The Maximum Clique Problem

PANOS M. PARDALOS
Department of Industrial and Systems Engineering, 303 Weil Hall, University of Florida, Gainesville, FL 32611, U.S.A.

JUE XUE
Graduate School of Management, Clark University, 950 Main Street, Worcester, MA 01610, U.S.A.

(Received: 6 September 1992; accepted: 5 July 1993)

Abstract. In this paper we present a survey of results concerning algorithms, complexity, and applications of the maximum clique problem. We discuss enumerative and exact algorithms, heuristics, and a variety of other proposed methods. An up to date bibliography on the maximum clique and related problems is also provided.

Key words. Maximum clique problem, graph coloring, heuristics, algorithms, *NP*-hard, bibliography, survey.

1. Introduction

Throughout this paper, $G = (V, E)$ is an arbitrary undirected and weighted graph unless otherwise specified. $V = \{1, 2, \ldots, n\}$ is the vertex set of G, and $E \subseteq V \times V$ is the edge set of G. For each vertex $i \in V$, a positive weight w_i is associated with i. $A_G = (a_{ij})_{n \times n}$ is the adjacency matrix of G, where $a_{ij} = 1$ if $(i, j) \in E$ is an edge of G, and $a_{ij} = 0$ if $(i, j) \notin E$.

The *complement graph of* $G = (V, E)$ is the graph $\bar{G} = (V, \bar{E})$, where $\bar{E} = \{(i, j) \mid i, j \in V, i \neq j \text{ and } (i, j) \notin E\}$. For a subset $S \subseteq V$, we define the weight of S to be $W(S) = \Sigma_{i \in S} w_i$. We call $G(S) = (S, E \cap S \times S)$ *the subgraph induced by* S.

A graph $G = (V, E)$ is *complete* if all its vertices are pairwise adjacent, i.e. $\forall i, j \in V$, $(i, j) \in E$. A *clique* C is a subset of V such that $G(C)$ is complete. The maximum clique problem asks for a clique of maximum weight.

An *independent set* (*stable set*, *vertex packing*) is a subset of V, whose elements are pairwise nonadjacent. The maximum independent set problem asks for an independent set of maximum cardinality. The size of a maximum independent set is the *stability number* of G (denoted by $\alpha(G)$). The maximum weight independent set problem asks for an independent set of maximum weight.

We should distinguish a *maximum* clique (independent set) from a *maximal* clique (independent set). A maximal clique (independent set) is a clique (independent set) that is not a subset of any other clique (independent set). A maximum clique (independent set) is a maximal clique (independent set) that has the maximum cardinality or weight.

A *vertex cover* is a subset of V, such that every edge $(i, j) \in E$ has at least one endpoint i or j in the subset. The minimum vertex cover problem asks for a vertex

Journal of Global Optimization **4**: 301–328, 1994.

cover of minimum cardinality. The minimum weighted vertex cover problem asks for vertex cover of minimum weight.

It is easy to see that S is a clique of G if and only if S is an independent set of $\bar{G}$, and if and only if $V \backslash S$ is a vertex cover of $\bar{G}$. Any result obtained for one of the above problems has its equivalent forms for the other problems. Furthermore, these problems are *NP*-complete on arbitrary graphs (see Gary and Johnson [79]).

The maximum clique problem has many equivalent formulations as an integer programming problem, or as a continuous nonconvex optimization problem. The simplest one is the following *edge formulation*:

$$\begin{aligned} &\max \sum_{i=1}^{n} w_i x_i , \\ &\text{s.t. } x_i + x_j \leq 1, \forall (i, j) \in \bar{E} , \\ &x_i \in \{0, 1\} , \quad i = 1, \ldots, n . \end{aligned} \tag{1}$$

A polyhedral result concerning formulation (1) is due to Nemhauser and Trotter ([185], [186]). In 1975 they found that if a variable x_i had integer value 1 in an optimal solution to the linear relaxation of (1), then $x_i = 1$ in at least one optimal solution to (1).

THEOREM 1.1 (see [186], also [210]). *Let x be an optimum $(0, \frac{1}{2}, 1)$-valued solution to the linear relaxation of* (1), *and let $P = \{j | x_j = 1\}$. There exists an optimum solution x^* to* (1) *such that $x_j^* = 1$, $\forall j \in P$.*

This theorem suggests an implicit enumerative algorithm for (1) via solving its linear relaxation problem. However, in most cases, few variables have integer values in an optimal solution to the linear relaxation of (1), and the gap between the optimal values of (1) and its linear relaxation problem is too large. These seriously restrict the use of this approach.

Let $\mathcal{S}$ denote the set of all maximal independent sets of G. An alternative formulation is the following *independent set* formulation.

$$\begin{aligned} &\max \sum_{i=1}^{n} w_i x_i , \\ &\text{s.t. } \sum_{i \in S} x_i \leq 1, \forall S \in \mathcal{S} , \\ &x_i \in \{0, 1\} , \quad i = 1, \ldots, n . \end{aligned} \tag{2}$$

The advantage of formulation (2) over (1) is a smaller gap between the optimal values of (2) and its linear relaxation. However, since the number of constraints in (2) is exponential, solving the linear relaxation of (2) is not an easy problem. In fact, Grötschel *et al.* ([98] and [99]) have shown that the linear relaxation

problem of (2) is *NP*-hard on general graphs. They have also shown that the same problem is polynomially solvable on *perfect* graphs. Furthermore, they have shown that a graph is perfect if and only if the optimal solution to the linear relaxation of (2) always assumes integer values. The following results can be found from [97], [98], [99], and [100].

THEOREM 1.2. *Let G be an arbitrary graph. The linear relaxation problem of* (2) *is NP-complete.*

THEOREM 1.3. *G is a perfect graph if and only if the linear relaxation of* (2) *has integer solutions for any $w \in R^n$.*

THEOREM 1.4. *The maximum weight clique problem for perfect graphs can be solved in polynomial time.*

Besides the above formulations for the maximum clique problem, we can also find in the literature many other formulations. For example, consider the following indefinite [205] quadratic problem

$$\begin{aligned} &\text{global max } f(x) = \tfrac{1}{2} x^T A_G x\,, \\ &\text{s.t.} \sum_{i=1}^{n} x_i = 1\,, \quad x_i \geq 0\,, \quad i = 1, \ldots, n\,. \end{aligned} \tag{3}$$

In Theorem 1 of Motzkin and Straus [180], the following result was proved (see also [1]).

THEOREM 1.5. *Let x^* and $\alpha = f(x^*)$ be the optimal solution and the corresponding objective value of problem* (3). *Then G has a maximum clique C of size $k = 1/(1-2\alpha)$. The global maximum of* (3) *can be attained by setting $x_i^* = \frac{1}{k}$ if $v_i \in C$, and $x_i^* = 0$ otherwise.*

Although this characterization does not provide an efficient approach to solve the maximum clique problem, it can be used to prove certain bounds on the size of the maximum clique. The following theorem is a consequence of a more general result from Hager *et al.* [103].

THEOREM 1.6. *If A_G has r negative eigenvalues, then at least $n - r$ constraints are active at any global maximum x^* of $f(x)$.*

Here, by active constraints of (3) at a global maximum x^*, we mean those constraints $x_i \geq 0$ satisfying $x_i^* = 0$, $i = 1, \ldots, n$. Note that the constraint $\Sigma_{i=1}^n x_i = 1$ is always active by definition. Combining theorems 1.5 and 1.6, we can obtain an upper bound on the size of the maximum clique of G (see also Pardalos and

Phillips [201]): If A_G has r negative eigenvalues, the size $|C|$ of the maximum clique C is bounded by $|C| \leq r+1$.

In 1990, Shor ([226]) considered an interesting formulation of the maximum weight independent set problem. Note that the maximum weight independent set problem can be formulated as

$$\begin{aligned} &\text{global min } f(x) = \sum_{i=1}^{n} w_i x_i , \\ &\text{s.t. } x_i + x_j \leq 1 , \quad \forall (i,j) \in E , \quad x \in \{0,1\}^n . \end{aligned} \tag{4}$$

The above formulation is equivalent to the following quadratically constrained global optimization problem

$$\begin{aligned} &\text{global min } f(x) = \sum_{i=1}^{n} w_i x_i , \\ &\text{s.t. } x_i x_j = 0 , \quad \forall (i,j) \in E , \\ &x_i^2 - x_i = 0 , \quad i = 1, 2, \ldots, n . \end{aligned} \tag{5}$$

Applying dual quadratic estimates, Shor reported very good computational results using (5). It seems that for the maximum clique problem, a *good* formulation of the problem is of crucial importance in solving the problem.

The general *quadratic optimization* problem is of the form

$$\begin{aligned} &\min f(x) = c^T x + \tfrac{1}{2} x^T Q x , \\ &\text{s.t. } Ax = b , \quad x \in D , \end{aligned} \tag{6}$$

where $c \in R^n$, $Q \in R^{n \times n}$, $A \in R^{m \times n}$, and $D \subseteq R^n$.

Next, we formulate the maximum clique, the maximum independent set and the maximum weight independent set problems as quadratic zero-one problems. We use I to denote the $n \times n$ identity matrix. To facilitate our discussion, define a transformation T from $\{0,1\}^n$ to 2^V,

$$T(x) = \{ i \,|\, x_i = 1, i \in V \} , \quad \forall x \in \{0,1\}^n .$$

Denote the inverse of T by T^{-1}. If $x = T^{-1}(S)$ for some $S \in 2^V$ then $x_i = 1$ if $i \in S$ and $x_i = 0$ if $i \notin S$, $i = 1, \ldots, n$.

We can rewrite the maximum problem (1) as a minimization problem (when $w_i = 1$)

$$\begin{aligned} &\text{global min } f(x) = -\sum_{i=1}^{n} x_i , \\ &\text{s.t. } x_i + x_j \leq 1 , \quad \forall (i,j) \in \bar{E} , \quad x \in \{0,1\}^n . \end{aligned} \tag{7}$$

If x^* solves (7), then the set $C = T(x^*)$ is a maximum clique of G with $|C| = -z = -f(x^*)$.

Another way of stating the constraints for (7) is to make use of the fact that the quadratic expressions $x_i x_j = 0$ for all $(i, j) \in \bar{E}$ since for $x_i, x_j \in \{0, 1\}$ $x_i + x_j \leq 1$ if and only if $x_i x_j = 0$. The constraints in (7) can be removed by adding two times the quadratic terms to the objective function, which is now

$$f(x) = -\sum_{i=1}^{n} x_i + 2 \sum_{(i, j) \in \bar{E},\, i>j} x_i x_j = x^T (A_{\bar{G}} - I) x .$$

The quadratic terms represent penalties for violations of $x_i x_j = 0$. This leads to the following theorem.

THEOREM 1.7. *The maximum clique problem is equivalent to the following global quadratic zero-one problem*

$$\begin{aligned} &\textit{global} \min f(x) = x^T A x , \\ &\text{s.t. } x \in \{0, 1\}^n , \quad \textit{where } A = A_{\bar{G}} - I . \end{aligned} \tag{8}$$

If x^* solves (8), *then the set C defined by $C = T(x^*)$ is a maximum clique of G with $|C| = -z = -f(x^*)$.*

The off-diagonal elements of the matrix A are the same as the adjacency matrix of $\bar{G}$. Hence, formulations (7) and (8) are advantageous for dense graphs because a sparse data structure can be used (for details, see [203]). Following the equivalence of the maximum clique problem with the maximum independent set problem, we have

THEOREM 1.8. *The maximum independent set problem is equivalent to the following global quadratic zero-one problem*

$$\begin{aligned} &\textit{global} \min f(x) = x^T A x \\ &\textit{s.t. } x \in \{0, 1\}^n, \textit{ where } A = A_G - I . \end{aligned} \tag{9}$$

If x^* solves (9), *then the set S defined by $S = T(x^*)$ is a maximum independent set of G with $|S| = -z = -f(x^*)$.*

Next, we discuss the maximum weight independent set problem. The above theorems for the maximum clique problem and the maximum independent set problem can be regarded as a special case by taking $w_i = 1$, $i = 1, 2, \ldots, n$.

THEOREM 1.9. *The maximum weight independent set problem is equivalent to the following global quadratic zero-one problem*

$$\begin{aligned} &\textit{global} \min f(x) = x^T A x\,, \\ &s.t.\ x \in \{0,1\}^n\,, \end{aligned} \tag{10}$$

where $a_{ii} = -w_i$, $i = 1, \ldots, n$, $a_{ij} = \frac{1}{2}(w_i + w_j)$, $\forall (i,j) \in E$, *and* $a_{ij} = 0$, $\forall (i,j) \in \bar{E}$.

If x^* *solves* (10), *then the set* S *defined by* $S = T(x^*)$ *is a maximum independent set of* G *with weight* $W(S) = -z = -f(x^*)$.

As with many problems of combinatorial optimization, using the appropriate formulation of the maximum clique problem is of crucial importance in solving the problem. In addition, using different formulations, we gain more insight into the problem's complexity and we can prove interesting results.

2. Complexity

The maximum clique, maximum independent set and minimum vertex cover problems are computationally equivalent on arbitrary graphs. They are also known to be *NP*-complete. Furthermore, for the maximum clique problem, the complexity of approximating remained an open question until recently. In [198], Papadimitriou and Yannakakis introduced the complexity class *MAX SNP* and showed that many natural problems are complete in this class, relative to a reducibility that preserves the quality of approximation. For example, the vertex cover problem (for constant degree graphs), min cut problem, dominating set problem, and the MAX 3-SAT problem are such complete problems [258].

If the solution to any of these complete problems can be approximated to arbitrary small constant factors, then the optimum solution to any problem in the class can be approximated to arbitrarily small constant factors. The question of whether such approximation schemes can be found for the complete problems in this class was left unresolved. In [29], Berman and Schnitger have shown that if one of the *MAX SNP* problems does not have polynomial time approximation schemes, then there is an $\epsilon > 0$ such that the maximum clique problem cannot be approximated in polynomial time with performance ratio

$$\frac{\textit{size of maximum clique}}{\textit{size of approximate clique}} = O(n^{\epsilon})\,,$$

where n is the number of vertices in the graph (see also Feige *et al.* [71], where a connection between approximation complexity and interactive proof systems is discussed).

A breakthrough in approximation complexity is the recent result by Arora *et al.* [6], [7]. It is shown that the maximum number of satisfiable clauses in a 3-SAT formula (MAX 3-SAT) cannot be approximated to arbitrary small constants

(unless $P = NP$), thus resolving the open question in [198]. This immediately shows the difficulty of finding good approximate solutions to all the above listed problems. In particular, it is shown that no polynomial time algorithm can approximate the maximum clique size within a factor of n^{ϵ} ($\epsilon > 0$), unless $P = NP$ (by using the results of Feige *et al.* [71]).

Although these complexity results characterize worst case instances, they nevertheless indicate that the maximum clique problem is indeed a very difficult problem to solve.

Some other results in the literature concerning the approximation of the maximum clique/independent set problem on arbitrary or special graphs can be found in [34], [47], [49], [57], [182], [214].

If we restrict ourselves to graphs with special structure, then in many cases the maximum clique/independent set problem can be solved in polynomial time. For example, Balas *et al.* ([15]) introduced several classes of graphs and showed that the maximum weight clique problem can be solved in polynomial time on them. In Balas and Yu ([21]), they discussed classes of graphs that have polynomially many maximal cliques. On these graphs, the maximum weight clique problem can also be solved in polynomial time.

A well known class of graphs where the maximum clique problem is polynomially solvable, is the class of perfect graphs ([27]). A graph G is perfect if every induced subgraph of G has the property that the size of its maximum clique equals the minimum number of independent sets needed to cover all the vertices (commonly called a *coloring* in the literature). Since the complement graph of a perfect graph is also perfect, Theorem 1.4 of the last section states that the maximum clique problem can be solved in polynomial time on perfect graphs and their complements. The class of perfect graphs contains many well known graphs in the literature (see [92]). For example, bipartite graphs, interval graphs, and triangulated graphs ([80], [74], [218], [219], [236]). Examples of more recently found perfect graphs are *Meyniel graphs* ([174], [40]), *quasi parity graphs* ([175]), *weakly triangulated graphs* ([112], [113]), *perfectly orderable graphs* ([55]) and *unimodular graphs* ([115]).

A class of graphs that is closely related to the perfect graphs is the *t-perfect* graphs. This class of graphs was defined in [52]. Polynomial algorithms for the maximum weight independent set problem on *t-perfect* graphs exist ([99]). The class of *t-perfect* graphs contains *bipartite graphs*, *series-parallel graphs* ([62], [52], [35]), and *strongly t-perfect graphs* ([86]). For a polynomial time algorithm for solving the maximum weight independent set problem on a bipartite graph $G(V_1, V_2)$ see the book by Lawler [148].

Other special classes of graphs where the maximum clique/independent set problem have been studied in the literature can be found in [14], [41], [47], [48], [51], [49], [63], [81], [82], [93], [96], [101], [116], [120], [121], [122], [138], [141], [166], [167], [177], [183], [188], [197], [213], [220], [221], [233], [227], and [260].

We should note here that the weighted or unweighted version of the maximum

clique problem, the maximum independent set problem, and the minimum vertex cover problem may not be equivalent on graphs with special structures.

3. Enumerative Algorithms

The first algorithm for enumerating all cliques of an arbitrary graph in the literature is probably due to Harary and Ross [110]. In 1957, they proposed an inductive method that first identified all the cliques of a special graph with no more than three cliques. Then the problem on general graphs is reduced to this special case. Their work was stimulated by the matrix manipulation problem of sociometric data to find a complete identification of cliques.

Early works following that of Harary and Ross [110] can be found in Maghout [164], Paull and Unger [206], Bonner [33], Marcus [165], and Bednarek and Taulbee [25], [64]. What Paull and Unger ([206]), and Marcus ([165]) proposed were algorithms to minimize the number of rows in a flow table for a sequential switching function. The problem addressed in Bonner ([33]) was the clustering problem in information systems. Bednarek and Taulbee [25] proposed algorithms for generating all maximal chains of a set with a binary relation defined on it. Although these problems come from different fields and appear dealing with different problems, they are solving the same problem of enumerating all cliques of a graph. With the technology at that time, these early algorithms could only be tested on special graphs.

In 1970, Auguston and Minker [8] investigated several graph theoretic clustering techniques used in information systems. In their work, the algorithm of Bierstone (see [8], [181]) and the algorithm of Bonner [33] were tested. The method used in both algorithms was called the *vertex sequence method* or *point removal method*. This method produces cliques of G from the cliques of $G - v$, $v \in G$. From their computational results, they found the algorithm of Bierstone was more efficient. The original work of Bierstone was not published. The version of Bierstone's algorithm contained in Auguston and Minker [8] had two errors that were corrected by Mulligan and Corneil [181] in 1972.

Then in 1973, two new algorithms using the *backtracking method* were proposed by Akkoyunlu [2], and by Bron and Kerbosch [37]. The advantage of the backtracking method is the elimination of the redundancy in generating the same clique. What was more important for these two algorithms was their polynomial storage requirements. For example, the Bron and Kerbosch algorithm requires at most $\frac{1}{2}n(n+3)$ storage space. Bron and Kerbosch tested their algorithm on graphs of 10 to 50 vertices and densities ranging from 10% to 95%. Here the density was defined as the probability of a pair of vertices being connected. They found their algorithm was much more efficient than Bierstone's algorithm. One very interesting phenomenon from their test was the ratio of CPU time over the number of cliques of the graph, as they put it, "hardly dependent

on the size of the graph". Bron and Kerbosch's algorithm is *Algorithm* 457 in the ACM collection.

More enumerative algorithms were proposed in the 70's following that of Bron and Kerbosch. For example, the works by Osteen and Tou [191], Osteen [190], Meeusen and Cuyvers [171], Johnson [132], Johnston [131], Leifman [151], Tsukiyama *et al.* [244], Gerhards and Lindenberg [85].

The algorithm of Osteen and Tou [191] was an improved version of the point removal method. Osteen's [190] algorithm was designed for a special class of graphs. The algorithm of Meeusen and Cuyvers [171] started with decomposing a graph into subgraphs satisfying *the chain of subsets in G* requirement. Such a decomposition had the property that every clique is contained completely in at least one subgraph. Based on this property, they proposed an algorithm to find all cliques of a graph. The work of Johnston [131], contains a family of algorithms that are variations of Bron and Kerbosch's algorithm. By comparing several algorithms computationally, Johnston [131] concluded that the Bron and Kerbosch algorithm was one of the most efficient algorithms.

Tsukiyama *et al.* [244] proposed an enumerative algorithm that combined the approaches used by Auguston and Minker [8], Akkoyunlu [2], and Bron and Kerbosch [37]. The result was an algorithm with time complexity of $O(nm\mu)$ and storage requirement of $O(n+m)$, where n, m, μ are the number of vertices, edges and maximal cliques of a graph. This bound is stronger than the earlier bound of $O(\mu^2)$ due to Auguston and Minker [8] (which was pointed out by Tsukiyama *et al.* [244]). The algorithm of Gerhards and Lindenberg [85] started with partial cliques related to fixed vertices of G. Then, cliques were generated from these partial cliques. Their computational results suggested their proposed algorithm was as efficient as that of Bron and Kerbosch [37] for general graphs, but more efficient on sparse graphs.

In 1980's, other proposed algorithms include the algorithms of Loukakis and Tsouros [156], Loukakis [155], Chiba and Nishizeki [46], Tomita *et al.* [240], and Johnson *et al.* [129].

Loukakis and Tsouros [156] proposed a depth-first enumerative algorithm that generated all maximal independent sets lexicographically. They compared their algorithm with the algorithm of Bron and Kerbosch [37], and the algorithm of Tsukiyama *et al.* [244]. Their computational results on graphs of up to 220 vertices suggested the superior efficiency of their algorithm. Namely, their algorithm was two to fifteen times faster than that of Bron and Kerbosch [37], and three times faster than that of Tsukiyama *et al.* [244]. Two years later, Loukakis [155] claimed an additional improvement of three folds of time saving over Loukakis and Tsouros [156]. Loukakis [155] tested his algorithm on graphs of 30 to 220 vertices and 10% (for small graphs) to 90% (for large graphs) densities.

In 1988, Johnson *et al.* [129] proposed an algorithm that enumerated all maximal independent sets in lexicographic order. The algorithm has an $O(n^3)$ delay between the generation of two subsequent independent sets. Based on one

of their results given below (see [129]), they also showed the nonexistence of a polynomial-delay algorithm for enumerating maximal cliques in reverse lexicographic order (if $P \neq NP$).

THEOREM 3.1. ([129]). *Given a graph G and a maximal independent set S, it is coNP-complete to prove whether S is the lexicographically last maximal independent set of G.*

Chiba and Nishizeki's [46] algorithm lists all cliques with time complexity of $O(a(G)m\mu)$, where $a(G)$ is the *arboricity* of graph G. This is an improvement over the time complexity of Tsukiyama *et al.* [244].

Finally, Tomita *et al.* [240] proposed a modified Bron and Kerbosch [37] algorithm and claimed its time complexity to be $O(3^{n/3})$. As they pointed out, this was the best one could hope for since the Moon and Moser graphs [178] have $3^{n/3}$ maximal cliques.

4. Exact Algorithms for the Unweighted Case

If our goal is to find a maximum clique or just the size of a maximum clique, a lot of work can be saved from the above enumerative algorithms. Because once we find a clique, we only need to enumerate cliques better than the current best clique. Modifying the enumerative algorithms based on this argument results in various implicit enumerative algorithms. This argument can also be used in designing implicit enumerative algorithms.

The most well known and commonly used implicit enumerative method for the maximum clique problem is the *branch and bound* method. Background information on how branch and bound method works can be found in, for example, [17] and [187]. The key issues in a branch and bound algorithm for the maximum clique problem are:

1. How to find a good lower bound, i.e. a clique of large size?
2. How to find a good upper bound on the size of maximum clique?
3. How to branch, i.e., break a problem into small subproblems?

Implicit enumerative algorithms for the maximum clique/independent set problem started in the 1970's by Desler and Hakimi [61], Tarjan [234], and Houck [118]. These early works were improved in 1977 by Tarjan and Trojanowski [235], and by Houck and Vemuganti [119]. In Tarjan and Trojanowski [235], they proposed a recursive algorithm for the maximum independent set problem. They showed their algorithm had a time complexity of $O(2^{n/3})$. This time bound illustrated that it was possible to solve a *NP*-complete problem much faster than the simple, enumerative approach. In the same year, Chvátal [53] showed that using a certain type of *recursive proofs* (see [53]) to show the upper bound on the stability number "has length at least $O(c^n)$", where $c > 1$ is a constant. The work of Houck and Vemuganti [119] exploited the relationship between the maximum

independent set and a special class of bipartite graphs. They used this relationship to find an initial solution in their algorithm for the maximum independent set problem.

Most algorithms in the literature for the maximum clique/independent set problem were proposed in the 1980's. For example, in 1982, Loukakis and Tsouros [157] proposed a *tree search* algorithm that finds the size of a maximum independent set. Then in 1984, Ebenegger *et al.* [65] proposed another algorithm for finding the stability number of a graph. Their approach is based on the relationship between the maximization of a pseudo-Boolean function and the stability number of a graph. Computational tests on graphs with up to 100 vertices were reported in Ebenegger *et al.* [65].

One of the most important contributions in the 1980's on practical algorithms for the maximum clique problem is due to Balas and Yu [20]. In their algorithm, the implicit enumeration was implemented in a new way. The idea of their approach is as follows. First, find a maximal induced triangulated subgraph T of G. Once T is found, find a maximum clique of T. This clique provides a lower bound and a feasible solution to the maximum clique problem. Then, they used a heuristic coloring procedure to extend T to a larger (maximal) subgraph that had no clique better than the current best clique. The importance of this second step is that it helps to reduce the number of subproblems generated from each node of the search tree, which in turn, reduces the size of the whole search tree. They solved the maximum clique problem on graphs of up to 400 vertices and 30,000 edges. Comparing their algorithm with other such algorithms, they found their algorithm not only generated a smaller search tree, but also required less CPU times.

In 1986, Kikusts [140] proposed a branch and bound algorithm for the maximum independent set problem based on a new recursive relation for the stability number of a graph G. Namely,

$$\alpha(G) = max\{ + \alpha(G - v - N(v)), \alpha'_v\} , \tag{11}$$

where $\alpha'_v = max\{|I| \ I \subseteq G - v$ is independent, $|I \cap N(v)| \geq 2\}$. This relation is different from the recursive relation

$$\alpha(G) = max\{1 + \alpha(G - v - N(v)), \alpha(G - v)\} , \tag{12}$$

traditionally used in designing branch and bound algorithms for the maximum independent set problem. Intuitively, relation (11) is stronger than relation (12). However, the trade off is a more complicated situation in (11). Some computational results were provided in [140] without comparing with other algorithms.

Also in 1986, Robson [217] proposed a modified recursive algorithm of Tarjan and Trojanowski [235]. Robson showed through a detailed case analysis that his algorithm had a time complexity of $O(2^{0.276n})$. This is an improvement over the time complexity of $O(2^{n/3})$ of Tarjan and Trojanowski [235]. Here we want to

mention the complexity proof of a similar recursive algorithm by Wilf [252]. Although Wilf's complexity of $O(1.39^n)$ is not as tight as that of Tarjan and Trojanowski [235], his proof is much simpler. Also in [252], Wilf proved (under certain probabilistic assumptions) that the average number of independent sets in a graph with n vertices is given by:

$$I_n = \sum_{k=0}^{n} \binom{n}{k} 2^{-k(k-1)/2}. \tag{13}$$

Using (13), it can be shown that the average complexity of a back tracking algorithm for the maximum independent set problem is subexponential (because I_n grows at the rate of $O(n^{\log n})$).

In the late 1980's, new algorithms were proposed by Pardalos and Rodgers [204] (which is published in 1992), Gendreau *et al.* [84] [83], and Tomita *et al.* [238]. The algorithm of Pardalos and Rodgers [204] was based on an unconstrained quadratic zero-one programming formulation of the maximum clique problem. In their work, the merit of two different branching rules, *greedy* and *nongreedy*, were tested. Some interesting results concerning the behavior of algorithms using these different rules were reported. The Gendreau *et al.* [84] algorithm was an implicit enumerative algorithm. Its branching rule (the selection of the next vertex to branch) is based on the number of triangles a vertex belongs to. The algorithm of Tomita et al [238] used a greedy coloring algorithm to get an upper bound on the size of the maximum clique. Some computational results could be found in [84] and [238].

In the 1990's, more algorithms were proposed, for example, the algorithms of Pardalos and Phillips [201], Friden et al. [76], Carraghan and Pardalos [43], Babel and Tinhofer [13], Babel [12], Xue [255].

Paradolos and Phillips [201] formulated the maximum clique problem as an indefinite quadratic global optimization problem with linear constrains. They were able to provide some theoretical upper bounds on the size of the maximum clique of a graph (see the paragraph after Theorem 1.6 in section 1). The algorithm of Friden *et al.* [76] was a branch and bound algorithm for the maximum independent set problem. They used the Tabu search technique in finding lower and upper bounds in their algorithm.

Carraghan and Pardalos [43] proposed an implicit enumerative algorithm. Their algorithm is very easy to understand and very efficient for sparse graphs. Their branching rule corresponds to the *nongreedy* rule described in Pardalos and Rodgers [204]. Using this algorithm, they were able to solve problems on graphs of 500 vertices. They have also tested their algorithm on instances of graphs with 1000 and 2000 vertices. Since the algorithm is easy to understand and their actual code is available, it can serve as a benchmark for comparing different algorithms.

Babel and Tinhofer [13] proposed a branch and bound algorithm for the maximum clique problem. The main ingredient of their algorithm is the use of a

fast and relatively good heuristic for the minimum coloring problem proposed by Brelaz [36]. The coloring heuristic is called the *degree of saturation largest first* (DSATUR). Applying DSATUR to a graph, one can find an upper bound on the size of the maximum clique as well as a maximal clique (thus, a lower bound). Babel and Tinhofer exploited this distinct feature and applied DSATUR at each node of the search tree. They tested their algorithm on graphs of 100 to 400 vertices with varying densities. In 1991, Babel [12] further refined and improved the algorithm of Babel and Tinhofer [13].

Also in 1991, based on the fact that a fractional coloring solution provides a tighter upper bound than an integer coloring solution for the maximum clique problem, a heuristic for the fractional coloring problem is proposed in [255] and used in a branch and bound algorithm for the maximum clique problem. Substantial reduction in the search tree size and the improvement in efficiency of the branch and bound algorithm are observed because of the use of this new bounding procedure. Details about the method and how to extend it to the weighted case can also be found in [19].

5. Exact Algorithms for the Weighted Case

Algorithms for finding a maximum weight independent set of an arbitrary graph started in 1975 by Nemhauser and Trotter [186]. They considered the polyhedron relationships between the edge formulation (1) of the maximum weight independent set problem and its linear relaxation problem. Their main results were given as Theorem 1.1 in Section 1 of the present paper. Based on this result, they proposed an algorithm for the maximum weight independent set problem.

In 1977, Balas and Samuelson [16] proposed an algorithm that solved the minimum vertex covering problem (a weighted version was available as mentioned [16]). Their algorithm was based on the relationships between the integer dual feasible solution and an equivalent linear programming for the vertex covering problem. Labeling procedures were designed to generate and improve vertex covers. When these labeling procedures could not continue, branch and bound was used. The computational tests in [186] and [16] were conducted on unweighted graphs of up to 50 vertices.

In 1983, Loukakis and Tsouros [158] proposed an algorithm for the maximum weight independent set problem. It seems that almost nothing else appeared in the literature until the late 1980's and early 1990's. Recently published algorithms we are aware of for the maximum weight clique/independent set problem are due to Pardalos and Desai [199], Balas and Xue [18], and Nemhauser and Sigismondi [184].

The algorithm proposed by Pardalos and Desai [19] was based on an unconstrained quadratic 0–1 formulation of the maximum weight independent set problem. In addition, they established an interesting relationship between the

local minima of the quadratic problem and the maximal independent sets of a graph as follows:

THEOREM 5.1. *x is a discrete local minimum for problem* (10) *if and only if x represents a maximal independent set of G.*

Their algorithm (for maximum weight independent set) used the *nongreedy* search strategy described in Pardalos and Rodgers [204]. With this algorithm they were able to solve problems of up to 500 vertices with different densities. Here we want to mention an unpublished work by Carraghan and Pardalos that was the weighted version of the Carraghan and Pardalos [44] algorithm. As its unweighted version, it is also very simple to understand. Computational tests show that the weighted version of Carraghan and Pardalos algorithm is more efficient than that of Pardalos and Desai [199].

Balas and Xue [18] algorithm extends the algorithm of Balas and Yu [20] to the weighted case. To accomplish this, a minimum weighted coloring of a triangulated graph was needed. Although the minimum weighted coloring problem on triangulated graphs is known to be in class P (see [92]), there was no algorithm in the literature that had reasonable time complexity. Balas and Xue [18] proposed a combinatorial algorithm for this problem with a time complexity of $O(n^2)$. With this algorithm, they extended the Balas and Yu algorithm to the weighted case. Their computational results show that the size of the search tree is greatly reduced and the CPU time is much smaller than other such algorithms, especially for larger, dense graphs. Graphs of size up to 2000 vertices are solved on a workstation.

Recently, Balas and Xue [19] proposed another branch and bound algorithm for the maximum weight clique problem. The algorithm is based on the notion of a weighted fractional coloring of a graph and the fact that such a coloring provides a tighter upper bound than its integer correspondence. In [19], they proposed a fast heuristic for the weighted fractional coloring problem and used this heuristic as an upper bounding procedure in a branch and bound algorithm for the maximum weight clique problem. Comparing with the method in [18], computational results show the reduction in search tree size and the improved efficiency of the resulting algorithm.

The algorithm of Nemhauser and Sigismondi [184] used the polyhedron approach. They tried to solve the problem by first solving the linear relaxation of the corresponding integer programming problem. If the optimal solution to the relaxation problem was integer, it was done. Otherwise, sets of valid inequalities were generated and added into the relaxed problem to cut off the current fractional solution. They restricted themselves to some classes of facet defining inequalities for the maximum independent set problem. Since not all facets for the clique/independent set polytope are known, there was no guarantee that a fractional solution would always be cut off. When this happened, they switched to

branch and bound method. From their computational test, they found too many iterations in solving the linear relaxation problems. The largest graphs they tried to solve had up to 120 vertices.

6. Heuristics

In a branch and bound algorithm for the maximum clique problem, its lower bounding procedure usually provides a maximal clique which can be used to approximate the maximum clique of G. Since different branch and bound algorithms tend to have different bounding procedures, they provide us different heuristics for the maximum clique problem. On the other hand, there are many heuristics in the literature designed to find an approximate solution to the maximum clique problem. These heuristics are usually more complicated than the lower bounding procedures from a branch and bound algorithm. The reason is that they will be run only once.

The majority of approximation algorithms in the literature for the maximum clique problem are called *sequential greedy heuristics*. These heuristics generate a maximal clique through the repeated addition of a vertex into a partial clique, or the repeated deletion of a vertex from a set that is not a clique.

Kopf and Ruhe [144] named these two classes of heuristics the *Best in* and the *Worst out* heuristics. Decisions on which vertex to be *added in* or *moved out* next are based on certain indicators associated with candidate vertices. For example, a possible *Best in* heuristic constructs a maximal clique by repeatedly *adding in* a vertex that has the largest degree among candidate vertices. In this case, the indicator is the degree of a vertex. On the other hand, a possible *Worst out* heuristic can start with the whole vertex set V. It will repeatedly remove a vertex out of V until V becomes a clique.

Kopf and Ruhe [144] further divided the above two classes of heuristics into *New* and *Old* (*Best in* or *Worst out*) heuristics. Namely, if the indicators are updated every time a vertex is added in or moved out, then the heuristic is called a *New* heuristic. Otherwise it is called an *Old* heuristic. We can find in the literature that most heuristics for the maximum clique problem fall in one or the other classes. See for example, the approximation algorithm of Johnson [127], and the approximation algorithm of Tomita *et al.* [239]. The differences among these heuristics are their choice of indicators and how indicators are updated. A heuristic of this type can run very fast.

A common feature of the sequential heuristics is that they all find only one maximal clique. Once a maximal clique is found, the search stops. We can view this type of heuristics from a different point of view. Let us define S_G to be the space consisting of all the maximal cliques of G. What a sequential greedy heuristic does is to find one point in S_G, hoping it is (close to) the optimal point. This suggests to us a possible way to improve our approximation solutions,

namely, expand the search in S_G. For example, once we find a point $x \in S_G$, we can search its neighbors to improve x. This leads to the class of the *local search heuristics*.

In a local search heuristic, the more neighbors of $x \in S_G$ we search, the greater chance of finding a better solution. Depending on the neighborhood definition of a point $x \in S_G$, and how the search is performed, different local search heuristics result. A well known class of local search heuristics in the literature is the *k-interchange* heuristics. They are based on the *k-neighbor* of a feasible solution. In the case of the maximum clique problem, a k-neighbor of $x \in S_G$ is defined as follows. Let $y \in S_G$, y is a k-neighbor of x if $|x - y| \leq k$, where $k \leq |x|$. A k-interchange heuristic first finds a maximal clique $x \in S_G$. Then it searches all the k-neighbors of x and outputs the best clique found. As one will expect, the main factors for the complexity of this class of heuristics are the size of the neighborhood and the searches involved. For example, in the k-interchange heuristic, the complexity grows roughly with $O(n^k)$.

The solution quality of a local search heuristic directly depends on the starting point $x \in S_G$ and the neighborhood of x. To improve the quality of its solution, we need to increase the neighborhood of x (the starting point) to include a "better" point. If we want to look at various points spread over S_G, we need to have a very large neighborhood. The problem is when the size of the neighborhood increases, the search effort increases so rapidly that one could not afford it.

A class of heuristics designed to search various points of S_G is called the *randomized heuristics*. The main ingredient of this class of heuristics is the part that finds a random point in S_G. A possible to way to do that is to include some random factors in the generation of a point of S_G. A randomized heuristic runs a heuristic (with random factors included) a number of times to find different points over S_G. For example, we can randomize a sequential greedy heuristic and let it run N times. The complexity of a randomized heuristic depends on the complexity of the heuristic and the number N.

An elaborated implementation of the randomized heuristic for the maximum independent set problem can be found in Feo *et al.* [72] where local search is combined with randomized heuristic. Their computational results indicated that their approach was effective in finding large cliques of randomly generated graphs. For example, for randomly generated graphs with 1000 vertices and 50% density, their approach found cliques of size 15 or larger in most cases. Here, 15 is a bound derived from the probabilistic analysis of this class of graphs (see [32], [31], [77]). A different implementation of a randomized algorithm for the maximum independent set problem can be found in [4].

Recently, Tabu search and Neural Networks have also been used to find an approximate solution for the maximum clique problem. For example, Friden *et al.* [75] proposed a heuristic based on Tabu search technique. A different implementation of Tabu search for the same problem was proposed by Gendreau *et*

al. [84]. Ramanujam and Sadayappan [215], Jagota [124], and Jagota and Regan [125] proposed heuristics based on Neural Networks. What these methods accomplished is to search various points of S_G. However, since there are too many parameters affecting the search pattern (and the result), it is hard to tell what points in S_G are being searched. Some computational results from these methods are encouraging.

Another type of heuristics that finds a maximal clique of G is called the *subgraph approach* (see [20]). It is based on the fact that a maximum clique C of a subgraph $G' \subseteq G$ is also a clique of G. The subgraph approach first finds a subgraph $G' \subseteq G$ such that the maximum clique of G' can be found in polynomial time. Then it finds a maximum clique of G' and uses it as the approximation solution. The advantage of this approach is that in finding the maximum clique $C \subseteq G'$, one has (implicitly) searched many other cliques of G' ($S_{G'} \subseteq S_G$). Because of the special structure of G', this implicit search can be done efficiently. In Balas and Yu [20], G' is a maximal induced triangulated subgraph of G. Since many classes of graphs have polynomial algorithms for the maximum clique problem, the same idea also applies there. For example, the class of edge-maximal triangulated subgraphs was chosen in [14], [255], and [256]. Some of the greedy heuristics, randomized heuristics and subgraph approach heuristics are compared in [255] and [256] on randomly generated weighted and unweighted graphs.

Additional heuristics for the maximum clique/independent set and related problems on arbitrary or special class of graphs can be found in [48], [51], [54], [73], [233].

7. Applications

Practical applications of the maximum clique and related problems include project selection, classification theory, fault tolerance, coding theory, computer vision, economics, information retrieval, signal transmission theory, aligning DNA and protein sequences, and other specific problems. Some of these applications can be found in [10], [22], [24], [28], [50], [60], [150], [159], [163], [176], [216], [228], [247], [248] and [254].

Next we provide some very interesting applications of the maximum clique problem. In addition, some problems arising in these applications can be used as test problems for algorithm comparison and correctness.

We start with an application in computing binary (constant weight) codes [38]. Let $A(n, d, w)$ be the maximal number of binary vectors of length n, with Hamming distance at least d apart, and constant weight w. Also let $A(n, d)$ be the maximal possible number of binary vectors of length n and Hamming distance at least d apart (with no restriction on weight).

To compute $A(n, d)$ or $A(n, d, w)$ one forms the graph with 2^n or $\binom{n}{w}$ vertices, corresponding to all possible code-words, joins two vertices by an edge if their Hamming distance is at least d, and finds the maximum clique. Similarly, finding

the largest code invariant under a given permutation group, requires finding the maximum clique in a graph with weights attached to the vertices.

The exact values of $A(n, d, w)$ are known for all $n \leq 11$ (the first undetermined values being $80 \leq A(12, 4, 5) \leq 84$). Similarly, for d even, $A(n, d)$ is known exactly for $n \leq 10$ (the first undetermined values being $72 \leq A(11, 4) \leq 79$). For more details and other references see the paper by Brouwer *et al.* [38].

The next application occurs in geometry [231]. Minkowski conjectured that all extremal lattices for the supermum norm were of a certain simple form, and observed that this conjecture had a simple geometric interpretation, that is, in any tiling of R^n with unit n-cubes, there must exist two cubes having a complete facet in common. A family of cubes whose interiors are disjoint and whose union is R^n is called a tiling. If the centers of the cubes form a lattice, then we have a lattice tiling. Although Minkowski's conjecture was proved in 1942, Keller generalized it to conjecture that any tiling in R^n by unit n-cubes contains two cubes having a complete facet in common. Corradi and Szabo [56] proved that there is a counterexample to this conjecture if and only if the following graphs Γ_n (of 4^n vertices) has a 2^n size maximum clique. The 4^n vertices of Γ^n are n-tuples of integers 0, 1, 2 and 3. A pair of these n-tuples are adjacent if there is a position at which the corresponding components are 2 modulo 4 and if there is a further position at which the corresponding components are different. The conjecture was proved by Peron [209] to be true for $n \leq 6$ and was proved by Lagarias and Shor [147] to be false for $n \geq 10$. The conjecture remains open for the values of n equal to 7, 8, 9.

Another interesting application appears in the study of assignment polytopes (see the paper by S. Onn, [189]). In particular, consider the assignment polytope $P(n-1, 1)$. The 1-skeleton of P is the graph of the set of vertices of P in which the edges are the 1-faces of P. Onn proved that large independent sets in the 1-skeleton of P are exhibited, by proving that its stability number is $2^{\Omega(\sqrt{n}\log n)}$. More details on these applications and computational results with exact algorithms have appeared in [111].

8. Concluding Remarks

While a variety of algorithms and heuristics have been proposed for the solution of the maximum clique problem, only a few of the suggested algorithms have been programmed and tested on graphs where the problem is difficult to solve. Extensive computational results are needed to evaluate the average performance of the algorithms, not only on randomly generated graphs but also on problems from a diverse spectrum of applications.

In this bibliographic survey, we have attempted to briefly summarize the main ideas in each of the proposed algorithms. Many applications and algorithms on problems related to the maximum clique problem can be found in the list of references below.

An algorithm implementation challenge is taking place during 1993 at DIMACS (Center for Discrete Mathematics and Theoretical Computer Science), Rutgers University, New Jersey. One of the three topics of the Challenge is the maximum clique problem. The proceedings of this implementation challenge will be published in a DIMACS volume by the American Mathematical Society [128].

References

[1] R. Aharoni, P. Erdös, and N. Linial (1988), Optima of Dual Integer Linear Programs, *Combinatorica* **8** (1), 13–20.

[2] E. A. Akkoyunlu (1973), The Enumeration of Maximal Cliques of Large Graphs, *SIAM Journal on Computing* **2**, 1–6.

[3] M. O. Albertson and J. P. Hutchinson (1978), On the Independence Ratio of a Graph, *J. Graph Theory* **2**, 1–8.

[4] N. Alon, L. Babai, and A. Itai (1986), A Fast and Simple Randomized Parallel Algorithm for the Maximal Independent Set Problem, *J. Algorithms* **7**, 567–583.

[5] A. T. Amin and S. L. Hakimi (1972), Upper Bounds on the Order of a Clique of a Graph, *SIAM J. of Appl. Math.* **22** (4), 569–573.

[6] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy (1992), On the Intractability of Approximation Problems, Preliminary Draft, University of Rochester, NY.

[7] S. Arora and S. Safra (1992), Approximating the Maximum Clique is *NP*-Complete, Preliminary Draft, University of Rochester, NY.

[8] J. G. Auguston and J. Minker (1970), An Analysis of Some Graph Theoretical Cluster Techniques, *J. Assoc. Comput. Mach.* **17**, 571–588.

[9] G. Ausiello, A. D'Atr, and M. Protasi (1980), Structure Preserving Reductions among Convex Optimization Problems, *J. Comput. and Syst. Sci.* **21**, 136–153.

[10] G. Avondo-Bodeno (1962), *Economic Applications of the Theory of Graphs*, Gordon and Breach, Science Publishers, New York.

[11] S. M. Baas, M. C. Bonvanie, and A. J. Verschoor (1988) A Relaxation Method for the Set Packing Problem Using Polyhedron Characteristic, Technical Report 699, University of Twente, Netherlands.

[12] L. Babel (1991), Finding Maximum Cliques in Arbitrary and in Special Graphs, *Computing* **46**, 321–341.

[13] L. Babel and G. Tinhofer (1990), A Branch and Bound Algorithm for the Maximum Clique Problem, *ZOR-Methods and Models of Operations Research*, **34**, 207–217.

[14] E. Balas (1986), A Fast Algorithm for Finding an Edge-Maximal Subgraph with a TR-Formative Coloring, *Discr. Appl. Math.* **15**, 123–134.

[15] E. Balas, V. Chvátal, and J. Nešetril (1987), On the Maximum Weight Clique Problem, *Math. of Operations Research* **12** (3), 522–535.

[16] E. Balas and H. Samuelsson (1977), A Node Covering Algorithm, *Naval Research Logistics Quarterly* **24**, 213–233.

[17] E. Balas and P. Toth (1985), Branch and Bound Methods, in: Lawler, Lenstra, Rinnooy Kan, and Shmoys (eds.), *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*, Wiley: 361–403.

[18] E. Balas and J. Xue (1991), Minimum Weighted Coloring of Triangulated Graphs, with Application to Maximum Weight Vertex Packing and Clique Finding in Arbitrary Graphs, *SIAM J. Comput.* **2**, 209–221; "Addendum," *SIAM J. Comput.* **21**: 1000, 1992.

[19] E. Balas and J. Xue (1993), Weighted and Unweighted Maximum Clique Algorithms with Upper Bounds from Fractional Coloring, Management Science Research Report 590, GSIA, Carnegie Mellon University, April 1993.

[20] E. Balas and C. S. Yu (1986), Finding a Maximum Clique in an Arbitrary Graph, *SIAM J. Computing* **14** (4), 1054–1068.

[21] E. Balas and C. S. Yu (1989), On Graphs with Polynomially Solvable Maximum-Weight Clique Problem, *Networks*, **19**, 247–253.
[22] D. H. Ballard and M. Brown, (1982), *Computer Vision*, Prentice-Hall, Englewood Cliffs, N.J.
[23] R. Bar-Yehuda and S. Even (1985), A Local-Ratio Theorem for Approximating the Weighted Vertex Cover Problem, *Annals of Discrete Mathemtics* **25**, 27–46.
[24] F. Barahona, A. Weintraub, and R. Epstein (1992), Habitat Dispersion in Forest Planning and the Stable Set Problem, *Operations Research* **40**, Supp. 1: S14–S21.
[25] A. R. Bednarek and O. E. Taulbee, (1966), On Maximal Chains, *Roum. Math. Pres et Appl.* **11**, 23–25.
[26] L. W. Beineke (1969), A Survey of Packings and Coverings in Graphs in the Many Facets of Graph Theory, G. Chartrand and S. Kapoor (eds.), Springer-Verlag: 45–53.
[27] C. Berge and V. Chvátal (eds.), (1984), Topics on Perfect Graphs, *Annals of Discrete Mathematics*, **21**.
[28] P. Berman and A. Pelc (1990), Distribution Fault Diagnosis for Multiprocessor Systems, *Proc. of the 20th Annual Intern. Symp. on Fault-Tolerant Computing* (Newcastle, UK): 340–346.
[29] P. Berman and G. Schnitger (1989), On the Complexity of Approximating the Independent Set Problem, In *Proceedings of the 1989 Symposium on Theoret. Aspects of Comp. Sci.*, Springer-Verlag, Lecture Notes in Computer Sciences 349: 256–268.
[30] A. Billionnet (1981), An Upper Bound on the Size of the Largest Cliques in a Graph, *J. Graph Theory*, **5**, 165–169.
[31] B. Bollobás and P. Erdös (1976), Cliques in Random Graphs, *Math. Proc. Cambridge Philos. Soc.* **80**, 419–427.
[32] B. Bollobás (1985), *Random Graphs*, Academic Press, NY.
[33] R. E. Bonner (1964), On Some Clustering Techniques, *IBM J. Res. Develop.* **8** (1), 22–32.
[34] R. Boppana and M. M. Halldórsson (1990), Approximating Maximum Independent Sets by Excluding Subgraphs, *Proc. SWAT 90*, Lecture Notes in Computer Sciences: 11–25, Springer.
[35] M. Boulala and J. P. Uhry (1979), Polytope des Independants dans un Graphe Serie-Paralle, *Discrete Mathematics*, **27**, 225–243.
[36] D. Brelaz (1979), New Methods to Color the Vertices of a Graph, *Comm. of ACM* **22** (4), 251–256.
[37] C. Bron and J. Kerbosch (1973), Algorithm 457: Finding All Cliques of an Undirected Graph, *Comm. of ACM*, **16**, 575–577.
[38] A. E. Brouwer, J. B. Shearer, N. J. A. Sloane, and W. D. Smith, (1990), A New Table of Constant Weight Codes, *J. IEEE Trans. Information Theory* **36**, 1334–1380.
[39] M. A. Buckingham (1981), Efficient Stable Set and Clique Finding Algorithms for Overlap Graphs, Technical Report, Dept. Computer Science, New York Univ.
[40] M. Burlet and J. Fonlupt (1984), Polynomial Algorithm to Recognize a Meyniel Graph, in W. R. Pulleyblank (ed.), *Progress in Combinatorial Optimization*, Academic Press, Toronto: 69–99.
[41] J. E. Burns (1989), The Maximum Independent Set Problem for Cubic Planar Graph, *Networks* **9**, 373–378.
[42] L. Butz, P. L. Hammer, and D. Hausmann (1975), Reduction Methods for the Vertex Packing Problem, Technical Report 7540, Univ. of Bonn, Institut für Ekonometric und Operations Research.
[43] R. Carraghan and P. M. Pardalos (1990), An Exact Algorithm for the Maximum Clique Problem, *Operations Research Letters* **9**, 375–382.
[44] R. Carraghan and P. M. Pardalos (1990), A Parallel Algorithm for the Maximum Weight Clique Problem, Technical Report CS-90-40, Dept. of Computer Science, Penn. State Univ.
[45] R. C. Chang and H. S. Lee (1992), Finding a Maximum Set of Independent Chords in a Circle, *Info. Proc. Letters* **41**, 99–102.
[46] N. Chiba and T. Nishizeki (1985), Arboricity and Subgraph Listing Algorithms, *SIAM J. Comput.* **14**, 210–223.
[47] N. Chiba, T. Nishizeki, and N. Saito (1982), An Approximation Algorithm for the Maximum Independent Set Problem on Planar Graphs, *SIAM. J. Comput.* **11** (4), 663–675.
[48] N. Chiba, T. Nishizeki, and N. Saito (1983), An Algorithm for Finding a Large Independent Set in Planar Graphs, *Networks*, **13**, 247–252.

[49] E. Choukhmane and J. Franco (1986), An Approximation Algorithm for the Maximum Independent Set Problem in Cubic Planar Graphs, *Networks* **16**, 349–356.
[50] N. Christofides (1975), *Graph theory, An algorithmic Approach*, Academic Press.
[51] M. Chrobak and J. Naor (1991), An Efficient Parallel Algorithm for Computing a Large Independent Set in a Planar Graph, *Algorithmica*, **6**, 801–815.
[52] V. Chvátal (1975), On Certain Polytopes Associated with Graphs, *J. of Combin. Theory B* **18**, 138–154.
[53] V. Chvátal (1977), Determining the Stability Number of a Graph, *SIAM J. on Comput.* **6**, 643–662.
[54] V. Chvátal (1979), A Greedy Heuristic for the Set-Covering Problem, *Math. Oper. Res.* **4**, 233–243.
[55] V. Chvátal (1985), Perfectly Orderable Graphs, *Annals of Discrete Math.* **21**, 63–65.
[56] K. Corradi and S. Szabo (1990), A Combinatorial Approach for Keller's Conjecture, *Periodica Mathematica Hungarica* **21** (2), 95–100.
[57] P. Crescenzi, C. Fiorini, and R. Silvestri (1991), A Note on the Approximation of the MAX CLIQUE Problem, *Infor. Proces. Letters* **40**, 1–5.
[58] E. Dahlhaus and M. Karpinski (1987), A Fast Parallel Algorithms for Computing All Maximal Cliques in a Graph and the Related Problems, Technical Report 8516-CS, Institut für Informatik, Universität Bonn, Römerstrasse 164, 5300 Bonn 1.
[59] V. Degot and J. M. Hualde (1975), De l'utilisation de la Notion de Clique en Matière de Typologie des Populations, R.A.I.R.O., 1.
[60] N. Deo (1974), *Graph Theory with Applications to Engineering and Computer Science*, Prentice-Hall, Engelwood Cliffs, NJ.
[61] J. F. Desler and S. L. Hakimi (1970), On Finding a Maximum Internally Stable Set on a Graph, *Proc. of Fourth Annual Princeton Conference on Information Sciences and Systems* **4**, 459–462, Princeton, NJ.
[62] G. A. Dirac (1952), A Property of 4-Chromatic Graphs and Some Remarks on Critical Graphs, *J. London Math. Society* **27**, 85–92.
[63] G. A. Dirac (1961), On Rigid Circuit Graphs, *Abh. Math. Sem., Univ. Hamburg*, **25**, 71–76.
[64] P. Doreian (1969), A Note on the Detection of Cliques in Valued Graphs, *Sociometry* **32**, 237–242.
[65] C. Ebenegger, P. L. Hammer, and D. de Werra, (1984), Pseudo-Boolean Functions and Stability of Graphs, *Annals of Discrete Mathematics* **19**, 83–98.
[66] J. Edmonds (1962), Covers and Packings in a Family of Sets, *Am. Math. Soc. Bull.* **68**, 494–497.
[67] C. S. Edwards and C. H. Elphick (1983), Lower Bounds for the Clique and the Chromatic Numbers of a Graph, *Disc. Math.* **5**, 51–64.
[68] P. Erdös and M. Erne (1986), Clique Numbers of Graphs, *Discrete Mathematics* **59**, 235–242.
[69] F. Escalante (1973), Über Iterierte Clique-Graphen, *Abh. Math. Sem. Univ. Hamburg* **39**, 59–68.
[70] S. Fajtlowicz (1978), On the Size of Independent Sets in Graphs, *Proc. 9th S.E. Conf. on Combin., Graph Theory and Comput.*: 269–274, Boca Raton.
[71] U. Feige, S. Goldwasser, L. Lovász, S. Safra, and M. Szegedy (1991), Approximating the Maximum Clique is Almost *NP*-Complete, *Proc. 32nd IEEE Symp. on Foundations of Computer Science*: 2–12.
[72] T. A. Feo, M. G. C. Resende, and S. H. Smith (1994), A Greedy Randomized Adaptive Search Procedure for Maximum Independent Set, *Operations Research*, to appear.
[73] M. L. Fisher and L. A. Wolsey (1982), On the Greedy Heuristic for Continuous Covering and Packing Problems, *SIAM J. Alg. Disc. Meth.* **3**, 584–591.
[74] A. Frank (1976), Some Polynomial Algorithms for Certain Graphs and Hypergraphs, *Proc. 5th British Combin. Conf.*: 211–226.
[75] C. Friden, A. Hertz, and D. de Werra (1989), Stabulus: A Technique for Finding Stable Sets in Large Graphs with Tabu Search, *Computing* **42**, 35–4.
[76] C. Friden, A. Hertz, and M. de Werra (1990), TABARIS: An Exact Algorithm Based on Tabu Search For Finding a Maximum Independent Set in a Graph, *Computers and Operations Research* **17** (5), 437–445.

[77] A. Frieze (1990), On the Independence Number of Random Graphs, *Discrete Math.* **81**, 171–175.
[78] Z. Füredi (1987), The Number of Maximal Independent Sets in Connected Graphs, *J. Graph Theory* **11**, 463–470.
[79] M. Garey and D. Johnson (1979), *Computers and Intractability, A Guide to The Theory of NP-Completeness*, Freeman, San Francisco.
[80] F. Gavrill (1992), Algorithms for Minimum Colouring, Maximum Clique, Minimum Covering by Cliques, and Maximum Independent Set of a Chordal Graph, *SIAM J. Comput.* **1**, 180–187.
[81] F. Gavril (1973), Algorithms for a Maximum Clique and a Maximum Independent Set of a Circle Graph, *Networks* **3**, 261–273.
[82] F. Gavril (1974), Algorithms on Circular Arc Graphs, *Networks* **4**, 357–369.
[83] M. Gendreau, J. C. Picard, and L. Zubieta (1988), An Efficient Implicit Enumeration Algorithm for the Maximum Clique Problem, A. Kurzhanski *et al.* (eds), *Lecture Notes in Economics and Mathematical Systems*, **304**, 70–91, Springer-Verlag.
[84] A. Gendreau, L. Salvail, and P. Soriano (1989), Solving the Maximum Clique Problem Using a Tabu Search Approach, Technical Report 675, Center for Research on Transportation, University of Montreal, Montreal, Canada.
[85] L. Gerhards and W. Lindenberg (1979). Clique Detection for Nondirected Graphs: Two New Algorithms, *Computing* **21**, 295–322.
[86] A. M. H. Gerards and A. Schrijver (1986), Matrices with the Edmonds–Johnson Property, *Combinatorica* **6**, 403–417.
[87] L. E. Gibbons, D. W. Hearn, and P. M. Pardalos (1993), A Continuous Optimization Approach for Finding Large Cliques in Dense Graphs, Working Paper, University of Florida.
[88] R. Giles and L. R. Trotter (1981), On Stable Set Polyhedra for $K_{1,3}$ Free Graphs, *J. Combin. Theory B* **31**, 313–326.
[89] M. Goldberg and T. Spencer (1989), A New Parallel Algorithm for the Maximal Independent Set Problem, *SIAM J. Comput.* **18** (2), 419–427.
[90] M. Goldberg and T. Spencer (1989), Constructing a Maximal Independent Set in Parallel, *SIAM J. Disc. Math.* **2** (3), 322–328.
[91] M. Goldman and J. Hastad (1992), A Simple Lower Bound for Monotone Clique Using a Communication Game, *Info. Proc. Letters* 41, 221–226.
[92] M. C. Golumbic (1980), *Algorithmic Graph Theory and Perfect Graphs*, Academic Press, New York.
[93] M. C. Golumbic and P. L. Hammer (1988), Stability in Circular-Arc-Graphs, *J. Algorithm* **9**, 314–320.
[94] J. R. Giggs (1983), Lower Bounds on the Independence Number in Terms of the Degrees, *J. Combin. Theory B* **34**, 22–39.
[95] J. R. Griggs, M. Grinstead, and D. Guichard (1988), The Maximum Number of Maximal Independent Sets in a Connected Graph, *Discrete Mathematics* **68**, 211–220.
[96] G. R. Grimmett and W. R. Pulleyblank (1985). Random Near-Regular Graphs and the Node Packing Problem, *Operations Research Letters* 4, 169–174.
[97] M. Grötschel, L. Lovász, and A. Schrijver (1987), The Ellipsoid Method and Its Consequences in Combinatorial Optimization, *Combinatorica* **1**, 169–197.
[98] M. Grötschel, L. Lovász, and A. Schrijver (1986), Relaxations of Vertex Packing, *J. Combin. Theory B* **40**, 330–343.
[99] M. Grötschel, L. Lovász, and A. Schrijver (1988), *Geometric Algorithms and Combinatorial Optimization*, Springer-Verlag.
[100] M. Grötschel, L. Lovász, and A. Schrijver (1989), Polynomial Algorithms for Perfect Graphs, *Annals of Discrete Mathematics* **21**, 325–356.
[101] U. I. Gupta, D. T. Lee, and J. Y. T. Leung (1982), Efficient Algorithms for Interval Graphs and Circular-Arc Graphs, *Networks* **12**, 459–467.
[102] Y. Gurevich and S. Shelah (1984), Expected Computation Time for Hamiltonian Path Problem and Clique Problem, Technical Report, Computing Research Laboratory, The Univ. of Michigan.
[103] W. W. Hager, P. M. Pardalos, I. M. Roussos, and H. D. Sahinoglou (1991), Active Constraints,

Indefinite Quadratic Programming and Test Problems, *Journal of Optimization Theory and Applications*, **68** (3), 499–511.

[104] S. L. Hakimi and H. Frank (1969), Maximum Internally Stable Sets of a Graph, *J. of Math. Anal. and Appl.* **25**, 296–308.

[105] P. L. Hammer, P. Hansen, and B. Simeone (1980). Vertices Belonging to All or to No Maximum Stable Sets of a Graph, FUCAM Research Report.

[106] P. Hansen (1975), Degrés et Nombre de Stabilité d'un Graph, *Cahiers du CERP* **17**, 213–220.

[107] P. Hansen (1979), Upper Bounds for the Stability Number of a Graph, *Revue Roumaine de Math. Pures et Appliquees* **24**, 1195–1199.

[108] P. Hansen (1980), Bornes et Algorithmes Pour les Stable d'un Graphe, In *Regards sur la Theory des Graph* (eds. P. Hansen and D. de Werra), Presses Polytechniques Romandes.

[109] P. Hansen and N. Mladenovic (1992), Two Algorithms for Maximum Cliques in Dense Graphs, Technical Report, Ecole des Hautes Etudes Commerciales, Montreal, Canada.

[110] F. Harary and I. C. Ross (1957). A Procedure for Clique Detection Using the Group Matrix, *Sociometry* 20, 205–215.

[111] J. Hasselberg, P. M. Pardalos, and G. Vairaktarakis (1993), Test Case Generators and Computational Results for the Maximum Clique Problem, *Journal of Global Optimization* **3**, 463–482.

[112] R. B. Hayward (1985), Weakly Triangulated Graphs, *J. of Combin. Theory B* **39**, 200–209.

[113] R. Hayward, C. T. Hoang, and F. Maffray (1989), Optimizing Weakly Triangulated Graphs, *Graphs and Combinatorics* **5**, 339–349. See also *Erratum* **6**, 33–35, 1990.

[114] B. Hedman (1985), The Maximum Number of Cliques in Dense Graphs, *Discrete Mathematics* **54**, 161–166.

[115] I. Heller (1957), On Linear Systems with Integral Valued Solutions, *Pacific J. of Math.* **7**, 1351–1364.

[116] C. W. Ho and R. C. T. Lee (1988), Efficient Parallel Algorithms for Finding Maximal Cliques, Clique Trees, and Minimum Coloring of Chordal Graphs, *Inform. Process. Letter* **28**, 301–309.

[117] C. Hoded (1988), Hard Graphs for the Maximum Clique Problem, *Discrete Mathematics*, **72**, 175–179.

[118] D. J. Houck (1974), On the Vertex Packing Problem, Ph.D. dissertation, The Johns Hopkins University, Baltimore.

[119] D. J. Houck and R. R. Vemuganti (1977). An Algorithm for the Vertex Packing Problem, *Operations Research* **25**, 773–787.

[120] W. L. Hsu (1985), Maximum Weight Clique Algorithms for Circular-Arc Graphs and Circle Graphs, *SIAM J. Comput.*, **14** (1) 224–231.

[121] W. L. Hsu (1988), The Coloring and Maximum Independent Set Problems on Planar Perfect Graphs, *J. ACM* **35** (3), 535–563.

[122] W. Hsu, Y. Ikura, and G. L. Nemhauser (1981), A Polynomial Algorithm for Maximum Weighted Vertex Packings on Graphs without Long Odd Cycles, *Math. Programming* **20**, 225–232.

[123] C. H. Hubbell (1965), An Input-Output Approach to Clique Identification, *Sociometry* **28**, 377–399.

[124] A. Jagota (1992), Efficiently Approximating Max-Clique in a Hopfield-style Network, In *Inter. Joint Conf. on Neural Networks* **2**, 248–253, NY.

[125] A. Jagota and K. W. Regan (1992). Performance of Max-Clique Heuristics under Description – Length Weighted Distributions, Technical Report 92–24, Dept. of Comput. Sci, SUNY Buffalo, NY.

[126] M. Jerrum (1990), The Elusiveness of Large Cliques in a Random Graph, Technical Report, Dept. of Comput. Sci. Univ. of Edinburgh.

[127] D. S. Johnson (1974), Approximation Algorithms for Combinatorial Problems, *J. Computer. and System Sciences* **9**, 256–278.

[128] D. S. Johnson and M. A. Trick (eds.), (1994), Clique, Graph Coloring, and Satisfiability: Second DIMACS Implementation Challenge, *DIMACS Series, American Mathematical Society*, to appear.

[129] D. S. Johnson, M. Yannakakis, and C. H. Papadimitriou (1988), On Generating All Maximal Independent Sets, *Information Processing Letters* **27**, 119–123.

[130] E. L. Johnson and M. W. Padberg (1983), Degree-Two Inequalities, Clique Facets, and Bipartite Graphs, *Annals of Discrete Mathematics* **16**, 169–188.
[131] H. C. Johnston (1976). Cliques of a Graph: Variations on the Bron–Kerbosch Algorithm, *Internat. J. Comput. and Inform Sci.* **5**, 209–238.
[132] L. F. Johnson (1975). Determining Cliques of a Graph, *Proc. of the fifth Manitoba Conf. on Num. Math.*: 429–437.
[133] V. Kann (1992), On the Approximability of the Maximum Common Sub-graph Problem, submitted to STACS.
[134] N. Karmarkar (1990), An Interior Point Approach to *NP*-Complete Problems – Part I, *Contemporary Mathematics* **114**, 297–308.
[135] N. Karmarkar, M. G. C. Resende, and K. G. Ramakrishnan (1989), An Interior Point Approach to the Maximum Independent Set Problem in Dense Random Graphs, IX Intern. Conf. of the Chilean Comp. Sci. Society and XV Latin American Conf. on Informatics, Santiago, Chile.
[136] R. M. Karp and A. Wigderson (1985), A Fast Parallel Algorithm for the Maximal Independent Set Problem, *J. ACM* **32** (4).
[137] T. Kashiwabara and T. Fujisawa (1979), *NP*-Completeness of the Problem of Finding a Minimum-Clique-Number Interval Graph Containing a Given Graph as a Subgraph, *Proceedings International Conference on Circuits and Systems*: 657–660.
[138] T. Kashiwabara, S. Masuda, K. Nakajima, and T. Fujisawa (1992), Generation of Maximum Independent Sets of a Bipartite Graph and Maximum Cliques of a Circular-Arc-Graph, *Journal of Algorithms* **13**, 161–174.
[139] B. Khoury and P. M. Pardalos (1993), An Algorithm for Finding the Maximum Clique on an Arbitrary Graph, Working Paper, University of Florida.
[140] P. Kikusts (1986), Another Algorithm Determining the Independence Number of a Graph, *Elektron. Inf. Verarb. Kybern.* ELK 22: 157–166.
[141] P. N. Klein (1988), Efficient Parallel Algorithms for Chordal Graphs, *Proc. 29th Annual Symp. on Foundations of Computer Science*, IEEE: 150–161.
[142] P. G. Kolaitis and M. N. Thakur (1991), Approximation properties of *NP* Minimization Problems, *Proc. of 6th Annual Conf. on Structures in Computer Science*: 353–366.
[143] R. Kopf (1987), Numerical Methods for Determining Independent Sets in Graphs (in German), Ph.D. Thesis, TH Leipzig.
[144] R. Kopf and G. Ruhe (1987), A Computational Study of the Weighted Independent Set Problem for General Graphs, *Foundations of Control Engineering* **12** (4), 167–180.
[145] S. M. Korman (1979), The Graph-Colouring Problem, in *Combinatorial Optimization* (N. Christofides, P. Toth, and C. Sandi, eds.), John Wiley & Sons, New York: 211–235.
[146] M. Kubal and B. Jackowski (1985), A Generalized Implicit Enumeration Algorithm for Graph Coloring, *Commun. ACM* **28** (4), 412–418.
[147] J. C. Lagarias and P. W. Shor (1992), Keller's Cube-Tilling Conjecture Is False in High Dimensions, Manuscript, Bell Laboratories.
[148] E. L. Lawler (1976), *Combinatorial Optimization*: *Networks and Matroids*, Holt, Rinehart and Winston.
[149] E. L. Lawler, J. K. Lenstra, and H. G. R. Kan (1980), Generating All Maximal Independent Sets: *NP*-Hardness and Polynomial-Time Algorithms, *SIAM J. Comput.* **9**, 558–565.
[150] Lecky, Murphy and Absher (1989), Graph Theoretic Algorithms for the PLA Folding Problem, *IEEE Trans on Computer-Aided Design* **8** (9), 1014–1021.
[151] L. J. Leifman (1976), On Construction of All Maximal Complete Subgraphs (Cliques) of a Graph, Technical Report, Dept. of Math., Univ. Haifa, Israel.
[152] N. Linial (1986), Hard Enumeration Problems in Geometry and Combinatorics, *SIAM J. Alg. Disc. Math.*, **7** (2), 331–335.
[153] R. Lipton and R. Tarjan (1979), A Separator Theorem for Planar Graphs, *SIAM J. Appl. Math.* **36**, 346–358.
[154] R. Lipton and R. Tarjan (1980), Applications of a Planar Separator Theorem, *SIAM J. on Comput.* **9** (3), 615–626.
[155] E. Loukakis (1983), A New Backtracking Algorithm for Generating the Family of Maximal Independent Sets of a Graph, *Comp. and Maths. with Appls.* **9** (4), 583–589.

[156] E. Loukakis and C. Tsouros (1981), A Depth First Search Algorithm to Generate the Family of Maximal Independent Sets of a Graph Lexicographically, *Computing* **27**, 249–266.
[157] E. Loukakis and C. Tsouros (1982), Determining the Number of Internal Stability of a Graph, *Intern. J. Comp. Math.* **11**, 207–220.
[158] E. Loukakis and C. Tsouros (1983), An Algorithm for the Maximum Internally Stable Set in a Weighted Graph, *Int. J. Comput. Math.* **13**, 117–12.
[159] L. Lovász (1979), On the Shannon Capacity of a Graph, *IEEE Transactions on Information Theory* **25**, 1–7.
[160] L. Lovász (1986), *An Algorithmic Theory of Numbers, Graphs and Convexity* SIAM, Philadelphia.
[161] M. Luby (1986), A Simple Parallel Algorithm for the Maximum Independent Set Problem, *SIAM J. Comput.* **15** (4), 1036–1053.
[162] R. D. Luce (1950), Connectivity and Generalized Cliques in Sociometric Group Structure, *Psychometrika* **15**, 169–190.
[163] J. MacWilliams and N. J. A. Sloane (1979), *The Theory of Error Correcting Codes* (Elsevier, American ed.), Amsterdam, North-Holland.
[164] K. Maghout (1959), Sur la Determination des Nombres de Stabilité et du Nombre Chromatique d'un Graphe, *C.R. Acad. Sci., Paris* **248**, 2522–2523.
[165] P. M. Marcus (1964), Derivation of Maximal Compatibles Using Boolean Algebra, *IBM J. Res. Develop.* **8**, 537–538.
[166] S. Masuda and K. Nakajima (1988), An Optimal Algorithm for Finding a Maximum Independent Set of a Cricular-Arc Graph, *SIAM J. Comput.* **17** (1), 41–52.
[167] S. Masuda, K. Nakajima, T. Kashiwabara, and T. Fujisawa (1990), Efficient Algorithms for Finding Maximum Cliques of an Overlap Graph, *Networks* **20**, 157–171.
[168] D. W. Matula (1970), On the Complete Subgraphs of a Random Graph, *Proc. 2nd Conf. Combin. Theory and Appl.*, Chapel Hill, NC: 356–369.
[169] D. W. Matula (1976), The Largest Clique Size in a Random Graph, Technical Report CS 7608, Department of Computer Science, Southern Methodist University.
[170] J. A. McHugh (1990), *Algorithmic Graph Theory*, Prentice-Hall.
[171] W. Meeusen and L. Cuyvers (1975), Clique Detection in Directed Graphs: a New Algorithm, *J. of Comp. and Appl. Math.* **1**, 185–193.
[172] W. Meeusen and L. Cuyvers (1979), An Annotated Bibliograph of Clique-Detection Algorithms and Related Graph-Theoretical Problems, CAM Discussion Paper 7909, State University Center Antwerp.
[173] J. C. Meyer (1972), Ensembles Stables Maximaux dans les Hypergraphes, Comptes Rendus Acad. Sci. Paris **274**, 144–147.
[174] H. Meyniel (1976), On the Perfect Graph Conjecture, *Discrete Mathematics* **16**, 339–342.
[175] H. Meyniel (1985), A New Property of Imperfect Graphs and Some Consequences, Technical Report, E.R. 1975 CNRS, Paris.
[176] W. Miller (1994), Building Multiple Alignments from Pairwise Alignments, to appear in *Computer Applications in the Biosciences*.
[177] G. J. Minty (1980), On Maximal Independent Sets of Vertices in Claw-Free Graphs, *J. Combin. Theory B* **28**, 284–304.
[178] J. W. Moon and L. Moser, (1965), On Cliques in Graphs, *Israel Journal of Mathematics* **3**, 23–28.
[179] R. Motwani (1991), Clique Partitions, Graph Compression and Speeding-up Algorithms, *STOC* (Feder-Motwani).
[180] T. S. Motzkin and E. G. Straus (1965), Maxima for Graphs and a New Proof of a Theorem of Turan, *Canadian Journal of Mathematics* **17** (4), 533–540.
[181] G. D. Mulligan and D. G. Corneil (1972), Corrections to Bierston's Algorithm for Generating Cliques, *J. Assoc. Comput. Mach.* **19**, 244–247.
[182] W. J. Murphy (1992), Computing Independent Sets in Graphs with Large Girth, *Discrete Applied Math.* **36** (2), 167–170.
[183] J. Naor, M. Naor, and A. A. Schaffer (1987), Fast Parallel Algorithms for Chordal Graphs, *proc. 19th Annual ACM Symp. on Theory of Computing*, ACM: 355–364.

[184] G. L. Nemhauser and G. Sigismondi (1992), A Strong Cutting Plane/Branch-and-Bound Algorithm for Node Packing, *J. Opl Res. Soc.* **43** (5), 443–457.
[185] G. L. Nemhauser and L. E. Trotter, Jr. (1974), Properties of Vertex Packings and Independence System Polyhedra, *Math. Programming* **6**, 48–61.
[186] G. L. Nemhauser and L. E. Trotter (1975), Vertex Packings: Structural Properties and Algorithms, *Math, Programming* **8**, 232–248.
[187] G. L. Nemhauser and L.A. Wolsey (1988), *Integer and Combinatorial Optimization*, Wiley, New York.
[188] S. Olariu (1989), Weak Bipolarizable Graphs, *Discrete Mathematics* **74**, 159–171.
[189] S. Onn (1992), On the Combinatorics of Permutation Polytopes, *Journal of Combinatorial Theory, Series A*.
[190] R. E. Osteen (1974), Clique Detection Algorithms Based on Line Addition and Line Removal, *SIAM J. Appl. Math.* **26**, 126–135.
[191] R. E. Osteen and J. T. Tou (1973), A Clique-Detection Algorithm Based on Neighborhoods in Graphs, *Intern. J. Comput. Inf. Sci.* **2**, 257–268.
[192] M. W. Padberg (1973), On the Facial Structure of Set Packing Polyhedra, *Mathematical Programming* **5**, 199–216.
[193] M. W. Padberg (1977), On the Complexity of Set Packing Polyhedra, *Annals of Discrete Mathematics* **1**, 421–434.
[194] M. W. Padberg (1979), Covering, Packing and Knapsack Problems, *Annals of Discrete Mathematics* **4**, 265–287.
[195] M. W. Padberg (1980), $(1,k)$-Configurations and Facets for Packing Problems, *Mathematical Programming* **18**, 94–99.
[196] Panconesi and Ranjan (1990), quantifers and Approximation, *Proc. STOC* 90.
[197] C. Papadimitriou and M. Yannakakis (1981), The Clique Problem for Planar Graphs, *Inform. Proc. Letters* **13**, 131–133.
[198] C. Papadimitriou and M. Yannakakis (1988), Optimization, Approximation and Complexity Classes, Proc. of the Twentieth Annual ACM STOC: 229–234.
[199] P. M. Pardalos and N. Desai (1991), An Algorithm for Finding a Maximum Weighted Independent Set in an Arbitrary Graph, *Intern. J. Computer Math.* **38**, 163–175.
[200] P. M. Pardalos and Y. Li (1991), Global Quadratic Optimization and the Maximum Clique Problem, Technical Report, Computer Science Department, The Penn. State Univ.
[201] P. M. Pardalos and A. T. Phillips (1990), A Global Optimization Approach for Solving the Maximum Clique Problem, *Intern. J. Computer Math.* **33**, 209–216.
[202] P. M. Pardalos and G. Rodgers (1990), Parallel Branch and Bound algorithms for Quadratic Zero-One Programming on a Hypercube Architecture, *Annals of Operations Research* **22**, 271–292.
[203] P. M. Pardalos and G. Rodgers (1990), Computational Aspects of a Branch and Bound Algorithm for Quadratic Zero-One Programming, *Computing* **45**, 131–144.
[204] P. M. Pardalos and G. P. Rodgers (1992), A Branch and Bound Algorithm for the Maximum Clique Problem, *Computers and Operations Research* **19** (5), 363–375.
[205] P. M. Pardalos and J. B. Rosen (1987), *Constrained Global Optimization: Algorithms and Applications*, Springer-verlag, Lecture Notes in Computer Science, Vol. 25.
[206] M. C. Paull and S. H. Unger (1959), Minimizing the Number of States in Incompletely Specified Sequential Switching Functions, *IRE TRANS. Electronic Computers*, EC-8: 356–367.
[207] E. R. Peay, Jr. (1970), An Iterative Clique Detection Procedure, Michigan Math. Physcol. Program: MMPP 70–4.
[208] E. R. Peay, Jr. (1974), Hierarchical Clique Structures, *Sociometry* **37**, 54–65.
[209] O. Perron (1940), Über Lückenlose Ausfüllung des n-diemsnionalen Raumes durch kongruente Wurfel, *Math. Z.* 46, 1–26, 161–180. Zbl 22, 202.
[210] J. C. Picard and M. Queyranne (1977), On the Integer-Valued Variables in the Linear Vertex Packing Problem, *Math Prog.* **12**, 97–101.
[211] B. Pittel (1982), On the Probable Behavior of Some Algorithms for Finding the Stability Number of a Graph, *Math. Proc. Cambridge Philos. Soc.* **92**, 511–526.
[212] S. Provan and M. O. Ball (1983), On the Complexity of Counting Cuts and of Computing the Probability that a Graph is Connected, *SIAM J. Comput.* **12**, 777–788.

[213] W. R. Pulleyblank (1979), Minimum Node Covers and 2-Bicritical Graphs, *Mathematical Programming* **17**, 91–103.
[214] P. Raghavan (1988), Probabilistic Construction of Deterministic Algorithms: Approximating Packing Integer Programs, *J. Comput. System Sciences* **37**, 130–143.
[215] J. Ramanujam and P. Sadayappanv (1988), Optimization by Neural Networks, *IEEE International Conf. on Neural Networks*, San Diego, July 24–27, II: 325–332.
[216] E. M. Reigold, J. Nievergelt, and N. Deo (1977), *Combinatorial Algorithms: Theory and Practice*, Prentice-Hall, Englewood Cliffs, NJ.
[217] J. M. Robson (1986), Algorithms for Maximum Independent Sets, *J. of Algorithms* **7**, 425–440.
[218] D. J. Rose (1970), Triangulated Graphs and the Elimination Process, *J. Math. Anal. Appl.* **32**, 597–609.
[219] D. J. Rose, R. E. Tarjan, and G. S. Leuker (1976), Algorithmic Aspects of Vertex Elimination on Graphs, *SIAM J. Comput.* **5**, 266–283.
[220] D. Rotem and J. Urrutia (1981), Finding Maximum Cliques in Circle Graphs, *Networks* **11**, 269–278.
[221] B. E. Sagan (1988), A Note on Independent Sets in Trees, *SIAM J. Disc. Math.* **1** (1), 105–108.
[222] L. Sanchis (1990), On the Complexity of Test Case Generation for *NP*-hard Problems, *Inf. Processing Letters* **36**, 135–140.
[223] L. Sanchis (1992), Test Case Construction for the Vertex Cover Problem (extended abstract), DIMACS Workshop on Computational Support for Discrete Mathematics.
[224] N. Sbihi (1980), Algorithme de Recherche d'un Stable de Cardinalité Maximum dans un Graph sans Étoile, *Discrete Mathem.* **29**, 53–76.
[225] C. E. Shannon (1956), The Zero-Error Capacity of a Noisy Channel, *I.R.E. Transactions* **3**.
[226] N. Z. Shor (1990), Dual Quadratic Estimates in Polynomial and Boolean Programming, *Computational Methods in Global Optimization* (P. M. Pardalos and J. B. Rosen, eds.), *Annals of Operations Research* **25**, 163–168.
[227] C. De Simone and A. Sassano (1993), Stability Number and Bull-and-Chair-Free Graphs, *Discrete Applied Mathematics* **41** (2), 121–129.
[228] N. J. A. Sloane (1989), Unsolved Problems in Graph Theory Arising from the Study of Codes, *J. Graph Theory Notes of New York* **18**, 11–20.
[229] J. H. Spencer (1979), On Cliques in Graphs, *Israel J. Math.* **9**, 419–421.
[230] W. Staton (1979), Some Ramsey-Type Numbers and the Independence Ratio, *Trans. Amer. Math. Soc.* **256**, 353–370.
[231] S. K. Stein (1974), Algebraic Tiling, *Amer. Math. Monthly* **81**, 445–462.
[232] M. N. S. Swamy and K. Thulasiraman (1981), *Graphs, Networks, and Algorithms*, John Wiley and Sons.
[233] Y. Takefuji, L. Chen, K. Lee, and J. Huffman (1990), parallel Algorithms for Finding a Near-Maximum Independent Set of a Circle Graph, *IEEE Transactions on Neural Networks* **1**, 263–267.
[234] R. Tarjan (1972), Finding a Maximum Clique, Technical Report 72–123, Computer Sci. Dept., Cornell Univ., Ithaca, NY.
[235] R. E. Tajan and A. E. Trojanowski (1977), Finding a Maximum Independent Set, *SIAM Journal on Computing* **6**, 537–546.
[236] R. E. Tajan and M. Yannakakis (1984), Simple Linear-Time Algorithms to Test Chordality of Graphs, Test Acyclicity of Hypergraphs, and Selectively Reduce Acyclic Hypergraphs, *SIAM J. Computing* **13** (3), 566–579.
[237] I. Tomescu (1985), *Problems in Combinatorics and Graph Theory*, John Wiley and Sons.
[238] E. Tomita, Y. Kohata, and H. Takahashi (1988), A Simple Algorithm for Finding a Maximum Clique, Technical Report UEC-TR-C5.
[239] E. Tomita, S. Mitsuma, and H. Takahashi (1988), Two Algorithms for Finding a Near-Maximum Clique, Technical Report UEC-TR-C1.
[240] E. Tomita, A. Tanaka, and H. Takahashi (1988), The Worst-Case Time Complexity for Finding All the Cliques, Technical Report UEC-TR-C5.
[241] L. E. Trotter (1973), Solution Characteristics and Algorithms for the Vertex Packing Problem, Technical Report 168, Dept. of Operations Research, Cornell University, Ithaca, NY.

[242] L. E. Trotter (1974), Vertex Packings: Structural Properties and Algorithms, *Mathematical Programming* **8**, 232–248.
[243] L. E. Trotter (1975), A Class of Facet Producing Graphs for Vertex Packing Polyhedra, *Discrete Mathemtics* **12**, 373–388.
[244] S. Tsukiyama, M. Ide, H. Aviyoshi, and I. Shirakawa (1977), A New Algorithm for Generating All the Maximum Independent Sets, *SIAM J. on Comput.* **6** (3), 505–517.
[245] P. Turan (1954), On the Theory of Graphs, *Colloq. Math.* **3**, 19–30.
[246] L. G. Valiant (1979), The Complexity of Enumeration and Reliability Problems, *SIAM J. Comput.* **8**, 410–421.
[247] M. Vingron and P. Argos (1991), Motif Recognition and Alignment for Many Sequences by Comparison of Dot-Matrices, *J. Molecular Biology* **218**, 33–43.
[248] M. Vingron and P. A. Pevzner (1992), Multiple Sequence Comparison and n-Dimensional Image Reconstruction, Working paper.
[249] V. K. Wei (1981), Lower Bound on the Stability Number of a Simple Graph, Bell Laboratories Technical Memorandum, No. 81-11217-9.
[250] A. Wigderson (1983), Improving the Performance Guarantee for Approximate Graph Coloring, *Journal of the Association for Computing Machinery* **30** (3), 729–735.
[251] H. S. Wilf (1986), The Number of Maximal Independent Sets in a Tree, *SIAM J. Alg. Disc. Meth.* **7** (1), 125–130.
[252] H. S. Wilf (1986), *Algorithms and Complexity*, Prentice-Hall, Englewood Cliffs, New Jersey.
[253] H. S. Wilf (1986), Special Bounds for the Clique and Independence Numbers of Graphs, *J. Combin. Theory B* **40**, 113–117.
[254] S. Wimer, R. Y. Pinter, and J. Feldman (1987), Optimal Chaining of CMOS Transistors in a Functional Cell, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, CAD-6(5): 795–801.
[255] J. Xue (1991), Fast Algorithms for Vertex Packing and Related Problems, Ph.D. Thesis, GSIA, Carnegie Mellon University.
[256] J. Xue (1994), Edge-Maximal Triangulated Subgraphs and Heuristics for the Maximum Clique Problem, *Networks*, to appear.
[257] M. Yannakakis (1988), Expressing Combinatorial Optimization Problems as Linear Programs, *Proc. of the Twentieth Annual ACM STOC*: 223–228.
[258] M. Yannakakis (1992), On the Approximation of Maximum Satisfiability, *Proc. 3rd Annual ACM-SIAM Symp. on Discrete Algorithms*.
[259] G. Yu, P. Kouvelis, and S. Luo (1991), Weighted Vertex Packing Problem for Specially Structured Geometric Graphs, Technical Report, Dept. of Magmt. Sci. and Info. Syst., Univ. of Texas, Austin.
[260] M.-S. Yu, L. Yu, and S.-J. Chang (1993), Sequencial and Parallel Algorithms for the Maximum-Weight Independent Set Problem on Permutation Graphs, *Information Processing Letters* **46**, 7–11.