Generating Graph From Given Degree Sequences

Abhishek, Laura, Ravi

January 24, 2007

Abstract

We wish to find if we can generate a graph having red and blue colored edges, whose vertices satisfy 2 degree sequences, each corresponding to the red edges and blue edges respectively.

1 Introduction

Definition 1: A finite sequence $(d_1, d_2, \ldots, d_n)$ of nonnegative integers is called graphic (or realizable) if there is a labeled simple graph with vertex set $(v_1, v_2, \ldots, v_n)$ in which vertex v_i has degree d_i. Such graph is called realization of the given degree sequence $(d_1, d_2, \ldots, d_n)$.

We intend to test whether a given degree sequence is graphic. A simple recursive algorithm to test if the degree sequence is graphic was developed independently by Havel and Hakimi. We state their results in following theorem.

Theorem 1: Let **d**$= (d_1, d_2, \ldots, d_n)$ be a non increasing sequence of nonnegative integers $(n \geq 2)$ and denote the sequence $(d_2-1, d_3-1, \ldots d_{d_1+1}-1, d_{d_1+2}-1, \ldots, d_n) =$ **d'**. Then **d** is graphic if and only if **d'** is graphic.

Proof: It is immediate that if **d'** is graphic then **d** is graphic. Take a realization of **d'** with vertices $v_2, v_3, \ldots, v_n$. Introduce a new vertex v_1 and join v_1 to the d_1 vertices whose degree had 1 subtracted from them.

Now, assume **d** is graphic and let G with vertices $(v_1, v_2, \ldots, v_n)$ be realization of **d**. If vertex v_1 is connected to vertices $v_2, v_3, \ldots, v_{d_1+2}$, we are done, since we can delete v_1 and the corresponding d_1 edges adjacent to v_1

from G. Assume that there exists a vertex v_x which is adjacent to v_1 and v_x doesnt belong to $v_2, v_3, \ldots, v_{d_1+2}$. Let $v_y \in v_2, v_3, \ldots, v_{d_1+2}$ such that v_1 and v_y are not adjacent. If $degree(v_x) = degree(v_y$, we can interchange vertices v_x and v_y wihout affecting the degrees. If $degree(v_x) < degree(v_y)$, then there is a vertex $v_z \neq v_x$ joined by an edge to v_y but not to v_x. Perform a *switch*, by adding the edges (v_1, v_y), (v_z, v_x) and deleting the edges (v_1, v_x), (v_y, v_z). This doesn't affect the degrees. So, we still have the realization of $\mathbf{d}$. Repeating this we obtain a realization of $\mathbf{d}$ where vertex v_1 is joined to d_1 highest degree vertices other than v_1 itself.

Theorem 1 thus gives a recursive test for whether d is graphic: apply the theorem repeatedly until either the theorem reports that the sequence is not graphical (if there are not enough vertices available to connect to some vertex) or sequence becomes zero vector (in which case $\mathbf{d}$ is graphic).

A Polynomial Time Algorithm via Matchings

For $\mathbf{d} = (d_1, d_2, \ldots, d_n)$ as above, let M_d be the following graph. For each $1 \leq i \leq n$, M_d contains a complete bipartite graph $H_i = (L_i, R_i)$, where $|R_i|=n-1$ and $|L_i| = n - 1 - d_i$. The vertices of R_i are labeled so that there is a label for each $1 \leq j \leq n$ other than i; let us denote these labels by $u_{i,1}, \ldots, u_{i,i-1}, u_{i,i+1}, \ldots, u_{i,n}$. In addition, for each $1 \leq i, j \leq n$ with $j \neq i$, M_d has an edge between $u_{i,j}$ and $u_{j,i}$. Now, each perfect matching M of M_d gives rise to a unique realization G of $\mathbf{d}$ in the natural way: G has a link between v_i and v_j if and only if M contains the edge between $u_{i,j}$ and $u_{j,i}$.