Enumerating Non-Graceful Graphs Using Rosa's Parity Condition

Adam Drake (adrake@gmail.com) & Dr. Timothy Redl (redlt@uhd.edu)

Introduction:

A *vertex labeling* of a graph G with *n* vertices and *e* edges is an assignment *f* of labels to each of the vertices that produces for each edge *xy* a label depending on the vertex labels *f(x)* and *f(y)*. A vertex labeling *f* is called a *graceful labeling* of a graph G with *e* edges if *f* is an injection from the vertices of G to the set {0,1,...,*e*} such that when each edge *xy* is assigned the label |*f(x)* – *f(y)*| the resulting edge labels are distinct. If a graceful labeling of G exists, then G is called *graceful*.

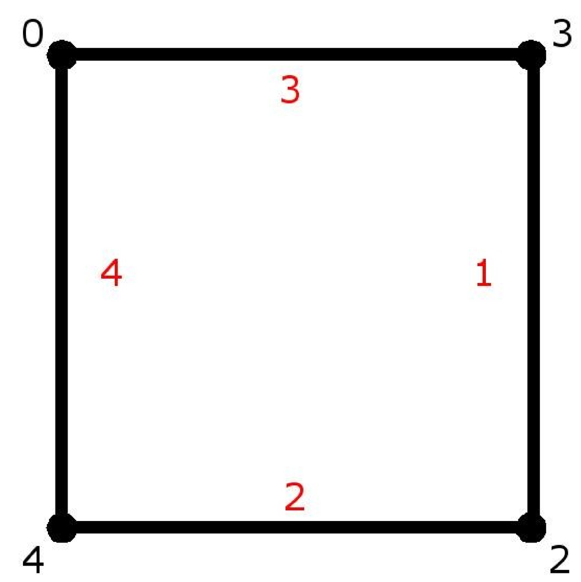

Fig. 1 – C_4 is a graceful graph

Research Motivation:

A popular problem in graph theory is to determine whether a graph G is graceful or not, either by producing a graceful labeling of G or by proving that no such labeling exists. Hundreds of papers have been published on graceful labeling methods, including a recent study on computing graceful labelings of graphs by Redl [4] using techniques of mathematical programming. Most of these papers focus on graceful graphs, but what about non-graceful graphs?

A parity condition by Rosa [5] provides a sufficient condition for a graph to be non-graceful: "If a graph G is simple, even, and has *e* edges, where *e* is congruent to 1 or 2 (mod 4), then G is not graceful." This parity condition was later reformulated as a necessary condition for graceful graphs by Golomb [1] and later by Redl [4]. Rosa's parity condition allows us to recognize and enumerate an infinite subclass of non-graceful graphs, which we define as *R*.

Degree Sequences and Graphic Sequences:

The degree of a vertex of a graph is the number of edges adjacent to that vertex. A degree sequence of a graph G is simply a list of the degrees of the vertices of G. Every graph has a unique degree sequence (with degrees listed in non-increasing order), but a particular degree sequence may represent more than one graph. If a degree sequence represents a simple graph, the sequence is called a *graphic* sequence. Not all degree sequences are graphic sequences. Degree sequences are strictly numerical representations of graphs and consequently, they naturally lend themselves to computational methods.

In Fig. 1 above, we have that *e* = 4 and *n* = 4. The degree sequence of C_4 is [2,2,2,2] because the degree of each of the 4 vertices of C_4 is 2. By the degree sum formula, the sum of the integers in a degree sequence of a graph is equal to twice the number of edges. It is also useful to note that [2,2,2,2] is a partition of 8 consisting of only even numbers. This partition is precisely the degree sequence of C_4.

Enumerating Non-Graceful Graphs in *R*:

Our iterative method for enumerating graphs in *R* proceeds as follows: Given a value of *e*, where *e* is congruent to 1 or 2 (mod 4), we generate all of the integer partitions (i.e., degree sequences) of 2*e* consisting of only even numbers using the partition() function in Maple 10. We plan to change this in the future to be a modified version of Knuth's Partitioning Algorithm P [3]. We then use the Havel-Hakimi Procedure [2] to determine which of these degree sequences are graphic sequences. The resulting graphic sequences are precisely the degree sequences of connected graphs in *R* with *e* edges (i.e., simple, even, with *e* congruent to 1 or 2 (mod 4)). At this time, we are only concerned with connected graphs in *R*; however, we are aware that *R* also contains some graphs which are not connected.

Below is a flowchart diagram of the enumeration procedure and a diagram of the cases where *e* = 5 and *e* = 6.

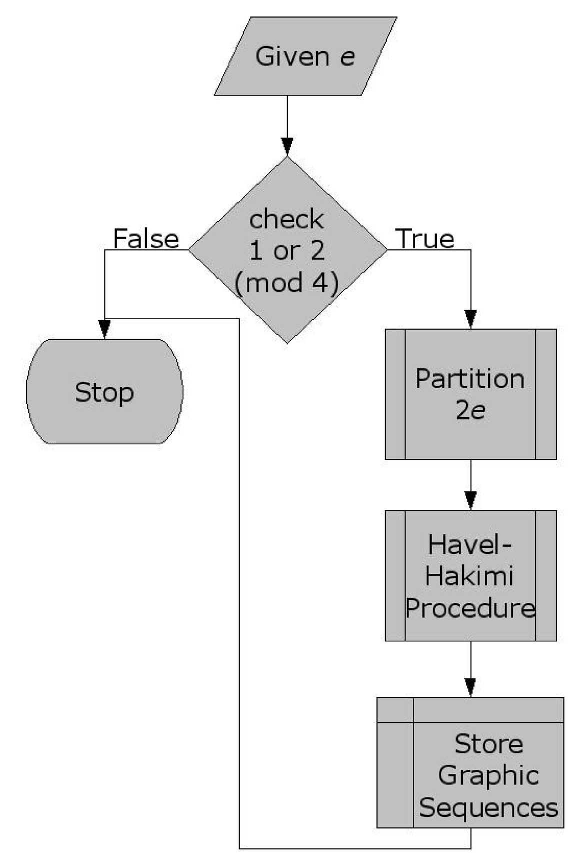

Fig. 2 – A single iteration of the enumeration procedure

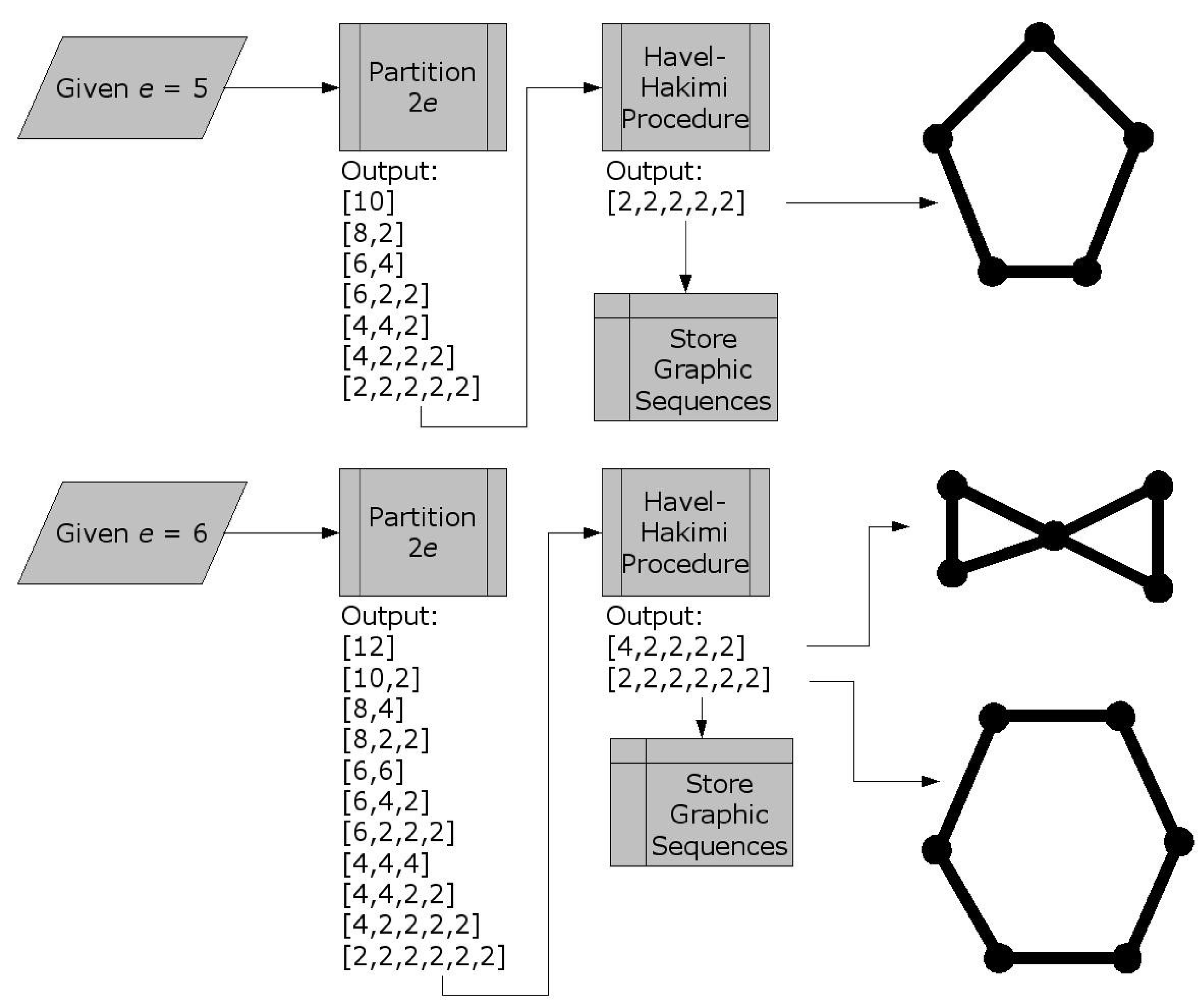

Fig. 3 – An example of the procedure for e=5 and e=6

Results and Further Questions:

With our present method, we can easily enumerate graphic sequences for values of *e* up to 62 on a commodity-hardware laptop. However, we are examining other partitioning algorithms that increase our capacity to work with larger values of *e* on commodity hardware. One of the things we are particularly interested in is looking for patterns or relationships between the value of *e* and the number of non-graceful graphs in *R* that have *e* edges. We plan to implement database storage of graphic sequences in the next version of our implementation.

Again, thus far we have only examined graphs that are connected. Many of the processes we have developed for our current research can be extended to examine graphs that contain isolated vertices or other disjoint subgraphs (the parity condition does not specify connectedness). We have found, for example, that given a graceful graph G, it is possible to "disgrace" G (i.e., make G non-graceful) by simply adding a number of isolated vertices *i* such that *i > e - n + 1*. Additionally, if n > 4 then adding sufficient edges to "complete" *G* (i.e., make *G* a complete graph) will also disgrace *G*.

In addition to considering disconnected graphs, we are also curious about using a distributed computing model for the enumeration process. The process we currently have naturally lends itself to discretization because we generate degree sequences for one value of *e* at a time. In a distributed computing environment, each processing unit could follow the process independently for a given *e*, and send its results back to a central location for archiving and further evaluation. This would allow us to generate a large number of graphic degree sequences in a short amount of time by having multiple processing units examine individual values of *e* simultaneously instead of a single computer examining them one at a time.

Another way to increase the speed of enumeration would be to improve the algorithms we are using. The partition function in Maple is relatively fast, but if we use Knuth's Algorithm P we have more options for distributed computing. In addition, it is possible that there is another that can perform the partitioning even faster than Algorithm P. In addition, we seek to improve our implementation of the Havel-Hakimi procedure, by integrating it with the partitioning algorithm, to increase the speed with which degree sequences are evaluated.

We look forward to continuing our work in this area, including developing the previously mentioned modifications to our procedure, and ultimately making further contributions to the field of graph labeling.

The authors thank the University of Houston-Downtown, the Scholars Academy, and the U.S. Army Research Office (grant W911NF-1-0024) for providing the funding for this research.

References:

[1] Golomb, S. W. *How to Number a Graph*, **Graph Theory and Computing**, R. C. Read, ed., Academic Press: 1972, 23-37
[2] Hakimi, S. L. *On the Realizability of a Set of Integers As Degrees of the Vertices of a Graph*, **SIAM J. Appl Math 10 (1962)** 496-506
[3] Knuth, D. E. **The Art of Computer Programming, Vol. 4, Fascicle 3,** Addison-Wesley Professional: 2005
[4] Redl, T. *Graceful Graphs and Graceful Labelings: Two Mathematical Programming Formulations and Some Other New Results*, **Congressus Numerantium 164 (2003)**, 17-31
[5] Rosa, A. *On Certain Valuations of the Vertices of a Graph*, **Theory of Graphs, International Symposium,** Rome, July 1966, Gordon and Breach: 1967, 349-355

University of Houston-Downtown
Department of Computer & Mathematical Sciences
www.cms.dt.uh.edu