Chapter 1

APPROXIMATING MAXIMUM STABLE SET AND MINIMUM GRAPH COLORING PROBLEMS WITH THE POSITIVE SEMIDEFINITE RELAXATION

S. J. Benson
Division of Mathematics and Computer Science
Argonne National Laboratory
Argonne, IL 60439
benson@mcs.anl.gov

Y. Ye
Department of Management Sciences
The University of Iowa
Iowa City, Iowa 52242
yyye@yuan.biz.uiowa.edu

Abstract We compute approximate solutions to the maximum stable set problem and the minimum graph coloring problem using a positive semidefinite relaxation. The positive semidefinite programs are solved using an implementation of the dual scaling algorithm that takes advantage of the sparsity inherent in most graphs and the structure inherent in the problem formulation. From the solution to the relaxation, we apply a randomized algorithm to find approximate maximum stable sets and a modification of a popular heuristic to find graph colorings. We obtained high quality answers for graphs with over 1000 vertices and over 6000 edges.

Keywords: Stable Set, Independent Set, Maximum Clique, Graph Coloring, Positive Semidefinite Relaxation.

1 INTRODUCTION

Given an undirected graph $G = (V, E)$, a *stable set* of vertices (or *vertex packing* or *independent set*) is a subset of V such that no two vertices are adjacent. The *Maximum Stable Set Problem* (MSS) asks for the stable set with the maximum cardinality. A *clique* of graph G is a subset set of vertices such that every pair of vertices is adjacent. A *vertex cover* is a subset of vertices that are incident to each edge in the graph. Denoting $\bar{G}$ as the graph complement of G, the following statements concerning any $S \subset V$ are known to be equivalent:

1. S is a stable set of G,
2. S is a clique of $\bar{G}$,
3. $V \setminus S$ is vertex cover of G

Accordingly, the problems of finding a maximum stable set of G, a maximum clique in $\bar{G}$, and a minimum vertex cover in G are equivalent.

A *vertex coloring* of a graph is an assignment of colors to the vertices V such that no two adjacent vertices receive the same color. Equivalently, the problem looks to partition the vertices into independent sets. The smallest number of colors needed for this coloring is called the *chromatic number* of G. A graph is *k-colorable* if it can be colored with k colors or less. Obviously, the cardinality of any clique in G is a lower bound on the chromatic number of G. When a graph, and every node induced subgraph, have a chromatic number that equals the cardinality of the largest clique, it is known as a *perfect graph*. For this special class of graphs, the MSS problem can be solved to optimality using a polynomial algorithm.

These problems are classical problems in combinatorial optimization and are well known to be NP-complete[19]. The MSS problem can be solved using polynomial time algorithms for special classes of graphs such as perfect graphs and t-perfect graphs, circle graphs and their complements, circular arc graphs and their complements, claw-free graphs, and graphs with long odd cycles[27], but the existence of a polynomial time algorithm for arbitrary graphs seems unlikely.

Various exact solution methods have been developed for these combinatorial optimization problems. An implicit enumeration technique of Carrahan and Pardalos[12], integer programming with branch and bound by Babel and Tinhofer[3][4], Balas, Xue, and Yu[6][7], Mannino and Sassano [27], and Nemhauser[30], integer programming with cutting planes by Balas [5], Nemhauser[31], and Nemhauser and Sigismondi [30], and a tabu search by Friden[17] have all been applied to the maximum stable

set problem. There effectiveness, however, has usually been limited to graphs with less than 500 vertices. For the minimum graph coloring problem, implicit enumeration and branch and bound based methods of Kubale[25] [26] have been limited to very small instances, and a column generation approach based upon the stable set formulation by Mehrotra and Trick[28] has been applied to graphs with up to 600 vertices. Of course all of these algorithms have exponential complexity, so for larger graphs, the only option available is heuristic methods [34] [21] [22] [29], which have the cost of regularly suboptimal solutions.

Aside from its theoretical interest, the MSS problem arises in applications in information retrieval, experimental design, signal transmission, and computer vision[7]. Graph coloring arises when using finite differences to approximate sparse Hessian matrices, and well as applications in computer register allocation[11][14][13], timetable scheduling[9][15][43], and electronic bandwidth allocation[18]. In many of these applications, it suffices to find an approximately optimal solution. This fact and the difficulty of finding exact solutions, have encouraged considerable effort on finding good approximation algorithms.

2 POSITIVE SEMIDEFINITE RELAXATIONS

The standard form of a positive semidefinite program is:

$$\text{(SDP)} \qquad \begin{aligned} &\text{Minimize} && C \bullet X \\ &\text{Subject to} && A_i \bullet X = b_i, \quad i = 1, \ldots, m, \\ & && X \in K \end{aligned}$$

where $K = K_1 \oplus K_2 \oplus \cdots \oplus K_r$ and K_l is the cone of $n_l \times n_l$ symmetric positive semidefinite matrices, $C, A_i \in \Re^{n \times n}$ are symmetric, and $A \bullet C = \text{tr}\ (A^T C)$.

The dual of (SDP) can be written as:

$$\text{(DSP)} \qquad \begin{aligned} &\text{Maximize} && b^T y \\ &\text{Subject to} && \sum_{i=1}^{m} y_i A_i + S = C, \quad S \in K, \end{aligned}$$

where $y \in \Re^m$.

There are some very strong connections between positive semidefinite programming and combinatorial optimization. The famous Lovász number, which provides an upper bound to the maximum stable set of

a graph and a lower bound to its chromatic number, is the solution to a positive semidefinite program. Many more combinatorial problems have can be relaxed into a positive semidefinite program, and some of these relaxations offer rounding techniques that are guaranteed to be within a specified fraction of optimality.

Most linear programming relaxations do not offer a performance guarantee, but Geomans and Williamson[20], in a now classic result, applied the solution of a maximum cut positive semidefinite relaxation to a randomized algorithm and proved that the answers it generates have an expectation greater than 0.878 of optimality. Although the stable set problem cannot be approximated within a constant fraction in polynomial time unless $P = NP$, provably good approximation algorithms using a positive semidefinite relaxation have been found for MAX-SAT, MAX-2-SAT, MAX-3-SAT, MAX-4-SAT, MAX k-CUT[36], MAX-3-CSP, minimum bandwidth, graph bisection, bound constrained quadratic programming[32][45], graph coloring[23], and some scheduling problems.

Much like the formulation of Kleinberg and Goemans[24] the SDP relaxation of the MSS problem will assign each vertex an integer value of -1 or $+1$. One of the two sets will be a stable set. Given a graph G with $n-1$ vertices, our formulation, adds an artificial vertex v_n with no edges connecting it to other vertices. Since the artificial vertex is obviously a member of the maximal stable set of the new graph, it will used to identify the stable set and enforce the constraints of the problem. The MSS problem can be stated as:

$$\text{(MSS)} \qquad \begin{aligned} \text{Maximize} \quad & \frac{1}{2}\left(\sum_{i=1}^{n-1} v_i^2 + v_n v_i\right) \\ \text{Subject to} \quad & v \in \{-1, 1\}^n, \\ & |v_i + v_j + v_n| = 1 \text{ if } (v_i, v_j) \in \mathrm{E} \end{aligned}$$

Denoting $e_{i,j,n} \in \Re^n$ as the vector with zeros at all indices except i, j, and n, whose elements equal 1, the positive semidefinite relaxation of

MSS is

$$\text{(MSSSDP)} \qquad \begin{array}{ll} \text{Maximize} & \begin{pmatrix} .5 & & & .25 \\ & \ddots & & \vdots \\ & & .5 & .25 \\ .25 & \cdots & .25 & 0 \end{pmatrix} \bullet X \\ \text{Subject to} & diag(X) = e, \\ & (e_{i,j,n} e_{i,j,n}^T) \bullet X = 1 \qquad \forall (i,j) \in \mathrm{E} \\ & X \succeq 0 \end{array}$$

Imposing the additional constraint upon (MSSSDP) that the matrix X have rank one would make it equivalent to (MSS). Relaxing this constraint to include all symmetric positive semidefinite matrices makes the feasible region convex, and the solution to this problem provides an upper bound to the integer program (MSS). A randomized algorithm uses a solution of the relaxed problem, X^*, to identify stable sets. The randomized algorithm goes as follows:

1. Given a solution X^* to (MSSSDP), find a $V \in \Re^{n \times n}$ such that $X^* = V^T V$.

2. Select a unit vector $u \in \Re^n$ from the unit sphere and let $v = sign(V^T u)$.

3. For each $(i,j) \in \mathrm{E}$, if $|v_i + v_j + v_n| \neq 1$, change the sign of either v_i or v_j.

The stable set will be the set of vertices with the same sign as v_n. For arbitrary graphs, the constraints corresponding to the edges of the graph will be satisfied with a frequency greater than 91% [10]. The third step of the randomized algorithm ensures that no edge connects vertices in the set by selectively removing vertices from the set. The choice of whether to switch vertices v_i or v_j may be arbitrary, but a better choice may be made by switching the vertex whose value is farthest from v_n: if $|v_i - v_n| > |v_j - v_n|$, change the sign of v_i, otherwise change the sign of v_j. This randomized algorithm can be applied multiple times to calculate multiple stable sets.

In the linear programming relaxation of the maximal stable set problem, utilizing larger cliques is crucial for a tight approximation to the convex hull of the integer program. These cliques can also improve the positive semidefinite relaxation. Given cliques $\mathcal{C}^1, \ldots, \mathcal{C}^d$, such that $\mathcal{C}^k$

has n_k vertices, stable sets $v \in \{-1, 1\}^n$ must satisfy

$$|(n_k - 1)v_n + \sum_{v_i \in \mathcal{C}^k} v_i| = 1$$

for $k = 1, \ldots, d$. This formulation has a positive semidefinite relaxation that more closely approximates the convex hull of the integer program. This formulation has fewer constraints which can significantly reduce the time required to solve the positive semidefinite program.

To favor the inclusion of selected vertices into the stable set, the *weighted maximal stable set* problem has a similar formulation. Given a weights w_i on the vertices, this problem seeks to maximize

$$\frac{1}{2} \sum_{i=1}^{n-1} w_i \left(v_i^2 + v_n v_i \right)$$

subject to the same constraints as (MSS). These problems can also be addressed using the positive semidefinite relaxation.

For the graph coloring problem, instead of assigning colors or integers to the vertices of the graph, a unit vector $v_i \in \Re^n$ is assigned to the each of the n vertices i in V. To capture the property of coloring, the vectors of adjacent vertices should different in a natural way. Using the definition of [23], the vector k- coloring of G is an assignment of unit vectors $v_i \in \Re^n$ to each vertex i in V such that for any two adjacent vertices i and j, the dot product of the vectors satisfies the inequality $v_i^T v_j \leq -\frac{1}{k-1}$. In other words, the angle between the vectors of adjacent vertices must be sufficiently large. Define the matrix V such that column i is given by v_i and let $X = V^T V$. The matrix X is positive semidefinite and satisfies the inequalities $X_{ij} = X_{ji} \leq -\frac{1}{k-1}$ for each pair of adjacent edges (i, j). Obviously, any matrix is n-colorable, so the graph coloring problem can be posed as:

$$\begin{array}{lll} & \text{Minimize} & \text{rank}(X) \\ \text{(COLOR)} & & \\ & \text{Subject to} & diag(X) = e, \\ & & X_{ij} \leq -\frac{1}{n-1} \text{ for } (i,j) \in \text{E} \\ & & X \succeq 0 \end{array} \tag{1.1}$$

Ignoring the objective function, the problem is now a positive semidefinite program which seeks to find a feasible point. Heuristic algorithms can then be applied to the solution to color the graph.

Let $a_{ij} \in \Re^n$ be a vector of zeros except indices i and j, whose elements equal one. A positive semidefinite relaxation of the graph k coloring

problem can be rewritten as:

$$\begin{array}{lll} & \text{Minimize} & 0 \bullet X \\ \text{(COLORSDP)} & & \\ & \text{Subject to} & diag(X) = e, \\ & & (a_{ij}a_{ij}^T) \bullet X \leq 2 - \frac{2}{k-1} \text{ if } (i,j) \in \text{E} \end{array} \tag{1.2}$$

A solution X^* with rank less than or equal to k, identifies a legal k-coloring. The problem can be solved exactly. More generally, Karger, Motwani, Sudan propose a randomized algorithm that produces a k-semicoloring, an assignment of colors with relatively few adjacent vertices with the same color. We propose a heuristic procedure for to obtain a legal coloring, albeit with more than k colors if necessary.

Coloring Algorithm For $k = 1, \ldots,$

1. Let U^k be the uncolored vertices. If U^k is empty, terminate the algorithm.

2. Sort the vertices of U^k in decreasing order of degree in $G[U^k]$, the graph induced by the uncolored vertices, and let i be the vertex with highest degree.

3. Build a vertex set W^k by examining vertices $j \in U^k$ in the decreasing order of X_{ij}. Add j to W^k if it is not adjacent to any of the vertices in W^k.

4. Assign the vertices in W^k color k.

This algorithm is a modification of the algorithm proposed by [35]. In their algorithm, only step 3 is different. Instead of using the solution to the a positive semidefinite program, they examine the vertices in decreasing order of degree in $G[U^k]$. This algorithm remains one of the simplest and most popular, although other heuristics have been proposed and can be modified to include information inherent in the positive semidefinite program.

3 POSITIVE SEMIDEFINITE PROGRAMMING ALGORITHMS

There are actually several polynomial algorithms that can solve positive semidefinite programs. One is the primal-scaling algorithm (Nesterov and Nemirovskii [33], Alizadeh [1], Vandenberghe and Boyd [42], and Ye [44]), which is the analogue of the primal potential reduction algorithm for linear programming. This algorithm uses X to generate

the iterate direction. Another is the dual-scaling algorithm (Vandenberghe and Boyd [42], Anstreicher and Fampa [2], and Ye [44]), which is the analogue of the dual-scaling algorithm for linear programming. The dual-scaling algorithm uses only S to generate the iterate direction. The third is the primal-dual scaling algorithm which uses both X and S to generate iterate directions, including Alizadeh-Haeberly-Overton, Helmberg-Rendl-Vanderbei-Wolkowicz/ Kojima-Shida-Hara/ Monteiro, Nesterov-Todd, Gu, and Toh directions, as well as directions called the MTW and Half directions (see Todd [39] and references therein). All these algorithms possess $O(\sqrt{n}\log(1/\epsilon))$ iteration complexity to yield accuracy ϵ.

The features of the positive semidefinite program should determine which algorithm and which implementation of the algorithm is most appropriate. In contrast to applications of SDP in control theory and truss topology design, positive semidefinite programs arising in combinatorial optimization typically have many variables, contain sparse low rank constraint matrices, and require relatively low precision solutions. Although rank one matrices reduce the complexity of interior point algorithms for positive semidefinite programming by a factor of n, not all implementations utilize this structure to reduce the complexity. Our implementation of the dual scaling algorithm explicitly accounts for these features[8]. Furthermore, the dual matrix S has a sparsity pattern like that of the graph's adjacency matrix. This sparsity offers the potential for savings in computation time and memory requirements, which the dual scaling algorithm can exploit better than primal dual algorithms. Although the rate of convergence of the dual algorithms is only linear, the relatively low precision required by combinatorial problems lessens the disadvantage of slower convergence.

One assumption for the convergence of the dual scaling algorithm is that the feasible primal region has a relative interior.

Theorem 1 *The positive semidefinite relaxation (MSSSDP) has a relative interior.*

Proof. 1 *Define the vectors* $v^1, v^2, \ldots, v^n$ *by*

$$v_i^j = \begin{cases} -1 & \text{if } i = j \text{ or } i = n+1 \\ 1 & \text{otherwise} \end{cases}$$

and

$$v_i^{n+1} = \begin{cases} -1 & \text{if } 1 \le i \le n \\ 1 & \text{if } i = n+1 \end{cases}$$

These vectors satisfy the constraints of (MSS) and the matrices $X^i = v^i(v^i)^T$ *satisfy the positive semidefinite relaxation (MSSSDP). Let*

$$X = \frac{1}{n+1}\sum_{i=1}^{n+1} X^i.$$

This matrix is a strict convex combination of symmetric rank one matrices and is therefore positive semidefinite. To prove it is positive definite, it suffices to show that $\{v^i : i \in \{1, 2, \ldots, n+1\}\}$ *is linearly independent. Linear independence can be shown by evaluating the determinant of* V^n*, whose columns are the vectors* v^i*. Since* $\det(V_n) = -(2)^n \neq 0$*, the convex hull of the feasible solutions of the nonconvex optimization problems in* $\Re^n$ *contains* $n+1$ *linearly independent vectors, so the SDP relaxation has a feasible solution that is positive definite.*

Theorem 2 *The feasible region of the n-coloring problem relaxation (COLORSDP) contains a positive definite matrix.*

Proof. 2 *Let*

$$X_{ij} = \begin{cases} 1 & \text{if } i = j \\ -\frac{1}{n+1} & \text{otherwise} \end{cases}$$

Then $X = -\frac{1}{n+1}ee^T + \frac{n+2}{n+1}I$*. The matrix has one eigenvalue of* $\frac{2}{n+1}$ *and* $n-1$ *eigenvalue equal to* $\frac{n+2}{n+1}$*, which implies it is positive definite.*

Since the primal and dual problems of these SDP relaxations always have a feasible solution whose S and X part is positive definite, it follows that the primal and dual optimal values are attained and equal[37]. (Quite recently, Tuncel extended these theorems to the SDP relaxations of rather general nonconvex sets [41].)

4 COMPUTATIONAL RESULTS

In our computational experiments, we used a variety of previously tested graphs drawn from a large number of sources. For each of these graphs, we formulated the positive semidefinite relaxation of the integer combinatorial problem and solved the relaxation until a relative duality gap of 10^{-3} has been achieved.

For the maximum stable set problems, most of the graphs are taken from the 2nd DIMACS Challenge [16]. These graphs were contributed as test problems for solving the maximum clique problem. For these graphs, we took the complement of these graphs and applied our maximum stable set algorithm. The results are supplied in Table 1.1. A second set of test problems are examples of Mycielski graphs[40]. These

graphs are interesting because they contain no cliques of size larger than 2. For these graphs, we expect our relaxation to be very tight. The results are also included in Table 1.1. A third set of graphs are line graphs created from other randomly generated graphs. Three line graphs were created from a graph with 100 vertices and 248 nonzero edges. Another three line graphs were created from graphs with 200 vertices and 597 nonzero edges. These line graphs are interesting because the SDP relaxation methods and successive relaxation methods perform very poorly for the maximum stable set problem in the worst case[38]. For these line graphs, lower bounds for the maximum stable set was calculated using the program "dfmax.c", also available from the DIMACS web site[16].

For each graph, we solved the positive semidefinite relaxation, without using cliques of size 3 or larger, and applied the randomized procedure for finding stable sets. Since the time required by the randomization procedure is very small relative to the time spent on solving the positive semidefinite program, we applied the randomized procedure n times on each problem. The data in Table 1.1 includes the number of vertices ($|V|$) and edges ($|E|$) in each graph, the upper bound provided by the semidefinite relaxation (SDP), the size of the maximum stable set (Optimal), and the size the the largest stable set found using our implementation of the algorithm (DSDP).

Of the 24 graphs, we solved (MSS) exactly 14 times. In 13 of those 14 instances, the positive semidefinite relaxation was extremely tight. These 13 instances include the five Mycielski graphs, which have no large cliques. This evidence demonstrates the importance of using large cliques when such knowledge is available. The ten instances in which DSDP did not find the optimal answer included all of the line graphs. Even in the line graphs, however, the SDP relaxation proved to be within about 10% of the optimal answer. The worst results were from the problem sanr200_0.7, whose SDP bound is 23.9, maximum stable set size is 18, and DSDP answer is 11. In most cases, however, the SDP relaxation was strong and our answers were good.

For the graph coloring problem, we used examples collected by Trick and Mehrotra [40]. For these problems, we formulated and solved the relaxed of the $n-$coloring problem (COLORSDP). From this solution, we applied the graph coloring heuristic to obtain one graph coloring. Table 1.2 shows the minimal number of colors used, the number of colors we used in DSDP, and the number of colors used by the heuristic [35].

Of these test problems, the optimal coloring is known for 34 of them. In 24 of these 34 problems, we correctly identified an optimal coloring of the graph. Although the heuristic also found an optimal coloring in many of these graphs, problem `queen5.5`, utilized the solution to

Table 1.1 Maximum Stable Set Problems

Graph	$\|V\|$	$\|E\|$	SDP	Optimal	DSDP
hamming10-2	1024	5120	512.1	512	512
hamming6-2	64	192	32.0	32	32
hamming6-4	64	1312	5.35	4	4
hamming8-2	256	1024	128.0	128	128
johnson16-2-4	120	1600	8.0	8	8
johnson8-2-4	28	168	4.0	4	4
brock200_1	200	5066	27.5	21	14
brock200_3	200	7852	18.8	15	9
brock200_4	200	6811	21.3	17	9
keller4	172	5100	14.0	11	7
san200_0.9_1	200	1990	70.0	70	70
san200_0.9_2	200	1990	60.0	60	60
san200_0.9_3	200	1990	44.1	44	44
sanr200_0.7	200	6032	23.9	18	11
sanr200_0.9	200	2037	49.3	40	34
myciel3	11	20	5.0	5	5
myciel4	23	71	11.0	11	11
myciel5	47	236	23.0	23	23
myciel6	95	755	47.0	47	47
myciel7	191	2360	95.0	95	95
line1	248	1202	50.0	≥ 47	39
line2	248	1220	49.5	≥ 47	40
line3	248	1212	49.5	≥ 47	42
line4	597	3414	100.0	≥ 89	79
line5	597	3481	100.0	≥ 85	76
line6	597	3635	100.0	≥ 85	82

the positive semidefinite program to find an optimal coloring which the heuristic could not do. In a total of four problems, the coloring obtained using the SDP relaxation was better than the coloring obtained by the heuristic, but in the three `DSJC125` graphs, the coloring was actually worse.

For seven of the graphs in which we definitely did not compute the optimal coloring, we formulated a tighter formulation. Instead of using the n-color formulation, we used the k-color formulation where k is the minimal graph coloring. We solved these tighter relaxations and applied our heuristic to these solutions, hoping to identify a better coloring. The results are in Table 1.3. The number of colors required when using the tighter formulation is in the last column (DSDP2). In only one of the seven instances did the tighter formulation actually improve the coloring. On the other hand, there was one instance where the tighter formulation actually worsened the coloring of the graph. Hence, it seems sufficient to pose the n-coloring relaxation.

The time required to solve these problems ranged from less than a second for `queen5.5` to over twelve hours to find the maximum stable set of `brock200_1`. The heuristic can find answers very quickly, but the positive semidefinite relaxation may offer improved answers. For other combinatorial problems, performance guarantees for algorithms using the positive semidefinite relaxation exist. The cost of these guarantees, however, is the significant additional cost in computation time and memory requirements. This contributes the growing mountain of evidence demonstrating the high quality of solutions that can be obtained from the semidefinite relaxation.

Acknowledgments

This work was supported by the Mathematical, Information, and Computational Sciences Division subprogram of the Office of Advanced Scientific Computing Research, U.S. Department of Energy, under Contract W-31-109-Eng-38.

Table 1.2 Graph Coloring Problems

Graph	$\|V\|$	$\|E\|$	Optimal	DSDP	Heuristic
anna	138	493	11	11	11
david	87	406	11	11	11
homer	561	1629	13	13	13
huck	74	301	11	11	11
jean	80	254	10	10	10
games120	120	638	9	9	9
miles250	128	387	8	8	8
miles500	128	1170	20	20	20
miles750	128	2113	31	32	32
miles1000	128	3216	42	42	42
miles1500	128	5198	73	73	73
queen5.5	25	160	5	5	7
queen6.6	36	290	7	9	9
queen7.7	49	476	7	11	11
queen8.8	64	728	9	11	12
queen9.9	81	1055	10	13	13
queen10.10	100	1470	?	14	14
queen11.11	121	1980	11	15	15
queen12.12	144	2596	?	17	17
queen13.13	169	3328	13	18	18
queen14.14	196	4186	?	19	19
myciel3	11	20	4	4	4
myciel4	23	71	5	5	5
myciel5	47	236	6	6	6
myciel6	95	755	7	7	7
myciel7	191	2360	8	8	8
zeroin.i.1	211	4100	49	49	49
zeroin.i.2	211	3541	30	30	30
zeroin.i.3	206	3540	30	30	30
mulsol.i.1	197	3925	49	49	49
mulsol.i.2	188	3885	31	31	31
mulsol.i.3	184	3916	31	31	31
mulsol.i.4	185	3946	31	31	31
mulsol.i.5	186	3973	31	31	31
DSJC125.1	125	736	?	6	7
DSJC125.5	125	3891	?	21	22
DSJC125.9	125	6961	?	49	50
DSJC250.1	250	3218	?	11	11
DSJR500.1	500	3555	?	13	13

Table 1.3 Graph Coloring Problems with a Tighter Relaxation

Graph	$\|V\|$	$\|E\|$	Optimal	DSDP	DSDP2
miles750	128	2113	31	32	32
queen6.6	36	290	7	9	9
queen7.7	49	476	7	11	10
queen8.8	64	728	9	11	12
queen9.9	81	1055	10	13	13
queen11.11	121	1980	11	15	15
queen13.13	169	3328	13	18	18

References

[1] F. Alizadeh. *Combinatorial optimization with interior point methods and semidefinite matrices.* PhD thesis, University of Minnesota, Minneapolis, MN, 1991.

[2] K. M. Anstreicher and M. Fampa. A long-step path following algorithm for semidefinite programming problems. Working Paper, Department of Management Science, The University of Iowa, Iowa City, IA, 1996.

[3] L. Babel. Finding maximum cliques in arbitrary and in special graphs. *Computing*, 46:321–341, 1991.

[4] L. Babel and G. Tinhofer. A branch and bound algorithm for the maximum clique problem. *J. of Global Optimization*, 4, 1994.

[5] Egon Balas and H. Samuelsson. A node covering algorithm. *Naval Research Logistics Quarterly*, 24(2):213–233, 1977.

[6] Egon Balas and Jue Xue. Minimum weighted coloring of triangulated graphs, with application to maximum weight vertex packing and clique finding in arbitrary graphs. *SIAM Journal on Computing*, 20(2):209–221, April 1991.

[7] Egon Balas and Chang Sung Yu. Finding a maximum clique in an arbitrary graph. *SIAM Journal on Computing*, 15(4):1054–1068, November 1986.

[8] S. Benson, Y. Ye, and X. Zhang. Solving large–scale sparse semidefinite programs for combinatorial optimization. Technical report, Department of Management Science, University of Iowa, Iowa City, IA 52242, USA, September 1997. To appear in *SIAM J. of Optimization.*

[9] C. Berge. *Graphs and Hypergraphs*. North–Holland, Amsterdam, 1973.

[10] D. Bertsimas and Y. Ye. Semidefinite relaxations, multivariate normal distributions, and order statistics. In D.-Z. Du and P.M. Pardalos, editors, *Handbook of Combinatorial Optimization*, volume 3, pages 1–19. Kluwer Academic Publishers, 1998.

[11] P. Briggs, K. Cooper, K. Kennedy, and L. Torczon. Coloring heuristics for register allocation. In *ASCM Conference on Program Language Design and Implementation*, pages 275–284. The Association for Computing Machinery, 1998.

[12] R. Carrahan and P. M. Pardalos. An exact algorithm for the maximum clique problem. *Operations Research Letters*, 9:375–382, 1990.

[13] G.J. Chaitin, M. Auslander, A.K. Chandra, J. Cocke, M.E. Hopkins, and P. Markstein. Register allocation via coloring. *Computer Languages*, 6:47–57, 1981.

[14] Gregory J. Chaitin. Register allocation and spilling via graph coloring. *SIGPLAN Notices (Proceedings of the SIGPLAN '82 Symposium on Compiler Construction, Boston, Mass.)*, 17(6):98–101, June 1982.

[15] D. De Werra. An introduction to timetabling. *European Journal of Operations Research*, 19:151–162, 1985.

[16] DIMACS Center Web Page. The Second DIMACS Implementation Challenge: 1992-1993. ftp://dimacs.rutgers.edu/pub/challenge/graph/benchmarks/clique/.

[17] C. Friden, A. Hertz, and D. de Werra. An exact algorithm based on tabu search for finding a maximum inderpendent set in a graph. *Computers Operations Research*, 17(5):375–382, 1990.

[18] Andreas Gamst. Some lower bounds for a class of frequency assignment problems. *IEEE Transactions of Vehicular Technology*, 35(1):8–14, 1986.

[19] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP–Completeness*. W.H Freeman, San Francisco, CA, 1979.

[20] M. X. Goemans and D. P. Williamson. .878-approximation for MAX CUT and MAX 2SAT. In *Proc.* 26^{th} *ACM Symp. Theor. Computing*, pages 422–431, 1994.

[21] David S. Johnson. Approximation algorithms for combinatorial problems. *Journal of Computer and System Sciences*, 9:256–278, 1974.

[22] David S. Johnson. Worst case behavior of graph coloring algorithms. In *Proceedings of 5th Southeastern Conference on Combinatorics, Graph Theory, and Computing*, pages 513–527. Utilitas Mathematica, Winnipeg, Canada, 1974.

[23] David Karger, Rajeev Motwani, and Madhu Sudan. Approximate graph coloring by semidefinite programming. Technical report, MIT, Cambridge, MA 52242, USA, 1994.

[24] Jon Kleinberg and Michel X. Goemans. The Lovász theta function and a semidefinite programming relaxation of vertex cover. *SIAM Journal on Discrete Mathematics*, 11(2):196–204, May 1998.

[25] M. Kubale and B. Jackowski. A generalized implicit enumeration algorithm for graph coloring. *Communications of the ACM*, 28:412–418, 1985.

[26] M. Kubale and E. Kusz. Computational experience with implicit enumeration algorithms for graph coloring. In *Proceedings of the WG'83 International Workshop on Graphtheoretic Concepts in Computer Science*, pages 167–176, Linz, 1983. Trauner Verlag.

[27] Carlo Mannino and Antonio Sassano. An exact algorithm for the maximum cardinality stable set problem. *Networks*, page (submitted), 1993. ftp://dimacs.rutgers.edu/pub/challenge/graph/ contributed.

[28] Anuj Mehrotra and Michael A. Trick. A column generation approach for graph coloring. http://mat.gsia.cmu.edu/trick.html, April 1995.

[29] Craig A. Morgenstern and Harry D. Shapiro. Coloration neighborhood structures for general graph coloring. In *Proceedings of the First Annual ACM-SIAM Symposium on Discrete Algorithms, San Francisco, Jan, 1990.* Society for Industrial and Applied Mathematics, Philadelpia, 1990.

[30] George L. Nemhauser and G. L. Sigismondi. A strong cutting plane / branch and bound algorithm for node packing. *Journal of the Operational Research Society*, 43(5), 1992.

[31] George. L. Nemhauser and Les. E. Trotter, Jr. Vertex packings: Structural properties and algorithms. *Mathematical Programming*, 8(2):232–248, 1975.

[32] Yu. E. Nesterov. Semidefinite relaxation and nonconvex quadratic optimization. *Optimization Methods and Software*, 9:141–160, 1998.

[33] Yu. E. Nesterov and A. S. Nemirovskii. *Interior Point Polynomial Methods in Convex Programming: Theory and Algorithms.* SIAM Publications, SIAM, Philadelphia, 1993.

[34] B. Pittel. On the probable behaviour of some algorithms for finding the stability number of a graph. *Mathematical Proceedings of the Cambridge Philosophical Society*, 92:511–526, 1982.

[35] M. J. D. Powell and P. L. Toint. On the estimation of sparse hessian matrices. *SIAM Journal on Numerical Analysis*, 16:1060–1074, 1979.

[36] Proc. 4th IPCO Conference. *Improved approximation algorithms for max k-cut and max bisection*, 1995.

[37] M. V. Ramana, L. Tunçel, and H. Wolkowicz. Strong duality for semidefinite programming. *SIAM Journal on Optimization*, 7:641–662, 1997.

[38] T. Stephen and L. Tuncel. On a representation of the matching polytope via semidefinite liftings. *Mathematics of Operations Research*, 24(1):1–7, 1999.

[39] M. J. Todd. On search directions in interior-point methods for semidefinite programming. Technical Report 1205, School of Operations Research and Industrial Engineering, Cornell University, Itheca, NY 14853–3801, October 1997.

[40] M. Trick. Graph coloring instances. http://mat.gsia.cmu.edu/COLOR/instances.html.

[41] L. Tuncel. On the slater condition for the sdp relaxations of nonconvex sets. Technical Report CORR2000-13, Department of Combinatorica and Optimization, University of Waterloo, Waterloo, Ontario N2L3G1, Canada, February 2000.

[42] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Review*, 38(1):49–95, 1996.

[43] D. C. Wood. A technique for coloring a graph applicable to large scale time-tabling problems. *The Computer Journal*, 3:317–319, 1969.

[44] Y. Ye. *Interior Point Algorithms : Theory and Analysis.* Wiley-Interscience Series in Discrete Mathematics and Optimization. John Wiley & Sons, New York, 1997.

[45] Y. Ye. Approximating quadratic programming with bound and quadratic constraints. *Mathematical Programming*, 84:219–226, 1999.