World Academy of Science, Engineering and Technology
International Journal of Electrical and Computer Engineering
Vol:4, No:3, 2010

Complexity Analysis of Some Known Graph Coloring Instances

Jeffrey L. Duffany

***Abstract*—**Graph coloring is an important problem in computer science and many algorithms are known for obtaining reasonably good solutions in polynomial time. One method of comparing different algorithms is to test them on a set of standard graphs where the optimal solution is already known. This investigation analyzes a set of 50 well known graph coloring instances according to a set of complexity measures. These instances come from a variety of sources some representing actual applications of graph coloring (register allocation) and others (mycieleski and leighton graphs) that are theoretically designed to be difficult to solve. The size of the graphs ranged from ranged from a low of 11 variables to a high of 864 variables. The method used to solve the coloring problem was the square of the adjacency (i.e., correlation) matrix. The results show that the most difficult graphs to solve were the leighton and the queen graphs. Complexity measures such as density, mobility, deviation from uniform color class size and number of block diagonal zeros are calculated for each graph. The results showed that the most difficult problems have low mobility (in the range of .2-.5) and relatively little deviation from uniform color class size.

***Keywords*—**graph coloring, complexity, algorithm.

I. Introduction

ASSUME a graph G=(V,E) where V is a set of vertices and E is a set of edges connecting some subset of n(n-1)/2 pairs of vertices. The idea is to assign each vertex a color in such a way that no two adjacent vertices (connected by an edge) have the same color. In addition, the optimal solution uses the minimum number of colors possible (k*). This is the well-known graph coloring problem which is an np-hard problem[1]. A graph G can be represented by an binary symmetric matrix called the adjacency matrix (A) where a 1 in row i and column j represents an edge in the graph between vertex i and j and a 0 representing the absence of an edge in G. A decision function f(A) is a polynomial function in A that allows a graph coloring algorithm to choose a pair of vertices to combine (for example $f(A)=\max(A^2)$)[2,3,4,5,6]. Each (i,j) element in A^2 represents the number of shared constraints between vertices i and j. Combining vertices that share the most constraints often leads to an optimal solution. The process is repeated until all vertices are combined.

J. L. Duffany is with the Electrical and Computer Engineering Department, Universidad del Turabo, Gurabo, PR 00778 USA, (e-mail: jduffany@suagm.edu).

A solution vector (s) is a mapping of each vertex x_i into an integer s_i such that $1 <= s_i <= k$ while ensuring that $s_i \neq s_j$ when $A[i,j] = 1$. If an optimal solution vector s* is permuted such that all equivalence classes are grouped together and the corresponding A matrix is permuted accordingly, it is seen that the A matrix takes on a block diagonal form. A zero inside the block diagonal represents a good decision which leads towards an optimal solution. A zero outside the block diagonal or any zero of a suboptimal solution may be good or bad and may or may not lead to an optimal solution. A typical problem of n=100 variables has n^2 = 10,000 elements each of which is a zero or a one. For example a system may have 5000 zeros of which 3000 are inside the block diagonal (z_{in}) and 2000 which are outside (z_{out}). Each zero represents a possible decision and at each step of the solution only one of these can be chosen. Clearly there can be a large number of choices and some kind of decision function is required to proceed towards a solution. A pair association {ij} occurs when a pair of variables x_i and x_j are combined into the same equivalence class. Generally speaking the lower the solution cardinality (k) the greater the number of zeros in the blocks of the block diagonal form.

The coloring instances to be analyzed were obtained from an archival website at Carnegie-Mellon University that is dedicated to this purpose and has compiled these graphs from a variety of sources such as the Stanford GraphBase[7]. This site includes a variety of graphs such as book graphs, game graphs, miles graphs and queen graphs[7] all represented in DIMACS (Discrete Mathematics and theoretical Computer Science) format. The selection process started out with all 79 graphs currently on the website and eliminated those were either (1) not in valid DIMACS format or (2) were not a valid graph or (3) did not have a known optimal coloring solution. This process eliminated a total of 29 of the graphs. The remaining 50 graphs were analyzed using the gcolor package of the R Language[8]. The gcolor (graph color) package of R Language provides tools for importing both ASCII and compressed binary DIMACS files into a matrix data structure in the R computing environment. The R Language gcolor package also provides the coloring algorithm (ineq) which implements the $f(A)=\max(A^2)$ (correlation matrix) coloring algorithm. The results are shown in Tables I and II.

TABLE I RESULTS OF DIMACS GRAPH COLORING ANALYSIS - PART 1

GRAPH	SIZE(n)	k*OPTIMAL/kfound	density/mobility	zozi/zratio
anna	138	11/11	.05/1.54	4.01/2.08
david	87	11/11	.11/1.18	5.10/1.61
homer	561	13/13	.01/2.24	4.16/2.49
huck	74	11/11	.11/1.35	6.32/1.34
jean	80	10/10	.08/1.57	3.66/1.98
fpsol21	496	65/65	.09/1.38	3.64/12.69
fpsol22	451	30/30	.09/.78	7.01/3.43
fpsol23	425	30/30	.10/.73	8.32/2.91
games120	120	9/9	.09/.85	6.39/1.11
inithx1	864	54/54	.05/1.25	3.41/11.63
inithx2	645	31/31	.07/.72	8.01/3.21
inithx3	621	31/31	.07/.69	8.35/3.07
mulsol1	197	49/49	.20/1.23	6.37/5.30
mulsol2	188	31/31	.22/.75	10.96/2.02
mulsol3	184	31/31	.23/.73	11.08/1.97
mulsol4	185	31/31	.23/.73	11.08/1.97
mulsol5	186	31/31	.23/.73	8.38/2.55
myciel3	11	4/4	.33/1.10	1.08/1.29
myciel4	23	5/5	.27/.81	1.87/1.28
myciel5	47	6/6	.21/.60	2.27/1.44
myciel6	95	7/7	.17/.44	3.10/1.42
myciel7	191	8/8	.13/.32	3.40/1.58
miles250	128	8/8	.05/1.32	4.94/1.28
miles500	128	20/20	.14/1.09	12.13/1.31
miles750	128	31/31	.26/.94	17.09/1.27
miles1000	128	42/42	.39/.84	18.90/1.28
miles1500	128	73/73	.63/.90	20.70/1.23
zeroin1	211	49/49	.18/1.26	4.82/6.86
zeroin2	211	30/30	.16/.89	3.38/5.76
zeroin3	206	30/30	.17/.87	3.65/5.37

II. RESULTS

The total of 50 graphs selected for analysis was further subdivided into two groups: those which had a 100% success rate (Table I) and those which had less than 100% success rate (Table II). For each instance the size of the graph (number of variables (n)), the known optimal coloring solution (k*), the actual coloring solution found by the algorithm (k), the constraint density (d), the average vertex mobility (k/(d*n)), the deviation from uniform color class size (zratio=zin/zmin) and the ratio of the number of zeros outside the block diagonal to inside the block diagonal (zozi=zout/zin) were tabulated. The constraint density (d) is the number of ones in the A matrix divided by the maximum possible number (n)(n-1)/2.

The vertex mobility (k/(d*n)) is the optimal number of color classes divided by the average vertex degree (n*d) and reflects the ability of a vertex to move between color classes (a measure of how constrained the system is).

The parameter z_{in} is the number of zeros in the block diagonal of an optimal solution whereas z_{out} is the number of zeros not in the block diagonal of an optimal solution and z_{min} is the minimum number of zeros in the block diagonal for a given k (as close to a uniform color class size as possible). A total of 30 out of the 50 graphs successfully solved using the $f(A)=\max(A^2)$ algorithm are found in Table I (including inithx1 which has 864 vertices).

TABLE II RESULTS OF DIMACS GRAPH COLORING - PART 2

GRAPH	SIZE(n)	k*OPTIMAL/kfound	density/mobility	zozi/zratio
le450_5a	450	5/7	.06/.28	4.64/1.17
le450_5b	450	5/5*	.06/.20	3.72/1.00
le450_5c	450	5/7	.10/.16	4.73/1.10
le450_5d	450	5/5*	.10/.12	3.52/1.00
le450_15a	450	15/17	.08/.47	13.27/1.10
le450_15b	450	15/16	.08/.44	13.27/1.03
le450_15c	450	15/24	.16/.32	17.78/1.07
le450_15d	450	15/23	.17/.31	17.93/1.01
le450_25a	450	25/25*	.08/.68	17.56/1.24
le450_25b	450	25/25*	.08/.68	20.47/1.07
le450_25c	450	25/29	.17/.38	21.77/1.06
le450_25d	450	25/29	.17/.37	21.86/1.05
queen5	25	5/5*	.51/.39	1.44/1.00
queen6	36	7/8	.45/.50	3.07/1.09
queen7	49	7/10	.40/.51	4.51/1.10
queen8	64	9/11	.36/.48	5.77/1.05
queen8x12	96	12/13	.30/.46	7.95/1.02
queen9	81	10/11	.32/.42	6.12/1.05
queen11	121	11/14	.27/.43	8.88/1.03
queen13	169	13/16	.23/.41	10.90/1.03

The $f(A)=\max(A^2)$ algorithm showed a 100% success rate on all graphs except for the set of Leighton graphs(4/12=33%) and the set of queen graphs(1/8=12.5%). This raises several important questions. First of all, what is inherently different about the graphs in Table 2 that makes them more difficult to solve? Clearly it is not simply the size of the problem (note queen6 with only 36 variables) although that might have something to do with it. Secondly, it might be asked (for example), why is it possible that a graph such as le450_5a cannot be solved by a particular algorithm while le450_5b (which is the same size and perhaps differing by a few edges) can be solved?

For insight into these questions it is helpful to go back to the $f(A)=\max(A^2)$ algorithm and see what that tells us about the problems it can solve. The $f(A)=\max(A^2)$ algorithm solves all complete k-partite graphs and all graphs reasonably close to complete k-partite (derived by removing some subset of edges). These are highly overconstrained systems with unique optimal solution (vertex mobility is zero). If enough edges are removed from a complete k-partite system it will move from being overconstrained to perfectly constrained and finally to an underconstrained system which is easily solved because there are so many optimal solutions. Somewhere in between there has to be a maximum difficulty which is somewhere in the region of a perfectly constrained system. In terms of vertex mobility it is expected that this would be near one (at vertex mobility = 1 the average vertex degree equals the optimal number of color classes).

Another important complexity metric is the difference between k* (optimal) and k (found). By this metric the most difficult istance would be le450_25c which had a difference kfound-k*optimal = 24-15 = 9. This is apparently a result of a cascading effect of making a poor decision near the start of the solution process. The equivalence class subset algorithm[6] has been developed to overcome this limitation of the $f(A)=\max(A^2)$ decision function. It is based on the fact that every problem that cannot be solved using the $f(A)=\max(A^2)$ decision function is near (in some sense) to one that can be solved by that decision function.

It should also be noted that for all complete k-partite graphs the zozi ratio is zero since there are no zeros outside the block diagonal. Any graph with zozi ratio of zero is trivially solvable. This implies that the most difficult systems to solve would be expected therefore have a high zozi ratio. A high zozi ratio can be achieved by either maximizing zo or minimizing zin. Maximizing zo has already been considered so the only other method is to reduce zin which brings us to the concept of the uniform color class size (which minimizes zin which is the number of zeros in the block diagonal). Notice that in Table 2 the zratio is very close to 1 while in Table 1 it is significantly greater than 1. zratio is the ratio of zin to zmin which is a measure of deviation from uniform color class size. zratio along with the vertex mobility are extremely important measures of complexity for graph coloring problems as will be seen in Figures 3 and 4.

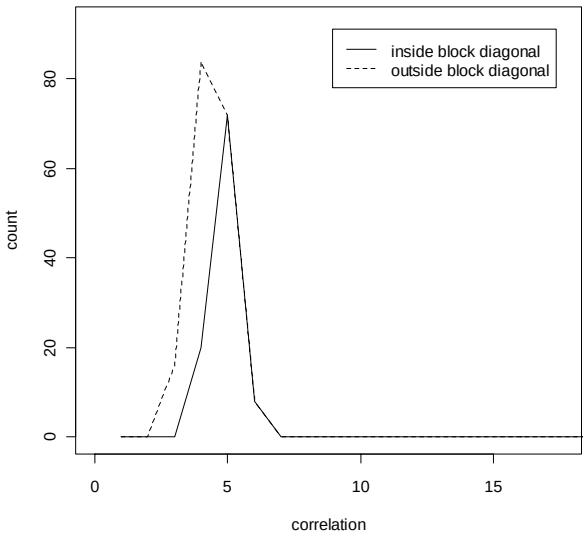

Fig. 1 A^2 values for the queen5 matrix

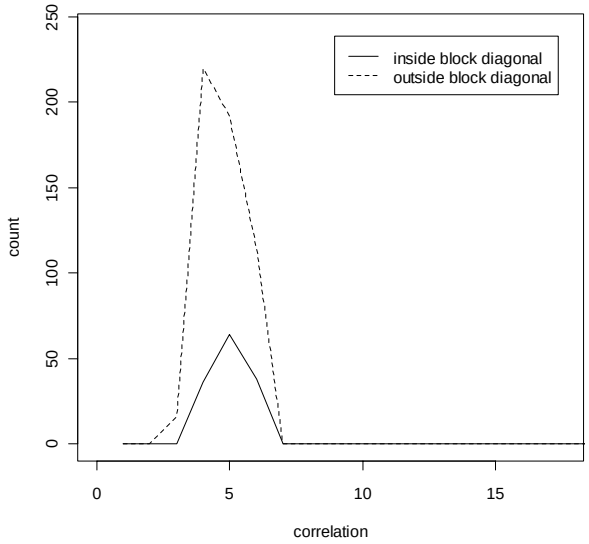

Fig. *2* A^2 values for the queen*6* matrix

III. DISCUSSION

Considerable discussion has been put forth on the ones density of the A matrix and the number of zeros inside and outside of the block diagonal. From a probabilistic viewpoint the odds of making a good decision depends on the ratio of good to bad which is related to the number of zeros inside and outside of the block diagonal. The decision function works most of the time but it most error prone for moderately low density systems. By contrast, in a complete k-partite system (the opposite extreme) there are only good decisions. For any given system and optimal solution vector s* there will be a zozi ratio given by equation (1):

$$zozi = z_{out}/z_{in} \quad (1)$$

The higher the zozi ratio the more difficult the problem. Recall for example that the easiest problems all have a zozi ratio of zozi=0. Difficulty can be defined as the average time complexity of solving a problem or as the difficulty of guessing at random and getting the correct answer. This leads to the question of how it is possible to construct problems which have the highest ratio of zeros outside to zeros inside the block diagonal. Simply decreasing the ones density does not necessarily accomplish the desired result as it can result in the trivial problem where the ones density is zero. Referring to equation (1) it can be seen that to maximize the zozi ratio it would be necessary to consider both maximizing the numerator and minimizing the denominator of the fraction. The formula for calculating the minimum number zeros in the block diagonal is given by equation (2):

$$z_{min} = k*(n/k)^2 = n^2/k \quad (2)$$

which gives approximately the minimum number of block diagonal zeros for solution cardinality k (exactly if n is divisible by k). This tells us that the minimum number of zeros on the block diagonal occurs when the size of all the equivalence classes are the same (n/k) which is the case of uniform color class size.

To illustrate these concepts further consider Figure 1 which shows the result of calculating the correlation decision function $f(A)=\max(A^2)$ for the queen5 graph. The solid curve shows the values of the A^2 matrix corresponding to vertices in the same color class of an optimal solution. On the other hand the dashed curve shows the values of the A^2 matrix corresponding to vertices that are not in the same color class of an optimal solution. It is seen by this that the decision function can reliably distinguish (at least in this case) between vertices in the same color class of an optimal solution and those that are not. This is typical of an easy problem. However if the same calculation is performed on the queen6 graph the distinction is not so clear. In fact, it is virtually impossible for $f(A)=\max(A^2)$ to tell the difference between the categories because they look the same. This is characteristic of a difficult problem.

There is another important observation concerning Figures 1 and 2 and it involves the zozi ratio. In Figures 1 and 2 it can be seen that zo is the area under the dashed line curve while zi is the area under the solid curve. In Figure 1 the zozi ratio is near 1 (1.44) but for Figure 2 it is significantly greater (3.07) which is another indication of increased problem difficulty. Looking at the zozi ratio in Table II shows that there is not a single easy problem among them based on the zozi metric (queen5 was actually the easiest of all).

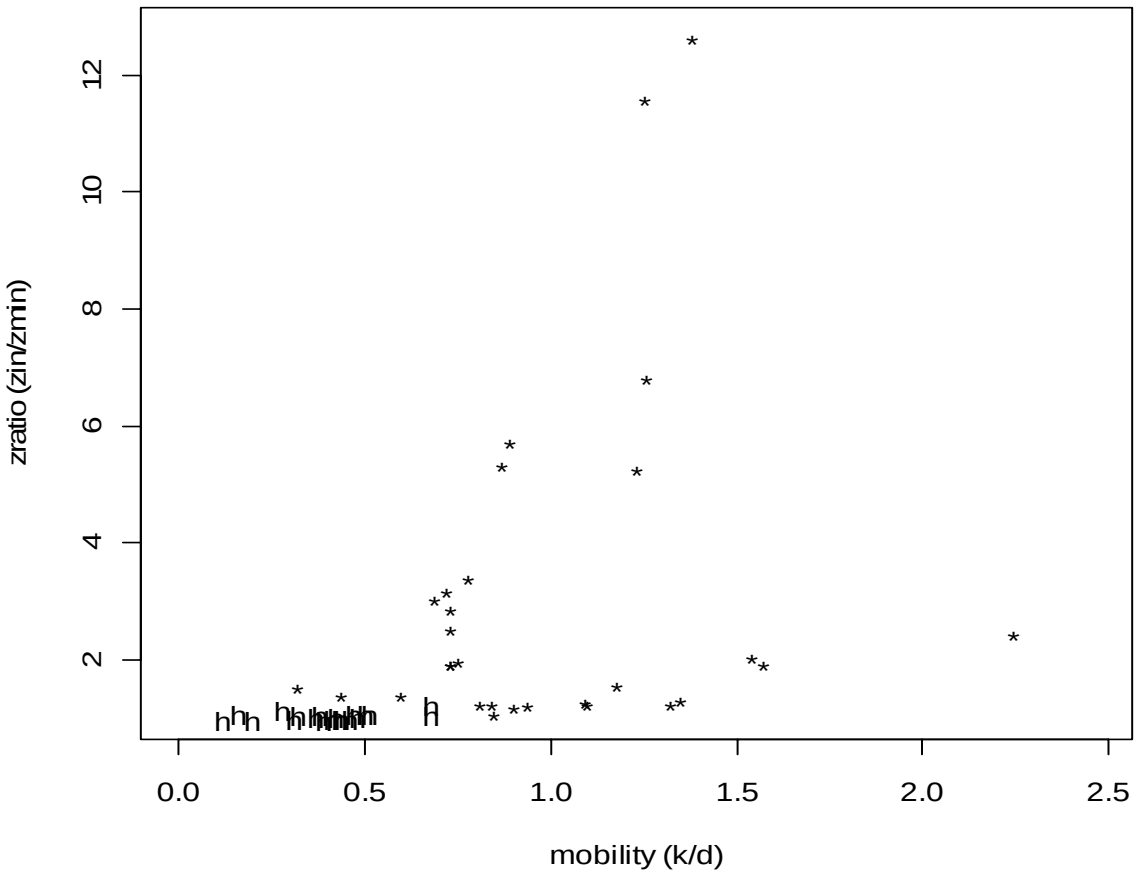

Fig. 3 Plot of zratio versus vertex mobility

Figures 3 and 4 show the plots of some key complexity measures across all 50 graphs in the study. The points marked with an asterisk are from Table I (easy) and those marked with with an h (hard) are from Table II.

It has been noted that that one of the main measures of problem complexity is the ratio of the number of zeros inside the block diagonal z_{in} to those outside the block diagonal z_{out} which can also be called the zozi ratio. It was further determined that the problems with the smallest number of zeros in the block diagonal would also be among the most difficult and this was corroborated by experimental results.

Figure 3 shows two complexity measures (vertex mobility and zratio) and how they differ between Tables I and II. Figure 3 shows very clearly the most difficult Table II (h) problems clustered in the lower left hand corner of the graph. All of the hard problems from Table II had very low deviation from uniform color class size. On the average the hard problems had about a 6% deviation from uniform color class size while the easy problems from Table I averaged about a 200% deviation (this effect is predictable from the nature of the decision function $f(A)=\max(A^2)$).

The other important factor is the mobility which was very low (<1) for all of the graphs in Table II. Also notice the wide range of mobility values for the Table I graphs as compared to the narrow range of the Table II graphs (this effect is predictable from the nature of the decision function $f(A)=\max(A^2)$). It is clearly seen in Figure 3 for example that the most difficult (h) problems had a very narrow range of vertex mobility between about .2-.5. This is exactly in accordance with the results in[2] which analyzed the complexity of randomly generated graphs of size n=100 and solution caridinality k=3. The only difference was that in[2] the range of mobility was shifted slightly higher such that the most difficult problems were in the vertex mobility range of .375-.75. However it does help to provide some confidence that randomly produced graphs can give a reasonable approximation to what might be considered the general case.

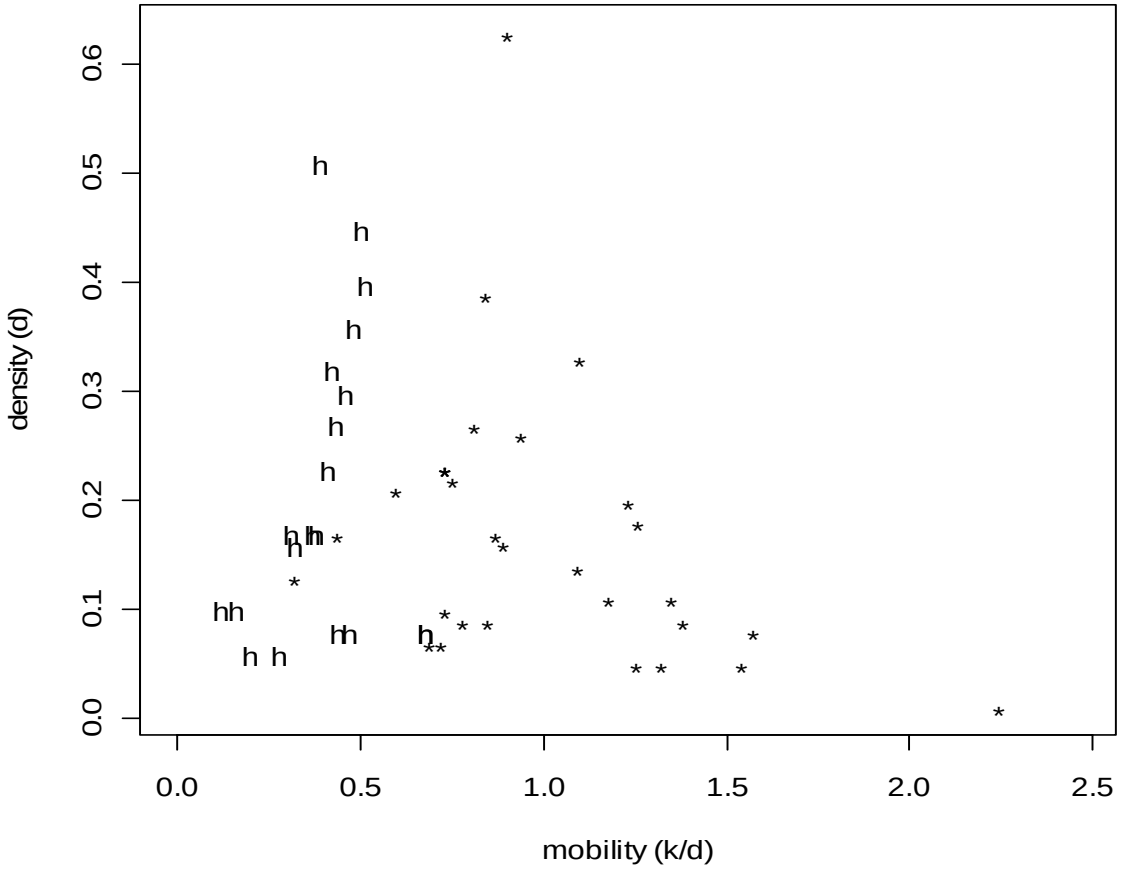

Fig. 4 Plot of Graph Constraint Density versus Vertex Mobility

Figure 4 illustrates the density versus the mobility and shows how the hard problems of Table II are spread out over a wide range of constraint density values. In addition Figure 4 does give an indication that while hard problems can be found over a wide range of constraint density values they are more likely to occur at lower constraint densities (as would be predicted). As seen in Tables I and II the values of both density and zozi vary widely and therefore by themselves are not clear indicators of problem difficulty. Hard problems can occur over a wide range of zozi values however they are more likely to occur at the higher zozi values. The best predictor od difficulty appears to be a combination of low mobility, high zozi and low zratio (e.g., comparing le450_25a,b,c,d).

A summary of the results in Tables I and II is given in Table III by type of graph. Out of the 50 graphs in the study 35/50 or 70% were solved by the decision function $f(A)=\max(A^2)$. All of the remaining queens problems in Table II were subsequently solved using a more sophisticated version of $f(A)=\max(A^2)$ called the equivalence class subset algorithm (ecsa) which is described in[6] although it took many iterations to do so. If these were added to the graphs successfully solved by $f(A)=\max(A^2)$ it would raise the overall success rate to 84%. A complete discussion of the application of ecsa to the queen graphs and the unsolved Leighton graphs of Table II have been deferred to a future study.

TABLE III SUMMARY OF RESULTS FOR ALL 50 DIMACS COLORING INSTANCES

CATEGORY	SIZE (n)	k OPTIMAL	SUCCESS	PERCENT
literature	74-561	11-13	5/5	100%
fpsol	425-496	30-65	3/3	100%
games	120	9	1/1	100%
inithx	621-864	31-54	3/3	100%
mulsol	184-197	31-49	5/5	100%
myciel	11-191	4-8	5/5	100%
miles	128	8-73	5/5	100%
zeroin	206-211	30-49	3/3	100%
leighton	450	5-25	4/12	33%
queen	25-169	5-11	1/8	12%

IV. SUMMARY AND CONCLUSIONS

This investigation analyzes a set of 50 well known graph coloring instances according to a set of complexity measures. These instances come from a variety of sources some representing actual applications of graph coloring (register allocation) and others (mycieleski and leighton graphs) that are theoretically designed to be difficult to solve. The size of the graphs ranged from ranged from a low of 11 variables to a high of 864 variables. The primary method used to solve the coloring problem was the square of the adjacency (i.e., correlation or A^2) matrix.

Complexity measures such as density, mobility, deviation from uniform color class size and number of block diagonal zeros were calculated for each graph. The results show that the most difficult to solve were the leighton graphs and the queen graphs. The results showed that the most difficult problems have low mobility (in the range of .2-.5), relatively little deviation from uniform color class size (zratio close to 1) and high zozi ratio. Based on percentages the queen graphs were the most difficult, despite their relatively small size. A discussion of the unsolved queen graphs and the unsolved Leighton graphs of Table II is deferred to a future study.

The results in this investigation were obtained using the gcolor (graph color) package in the R Language programming environment[8]. The gcolor (graph color) package of R Language provides tools for importing both ASCII and compressed binary DIMACS files into a matrix data structure in the R environment. The R Language gcolor package also provides the coloring algorithm (ineq) which implements the $f(A)=\max(A^2)$ decision function coloring algorithm.

REFERENCES

[1] C.H. Papadimitrou and K. Steiglitz, "Combinatorial Optimization: Algorithms and Complexity", Dover, ISBN 0-486-40258-4, pp. 344.
[2] Duffany, J.L., "Statistical Characterization of NP-Complete Problems", Foundations of Computer Science Conference, World Computer Congress, Las Vegas, Nevada, July 14-17, 2008.
[3] Duffany, J.L. "Systems of Inequations", 4th LACCEI Conference, Mayaguez, PR, June 21-23, 2006.
[4] Duffany, J.L. "Generalized Decision Function and Gradient Search Technique for NP-Complete Problems", XXXII CLEI Conference, Santiago Chile, August 20-23, 2006.
[5] Duffany, J.L., "Optimal Solution of Constraint Satisfaction Problems", International Conference on Applied Computer Science, Sharm el Sheik, Egypt, January, 2009.
[6] Duffany, J.L., "Equivalence Class Subset Algorithm", International Conference Computer Information Technology, Tokyo, Japan, May, 2009.
[7] http://mat.gsia.cmu.edu/COLOR/instances.html
[8] http://www.r-project.org

Jeffrey L. Duffany, Ph.D (M'77) became a Member (M) of **IEEE** in 1977. Dr. Duffany was born in Waterbury, CT and received the BSEE degree from the University of Connecticut in 1977. Dr. Duffany received the MSEE degree from Columbia University in 1979 and a dual Ph.D. in Computer and Information Engineering from Stevens Institute of Technology in 1996.

He joined the Bell Laboratories as a member of technical staff in 1977 where he worked in research and development publishing over 100 internal technical reports and receiving two US patents. He also spent a year working at Lucent Espana in Madrid, Spain. He joined the Universidad del Turabo in Gurabo PR in 2003. In 2005 he was a visiting scientist at Sandia National Laboratories Center for Cyber Defense in Albuquerque, NM. In 2006 he was a visiting faculty researcher at the University of Southern California Information Sciences Institute in Marina del Rey California. In 2008 and in 2009 he was named Office of Naval Research Faculty fellow and worked at the SPAWAR (Space and Naval Warfare Systems Center) in San Diego California. Currently he is teaching graduate and undergraduate classes in computer science, electrical engineering and computer engineering.

Currently Dr. Duffany is pursuing research interests in the areas of computer algorithms, artificial intelligence and network/computer security. Dr. Duffany is a member of the IEEE and the ACM.