Genetic Algorithms and Book Embeddings: A Dual Layered Approach

Shannon Overbay Paul De Palma Marshall Hurson Tiffany Arnold Andrew Pierce
Gonzaga University
Spokane, WA 99258 USA
overbay@gonzaga.edu

Abstract

The genetic algorithm (GA) has been applied to a wide variety of problems where truly optimal solutions are computationally intractable. One such problem is the book embedding problem from graph theory. A book embedding is an ordering of vertices along a line (the spine) with the edges embedded in half-planes (the pages) extruding from the line so that the edges do not cross. The goal is to find the minimal number of half-planes needed to embed a given graph. This problem is known to be NP-complete. The paper shows that the GA can be used to generate counter-examples to conjectured minimum bounds.

Introduction

The idea that there might be something to be gained by applying the principles of Darwinian natural selection to computing is not new. Turing himself proposed evolutionary search as early as 1948. Though John Holland at the University of Michigan coined the term "genetic algorithm" in the mid-seventies, the GA was not widely studied until 1989 when D.E. Goldberg showed that it could be used to solve a number of difficult problems (Holland, 1975; Goldberg, 1989; Luger and Stubblefield, 2009). At least some of those difficult problems are in the equivalence class "consisting of the 'hardest' problems in NP," namely the class of NP-complete problems (Garey and Johnson, 1979: 14). Researchers who investigate problems in this class must content themselves with heuristic approaches, constraint relaxation, and, crucially, with sub-optimal solutions.

This paper argues that the GA can be effectively used in a problem from graph theory known as book embedding. A book embedding of a graph is an ordering of the vertices along a line in 3-space (the spine) along with an assignment of each edge to a single half-plane extruding from the spine (a page) such that the edges do not cross each other or the spine. The goal is to find the minimal number of pages needed to embed a given graph in a book. The study of book embedding is of interest both as a theoretical area of topological graph theory and as a practical subject with numerous applications.

There has been a recent boom in interest in book embedding, motivated by its usage in modeling a variety of problems. Book embeddings have been applied to fault-tolerant VLSI design, sorting with parallel stacks, single-row routing, and complexity theory (Chung, Leighton, and Rosenberg, 1987). Beyond computer science applications, book embeddings can be used to model and solve traffic flow problems (Kainen, 1990) and to study RNA folding (Gliess and Stadler, 1999). Due to its contributions to both theory and application, book embedding has been the subject of extensive study. Dujmović and Wood (2004) give a summary of many of the known applications and results in book embeddings.

The book embedding problem is also known to be NP-complete (Garey, et al., 1980). Informally, this means that an exhaustive search through the space of possible embeddings for a minimal embedding is intractable. As an NP-complete problem, the task of determining an optimal book embedding for an arbitrary graph is difficult. This is where methods such as the GA may be of great assistance. The contribution of the GA to the book embedding problem is two-fold: 1) generating novel embeddings and 2) generating counter-examples to conjectured bounds. In this paper, we provide an overview of book embedding, an overview of the GA, and present very encouraging results for graphs of various configurations. We also describe a novel technique that we call the "Dual-Layered Approach" (DUA) which we are currently investigating.

The Book Embedding Problem

An n-book is a topological structure consisting of n half-planes (the pages) joined together at a common line (the spine). A graph is embedded in a book by placing the vertices along the spine and each edge on a single page of the book so that no two edges cross on any page. The book-thickness of a graph G, denoted $bt(G)$, is the smallest number n for which G has an n-book embedding.

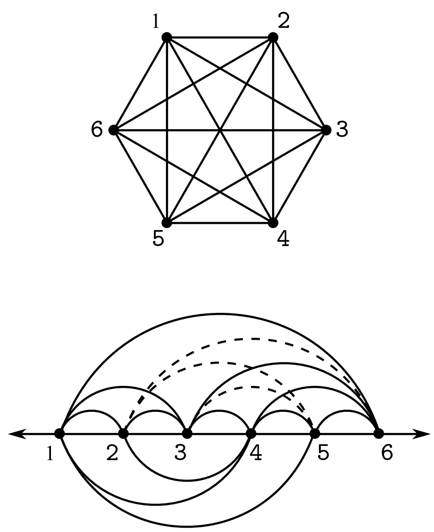

Figure 1: ***A three-page book embedding of K_6***

A book embedding of the complete graph on six vertices K_6 in a three-page book is given in Figure 1. The vertices of the graph lie on the spine. The first page of the book consists of the solid edges above the spine, the second page of the book is comprised of the solid edges below the spine, and the dashed edges above the spine form the third page of the book. These pages may be represented as lists of edges as follows:

Page 1: {(1,2), (1,3), (1,6), (2,3), (3,4), (3,6), (4,5), (4,6), (5,6)}

Page 2: {(1,4), (1,5), (2,4)}

Page 3: {(2,5), (2,6), (3,5)}

When embedding a graph in a book, there are two important considerations. First, the ordering of the vertices along the spine must be determined. For a graph with m vertices, there will be $m!$ possible arrangements of the vertices along the spine. Even if we account for the m cyclic rotations of this linear ordering and the reflections of each of these, there are still $(m-1)!/2$ vertex orderings to consider. Once the vertex order is determined, then the edges must be embedded on the pages of the book. As the numbers of vertices and edges increase, finding the book-thickness of a graph becomes computationally intractable. Garey, Johnson, Miller, and Papadimitriou (1980) proved that the problem of determining the book-thickness of an arbitrary graph is NP-complete, even with a pre-specified vertex ordering.

Despite the difficulty of the general book embedding problem, there are known bounds for the book-thickness of a graph with m vertices and q edges. We include Bernhart and Kainen's (1979) proof here since we use methods from this proof to form our custom cost function for our book embedding GA.

Theorem 1 *If G is a finite simple graph with $m \geq 4$ vertices and q edges, then*

$$bt(G) \geq \frac{q-m}{m-3}.$$

Proof: Place the m vertices on the spine of the book. The $m-1$ edges connecting consecutive vertices along the spine may be placed on any page of the book without creating edge crossings. The edge connecting the first and last vertex on the spine may also be placed on any page of the book, above all other edges, without causing crossings. Ignoring the m edges of this cycle, there may be at most $m-3$ additional edges on any page of the book, corresponding to a complete triangulation of the interior of this cycle. Thus an n-page book embedding of a graph with m vertices may have at most $m + n(m-3)$ edges; m for the outer cycle and $m-3$ for a complete triangulation of this cycle on each of the n pages.

Now we have:

$$q \leq m + n(m-3)$$

Solving for n yields the desired result:

$$n \geq \frac{q-m}{m-3},$$ thus completing the proof.

The complete graph on m vertices, K_m, is formed by connecting each pair of distinct vertices with an edge. The bound for book-thickness given in Theorem 1 may now be used to determine the optimal book-thickness of K_m in the following theorem (Bernhart and Kainen, 1979).

Theorem 2 *If* $m \geq 4$, *then* $bt(K_m) = \left\lceil \frac{m}{2} \right\rceil$.

Proof: The graph K_m has $q = \binom{m}{2} = \frac{m(m-1)}{2}$ edges, corresponding to each distinct pairs of vertices. From Theorem 1, it follows that

$$bt(K_m) \geq \frac{\left\lceil \frac{m(m-1)}{2} \right\rceil - m}{m-3} = \frac{m}{2}$$

Since the book-thickness must be an integer, it follows that $bt(K_m) \geq \left\lceil \frac{m}{2} \right\rceil$.

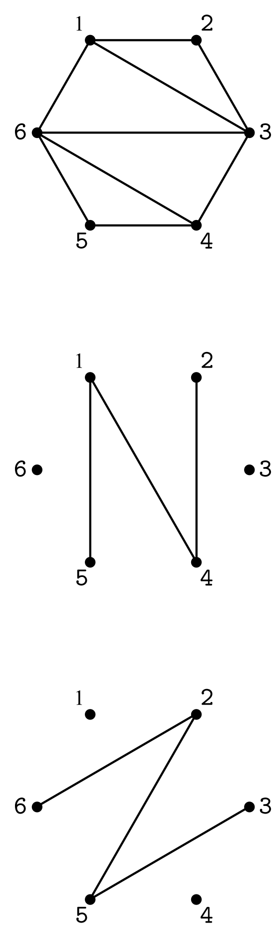

Figure 2: ***Rotated zig-zag triangulations of an m-cycle. Each rotation corresponds to one of the m/2 pages in a book embedding of*** K_m.

To show that $\left\lceil \frac{m}{2} \right\rceil$ pages are sufficient, we first observe that when m is even, then $\left\lceil \frac{m}{2} \right\rceil = \left\lceil \frac{m-1}{2} \right\rceil = \frac{m}{2}$. Hence, we may assume that m is even. Since the graph K_{m-1} is a sub-graph of K_m, we will show that K_m is embeddable in a book with $\frac{m}{2}$ pages. The corresponding embedding of K_{m-1} will follow after removing one vertex and its adjoining edges from the embedding of K_m.

The desired embedding of K_m is attained by rotating the interior edges of a zig-zag triangulation of the outer m-cycle through $\frac{m}{2}$ successive positions, as shown in Figure 2. The edges of each rotation are embedded on a separate page and the m edges of the outer cycle are placed on the first page. It is easily seen that each of the $m + \frac{m}{2}(m-3) = \binom{m}{2}$ edges of of K_m are embedded exactly once, showing that $bt(K_m) \leq \left\lceil \frac{m}{2} \right\rceil$. This gives us our desired result.

For example, the K_6 graph shown in Figure 1 has 15 edges, 6 on the outer cycle and $\frac{6}{2}(6-3) = 9$ in the interior of the cycle. By Theorem 2, the optimal book-thickness of this graph is $bt(K_6) = \left\lceil \frac{6}{2} \right\rceil = 3$. Figure 2 depicts the three rotated triangulations of the 6-cycle that correspond to each of the three pages of the book embedding of K_6 given in Figure 1.

The optimal book-thickness is known for several classes of graphs (Dujmović and Wood, 2004). When a graph is planar, it can be shown that the book thickness is never more than four pages (Yannakakis, 1986). Further, if the graph is planar and does not contain triangles, the book thickness is at most two pages (Kainen and Overbay, 2007). Although the optimal book-thickness is known for the complete graph, there are other similar graphs for which the optimal number is not known. One such graph is the complete bipartite graph, $K_{m,n}$. This graph consists of a set of m vertices and a set of n vertices, with all possible connections from the m-set to the n-set and no connections within each set. The book-thickness of $K_{m,n}$ has been shown to be at most the smaller of n and $\left\lceil \frac{2n+m}{4} \right\rceil$ (Muder, Weaver, and West, 1988). They originally conjectured that this bound was optimal, but it has been improved to $\left\lfloor \frac{2n}{3} \right\rfloor + 1$ when $m = n$ and in the case when $m = \left\lfloor \frac{n^2}{4} \right\rfloor$, embeddings in books with $n-1$ pages have been found (Enomoto, Nakamigawa, and Ota, 1997).

The Genetic Algorithm

Having provided an overview of the book embedding problem, we turn our attention to the Genetic Algorithm (GA). The GA is loosely based on the concept of Darwinian natural selection. Individual members of a species who are better adapted to a given environment reproduce more successfully and so pass their adaptations on to their offspring. Over time, individuals possessing the adaptation form interbreeding populations, that is, a new species. In keeping with the biological metaphor, a candidate solution in a GA is known as a *chromosome*. The chromosome is composed of multiple genes. A collection of chromosomes is called a *population*. The GA randomly generates an initial population of chromosomes that are then ranked according to a fitness (or cost) function (Haupt and Haupt, 1998). One of the truly marvelous things about the GA is its wide applicability. We have used it to optimize structural engineering components—an NP-Complete problem—and are currently applying it to model language change (Ganzerli, et al., 2003, 2005, 2008; Overbay, et al., 2006). For practical purposes, this means, of course, that those who attempt to solve these problems must be content with good-enough solutions. Though good-enough may not appeal to purists, it is exactly the kind of solution implicit in natural selection: a local adaptation to local constraints, where the structures undergoing change are themselves the product of a recursive sequence of adaptations. This can be expressed quite compactly:

```
GA()
{
    Initialize() //generate population
    ComputeCost() //of each member
    SortByCost() //entire population
    while (good enough solution has not appeared)
     {
        Winnow() //who reproduces?
        Pair() //pair up reproducers
        Mate() //pass properties to children
        Mutate()  //randomly perturb genes
        SortByCost() //entire population
        TestConvergence()  //good enough solution?
     }
}
```

As noted, the optimal book embedding for the complete bipartite graph $K_{m,n}$ is not known. The optimal book-thickness is known for small values of m and n, but even in cases as small as $K_{4,4}$ an unusual ordering of the vertices is needed to attain an optimal 3-page embedding. Using a dual-layered approach to our genetic algorithm, described later in the paper, we hope to improve upon the best known bounds.

The GA and Book Embedding

The most extensive application of the GA to the book embedding problem prior to our own work is found in Kapoor et al. (2002) and Kapoor (1999). Kapoor at al. (2002) algorithmically generate an edge ordering and use the GA solely for the vertex ordering. Their algorithm produced embeddings at the known optimal bound for certain families of graphs. They provide no examples on how their approach scales to other types of graphs. Further, Kapoor (1999) appears only to have found known optimums for relatively small graphs, such as the complete graph up to K_{10}. Kapoor's results may be limited since the edge ordering is fixed. It is also known that embedding with pre-set vertex ordering does not always achieve optimal results. Our approach seeks to vary both dimensions of the problem.

The Dual-Layered Approach

We use a novel application of the GA to the book embedding problem that we call the "Dual-Layered Approach" (DUA). DUA provides an outer GA, which is used to seek an "optimal" vertex ordering for the spine of the book, along with an inner GA which seeks the "best" edge ordering for any given vertex sequence. Each population in our experiments consists of 64 chromosomes. The outer GA generates an initial population of vertex orderings, referred to as outer chromosomes. In order to determine the fitness of these chromosomes, the inner GA is run using each individual member of the population as a vertex ordering. So, for each member of a population of 64 outer chromosomes, the inner GA is run 64 times. The fitness of each outer chromosome is equal to the fitness of the best solution found in the inner GA using that vertex sequence. This process is repeated in each generation of the outer GA.

The ultimate goal is to find a solution within the inner GA that is lower than theorized bounds for graphs such as complete bipartite graphs, $K_{m,n}$. DUA will be particularly useful in seeking an improvement on the best known bounds for the book thickness of complete bipartite graphs since it is known that naïve approaches to vertex ordering for this family of graphs does not lead to optimal results. In particular, orderings with high regularity do not lead to the smallest book thickness. We hope that DUA will help discover atypical vertex orderings for complete bipartite graphs that will produce book embeddings that require fewer pages than the best known bounds.

The Cost Function

We have applied optimizations to several aspects of the inner GA in order to improve its effectiveness and efficiency. The cost function received special attention. The fitness of any given solution can be seen as its distance away from the best known bound. The more accurately the cost algorithm is able to capture this distance, the more quickly the GA will converge on a local solution. If the cost algorithm does not capture this distance well, then the GA will approach a random search. Initially we attempted to measure the cost using a relatively naïve approach, that is, the cost was simply equal to the book-thickness for a given edge ordering. However, consider two edge orderings with the same book-thickness: ordering one is more tightly packed toward the first page, while ordering two is more thickly populated toward the end. Ordering one is probably closer to an optimal solution than ordering two, but by considering only the book-thickness, the genetic algorithm would be unable to differentiate between the two solutions.

In order to solve this problem, we developed a cost function that values both small book-thickness as well as books more tightly packed towards the top. This cost function is customized for each type of graph. For example, when evaluating the fitness of a particular book embedding of the complete graph, we remove the m cycle edges from our edge list, since these may be placed on any page. By the proof of Theorem 1, at most $m - 3$ additional edges may be placed on any page of the book. We assign a cost of 0 to any page that achieves this bound. Pages that have $m - 3 - k$ edges are assigned a cost of k. Since an optimal book embedding of K_m requires $w = \left\lceil \frac{m}{2} \right\rceil$ pages (see Theorem 2), any edges embedded on pages after page w are also included in the cost function. The cost function for K_m is given below:

t = total number of edges (not counting adjacent boundary edges)
e = max number of edges per page = m-3
p = max number of pages = ceiling(t/e)
n = number of edges on last page = e-(t mod e)

cases:
1. current page # < p
   cost = e - (number of edges on page)

2. current page number == p
   if current number of edges on page <= n
      cost = n-(current number of edges on page)
   if current number of edges on page is > n
      cost = (current number of edges on page) – n

3. current page number > p
   cost = current number of edges on page

Bipartite graphs, such as the hypercube and $K_{m,n}$ do not contain triangles, so the maximum number of edges per page of the book will be less than $m - 3$. For such graphs, the custom cost function is adjusted accordingly.

Mating Algorithms

We also explored several types of mating algorithms, finally settling on the Order Crossover approach, which is well suited to the book embedding problem due to its ability to maintain the relative order of genes (Davis, 1985). In Order Crossover, the construction of a child chromosome from its parent involves two main steps. First, a subsequence of genes with random start and end indices is selected from parent 1. Each gene in the range [start, end], is copied into the child at the same index as it was found in the parent (Figure 3-Step 1). Next, starting at end + 1, genes are selected from parent 2 and placed into the child at the first available index following the end index. If a selected gene is already contained in the child, then it is skipped. The algorithm continues to choose genes from parent 2 until the child chromosome has been filled, wrapping around to the beginning of the parent and child chromosomes as needed (Figure 3-Step 2).

Step 1:

Parent 1:	1 |	2	3	4 |	5	6	7	8
Parent 2:	3	2	5	6	4	8	1	7
Child:	_ |	2	3	4 |	_	_	_	_

Step 2:

Parent 1:	1 |	2	3	4 |	5	6	7	8
Parent 2:	3_x	2_x	5_4	6_5	4_x	8_1	1_2	7_3
Child:	6_5 |	2	3	4 |	8_1	1_2	7_3	5_4

Figure 3: ***The two main steps of Order Crossover. In Step 2, numbered subscripts indicate the order of insertion into the child, while the subscript "x" indicates a gene which was skipped.***

Normalization

The use of Order Crossover allowed us to optimize our cost function using a technique that we call "normalization." Because the edge ordering is independent of the page numbers of the edges, the effectiveness of the mating algorithm was diluted. Normalization is the process of grouping the edges by their page numbers. In other words, all edges that were embedded on page one occur first, followed by all of the edges from page two, etc. When the edges are grouped in this manner, any sub-sequence of edges that gets swapped by the parents in the mating algorithm contains edges that are closely related by page. Therefore, entire sections of the parent embedding can be preserved in the children. Normalization has enabled us to find optimal book embeddings for several kinds of graphs.

Results

We have explored several kinds of graphs thus far:

- Complete graphs up to K_{19}
- Complete bipartite graphs up to $K_{7,7}$
- Hypercubes up to Q_6
- Square grids up to 10×10
- X-trees up to height 8

Table 1 This shows the best results produced by our GA as they compare to the optimal bound for the book thickness of complete graphs.

Graph	Our Results	Optimal Bound
K_7	4	4
K_8	4	4
K_9	5	5
K_{10}	5	5
K_{11}	6	6
K_{12}	7	6
K_{13}	7	7
K_{19}	10	10
K_n		$\lceil n/2 \rceil$

Table 2 This shows the best results produced by our GA as they compare to the best known lower bound for the book thickness of complete bipartite graphs.

Graph	Our Results	Best Known Bound
$K_{5,5}$	4	4
$K_{6,6}$	5	5
$K_{7,7}$	5	5
$K_{n,n}$		$\lfloor 2n/3 \rfloor + 1$

Table 3 This shows the best results produced by our GA as they compare to the best known lower bound for the book thickness of hypercube graphs.

Graph	Our Results	Best Known Bound
Q_4	4	4
Q_5	4	4
Q_6	5	5
Q_7	7	6
Q_n		$n - 1$

Table 4 This shows the best results produced by our GA as they compare to the optimal bound for the book thickness of square grids.

Graph Size	Our Results	Optimal Bound
2×2	2	2
3×3	2	2
4×4	2	2
5×5	2	2
6×6	2	2
7×7	2	2
8×8	2	2
9×9	2	2
10×10	2	2
$n \times n$		2

Table 5 This shows the best results produced by our GA as they compare to the optimal lower bound for the book thickness of *X*-trees.

Graph Height	Our Results	Optimal Bound
2	2	2
3	2	2
4	2	2
5	2	2
6	2	2
7	2	2
8	2	2
n		2

In every case, with the exception of Q_7, our results have been equivalent to known or conjectured bounds. We also have attained optimal bounds for much larger graphs than in previously published results. Our GA has attained optimal book embeddings of the complete graph up to K_{19}, which has 19 vertices and 171 edges. Clearly, the possible orderings of 171 edges would make an exhaustive search of the solution space intractable.

It should be noted that for square grids and *X*-trees, convergence to an optimal two-page embedding occurred every time and the convergence time did not appear to increase as the size of the graph increased. For these graphs, the degrees of the vertices and the structure of the graph remain similar as the size increases. We would expect duplicate results for much larger graphs of these types. Whereas, for complete graphs, hypercubes, and complete bipartite graphs, the vertex degrees increase as the number of vertices grows. For this reason, these three types of graphs are of interest in our continued research. We are particularly interested in improving on the theoretical bounds for hypercubes and complete bipartite graphs, since the best bounds for these graphs are still unknown.

Conclusion and Future Research

Book embedding is easy to describe yet computationally intractable. It is exactly the kind of problem for which the genetic algorithm shines. Whether one is constructing a near-optimal truss, a near-optimal book embedding, or, indeed, an organism adapted to a set of local conditions, the genetic algorithm has proven to be a useful guide. We have shown that the GA can produce book embeddings that are as good as known optimal bounds on large graphs. Though we have yet to find a counter-example to conjectured bounds for other types of graphs, our dual-layered approach, a genetic algorithm within a genetic algorithm, represents a novel solution to the problem. We are currently working in two directions. We are attempting to generate book embeddings for complete bipartite graphs and hypercubes that improve upon known bounds for these graphs. We also observe that computing the same cost function for each of 64 chromosomes is embarrassingly parallel. Our major effort over the next year will be to parallelize DUA for execution on a cluster. Although the ability to search intractably large spaces will not necessarily generate a true optimal embedding, it should allow us to speak with confidence about currently conjectured bounds.

Acknowledgments

The authors wish to acknowledge the McDonald Family and the Gonzaga University Research Council which have funded this research for several years. Their confidence in the role of undergraduates in faculty research made the participation of our three undergraduate research assistants (Arnold, Hurson, and Pierce) possible.

References

Bernhart, F., and Kainen, P. C. 1979. The Book Thickness of a Graph. *Journal of Combinatorial Theory*, Series B 27 (3): 320-331.

Chung, F. R. K., Leighton, F. T., and Rosenberg, A. L. 1987. Embedding Graphs in Books: A Layout Problem with Applications to VLSI Design. *SIAM Journal on Algebraic and Discrete Methods* 8 (1): 33-58.

Davis, L. 1985. Applying Algorithms to Epistatic Domains. *Proceedings of the International Joint Conference on Artificial Intelligence*, 162-164.

Dujmović, V. and Wood, D. R. 2004. On Linear Layouts of Graphs. *Discrete Mathematics and Theoretical Computer Science* 6: 339-358.

Enomoto, H., Nakamigawa, T., and Ota, K. 1997. On the Pagenumber of Complete Bipartite Graphs. *Journal of Combinatorial Theory*, Series B 71 (1): 111-120.

Ganzerli, S., De Palma, Paul. 2008. Genetic Algorithms and Structural Design Using Convex Models of Uncertainty. In Y. Tsompanakis, N. Lagaros, M. Papadrakakis (eds.), *Structural Design Optimization Considering Uncertainties*. London: A.A. Balkema Publishers, A Member of the Taylor and Francis Group.

Ganzerli, S., De Palma, P., Stackle, P., & Brown, A. 2005. Info-Gap Uncertainty on Structural Optimization via Genetic Algorithms. *Proceedings of the Ninth International Conference on Structural Safety and Reliability*, Rome.

Ganzerli, S., De Palma, P., Smith, J., & Burkhart, M. 2003. Efficiency of genetic algorithms for optimal structural design considering convex modes of uncertainty. *Proceedings of The Ninth International Conference on Applications of Statistics and Probability in Civil Engineering,* San Francisco.

Garey, M. R. and Johnson, D. S. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* NY: W. H. Freeman and Co.

Garey, M. R., Johnson, D. S., Miller, G. L., Papadimitriou, C. H. 1980. The Complexity of Coloring Circular Arcs and Chords. *SIAM Journal on Algebraic and Discrete Methods* 1 (2): 216-227.

Gleiss, P. M. and Stadler, P. F., 1999. Relevant Cycles in Biopolymers and Random Graphs, Technical Report, 99-07-042, Santa Fe Institute, Santa Fe, NM.

Goldberg, D. E. 1989. *Genetic Algorithms in Search Optimization and machine Learning*. NY: Addison-Wesley.

Haupt, L., Haupt, S. 1998. *Practical Genetic Algorithms*. New York: John Wiley and Sons.

Holland, J. H. 1975. *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI: University of Michigan Press.

Kainen, P. C., 1990. The Book Thickness of a Graph, II. *Congressus Numerantium* 71: 127-132.

Kainen, P. C., and Overbay, S. R., 2007. Extension of a Theorem of Whitney. *Applied Mathematics Letters* 20 (7): 835-837.

Luger, G., Stubblefield, W. 2009. Genetic and Evolutionary Computing. In: *AI Algorithms, Data Structures, and Idioms in Prolog, Lisp, and Java*. Addison-Wesley Computing.

Kapoor, N. (1999). *Pagenumber Problem*. Master's thesis, SITE, University of Ottawa.

Kapoor, N., Russell, M., Stojmenovic, I. 2002. A Genetic Algoritym for Finding the Pagenumber of Interconnection Networks. *Journal of Parallel and Distributed Computing* 62: 267-283.

Muder, D. J., Weaver, M. L., and West, D. B. 1988. Pagenumber of Complete Bipartite Graphs. *Journal of Graph Theory* 12 (4): 469-489.

Overbay, S., Ganzerli, S., De Palma, P, Brown, A., & Stackle, P. 2006. Trusses, NP-Completeness, and Genetic Algorithms. *Proceedings of the 17th Analysis and Computation Specialty Conference.* St. Louis.

Yannakakis, M. 1986. Four Pages are Necessary and Sufficient for Planar Graphs. *Proceedings of the 18th Annual ACM Symposium on Theory of Computing*, 104-108.