2008

A computational investigation of graph reconstruction

David Rivshin

Master's Project Report

A Computational Investigation of Graph Reconstruction

David Rivshin
dfr2936@cs.rit.edu

August 5, 2007

Rochester Institute of Technology
Department of Computer Science
102 Lomb Memorial Drive
Rochester, NY 14623-5608

Project Committee

Chair: Stanisław P. Radziszowski _____

Reader: Edith Hemaspaandra _____

Observer: Piotr Faliszewski _____

Abstract

First proposed in 1941 by Kelly and Ulam, the *Graph Reconstruction Conjecture* has been called the major open problem in the field of Graph Theory. While the Graph Reconstruction Conjecture is still unproven it has spawned a number of related questions. In the classical vertex graph reconstruction number problem a vertex is deleted in every possible way from a graph G, and then it can be asked how many (both minimum and maximum values) of these subgraphs are required to uniquely reconstruct G (up to isomorphism). This problem can then be extended to k-vertex deletion (for $1 \leq k \leq |V(G)|$), and to k-edge deletion (for $1 \leq k \leq |E(G)|$). For some classes of graphs there is known a formula to directly compute its reconstruction numbers. However, for the vast majority of graphs the computation devolves to brute force exhaustive search.

Previous computer searches have computed the 1-vertex-deletion reconstruction numbers of all graphs of up to 10 vertices, as well as computing 2-vertex-deletion reconstructibility of all graphs on up to 9 vertices. In this project I have developed and implemented an improved algorithm to compute 1-vertex-deletion reconstruction numbers with an $O(|V(G)|)$ speedup, allowing their computation for all graphs of up to 11 vertices. In addition the ability to compute arbitrary k-vertex and edge deletion reconstruction numbers has been implemented, leading to many new results in these areas.

Contents

1 Introduction

In 1941 Kelly and Ulam proposed the Graph Reconstruction Conjecture, and it has remained an open problem to this day [3]. The Graph Reconstruction Conjecture states simply that any simple finite undirected graph on 3 or more vertices can be uniquely identified (up to isomorphism) by the multiset of its unlabeled 1-vertex-deleted subgraphs. There are no known counter-examples to the Conjecture, and it is widely believed to be true, although yet unproven; see [1]. However, for some classes of graphs the conjecture has been proven to hold; specifically disconnected graphs, regular graphs, trees, and maximal planar graphs [14, 2, 15, 1]. Through exhaustive computer search it has previously been shown that all graphs of between 3 and 11 vertices are reconstructible [10, 12], and that some classes of graphs of up to 16 vertices are reconstructible [10].

From the original Graph Reconstruction Conjecture, there has sprung many new related problems. In 1964 the Edge Reconstruction Conjecture was formulated, which state that all simple undirected graphs with 4 or more edges can be uniquely identified (up to isomorphism) by the multiset of its unlabeled 1-edge-deleted subgraphs [8]. In 1957 Kelly introduced the concept of k-vertex-reconstruction, where k vertices are deleted to form each subgraph, rather than 1. Kelly further proved that for $k > 0$ there is a graph on $2k$ vertices which is not k-reconstructible (see [11]), although reconstructibility for larger graphs is unproven.

More recently the question "if a graph is reconstructible, how many of its subgraphs are required to reconstruct it?" has been asked. This takes two forms, the *Existential (*or *Ally) Reconstruction Number* ($\exists rn$), and the *Universal (*or *Adversarial) Reconstruction Number* ($\forall rn$). The Existential Reconstruction Number is the minimum number of subgraphs required if they are carefully selected, while the Universal Reconstruction Number is the minimum number such that any multiset of subgraphs of that size can reconstruct the original graph. Similarly, the same question can be asked, but with the addition of the knowledge that a graph is in a certain class, leading to the concept of Class Reconstruction Numbers ($\mathcal{C}rn$). The concept of Edge Reconstruction Numbers exists as an edge-deleted analogue to what is more specifically called Vertex Reconstruction Numbers (vrn). Taken all together these concepts result in a veritable zoo of reconstruction numbers, many of which little is known about.

The goal of this project was to directly compute various reconstruction numbers for as many graphs as feasible. Given the many different possible reconstruction numbers, this goal became multifaceted. The primary facet was the computation of 1-vertex-deleted reconstruction numbers on all graphs of up to 11 vertices, which was a significant computational challenge. An important secondary goal was the computation of k-vertex-deleted reconstruction numbers for $2 \leq k \leq |V(G)| - 2$, which were computed for all graphs on up to 8 vertices. For all graphs on 9 vertices k-vertex-deleted reconstruction numbers were computed for $2 \leq k \leq 3$. In addition, a great many k-edge-deleted reconstruction numbers were computed for graphs on not more than 10 vertices. The results of these computations are presented in Section 6.

2 Terminology

2.1 Graph Notation

Traditional graph notation is primarily used in this report, and basic familiarity with graph theory and notation is assumed. In all cases graphs are assumed to be simple, undirected, and finite. Furthermore, graphs which are isomorphic are considered identical, except where explicitly noted. Additionally, $\mathcal{G}_n$ refer to the set of all graphs on n vertices, and $\mathcal{G}_{n,e}$ refers to the set of all graphs on n vertices which have exactly e edges. The following common graphs and combinations are also defined:

K_n: Clique on n vertices

P_n: Path on n vertices

S_n: Star on n vertices

C_n: Cycle on n vertices

mG: the disconnected graph which consists of $m \in \mathbb{Z}_{\geq 0}$ copies of the graph G

$G \cup H$: the disconnected graph which consists of one copy each of G and H

2.2 Set Notation

Graph reconstruction involves many instances of set and multiset manipulation. Traditional set notation is used troughout this report, as well as less common multiset constructs. A multiset is differentiated from a set by the use of square brackets ([]) instead of curly brackets ({}), by stating that a quantity is a multiset explicitly, or implicitly if it is the result of operations on a multiset. Each element x of a multiset $\mathcal{S}$ is said to have a multiplicity, given by $m(S;\, x) \in \mathbb{Z}_{>0}$. The number of unique elements in a multiset is given by $\|\mathcal{S}\|$, while the cardinality is given by $|\mathcal{S}| = \sum_{x \in \mathcal{S}} m(\mathcal{S};\, x)$. If $\mathcal{S}$ is a set then the equality $|\mathcal{S}| = \|\mathcal{S}\|$ always holds by definition.

The intersection ($\bigcap$) and union ($\bigcup$) of multisets preserves the minimal and maximal multiplicity of matching elements, while the additive union ($\biguplus$) sums the multiplicities of matching elements. The power set of a set $\mathcal{S}$ is represented by $\mathcal{P}(\mathcal{S})$, and consists of all sets which are a subset of $\mathcal{S}$. The power multiset of a set $\mathcal{S}$ represented by $\mathcal{M}_t^n(\mathcal{S})$, consists of all subsets of $\mathcal{S}$ where elements have multiplicities in the range $[1..n]$ and the sum of all the multiplicities is in the range $[0..t]$. It is self-evident that for a given n all values of $t \geq n|S|$ are equivalent, and for a given t all values of $n \geq t$ are equivalent.

In this report the notation $\mathcal{M}(\mathcal{S})$ shall be taken to mean $\mathcal{M}_\infty^n(\mathcal{S})$ for some finite, but unspecified, $n \in \mathbb{N}^*$. Furthermore, the notation $\mathcal{M}_t(\mathcal{S})$ shall be taken to mean $\mathcal{M}_t^t(\mathcal{S})$.

2.3 Graph Reconstruction

The multiset of all vertex deleted subgraphs is referred to as $Deck_k(G)$ or $D_k(G)$, where k is the number of vertices deleted from graph G to create each subgraph. Each $C \in Deck_k(G)$ is referred to as a $Card_k$ of G. Each $S \subseteq Deck_k(G)$ is referred to as a legitimate $Subdeck_k$ of G. The inverse operation to $Deck_k(G)$ is $Extensions_k(F)$, which is the set of all graphs where k vertices are added in every possible way to graph F.

$\mathcal{E}Deck_k(G)$, $\mathcal{E}Card_k$, $\mathcal{E}Subdeck_k$, and $\mathcal{E}Extensions_k(F)$ are the edge-deletion analogues of vertex-deletion definitions.

The $\exists vrn_k$ is the size of the smallest $Subdeck_k$ of G which can uniquely reconstruct G. The $\forall vrn_k$ is the smallest number such that all $Subdeck_k$ of G of size $\forall vrn_k$ can uniquely reconstruct G. Similarly $\exists ern_k$ is the size of the smallest $\mathcal{E}Subdeck_k$ of G which can uniquely reconstruct G, and $\forall ern_k$ is the smallest number such that all $\mathcal{E}Subdeck_k$ of G of size $\forall ern_k$ can uniquely reconstruct G. $\exists \mathcal{C}vrn_k(G)$, $\exists \mathcal{C}ern_k(G)$, $\forall \mathcal{C}vrn_k(G)$, and $\forall \mathcal{C}ern_k(G)$ have the same definition as $\exists vrn_k(G)$, $\exists ern_k(G)$, $\forall vrn_k(G)$, and $\forall ern_k(G)$, with the added knowledge that G is a member of some class $\mathcal{C}$.

A graph G is said to be k-reconstructible (k-edge-reconstructible) if it can be uniquely identified (up to isomorphism) from its full $Deck_k(G)$ ($\mathcal{E}Deck_k(G)$). If a graph G is not k-reconstructible (k-edge-reconstructible), then it is said that $\exists vrn_k(G) = \forall vrn_k(G) = \infty$ ($\exists ern_k(G) = \forall ern_k(G) = \infty$).

In this report if neither vertex nor edge reconstruction is specifically stated, then the statement applies to all forms of graph reconstruction. Similarly, if no specific value k for number of deletions is given, then the statement applies to all values of k which could apply given the context of the statement. Note that this differs from conventional notation, where 1-vertex-deletion is assumed if not otherwise specified. When neither existential nor universal qualifier is given to reconstruction numbers, then the statement applies to both, again, contrasted against the normal assumption of existential reconstruction number.

2.4 Reconstruction Function Pairs

As a generalization of graph reconstruction, one can speak of reconstruction over any pair of (f_D, f_E), which have the following properties:

1. $f_D : \mathcal{S}_1 \to \mathcal{M}(\mathcal{S}_2)$
2. $f_E : \mathcal{S}_2 \to \mathcal{P}(\mathcal{S}_1)$
3. $x_2 \in f_D(x_1) \iff x_1 \in f_E(x_2)$

Such functions shall henceforth be referred to as a *reconstruction function pair*. For example, the following are valid reconstruction function pairs, as described in Section 2.3:

$$\forall(n \geq k) \begin{pmatrix} Deck_k : & \mathcal{G}_n \to \mathcal{M}_{\mathrm{C}(n,k)}(\mathcal{G}_{n-k}) \\ Extensions_k : & \mathcal{G}_{n-k} \to \mathcal{P}(\mathcal{G}_n) \end{pmatrix}$$

$$\forall(n \geq 2, e \geq k) \begin{pmatrix} \mathcal{E}Deck_k : & \mathcal{G}_{n,e} \to \mathcal{M}_{\mathrm{C}(e,k)}(\mathcal{G}_{n,e-k}) \\ \mathcal{E}Extensions_k : & \mathcal{G}_{n,e-k} \to \mathcal{P}(\mathcal{G}_{n,e}) \end{pmatrix}$$

3 Reconstruction Algorithm

In order to determine both universal and existential reconstruction numbers the same primitive question is asked: "can a given *Subdeck* reconstruct G uniquely (up to isomorphism)?" In order for a *Subdeck* to not reconstruct G there must be another graph H which shares that same *Subdeck*. Therefore in order to answer the question, either an example of a graph which shares the same *Subdeck* must be found, or it must be proven that no such graph exists. The results presented in this report answer that question by computational search.

3.1 Previous Algorithm

The following is a simplified version of the algorithm which was used by Brian McMullen [11] to compute $\forall vrn_1(G)$ and $\exists vrn_1(G)$:

1. compute the multiset $\mathcal{D}_G \leftarrow Deck_1(G)$
2. compute set of possible matches $\mathcal{H} \leftarrow \left(\bigcup_{C \in \mathcal{D}_G} Extensions_1(C)\right) - G$
3. for each $H_i \in \mathcal{H}$ compute the multiset $\mathcal{D}_{H_i} \leftarrow Deck_1(H_i) \cap \mathcal{D}_G$
4. compute $\forall vrn_1(G) \leftarrow 1 + \max(|\mathcal{D}_{H_i}|,\ H_i \in \mathcal{H})$
5. compute $\exists vrn_1(G) \leftarrow \min(|S|;\ S \subseteq \mathcal{D}_G \wedge (\forall H_i) S \nsubseteq \mathcal{D}_{H_i})$

It is implicit that the operations *Deck* and *Extensions* return graphs in the canonical labeling for their automorphism group. This ensures that when graphs are checked for equivalence it may be done directly, rather than determing whether they are isomorphic. The operation of canonically labelling graphs reduces to GI, however graphs are generated many fewer times than they are compared, resulting in a large net gain in efficiency. As McMullen found, the process of canonical labelling is the dominant factor in the running time of this algorithm for 1-vertex-deletion reconstruction [11], and therefore a reduction in the number of canonical labellings that is required has a substantial effect on the overall run-time.

3.2 Improved Algorithm

The basic reconstruction algorithm used in the computations presented in this report is a optimized version if this algorithm. The primary optimization is the speedup of step 3, which in McMullen's algorithm requires canonically labelling $O(|V(G)|)$ graphs for each unique extension graph. The optimization involves greatly reducing the number of these canonical labelling operations by exploiting the properties of reconstruction function pairs, specifically that $H \in Extensions(C) \implies C \in Deck(H)$, as well as a property of 1-vertex-deleted subgraphs, specifically that most are unique within a *Deck* (see Section 6.1). By the combination of these two properties, we see that step 2 has, as a side effect computed the membership of each $\mathcal{D}_{H_i}$, as well as the multiplicity of the majority of those members. However, for any card C which is present in $\mathcal{D}_G$ more than once, C may also be present in $\mathcal{D}_{H_i}$ more than once, if it is present at least once. To compute the multiplicity in such cases, C is counted within $Deck(H_i)$ for each $H_i \in Extensions(C)$. This process is further optimized by only canonically labelling graphs in $Deck(H_i)$ which have the same number of edges as C.

In addition the implementation used was extended to handle any reconstruction function pair that operates on graphs. In principle the same core algorithm could be used for any reconstruction function pair, although the specific implementation used is optimized for graphs.

The following is the algorithm used to compute the reconstruction results in this project, given an appropriate reconstruction function pair (f_D, f_E):

1. compute the multiset $\mathcal{D}_G \leftarrow f_D(G)$
2. for each $C_i \in \mathcal{D}_G$:
   (a) compute the set $\mathcal{H}_i \leftarrow f_E(C_i) - G$, hereafter used as a multiset
   (b) for each $H \in \mathcal{H}_i$ set $m(\mathcal{H}_i;\, H) \leftarrow \min(\ m(f_D(H);\, C_i),\ m(\mathcal{D}_G;\, C_i)\ )$
3. compute the multiset of all extensions $\mathcal{H} \leftarrow \biguplus_{C_i \in \mathcal{D}_G} \mathcal{H}_i$
4. compute $\forall rn(G) \leftarrow 1 + \max(m(\mathcal{H};\, H);\ H \in \mathcal{H})$
5. compute $\exists rn(G) \leftarrow \min(\ |\mathcal{S}|;\ (\mathcal{S} \subseteq \mathcal{D}_G) \wedge (\bigcap_{C_i \in \mathcal{S}} \mathrm{T}_i(\mathcal{S}) = \emptyset)\ )$

   where $\mathrm{T}_i(\mathcal{S}) = \{H \mid H \in \mathcal{H}_i \wedge m(\mathcal{H}_i;\, H) \geq m(\mathcal{S};\, C_i)\}$

3.3 Runtime Complexity

Analyzing the run-time complexity of either algorithm is a complicated task due to the number of variables involved. These variables include mainly:

- the number of unique cards in $f_D(G)$
- the number of unique extensions of each of those cards
- the number of cards in $f_D(G)$ that are non-duplicates ($m(f_D(G);\, c) = 1$)
- the number of vertices and edges in each of those cards and extensions of cards
- the value of the existential reconstruction number

Many of those values are summarized in Section 6 for different reconstruction function pairs.

In this discussion of the run-time complexity of the improved algorithm used in this project, one major assumption is made: the vast majority of run-time in steps 1 and 2 is due to the canonical labelling of the resultant graphs. This behavior was previously described by Brian McMullen [11] in his implementation, and held true as well in the implementation used in this project. Furthermore, while the run-time complexity of each canonical labelling depends on the properties of the graph itself, especially it's order, canonical labelling will be treated as a constant time operation for simplicity.

The number of canonical labelling operations needed by the first section of the algorithm used in this project can be approximated as follows:

1: $\Theta(|f_D(G)|)$

2: $\Theta(\|f_D(G)\| \cdot (\mathrm{a} + \mathrm{b}))$

   a: $|f'_E(C)|$

   b: if $m(f_D(G);\, C) = 1$ then 0, else $\|f_E(C)\| \cdot |f_D(H)|$

Where G is the original graph, C is a representative card, and H is a representative extension of a representative card. f'_E represents the multiset of all extensions that are computed in order to determine f_E, and it is often the case that $|f'_E| \ll |f_E|$, although, by definition, $\|f'_E\| = \|f_E\| = |f_E|$.

The next section of the algorithm computes the universal reconstruction number, which can be described as simply one greater than the greatest number of cards the original graph shares in common with any other. Since each H_i effectively has a count of how many times a given card is shared with each possible other graph, the additive union of them gives the result almost directly. Computation of the additive union itself, while straightforward, is not trivial run-time, and can be most efficiently done as a form of insertion-sort when the set is large. The run-time complexity of this can be described as:

3: $O(|\mathcal{H}| \cdot \log_2 |\mathcal{H}|)$

4: $\Theta(|\mathcal{H}|)$

Where $|\mathcal{H}| \approx |f_E(C)| \cdot |f_D(H)|$ is the total number of unique extensions of all cards of G.

The final part of the algorithm computes the existential reconstruction number. In order to prove the existential reconstruction number is n, it must be shown that there is a *Subdeck* of G of size n which is not shared by any other graph, and also that there is no *Subdeck* of size $n-1$ that meets that same criteria. In general this requires that all *Subdeck*s of size $n-1$ and at least some *Subdeck*s of size n are evaluated. Since there is no pre-existing knowledge of what the existential reconstruction number will be, in practice at least all *Subdeck*s of size $\leq n-1$ will be evaluated while searching for the smallest *Subdeck* that reconstructs G. Each *Subdeck* evaluation, in turn, requires comparing $\|Subdeck\|$ extension sets, therefore the overall run-time complexity can be expressed as:

5: $\Omega\left(\sum_{n=1}^{\exists rn(G)-1} f(n)\right)$ and $O\left(\sum_{n=1}^{\exists rn(G)} f(n)\right)$

where $f(n) = O(n \cdot \binom{|f_D(G)|}{n})$. This gives an upper bound of:

$$O\left(\sum_{n=1}^{\exists rn(G)} n \cdot \binom{|f_D(G)|}{n}\right) = O\left(\exists rn(G) \cdot \sum_{n=1}^{\exists rn(G)} \binom{|f_D(G)|}{n}\right) = O\left(\exists rn(G) \cdot 2^{|f_D(G)|}\right)$$

In the common case where $\exists rn(G) \ll |f_D(G)|$ this can be approximated as:

$$\approx O\left(\exists rn(G) \cdot \binom{|f_D(G)|}{\exists rn(G)}\right)$$

However, in cases — such as 1-vertex and 1-edge deletion — where $\exists rn(G)$ is almost always some small integer, this can be treated as a constant time operation. In such cases the the computational time for step 5 is also very small in comparison to the rest of the algorithm.

This leads to a rough trichotomy in the run-time behavior of this algorithm. In cases where the reconstruction numbers have a very high probability of being small values also tend to be the cases where there are very few duplicate cards in a deck, and step 2a dominates the run-time. In cases where the reconstruction numbers are larger, then it is step 5 that dominates. The last category is cases where the original graph is not reconstructible, in which case step 5 can be skipped altogether ($\forall rn = \infty \iff \exists rn = \infty$), and there tend to be many duplicate cards in the deck, therefore step 2b tends to dominate the run-time.

4 Implementation

The programs used in the computation of the results presented in this report were implemented with three primary goals, in order of importance: efficiency, flexibility, and maintainability. Given these goals, the programs were primarily written in standards compliant C++, although not in a strictly object oriented fashion. Where effective, standard STL-based data structures where used, however in many cases it was found to be much more efficient to implement customized data structures and algorithms. In general the code was optimized for computing 1-vertex-deletion reconstruction numbers, as that was the major goal of this project. This sometimes involved tradeoffs that were less favorable to computing other reconstruction numbers, and in those cases the code often has selections for different algorithms at compile time. Error checking is liberal throughout the code, in order to isolate any logic errors. In cases where the error checking is detrimental to performance it can be disabled at compile time, and that is done once the correctness of the logic has been verified.

As mentioned in Section 3, canonical labelling of graphs is a critical aspect of this project. For this task Brendan McKay's *Nauty* [9] package was used. In some cases the code within the Nauty package needed to be modified to make it compatible with the programs written for this project. Specifically some functions were modified to take *const* parameters, and other functions were renamed to avoid conflict with standard functions. In addition, the Nauty package had various utilities which were useful, especially *geng* for producing graphs with various characteristics, and *listg* for displaying graphs in human readable formats (adjacency lists or matrices).

In the programs written for this project, graphs are represented internally in either Nauty's native representation, or in a more space efficient representation termed *compact*. The Nauty graph format is essentially an adjacency matrix with a minimum of one machine word per vertex. Since all graphs in this project are simple and undirected, the compact format uses $\binom{|V(G)|}{2}$ bits per graph, and is similar in definition to that used by Brian McMullen [11]. In contrast to the definition McMullen used, the compact format used in this project is defined as an array of bytes sufficient in size for the maximum graph size, allowing graphs of arbitrary size instead of being limited to at most 11 vertices (which can fit into McMullen's 64bit quantity). If the maximum graph size is defined such that it fits into 64 or 128 bits (11 and 16 vertices, respectively), then certain optimizations are enabled to speed comparisons between graphs by taking advantage of 64bit comparisons rather than bytewise comparisons. In practice this allows equal efficiency to McMullen's representation while still allowing greater flexibility.

The $Deck_1$ and $Extensions_1$ operations were carefully optimized for performance, as they are critical to the performance of computing the 1-vertex-deleted reconstruction numbers. In both cases it was found that computing successive graphs in-place (rather than copying from the original graph each time) was more efficient. In the case of the $Deck_1$ operation, this was accomplished by deleting a vertex, and then successively swapping the next vertex with one from the original graph. This has the effect of "bubbling" the hole through the adjacency matrix. In the case of $Extensions_1$ a binary reflected Gray code was used. The difference between successive values in the Gray code was pre-computed and stored as an array, such that at iteration n, edge $g[n]$ is to be complemented to generate the next graph. The $Deck_2$ and $Extensions_2$ operations were implemented similarly, and then $Deck_k$ and $Extensions_k$ operations were implemented as generalizations, although less time efficiently due to memory constraints.

The analogous operations for edge-reconstruction were implemented with similar algorithms, although by their nature they were less complex, and had less overall effect on performance.

During the computation of reconstruction numbers it is necessary to store and manipulate large sets of graphs. For this purpose a custom data structure which stores graphs in compact representation was implemented to maximize performance. The data structure operates essentially as a chained hashtable with a compile-time selectable number of buckets, each of which is implemented with an STL vector. The hash key is actually made up of two parts which are mixed together: the number of edges and a hash function consisting of the middle byte of the compact graph representation. The graphs stored in this data structure are always canonically labelled, and it was found the middle bits were the most random due to the canonical labelling placing vertices in order of degree. In practice this hash function was found to be very effective at distributing the results of the $Extensions_1$ operation efficiently, and both values are immediately available since the implementation of the *Extensions* operations keeps track of the number of edges added during graph generation.

4.1 Usage

There were 4 main programs written for this project, as well as a common library of functionality which they share. The header file *config.h* contains a number of compile-time settings which are used by various parts of the implementation. For instance, the size of hash tables, whether error checking is enabled for certain libraries, and which algorithm to use when multiple are available (which may be more efficient in some situations versus others).

The first program is *gconvert* which converts graphs between 3 different graph formats: raw Nauty format, Nauty *graph6* format, and a custom compact format which is optimally space efficient. The compact format is the format that all the other programs read, while the first two formats are ones that the Nauty package can use. The next two programs are *deck* and *extend* which compute *Deck*s and *Extensions* of input graphs, and produce statistics relating to those operations. The final program is *rncompute* which computes reconstruction numbers of input graphs. Each of these programs takes command line parameters to indicate the reconstruction function pairs to use, as well as to control the type of output provided. The command line parameters of each program implemented are as follows:

```
Usage: gconvert <intype> <outtype> [infilename] [outfilename]
where <intype> and <outtype> are one of:
  n[auty]     = raw nauty format
  c[ompact]   = compact format
  g[raph6]    = nauty graph6 format
  N[ull]      = NULL/bit-bucket (ignores filename)
and indicate the format of the input and output files.
If [infilename] and/or [outfilename] are missing or "-"
then stdin and stdout will be used in their place
```

```
Usage: deck [options] [infilename]
where [options] can be:
  -k <k> = compute k-deletion decks
  -e     = compute edge-deletion decks
  -v     = increase verbosity level (can be used multiple times)
  -n     = no count summary is printed
  -h     = print this help text
If [infilename] is missing or "-" then stdin will be used.
Unless -e is specified, vertex-deletion is assumed, and the default
value for -k is 1. Multiple arguments can be combined with a single
dash, if so desired. For instance: -ek3 for 3-edge-deletion.
At verbosity 1 each card is output in graph6 format.
At verbosity 2 each input graph is output before its cards.
```

```
Usage: extend [options] [infilename]
where [options] can be:
  -k <k> = compute k-deletion extension sets
  -e     = compute edge-deletion extension sets
  -v     = increase verbosity level (can be used multiple times)
  -n     = no count summary is printed
  -h     = print this help text
If [infilename] is missing or "-" then stdin will be used.
Unless -e is specified, vertex-deletion is assumed, and the default
value for -k is 1. Multiple arguments can be combined with a single
dash, if so desired. For instance: -ek3 for 3-edge-deletion.
At verbosity 1 each extension is output in graph6 format.
At verbosity 2 each input graph is output before its extensions.
```

```
Usage: rncompute [options] [infilename]
where [options] can be:
  -E <n> = print graph if Existential RN >= n
  -A <n> = print graph if Universal RN >= n
  -k <k> = compute k-reconstruction numbers
  -e     = compute edge-reconstruction numbers
  -v     = increase verbosity level (can be used multiple times)
  -n     = no count summary is printed
  -q     = quiet, none of the standard output is printed
  -h     = print this help text
If [infilename] is missing or "-" then stdin will be used.
Unless -e is specified, vertex-deletion reconstruction numbers
will be computed. The default value for -E and -A is infinity,
and for -k is 1. Multiple arguments can be combined with a single
dash, if so desired. For instance: -ek3 for 3-edge-deletion.
```

5 Performance

Due to the nature of this project, a great deal of emphasis was placed on run-time optimization. The results of the algorithm, implementation, and compiler-assisted optimizations was impressive when compared to previous efforts.

All computations for this project were done on one of two systems. The first system is single machine with dual AMD Opteron 248 processors and 1GB of RAM, running a standard installation of Fedora Core 6 Linux and using GCC version 4.1.1 as the compiler. The second system is the compute cluster at the Rochester Institute of Technology run by RIT's Center for Advancing the Study of Cyberinfrastructure (CASCI) [16]. The cluster consists of 94 1.4GHz Pentium IV processors arranged in 47 dual processor compute nodes, each with 512MiB of main memory, and runs the ROCKS HPC software [16, 17] with GCC version 3.4.5 as the available compiler. It was found that each of the CPUs in the CASCI cluster performed at approximately 50% the speed of the Opteron 248 processors when computing 1-vertex-deletion reconstruction numbers, giving the cluster overall performance of 23.5 times that of the first system. All compute time results presented are normalized to a single Opteron 248 CPU, as the first system performed the majority of the computations, and it will be noted which were computed on the cluster.

Table 1 compares the runtime between the implementation used by Brian McMullen in [11] and that used in this project. In both cases only graphs on $\leq \binom{|V(G)|}{2}$ edges were directly computed, as $vrn(G) = vrn(\overline{G})$ [6].

$\lvert V(G)\rvert$	unique graphs	computed graphs	McMullen's implementation	new implementation	speedup
6	156	78	0.07 seconds	0.02 seconds	3.5
7	1044	522	1.84 seconds	0.52 seconds	3.5
8	12346	6996	1.50 minutes	16.8 seconds	5.4
9	274668	154354	1.83 hours	14.0 minutes	7.9
10	12005168	6002584	9.19 days	20.9 hours	10.5
11	1018997864	509498932	6.12 years‡	174 days†	12.8

†Computations performed on the CASCI cluster.

‡Estimated by computing 0.02% of graphs

Table 1: Comparison of run-time for computing 1-vertex-deletion results against McMullen's implementation

Presented in tables 2 and 3 are compute times for certain results which are presented in Section 6. These compute times are presented as the total time to compute k-vertex,edge-deletion reconstruction numbers for all graphs of a given order. Since the number of graphs increases exponentially with order, tables 4 and 5 give the same results in terms of the average time to compute each graph in milliseconds.

$\|V(G)\|$	k=1	k=2	k=3	k=4
6	0.02 seconds	0.45 seconds	0.88 seconds	0.72 seconds
7	0.52 seconds	23.3 seconds	1.7 minutes	1.2 minutes
8	16.8 seconds	42.5 minutes	19.2 hours	9.5 days
9	14.0 minutes	4.6 days	379 days†	
10	20.9 hours			
11	174 days†			

†Computations performed on the CASCI cluster.

Table 2: Run-times for computing k-vertex-deletion reconstruction numbers for all graphs on 6–11 vertices

$\|V(G)\|$	k=1	k=2	k=3	k=4
6	0.04 seconds	0.56 seconds	3.2 seconds	68.9 seconds
7	0.74 seconds	40.7 seconds	11.9 minutes	6.2 hours
8	16.3 seconds	58.2 minutes	3.1 days	
9	10 minutes	3.9 days		
10	10.7 hours			
11	23.7 days†			

†Computations performed on the CASCI cluster.

Table 3: Run-times for computing k-edge-deletion reconstruction numbers for all graphs on 6–10 vertices

$\|V(G)\|$	k=1	k=2	k=3	k=4
6	0.26	5.89	12.7	9.74
7	1.07	44.7	193	135
8	2.45	364	9892	116955
9	5.52	2549	212137†	
10	12.6			
11	29.5†			

†Computations performed on the CASCI cluster.
All values in units of milliseconds.

Table 4: Average per-graph compute-time for k-vertex-deletion reconstruction numbers for graphs on 6–11 vertices

$\|V(G)\|$	k=1	k=2	k=3	k=4
6	0.25	3.64	21.2	469
7	0.71	39.1	690	10869
8	1.32	283	22039	
9	2.18	1235		
10	3.21			
11	4.02†			

†Computations performed on the CASCI cluster.
All values in units of milliseconds.

Table 5: Average per-graph compute-time for k-edge-deletion reconstruction numbers for graphs on 6–10 vertices

Compute time for edge-deletion reconstruction numbers depends strongly on the number of edges in the original graph, as $|f_D(G)| = |\mathcal{E}Deck_k(G)| = \mathrm{C}(|E(G)|, k)$ and $|f'_E(C)| = |\mathcal{E}Extensions'_k(C)| = \mathrm{C}(|E(\overline{C})|, k)$. To illustrate this, table 6 shows the average time to compute k-edge-deletion reconstruction numbers for all graphs on 7 vertices according to the number of edges. As described in Section 3.3, the compute-time can also be heavily dependent on the actual value of the existential reconstruction number, which explains the extremely large value for $ern_4(\mathcal{G}_{7,16})$, where the single graph, $\overline{P_5 \cup K_2}$, with $\exists ern = 7$ (out of a possible $\binom{16}{4} = 1820$) dominates the run-time. Table 7 breaks down the computation time for $ern_4(\mathcal{G}_{7,16})$ according to $\exists ern_4$.

$\lvert E(G)\rvert$	k=1	k=2	k=3	k=4
5	0.33	2.52	3.81	2.4
6	0.39	5.04	11.7	9.3
7	0.51	10.6	41.2	43
8	0.55	17.8	115	701
9	0.58	26.7	261	2804
10	0.65	38.0	534	2064
11	0.70	48.6	871	6519
12	0.74	56.9	1175	23326
13	0.82	65.3	1477	11305
14	0.86	64.5	1447	33135
15	0.80	52.4	1155	12800
16	0.86	42.5	810	108815
17	0.80	25.6	450	4552
18	0.60	12.6	220	1708
19	0.50	6.0	75	545

All values in units of milliseconds.

Table 6: Average per-graph compute-time for k-edge-deletion reconstruction numbers for graphs on 7 vertices and 5-19 edges

$\exists ern_4$	number of graphs	total run time (seconds)	avg. per graph (seconds)
2	6	35.9	6.0
3	11	62.3	5.7
4	2	8.6	4.3
5	1	11.3	11.3
7	1	2158.7	2158.7

Table 7: Average per-graph compute-time for 4-edge-deletion reconstruction numbers for graphs in $\mathcal{G}_{7,16}$ according to $\exists ern_4$

6 Results

The results presented in this section are organized primarily by the various reconstruction function pairs that were considered in this project. Within each subsection results are presented as 3 tables, the first containing statistics from f_D, the second containing statistics from f_E, and the last containing counts of existential and universal reconstruction numbers.

For each table the data is broken out into columns according to the number of vertices, and tabulates results for all graphs of that order. The first row of each table lists the number of unique (non-isomorphic) graphs of each order to which the function in question is applicable. Examples of graphs to which certain functions are not applicable, are graphs with fewer than k edges for the $\mathcal{E}Deck_k$ family of functions. The set of useful graphs of a given order under a given function will be referred to use $\mathcal{G}_u$ for convenience.

For the tables summarizing the f_D function, the following additional rows are given:

total cards: $T_C = \sum_{g \in \mathcal{G}_u} \|f_D(g)\|$

non-duplicates: $T_{nd} = \sum_{g \in \mathcal{G}_u} \|\{c \mid \ m(f_D(g);\, c) = 1\}\|$

average cards per deck: $\dfrac{T_C}{|\mathcal{G}_u|}$

percentage of non-duplicate cards: $\dfrac{T_{nd}}{|\mathcal{G}_u|}$

For the tables summarizing the f_E function, the following additional rows are given:

total extensions: $T_E = \sum_{g \in \mathcal{G}_u} \|f_E(g)\|$

average unique extensions per graph: $\dfrac{T_E}{|\mathcal{G}_u|}$

Tables summarizing reconstruction numbers have two primary sets of rows. In the first set each row counts the number of graph with a given existential reconstruction number, and in the second set each row counts the number of graphs with a given universal reconstruction number. If the table is too long to display on a single page, then it is be broken into separate tables for existential and universal reconstruction numbers. If a table is still too long to display on a single page, then it is further subdivided by the range of reconstruction numbers. If there is a sequence of 2 or more rows which are empty, then they are collapsed into a single line.

In addition, these tables may have additional optional rows. The first optional row appears at the top of the table, and shows the number of graphs which are not reconstructible. The other type of optional row appears at the bottom of the table, and shows the percentage of graphs which have some interesting or common existential or universal reconstruction number.

6.1 Results for 1-Vertex Deletion

Deck$_1$ **Counts**

	graph order								
	3	4	5	6	7	8	9	10	11
unique graphs	4	11	34	156	1044	12346	274668	12005168	1018997864
total cards	6	20	90	544	5096	79220	2208158	113737744	10926144928
non-duplicates	2	6	39	278	3370	62508	1965149	107681508	10649004558
avg. cards/deck	1.5	1.8	2.6	3.5	4.9	6.42	8.039	9.474	10.722
% non-duplicate	33	30	43	51	66.1	78.90	88.995	94.675	97.464

Extensions$_1$ **Counts**

	graph order								
	2	3	4	5	6	7	8	9	10
unique graphs	2	4	11	34	156	1044	12346	274668	12005168
total extensions	6	20	90	544	5096	79220	2208158	113737744	10926144928
avg. extensions	3	5	8	16	32.7	75.88	178.856	414.092	910.120

1-Vertex Deletion Reconstruction Numbers

		graph order								
		3	4	5	6	7	8	9	10	11
unique graphs		4	11	34	156	1044	12346	274654	12005168	1018997864
$\exists vrn_1$	3	4	8	34	150	1044	12334	274652	12005156	1018997864
	4		3		4		8		6	
	5				2		2	2	4	
	6						2			
	7								2	
$\forall vrn_1$	3	4	2	7	8	16	266	45186	6054148	815604300
	4		9	19	56	496	8208	199247	5637886	199382868
	5			8	90	520	3584	28777	301530	3922130
	6				2	12	284	1426	10686	83730
	7						4	18	914	4824
	8								4	12
% $\exists vrn_1 = 3$		100	73	100	96.2	100.0	99.903	99.9993	99.9999	100.0000
% $\forall vrn_1 = 3$		100	18	21	5.1	1.53	2.155	16.4520	50.4295	80.0398

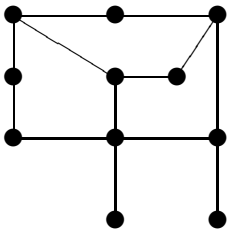
Discussion

There are a number of items of interest in these results. Foremost is that they agree with previous theoretical and empirical results. In 1990 Bollobás proved that almost every graph can be reconstructed from any 3 cards in its deck [4], and the counts presented here bear that out in striking fashion. In addition the reconstruction number counts agree with those collected by Brian McMullen for $|V(G)| \leq 10$ in 2005 [11, 12].

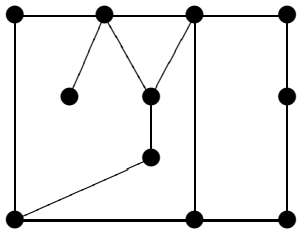
Graphs which have high (> 3) $\exists vrn_1$ are of special interest. Myrvold showed that for any disconnected graph of the form pK_c that $\exists vrn_1 = c + 2$ [14] (proof corrected by Molina [13]). McMullen later described some additional infinite families of graphs which have high $\exists vrn_1$ [12]. In all cases the construction of graphs with high $\exists vrn_1$ requires the graphs to have a non-prime order, and showed that all graphs where $5 \leq |V(G)| \leq 10$ and $\exists vrn_1 > 3$) are members of one or more of the known families. The results presented here prove that no graph on 11 vertices has $\exists vrn_1 > 3$, strengthening the previous conjecture that all graphs of prime order have $\exists vrn_1 = 3$ [11].

The $Deck_1$ and $Extensions_1$ tables reveal some interesting details of their own. The average number of unique cards per deck and the average number of unique extensions of a graph rapidly approach their maximum values ($|V(G)|$ and $2^{|V(G)|}$, respectively). This has strong implications for future work in computing reconstruction numbers, as the number of graphs which will need to be canonically labelled for any graph G will rapidly approach $|V(G)|^2 \cdot 2^{|V(G)|-1}$ as $|V(G)|$ increases further. More favorably, the probability that any given card in a deck is isomorphically unique within that deck also increases rapidly towards 1, indicating that the optimizations listed in Section 3 are very effective for computing $\exists vrn_1$ and $\forall vrn_1$.

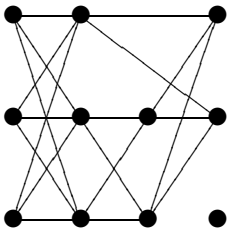
Recent work has focused on graphs which have high (maximal or nearly maximal) $\forall vrn_1$ [5]. The results presented here prove that there are 12 graphs of order 11 with maximal $\forall vrn_1 = 8$. The following graphs and their complements are all graphs of order 11 with $\forall vrn_1$=8. Graphs are presented in pairs, each of which has exactly 7 cards in common. Below each graph is listed its number of edges, and its Nauty graph6 representation.



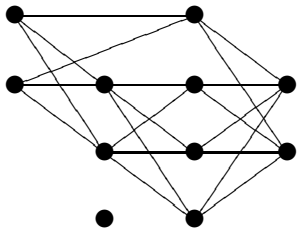
$|E(G)| = 13$, `J@?A?OTBdE?`



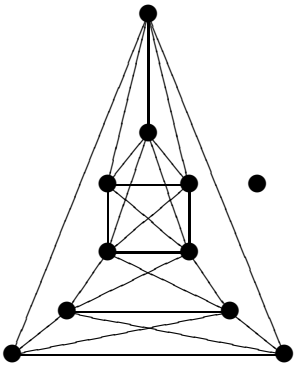
$|E(G)| = 14$, `J??_r?HDMK?`



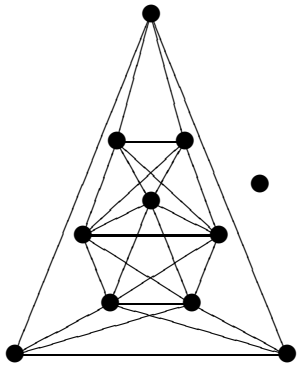
$|E(G)| = 19$, `J???WX{[re?`



$|E(G)| = 20$, `J??WopwXrM?`



$|E(G)| = 25$, `JGCZz@‘FWz_`



$|E(G)| = 26$, `JJCWW[NWzF_`

It should be noted that the 4 such graphs with the highest edge counts each have a single disconnected vertex. If that vertex is removed, the resulting graph on 10 vertices also have $\forall vrn_1 = 8$, and are the only such graphs on 10 vertices.

6.2 Results for 2-Vertex Deletion

$Deck_2$ Counts

	graph order						
	4	5	6	7	8	9	10
unique graphs	11	34	156	1044	12346	274668	12005168
total cards	20	94	766	9020	193144	7143022	460986428
non-duplicates	2	18	208	3156	101534	5091782	392761878
avg. cards/deck	1.8	2.8	4.9	8.6	15.644	26.006	38.399
% non-duplicate	10	19	27	35.0	52.569	71.283	85.200

$Extensions_2$ Counts

	graph order							
	2	3	4	5	6	7	8	9
unique graphs	2	4	11	34	156	1044	12346	274668
total extensions	20	94	766	9020	193144	7143022	460986428	52001662796
avg. extensions	10	24	70	265	1238.1	6841.98	37338.930	189325.523

2-Vertex Deletion Reconstruction Numbers

		graph order					
		4	5	6	7	8	9
unique graphs		11	34	156	1044	12346	274654
not reconstructible		7	4	0	0	0	0
$\exists vrn_2$	3			8	240	9592	270869
	4		2	30	396	2464	3448
	5			34	216	216	228
	6	4	4	30	106	36	48
	7		8	32	44	18	18
	8		9	16	20	8	16
	9		7	2	10	2	3
	10			2	4	4	12
	11			2	2	4	4
	12				6		2
	13						
	14					2	2
	15						
	16						4
$\forall vrn_2$	6	4					
	7						
	8		6				
	9		9				
	10		15	6			
	11			2			
	12			4	4		
	13			98	2		
	14			46	14	5	
	15				76	4	
	16				216	36	9
	17				532	111	271
	18				172	1020	3704
	19				28	2820	14270
	20					3598	21982
	21					3212	60137
	22					1254	79796
	23					248	48632
	24					32	20508
	25					6	17345
	26						5768
	27						1820
	28						316
	29						92
	30						4
% $\exists vrn_2 = 3$		0	0	5.1	22.99	77.693	98.6219

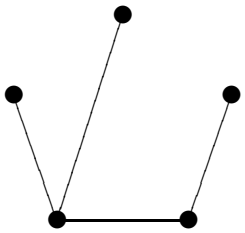
Discussion

These results begin to show a dichotomy between the patterns of existential and universal k-vertex-deleted reconstruction numbers, which grows greater as k increases. While the distribution of $\exists vrn_2$ is very similar to $\exists vrn_1$, the distribution of $\forall vrn_2$ is greatly different from that of $\forall vrn_1$. Indeed, just a casual inspection leads one to conjecture that almost every $\exists vrn_2(G) = 3$, just as almost every $\exists vrn_1(G) = 3$ [4]. In contrast, while it has also been shown that almost every $\forall vrn_1(G) = 3$ [4], both extents of the range of $\forall vrn_2(G)$ seems to increase without bound as $|V(G)|$ increases.

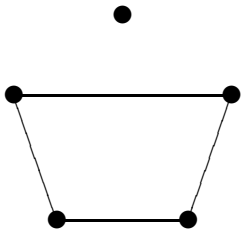
Another pattern that becomes more obvious in k-vertex-deleted results as k in-

creases, is that if $k = |V(G)| - 2$, there are always exactly 4 graphs that are reconstructible (see appendix A.1). These 4 graphs are those with 0 or 1 edges, and their complements. It is clear that $Deck_{|V(G)|-2}(G)$ holds no information other than $|V(G)|$ and $|E(G)|$, and it is also clear to see that for any fixed $|V(G)| \geq 3$, the only graphs which have a unique $|E(G)|$ are those on 0, 1, $\binom{|V(G)|}{2}$, and $\binom{|V(G)|}{2} - 1$ edges.

More interesting in these results is those 4 graphs on 5 vertices which are not 2-vertex-deletion reconstructible. The following two graphs, and their complements are not 2-vertex-deletion reconstructible. It is trivial to see that these two graphs share the same $Deck_2$, which proves that they are not 2-vertex-deletion reconstructible.



$|E(G)| = 4$, `D@s`



$|E(G)| = 4$, `DIK`

6.3 Results for 3-Vertex Deletion

$Deck_3$ Counts

	graph order					
	5	6	7	8	9	10
unique graphs	34	156	1044	12346	274668	12005168
total cards	66	526	7552	190579	9905170	914086960
non-duplicates	2	52	1318	44122	4001616	575295558
avg. cards/deck	1.9	3.4	7.2	15.436	36.062	76.141
% non-duplicate	3.0	10	17.5	23.152	40.399	62.937

$Extensions_3$ Counts

	graph order					
	2	3	4	5	6	7
unique graphs	2	4	11	34	156	1044
total extensions	66	526	7552	190579	9905170	914086960
avg. extensions	33	132	687	5605.3	63494.7	875562.22

3-Vertex Deletion Reconstruction Numbers

		graph order				
		5	6	7	8	9
unique graphs		34	156	1044	12346	274668
not reconstructible		30	78	20	8	0
$\exists vrn_3$	3					2760
	4				128	45713
	5			10	652	145271
	6			12	1738	62156
	7			24	2290	14434
	8		2	66	2285	3018
	9		2	90	1874	678
	10	4	4	126	1216	244
	11			88	755	160
	12		2	96	490	68
	13		8	70	304	46
	14		2	76	207	34
	15		10	54	152	26
	16		8	66	72	20
	17		14	74	40	8
	18		22	54	38	2
	19		4	62	36	8
	20			30	5	2
	21			14	16	2
	22			6	6	4
	23				6	2
	24			4	6	4
	25				2	
	26			2	4	
	27					
	28				4	
	29				4	
	30				2	
	31					2
	32					
	33				2	
	34–35					
	36					4
	37				2	
	38–40					
	41				2	
	42–49					
	50					2
% $\exists vrn_3 = 3$		0	0.0	0.00	0.000	1.0048

		graph order				
		5	6	7	8	9
unique graphs		34	156	1044	12346	274668
not reconstructible		30	78	20	8	0
$\forall vrn_3$	10	4				
	11–16					
	17		4			
	18		6			
	19		68			
	20–25					
	26			6		
	27					
	28			4		
	29			4		
	30			38		
	31			88		
	32			400		
	33			342		
	34			142		
	35–36					
	37				8	
	38					
	39				4	
	40				3	
	41				16	
	42				6	
	43				51	
	44				76	
	45				263	
	46				532	
	47				1282	
	48				2451	
	49				3902	
	50				2602	5
	51				840	
	52				118	2
	53				96	8
	54				88	4
	55					75
	56					98
	57					157
	58					242
	59					360
	60					1940
	61					3798
	62					6426
	63					11409
	64					21181
	65					32518
	66					42127
	67					46011
	68					38908
	69					30087
	70					18289

		graph order				
		5	6	7	8	9
unique graphs		34	156	1044	12346	274668
not reconstructible		30	78	20	8	0
$\forall vrn_3$	71					10642
	72					5843
	73					2984
	74					1216
	75					224
	76					64
	77					46
	78					
	79					4

6.4 Results for 4-Vertex Deletion

$Deck_4$ Counts

	graph order				
	6	7	8	9	10
unique graphs	156	1044	12346	274668	12005168
total cards	310	3932	110632	6008449	758176916
non-duplicates	2	214	12052	937719	221472810
avg. cards/deck	2.0	3.8	8.961	21.875	63.154
% non-duplicate	0.6	5.4	10.894	15.607	29.211

$Extensions_4$ Counts

	graph order				
	2	3	4	5	6
unique graphs	2	4	11	34	156
total extensions	310	3932	110632	6008449	758176916
avg. extensions	155	983	10057	176719	4860108.4

4-Vertex Deletion Reconstruction Numbers

		graph order		
		6	7	8
unique graphs		156	1044	12346
not reconstructible		152	854	1937
$\exists vrn_4$	7			6
	8			6
	9			10
	10			21
	11			8
	12			16
	13			48
	14			66
	15	4	4	100
	16		6	170
	17		2	193
	18		2	212
	19		2	346
	20		2	440
	21		4	368
	22			310
	23		2	318
	24			365
	25		14	375
	26		2	436
	27		6	322
	28		22	420
	29		8	460
	30		16	488
	31		30	452
	32		22	434
	33		44	442
	34		2	442
	35			450
	36			354
	37			370
	38			351
	39			403
	40			300
	41			304
	42			212
	43			169
	44			70
	45			58
	46			36
	47			22
	48			20
	49			6
	50			4
	51			2
	52			2
	53–55			
	56			2
% $\exists vrn_4 = 3$		0.0	0.00	0.000

		graph order		
		6	7	8
unique graphs		156	1044	12346
not reconstructible		152	854	1937
$\forall vrn_4$	15	4		
	16–30			
	31		8	
	32		6	
	33		16	
	34		160	
	35–55			
	56			8
	57			2
	58–59			
	60			4
	61			14
	62			22
	63			98
	64			214
	65			548
	66			1065
	67			3062
	68			3362
	69			2010

6.5 Results for 5-Vertex Deletion

$Deck_5$ Counts

	graph order			
	7	8	9	10
unique graphs	1044	12346	274668	12005168
total cards	2086	48520	2735466	318081750
non-duplicates	2	1066	176354	32690616
avg. cards/deck	2.0	3.93	9.959	26.495
% non-duplicate	0.1	2.20	6.447	10.277

$Extensions_5$ Counts

	graph order			
	2	3	4	5
unique graphs	2	4	11	34
total extensions	2086	48520	2735466	318081750
avg. extensions	1043	12130	248679	9355345.6

5-Vertex Deletion Reconstruction Numbers

		graph order	
		7	8
unique graphs		1044	12346
not reconstructible		1040	11935
$\exists vrn_5$	21	4	
	22		
	23		2
	24		4
	25		4
	26		4
	27		
	28		2
	29		6
	30		4
	31		2
	32		2
	33		
	34		4
	35		2
	36		2
	37		2
	38		4
	39		4
	40		6
	41		2
	42		8
	43		8
	44		8
	45		8
	46		12
	47		32
	48		10
	49		28
	50		30
	51		24
	52		56
	53		66
	54		65
$\forall vrn_5$	21	4	
	22–50		
	51		6
	52		6
	53		18
	54		24
	55		357
% $\exists vrn_5 = 3$		0.00	0.000

6.6 Results for 6-Vertex Deletion

$Deck_6$ Counts

	graph order		
	8	9	10
unique graphs	12346	274668	12005168
total cards	24690	1094818	125536992
non-duplicates	2	7270	4935086
avg. cards/deck	2.00	3.986	10.457
% non-duplicate	0.01	0.664	3.931

$Extensions_6$ Counts

	graph order		
	2	3	4
unique graphs	2	4	11
total extensions	24690	1094818	125536992
avg. extensions	12345	273704	11412454

6-Vertex Deletion Reconstruction Numbers

		graph order
		8
unique graphs		12346
not reconstructible		12342
$\exists vrn_6$	28	4
$\forall vrn_6$	28	4
% $\exists vrn_6 = 3$		0.000

6.7 Results for 1-Edge Deletion

$\mathcal{E}Deck_1$ Counts

	graph order								
	3	4	5	6	7	8	9	10	11
unique graphs	3	10	33	155	1043	12345	274667	12005167	1018997863
total cards	3	14	74	571	6558	125066	4147388	247179594	26814371140
non-duplicates	1	5	27	266	3836	90919	3497830	226766172	25678801469
avg. cards/deck	1	1.4	2.2	3.7	6.3	10.131	15.100	20.589	26.314
% non-duplicate	33	36	36	47	58.5	72.697	84.338	91.741	95.765

$\mathcal{E}Extensions_1$ Counts

	graph order									
	2	3	4	5	6	7	8	9	10	11
unique graphs	1	3	10	33	155	1043	12345	274667	12005167	1018997863
total extensions	1	3	14	74	571	6558	125066	4147388	247179594	26814371140
avg. extensions	1	1	1.4	2.2	4	6.3	10.131	15.100	20.589	26.314

1-Edge Deletion Reconstruction Numbers

		graph order								
		3	4	5	6	7	8	9	10	11
unique graphs		3	10	33	155	1043	12345	274667	12005167	1018997863
not reconstructible		0	4	4	4	4	4	4	4	4
$\exists ern_8$	1	3	5	9	18	23	35	46	64	71
	2			14	115	980	12242	274523	12004951	1018997596
	3		1	6	16	31	57	81	130	167
	4				2	5	4	9	10	15
	5						3	3	5	6
	6							1	2	2
	7								1	1
	8									1
$\forall ern_8$	1	3	3	3	3	3	3	3	3	3
	2			2	14	19	51	152	1591	2479879
	3		3	8	28	131	1622	65814	5895154	748858136
	4			6	36	285	5059	141767	4976002	239960040
	5			8	46	394	3880	50196	925253	24213068
	6			2	15	128	952	10379	138350	2533007
	7				5	41	520	4171	47953	711284
	8				4	20	136	1228	11382	141498
	9					12	55	521	5704	67083
	10					4	26	202	1854	18352
	11					2	21	110	1070	9050
	12						8	57	359	2615
	13						4	37	292	2562
	14						2	10	68	512
	15						2	10	66	376
	16							2	23	188
	17								19	106
	18							2	8	30
	19								2	26
	20							2	2	10
	21								2	6
	22								2	8
	23								2	4
	24									2
	25									6
	26								2	2
	27									
	28									4
	29–32									
	33									2
% $\exists ern_8 = 2$		0	0	42	74.2	93.96	99.166	99.9476	99.9982	100.0000
% $\forall ern_8 = 2$		0	0	6	9.0	1.82	0.413	0.0553	0.0133	0.2434

Discussion

Compared to vertex-deletion reconstruction, edge-deletion reconstruction has received comparatively little attention. Lauri has, however, recently shown that $\exists ern_1$ and $\forall ern_1$ are almost always 2 [7] (see also [1]). While the results presented here agree with that theoretical result for $\exists ern_1$, the values of $\forall ern_1$ appear not to agree. Indeed, it appears, from these results, that $\forall ern_1$ may tend more strongly torwards 3 as graph order increases. It should be noted that Lauri's proof relies on graphs with more than 13 vertices, so significantly larger graphs than presented here would need to be explored in order to determine whether there is a real discrepency.

These results also illustrate that for any order $n \geq 4$ there are 4 unique graphs that are not 1-edge-deletion reconstructible. These graphs are listed here, with those sharing equivalent $\mathcal{E}Deck_1$s grouped together, also serving as a proof of their non-reconstructibility:

$$\left.\begin{array}{r} P_3 \cup (n-3)K_1 \\ 2K_2 \cup (n-4)K_1 \end{array}\right\} \text{cards: } \mathbf{2} \qquad K_2 \cup (n-2)K_1$$

$$\left.\begin{array}{r} K_3 \cup (n-3)K_1 \\ S_4 \cup (n-4)K_1 \end{array}\right\} \text{cards: } \mathbf{3} \qquad P_3 \cup (n-3)K_1$$

In addition there are 3 graphs for any order $n \geq 3$ which have $\exists ern_1 = \forall ern_1 = 1$:

- $K_2 \cup (n-2)K_1$
- $\overline{K_2 \cup (n-2)K_1}$
- K_n

6.8 Results for 2-Edge Deletion

$\mathcal{E}Deck_2$ Counts

	graph order							
	3	4	5	6	7	8	9	10
unique graphs	2	9	32	154	1042	12344	274666	12005166
total cards	2	14	99	1162	21126	637576	31111903	2511542576
non-duplicates	1	3	20	275	7088	340286	22496705	2133059771
avg. cards/deck	1	1.6	3.1	7.5	20.27	51.651	113.272	209.205
% non-duplicate	50	21	20	23.7	33.55	53.372	72.309	84.930

$\mathcal{E}Extensions_2$ Counts

	graph order							
	3	4	5	6	7	8	9	10
unique graphs	2	9	32	154	1042	12344	274666	12005166
total extensions	2	14	99	1162	21126	637576	31111903	2511542576
avg. extensions	1	1.6	3	7.5	20.27	51.651	113.272	209.205

2-Edge Deletion Reconstruction Numbers

		graph order						
		3	4	5	6	7	8	9
unique graphs		2	9	32	154	1042	12344	274666
not reconstructible		0	5	10	14	14	14	14
$\exists ern_8$	1	2	2	3	4	5	7	8
	2		1	6	53	710	11567	273165
	3			3	49	227	605	1259
	4			4	9	40	69	103
	5		1	2	11	21	31	48
	6			3	8	9	24	27
	7			1	5	9	15	19
	8					5	7	13
	9				1	2	3	3
	10							2
	11						2	3
	12							
	13							2
% $\exists ern_8 = 2$		0	11	19	34.4	68.14	93.705	99.4535

		graph order						
		3	4	5	6	7	8	9
unique graphs		2	9	32	154	1042	12344	274666
not reconstructible		0	5	10	14	14	14	14
$\forall ern_8$	1	2	2	2	2	2	2	2
	2–4							
	5				1	1	1	1
	6		2	2	3	4	5	5
	7				2		1	1
	8			1	3	2	2	3
	9			6	9	9	8	10
	10				9	17	20	21
	11				1	4	6	3
	12			1	6	12	13	11
	13			2	11	28	44	51
	14			2	3	11	16	20
	15				1	6	22	26
	16			1	4	13	36	61
	17			3	17	43	75	95
	18				1	9	29	70
	19				5	26	64	88
	20				3	32	91	156
	21				10	41	119	242
	22			2	7	40	150	343
	23				2	16	107	322
	24				1	44	191	668
	25				3	43	232	812
	26				5	25	255	1213
	27				7	40	222	1420
	28					35	298	2278
	29				1	28	281	2599
	30				2	35	392	3776
	31				5	47	357	4075
	32				2	20	345	5024
	33				2	20	391	5959
	34				3	23	413	6828
	35					30	442	6368
	36				1	29	437	8351
	37					22	372	7633
	38				1	15	385	7426
	39					22	343	8318
	40				2	18	340	8053
	41					17	293	6557
	42					18	337	8879
	43					13	331	7315
	44					15	243	5976
	45					7	309	8809
	46					6	308	7499
	47					9	212	5420
	48				1	11	216	8175
	49				2	7	225	7677
	50					14	217	5949
	51					6	226	7612
	52					8	184	6801

		graph order						
		3	4	5	6	7	8	9
unique graphs		2	9	32	154	1042	12344	274666
not reconstructible		0	5	10	14	14	14	14
$\forall ern_8$	53					7	165	6250
	54					4	121	6750
	55					6	188	6061
	56				2	10	143	5010
	57					2	112	5964
	58					8	123	5347
	59					2	102	4394
	60						109	4325
	61					4	98	4400
	62						87	3841
	63						83	3246
	64					3	79	2854
	65					6	96	3352
	66						60	2606
	67					2	108	2508
	68					1	44	2101
	69					1	68	2503
	70					3	59	1951
	71						54	1727
	72						50	1551
	73					1	43	2029
	74					1	43	1416
	75					2	47	1534
	76					2	50	1225
	77					3	39	1631
	78					1	24	1036
	79						16	1171
	80					2	21	998
	81						39	1157
	82					5	27	861
	83						28	990
	84					1	12	740
	85						17	987
	86						23	709
	87						18	744
	88					1	26	611
	89						16	629
	90						8	535
	91					2	20	643
	92						22	533
	93						14	481
	94						10	372
	95						8	489
	96						10	311
	97						9	386
	98						13	325
	99						4	349
	100						13	347
	101						6	284
	102						6	215

		graph order						
		3	4	5	6	7	8	9
unique graphs		2	9	32	154	1042	12344	274666
not reconstructible		0	5	10	14	14	14	14
$\forall ern_8$	103					1	6	273
	104						2	213
	105					2	8	272
	106						9	209
	107						5	212
	108						7	136
	109						4	154
	110						4	140
	111						6	174
	112						2	150
	113						4	112
	114						7	172
	115						3	145
	116						4	105
	117							116
	118						3	123
	119						4	88
	120						3	81
	121						2	128
	122						3	110
	123							99
	124					2	4	79
	125						1	69
	126						2	53
	127						2	83
	128						2	66
	129							49
	130						2	76
	131						1	56
	132						4	63
	133						2	58
	134						1	61
	135							57
	136						2	40
	137							40
	138						2	46
	139							44
	140							22
	141						1	58
	142						2	33
	143							40
	144						1	33
	145							30
	146							28
	147							23
	148							26
	149						1	23
	150						4	28
	151						2	30
	152							25

		graph order						
		3	4	5	6	7	8	9
unique graphs		2	9	32	154	1042	12344	274666
not reconstructible		0	5	10	14	14	14	14
$\forall ern_8$	153						2	27
	154						1	9
	155							26
	156						2	24
	157						1	26
	158							21
	159							26
	160							4
	161							17
	162						2	15
	163							7
	164							10
	165							14
	166						1	16
	167							12
	168							12
	169							33
	170							6
	171							5
	172						1	8
	173							9
	174							10
	175							9
	176							12
	177							8
	178							7
	179						1	7
	180							9
	181							10
	182							9
	183							8
	184						2	12
	185							9
	186							3
	187						1	3
	188						1	4
	189							7
	190							2
	191							4
	192							7
	193							6
	194							2
	195							1
	196							8
	197							5
	198							1
	199							5
	200							2
	201							
	202							3

		graph order						
		3	4	5	6	7	8	9
unique graphs		2	9	32	154	1042	12344	274666
not reconstructible		0	5	10	14	14	14	14
$\forall ern_8$	203							4
	204							4
	205							5
	206							3
	207							
	208						2	10
	209							4
	210							
	211							1
	212							1
	213							3
	214							
	215							1
	216							2
	217							6
	218							
	219							2
	220							7
	221							
	222							1
	223							2
	224						2	4
	225							
	226							4
	227							
	228							5
	229							2
	230–231							
	232							1
	233							3
	234							2
	235							
	236							4
	237–239							
	240							2
	241							2
	242–245							
	246							2
	247–250							
	251							1
	252–255							
	256							2
	257							1
	258							
	259							2
	260							1
	261							1
	262							2
	263							1
	264							

		graph order						
		3	4	5	6	7	8	9
unique graphs		2	9	32	154	1042	12344	274666
not reconstructible		0	5	10	14	14	14	14
$\forall ern_8$	265							3
	266							
	267							2
	268–270							
	271							1
	272							
	273							2
	274–281							
	282							2
	283–284							
	285							2
	286–288							
	289							1
	290–291							
	292							1
	293–308							
	309							1
	310–316							
	317							1
	318–321							
	322							2
	323–337							
	338							1
	339–354							
	355							2
	356–381							
	382							2

Discussion

The following are the graphs of order n which are not 2-edge-deletion-reconstructible, along with their $\mathcal{E}Deck_2$s:

$$\left.\begin{array}{ll} (\forall n \geq 4) & P_3 \cup (n-3)K_1 \\ (\forall n \geq 4) & 2K_2 \cup (n-4)K_1 \end{array}\right\} \text{ cards: } \mathbf{1} \quad nK_1$$

$$\left.\begin{array}{ll} (\forall n \geq 4) & K_3 \cup (n-3)K_1 \\ (\forall n \geq 4) & S_4 \cup (n-4)K_1 \\ (\forall n \geq 4) & P_4 \cup (n-4)K_1 \\ (\forall n \geq 5) & P_3 \cup K_2 \cup (n-5)K_1 \\ (\forall n \geq 6) & 3K_2 \cup (n-6)K_1 \end{array}\right\} \text{ cards: } \mathbf{3} \quad K_2 \cup (n-2)K_1$$

$$\left.\begin{array}{ll} (\forall n \geq 5) & C_4 \cup (n-4)K_1 \\ (\forall n \geq 5) & \text{  } \cup (n-5)K_1 \end{array}\right\} \text{ cards: } \left\{\begin{array}{ll} \mathbf{4} & P_3 \cup (n-3)K_1 \\ \mathbf{2} & 2K_2 \cup (n-4)K_1 \end{array}\right.$$

$$\left.\begin{array}{ll} (\forall n \geq 5) & P_5 \cup (n-5)K_1 \\ (\forall n \geq 5) & K_3 \cup K_2 \cup (n-5)K_1 \\ (\forall n \geq 6) & S_4 \cup K_2 \cup (n-6)K_1 \end{array}\right\} \text{ cards: } \left\{\begin{array}{ll} \mathbf{3} & P_3 \cup (n-3)K_1 \\ \mathbf{3} & 2K_2 \cup (n-4)K_1 \end{array}\right.$$

$$\left.\begin{array}{ll} (\forall n \geq 6) & 2P_3 \cup (n-6)K_1 \\ (\forall n \geq 6) & P_4 \cup K_2 \cup (n-6)K_1 \end{array}\right\} \text{ cards: } \left\{\begin{array}{ll} \mathbf{2} & P_3 \cup (n-3)K_1 \\ \mathbf{4} & 2K_2 \cup (n-4)K_1 \end{array}\right.$$

There are 2 graphs for any order $n \geq 3$ which have $\exists ern_2 = \forall ern_2 = 1$:

- $\overline{K_2 \cup (n-2)K_1}$
- K_n

Another set of graphs with interesting behavior are those of order $n \geq 5$ with the form

$$\overline{S_i \cup (n-i)K_1} \quad \forall i (3 \leq i \leq n-2)$$

which are the only examples of graphs with $\exists ern_2 = 1$ and $\forall ern_2 > 1$. The $\forall ern_2$ of these graphs tends to be very high, and indeed if $i = 3$ then the graph has the maximal $\forall ern_2$ for that order. The other graph with maximal $\forall ern_2$ is of the form

$$\overline{2K_2 \cup (n-4)K_1}$$

which also has a low $\exists ern_2$, especially for $n \geq 8$ where $\exists ern_2 = 1$ ($\exists ern_2 = 2$ for $n = 5$, and $\exists ern_2 = 4$ for $6 \leq n \leq 7$).

6.9 Results for 3-Edge Deletion

$\mathcal{E}Deck_3$ Counts

	graph order							
	3	4	5	6	7	8	9	10
unique graphs	1	7	30	152	1040	12342	274664	12005164
total cards	1	10	88	1416	38388	1912484	145990314	16449689951
non-duplicates	1	3	8	159	5813	640649	86364522	12775798472
avg. cards/deck	1	1.4	2.9	9.3	36.91	154.957	531.523	1370.218
% non-duplicate	100	30	9	11.2	15.14	33.498	59.158	77.666

$\mathcal{E}Extensions_3$ Counts

	graph order							
	3	4	5	6	7	8	9	10
unique graphs	1	7	30	152	1040	12342	274664	12005164
total extensions	1	10	88	1416	38388	1912484	145990314	16449689951
avg. extensions	1	1.4	2.9	9.3	36.91	154.957	531.523	1370.218

3-Edge Deletion Reconstruction Numbers

		graph order					
		3	4	5	6	7	8
unique graphs		1	7	30	152	1040	12342
not reconstructible		0	5	14	28	39	45
$\exists ern_8$	1	1	2	2	3	4	5
	2			1	11	167	6553
	3			3	14	428	4721
	4				40	209	578
	5			5	17	72	169
	6				9	36	75
	7			1	2	18	59
	8			1	5	11	25
	9				6	13	20
	10			1	4	9	17
	11				2	6	13
	12			1	4	4	6
	13				1	5	10
	14					2	11
	15			1	2	7	7
	16				1	4	11
	17				1	2	10
	18				2	4	5
	19						2
% $\exists ern_8 = 2$		0	0	3	7.2	16.06	53.095

		graph order					
		3	4	5	6	7	8
unique graphs		1	7	30	152	1040	12342
not reconstructible		0	5	14	28	39	45
$\forall ern_8$	1	1	2	2	2	2	2
	2–8						
	9				1	1	1
	10			2			
	11					1	1
	12–13						
	14				1		
	15						
	16				1		1
	17			1	1	1	2
	18					2	2
	19			5	16	29	36
	20				3	4	9
	21–25						
	26				1	3	2
	27						
	28				2	3	5
	29					4	8
	30				1	3	6
	31			2	8	25	37
	32			2	11	22	38
	33				2	4	16
	34				1	2	2
	35					1	
	36						1
	37				1	6	4
	38					1	
	39				1	2	2
	40						1
	41					2	6
	42				1		1
	43						7
	44				1	1	4
	45				1	5	14
	46				1	3	13
	47			2	11	32	56
	48				5	26	61
	49				3	24	43
	50					4	14
	51						2
	52						2
	53						1
	54						
	55						3
	56					1	2
	57					1	2
	58						3
	59					1	4
	60					2	5
	61					4	10

		graph order					
		3	4	5	6	7	8
unique graphs		1	7	30	152	1040	12342
not reconstructible		0	5	14	28	39	45
$\forall ern_8$	62						8
	63				1	1	7
	64				1	9	32
	65					10	30
	66				1	15	37
	67				5	15	57
	68				2	14	50
	69				6	26	69
	70					13	44
	71				2	14	26
	72					2	13
	73						4
	74						5
	75						8
	76					2	7
	77						2
	78					2	4
	79					1	13
	80					3	8
	81					2	6
	82					1	11
	83					2	21
	84					4	24
	85					4	25
	86				1	8	45
	87					4	26
	88				1	3	45
	89					8	42
	90					7	45
	91					10	44
	92					21	60
	93				1	13	60
	94					11	49
	95				2	10	40
	96				1	7	31
	97				4	17	40
	98				2	9	24
	99				2	3	24
	100						13
	101						16
	102					1	8
	103						10
	104					1	15
	105					1	12
	106					1	22
	107					2	23
	108					1	22
	109					6	35
	110				1	4	36
	111					3	28

		graph order					
		3	4	5	6	7	8
unique graphs		1	7	30	152	1040	12342
not reconstructible		0	5	14	28	39	45
$\forall ern_8$	112					1	35
	113					4	36
	114					1	45
	115					1	43
	116					10	74
	117					6	58
	118					7	49
	119					7	49
	120				1	9	45
	121					10	63
	122					19	53
	123					7	48
	124					13	45
	125					5	35
	126					4	31
	127				2	9	38
	128				1	5	25
	129					2	21
	130				2	6	28
	131						23
	132						26
	133					1	32
	134						21
	135				2	5	47
	136					5	48
	137					1	37
	138					1	44
	139					1	34
	140					1	33
	141						32
	142						38
	143				1	3	34
	144					2	39
	145					2	54
	146					2	45
	147					5	58
	148					3	67
	149					2	56
	150						43
	151					4	58
	152				1	3	35
	153					5	47
	154					2	41
	155					10	54
	156					9	58
	157					6	39
	158					6	39
	159					8	45
	160					4	22
	161					6	37

		graph order					
		3	4	5	6	7	8
unique graphs		1	7	30	152	1040	12342
not reconstructible		0	5	14	28	39	45
$\forall ern_8$	162					7	34
	163					8	43
	164					4	34
	165					6	38
	166						41
	167					1	46
	168					4	46
	169					2	29
	170						38
	171					2	52
	172					1	49
	173				1	3	45
	174					1	51
	175						54
	176						48
	177						57
	178				2	4	55
	179					1	54
	180						45
	181						58
	182						54
	183					1	32
	184						35
	185					3	49
	186						29
	187					2	41
	188						29
	189					1	38
	190					2	36
	191					2	41
	192						53
	193					1	35
	194					3	31
	195					1	37
	196					2	43
	197					3	45
	198					5	32
	199					8	54
	200					5	51
	201					5	46
	202					2	31
	203					6	41
	204					2	31
	205					3	43
	206					8	43
	207					4	49
	208					3	51
	209					1	41
	210					2	59
	211						54

		graph order					
		3	4	5	6	7	8
unique graphs		1	7	30	152	1040	12342
not reconstructible		0	5	14	28	39	45
$\forall ern_8$	212						33
	213					3	56
	214					4	39
	215					1	31
	216						47
	217						40
	218						47
	219					1	19
	220					3	49
	221					1	31
	222					2	35
	223						27
	224					1	25
	225					2	28
	226						43
	227					2	39
	228						41
	229				2	3	37
	230						29
	231						36
	232						21
	233					1	29
	234						34
	235						33
	236					2	38
	237						33
	238					2	41
	239					2	35
	240						35
	241					1	44
	242					1	34
	243						24
	244					1	42
	245						28
	246						33
	247					1	39
	248					2	28
	249					3	41
	250						35
	251					1	36
	252					3	24
	253					2	37
	254					1	21
	255						32
	256					2	46
	257					6	46
	258					3	26
	259					3	46
	260					2	36
	261					3	37

		graph order					
		3	4	5	6	7	8
unique graphs		1	7	30	152	1040	12342
not reconstructible		0	5	14	28	39	45
$\forall ern_8$	262					4	39
	263						29
	264						26
	265						25
	266						35
	267						36
	268					1	30
	269					1	52
	270						27
	271						22
	272					2	48
	273						37
	274						36
	275						45
	276						23
	277					1	24
	278						27
	279					2	24
	280					2	24
	281						23
	282					1	29
	283						11
	284					1	25
	285						29
	286						21
	287					2	27
	288						19
	289						18
	290					1	18
	291						30
	292						20
	293						20
	294						21
	295						26
	296						14
	297						23
	298						17
	299						29
	300					1	27
	301					3	20
	302						27
	303					1	18
	304						30
	305						17
	306						20
	307						35
	308						36
	309						26
	310						18
	311						25

		graph order					
		3	4	5	6	7	8
unique graphs		1	7	30	152	1040	12342
not reconstructible		0	5	14	28	39	45
$\forall ern_8$	312					2	36
	313					1	36
	314					1	25
	315					1	24
	316					1	24
	317						28
	318					1	21
	319					1	21
	320					2	32
	321						18
	322						11
	323						20
	324						25
	325						17
	326						17
	327						24
	328						20
	329						25
	330						22
	331					3	20
	332						12
	333					1	19
	334					2	18
	335					1	18
	336						19
	337					2	12
	338						9
	339						18
	340					2	19
	341						23
	342					1	15
	343						13
	344					1	21
	345					1	21
	346						16
	347						16
	348						16
	349						19
	350						8
	351						13
	352					1	17
	353						16
	354						16
	355						15
	356						16
	357					2	18
	358						6
	359						14
	360						17
	361						18

		graph order					
		3	4	5	6	7	8
unique graphs		1	7	30	152	1040	12342
not reconstructible		0	5	14	28	39	45
$\forall ern_8$	362						15
	363						16
	364						21
	365						19
	366						14
	367						19
	368						20
	369					1	21
	370						10
	371						18
	372						11
	373						15
	374						20
	375						12
	376						21
	377						9
	378						7
	379						12
	380						14
	381						14
	382						20
	383						9
	384						12
	385						10
	386					1	22
	387						15
	388					1	15
	389					1	24
	390						15
	391						11
	392						10
	393						4
	394						11
	395						9
	396						12
	397						7
	398						8
	399						15
	400						10
	401					1	12
	402						11
	403						5
	404						11
	405						16
	406						14
	407						11
	408					1	10
	409						17
	410						20
	411					2	11

		graph order					
		3	4	5	6	7	8
unique graphs		1	7	30	152	1040	12342
not reconstructible		0	5	14	28	39	45
$\forall ern_8$	412						10
	413					1	11
	414						17
	415						5
	416						8
	417						15
	418						7
	419						3
	420						9
	421						6
	422					1	11
	423						19
	424					2	19
	425						10
	426					2	7
	427					3	14
	428						11
	429						9
	430						7
	431						2
	432						4
	433						4
	434						14
	435						4
	436						7
	437						7
	438						5
	439						13
	440					1	8
	441						5
	442						6
	443						9
	444						5
	445						5
	446						7
	447					1	11
	448						3
	449						14
	450						5
	451						9
	452						7
	453						7
	454						5
	455						4
	456						7
	457						5
	458						8
	459						8
	460						14
	461						12

		graph order					
		3	4	5	6	7	8
unique graphs		1	7	30	152	1040	12342
not reconstructible		0	5	14	28	39	45
$\forall ern_8$	462						2
	463						6
	464						4
	465						12
	466						7
	467						4
	468						2
	469						10
	470						8
	471						13
	472						8
	473						11
	474						10
	475						4
	476						9
	477						12
	478						12
	479						3
	480						3
	481						8
	482						6
	483						6
	484						4
	485						5
	486						4
	487						14
	488					1	7
	489						10
	490						4
	491						6
	492						1
	493						3
	494						8
	495						11
	496						4
	497						12
	498					2	5
	499						6
	500						3
	501						3
	502					1	2
	503						
	504						4
	505						4
	506						9
	507						8
	508						7
	509						2
	510						6
	511						6

		graph order					
		3	4	5	6	7	8
unique graphs		1	7	30	152	1040	12342
not reconstructible		0	5	14	28	39	45
$\forall ern_8$	512						4
	513						5
	514						4
	515						15
	516						1
	517						5
	518						2
	519						2
	520						1
	521						9
	522						5
	523					2	3
	524						5
	525						11
	526						4
	527						3
	528						7
	529						7
	530						6
	531						3
	532						6
	533						6
	534						3
	535						6
	536						3
	537						5
	538						2
	539						5
	540						7
	541						8
	542						5
	543						2
	544						2
	545						2
	546						2
	547						5
	548						4
	549						9
	550						3
	551						10
	552						2
	553						4
	554						2
	555						7
	556						
	557						5
	558						1
	559						3
	560						2
	561						2

		graph order					
		3	4	5	6	7	8
unique graphs		1	7	30	152	1040	12342
not reconstructible		0	5	14	28	39	45
$\forall ern_8$	562						4
	563						3
	564						3
	565						1
	566					1	5
	567						2
	568						5
	569						5
	570						1
	571						2
	572						5
	573						1
	574						
	575						2
	576						2
	577						1
	578						1
	579						8
	580						1
	581						7
	582						4
	583						1
	584						2
	585						2
	586						2
	587						2
	588						4
	589						11
	590						2
	591						3
	592						2
	593						1
	594						3
	595						3
	596						4
	597						5
	598						
	599						2
	600						5
	601						4
	602						12
	603						1
	604						1
	605						3
	606						2
	607						6
	608						3
	609						1
	610						1
	611						4

		graph order					
		3	4	5	6	7	8
unique graphs		1	7	30	152	1040	12342
not reconstructible		0	5	14	28	39	45
$\forall ern_8$	612					1	3
	613						2
	614						2
	615						1
	616						
	617						2
	618						3
	619						3
	620						
	621						3
	622						
	623						1
	624						
	625						4
	626						4
	627						3
	628						2
	629						
	630						2
	631						3
	632						2
	633						
	634						1
	635						3
	636					2	5
	637						
	638						6
	639						2
	640						4
	641						3
	642						
	643						1
	644						2
	645						3
	646						1
	647						2
	648						2
	649						1
	650						8
	651						4
	652						
	653						1
	654						4
	655						
	656						1
	657						4
	658						4
	659						1
	660–661						
	662						1

		graph order					
		3	4	5	6	7	8
unique graphs		1	7	30	152	1040	12342
not reconstructible		0	5	14	28	39	45
$\forall ern_8$	663						2
	664						
	665						4
	666						2
	667						2
	668						3
	669						
	670						1
	671–672						
	673						2
	674						1
	675						2
	676						6
	677						1
	678						3
	679–680						
	681						1
	682						1
	683						2
	684						
	685						2
	686						2
	687						1
	688						
	689						2
	690						2
	691						4
	692						6
	693						5
	694						2
	695						1
	696						1
	697						4
	698–699						
	700						3
	701						
	702						2
	703						2
	704						
	705						1
	706						2
	707						2
	708						
	709						3
	710						1
	711						
	712						2
	713						3
	714						3
	715						1

		graph order					
		3	4	5	6	7	8
unique graphs		1	7	30	152	1040	12342
not reconstructible		0	5	14	28	39	45
$\forall ern_8$	716						4
	717–720						
	721						4
	722						
	723						2
	724						3
	725–726						
	727						2
	728						2
	729–732						
	733						3
	734						
	735						3
	736–738						
	739						2
	740						
	741						2
	742–744						
	745						1
	746						2
	747–748						
	749						1
	750						
	751						2
	752						4
	753						
	754						2
	755						
	756						1
	757						2
	758						1
	759–761						
	762						2
	763						1
	764–766						
	767						2
	768–769						
	770						2
	771						2
	772–773						
	774						3
	775						
	776					2	3
	777						1
	778–779						
	780						3
	781						2
	782						2
	783						
	784						1

		graph order					
		3	4	5	6	7	8
unique graphs		1	7	30	152	1040	12342
not reconstructible		0	5	14	28	39	45
$\forall ern_8$	785						1
	786						3
	787						3
	788						
	789						1
	790						1
	791						
	792						1
	793						2
	794–796						
	797						2
	798						1
	799						
	800						1
	801–802						
	803						1
	804						1
	805						1
	806						2
	807						3
	808–816						
	817						2
	818						
	819						4
	820						1
	821–822						
	823						2
	824–830						
	831						1
	832						
	833						2
	834						2
	835–839						
	840						2
	841–845						
	846						1
	847						
	848						1
	849–850						
	851						1
	852–856						
	857						5
	858–861						
	862						1
	863						2
	864–865						
	866						2
	867						1
	868						
	869						1

		graph order					
		3	4	5	6	7	8
unique graphs		1	7	30	152	1040	12342
not reconstructible		0	5	14	28	39	45
	870						
	871						1
	872						3
	873						1
	874						1
	875						
	876						2
	877						2
	878						1
	879						1
	880						
	881						1
	882						1
	883						1
	884–886						
	887						2
	888–889						
	890						2
	891–895						
	896						2
	897						
	898						1
	899–900						
	901						1
	902–904						
$\forall ern_8$	905						2
	906–907						
	908						2
	909–915						
	916						2
	917						1
	918–920						
	921						2
	922						2
	923–928						
	929						1
	930–931						
	932						1
	933–937						
	938						1
	939						
	940						2
	941–945						
	946						2
	947–970						
	971						1
	972–985						
	986						1
	987						1
	988–991						

		graph order					
		3	4	5	6	7	8
unique graphs		1	7	30	152	1040	12342
not reconstructible		0	5	14	28	39	45
$\forall ern_8$	992						1
	993–995						
	996						2
	997						1
	998–1004						
	1005						1
	1006						1
	1007–1018						
	1019						1
	1020–1022						
	1023						2
	1024–1027						
	1028						2
	1029–1043						
	1044						3
	1045						2
	1046						3
	1047–1060						
	1061						2
	1062						
	1063						1
	1064–1070						
	1071						1
	1072						
	1073						1
	1074–1076						
	1077						2
	1078–1082						
	1083						1
	1084–1091						
	1092						1
	1093–1109						
	1110						1
	1111–1130						
	1131						1
	1132–1136						
	1137						1
	1138–1145						
	1146						1
	1147–1159						
	1160						2
	1161–1164						
	1165						2
	1166–1194						
	1195						1
	1196						
	1197						1
	1198						
	1199						1
	1200–1247						

		graph order					
		3	4	5	6	7	8
unique graphs		1	7	30	152	1040	12342
not reconstructible		0	5	14	28	39	45
$\forall ern_8$	1248						2
	1249–1251						
	1252						2
	1253–1254						
	1255						1
	1256–1265						
	1266						1
	1267–1287						
	1288						2
	1289–1306						
	1307						2
	1308–1417						
	1418						1
	1419–1480						
	1481						2
	1482–1499						
	1500						1
	1501–1522						
	1523						2
	1524–1654						
	1655						1
	1656–1657						
	1658						1
	1659–1740						
	1741						2
	1742–2004						
	2005						2

6.10 Results for 4-Edge Deletion

$\mathcal{E}Deck_4$ Counts

	graph order					
	4	5	6	7	8	9
unique graphs	4	26	147	1035	12337	274659
total cards	5	64	1278	45762	3576158	459990744
non-duplicates	2	7	75	3059	580124	199899308
avg. cards/deck	1.2	2.5	8.7	44.21	289.873	1674.770
% non-duplicate	40	11	5.9	6.68	16.222	43.457

$\mathcal{E}Extensions_4$ Counts

	graph order					
	4	5	6	7	8	9
unique graphs	4	26	147	1035	12337	274659
total extensions	5	64	1278	45762	3576158	459990744
avg. extensions	1.2	2.5	8.7	44.21	289.873	1674.770

4-Edge Deletion Reconstruction Numbers

		graph order			
		4	5	6	7
unique graphs		4	26	147	1035
not reconstructible		2	16	44	80
$\exists ern_8$	1	2	2	2	3
	2		1	5	23
	3			5	137
	4		2	8	199
	5			5	177
	6			11	109
	7			14	78
	8		1	9	45
	9			7	32
	10		2	5	21
	11			8	22
	12		1		9
	13				13
	14				5
	15		1	2	6
	16			2	6
	17			4	7
	18			2	3
	19			2	4
	20			3	4
	21			1	2
	22			1	11
	23				5
	24				5
	25			1	9
	26				3
	27			1	2
	28				
	29			2	3
	30			2	6
	31				1
	32			1	4
	33				1
% $\exists ern_8 = 2$		0	4	3.4	2.22

		graph order			
		4	5	6	7
unique graphs		4	26	147	1035
not reconstructible		2	16	44	80
$\forall ern_8$	1	2	2	2	2
	2–12				
	13				1
	14				
	15		2	1	
	16–26				
	27				1
	28–30				
	31				1
	32			1	
	33		2	5	10
	34			12	29
	35		2	6	15
	36–52				
	53				1
	54–55				
	56			1	
	57				
	58			1	
	59			1	2
	60				2
	61				
	62				1
	63			1	9
	64		2	6	12
	65			4	17
	66			6	24
	67			2	25
	68			2	2
	69				2
	70–71				
	72				1
	73				1
	74				
	75			1	
	76–81				
	82			1	2
	83–89				
	90				1
	91				
	92			1	
	93–94				
	95				1
	96				
	97				1
	98				
	99			1	1
	100				1
	101				
	102			1	1

		graph order			
		4	5	6	7
unique graphs		4	26	147	1035
not reconstructible		2	16	44	80
$\forall ern_8$	103				1
	104				2
	105				2
	106				1
	107				
	108				2
	109			2	3
	110			1	7
	111			3	7
	112			4	23
	113			5	23
	114			2	27
	115				8
	116				9
	117				1
	118				8
	119–122				
	123				2
	124–135				
	136				1
	137–159				
	160				1
	161–162				
	163				1
	164				
	165			1	3
	166–169				
	170				2
	171				2
	172				1
	173				1
	174				2
	175				6
	176			1	1
	177				4
	178				9
	179				4
	180			2	8
	181				22
	182				9
	183				10
	184			2	12
	185			2	19
	186			4	14
	187			2	3
	188				3
	189				4
	190				3
	191–212				
	213				1

		graph order			
		4	5	6	7
unique graphs		4	26	147	1035
not reconstructible		2	16	44	80
$\forall ern_8$	214–218				
	219				1
	220				
	221				1
	222–225				
	226				1
	227–230				
	231				2
	232–241				
	242				1
	243–245				
	246				1
	247–248				
	249			1	3
	250				
	251				1
	252				
	253				1
	254–255				
	256				1
	257–258				
	259				1
	260				
	261				1
	262				2
	263				2
	264				1
	265				2
	266				7
	267				
	268				2
	269				3
	270				5
	271				3
	272			1	8
	273				7
	274				5
	275				7
	276				5
	277				5
	278				7
	279			1	11
	280				8
	281				8
	282				4
	283				6
	284			1	7
	285				2
	286			1	5
	287			1	3

		graph order			
		4	5	6	7
unique graphs		4	26	147	1035
not reconstructible		2	16	44	80
$\forall ern_8$	288				2
	289			1	3
	290–292				
	293			2	2
	294–298				
	299				1
	300–314				
	315				1
	316–321				
	322				1
	323–324				
	325				1
	326–346				
	347				1
	348				2
	349–350				
	351				2
	352–360				
	361				1
	362–363				
	364				1
	365–366				
	367				3
	368–378				
	379			1	2
	380–382				
	383				1
	384–385				
	386				1
	387				1
	388				1
	389				1
	390				1
	391				5
	392				1
	393				2
	394				3
	395				6
	396				4
	397				3
	398				1
	399				
	400				11
	401				7
	402				4
	403				1
	404				5
	405				4
	406				1
	407				3

		graph order			
		4	5	6	7
unique graphs		4	26	147	1035
not reconstructible		2	16	44	80
$\forall ern_8$	408				2
	409				7
	410				3
	411			1	4
	412				10
	413				2
	414				1
	415				
	416				6
	417				
	418				2
	419				4
	420				
	421				2
	422				2
	423			1	1
	424–425				
	426				2
	427–429				
	430			2	2
	431–451				
	452				1
	453–471				
	472				1
	473–499				
	500				2
	501				
	502				1
	503–511				
	512				1
	513				2
	514–522				
	523				1
	524–526				
	527				1
	528				1
	529–533				
	534				2
	535				
	536				1
	537–538				
	539				1
	540				1
	541				
	542				1
	543–548				
	549				1
	550				
	551				2
	552				

		graph order			
		4	5	6	7
unique graphs		4	26	147	1035
not reconstructible		2	16	44	80
$\forall ern_8$	553				1
	554				
	555				1
	556				3
	557				1
	558				
	559				1
	560				
	561				1
	562				1
	563				3
	564				1
	565				2
	566				4
	567–568				
	569				2
	570				
	571				1
	572				3
	573–574				
	575				4
	576				2
	577				1
	578				
	579				4
	580				3
	581				
	582				6
	583				1
	584				4
	585				4
	586				1
	587				2
	588				4
	589				2
	590				4
	591–592				
	593				2
	594–595				
	596				2
	597				2
	598–604				
	605				2
	606–608				
	609			2	2
	610–676				
	677				1
	678				2
	679–691				
	692				1

		graph order			
		4	5	6	7
unique graphs		4	26	147	1035
not reconstructible		2	16	44	80
$\forall ern_8$	693–694				
	695				1
	696–704				
	705				1
	706–717				
	718				1
	719–721				
	722				2
	723–726				
	727				1
	728				1
	729				1
	730–734				
	735				1
	736–741				
	742				1
	743–746				
	747				1
	748–755				
	756				1
	757				1
	758				1
	759–768				
	769				1
	770–774				
	775				1
	776–779				
	780				1
	781				1
	782				1
	783				2
	784–785				
	786				1
	787–789				
	790				3
	791–794				
	795				3
	796–798				
	799				1
	800				3
	801				2
	802				3
	803				1
	804				
	805				2
	806–808				
	809				2
	810				1
	811				1
	812–813				

		graph order			
		4	5	6	7
unique graphs		4	26	147	1035
not reconstructible		2	16	44	80
$\forall ern_8$	814				2
	815				1
	816				1
	817				2
	818				
	819				2
	820–823				
	824				1
	825–831				
	832				2
	833				
	834				2
	835–837				
	838				2
	839–888				
	889				2
	890–907				
	908				1
	909–955				
	956				1
	957–963				
	964				1
	965–970				
	971				1
	972–989				
	990				1
	991–995				
	996				1
	997–999				
	1000				1
	1001–1005				
	1006				1
	1007–1011				
	1012				1
	1013				
	1014				1
	1015–1023				
	1024				1
	1025–1035				
	1036				2
	1037–1047				
	1048				1
	1049–1054				
	1055				1
	1056–1062				
	1063				1
	1064–1066				
	1067				1
	1068–1069				
	1070				1

		graph order			
		4	5	6	7
unique graphs		4	26	147	1035
not reconstructible		2	16	44	80
	1071				
	1072				1
	1073–1076				
	1077				1
	1078–1079				
	1080				1
	1081				1
	1082				1
	1083–1084				
	1085				1
	1086–1089				
	1090				2
	1091–1092				
	1093				1
	1094				
	1095				1
	1096–1103				
	1104				1
	1105				1
	1106–1107				
	1108				1
	1109–1117				
	1118				2
	1119–1128				
$\forall ern_8$	1129				1
	1130–1134				
	1135				1
	1136				2
	1137				2
	1138–1205				
	1206				1
	1207–1282				
	1283				1
	1284–1361				
	1362				1
	1363–1372				
	1373				1
	1374–1383				
	1384				1
	1385–1388				
	1389				1
	1390–1399				
	1400				1
	1401–1414				
	1415				1
	1416–1435				
	1436				1
	1437–1448				
	1449				1
	1450–1470				

		graph order			
		4	5	6	7
unique graphs		4	26	147	1035
not reconstructible		2	16	44	80
$\forall ern_8$	1471				1
	1472–1474				
	1475				1
	1476–1486				
	1487				2
	1488				
	1489				1
	1490–1493				
	1494				1
	1495–1506				
	1507				1
	1508–1517				
	1518				2
	1519–1628				
	1629				1
	1630–1722				
	1723				1
	1724–1778				
	1779				1
	1780				1
	1781–1896				
	1897				1
	1898–1931				
	1932				1
	1933–1941				
	1942				2
	1943–1972				
	1973				2
	1974–2103				
	2104				1
	2105–2358				
	2359				1
	2360–2483				
	2484				1
	2485–2554				
	2555				2
	2556–3229				
	3230				2

6.11 Results for 5-Edge Deletion

$\mathcal{E}Deck_5$ Counts

	graph order					
	4	5	6	7	8	9
unique graphs	2	20	138	1025	12326	274648
total cards	2	40	960	41385	4467068	979474682
non-duplicates	1	6	36	1281	280038	254419388
avg. cards/deck	1	2	7	40.38	362.410	3566.291
% non-duplicate	50	15	3.8	3.10	6.269	25.975

$\mathcal{E}Extensions_5$ Counts

	graph order					
	4	5	6	7	8	9
unique graphs	2	20	138	1025	12326	274648
total extensions	2	40	960	41385	4467068	979474682
avg. extensions	1	2	7	40.38	362.410	3566.291

5-Edge Deletion Reconstruction Numbers

		graph order	
		4	5
unique graphs		2	20
not reconstructible		0	14
$\exists ern_8$	1	2	2
	2–4		
	5		1
	6–7		
	8		1
	9–13		
	14		1
	15–20		
	21		1
$\forall ern_8$	1	2	2
	2–20		
	21		2
	22–53		
	54		2
% $\exists ern_8 = 2$		0	0

6.12 Results for 6-Edge Deletion

$\mathcal{E}Deck_6$ Counts

	graph order					
	4	5	6	7	8	9
unique graphs	1	14	123	1004	12302	274623
total cards	1	24	654	31874	4193822	1413960799
non-duplicates	1	6	27	547	101581	162168340
avg. cards/deck	1	1.7	5	31.75	340.906	5148.734
% non-duplicate	100	25	4.1	1.72	2.422	11.469

$\mathcal{E}Extensions_6$ Counts

	graph order					
	4	5	6	7	8	9
unique graphs	1	14	123	1004	12302	274623
total extensions	1	24	654	31874	4193822	1413960799
avg. extensions	1	1.7	5	31.75	340.906	5148.734

6-Edge Deletion Reconstruction Numbers

		graph order	
		4	5
unique graphs		1	14
not reconstructible		0	10
$\exists ern_8$	1	1	2
	2–9		
	10		1
	11–18		
	19		1
$\forall ern_8$	1	1	2
	2–27		
	28		2
% $\exists ern_8 = 2$		0	0

6.13 Results for 7-Edge Deletion

$\mathcal{E}Deck_7$ Counts

	graph order			
	5	6	7	8
unique graphs	8	102	963	12246
total cards	12	409	22121	3299474
non-duplicates	4	25	233	36231
avg. cards/deck	1.5	4.0	22.97	269.433
% non-duplicate	33	6	1.05	1.098

$\mathcal{E}Extensions_7$ Counts

	graph order			
	5	6	7	8
unique graphs	8	102	963	12246
total extensions	12	409	22121	3299474
avg. extensions	1.5	4	22.97	269.433

7-Edge Deletion Reconstruction Numbers

		graph order
		5
unique graphs		8
not reconstructible		6
$\exists ern_8$	1	2
$\forall ern_8$	1	2
% $\exists ern_8 = 2$		0

6.14 Results for 8-Edge Deletion

$\mathcal{E}Deck_8$ Counts

	graph order			
	5	6	7	8
unique graphs	4	78	898	12131
total cards	5	240	14292	2328920
non-duplicates	2	24	160	14519
avg. cards/deck	1.2	3.1	15.92	191.981
% non-duplicate	40	10	1.12	0.623

$\mathcal{E}Extensions_8$ Counts

	graph order			
	5	6	7	8
unique graphs	4	78	898	12131
total extensions	5	240	14292	2328920
avg. extensions	1.2	3	15.92	191.981

8-Edge Deletion Reconstruction Numbers

		graph order
		5
unique graphs		4
not reconstructible		2
$\exists ern_8$	1	2
$\forall ern_8$	1	2
% $\exists ern_8 = 2$		0

7 Open Questions and Future Directions

In the course of this project a number of new questions have arisen, many of which (to the author's knowledge) have not been posed in the past. Some of these questions are theoretical, while others can be investigated computationally by building upon the programs developed for this project.

Amongst those questions which might be answered via computational means, are the following types of reconstruction numbers, which apply to all forms of graph reconstruction:

- For a given graph G, what is the number of subdecks of size $\exists rn(G)$ which reconstruct G?
- For a given graph G and $\exists rn(G) \leq s \lneq \forall rn(G)$, what is the number of subdecks of size s which reconstruct G?

A number of the techniques explored in this project would be useful for reconstruction of structures other than graphs, but even within graph reconstruction there is still much to be explored. For instance, so far all descriptions of reconstruction have involved constructing the deck by removing some element of the graph in every possible way. This thought brings to mind the following new ways of constructing a deck:

- Add a vertex in every possible way
- Complement an edge in every possible way
- Add or subtract a vertex in every possible way

These can then be iterated to form k-reconstruction forms in the obvious ways. It should be noted that edge-addition reconstruction is the same as edge-deletion reconstruction of the complement.

Some questions arose regarding the relationships between the graphs. These questions seemed best explored by constructing meta-graphs, where each graph of interest is a vertex, and each edge denotes a relationship between them. These questions include the following, and their generalizations to other forms of reconstruction:

- A meta-graph of all graphs on n vertices, where each edge is colored according to the number of shared cards. Of special interest is those edges colored with the maximal $\forall rn - 1$ for that order.
- A meta-digraph of all graphs on $\leq n$ vertices, where a edge from G to C is colored according to $m(Deck(G);\, C)$
- A meta-digraph of all graphs on n vertices and e edges, where a edge from G to C is colored according to $m(\mathcal{E}Deck(G);\, C)$

Forms of these questions can be asked for any form of reconstruction, although they are only answerable computationally if the meta-graph is finite.

One of the major goals of this project was to expose patterns in reconstruction numbers which would then be investigated theoretically. For instance, when analyzing the results presented in Section 6.1, it became apparent that there was a non-trivial relationship between the universal reconstruction number of graphs and graphs constructed from them by adding a disconnected or completely-connected vertex. k-vertex-deletion reconstruction numbers. These patterns gave rise to questions such as these:

- What is the relationship between $\forall vrn_1(G)$ and $\forall vrn_1(G \cup K_1)$ or $\forall vrn_1(\overline{\overline{G} \cup K_1})$?
- $\exists vrn_{k>1}$ appears to almost always be 3, is there a proof of this?
- The smallest graph for which $\exists vrn_k = 3$ appears to follow a strictly monotonic progression as k increases. For any given k, what is the smallest $|V(G)|$ such that $\exists vrn_k(G) = 3$?
- For any given k, what is the smallest n such that all graphs on n vertices are reconstructible? In the results so far the value is the same as for the previous question, does this hold in general?
- $\forall vrn_k$ behaves very differently from $\exists vrn_3$. What is is asymptotic behavior as $|V(G)|$ increases?

While extensive effort has been put forth in this project to optimize both the algorithms and the implementation thereof, there is yet room for improvement. Since majority of attention was paid to those aspects which were measured to be critical to 1-vertex-deletion computations, there are a number of areas in k-vertex-deletion and k-edge-deletion which are not optimal. A prime example is the computation of the existential reconstruction number, which in many of the high-k cases was, by far, the most computationally expensive step. It is believed that computing $Extensions_{k>2}$, $\mathcal{E}Extensions_{k>2}$, $Deck_{k>2}$, and $\mathcal{E}Deck_{k>2}$ could also be improved considerably. Even given the existing implementation, it is believed that computations of $(k \geq 4)$-vertex-deletion results for all graphs on 9 vertices could be achieved with use of the CASCI cluster [16] within a time-span of a few months.

A Vertex Reconstruction Results With Fixed Sized Cards

In this section some of the same results presented in Section 6 are reorganized according to the size of the cards. For example, in appendix A.1 the number of vertices deleted is $|V(G)| - 2$ so that in each column all cards have exactly 2 vertices.

A.1 Results for $(|V(G)| - 2)$-Vertex Deletion

		graph order					
		3	4	5	6	7	8
unique graphs		4	11	34	156	1044	12346
not reconstructible		0	7	30	152	1040	12342
$\exists vrn_{\|V(G)\|-2}$	3	4					
	4–5						
	6		4				
	7–9						
	10			4			
	11–14						
	15				4		
	16–20						
	21					4	
	22–27						
	28						4
$\forall vrn_{\|V(G)\|-2}$	3	4					
	4–5						
	6		4				
	7–9						
	10			4			
	11–14						
	15				4		
	16–20						
	21					4	
	22–27						
	28						4

A.2 Results for $(|V(G)|-3)$-Vertex Deletion

		graph order				
		4	5	6	7	8
unique graphs		11	34	156	1044	12346
not reconstructible		0	4	78	854	11935
$\exists vrn_{\|V(G)\|-3}$	3	8				
	4	3	2			
	5					
	6		4			
	7		8			
	8		9	2		
	9		7	2		
	10			4		
	11					
	12			2		
	13			8		
	14			2		
	15			10	4	
	16			8	6	
	17			14	2	
	18			22	2	
	19			4	2	
	20				2	
	21				4	
	22					
	23				2	2
	24					4
	25				14	4
	26				2	4
	27				6	
	28				22	2
	29				8	6
	30				16	4
	31				30	2
	32				22	2
	33				44	
	34				2	4
	35					2
	36					2
	37					2
	38					4
	39					4
	40					6
	41					2
	42					8
	43					8
	44					8
	45					8
	46					12
	47					32
	48					10
	49					28
	50					30
	51					24
	52					56
	53					66
	54					65

		graph order				
		4	5	6	7	8
unique graphs		11	34	156	1044	12346
not reconstructible		0	4	78	854	11935
$\forall vrn_{\|V(G)\|-3}$	3	2				
	4	9				
	5–7					
	8		6			
	9		9			
	10		15			
	11–16					
	17			4		
	18			6		
	19			68		
	20–30					
	31				8	
	32				6	
	33				16	
	34				160	
	35–50					
	51					6
	52					6
	53					18
	54					24
	55					357

A.3 Results for $(|V(G)| - 4)$-Vertex Deletion

		graph order			
		5	6	7	8
unique graphs		34	156	1044	12346
not reconstructible		0	0	20	1937
$\exists vrn_{\lvert V(G)\rvert-4}$	3	34	8		
	4		30		
	5		34	10	
	6		30	12	
	7		32	24	6
	8		16	66	6
	9		2	90	10
	10		2	126	21
	11		2	88	8
	12			96	16
	13			70	48
	14			76	66
	15			54	100
	16			66	170
	17			74	193
	18			54	212
	19			62	346
	20			30	440
	21			14	368
	22			6	310
	23				318
	24			4	365
	25				375
	26			2	436
	27				322
	28				420
	29				460
	30				488
	31				452
	32				434
	33				442
	34				442
	35				450
	36				354
	37				370
	38				351
	39				403
	40				300
	41				304
	42				212
	43				169
	44				70
	45				58
	46				36
	47				22
	48				20
	49				6
	50				4
	51				2
	52				2
	53–55				
	56				2

		graph order			
		5	6	7	8
unique graphs		34	156	1044	12346
not reconstructible		0	0	20	1937
$\forall vrn_{\lvert V(G)\rvert-4}$	3	7			
	4	19			
	5	8			
	6–9				
	10		6		
	11		2		
	12		4		
	13		98		
	14		46		
	15–25				
	26			6	
	27				
	28			4	
	29			4	
	30			38	
	31			88	
	32			400	
	33			342	
	34			142	
	35–55				
	56				8
	57				2
	58–59				
	60				4
	61				14
	62				22
	63				98
	64				214
	65				548
	66				1065
	67				3062
	68				3362
	69				2010

A.4 Results for $(|V(G)| - 5)$-Vertex Deletion

		graph order		
		6	7	8
unique graphs		156	1044	12346
not reconstructible		0	0	8
$\exists vrn_{\|V(G)\|-5}$	3	150	240	
	4	4	396	128
	5	2	216	652
	6		106	1738
	7		44	2290
	8		20	2285
	9		10	1874
	10		4	1216
	11		2	755
	12		6	490
	13			304
	14			207
	15			152
	16			72
	17			40
	18			38
	19			36
	20			5
	21			16
	22			6
	23			6
	24			6
	25			2
	26			4
	27			
	28			4
	29			4
	30			2
	31–32			
	33			2
	34–36			
	37			2
	38–40			
	41			2

		graph order		
		6	7	8
unique graphs		156	1044	12346
not reconstructible		0	0	8
$\forall vrn_{\lvert V(G)\rvert-5}$	3	8		
	4	56		
	5	90		
	6	2		
	7–11			
	12		4	
	13		2	
	14		14	
	15		76	
	16		216	
	17		532	
	18		172	
	19		28	
	20–36			
	37			8
	38			
	39			4
	40			3
	41			16
	42			6
	43			51
	44			76
	45			263
	46			532
	47			1282
	48			2451
	49			3902
	50			2602
	51			840
	52			118
	53			96
	54			88

A.5 Results for $(|V(G)| - 6)$-Vertex Deletion

		graph order		
		7	8	9
unique graphs		1044	12346	274668
$\exists vrn_{\lvert V(G)\rvert-6}$	3	1044	9592	2760
	4		2464	45713
	5		216	145271
	6		36	62156
	7		18	14434
	8		8	3018
	9		2	678
	10		4	244
	11		4	160
	12			68
	13			46
	14		2	34
	15			26
	16			20
	17			8
	18			2
	19			8
	20			2
	21			2
	22			4
	23			2
	24			4
	25–30			
	31			2
	32–35			
	36			4
	37–49			
	50			2

		graph order		
		7	8	9
unique graphs		1044	12346	274668
$\forall vrn_{\lvert V(G)\rvert-6}$	3	16		
	4	496		
	5	520		
	6	12		
	7–13			
	14		5	
	15		4	
	16		36	
	17		111	
	18		1020	
	19		2820	
	20		3598	
	21		3212	
	22		1254	
	23		248	
	24		32	
	25		6	
	26–49			
	50			5
	51			
	52			2
	53			8
	54			4
	55			75
	56			98
	57			157
	58			242
	59			360
	60			1940
	61			3798
	62			6426
	63			11409
	64			21181
	65			32518
	66			42127
	67			46011
	68			38908
	69			30087
	70			18289
	71			10642
	72			5843
	73			2984
	74			1216
	75			224
	76			64
	77			46
	78			
	79			4

A.6 Results for $(|V(G)| - 7)$-Vertex Deletion

		graph order	
		8	9
unique graphs		12346	274654
$\exists vrn_{\|V(G)\|-7}$	3	12334	270869
	4	8	3448
	5	2	228
	6	2	48
	7		18
	8		16
	9		3
	10		12
	11		4
	12		2
	13		
	14		2
	15		
	16		4
$\forall vrn_{\|V(G)\|-7}$	3	266	
	4	8208	
	5	3584	
	6	284	
	7	4	
	8–15		
	16		9
	17		271
	18		3704
	19		14270
	20		21982
	21		60137
	22		79796
	23		48632
	24		20508
	25		17345
	26		5768
	27		1820
	28		316
	29		92
	30		4

References

[1] K. Asciak, M. Francalanza, J. Lauri, and W. Myrvold. A survey of some open questions in reconstruction numbers. *Pre-print*, 2006.

[2] K. Asciak and J. Lauri. On disconnected graph with large reconstruction number. *Ars Combinatoria*, 62:173–181, 2002.

[3] J. Baldwin. Graph Reconstruction Numbers. Master's project, Rochester Institute of Technology, 102 Lomb Memorial Drive, Rochester, NY 14623-5608, July 2004. http://www.cs.rit.edu:8080/ms/static/eh/2003/4/jlb5851/.

[4] B. Bollobás. Almost every graph has reconstruction number three. *J. Graph Theory*, 14(1):1–4, 1990.

[5] A. Bowler, P. Brown, and T. Fenner. Families of pairs of graphs with a large number of common cards. *Pre-Print*, 2006.

[6] F. Harary and M. Plantholt. The graph reconstruction number. *J. Graph Theory*, 9(4):451–454, 1985.

[7] J. Lauri. Vertex-deleted and edge-deleted subgraphs. http://staff.um.edu.mt/jlau/research/research.html, 1992. University of Malta.

[8] J. Lauri and R. Scapellato. *Topics in Graph Automorphisms and Reconstruction*. Cambridge University Press, The Edinburgh Building, Cambridge CB2 2RU, UK, May 2003.

[9] B. D. McKay. *nauty User's Guide Version 2.2*. Computer Science Department, Australian National University, ACT 0200, Australia. http://cs.anu.edu.au/∼bdm/nauty.

[10] B. D. McKay. Small graphs are reconstructible. *Australas. J. Combin.*, 15:123–126, 1997.

[11] B. McMullen. Graph Reconstruction Numbers. Master's project, Rochester Institute of Technology, 102 Lomb Memorial Drive, Rochester, NY 14623-5608, June 2005. http://www.cs.rit.edu:8080/ms/static/spr/2004/4/bmm3056/.

[12] B. McMullen and S. P. Radziszowski. Graph reconstruction numbers. *Journal of Combinatorial Mathematics and Combinatorial Computing*, 62, August 2007.

[13] R. Molina. Correction of a proof on the ally-reconstruction number of a disconnected graph. *Ars Combinatoria*, 40:59–64, 1995.

[14] W. Myrvold. The ally-reconstruction number of a disconnected graph. *Ars Combinatoria*, 28:123–127, 1989.

[15] W. Myrvold. The ally-reconstruction number of a tree with five or more vertices is three. *J. Graph Theory*, 14(2):149–166, 1990.

[16] RIT. RIT CASCI cluster. http://cluster.rit.edu.

[17] ROCKS clustering software. http://www.rocksclusters.org/.