

Research Article

An Evolutionary Formulation of the Crossing Number Problem

**Che Sheng Gan, Wee Chiat Alan Tan, Voon Chet Koo,
and Kwang Hwai Mak**

*Faculty of Engineering and Technology, Multimedia University,
Jalan Ayer Keroh Lama, 75450 Melaka, Malaysia*

Correspondence should be addressed to Voon Chet Koo, vckoo@mmu.edu.my

Received 24 September 2008; Revised 31 March 2009; Accepted 2 July 2009

Recommended by B. Sagar

A graph drawing algorithm is presented which results in complete graphs having minimum crossings equal to that of Guy's conjecture. It is then generalized and formulated in an evolutionary algorithm (EA) to perform constrained search for the crossing numbers. The main objective of this work is to present a suitable two-dimensional scheme which can greatly reduce the complexity of finding crossing numbers by using computer. Program performance criteria are presented and discussed. It is shown that the EA implementation provides good confirmation of the predicted crossing numbers.

Copyright © 2009 Che Sheng Gan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. Introduction

Let G be a graph. The crossing number of G , denoted by $cr(G)$, is the minimum number of pairwise intersections of its edges when G is drawn in the plane. Undefined terms in this paper can be found in [1].

Problem about the crossing number of a graph is an area of active research in both applied and algorithmic graph theory. Beside being a challenging problem in itself, it also finds applications in areas of network design and circuit layout. The crossing numbers problem was proposed in early 1970s [2, 3]. It has been shown that the problem of determining the crossing number of a given graph is NP-complete [4].

There are some known results or conjectures for some families of graphs including complete graphs, bipartite graphs, and product of cycles. In particular, the following is known as Guy's conjecture [5]:

$$cr(K_p) = \frac{\lfloor p/2 \rfloor \cdot \lfloor (p-1)/2 \rfloor \cdot \lfloor (p-2)/2 \rfloor \cdot \lfloor (p-3)/2 \rfloor}{4}, \quad (1.1)$$

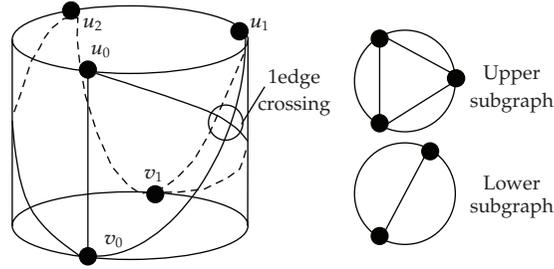


Figure 1: K_5 drawn on a cylinder using the minimum-length rule.

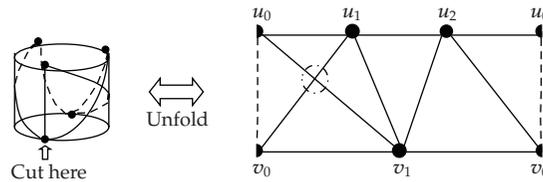


Figure 2: Mapping of K_5 from cylinder to plane.

where K_p is the complete graph of order p . Some values of (1.1) for $p = 5, 6, \dots, 12$ are 1, 3, 9, 18, 36, 60, 100, and 150. Guy's conjecture holds true for complete graphs of up to $p = 12$ and provides a known upper bound for $p > 12$ [6]. Various optimization algorithms [7–9] have been used in the search for the crossing numbers although none has successfully violated the conjecture. In this paper, a graph drawing algorithm is presented which results in complete graphs having the same crossing number given by Guy's conjecture. A natural extension of this algorithm is then used as the underlying scheme in an evolutionary algorithm (EA) that performs a constrained search for the crossing numbers of complete graphs.

2. Graph Drawing Algorithm

In this section, a graph drawing algorithm is proposed, which achieves Guy's conjecture. An article by Archdeacon [10] provided a helpful starting point for the graph drawing algorithm. The key strategy of this drawing method lies in the connection of vertices by weighing possible edge directions by their positive winding number. In other words, the direction of an edge connecting a top vertex to a bottom one is chosen according to the *minimum-length* criterion. It is plausible that such a method could produce a drawing with the minimum number of crossings since an edge of minimum length would seem to have lessened its chance of intersecting with other edges. A complete graph of order 5 drawn using this method is shown in Figure 1. Note also that there are two groups of vertices on the top and bottom surfaces of the cylinder connected amongst each other.

The three-dimensional visualization required by such a method is quite difficult for large p . We simplify the drawing by making a vertical cut on the curved surface of the cylinder along its entire length and “unfolding” it into a rectangular plane as in Figure 2. The resulting vertical boundaries are now represented by vertical dotted lines.

In such drawings, edges corresponding to those on the top and bottom surfaces of the cylinder are not shown and their crossings will be referred to as *external* crossings as opposed

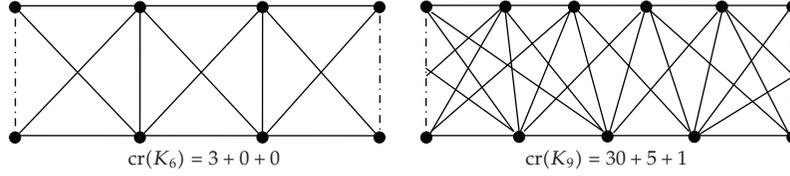


Figure 3: K_6 and K_9 drawn using Algorithm A.

to those on the curved surface of the cylinder which will be referred to as *internal* crossings. The reason for the absence of external crossings is due to the fact that their number can easily be determined by ${}^r C_4$ where r is the number of top or bottom vertices.

It should be noted that this drawing method serves no purpose other than to verify the upper bound according to Guy's conjecture (for $p \leq 12$ it also verifies the actual crossing number). It is deterministic and hence it does not actually search for a drawing that satisfies the minimum number of crossings. Of course, if the conjecture is proven to be true, then the upper bound is equal to the minimum crossing numbers. The corresponding algorithm is presented as follows.

Algorithm A (drawing the complete graph with minimum crossing).

- A1. Partitions the vertices of K_p into U (the set of upper vertices) and V (the set of lower vertices):
 - A1.1. $U = \{u_i\}$ for $0 \leq i \leq m - 1$, where $m = \lfloor p/2 \rfloor$,
 - A1.2. $V = \{v_j\}$ for $0 \leq j \leq n - 1$, where $n = p - m$.
- A2. Place the vertices of U and V on two parallel lines (called the upper and the lower lines) that are separated by unity distance:
 - A2.1. place u_0 (v_0) at $x = 0$ and $x = m \times n$ on the upper (lower) line,
 - A2.2. place the remaining u_i (v_j) between $x = 0$ and $x = m \times n$ at intervals of length $n(m)$ on the upper (lower) line,
 - A2.3. u_0 at $x = 0$ ($x = m \times n$) is joined to v_0 at $x = 0$ ($x = m \times n$),
 - A2.4. u_i is joined to each v_j either by a continuous line or a broken line which crosses the line $x = 0$ or $x = m \times n$ subject to the minimum length or the positive direction criteria (the direction is positive for the right going lines).

Figure 3 shows the application of this drawing algorithm on K_6 and K_9 . The total number of crossings is calculated as the sum of the internal crossings and the upper and lower external crossings.

3. The Evolutionary Algorithm (EA)

3.1. Introduction

In this section, we present an evolutionary formulation for the crossing number problem. The proposed algorithm initializes by randomly generating a population of genomes. As shown in Figure 4, initial populations of size N are selected from the solution space of 2^p , where

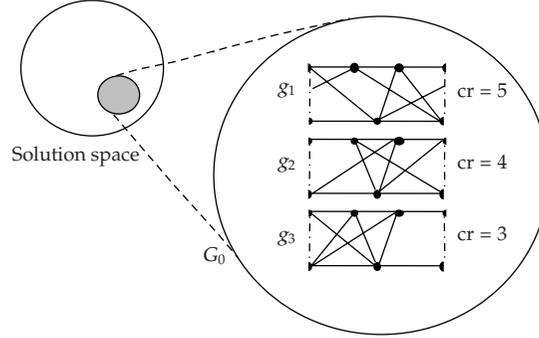


Figure 4: An initial population G_0 containing N genomes is selected from the solution space of 2^p .

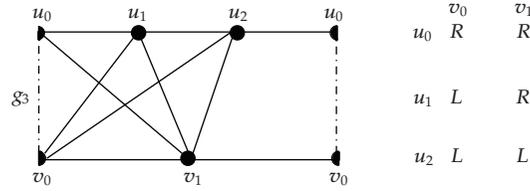


Figure 5: The coding convention for genome $g_3 \in K_5$; coding conventions, R: right-going edge, L: left-going edge, for the case where (u_i, v_i) forms a vertical line, either R or L can be used.

each genome encodes a complete graph of order p by a binary-valued $\lfloor p/2 \rfloor \times \lfloor p/2 \rfloor$ adjacency matrix. Figure 5 illustrates an example of the coding convention for K_5 .

From the initial population, the evolutionary algorithm (Algorithm B) recombines, mutates, and selects genomes for the population of subsequent generations. The algorithm terminates after K generations, or when the optimum solution (the conjecture upper bound) is achieved.

Algorithm B (optimization search for minimum crossing based on EA).

B1 Initialization.

An initial population, G_0 , of N genomes is randomly drawn,
Set $G_0 \leftarrow \{g_1, g_2, \dots, g_N\}$.

B2 For $k = 1$ to K , make the following:

B2.1 Recombination.

two parent genomes g_a and g_b are selected from the population G_{k-1} using *Roulette selection* and recombined to produce an offspring genome g_r . This process is repeated N_r times:

Initialize $G_k^r \leftarrow 0$.

For $n = 1$ to N_r , do:

Pick $g_a \leftarrow \text{roulette}(G_{k-1})$ and $g_b \leftarrow \text{roulette}(G_{k-1} - g_a)$.

Make $g_r \leftarrow \text{recombine}(g_a, g_b)$.

Update $G_k^r \leftarrow G_k^r + g_r$.

End For
 Update $G_k^r \leftarrow G_{k-1} + G_k^r$.

B2.2 Mutation.

Every genome in the population G_k^r is mutated at a (preset) mutation rate P_{mut} :
 Initialize $G_k^m \leftarrow 0$.
 For every $g_n \in G_k^r$, do:
 Make $g_m \leftarrow \text{mutate}(P_{\text{mut}}, g_n)$.
 Update $G_k^m \leftarrow G_k^m + g_m$.
 End For

B2.3 Selection.

The selection stage is divided into two parts, that is, *Elite selection* and *Non-elite selection*. In the elite selection, the best N_s genomes of the population G_k^m is initially selected and removed from G_k^m . The remaining $N - N_s$ genomes are then selected from the depleted G_k^m to retain the population size at N :

Initialize $G_k \leftarrow 0$.
 For $n = 1$ to N_s , do: } *Elite Selection*
 Pick $g_s \leftarrow \text{best of}(G_k^m)$.
 Update $G_k^m \leftarrow G_k^m - g_s$.
 Update $G_k \leftarrow G_k + g_s$.
 End For
 For $n = 1$ to $N - N_s$, do: } *Non-elite Selection*
 Pick $g_s \leftarrow \text{roulette}(G_k^m)$.
 Update $G_k^m \leftarrow G_k^m - g_s$.
 Update $G_k \leftarrow G_k + g_s$.
 End For

End For

3.2. Evolutionary Functions

To evaluate the “goodness” of a genome in a population, a *fitness function* is required. A genome with a high fitness value is more likely to survive than one with a low fitness value. For this particular problem, the following fitness function F is used to evaluate the fitness of graph g :

$$F(g) = \max\left(0, 1 - \log_2\left(\frac{\text{cr}(g)}{\text{cr}_0}\right)\right), \quad (3.1)$$

where $\text{cr}(g)$ is the crossing number of graph g and cr_0 is the conjectured lower bound of $\text{cr}(g)$. As shown in Figure 6, the fitness function is a monotone function which has the widest dynamic range as compared to other functions. If Guy’s conjecture holds true, the

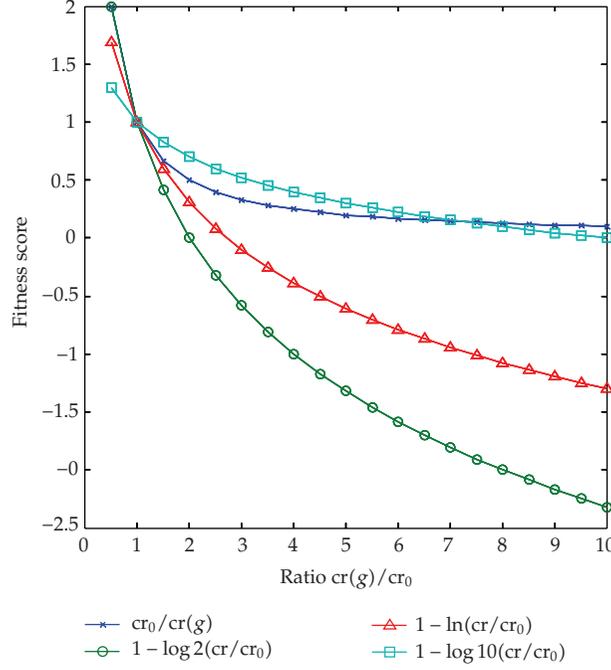


Figure 6: Fitness Functions.

optimum (maximum) value for $F(g)$ is 1. A value greater than 1 will indicate a violation of the conjecture.

Roulette selection is a selection mechanism in which a genome of population G is picked based on some fitness measure. To select a genome from the population $G = \{g_1, g_2, \dots, g_N\}$, a random number r is drawn from the uniform distribution $[0, 1]$. Genome g_n is selected if

$$\frac{\sum_{i=1}^{n-1} F(g_i)}{F_0} \leq r < \frac{\sum_{i=1}^n F(g_i)}{F_0}, \quad (3.2)$$

where

$$F_0 = \sum_{i=1}^N F(g_i). \quad (3.3)$$

The function recombine (g_a, g_b) combines two parent genomes g_a and g_b to produce an offspring genome. The offspring g_r inherits genetic material from g_a with a probability $F(g_a)/F(g_a) + F(g_b)$, otherwise, genetic material from g_b is inherited. The recombination process is illustrated in Figure 7.

The function mutate (P_{mut}, g) mutates genome g with mutation rate P_{mut} . Mutation generally involves toggling the binary value of the coded genome. The function bestof (G) picks the best genome g_b from population G , where

$$b = \arg \max \{F(g_n) : \forall g_n \in G\}. \quad (3.4)$$

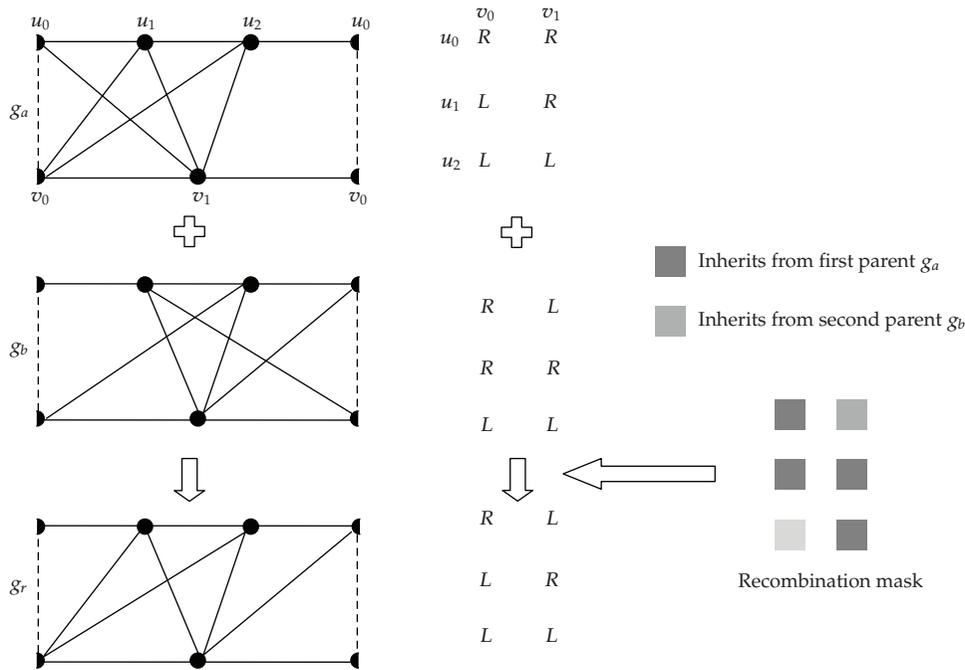


Figure 7: The recombination process.

4. Performance Evaluation

The EA algorithm is deemed successful when it finds the number of crossings equal to (1.1). We will examine the performance of the algorithm based on the number of functions evaluated to achieve the conjectured upper bound. Four variables are considered for such evaluation: graph order (p), mating probability (P_{mate}) defined as N_r/N , mutation rate (P_{mut}), and elite percentage (P_{elite}) defined as N_s/N . In each experiment, three of the four input variables are held constant while the variable selected for evaluation takes on values of a given range. Figures 8, 9, and 10 summarizes the experiment results by varying the P_{mate} , P_{mut} , and P_{elite} , at $p = 12$. Each experiment is conducted for 50 trials and the final result is averaged over these 50 trials. In a subsequent experiment shown in Figure 11, a set of optimum parameters ($P_{mate} = 0.5, P_{mut} = 0.05$, and $P_{elite} = 0.1$) is applied to search for the conjecture upper bound. Finally, Table 1 summarizes the success rates of the proposed formulation for graph order up to 20. The success rates improve as the population size (N) and the number of iterations increase. However, this will increase the computational time since larger solution space is evaluated. Based on the empirical data, $N = 50$ and Iteration = 200 are good compromise for this problem.

5. Conclusions and Future Work

In this paper a graph-drawing algorithm is proposed to represent the graphical equivalent of Guy’s conjecture. The results are then extended, generalized, and employed in an evolutionary algorithm that performed a constrained search for the crossing number.

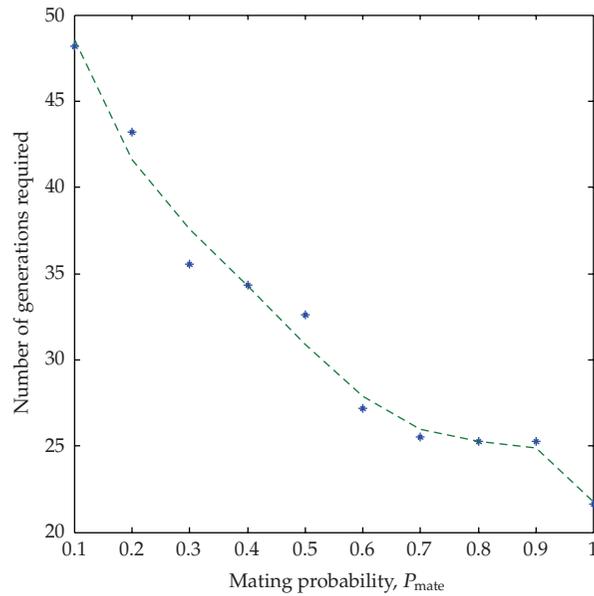


Figure 8: Plot of number of functions evaluated against mating probability ($p = 12$, $P_{\text{mut}} = 0.05$, $P_{\text{elite}} = 0.1$).

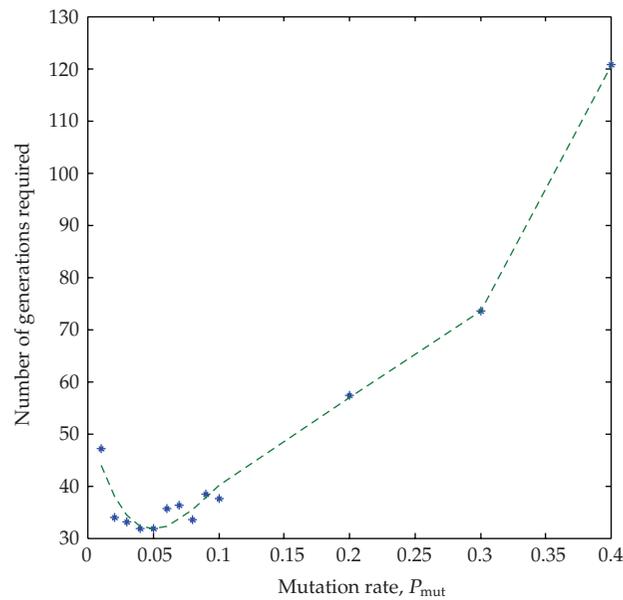


Figure 9: Plot of number of functions evaluated against mutation rate ($p = 12$, $P_{\text{mate}} = 0.5$, $P_{\text{elite}} = 0.1$).

Performance of the EA is evaluated and presented. It is demonstrated that the EA is successful in finding the upper bound as set by (1.1). The success of the EA is dependent on the actual evolutionary mechanisms of selection, recombination, mutation, and reinsertion. As expected, the number of generations required to achieve optimum solution increases as

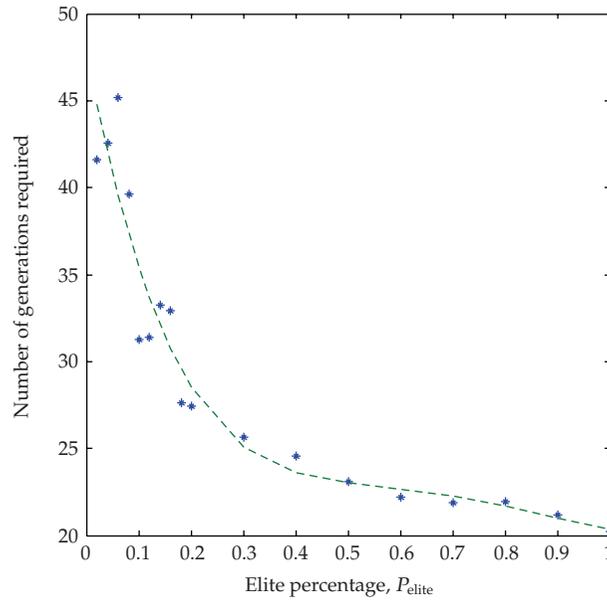


Figure 10: Plot of number of functions evaluated against elite percentage ($p = 12, P_{mate} = 0.5, P_{mut} = 0.05$).

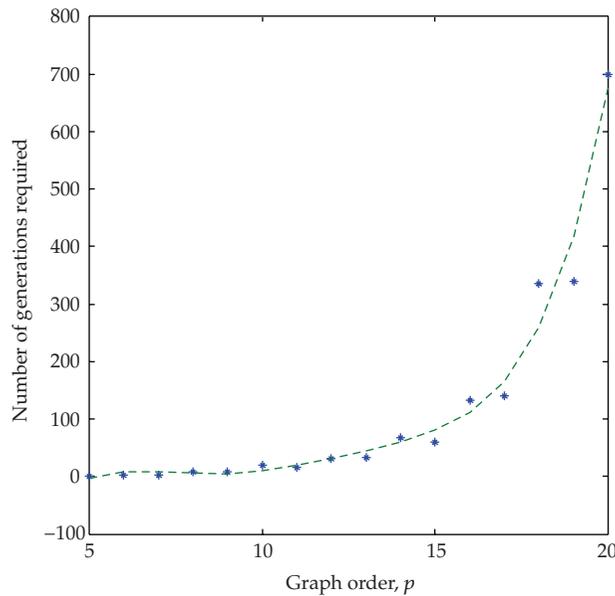


Figure 11: Plot of number of functions evaluated against graph order ($P_{mate} = 0.5, P_{mut} = 0.05, P_{elite} = 0.1$).

the graph order becomes larger. The algorithm performed well at lower rates of mutation, which prevents the EA from degenerating into a random search.

It is natural to ask whether a graph with fewer crossings can be found by removing adherence to the conjecture upper bound. The future work is focused on optimizing various

Table 1: Success Rates of the EA.

Graph order, p	$N = 25,$ Max iteration = 100		$N = 25,$ Max iteration = 200		$N = 50,$ Max iteration = 100		$N = 50,$ Max iteration = 200	
	Average iteration	Success rate (%)						
5	1	100	1	100	1	100	1	100
6	1	100	1	100	1	100	1	100
7	1	100	1	100	1	100	1	100
8	3	100	4	100	2	100	2	100
9	4	100	5	100	3	100	3	100
10	9	100	9	100	6	100	7	100
11	11	100	11	100	8	100	8	100
12	21	100	20	100	15	100	15	100
13	25	100	24	100	18	100	17	100
14	42	98	42	100	30	100	28	100
15	47	100	48	100	33	100	35	100
16	72	84	73	100	50	98	59	100
17	75	66	98	100	58	94	68	100
18	89	16	137	82	77	46	108	96
19	82	2	147	64	79	3	115	94
20	100	0	154	18	89	2	156	64

parameters and generalizing the algorithm such that given an initial set of randomly drawn complete graphs, an optimized solution is to be found with minimum crossing numbers.

References

- [1] G. Chartrand and L. Lesniak, *Graphs & Digraphs*, Chapman & Hall, London, UK, 3rd edition, 1996.
- [2] W. T. Tutte, "Toward a theory of crossing numbers," *Journal of Combinatorial Theory Series B*, vol. 8, pp. 45–53, 1970.
- [3] P. Turán, "A note of welcome," *Journal of Graph Theory*, vol. 1, pp. 7–9, 1977.
- [4] M. R. Garey and D. S. Johnson, "Crossing number is NP-complete," *SIAM Journal on Algebraic and Discrete Methods*, vol. 4, no. 3, pp. 312–316, 1983.
- [5] R. K. Guy, "Crossing numbers of graphs," in *Graph Theory and Applications*, pp. 111–124, Springer, New York, NY, USA, 1972.
- [6] S. Pan and R. B. Richter, "The crossing number of K_{11} is 100," *Journal of Graph Theory*, vol. 56, no. 2, pp. 128–134, 2007.
- [7] F. C. Harris Jr. and R. H. Cynthia, "A proposed algorithm for calculating the minimum crossing number of a graph," in *Proceedings of the 8th Quadrennial International Conference on Graph Theory, Combinatorics, Algorithms, and Applications*, Kalamazoo, Mich, USA, June 1996.
- [8] C. Gutwenger and P. Mutzel, "An experimental study of crossing minimization heuristics," in *Graph Drawing*, G. Liotta, Ed., vol. 2912 of *Lecture Notes in Computer Science*, pp. 13–24, Springer, Berlin, Germany, 2004.
- [9] U. Tadjiev, *Parallel computation and graphical visualization of the minimum crossing number of a graph*, M.S. thesis, University of Nevada, 1998.
- [10] D. Archdeacon, December 2003, <http://www.emba.uvm.edu/~archdeac/problems/crosskn.htm>.