ pairs (i, j) with $1 \le i \le m$, $1 \le j \le n$, and $\lambda_i + \mu_j = 0$. Hence by Lemma 2, $\text{rank}(P_m \oplus P_n) = mn - (g-1)$.

Next let $\lambda_i = 2\cos\left(\frac{\pi i}{m+1}\right)$ for $i = 1, 2, \ldots, m$ and $\mu_j = 2\cos\left(\frac{2\pi j}{n}\right)$ for $j = 1, 2, \ldots, \lfloor (n-1)/2 \rfloor$ denote distinct eigenvalues of P_m and C_n respectively.

Since $\lambda_i \neq \pm 2$, there is no need to consider the (possible) eigenvalues ± 2 of C_n. Now $\lambda_i + \mu_j = 0 \Leftrightarrow \cos\left(\frac{\pi i}{m+1}\right) = \cos\left(\frac{\pi(n-2j)}{n}\right) \Leftrightarrow \frac{i}{m+1} = \frac{n-2j}{n}$ since the cosine is monotone on the interval $(0, \pi)$. Let $g = gcd(m+1, n)$, $p = (m+1)/g$, and $q = n/g$. Thus $\lambda_i + \mu_j = 0 \Leftrightarrow i = p(n-2j)/q$. As before, this equation has integer solutions only when q divides $2j$. When q is odd j is a multiple of q, and when q is even j is a multiple of $q/2$. The appropriate values of j in these two cases are $j = 1q, 2q, \ldots, \lfloor (g-1)/2 \rfloor q$ and $j = 1(q/2), 2(q/2), \ldots, (g-1)(q/2)$ respectively. As above the reader can verify that these are exactly the values of j for which $1 \leq i \leq m$ and $1 \leq j \leq n$. Now each of the μ_j's has multiplicity two as an eigenvalue of C_n, and according to Lemma 2, the rank of $P_m \oplus C_n$ is $mn - 2\lfloor (g-1)/2 \rfloor$ or $mn - 2(g-1)$ depending on whether $q = n/g$ is odd or even.

We now consider $C_m \oplus C_n$. List the distinct eigenvalues of C_m and C_n respectively as $\lambda_{-1} = -2$ if m is even, $\lambda_0 = 2$, $\lambda_i = 2\cos\left(\frac{2\pi i}{m}\right)$ for $i = 1, 2, \ldots, \lfloor (m-1)/2 \rfloor$ and $\mu_{-1} = -2$ if n is even, $\mu_0 = 2$, $\mu_j = 2\cos\left(\frac{2\pi j}{n}\right)$ for $j = 1, 2, \ldots, \lfloor (n-1)/2 \rfloor$. Now $-2 < \lambda_i, \mu_j < 2$ for $i, j \geq 1$ so we only need to consider the possible relations $\lambda_{-1} + \mu_0 = 0$, $\lambda_0 + \mu_{-1} = 0$, and $\lambda_i + \mu_j = 0$ for $i, j \geq 1$. When $i, j \geq 1$, $\lambda_i + \mu_j = 0 \Leftrightarrow \cos\left(\frac{2\pi i}{m}\right) = \cos\left(\frac{\pi(n-2j)}{n}\right) \Leftrightarrow \frac{2i}{m} = \frac{n-2j}{n}$. Let $g = gcd(m, n)$, $p = m/g$, and $q = n/g$. Thus for $i, j \geq 1$, $\lambda_i + \mu_j = 0 \Leftrightarrow i = p(n-2j)/(2q)$. When m and n are both odd, neither p nor $n - 2j$ is divisible by two and there is no solution. Hence $\text{rank}(C_m \oplus C_n) = mn$ when m and n are both odd. Now suppose that at least one of m and n is even. As before q must divide $2j$. If n is odd (so that m and p are both even) then q is odd, and j must be a multiple of q. The case with m odd and n even is symmetric to this one and is omitted. If m and n are both even and q is odd then j is a multiple of q as before. If m, n and q are all even, then p is odd so that $(n-2j)/q = g - 2(j/q)$ must be even. Now g is even in this last case so j must still be a multiple of q. When m is even the appropriate values of j are $j = 1q, 2q, \ldots, \lfloor (g-1)/2 \rfloor q$ regardless of whether n is odd or even. As above the reader can verify that these are exactly the values of j for which $1 \leq i \leq m$ and $1 \leq j \leq n$. Since the multiplicities of λ_i and μ_j, $i, j \geq 1$, as eigenvalues of C_m and C_m are both two, we obtain four pairs of eigenvalues that sum to zero for each i, j pair such that $i = p(n-2j)/(2q)$. We also have one of the relations $\lambda_0 + \mu_{-1} = 0$ or $\lambda_{-1} + \mu_0 = 0$ when m and n have opposite parity. Thus, in this case, $\text{rank}(C_m \oplus Cn) = mn - [4\lfloor (g-1)/2 \rfloor + 1]$. Finally in the case where m and n are both even we have both relations $\lambda_0 + \mu_{-1} = 0$ and $\lambda_{-1} + \mu_0 = 0$, and in this case $\text{rank}(C_m \oplus C_n) = mn - [4\lfloor (g-1)/2 \rfloor + 2]$.

Corollary 4. *Grid graphs have full rank, that is their rank is the same as* *the number of vertices, in the following cases.*

$Rank(P_m \oplus P_n) = mn$, *if and only if* $m+1$ *and* $n+1$ *are relatively prime.*

$Rank(P_m \oplus C_n) = mn$, *if and only if* $m+1$ *and* n *are relatively prime, or* $gcd(m+1, n) = 2$ *and* $n/2$ *is odd.*

$Rank(C_m \oplus C_n) = mn$, *if and only if* m *and* n *are both odd.*

References

[1] N. Avon and P.D. Seymour, *A Counterexample to the Rank-Coloring Conjecture* J. Graph Theory **13** (1989), 523 –525.

[2] J.H. Bevis and F.J. Hall, *Integer LU-factorizations*, Linear Alg. and Its Applications **150** (1991), 267–286.

[3] Y. Caro and J. Schoenheim, *Generalized* 1*-factorization of a tree* Discrete Math. **33** (1981), 319–321.

[4] D.M. Cvetkovic, M. Doob, and H. Sachs, *Spectra of Graphs*, Academic Press, New York (1979).

[5] M.N. Ellingham, *Basic subgraphs and adjacency matrix rank* (preprint, 1990).

[6] S. Fajtlowicz *On Conjectures of Graffiti II*, Congressus Numeratium **60** (1987), 187–197.

[7] F. Harary *The determinant of the adjacency matrix of a graph* SIAM Review **4** (1962), 202–210.

[8] S.D. Hedetniemi, D.P. Jacobs, and R. Laskar, *Inequalities involving the rank of a graph* J. Combin. Math. Combin. Comput. **6** (1989), 173–176.

[9] M. Iri *A min-max theorem for the ranks and term-ranks of a class of matrices*, Electronics and Communications in Japan **51-A** (1968), 18–25.

[10] P. Lancaster and M. Tismenetsky *The Theory of Matrices* 2nd Ed, Academic Press, Orlando, (1985).

[11] S. Mitchell, S. Hedetniemi, and S. Goodman *Some Linear Algorithms on Trees* Proc. 6th S. E. Conference on Combinatorics, Graph Theory, and Computing, (1975), 467–483.

[12] C. Van Nuffelen *A bound for the chromatic number of a graph*, Amer. Math. Month. **83**, (1976). 265–266.

[13] C. Van Nuffelen *Some bounds for fundamental numbers in graph theory* Bull. Soc. Math. Belg. , **Ser. B 32** (1980), 251–257.

[14] C. Van Nuffelin *Rank and diameter of a graph* Bull. Soc. Math. Belg., **34** (1982), 105–111.

[15] C. Van Nuffelin *The rank of the adjacency matrix of a graph* Bull. Soc. Math. Belg., **Ser. B 35**, (1983), 219 –225.