On Node Ranking of Complete r-partite Graphs

Yung-Ling Lai*and Yi-Ming Chen
Computer Science and Information Engineering,
National Chia-Yi University, Chiayi, Taiwan.

Abstract

A node ranking problem is also called *ordered coloring problem* [6], which labels a graph $G = (V, E)$ with $C : V \to \{1, 2, \cdots k\}$ such that for every path between any two nodes u and v, with $C(u) = C(v)$, there is a node w on the path with $C(w) > C(u) = C(v)$. The value $C(v)$ is called the *rank* or *color* of the node v. Node ranking is the problem of finding minimum k such that the maximum rank in G is k. There are two versions of node ranking problem, off-line and on-line. In the off-line version, all the vertices and edge are given in advance. In the on-line version, the vertices are given one by one in an arbitrary order (say $v_1, v_2, \cdots, v_n$) and only the edges of the induced subgraph $\langle\{v_1, v_2, \cdots v_i\}\rangle_G$ are known when the rank of v_i has to be chosen. This paper establishes the node ranking number of complete r-partite graphs for off-line version and gives a tight bound for on-line version with the algorithms to accomplish them in linear time.

Keywords: on-line node ranking, off-line node ranking, complete r-partite graph

1 Introduction

A graph $G = (V, E)$ is called *k-rankable* if there is a mapping $C : V \to \{1, 2, \cdots k\}$ such that for any path of G with end-vertices u, v and $C(u) = C(v)$, there is a vertex w lies on the path with $C(w) > C(u) = C(v)$. The mapping C is called a *node ranking labeling of G* and the value $C(v)$ is called the *rank* or *color* of the vertex v. The *ranking number* $\chi_r(G)$ of a

*Email: yllai@mail.ncyu.edu.tw. Supposed in part by the National Science Council under grant NSC 95-2115-M-415-004-

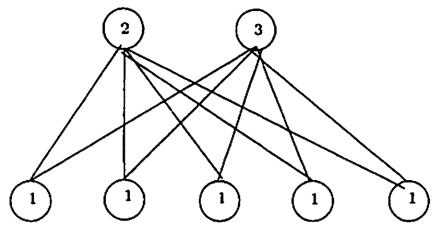

Figure 1: Off-line node ranking of complete bipartite graph $K_{2,5}$.

graph G is the smallest integer k such that G is k-rankable. A node ranking labeling is called an *optimal node ranking labeling of G* if its maximum rank is $\chi_r(G)$. The *node ranking problem* (also called *ordered coloring problem* [6]) is the problem of finding the ranking number $\chi_r(G)$ of a graph G. For an arbitrary graph G, the decision version of the node ranking problem is NP-complete [1]. In fact, this problem remains NP-Complete even with restriction of the graph to be co-bipartite graphs [13]. The node ranking problem is an interested problem and it can be applied on communication network design [17][5][12][15], computing Cholesky factorizations of matrices in parallel [1][4][11], finding the minimum height elimination tree of a graph [17][3], and VLSI layout problem [10][16] etc. There are two versions of node ranking problem, namely off-line and on-line versions. In off-line version, we consider the node ranking problem of a graph with vertices and edges given in advance. On the contrary, the vertices are given one by one in an arbitrary order together with the edges adjacent to the vertices that are already given in the on-line version. Assuming the vertices are given in the order $v_1, v_2, \cdots, v_n$, then only the edges of the induced subgraph $\langle\{v_1, v_2, \cdots v_i\}\rangle_G$ are known when vertex v_i gets its rank assignment. We want to assign the rank of the vertex in real time and the rank of a vertex cannot be changed once it gets an assignment. We use $\chi_r^*(G)$ to denote the on-line ranking number of graph G. Figure 1 and Figure 2 shows an example of node ranking of a complete bipartite graph with off-line and on-line versions respectively.

Node ranking problem has been studied since 1980s. Most of them worked on off-line version in which the whole graph is given beforehand. It is known that $\chi_r(P_n) = \lfloor \log n \rfloor + 1$ for $n \geq 1$ [6] and $\chi_r(C_n) = \lfloor \log(n-1) \rfloor + 2$ for $n \geq 3$ [2]. An upper bound of the ranking number of an arbitrary tree with n nodes was given by [5], they proposed an $O(n \log n)$ time optimal node ranking labeling algorithm of trees, which was further improved to $O(n)$ by [14]. For the on-line version, [2] gave bounds of the node ranking number of paths and cycles as $\lfloor \log n \rfloor + 1 \leq \chi_r^*(P_n) \leq 2 \lfloor \log n \rfloor + 1$ and $\lfloor \log(n-1) \rfloor + 2 \leq \chi_r^*(C_n) \leq 2 \lfloor \log(n-1) \rfloor + 1$ respectively. An on-line

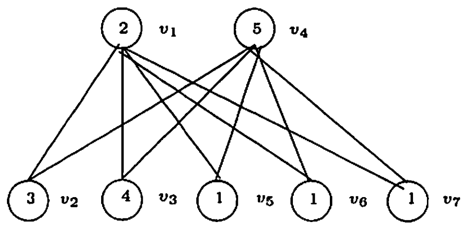

Figure 2: On-line node ranking of complete bipartite graph $K_{2,5}$.

ranking algorithm for trees using CREW PRAM model with $O(n^3/\log^2 n)$ processors was given by [8]. And [9] provided an optimal on-line ranking algorithm for stars and an $O(n^3)$ time algorithm for arbitrary trees. [7] has further improved the algorithm to $O(n^2)$. Although some simple graphs' ranking problems have been solved, we have not seen any discussion about complete r-partite graphs. This paper establishes the node ranking number of complete bipartite graphs for off-line version and gives a tight bound for on-line version with the algorithms to accomplish them in linear time and extends the results to complete r-partite graphs. The rest of this paper is structured as follows. Section 2 solves the node ranking number of complete bipartite graphs for off-line version and extends to complete r-partite graphs. Section 3 gives a tight bound of the node ranking number for on-line versions and presents the algorithm to achieve it in linear time. Then section 4 concludes the whole work.

2 Off-line Version

In this section, we present an off-line optimal node ranking algorithm for complete bipartite graphs.

Algorithm Off_Kmn:
Input: A complete bipartite graph $G = K_{m,n}$ for $m \leq n$ with two partite sets $V_m = \{v_1, v_2, \cdots v_m\}$ and $V_n = \{u_1, u_2, \cdots u_n\}$.
Output: A rank assignment of G.
Method:

1. For all vertices $v_i \in V_m$, $C(v_i) = i + 1$.
2. For all vertices $u_i \in V_n$, $C(u_i) = 1$.

End of Algorithm Off_Kmn

Theorem 1 *Algorithm Off_Kmn produces a node ranking labeling for complete bipartite graph $G = K_{m,n}$ for $m \leq n$ with maximum rank $m + 1$ in linear time.*

Proof: Since each vertex gets its rank according to the partite set it belongs to, the algorithm Off_Kmn takes only linear time. Since only the vertices in partite sets V_n gets the same rank ($= 1$), and all vertices in the other partite set has rank greater than 1, the rank assignment produced by Algorithm Off_Kmn satisfy every path between any two nodes u and v, with $C(u) = C(v)$, there is a node w on the path with $C(w) > C(u) = C(v)$ which implies it produces a node ranking labeling of complete bipartite graphs. □

Lemma 1 comes directly from the definition of node ranking.

Lemma 1 *Let $G = K_{m,n}$ be a complete bipartite graph with $m \leq n$. Let V_m, V_n be two partite sets of G with m and n vertices respectively. Let C be a node ranking labeling of G. Then $C(v) \neq C(u)$ for all $v \in V_m$ and $u \in V_n$.*

It's trivial since every vertex $v \in V_m$ is adjacent to every vertex $u \in V_n$, which implies that $C(v) \neq C(u)$.

Lemma 2 *Let $G = K_{m,n}$ be a complete bipartite graph with $m \leq n$ and V_m, V_n be two partite sets of G with m and n vertices respectively. Let C be a node ranking labeling of G. If two vertices in the same partite set have the same rank r, then all vertices in the other partite set must have different ranks greater than r.*

Proof: Let C be a node ranking labeling of G with $C(u) = C(v) = r$ for some $u, v \in V_m$. Then $\forall x \in V_n$ there is a path $uxv \in G$ which implies $C(x) > C(u) = C(v) \quad \forall x \in V_n$. Now consider $x, y \in V_n$ such that $x \neq y$. Since xuy is an induced path in G, and we already know that $C(u) < C(x)$ and $C(u) < C(y)$, if $C(x) = C(y)$ it will contradict to the fact that C is a node ranking labeling of G, so we have $C(x) \neq C(y)$. □

Lemma 3 *Let G be a graph and $C : V(G) \rightarrow \{1, 2, \cdots k\}$ be an optimal node ranking labeling of G. Then for all i, $1 \leq i \leq k$, there is a vertex $v \in V(G)$ such that $C(v) = i$.*

Proof: Assume that $C : V(G) \rightarrow \{1, 2, \cdots k\}$ is an optimal node ranking labeling of G, such that $\exists i$, $1 \leq i \leq k$ and $\forall v \in V(G)$, $C(v) \neq i$. Then this mapping can be shown as $C : V(G) \rightarrow \{1, 2, \cdots i-1, i+1, \cdots k\}$ and $\chi_r(G) = k$. Consider another labeling C' of graph G, $C' : V(G) \rightarrow \{1, 2, \cdots k-1\}$ such that $C'(v) = C(v)$ for all $v \in V(G)$ with $C(v) < i$ and $C'(v) = C(v) - 1$ for all $v \in V(G)$ with $C(v) > i$. Then if $\exists\, x, y \in V(G)$ such that $C'(x) = C'(y)$, we must have $C(x) = C(y)$. Since C is an optimal node ranking labeling of G, there exists a vertex w such that $C(w) > C(x) = C(y)$. If $C(w) < i$, then $C'(w) = C(w) > C'(x) = C'(y) = C(x) = C(y)$.

If $C(w) > i$ and $C(x) < i$, then $C'(w) = C(w) - 1 \geq i > C'(x) = C(x)$. If $C(w) > i$ and $C(x) > i$, then $C'(w) = C(w) - 1 > C(x) - 1 = C'(x)$. So C' is a node ranking labeling of G with maximum rank less than k which is the node ranking number of G, which is a contradiction. Hence the lemma implies. □

Theorem 2 *Let $G = K_{m,n}$ be a complete bipartite graph with $m \leq n$. Then $\chi_r(G) = m + 1$.*

Proof: By lemma 2, at least one partite set have verities with all different ranks, since $m \leq n$, we have $\chi_r(G) \geq m$. By lemma 1, the vertices in different partite set must have different ranks, which then implies $\chi_r(G) \geq m + 1$. By theorem 1, algorithm Off_Kmn produces a node ranking labeling with maximum rank $m + 1$, which implies $\chi_r(G) \leq m + 1$. Therefore we have $\chi_r(G) = m + 1$. □

Corollary 1 *Let $G = K_{n_1,n_2,\ldots,n_r}$ be a complete r-partite graph with $n_1 \leq n_2 \leq \cdot \leq n_r$. Then $\chi_r(G) = \sum_{i=1}^{r-1} n_i + 1$.*

Proof: By lemma 1, if there are any two vertices u, v have the same labeling, these two vertices must be in the same partite set. By considering these two vertices are adjacent to all vertices in other partite sets, according to lemma 2, all vertices in others partite set must have different ranks which are greater than $C(u) = C(v)$. Since $n_1 \leq n_2 \leq .... \leq n_r$, we have $\chi_r(G) \geq \sum_{i=1}^{r-1} n_i$. By lemma 1, the vertices in different partite set must have different ranks, which then implies $\chi_r(G) \geq \sum_{i=1}^{r-1} n_i + 1$. Consider a labeling C which assign all vertices in the greatest partite set with rank 1 and all other vertices with rank from 2 to $\sum_{i=1}^{r-1} n_i + 1$. Since only the vertices in the partite set with n_r vertices gets rank 1, and all vertices in other partite sets have different ranks greater than 1, C satisfies every path between any two nodes u and v, with $C(u) = C(v)$, there is a node w on the path with $C(w) > C(u) = C(v)$, hence C is a node ranking labeling of G with $\max_{v \in V(G)} C(v) = \sum_{i=1}^{r-1} n_i + 1$. That is $\chi_r(G) \leq \sum_{i=1}^{r-1} n_i + 1$, which then implies $\chi_r(G) = \sum_{i=1}^{r-1} n_i + 1$. □

3 On-line Version

In this section, we propose an on-line ranking algorithm for complete bipartite graphs, then give the bound for complete bipartite graph in on-line version.

Algorithm On_Kmn
Input: Two integers m and n with $m \leq n$, which indicate the number of vertices in each partite set. The complete bipartite graph $K_{m,n}$ where the vertices $v_1, v_2, \cdots v_{m+n}$ are entered one by one with the edges adjacent to those vertices that have been entered.
Output: An on-line rank assignment of G.
Method:

```
If (2m ≥ n) then {
   C(v1) = 1;
   t = 2;
   For i = 2 to m + n do
     If (vi v1) ∈ E(G) then {
       C(vi) = t; t = t + 1;}
     Else C(vi) = 1
}
Else { // 2m < n
   x = 1; // x = |Vx|, v1 ∈ Vx
   y = 0; // y = |Vy|, v1 ∉ Vy
   C(v1) = 2; t = 3;
   For i = 2 to m + n do {
     If (vi v1) ∈ E(G) then {
       y = y + 1;
       If (y > m) then
         C(vi) = 1;
       Else {
         vi = t; t = t + 1;}
     } // vi ∈ Vy
     Else {
       x = x + 1;
       If (x > m) then
         C(vi) = 1;
       Else {
         vi = t; t = t + 1;}
     } //vi ∈ Vx
   } //end for
} //end else
```

End of Algorithm On_Kmn

Lemma 4 *Let $G = K_{m,n}$ be a complete bipartite graph with $m \leq n$. Algorithm On_Kmn produces an on-line node ranking labeling for G in linear time with*

$$\max_{v \in V(G)} C(v) \leq \begin{cases} n+1 & \text{if } 2m \geq n; \\ 2m+1 & \text{otherwise.} \end{cases}$$

Proof: Since each vertex gets its rank assignment for only a constant number of comparisons, algorithm On_Kmn only takes linear time. To see that the algorithm On_Kmn produces an on-line node ranking labeling, we have following two cases.

Case 1: $2m \geq n$. Since all vertices in the same partite set as v_1 gets the same rank 1, and all other vertices in the other partite set gets ranks from 2 to $m+1$ or $n+1$ (depend on which partite set v_1 is), the algorithm has maximum rank either $n+1$ or $m+1$.

Case 2: $2m < n$. In this case, since the only duplicate rank of the vertices is 1 and all these vertices are in V_n, by lemma 2 we know that algorithm On_Kmn produce an on-line node ranking labeling. Since each partite set has m vertices with different ranks greater than 1, there are two partite sets, we have to use $2m+1$ different ranks. That is $\max_{v \in V(G)} C(v) = 2m+1$ for $2m < n$. □

Theorem 3 *Let $G = K_{m,n}$ be a complete bipartite graph with $m \leq n$. Then*

$$m + 1 \leq \chi_r^*(G) \leq \min\{n+1,\ 2m+1\}.$$

Proof: The lower bound is trivial since for all graph G, $\chi_r(G) \leq \chi_r^*(G)$ and by theorem 2, $\chi_r(K_{m,n}) = m+1$. By lemma 4, we know the upper bound is achievable. Hence the proof is completed. □

Corollary 2 *Let $G = K_{n_1,n_2,\ldots,n_r}$ be a complete r-partite graph with $n_1 \leq n_2 \leq \ldots . \leq n_r$. Then*

$$\sum_{i=1}^{r-1} n_i + 1 \leq \chi_r^*(G) \leq \min\left\{\sum_{i=2}^{r} n_i + 1, \sum_{i=1}^{r-1} n_i + n_{r-1} + 1\right\}.$$

Proof: The lower bound is trivial since for all graph G, $\chi_r(G) \leq \chi_r^*(G)$. Similar to the algorithm On_Kmn, when $n_r < n_1 + n_{r-1}$, just assign rank 1 to the vertices with the same partite set of the vertex that entered first, and all others a different rank greater than 1, then we have a node ranking assignment which has maximum rank no more than $\sum_{i=2}^{r} n_i + 1$. If $n_r \geq n_1 + n_{r-1}$, then we assign each vertex different ranks greater than 1 until the biggest partite set has been determined, then assign rank 1 to the vertex of the rest of biggest partite set. Then we have a node ranking assignment which has maximum rank no more than $\sum_{i=1}^{r-1} n_i + n_{r-1} + 1$. Hence, we have $\sum_{i=1}^{r-1} n_i + 1 \leq \chi_r^*(G) \leq \min\left\{\sum_{i=2}^{r} n_i + 1,\ \sum_{i=1}^{r-1} n_i + n_{r-1} + 1\right\}$. □

4 Conclusion

In this paper, we presented both off-line and on-line node ranking algorithms for complete bipartite graphs and extend the results to complete r-partite graphs. The node ranking number of both complete bipartite graphs and complete r-partite graphs are established for off-line version and the tight bounds are given for the on-line version.

References

[1] H.L. Bodlaender, J.R. Gilbert, H. Hafsteninsson and T. Kloks, "Approximating treewidth, pathwidth, frontsize and shortest elimination tree," *Journal of Algorithms* 18, pp. 238-255, 1995.

[2] E. Bruoth and M. Horák, "On-line ranking number for cycles and paths," *Discussiones Mathematicae, Graph Theory* 19, pp. 175-197, 1999.

[3] J.S. Deogun, T. Kloks, D. Kratsch, and H. Muller, "On vertex ranking for permutation and other graphs," *Proc. of the 11th Annual Symposium on Theoretical Aspects of Computer Science*, P. Enjalbert, E.W. Mayr, K.W. Wagner, Lecture Notes in Computer Science 775, Springer-Verlag, Berlin, pp. 747-758, 1994.

[4] I.S. Duff and J.K. Reid, "The multifrontal solution of indefinite sparse symmetric linear equations," *ACM Transactions on Mathematical Software* 9, pp. 302-325, 1983.

[5] A.V. Iyer, H.D. Ratliff and G. Vijayan, "Optimal node ranking of trees," *Inform. Process. Lett.* 28 pp.225-229, 1988.

[6] M. Katchalski, W. McCuaig, and S. Seager, "Ordered colorings," *Discrete Mathematics* 142, pp. 141-154, 1995.

[7] Y.L. Lai, and Y.M. Chen, "An Improved On-line Node Ranking Algorithm of Trees," *Proceedings of the 23rd Workshop on Combinatorial Mathematics and Computation Theory*, pp.345-348, Taiwan.

[8] C.W. Lee and J.S.T. Juan, "Parallel Algorithm for On-Line Ranking in Trees," *Proc. of the 22nd Workshop on Combinatorial Mathematics and Computational Theory*, National Cheng Kung University, Tainan, Taiwan, May 20-21, pp. 151-156, 2005.

[9] C.W. Lee and J.S.T. Juan, "On-Line Ranking Algorithms for Trees," *Proc. of Int. Conf. on Foundations of Computer Science*, Monte Carlo Resort, Las Vegas, Nevada, USA,. June 27-30, pp. 46-51, 2005.

[10] C.E. Leiserson, "Area-efficient graph layouts for VLSI," *Proceedings of the 21st Annual IEEE Symposium on Foundations of Computer Science*, pp. 270-281, 1980.

[11] J.W.H. Liu, "The role of elimination trees in spare factorization," *SIAM Journal of Matrix Analysis and Applications* 11, pp. 134-172, 1990.

[12] J. Nevins, D. Whitney (Eds.), *Concurrent Design of Products and processes*, McGraw-Hill, New York, 1989.

[13] A. Pothen, "The complexity of optimal elimination trees," *Technical Report* CS-88-13, Pennsylvania State University, University Park, PA, 1988.

[14] A.A. Schäffer, "Optimal node ranking of trees in linear time," *Information Processing Letters* 33, pp. 91-96, 1989.

[15] P. Scheffler, "Node ranking and searching on cographs," in: U. Faigle and C. Hoede (Eds.), *3rd Twenty Workshop on Graph and Combinatorial Optimization*, 1993.

[16] A. Sen, H. Deng, and S. Guha, "On a graph partition problem with application to VLSI layout," *Information Processing Letters* 43, pp. 87-94, 1992.

[17] P. Torre, R. Greenlaw, A.A. Schäffer, "Optimal ranking of trees in polynomial time," *Proceedings of the 4th Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 138-144, 1993.