A Survey of Approximately Optimal Solutions to Some Covering and Packing Problems

VANGELIS T. PASCHOS

LAMSADE, Université Paris-Dauphine

We survey approximation algorithms for some well-known and very natural combinatorial optimization problems, the minimum set covering, the minimum vertex covering, the maximum set packing, and maximum independent set problems; we discuss their approximation performance and their complexity. For already known results, any time we have conceived simpler proofs than those already published, we give these proofs, and, for the rest, we cite the simpler published ones. Finally, we discuss how one can relate the approximability behavior (from both a positive and a negative point of view) of vertex covering to the approximability behavior of a restricted class of independent set problems.

Categories and Subject Descriptors: F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems—*computations on discrete structures*; G.1.6 [**Numerical Analysis**]: Optimization—*integer programming, linear programming*; G.2.1 [**Discrete Mathematics**]: Combinatorics—*combinatorial algorithms*; G.4 [**Mathematics of Computing**]: Mathematical Software—*algorithm analysis*

General Terms: Algorithms, Theory

Additional Key Words and Phrases: Algorithm analysis, approximation algorithms, combinatorial algorithms, constrained optimization, problem complexity

1. INTRODUCTION

1.1 The Problems

Set cover. Given a collection $\mathcal{S}$ ($|\mathcal{S}| = n$) of subsets of a finite set C ($|C| = m$), a set cover is a subcollection $\mathcal{S}' \subseteq \mathcal{S}$ such that $\cup_{S_i \in \mathcal{S}'} S_i = C$, and the minimum set cover problem (SC) is to find a cover of minimum size.

Set packing. Given a collection $\mathcal{S}$ ($|\mathcal{S}| = n$) of finite sets, a packing is a subcollection $\mathcal{S}' \subseteq \mathcal{S}$, all members of which are mutually disjoint and the maximum set packing problem (SP) is to find a packing of maximum size.[1]

Vertex cover. In a graph $G = (V, E)$, a vertex cover is a subset $V' \subseteq V$ such that, for each edge $uv \in E$, at least one of u and v belongs to V' and the minimum vertex cover problem (VC) is to find a minimum size vertex cover.

Independent set. An independent set in graph $G = (V, E)$ is a subset $V' \in V$ such that, for every pair (v_i, v_j) of verti-

[1] In fact, the dual integer linear program of SC is the set packing problem.

Author's address: LAMSADE, Université Paris-Dauphine, Place du Maréchal De Lattre de Tassigny, 75775 Paris Cedex 16, France; ⟨paschos@lamsade.dauphine.fr⟩.

ces in V', the edge v_iv_j does not belong to E, and the maximum independent set problem (IS) is to find a maximum size independent set.

Hitting set. Let $\mathcal{S}$ be a family of n subsets drawn from an m-element set C. A subset $H \subseteq C$ is a hitting set for the family $\mathcal{S}$ if H has a nonempty intersection with each element of this family, and the minimum hitting set problem (HS) is that of finding a hitting set of minimum cardinality.

Clique. Consider a graph $G = (V, E)$. A clique of G is a subset $V' \subseteq V$ such that every two vertices of V' are joined by an edge in E, and the maximum clique problem (C) is to find a maximum size set V' inducing a clique in G (a maximum size clique).

Dominating set. A dominating set of G is a subset $V' \subseteq V$ such that $\Gamma(V') = V \setminus V'$, and the minimum dominating set problem (DS) is to find a dominating set of minimum size (where by $\Gamma(V')$, we denote the set of neighbors of the elements of V').

All the preceding problems admit natural generalizations, where we associate positive weights on the data describing them (vertices for VC, IS, C, and DS sets for SC and SP, elements for HS), and then the objective becomes to optimize (minimize for SC, VC, HS, and DS, maximize for SP, IS, and C) the sum of the weights of the solution (we denote by WSC, WSP, WVC, WIS, WHS, WC, and WDS the weighted versions of the corresponding problems). The original NP-completeness proofs are found either in Karp [1972] or in Garey and Johnson [1979].

IS is approximate equivalent to SP, in the sense that every approximation algorithm solving the former also solves the latter within the same approximation ratio; this equivalence becomes very clear and intuitive by means of a graph (see Definition 3 at the beginning of Part II) defined for every SP-instance; proofs of this equivalence are found in Berge [1973] and Simon [1990]. Problem HS is the dual of SC, via the interchanging of the roles of the sets $\mathcal{S}$ and C; consequently, SC and HS are also approximate equivalent. Problem C is approximate equivalent to IS in the sense that an independent set in a graph G corresponds to a clique in graph $\bar{G} = (V, \{v_iv_j : i \neq j, v_iv_j \notin E\})$. Finally, DS is approximate equivalent to SC via the following reduction described in Paz and Moran [1981].

Given an instance ($\mathcal{S} = \{S_i : i = 1, \ldots, n\}$, $C = \{c_j : j = 1, \ldots, m\}$) of SC, one can construct an instance of $G = (V, E)$ of DS as follows: $V = \{v_1, \ldots, v_n, v_{n+1}, \ldots, v_{n+m}\}$, $E = \{v_iv_j : 1 \leq i < j \leq n\} \cup \{v_iv_{n+j} : c_j \in S_i, i = 1, \ldots, n, j = 1, \ldots, m\}$; it is easy to see that via this reduction, every solution of SC corresponds to a solution of DS of the same cardinality, thus the reduction is approximation preserving.

1.2 A Short Overview of the Main Negative Results

The approximability behavior of SC, IS, and SP has remained for a long time a very (perhaps the most) interesting open problem.

In 1992, Berman and Schnitger renewed the interest of the scientific community about the approximability of IS, by proving that for some positive constant c it is not possible to approximate IS within a factor of n^c, provided maximum 2-satisfiability does not admit a randomized polynomial time approximation schema.

This fascinating result has, among others, the great merit of associating the approximability of IS with the one of maximum 2-satisfiability which is in max-SNP, the complexity class defined in Papadimitriou and Yannakakis [1988].

This association lies at the heart of the strongest approximation result for IS. In fact, Arora et al. [1992] proved that all the problems that are hard for max-SNP are not approximable by polynomial time approximation schemas (unless P = NP); furthermore, IS and VC, both for bounded-degree graphs, are hard for max-SNP and the following result holds.

THEOREM 1 [Arora et al. 1992]. *VC, even for bounded degree graphs, and IS for bounded-degree graphs do not admit a polynomial time approximation schema*[2] *unless P = NP.*

On the other hand, the following theorem is classical in the domain of polynomial approximation theory.

THEOREM 2 [Garey and Johnson 1979; Papadimitriou and Steiglitz 1981]. *Either IS can be solved by a polynomial time approximation schema, or there is no polynomial time approximation algorithm for IS achieving an approximation ratio bounded below by a fixed positive constant.*

The combination of Theorems 1 and 2 results in the nonexistence of a constant-ratio[3] polynomial time approximation algorithm (PTAA) for IS. On the other hand, since SP is approximately equivalent to IS, the negative results for the latter also hold for the former.

Let us note here that, once the impossibility of approximating VC by polynomial time approximation schema is established, very natural questions arise: the first one is "what is the constant hardness factor[4] for VC?" and the second one is "which could be the best approximation ratio for it?"

For the first question, Bellare et al. [1995] supply the answer by proving the following theorem.

THEOREM 3 [Bellare et al. 1995]. *VC is not approximable with a ratio smaller than* $1.038512 > 27/26$*, unless P = NP.*

The second question is discussed in detail in Part V of this article.

For SC, a first negative result can be easily obtained from Theorem 1. Since VC is a subproblem of SC, the negative result for the former is carried over the latter, thus the problem of finding a polynomial time approximation schema for SC is NP-complete.

About this problem, Lund and Yannakakis [1992] (see also Johnson [1992]) have proved a tighter and stronger negative result for SC's approximability.

THEOREM 4 [Lund and Yannakakis 1992]. *SC cannot be approximated with ratio* $c \log n$ *for any* $c < 1/4$ *unless* $NP \subseteq DTIME[n^{\text{poly} \log n}]$.[5]

Let us note here that in Paschos and Renotte [1995] we proposed as an open problem the strengthening (transformation of the condition NP $\subseteq$ DTIME[$n^{\text{poly} \log n}$] into the one P = NP) of the result of Theorem 4 via the study of approximation-preserving reductions linking SC to the coloring problem. Recent results of Halldórsson [1995a, b] support the idea of Paschos and Renotte [1995].

The following theorem summarizes the main points of the preceding brief discussion.

THEOREM 5

—*SP, IS, and C do not admit constant-ratio PTAA unless P = NP.*

—*SC, HS, as well as DS and VC, do not admit a polynomial time approximation schema unless P = NP.*

—*Moreover, SC, HS, DS do not admit PTAA with ratio c log n for any* $c < 1/4$ *unless* $NP \subseteq DTIME[n^{poly \log n}]$.

[2] A sequence of algorithms approximating the optimal solutions within an error arbitrarily close to 1.

[3] To evaluate the approximation performance of an algorithm, we use the measure $A(I)/OPT(I)$, where $A(I)$ denotes the value of the solution provided by the approximation algorithm and $OPT(I)$ the value of the optimal one.

[4] In fact, by rephrasing the result of Theorem 1 for VC, we have: " ϵ such that VC cannot be approximated within a factor less than, or equal to, $1 - \epsilon$;" the quantity ϵ is the constant hardness factor.

[5] The conjecture NP $\not\subseteq$ DTIME[$n^{\text{poly} \log n}$] is weaker than P $\neq$ NP, but that NP is included in DTIME[$n^{\text{poly} \log n}$] is highly improbable.

1.3 Scope

Among all the problems cited, only VC admits a number of PTAAs providing solutions of size at most twice the size of the optimal one but, up to now, researchers have been unable to improve this ratio significantly, or to prove lower bounds, stronger than that of Theorem 3, for the approximation ratio of every algorithm supposed to solve VC. In Part V of this article we give sufficient conditions for both these eventualities.

On the other hand, let us quote here (without further discussion, such a discussion being beyond the scope of this article) the very nice results of Baker [1994] about the approximability of some of the preceding problems on *planar graphs.*[6] In fact, in Baker [1994], *polynomial time approximation schemas*, based on dynamic programming arguments, are devised for IS, VC, DS, and a number of other combinatorial optimization problems defined on planar graphs.

In this strongly negative context (for the general approximability behavior of the problems studied), and since all of these problems are from both theoretical and practical points of view very interesting (problems on operating systems, databases, computer-aided systems, satellite imaging systems, etc., can be represented and treated in terms of one of these problems), we think that it is crucial (and challenging) to study existing approximation algorithms or to devise new ones, always with the objective of designing efficient algorithmic methods (or improving existing ones) that find "reasonable" solutions for the instances of the problems.

In this article we do not further survey the recent negative results already mentioned. A broad discussion of these results, as well as of the origins of the mathematical tools allowing us to obtain them, is found in the literature.[7]

Also, we do not survey approximation results on random instances for the problems under discussion; such results for SC can be found in Blot et al. [1995], Fernandez de la Vega et al. [1992], and Karp [1976] (in Blot et al. [1995], we propose a framework for the solution of random optimization problems, this framework also including SP and the hypergraph independent set), for SP in Fernandez de la Vega [1992] (an extension of the main theorem of Fernandez de la Vega [1982] is found in Paschos [1993]), and for VC and IS, a broad discussion and presentation of classical results is found in Bollobás [1985]. Finally, we note that a small survey for SC was published in Paschos and Demange [1994]; however, for purposes of this article's autonomy and since, sometimes, when discussing VC we make references to SC algorithms, we survey, in Section 2, the most well-known approximation algorithms for the latter problem.

1.4 Notation

Let us consider a graph $G = (V, E)$. In what follows, given a set of edges $A \subseteq E$, we denote by $T[A]$ the set of the endpoints of the edges of A ($T[A] \subseteq V$); given a set V' of vertices of a graph G, we denote by $G[V']$ the partial subgraph of G induced by V'. Given a maximum (or maximal) matching M of G and an edge $uv \in M$, we denote the vertex v (resp., vertex u) by $m(u)$ (resp., $m(v)$); given a vertex x, exposed with respect to M, we denote by $M[x]$ the subset of M such that, given an edge $e \in M$, at least one endpoint of e is incident to x. We denote by $E(G)$ (resp., $V(G)$) the edge (resp., vertex) set of G; we denote by $v_i - v_{i_1} - \ldots - v_j$ an elementary path from vertex v_i to vertex v_j in G. Also, for every vertex $v \in V$,

[6] A graph is called planar if it can be represented on a plane in such a way that its vertices are distinct points, its edges are simple curves, and every pair of edges does not admit intersection points other than their endpoints.

[7] See Arora et al. [1992], Johnson [1992], Lund and Yannakakis [1992], Papadimitriou [1994], and Papadimitriou and Yannakakis [1988, 1993].

we denote by $\Gamma(v)$, the set of neighbors of v, by δ_v the quantity $|\Gamma(v)|$, by Δ, the quantity $\max_{v \in V} \{|\Gamma(v)|\}$, that is, the maximum degree of G and by μ the quantity $(\Sigma_{v \in V} \delta_v)/n$, the average degree of G. Finally, given a set $V' \subseteq V$, we denote by $\Gamma(V')$, the set $\cup_{x \in V'} \Gamma(x) \setminus V'$. Especially for the cases of SC and SP, we denote by Δ the quantity $\max_{S_i \in \mathcal{S}} \{|S_i|\}$.

Also, we denote by $\alpha(G)$ the stability number (cardinality of a maximum independent set) of G and by $\iota(G)$ its stability ratio, that is, the quantity $\alpha(G)/n$. Also, given an instance G of a problem Π, we denote sometimes by $v(\Pi(G))$ its optimal (objective) value (so, for the unweighted case of IS, $v(IS(G)) = \alpha(G)$). Finally, whenever vector $\vec{1}$ is indexed by a set of vertices, it denotes the characteristic vector of this set. Especially for the linear programs of Section 8, since the dimensions of the vectors $\vec{1}$ and $\vec{0}$ are not always the same, these vectors are indexed by their dimension.

Given a graph G, let us consider a minimum vertex cover C^* ($|C^*| = \tau(G)$) and the corresponding maximum independent set S^*. Let us also suppose that, given a maximum matching M ($|M| = m$), there are f matching edges such that both their endpoints belong to C^*, for the remaining ones, one of their endpoints belonging to C^* and the other one to S^*. Let us call these edges "dissymmetric" and denote by F the set of these "dissymmetric" edges ($|F| = f$). For M (whenever it is not perfect), let us denote by X the set of the exposed (nonsaturated) vertices of G with respect to M, and by X_C ($|X_C| = g$) and X_S the subsets of X belonging to C^* and S^*, respectively (of course, $X = X_C \cup X_S$).

PART I. SET COVERING

2. Former Approximation Algorithms

The approximation of SC was originally studied by Johnson [1974] and also by Lovász [1975b] in the unweighted case, and by Chvátal [1979] in the weighted one. All of these authors study a natural greedy algorithm for SC and compute its approximation ratio in both cases. We give this algorithm in the weighted case (Algorithm 1). It is easy to see that in the unweighted case, the set chosen on the fourth line of Algorithm 1 is the maximum cardinality set first in the initial instance and, after, in the surviving ones. The complexity of Algorithm 1 is of $O(mn)$.

Algorithm 1. The greedy SC-algorithm.

begin
  $S' \leftarrow \emptyset$;
  while $S_i \neq \emptyset$, $i \leq n$ **do**
    choose S_j such that $|S_j|/w_j = \max_{1 \leq i \leq n} \{|S_i|/w_i\}$;
    $\mathcal{S}' \leftarrow \mathcal{S}' \cup \{S_j\}$;
    for all $i \leq n$ **do** $S_i \leftarrow S_i \setminus S_j$ **od**
  od
end.

Both the weighted and unweighted versions of Algorithm 1 lead to approximation ratios of $1 + \ln(\max_{i \leq n} |S_i|)$ and these bounds are tight for both versions.

Hochbaum [1982] presents an approximation algorithm for WSC using a variant of the Russian method Aspvall and Stone [1980] and taking, at worst, $O(n^3 \log n)$ steps.

In fact, she considers the linear-integer program describing WSC; she passes on the relaxed (continuous) version WSC_r of WSC, and by solving its dual one WSC_r^d, she obtains a suboptimal solution for WSC, the value of which does not exceed the maximum number of sets covering an element (in this solution) times the value of the optimal solution.

Bar-Yehuda and Even [1981] propose Algorithm 2 that attains the same approximation ratio as Hochbaum's algorithm, but in time complexity of $O(n^2)$.

In Paschos and Demange [1994] we have somewhat changed the proof of the theorem characterizing the approximation performance of Algorithm 2 in order to make it completely combinatorial (even if the algorithm in Bar-Yehuda and Even [1981] does not involve a solution of a linear program, the elegant

Algorithm 2

begin
  for each c_k **in** C **do**
    $\boldsymbol{\omega} \leftarrow \mathbf{min}_{s_i \in \Gamma(c_k)} \{w(S_i)\}$;
    for each $s_i \in \Gamma(c_k)$ **do** $w(s_i) \leftarrow w(s_i) - \omega$ **od**
  od
  $S' \leftarrow \{s_i : w(s_i) = 0\}$;
end.

Algorithm 3

begin
  execute algorithm 1 until set size becomes at most 2;
  let S_1 be the obtained solution on the sub-instance with $\Delta > 2$;
  starting from the surviving SC-instance apply definition 2;
  solve the minimum edge covering on G_C;
  let S_2 be the obtained edge covering-solution on G_C;
  $S' \leftarrow S_1 \cup S_2$;
end.

proof of their theorem uses linear programming techniques (and the authors admit this fact)).

We consider, for the purposes of Algorithm 2, an instance of SC in terms of its characteristic (or membership) bipartite graph $B = (S, C, E)$ defined in Definition 1. By definition of B, a subset $S' \subseteq S$ sending edges to all vertices of the color class C corresponds to a cover of the set C.

Of course, such a graph B can be constructed for every problem defined on set systems as, for example, SP. For SP, we can consider that $C = \cup_{S_i \in S} S_i$, and a feasible solution, with respect to B, is a set S' of S-vertices such that, for every two vertices s_i, s_j of S', there is no simple path on two edges in B linking s_i and s_j.

Definition 1. The characteristic (or membership) bipartite graph $B = (S, C, E)$ of SC is the bipartite graph with color classes S representing the family $\mathcal{S}$, and C representing the set C, and with edge-set E containing an edge $s_i c_j$ if the set S_i contains the element c_j. In the case of WSC, the set S of vertices of B is weighted.

Algorithm 2 also admits a counterexample showing that the attained ratio is tight.

3. RECENT APPROXIMATION RESULTS FOR (UNWEIGHTED) SET COVER

Goldsmidt et al. [1993] (see also Paschos and Demange [1994]) proposed the $O(m^{2.5})$ Algorithm 3, achieving a slightly better worst-case approximation ratio bounded above by $(5/6) + \ln \Delta$ (recall that, for SC and SP, $\Delta = \max_{i \leq n} |S_i|$).

The key idea of this work lies in the construction of an algorithmic schema including the combination of more than one (approximation or exact) algorithm applied on different subinstances of an instance of SC. They then consider, as a final SC-solution, the union of the outputs of the different components of the schema.

The edge-covering algorithm called by Algorithm 3 works on a graph derived from an instance of SC by means of the following definition.

Definition 2. Given an instance $(\mathcal{S}, C)$ of SC, the graph $G_C = (V, E)$ can be constructed as follows: $V = C$; $E = \{c_i c_j : S_k \in \mathcal{S}, \{ci, c_j\} \subseteq S_k\}$.

It is easy to see that whenever $\Delta \leq 2$, SC reduces to the well-known *minimum edge covering* problem which is polynomial [Berge 1973; Garey and Johnson 1979]. In fact, this problem implies the computation of a maximum matching which, in a graph of order n, can be done in $O(n^{2.5})$ (this explains the overall complexity of Algorithm 3). The use of the edge-covering algorithm in place of Algorithm 1, for $\Delta \leq 2$, induces a gain of 1/6 times the cardinality of the optimal solution, with respect to the ratio of Algorithm 1.

More recently, a further improvement of SC's approximation ratio has been proposed in Halldórsson [1995a]. The

Algorithm 4. A 3SC-algorithm parametrized by t.

begin
find a maximal set packing S_1 for the instance (S, C) of 3SC;
$C' \leftarrow C \setminus \cup_{S_i \in S_1} S_i$;
$\mathcal{S}' = \{S_i \cap C' : S_i \in \mathcal{S}, S_i \cap C' \neq \emptyset$;
apply definition 2 on $(\mathcal{S}', C')$;
solve the minimum edge covering on G_C;
let $\mathcal{S}_2$ be the obtained edge covering-solution on G_C;
repeat apply local t-improvements to $\mathcal{S}_1 \cup \mathcal{S}_2$ until t-optimality;
output the t-optimal solution $\tilde{S}'$;
end.

basic thought process of Halldórsson [1995a] is the same as in Goldsmidt et al. [1993], that is, the combination of different algorithms in several subinstances of the initial SC-instance. More precisely, it combines Algorithm 1 with an algorithmic schema approximately solving SC on set systems with $\Delta \leq 3$ (let us denote by 3SC the restriction of SC to instances verifying $\Delta \leq 3$). This latter algorithm (Algorithm 4) is based upon a clever strategy consisting of starting with an arbitrary initial SC-solution and then iteratively searching for a solution of smaller cardinality.

To do this Halldórsson [1995a] defines and uses the notion of *t-improvement*. A t-improvement of a cover $\mathcal{S}'$ is formed by sets $\tilde{S}'_1, \tilde{S}'_2, \ldots, \tilde{S}'_t$ in $\mathcal{S}'$ and by sets $S_1, S_2, \ldots, S_{t-1}$ in $\mathcal{S}$ such that $\mathcal{S}'' = (\mathcal{S}' \setminus \{\tilde{S}'_1, \tilde{S}'_2, \ldots, \tilde{S}'_t\}) \cup \{S_1, S_2, \ldots, S_{t-1}\}$ is also a cover. Obviously, $|\mathcal{S}''| < |\mathcal{S}'|$. A cover is t-optimal if it contains no t-improvement. Moreover, for fixed t, t-improvements can be done in polynomial time, as does the verification of t-optimality.

The very elegant analysis of Algorithm 4 (parametrized by t), performed in Halldórsson [1995a] produces an approximation ratio at most $(7/5) + \epsilon(4/t)$, which also constitutes an interesting improvement for the approximation ratio of 3SC. The only drawback of this result is that the complexity of the algorithm is of $O(n^{O(1/\epsilon)})$ and therefore the closer to 7/5 the ratio, the higher the execution time of Algorithm 4.

Algorithm 5

begin
execute algorithm 1 until set size becomes at most 3;
let $\mathcal{S}_1$ be the obtained SC-solution;
let $(\mathcal{S}, C)$ be the surviving SC-instance;
apply algorithm 4 to $(\mathcal{S}, C)$;
let $\mathcal{S}_2$ be the obtained 3SC-solution;
$\mathcal{S}' = \mathcal{S}_1 \cup \mathcal{S}_2$ is the final SC-solution;
end.

Based on Algorithm 4, Algorithm 5 devised in Halldórsson [1995a] is a PTAA for SC attaining a ratio at most $\ln \Delta + (17/30 + \epsilon$, in time of $O(n^{O(1/\epsilon)})$. This, from the (worst case) approximation performance point of view, is, to our knowledge, the best PTAA for SC.

In Paschos and Demange [1994], we devise a new efficient $O(nm)$-approximation algorithm for SC relying on the transformation of SC to a particular flow problem. Up to now, we have not been able to produce an interesting expression for the approximation ratio of this algorithm; however, experimental results show that it is very performant (more performant than Algorithm 1, for example). We think that a fine and detailed characterization of this behavior is very interesting and, in Paschos and Demange [1994], we propose it as a nice open problem.

Finally, the analysis of Algorithm 3 performed in Paschos and Demange [1994] brought to the fore the fact that, whenever G_C (Definition 2) admits a perfect matching, the approximation ratio of the algorithm is at most $\ln \Delta + (1/2)$. Of course, this quantity does not constitute a worst case guarantee; but the case of a perfect matching occurs very frequently given that G_C is, a priori, dense.

PART II. VERTEX COVERING

Let us consider the instances of SC with family size n. If we take into account the ones verifying the restriction "for every $c \in C$, c belongs to exactly two sets of the family and moreover there are no two elements contained in the same pair of sets," we then obtain all the instances of size n of VC, or equivalently, all the graphs (without loops) of order n. This is easy to see if we consider the bipartite graph B constructed by Definition 1. The previously mentioned restriction creates the bipartite graphs regular on C, with c-degrees all equal to 2. Let us now transform B following Definition 3.

Definition 3. Given a bipartite graph $B = (S, C, E')$, a graph $G = (V, E, \ell)$ can be constructed as follows: $V = S$; $E = \{v_i v_j : \quad c_k, s_i - C_k - s_j$ is a path of $B\}$; $\ell : \{1, \ldots, m\} \rightarrow E$ is an edge-labeling such that, $\quad v_i v_j \in E$, $\ell(v_i v_j) = k$ iff $s_i - c_k - s_j$ is a path of B. If B is weighted on its S-vertices, then G is also weighted (on V).

The transformation implied by Definition 3 is valid for every bipartite graph, hence for every instance of SC, the graph G being a multigraph in the sense that every edge in E may have multiple labels. If, in addition, the restriction previously mentioned is considered, then the labeling ℓ is redundant, given that each edge of G has, in this case, a unique label. Thus G is a simple graph; consequently, the transformation implied by Definition 3 produces all the instances of VC of order n.

Of course, because of the remarks just before Definition 1, Definition 3 applies also for SP; moreover, a feasible solution for SP is, with respect to G, an independent set of the same size.

4. ALGORITHMS FOR (UNWEIGHTED) VERTEX COVERING

A consequence of the preceding remarks is that all the algorithms studied in Section 2 also solve VC. In what concerns Algorithm 1 in its unweighted version, its VC-version consists of simply picking the maximum degree vertex in the survived graph.

Relative to the unweighted VC-version of Algorithm 1, let us consider the following configuration, inspired by Papadimitriou and Steiglitz [1981].

For a given k, let us consider a perfect matching $\{c_1 b_1, \ldots, c_K b_K\}$, on $K = 2.3 \ldots (k-1).k$ vertices. Next we create a new set of vertices; let us call these new vertices a-vertices; we partition the K b-vertices into $3 \ldots k - 1.k$ pairs and we link each of them to a new vertex (we use for this purpose $3 \ldots (k-1).k$ a-vertices); we partition, once more, the b-vertices into $2.4 \ldots (k-1).k$ triples and we join each such triple to a new a-vertex (for this purpose we use $2.4 \ldots (k-1).k$ new a-vertices), ...; finally, we partition the set of b-vertices into $2.3 \ldots (k-1)$ k-tuples and we append the vertices of each k-tuple to a newly created a-vertex (we use thus $2.3 \ldots (k-1)$ new vertices). Let us denote by $F(K)$ the number of the a-vertices so created.

In the so-constructed graph, for the unweighted VC-version of Algorithm 1 we could choose as solution the union of the sets of the a- and c-vertices, whereas the optimal one is the set of the b-vertices. Then it is easy to see that the approximation ratio of the algorithm is $(F(K) + K)/K$ and this quantity grows as fast as $O(\ln K)$.

Algorithm 6. The best known VC algorithm.

begin
compute a maximal matching M in G;
$C \leftarrow T[M]$;
end.

Algorithm 6 is the simplest and most known linear time (in $|E|$) approximation algorithm (due to Gavril [1979]), for finding a solution of the unweighted VC, the cardinality of which is at most twice as large as the cardinality of the optimal one.

This algorithm also admits a counterexample showing that the ratio 2 is tight. In fact, let us consider a graph $G = (V, E)$ of order n admitting minimum vertex covering of cardinality $n/2$

and a perfect matching. This graph can be easily constructed by taking two independent sets V_1 and V_2 of cardinality $n/2$, and by first adding a perfect matching between V_1 and V_2 and finally by completing the graph with edges lying in the interior of V_1 or linking one vertex of V_1 to a vertex of V_2. We can see that picking a maximal matching can lead to picking the perfect one, constructing a VC-solution of cardinality n.

Another algorithm achieving ratio 2 for VC has been conceived by Savage [1982]. It consists of a simple depth-first-search (DFS, [Aho et al. 1975]) on G, and the solution is formed by the nonleaf vertices of the obtained DFS-tree; let us denote this tree by $T = (V, E_T)$.

To see that the nonleaf vertices constitute a cover, we have to prove that there is no edge in E linking two leaves. Plainly, DFS implies that edges of $E \setminus E_T$ link the root of a subtree to vertices contained in this subtree; hence, since every leaf can be considered as the root of a (empty) subtree, two leaves cannot be linked by an edge in $E \setminus E_T$.

Let us prove now that the ratio of DFS is always smaller than or equal to 2. We reason with respect to a maximum matching M on T:

—suppose that M saturates all the nonleaf vertices of T and none of the leaves; then, since there are no edges linking any two leaves, M is maximal for G, and the arguments of Algorithm 6 suffice to let us conclude an approximation ratio equal to 2;

—suppose now that M saturates some leaves, some nonleaf vertices being left unsaturated, and also suppose that there exist edges in $E \setminus E_T$ such that their endpoints are both unsaturated by M; then, if all of these edges are added to M, the resulting matching M' is maximal for G. The edges of M' can be divided into two subsets, say M'_1 and M'_2, where the members of M'_1 have both their endpoints in the covering (they are nonleaves), whereas the edges of M'_2 have only one endpoint in the covering (this is the case when some leaves are saturated either by M or by $M' \setminus M$). Since $M'_1 \cup M'_2 = M'$, which is maximal for G, the cardinality of the DFS-solution for VC is smaller than or equal to $2|M'|$, whereas the cardinality of the optimal one is greater than or equal to $|M'|$; so, the ratio 2 is immediately deduced.

The bound found by the DFS algorithm is tight. In fact, if we consider the case where G is a path of length $2k + 1$ and the DFS starts from one of the two vertices of degree 1, then we take $2k$ vertices in the approximate solution, and the optimal one has cardinality k.

5. WEIGHTED VERTEX COVERING

5.1 Weighted Vertex Covering and the Semiintegral Property

The first real study of the approximation of WVC is implied by Nemhauser and Trotter [1975], even if the main purpose of their work was not exactly to study this problem.

In Nemhauser and Trotter [1975], the authors consider the following integer linear program for WVC:

$$\text{WVC} = \begin{cases} \min & \vec{w} \cdot \vec{x} \\ & A \cdot \vec{x} \geq \vec{1} \\ & \vec{x} \in \{0, 1\}^n, \end{cases}$$

where A is the edge-vertex incidence matrix of G.

They show that the LP-relaxation WVC_r of WVC has the semi-integral property; that is, the basic feasible optimal solution of WVC_r assigns to the variables values drawn from the set {0,1/2,1} (we show later how Hochbaum uses the semi-integral property to slightly improve the approximation ratio of WVC). It turns out that the semi-integral solution for WVC_r can be used to obtain a suboptimal solution for WVC by picking a vertex, the corresponding variable of which has nonzero value.

Algorithm 7. The input of the algorithm is a connected graph $G = (V, E)$.

begin
create a copy V' of V;
compute the bipartite graph $B_F = (V, V', E_{BF})$, where $E_{BF} = \{vu' : \ vu \in E\}$;
compute an optimal vertex cover for B_F by a maximum flow computation;
let C_{BF} be the vertex cover computed;
$V_C \leftarrow \{v \in V : v \in C_{BF} \vee v' \in C_{BF}\}$;
end.

In Motwani [1993], a combinatorial interpretation of the Nemhauser-Trotter work is given, and a 2-approximation algorithm (Algorithm 7, originally devised in Bar-Yehuda and Even [1985]) is described without referring to the linear programming formulation. In Algorithm 7, the phase comprising the computation of C_{BF} is polynomial, since WVC in bipartite graphs can be optimally solved in polynomial time via a reduction to the maximum flow problem.

Theorem 6 summarizes the results of Nemhauser and Trotter [1975] concerning WVC (see also Bar-Yehuda and Even [1985], and Motwani [1993], where elegant combinatorial proofs are given for Theorem 6).

THEOREM 6. [Nemhauser and Trotter 1975]. *Let $C^2_{BF} = \{v \in V : \{v, v'\} \subseteq C_{BF}\}$ and $C^1_{BF} = \{v \notin C^2_{BF} : \{v, v'\} \cap C_{BF} \neq \emptyset\}$. Then C^1_{BF} and C^2_{BF} have the following properties:*

(*i*) *if $D \subseteq C^1_{BF}$ is a vertex cover for $G[C^1_{BF}]$, then $V' = D \cup C^2_{BF}$ is a vertex cover for G;*
(*ii*) *there exists some optimal vertex cover V^* for G such that $C^2_{BF} \subseteq V^*$;*
(*iii*) *the optimal solution for $G[C^1_{BF}]$ has weight at least half as much as the total weight of the vertices of $G[C^1_{BF}]$.*

Starting from the sets C^1_{BF} and C^2_{BF}, we can consider a partition of V into three sets C^1, C^2 and $V \setminus (C^1 \cup C^2)$ (where C^1 and C^2 are the projections on V of the sets C^1_{BF} and C^2_{BF}, resp.).

In terms of the Nemhauser-Trotter proof, we can interpret this partition as follows (see also Hochbaum [1983]): the vertices of C^1 are the vertices whose values in the solution of the LP-relaxation for WVC correspond to variables of value 1/2, the ones of C^2 correspond to variables of value 1, and, finally, the vertices of $V \setminus (C^1 \cup C^2)$ correspond to variables of value 0. For the graph $G[C^2]$, Algorithm 7 has found the optimal solution, whereas for completing the solution on G, we have to compute the solution of $G[C^1]$; from condition (*iii*) of Theorem 6, the value of this solution is at most twice as much as the value of the optimal solution for $G[C^1]$.

On the other hand, by the constraints on the edges (see the integer linear program formulation of WVC), the vertices of $V \setminus (C^1 \cup C^2)$ form an independent set and, moreover, all their neighbors are in C^2; therefore, all the edges between C^2 and $V \setminus (C^1 \cup C^2)$ are covered by C^2. Consequently, the approximation ratio of Algorithm 7 is $[w(C^1) + w(C^2)]/[(1/2)w(C^1) + w(C^2)] \leq 2$, where by $w(X)$, $X \subseteq V$, we denote the sum of the weights of the vertices of X.

5.2 On the Approximation of Weighted Vertex Covering

Before discussing the several very interesting algorithms existing in the literature, let us notice that given the polynomial 2-approximability of VC, one can prove directly that WVC does so, by constructing an approximation-ratio-preserving reduction between VC and WVC (more details on this domain of approximability preserving reductions can be found in the literature).[8]

It is easy to see that, since VC is a restricted case of WVC, every algorithm solving the latter solves the former within the same approximation ratio.

[8] See Crescenzi and Panconesi [1989], Kann [1992], Orponen and Mannila [1987], Paschos and Renotte [1995], and Simon [1990].

However, given an instance G_w of WVC, we can construct an instance G of VC by replacing every vertex v_i of weight w_i by an independent set W_i of size w_i; next, if two vertices v_i and v_j are linked by an edge in G_w, we draw a complete bipartite graph between the two independent sets W_i and W_j of sizes w_i and w_j, respectively.

Let us now consider a solution of G; of course, this solution will be constituted from vertices of the several independent sets replacing vertices of G_w. Moreover, let us consider a complete bipartite graph formed by the sets, say W_i and W_j (standing for an edge of G_w); obviously, there do not exist two vertices $w_{i_k} \in W_i$ and $w_{j_l} \in W_j$ such that neither w_{i_k} nor w_{j_l} belong to the solution considered because, in this case, the edge $w_{i_k}w_{j_l}$ would not be covered; so, for each edge v_iv_j of G_w (with $w(v_i) = w_i$ and $w(v_j) = w_j$), at least one of the sets W_i, W_j will be entirely included in the solution for G; consequently, every time a set W_k is not entirely included in the solution for G, one can immediately delete this group, thus obtaining a new smaller feasible solution for G.

Henceforth, if an algorithm delivers a solution for G (instance of VC), one can directly obtain a solution of G_w by simply replacing the independent sets of the former by the corresponding vertices of the latter, this operation leading to a solution for G_w with the same value as the cardinality of the solution for G. Thus the approximation ratios for the two solutions remain the same.

Despite the preceding discussion, it seems worthwhile to study autonomous techniques for the approximation of VC. The fact is that a number of very nice results and very elegant strategies have been developed by trying to directly approximate WVC. We show, in what follows, the most known among them.

Of course, Algorithm 1, in its WVC-version, constitutes an $O(\log n)$ PTAA also for WVC. On the other hand, in Section 2, we have mentioned Hochbaum's SC linear programming algorithm and also Algorithm 2, both algorithms achieving a ratio equal to the largest C-vertex degree of B; obviously, these algorithms, when used for WVC, achieve a ratio 2.

In Hochbaum [1982], a slight improvement of VC's approximation ratio is described, and a ratio of $2 - [2/(r + 1)]$, where r is the length of the largest unweighted odd cycle of G, is obtained.

5.2.1 *Local Ratio Techniques and The Algorithm of Bar-Yehuda and Even.* We now describe a technique that results in slight improvements of the VC ratio producing ratios equal to $2 - o(1)$, the $o(1)$ functions being decreasing functions of n. This technique, called the "local ratio technique" (LRT), was originally introduced by Bar-Yehuda and Even [1985]. However, as we show, many approximation algorithms use LRTs implicitly.

A key point of LRT is the following Lemma 1 [Bar-Yehuda and Even 1985], where it is shown that any partition of the weight function $w : V \to \mathbb{R}$ results in two instances of WVC, from the optimal solutions of which one can derive a lower bound for the value of the optimal solution of the initial VC-instance.

LEMMA 1 [Bar-Yehuda and Even 1985]. *Let $G = (V, E)$ be a graph and let w, w_1, w_2 be weight functions on V such that $\forall v \in V$, $w(v) \geq w_1(v) + w_2(v)$. Let V^*, V_1^*, and V_2^* be optimal covers of G with respect to w, w_1, and w_2. Then $w(V^*) \geq w_1(V_1^*) + w_2(V_2^*)$.*

Let us now see how the local ratio lemma can be applied to give efficient approximation techniques for WVC.

Consider a graph $G = (V, E)$ and a weight function w, (G, w) being an instance of WVC; consider also an unweighted graph $G[V'] = G' = (V', E')$ induced by a subset $V' \subseteq V$; let n' be the order of G'; let τ' be the cardinality of an optimal vertex covering on G', and let $\rho_{G'} = n'/\tau'$. Let $\mathcal{A}$ be an approximation algorithm for WVC and $\rho_{\mathcal{A}}$ its approximation ratio. Consider now Algo-

Algorithm 8

begin
find a subgraph $G'_I = (V'_I, E'_I)$ of G isomorphic to G';
$\varepsilon \leftarrow \min \{w(v): v \in V'_I\}$;
for $u \in V$ **do**
  if $u \in V'_I$ **then** $w(u) \leftarrow w(u) - \varepsilon$;
    else $w(u) \leftarrow w(u)$;
  fi
od
call $\mathcal{A}$ to find a solution Λ' for the (modified) instance (G, w);
consider Λ' as the solution of the initial instance of WVC;
end.

Algorithm 9

begin
for every subgraph $G'_I = (V'_I, E'_I)$ of G isomorphic to some $G' \in \mathcal{F}$ **do**
  $\boldsymbol{\varepsilon} \leftarrow$ **min**$\{w(u) : u \in V'_I\}$;
  for every $v \in V'_I$ **do** $w(v) \leftarrow w(v) - \varepsilon$ **od**
od
$\Lambda_1 \leftarrow \{v : w(v) = 0\}$;
$V \leftarrow V \setminus \Lambda_1$;
call $\mathcal{A}$ to **find** a solution Λ_2 for the (modified) instance (G, w);
consider $\Lambda' = \Lambda_1 \cup \Lambda_2$ as the solution of the initial instance of WVC;
end.

rithm 8, where we suppose that the graph G' is fixed in advance, and let us denote by ρ its approximation ratio.

The approximation behavior of Algorithm 8 is characterized in Theorem 7, called the *local ratio theorem* in Bar-Yehuda and Even [1985] (and in Motwani [1993]).

THEOREM 7 [Bar-Yehuda and Even 1985]. $\rho \leq \max \{p_{G'}, \rho_{\mathcal{A}}\}$.

We now consider Algorithm 9, an extension of Algorithm 8. Let $\mathcal{F}$ be a finite family of graphs, let $\mathcal{A}$, $\rho_{\mathcal{A}}$, and $\rho_{G'}$ be as previously; finally, let ρ be the approximation ratio of Algorithm 9.

Corollary 1 characterizes the approximation efficiency of Algorithm 9; in Bar-Yehuda and Even [1985], this corollary is called the *local ratio corollary*.

COROLLARY 1 [Bar-Yehuda and Even 1985]. $\rho \leq max\{max\{\rho_{G'} : G' \in \mathcal{F}\}, \rho_{\mathcal{A}}\}$.

Many algorithms can be seen as particular applications of Algorithm 9 under particular choices of the family $\mathcal{F}$. For instance, Algorithm 2 in Section 2, and more particularly its variant for WVC, is such an application, where the family $\mathcal{F}$ is the family of edges of the instance of WVC.

In Bar-Yehuda and Even [1985] also, a $[2 - (1/n^{1/2})]$-polynomial time approximation algorithm for WVC is presented, where $\mathcal{F}$ is the set of all triangles of the WVC instance.

Moreover, the combination of Algorithm 9 together with Algorithm 7 (if we consider Algorithm 7 as the Algorithm $\mathcal{A}$ called by the Algorithms 8 and 9) seems to be a very good idea, since this combination has produced the better known approximation results for WVC.

Plainly, the idea of using this combination is due to Bar-Yehuda and Even [1985]. The key point is to use Algorithm 9 to eliminate, in a systematic way, all small-length odd cycles in an instance G of WVC.

Let us remark here that given an integer k, one can find all odd cycles of size at most $2k - 1$ in the following way: perform a breadth-first search (BFS; [Aho et al. 1975; Even 1979]) on G; for each vertex v, number the levels of BFS-subtree rooted at v, v being on level 1; if there exists an edge uw linking two vertices on level k, then the cycle consisting of the BFS-path $v - \ldots - u$, the edge uw, and the BFS-path $w - \ldots - v$ is an elementary odd cycle of length $2k - 1$.

Let us give a sketch of the method of Bar-Yehuda and Even [1985], this method constituting, as we have already mentioned, the more efficient LRT for approximating WVC.

Definition 4. Consider a graph $G = (V, E)$ of order n with vertex weight function w and a positive integer k; G is called proper (or k-proper, [Motwani 1993]) if the following conditions hold:

Algorithm 10. Algorithm k_PROPER.

begin
  $\Lambda_k \leftarrow \emptyset$;
  while $V \neq \emptyset$ **do**
    $v \leftarrow \text{argmax}_{u \in V}\{w(u)\}$;
    perform a BFS in G starting from v;
    for $i = 0, \ldots, k$ **do** $l_i = v$ $\{v : v$ is in BFS-level $i\}$ **od**;
    for $0 \leq t \leq \lfloor k/2 \rfloor$, $t \in \mathbb{N}$ **do** $B_{2t} \leftarrow \cup_{i=0}^{t} l_{2i}$ **od**
    for $0 \leq t \leq \lfloor (k-1)/2 \rfloor$, $t \in \mathbb{N}$ **do** $B_{2t+1} \leftarrow \cup_{i=0}^{t} l_{2i+1}$ **od**
    $f \leftarrow \min\{s : w(B_s) \leq (2k-1) w(B_{s-1})$;
    $\Lambda_k \leftarrow \Lambda_k \cup B_f$;
    $V \leftarrow V \setminus (B_f \cup B_{f-1})$;
    $G \leftarrow G[V]$;
  od
end.

(i) $(2k-1)^k \geq n$;
(ii) G does not contain odd circuits of length smaller than or equal to $2k-1$;
(iii) the weight of an optimal vertex cover of G is at least the half the sum of weights of V.

Algorithm 10 (k_PROPER) finds a vertex covering Λ_k in a k-proper graph [Bar-Yehuda and Even 1985], and Lemma 2, the proof of which is found in Bar-Yehuda and Even [1985], describes the properties satisfied by Algorithm 10.

LEMMA 2 [Bar-Yehuda and Even 1985]. *Algorithm* 10 *satisfies the following properties.*

(*i*) *in every application of the* while *loop, $f \leq k$ and, moreover, B_{f-1} is an independent set of the surviving graph G;*
(*ii*) *Λ_k is a vertex covering for G, the total weight of which is at most* $[1 - (1/2k)]w(V)$;
(*iii*) *the approximation ratio of Algorithm 10 is at most 2 − (1/k), and its time complexity is* $O(n\log n + |E|)$.

Algorithm 11. w is the weight-system of G after the end of the outer **for** loop of algorithm k_PROPER.

begin
  find k the smallest integer such that $(2k-1)^k \geq n$;
  for $1 \leq i \leq k-1$ **do** $\mathcal{F} \leftarrow \{C : C$ is an odd cycle of size $i\}$ **od**
  call algorithm 9 on G to obtain the set Λ_1 of vertices of zero weights;
  $G \leftarrow G[V \setminus \Lambda_1]$;
  call algorithm 7 on (G, w) to obtain the sets C_{BF}^1 and C_{BF}^2;
  call algorithm 10 on the weighted instance $G[C_{BF}^1]$ to obtain the set Λ_k;
  $\Lambda = \Lambda_k \cup C_{BF}^2 \cup \Lambda_1$;
end.

We can now specify the overall algorithm (Algorithm 11) proposed in Bar-Yehuda and Even [1985], and revisited in Motwani [1993], and characterize its approximation performance (Theorem 8, proved in Bar-Yehuda and Even [1985]).

THEOREM 8 [Bar-Yehuda and Even 1985]. *Algorithm* 11 *is an* $O(\max\{n^{5/3}|E|^{2/3}, n|E|\log n\})$, $[2 - (1/k)]$-*approximation algorithm for WVC. For VC, its complexity is of* $O(n|E|)$.

It is easy to see that the way k has been chosen in Algorithm 11 implies that the approximation ratio of Algorithm 11 is always smaller than $2 - [(\log\log n)/\log n]$. This ratio, even if it is bounded by 2, induces a significant improvement when dealing with real VC-instances; for example, when $n = 10^{12}$, this ratio is smaller than 1.9.

5.2.2 *The Monien and Speckenmeyer Algorithm.* In the same year Bar-Yehuda and Even published the result we have just seen, two other researchers, Monien and Speckenmeyer [1985], independently devised another approximation algorithm for (unweighted) VC with the same approximation ratio. The underlying idea of this algorithm is quite similar to the one in Bar-Yehuda and Even [1985], at least in what concerns

Algorithm 12

begin
$\Lambda' \leftarrow \emptyset$;
find the set $\mathscr{C}$ of all short cycles of G (for example using BFS);
for every cycle $C \in \mathscr{C}$ **do**
$\Lambda' \leftarrow \Lambda' \cup C$;
$V \leftarrow V \setminus C$;
od
let $G = (V, E)$ be the survived graph after the execution of the for loop;
call algorithm 7 on G to obtain sets C^1_{BF} and C^2_{BF};
$\Lambda' \leftarrow \Lambda' \cup C^2_{BF}$;
$V \leftarrow V \setminus (C^1_{BF} \cup C^2_{BF})$;
let $G = (V, E)$ be the survived graph;
compute an independent set S' for G;
$\Lambda' \leftarrow \Lambda' \cup (V \setminus S')$;
end.

the removal of all odd size short cycles, the removal common in both methods.

Whenever an odd short cycle is discovered, its vertices are added to the covering and removed from the graph, this procedure being repeated until all odd short cycles have been eliminated; after Monien and Speckenmeyer [1985] use Algorithm 7 to obtain sets C^1_{BF} and C^2_{BF}, they add C^2_{BF} in the vertex covering and remove $C^1_{BF} \cup C^2_{BF}$ from G; finally, they polynomially compute an independent set for the surviving instance (and, generally, for graphs without odd short cycles) and put its complement in the vertex covering.

Algorithm 12 is a sketch of the VC-algorithm presented in Monien and Speckenmeyer [1985].

We simply quote here Hochbaum's [1983] algorithm for WVC and we discuss it in Section 10.1. But we have to mention that she is the first author who has shown how we can use the Nemhauser-Trotter method to obtain efficient bounds for some combinatorial problems, more particularly for WVC and WIS.

PART III. SET PACKING

Let us note that, since IS and SP are approximate equivalent, a result, similar to the one of Theorem 2, can be immediately deduced for SP (an "autonomous" proof[9] of the result of Theorem 2 for SP proving the existence of a product construction working for SP is given in Paschos [1993]).

Algorithm 13. The greedy algorithm for SP.

begin
$S' \leftarrow \emptyset$;
repeat
$\mathscr{S}_i \leftarrow \mathrm{argmax}_{S_j \in \mathscr{S}}\{w(S_j)\}$;
$\mathscr{S}' \leftarrow \mathscr{S}' \cup \{S_i\}$;
$\mathscr{S} \leftarrow \mathscr{S} \setminus (\{S_i\} \cup \{S_j : S_i \cap S_j \neq \emptyset\})$;
until $\mathscr{S} = \emptyset$;
end.

6. THE GREEDY ALGORITHM

We first discuss the approximation performance of the natural greedy Algorithm 13 for WSP. In what follows, we denote by m the cardinality of the set $\cup_{S_i \in \mathscr{S}} S_i$. We note here that the characteristic bipartite graph B (Definition 1) constructed for SC can be defined also for SP by considering as ground set $C = \cup_{S_i \in \mathscr{S}} S_i$.

THEOREM 9. *Algorithm* 13 *is an* $O(n^2)$, $(1/\Delta)$-*approximation algorithm for SP.*

PROOF. *Let* S_j be the set added in $\mathscr{S}'$ during the jth iteration of Algorithm 13, and let $\mathscr{S}_j$ be the subfamily removed from $\mathscr{S}$ during the jth iteration.

If we denote by $\mathscr{SP}$ the optimal solution of SP, then $|\mathscr{SP} \cap \mathscr{S}_j| \leq \Delta$. The arguments: let $S_j = \{c_{j_1}, \ldots, c_{j_{|S_j|}}\}$ and let us denote by $\mathscr{S}_j|_{c_j}$ the subset of $\mathscr{S}_j$ interesecting S_j in c_{j_i} (note that the sets $\mathscr{S}_j|_{c_j i}$, $i = 1, \ldots |S_j|$ may be not disjoint); for every set packing $\hat{S}$, only one among the members of every $\mathscr{S}_j|_{c_{ji}}$ can be included in $\hat{S}$; thus, the inclusion of S_j in $\mathscr{S}'$ performed by Algorithm 13 excludes at most $|S_j| \leq \Delta$ sets which, in the best

[9] Of course this proof has only mathematical and no practical character, given that SP is not polynomially constant approximable unless P = NP.

case, would be included in the optimal solution.

Moreover, since Algorithm 13 selects the set of maximum weight in the surviving instance, $w(S_k) \leq w(S_j)$, $S_k \in \mathcal{SP} \cap \mathcal{S}_j$. Therefore, in the best case where $|S_j|$ among the excluded vertices would belong to the optimal solution (let us denote this set by $\mathcal{SP}|_{S_j}$), we would have $w(S_j) \geq (1/|S_j|)w(\mathcal{SP}|_{S_j}) \geq (1/\Delta)w(\mathcal{SP}|_{S_j})$.

Summation over all steps of Algorithm 13 induces that the quantity $1/\Delta$ also constitutes a bound for the whole approximation ratio.

The construction of G_S takes, at worst, $O(n^2)$; also, the total execution time of the **repeat** loop is of $O(E(G_S))$. However, for the choice of maximum-weight set, we can suppose that the vertices are pre-ordered following their weights with a cost of $O(n \log n)$; since there is no weight-modification during the algorithm's execution, the research of the maximum-weight vertex is performed in constant time. Thus the total complexity of Algorithm 13 is of $O(n^2)$. □

Let us note that Hochbaum [1983], based upon Definition 3 and using a nice approximation result for WIS in $(\Delta + 1)$-claw-free graphs,[10] gives a graph-theoretic interpretation of Algorithm 13 and proves the same approximation bound as the one of Theorem 9.

We now prove that the bound $1/\Delta$ can be attained. Let us consider an instance for SP consisting first of a family $\mathcal{S}' = \{S_1, S_2, \ldots, S_{n/\Delta}\}$ of n/Δ mutually disjoint sets, all of cardinality Δ (we suppose that n is a multiple of Δ^2).

Without loss of generality, we can suppose that the ground set, formed by the union of these sets, is the set $C = \{1, 2, \ldots, n\}$, and that $S_i = \{(i-1)\Delta + 1, (i-1)\Delta + 2, \ldots, i\Delta\}$, $i = 1, \ldots, n/\Delta$.

We then complete the instance by adding the family $\mathcal{S}'' = \{S_{(n/\Delta)+1}, \ldots, S_{(n/\Delta)+(n/\Delta^2)}\}$ where:

$$S_{(n/\Delta)+1} = \{1, \Delta + 1, \ldots, (\Delta - 1)\Delta + 1\},$$

$$S_{(n/\Delta)+2} = \{\Delta^2 + 1, \Delta(\Delta + 1) + 1, \ldots, \Delta(2\Delta - 1) + 1, \ldots,$$

$$\vdots$$

$$S_{(n/\Delta)+(n/\Delta^2)} = \{n - \Delta^2 + 1, n - \Delta^2 + \Delta + 1, \ldots, n - \Delta + 1\}.$$

Then it is easy to see that when searching for a set packing on $\mathcal{S} = \mathcal{S}' \cup \mathcal{S}''$ (where the maximum set cardinality is Δ), since there is no selection condition on the set cardinalities, Algorithm 13 could provide set $\mathcal{S}''$ as the solution of the problem; on the other hand, the optimal solution is exactly the set $\mathcal{S}'$, and therefore an approximation ratio equal to $1/\Delta$ is attained by the algorithm.

7. IMPROVEMENT OF THE SET PACKING APPROXIMATION RATIO

Recently, Yu and Goldschmidt [1993] have proposed Algorithm 14 approximating, in $O(n^3)$, independent sets in Δ-claw-free graphs within a ratio of $2/\Delta$.

A key notion in Yu and Goldschmidt [1993] is the one of locally optimal independent set. In fact, an independent set S is locally optimal if, for every $u \in S$ and for every independent set $S' \subseteq \Gamma(u)$, $|S'| \leq |S \cap \Gamma(S')|$ (note that $S' \subseteq V \setminus S$). The grounds for this definition are such that if S is not locally optimal, then there exists a set $S' \subseteq \Gamma(u)$ of cardinality strictly greater than the one of the set $S \cap \Gamma(S')$ and, consequently, we can form an independent set larger than S and equal to $(S \cup S') \setminus (S \cap \Gamma(S'))$.

Let us remark here that a locally optimal independent set can be found in polynomial time for the class of graphs for which a maximum independent set in $\Gamma(v)$, $v \in V$, can be polynomially derived; this class includes, more particularly, the class of k-claw-free graphs

[10] A graph is called k-claw-free if it contains no independent set of k vertices, all adjacent to a common vertex.

Algorithm 14

begin
  $S' \leftarrow \emptyset$;
  for every $v \in V \setminus S'$ **do** $\mathbf{\Gamma}_v \leftarrow \Gamma(v) \cap S'$
   od
  $\mathbb{L} \leftarrow \{(u, v) : u, v \notin S' \bigvee uv \notin E\}$;
  mark all elements of *IL* by new;
  if all elements of $\mathbb{L}$ are marked old **then**
   output S' **fi**
  pick a new element (u, v) of *IL*;
  if $|\mathbf{\Gamma}_u| > 1$ or $|\mathbf{\Gamma}_v| > 1$ or $|\Gamma(u) \cap \Gamma(v) \cap S'| > 1$
   then
     mark (u, v) **old**;
     go to *;
   else
     $S' \leftarrow (S' \setminus (\Gamma(u) \cap \Gamma(v) \cap S')) \cup \{u, v\}$;
     remove from $\mathbb{L}$ all pairs containing u or v;
     $w \leftarrow \Gamma(u) \cap \Gamma(v) \cap S'$;
     $\mathbb{L} \leftarrow \mathbb{L} \cup \{(\mathrm{x}, \mathrm{w}) : \chi \notin \mathrm{S}' \bigvee \chi\mathrm{w} \notin \mathrm{E}\}$;
     for $(p, q) \in \mathbb{L}$ **do**
      if $\{u, v\} \cap \Gamma(p) \neq \emptyset$ or $\{u, v\} \cap \Gamma(q) \neq \emptyset$ then mark $(p. q)$ new **fi**
     od
  fi
  maximalize S' by adding vertices of $V \setminus S'$ not in $\Gamma(S')$;
  go to *;
end.

for every fixed constant k. For the case where k is not a fixed constant, Yu and Goldschmidt [1993] propose Algorithm 14 for finding a quasilocally optimal independent set.

Theorem 10 characterizes the performance of Algorithm 14 in Δ-claw-free graphs.

THEOREM 10 [Yu and Goldschmidt 1993]. *Let α be the cardinality of a maximum independent set of G in a Δ-claw-free graph of order n, and α′ be the cardinality of a quasilocally optimal independent set derived from Algorithm 14. Then* $\alpha' \geq \max \{(2/\Delta)\alpha, (\Delta/2)\alpha - n[(\Delta/2) - 1]\}$.

Even if Yu and Goldschmidt [1993] do not make this remark, the proof of Hochbaum [1983] for the ratio of SP remains valid; hence, Algorithm 15, constitutes a $[2/(\Delta + 1)]$-approximation for SP (recall that G_S is $(\Delta + 1)$-claw-free).

Algorithm 15. A $[2/(\Delta + 1)]$-approximation algorithm for SP; by $\mathcal{S}'$, we denote the SP-solution.

begin
  apply definition 3 on the SP-instance;
  let G_S the obtained graph;
  apply algorithm 14 on G_S;
  $\mathcal{S}'$ is the family the sets corresponding to the vertices of S';
end.

PART IV. INDEPENDENT SET

For IS's approximation, the main open problem (given IS's nonconstant approximability) is to determine the best approximation ratio. Since such a ratio necessarily depends on input parameters, there exist three parameters commonly used to express the approximation behavior of IS-algorithms; the order n of the graph, its maximum degree, and the graph's average degree.

Concerning ratios depending on n, the major question is if an independent set of size $\Omega(n^{1/\kappa})$ can be found in polynomial time and, consequently, if one can approximate this problem within the ratio $O(n^{(1-\kappa)/\kappa})$, for a fixed constant κ (see Bellare et al. [1995] and Shmoys and Tardos for some interesting remarks about this question).

We quote here the paper of Bellare and Sudan [1994], where it is proved that IS cannot be approximated within n^k for a $k \geq (1/6) - \eta$, unless P = NP. Up to now, despite the impressive results produced during the last couple of years, researchers have been unable to prove something about these questions.

Let us, for the history, mention that the Lovász θ-function,[11] introduced in

[11] For a graph $G = (V, E)$, an "orthonormal labelling" is an assignment of a unit vector $\vec{a}_i$ (in a Euclidean space) to each vertex $v_i \in V$, such that if $v_i \neq v_j$ and $v_iv_j \notin E$, then $\vec{a}_i \cdot \vec{a}_j = 0$; the θ-function $\theta(G)$ equals the minimum, over all unit vectors $\vec{b}$ and all orthonormal labelings of G, of the quantity $\max_{v_i \in V}\{1/(\vec{b} \cdot \vec{a}_i)^2\}$; moreover, $\alpha(G)$

Lovász [1979], had been conjectured to approximate IS within $n^{1/2}$ [Boppana and Halldórsson 1992], but this conjecture has been refuted by Feige [1995].

Concerning ratios depending on graph degrees, the main open problems are: "which is the largest constant c for which IS-ratio is greater than, or equal to, c/Δ?" and "do there exist algorithms attaining ratios $o(\Delta)/\Delta$?"

8. INDEPENDENT SET AND LINEAR PROGRAMMING

8.1 The Nonweighted Case

A general instance of IS defined by a graph $G = (V, E)$ can be written as a 0–1 linear problem as follows

$$\mathrm{IS}(G) = \begin{cases} \max & \vec{1}_n \cdot \vec{x} \\ & A \cdot \vec{x} \leq \vec{1}_{|E|} \\ & \vec{x} \in \{0, 1\}^n, \end{cases}$$

where A is the edge-vertex incidence matrix of G.

Let us denote by IS_r the following relaxed version of IS

$$\mathrm{IS}_r(G) = \begin{cases} \max & \vec{1}_n \cdot \vec{x} \\ & A \cdot \vec{x} \leq \vec{1}_{|E|} \\ & \vec{x} \geq \vec{0}_n. \end{cases}$$

The dual of IS_r denoted by EC_r is

$$\mathrm{EC}_r(G) = \begin{cases} \min & \vec{1}_{|E|} \cdot \vec{x} \\ & A^T \cdot \vec{x} \geq \vec{1}_n \\ & \vec{x} \geq \vec{0}_{|E|}, \end{cases}$$

where this problem is denoted by EC_r in order to indicate that it is the relaxed version of the minimum edge covering problem (EC) which has the linear programming formulation:

$$\mathrm{EC}(G) = \begin{cases} \min & \vec{1}_{|E|} \cdot \vec{x} \\ & A^T \cdot \vec{x} \geq \vec{1}_n \\ & \vec{x} \in \{0, 1\}^{|E|}. \end{cases}$$

$\leq \theta(G)$ and the latter can be computed in polynomial time at an arbitrary precision [Grötschel et al. 1981].

Remark that $\vec{0}_n$ and $\vec{1}_{|E|}$ are feasible for IS_r and EC_r, respectively. As these dual instances have their respective constraint sets nonempty, they have the same optimal value.

Then the following inequalities hold[12]

$$\alpha(G) \leq v(\mathrm{IS}_r(G)) = v(\mathrm{EC}_r(G)) \leq v(\mathrm{EC}(G)).$$

Let us call the difference $v(\mathrm{EC}(G)) - \alpha(G)$, the *discrete duality gap* (see Demange and Paschos [1996c] for more details on this matter).

Moreover, observe that a similar schema holds also for VC by substituting VC for IS and *maximum matching* for EC.

Let us now try a combinatorial interpretation of the discrete duality gap $v(\mathrm{EC}(G)) - \alpha(G)$. Consider the sets F and X_C defined in the introduction (Section 1.4). By their definition, these sets, consequently the numbers f and g, depend not only on M but also, for a fixed matching, on the sets C^* (and S^*) considered.

In fact, for every graph G,

$$\tau(G) = m + (f + g)$$

$$|S^*| = \alpha(G) = n - m - (f + g).$$

Since $v(\mathrm{EC}(G)) = n - m$ [Berge 1973], the second of the preceding expressions gives: $v(\mathrm{EC}(G)) - \alpha(G) = f + g$. Consequently, the discrete duality gap $f + g$ depends only on G.

The case where a graph has zero duality gap ($f + g = 0$) generates a famous and well-studied class of graphs, the König-Egervary graphs (KE-graphs). For KE-graphs, the problem of recognizing them as well as the one of finding a maximum independent set are both polynomial [Deming 1979].

If we relax the constraint $f + g = 0$ by allowing a positive discrete duality gap bounded above, we get the following

[12] Recall that, following our notations, $v(\mathrm{IS}(G)) = \alpha(G)$.

result, the proof of which is given in Demange and Paschos [1996c].

PROPOSITION 1 [Demange and Paschos 1996c]. *Consider a graph* $G = (V, E)$ *such that* $0 \leq \tau(G) - m = f + g \leq \kappa$. *Then*

(*i*) *if* κ *is a fixed positive integer constant, there exists an exact polynomial algorithm for maximum independent set problem in G;*
(*ii*) *otherwise (*κ *depends on n), there exists a polynomial time approximation algorithm (having* κ *among its input parameters) providing an independent set of cardinality, at least equal to* $\lceil n/[2(\kappa + 1)] - 2\rceil$.

From an immediate application of part (i) of Proposition 1, we obtain the following corollary.

COROLLARY 2. *Given a fixed positive constant* κ, *deciding if a graph G verifies* $0 \leq f + g \leq \kappa$ *is polynomial.*

8.2 The Weighted Case

The weighted versions of WIS and WIS_r are defined as follows:

$$\text{WIS}(G) = \begin{cases} \max & \vec{w}_n \cdot \vec{x} \\ & A \cdot \vec{x} \leq \vec{1}_{|E|} \\ & \vec{x} \in \{0, 1\}^n \end{cases}$$

$$\text{WIS}_r(G) = \begin{cases} \max & \vec{w}_n \cdot \vec{x} \\ & A \cdot \vec{x} \leq \vec{1}_{|E|}. \\ & \vec{x} \geq \vec{0}_n. \end{cases}$$

Let us revisit the results presented in Section 5.1. Naturally, the same results hold for WIS, and the following Theorem 11 is the rephrasing of Theorem 6 for the case of WIS (and equally for IS, up to the replacement of the vector $\vec{c}$ by the unit vector).

THEOREM 11. *The basic feasible solution of* WIS_r *assigns to the vertices of V basic feasible values drawn from the set* $\{0, 1/2, 1\}$, *thus creating a partition of V to three sets P, Q, and R, corresponding to these values. Then there exists at least a maximum independent set S of G such that* $P \subseteq S$ *and* $R \subseteq V \setminus S$ *(the minimum vertex cover of G associated with S).*

We have seen that the KE-graphs are the ones for which the discrete duality gap $v(\text{EC}(G)) - \alpha(G)$ is equal to 0. However, $v(\text{IS}_r(G)) - \alpha(G) \leq v(\text{EC}(G)) - \alpha(G)$; consequently, the KE-graphs are the ones for which $v(\text{IS}_r(G)) - \alpha(G) = 0$.

Of course, the question of the difference between the value of IS and the one of its linear relaxation can be also posed for WIS.

Bourjolly et al. [1984] introduce the class of graphs where $v(\text{WIS}(G)) = v(\text{WIS}_r(G))$; they call it the class of $\vec{b}$-KE-graphs, and they conceive an $O(n^{2.5})$ exact WIS-algorithm for this class.

As in the previous section where we relaxed the condition $v(\text{EC}(G)) - \alpha(G) = 0$ by allowing this gap to be bounded above by a fixed positive constant, in the weighted case one can also follow the same thought process.

In Demange and Paschos [1996a], we relax the condition $v(\text{WIS}(G)) = v(\text{WIS}_r(G))$ by a less restrictive one and we prove that, in the class of graphs defined by this relaxed information, WIS remains polynomial.

Let us revisit, once more, the work of Nemhauser and Trotter [1975]. There it is also shown how, given an instance $G = (V, E)$ of WIS, one can:

—either determine (in $O(n^{4.5})$) a partition of V into sets P, Q, and R (with feasible values 1, 1/2, and 0, resp.), such that $Q = \emptyset$ and so WIS(G) is solved to optimality;
—or reduce G to a subgraph where the unique optimal solution for WIS_r is formed by assigning to all of its vertices the value 1/2.

This can be done by Procedure 1, strongly inspired by Nemhauser and Trotter [1975].

In what follows, WIS_r-solution refers to the Nemhauser-Trotter method [1975].

Procedure 1

```
begin
    solve WIS_r(G) to obtain sets P, Q, r;
    stop ← flase;
    while stop = false and Q ≠ ∅ do
        N' ← Q;
        repeat
                choose v_0 ∈ N';
                N' ← N'∖{v_0};
                N ← Q∖({v_0}∪ Γ
                  (v_0));
                solve WIS_r(G[N]);
{*R', P', Q' are the basic feassible sets of
  Q∖({v_0}∪ Γ(v_0))*}
            until w⃗ · (1⃗_{P'} + (1/2)1⃗_{Q'} +
              1⃗_{v_0}) = (1/2)w⃗ · 1⃗_Q or N' = ∅;
            if w⃗ · (1⃗_{P'} + (1/2)1⃗_{Q'} + 1⃗_{v_0}) =
                (1/2)w⃗ · 1⃗_Q
              then
                  P ← P ∪ P' ∪ {v_0};
                  R ← R ∪ R';
                  Q ← Q';
              else stop ← true;
            fi
    od
end;
```

Algorithm 16. Algorithm STABLE.

```
begin
    if κ ≤ 0 then call the procedure of
      [16] to optimally solve WIS(G);
      else
          choose v_i v_j ∈ E;
          for x ∈ {i, j} do
            V_x ← V∖{v_x};
            G_x ← G[V_x];
            call procedure 1 on G_x;
{*P_x, Q_x, R_x are the obtained sets *}
          if Q_x = ∅ then S_x ← ∅;
              else
                V'_x ← Q_x;
                G'_x ← G[V'_x];
                κ ← κ − (1/2);
                S_x ← STABLE(κ, G'_x);
          fi
      od
      x̂ ← argmax_{x∈{i,j}}{w⃗ · (1⃗_{P_x} + 1⃗_{S_x})};
      S ← P_x̂ ∪ S_x̂;
    fi
end.
```

LEMMA 3 [Nemhauser and Trotter 1975]. *Given a graph G, the* $O(n^{4.5})$ *Procedure* 1 *determines a partition of V into sets P, Q, and R, and a solution for* $WIS_r(G)$ *corresponding to this partition, such that*

—*either* $Q = \emptyset$ *and so WIS(G) is solved to optimality* $(v(WIS(G)) = \Sigma_{v_i \in P} w_i)$;

—*or the unique optimal solution for* $WIS_r\ (G[Q])$ *is formed by assigning, to all of the vertices of Q, the value* 1/2.

Let us now introduce a new class of graphs generalizing the one of $\vec{b}$-KE-graphs treated in Bourjolly et al. [1984].

Definition 5. Let κ be a fixed positive constant. The class κ-KE is defined as the class of graphs $G = (V, E)$, of order n, such that:

(i) the solution of $\text{WIS}_r(G)$ consisting of assigning value 1/2 on all of the vertices of V is unique for $\text{WIS}_r(G)$; and

(ii) $(1/2)\vec{w} \cdot \vec{1}_n - \kappa \le v(\text{WIS}(G)) \le (1/2)\vec{w} \cdot \vec{1}_n$.

Theorem 12 (the proof is found in Demange and Paschos [1996a]) proves that WIS is polynomial for the class of κ-KE-graphs.

THEOREM 12 [Demange and Paschos 1996a]. *Let* κ *be a fixed positive constant and let* $G = (V, E)$ *be a* κ*-KE-graph of order n. Algorithm* 16 *determines a maximum-weight independent set for G in* $O(n^{4.5})$*. Also, it polynomially decides if a graph is* κ*-KE.*

Consider a constant κ and a graph $G = (V, E)$ such that $v(\text{WIS}(G)) \ge v(\text{WIS}_r(G)) - \kappa$. Procedure 1 allows us to partition V into three sets R, P, and Q in such a way that $v(\text{WIS}(G)) = v(\text{WIS}(G[Q])) + \vec{w} \cdot \vec{1}_P$ and, furthermore, $v(\text{WIS}_r(G)) = v(\text{WIS}_r(G[Q])) + \vec{w} \cdot \vec{1}_P$. Moreover, in order to construct an optimal solution of WIS(G), one has simply to complete an optimal solution of WIS($G[Q]$) with the vertices of P. The preceding allows us to establish that $v(\text{WIS}(G[Q])) \ge v(\text{WIS}_r(G[Q])) - \kappa$;

furthermore, using Lemma 3 one can be sure that the graph $G[Q]$ is κ-KE. Then the application of Theorem 12 results in the following Theorem 13, the complete proof of which is found in Demange and Paschos [1996a].

THEOREM 13 [Demange and Paschos 1996a]. *Consider the class of graphs* $G = (V, E)$ *satisfying* $v(WIS(G)) \geq v(WIS_r(G)) - \kappa$, *where* κ *is a fixed constant. Then the problems:*

(*i*) *decide if a graph G belongs to this class, and*
(*ii*) *solve WIS in this class are both polynomial.*

Although the proof of the similar result in the nonweighted case of Section 8.1 resulted from a combinatorial interpretation leading to the consideration of an upper bound of the quantity $\alpha(G) - v(\mathrm{IS}_r(G))$, such a combinatorial interpretation for the weighted case, or, more generally, for the primal–dual approach, is much less natural. So, in Demange and Paschos [1996a], we give straightforward proofs for Theorems 12 and 13, both based upon linear programming arguments.

A Final Remark. Let us finish this section by a remark on the approximability of IS. Recall the result of Arora et al. [1992] on the nonconstant-ratio approximation of IS and observe that, as we have brought to the fore in this section, the "hard" (from an approximability point of view) instances of IS are the ones where, for a fixed integer constant κ, $\alpha(G) \leq (n/2) - \kappa$ ("hard" in the sense that the rest of the IS-instances can be polynomially solved).

For $\alpha(G) \leq (n/2) - \kappa$ (κ increasing function of n), IS is very probably NP-complete. Is it also nonconstant approximable, or does there exist a fixed constant $\nu > 2$ such that IS in graphs with $(n/2) - \kappa \geq \alpha(G) \geq n/\nu$ admits a constant-ratio PTAA and, if yes, which is the least ν for which such a positive result holds? (In other words, is ν equal to 2, or is it strictly greater than 2?)

Algorithm 17. The greedy IS algorithm.

begin
  $S' \leftarrow \emptyset$;
  repeat
    $v_j \leftarrow \mathrm{argmin}_{v_i \in V}\{|\Gamma(v_i)|\}$;
    $S' \leftarrow S' \cup \{v_j\}$;
    $V \leftarrow V \setminus (\{v_j\} \cup \Gamma(v_j))$;
    delete from E all edges incident to $\{v_j\} \cup \Gamma(v_j)$;
    update the degrees of the vertices in V;
  until $V = \emptyset$;
end.

In Part V, we show that determining such a $\nu \neq 2$ plays a crucial role in the approximability of a minimum vertex covering, since it simultaneously allows us to define lower bounds on the approximation ratio of the problem and to improve the known approximation ratio; on the other hand, if $\nu = 2$, then no polynomial time approximation algorithm can guarantee an approximation ratio equal to $2 - \epsilon$ for any fixed positive constant ϵ.

9. UNWEIGHTED INDEPENDENT SET

9.1 Approximation Ratio of the Greedy IS-Algorithm.

The natural greedy Algorithm 17 has been considered for a long time to be the most efficient approximation algorithm for IS (this was also supported by the excellent experimental behavior of the algorithm).

In this section we survey the main results concerning Algorithm 17.

9.1.1 *Old (but Always Interesting) Results About the Greedy Algorithm.* For Algorithm 17, an approximation ratio at most Δ holds, because out of a vertex deleted in each iteration of the **repeat** loop, at most Δ ones can belong to the optimal solution.

Simon [1990] gives an approximation ratio equal to $1/(\Delta - 1)$ for Algorithm 17.

To prove this ratio, let us first remark that (by an elementary counting of the edges between a maximum independent set and its associated vertex covering), if we denote by δ the minimum graph degree, $\alpha(G) \leq n\Delta/(\Delta + \delta)$. If G is not regular, then Algorithm 17 will never choose to include in S' a vertex of degree Δ so, at each step the greedy algorithm removes, at most, Δ vertices; consequently, $|S'| \geq n/\Delta$ and the approximation ratio of Algorithm 17 is bounded below by $(\Delta + 2)/\Delta^2 \geq \Delta - 1$, for $\delta \geq 2$. On the other hand, if $\delta < 2$, it suffices to observe that Algorithm 17 optimally handles isolated and degree-one vertices. Lastly, if G is regular, the approximation ratio of the greedy algorithm is at least $2/(\Delta + 1) \geq 1/(\Delta - 1)$ for $\Delta \geq 3$; if $\Delta < 3$, the discussed algorithm optimally solves IS.

Let us remark that from Turán's [1941] theorem an approximation ratio $\mu + 1$ is admitted by Algorithm 17.

An approximation ratio equal to $\Delta + 1$ is immediately deduced from the works of Berge [1973] and Hajnal and Szemeredi [1970].

Even if such an approximation bound is not so interesting, the works of Berge [1973] and Hajnal and Szemeredi [1970] have the great interest of providing lower bounds for the stability number of a graph G.

In the same vein lie the works of Hopkins and Staton [1982], Fajtlowicz [1978], and Erdös [1959]. All these works obtain lower bounds for the quantity $\iota(G) = \alpha(G)/n$, which are functions of Δ and of the girth[13] of G.

Let us remark here that such bounds are, up to now, only very infrequently used to devise approximation strategies for IS (since a common thought process to well approximate the problem is rather to search for nontrivial upper bounds, at least for natural classes of graphs, than for lower ones).

However, there is a very interesting underlying idea, common to all these results, namely, there exist graphs with very large girth and yet very small stability number (with respect to n) and this fact could be efficiently exploitable to approximate IS (at least on classes of graphs).

Moreover, as we see in Part V, the approximability of IS in graphs with large $\iota(G)$ plays an important role in the approximability of VC from both positive and negative points of view.

9.1.2 *A Tight Bound for the Greedy Algorithm.* Halldórsson and Radhakrishnan [1994] performed a nice analysis of Algorithm 17. This analysis has produced the best known lower bound for $|S'|$ (the size of the greedy solution): $|S'| \geq [1 - \iota(G)(1 - \iota(G))]/[(1 - \iota(G))\Delta + 1]n$.

Next, by a very fine counting of the vertices and edges deleted during the execution of Algorithm 17, they form a system of linear and quadratic equalities and inequalities, the solution[14] of which determines an approximation ratio $|S'|/\alpha(G) \geq 3/(\Delta + 2)$ for the greedy IS-algorithm. This ratio is tight as proved in Halldórsson and Radhakrishnan [1994a].

In fact, consider a graph G, parametrized by an integer $\ell \geq 2$, consisting of k copies of a pair of subgraphs: a clique K_ℓ and an independent set S_ℓ, on ℓ vertices. Let us denote by (K_ℓ^i, S_ℓ^i), $i = 1, \ldots, k$, the pair of the ith copy. For all i, the vertices of K_ℓ^i and S_ℓ^i are completely connected, whereas the connections between S_ℓ^{i-1} and K_ℓ^i miss only a single matching. For such a graph, $\Delta = 3\ell - 2$. Moreover, observe that the minimum degree of this graph is the one of the vertices of S_ℓ^i, $i = 1, \ldots, k$, which is equal to the degrees of the vertices of K_ℓ^1, equal $2\ell - 1$.

Algorithm 17 could pick one of the vertices of K_ℓ^1, thus deleting all the vertices of S_ℓ^1 as well as the rest of the ones of K_ℓ^1; therefore the first pair will be deleted and the algorithm can behave identically for the rest of the pairs, until

[13] The size of the longest cycle of G.

[14] Let us note here that, to solve this system, they use the method of multipliers described by Chvatal [1983] and used by the same author for the analysis of Algorithm 1.

all the vertices are deleted. Such an execution will produce an independent set of size k, and the optimal solution of G equals $\cup_{i=1}^{k} S_{\ell}^{i}$, of size ℓk. In this case, the approximation ratio of Algorithm 17 is $1/\ell = 3/(\Delta + 2)$.

9.2 Efficient Approximations for an Independent Set via Ramsey Numbers

Boppana and Halldórsson [1992] propose a PTAA for IS, guaranteeing an approximation ratio $O(n/(\log n)^2)$, which is the best known ratio, function of n.

This algorithm is based upon the famous result of Ramsey:[15] "for any pair (s, t) of integers, there is an integer n for which every graph of order n contains either a clique K_s, or an independent set S_t."

Let us denote by $R(s, t)$ the minimal value of n for which Ramsey's result holds. The following upper bound (the best known) for $R(s, t)$: $R(s, t) \le C_{t-1}^{s-t+2}$ is due to Erdös and Szekeres [1935].

Let us denote by $r(s, t)$ the quantity C_{t-1}^{s-t+2} for all positive integers s and t (by convention, if one of s and t is negative or zero, then $C_{t-1}^{s-t+2} = 1$), and let $t_k(n) = \min\{t : r(k, t) \ge n\}$. Note that $t_k(n) \approx kn^{1/(k-1)}$ if $k \le 2 \log n$, and $t_k(n) \approx (\log n)/[\log(k/\log n)]$, otherwise.

Given a graph $G = (V, E)$, Algorithm 18 (with inputs a graph G of order n and a positive number k) developed in Boppana and Halldórsson [1992] and strongly inspired by the proof of the upper bound for $R(s, t)$ [Erdös and Szekeres 1935], finds, in time $O(|E|)$, either a clique K of size k, or an independent set S' of size t. Moreover, as it is proved in Boppana and Halldórsson [1992], $r(|K|, |S'|) \ge n$, so $|K||S'| \ge c(\log n)^2$, for some constant c.

Unfortunately, since one cannot a priori know which among K and S' Algorithm 18 will output, Boppana and Halldórsson [1992] had the elegant idea to

Algorithm 18. Algorithm RAMSEY.

begin
  $S' \leftarrow \emptyset$;
  $K \leftarrow \emptyset$;
  while $|V| > 1$ **do**
    choose $v \in V(G)$;
    $t \leftarrow t_k(n)$;
    if $|\Gamma(v)| \ge r(k - 1, t)$
      then
        $K \leftarrow K \cup \{v\}$;
        $G \leftarrow G[\Gamma(v)]$;
        $k \leftarrow k - 1$;
      else
        $S' \leftarrow S' \cup \{v\}$;
        $G \leftarrow G[V \setminus (\{v\} \cup \Gamma(v))]$;
  od
  $K \leftarrow K \cup V(G)$;
  $S' \leftarrow S' \cup V(G)$;
end.

influence the output by eliminating choice; that is, for the case of IS, if one removes all cliques of size k, the resulting independent set will verify $|S'| \ge t_k(n)$.

The combination of Algorithm 18 with a strategy of clique removal (performed in Boppana and Halldórsson [1992]) is described in Algorithm 19.

As Boppana and Halldórsson [1992] remark, it is somewhat difficult to find the value of k for which the approximation ratio of Algorithm 19 is the best possible. In order to remedy this obstacle, they embed in Algorithm 19 a kind of binary search (allowing us to define a "good" κ) originally proposed in Wigderson [1983] for minimum graph coloring. The thus-strengthened version of Algorithm 19 has complexity $O(|E|n \log n)$ and yields approximation ratio $O(n/(\log n)^2)$.

Finally, let us remark that if we consider Algorithm 19 on graphs with a "high" stability ratio, for instance, if $\iota(G) \ge (1/k) + \epsilon$, $\epsilon > 0$, then as Boppana and Halldórsson [1992] prove, its approximation ratio becomes $\Omega((\epsilon n)^{1/(k-1)})$.

[15] Using Ramsey's numbers to estimate independent sets sizes is an old and fruitful thought process; see for instance Staton [1979], or even Monien and Speckenmeyer [1985].

Algorithm 19

begin
    $(S', K) \leftarrow \text{RAMSEY}(G, k)$;
    while $|K| \geq k$ **do**
        $G \leftarrow G[V \setminus K]$;
        $(S', K) \leftarrow \text{RAMSEY}(G, k)$;
    od
end.

9.3 Algorithmic Schemas for an Independent Set

During the last couple of years, a great number of researchers have produced numerous and very worthy "positive" approximation results for IS.

The underlying key idea for the most of the recently proposed methods is, given an instance G of IS, the execution of several approximation (or exact) algorithms on several subgraphs of G (or on G itself). In what follows, we call such an algorithm (which mainly consists of combining the execution of different algorithms on subinstances of IS) an *algorithmic schema*.[16] The devised algorithmic schemas for IS undoubtedly produce improved approximation ratios, but aggravate (sometimes very heavily, as we see) complexities.

The development of the different algorithmic schemas for IS relies on a number of simple and natural ideas. A careful examination of the quasitotality of these schemas in the literature brings to the fore three types of such ideas:

—try to "separate" the input graph G in i components, $G_1, \ldots, G_i$, such that for the $i - 1$ first ones, a maximum independent set can be polynomially found (or, at least, can be approximated with high accuracy); if, furthermore, one can prove that these $(i - 1)$ maximum independent sets belong to at least an optimal IS-solution of G (the initial graph), then the approximation ratio of the proposed algorithmic schema on G will be greater than the ratio on the ith (last) component of the performed separation; this thought process has potential advantages: first, the approximation of IS in G_i could be easier (more efficient) than the approximation of IS in G itself, and second, a better (smaller) upper bound could hold for $\alpha(G_i)$ (Section 9.3.1);

—run an algorithm on G and next, perform local improvements of the provided solution; the main potential advantage of this type is that if we have a good (tight) evaluation of the solution found by the basic algorithm used by the schema and if, furthermore, we succeed to measure the impact well of the performed local improvements, then we can easily obtain an improved approximation ratio (Section 9.3.2);

—run an algorithm on G and stock the provided solution; then remove some "problematic" configurations from G (problematic in the sense that they contain small independent sets) and run several judiciously chosen IS-algorithms in the surviving instances, always stocking the obtained solutions; next retain the largest among the solutions stocked at the different steps of the schema (Section 9.3.3).

9.3.1 *Using the Nemhauser–Trotter Method.* In the case of IS, a very natural thought process conforming to the first of the above items is the preprocessing of G using the method of Nemhauser and Trotter [1975]. In fact, one can: solve (in polynomial time) IS_r in order to obtain sets P, Q, and R, stock P, call an IS-algorithm on $G[Q]$ and, finally, retain the union of P with the IS-solution of $G[Q]$ as approximate IS-solution. It is easy to see that, since solving IS_r can be done in polynomial time, if the algorithm operating on $G[Q]$ is also polynomial, then the whole process remains of polynomial complexity.

Let us note that this type of schema (including solution of IS_r) is originally

[16] Of course, Algorithm 19 could be seen as such a schema; but since all the discussed schemas in the sequel have ratios functions of degree, we have preferred to mention Algorithm 19 separately.

Algorithm 20

```
begin
    execute RIS on G to obtain sets P,
      Q, R;
    if Q = ∅
      then output P as IS-solution;
      else
        G ← G[Q];
        execute ISALGO on G[Q];
        let S′ the obtained
          IS-solution;
     fi
     S′ ← S′ ∪ P;
end.
```

proposed in Hochbaum [1983] (we speak in detail of this work in Section 10.1).

Algorithm 20, parametrized by algorithms RIS (solving IS_r) and ISALGO (which can be any PTAA for IS), formalizes the preceding discussion. If we note by $T_{RIS}(G)$ and by $T_{ISALGO}(G)$ the complexities of RIS and ISALGO, respectively, the whole complexity of Algorithm 20 is $\max\{T_{RIS}(G), T_{ISALGO}(G)\} = max\{n^{2.5}, T_{ISALGO}(G)\}$. Moreover, by Theorem 11, since P belongs to both an optimal IS-solution of G and to the solution constructed by Algorithm 20, the approximation ratio of this algorithm will be at least equal to one of algorithm ISALGO (operating on $G[Q]$). Moreover, $\alpha(G[Q]) \leq v(\mathrm{WIS}_r(G[Q]) = |Q|/2$. So a better bound than the trivial one, $|Q|$), can be used for $\alpha(G[Q])$.

In Haldorsson and Radhakrishnan [1994a] (see also Hochbaum [1983]), it is shown that if Algorithm 20 is parametrized by RIS and Algorithm 17, then its approximation ratio is bounded below by $5/(2\mu + 3)$.

Let us also note that in the vein of improving bounds for $\alpha(G)$, another idea could be to introduce the discrete duality gap, discussed in Section 8, which diminishes the value of $\alpha(G)$. This way has not led, up to now, to notable improvements [Paschos 1993].

9.3.2 *Performing Local Improvements.* Algorithm 21 is a specification of the local improvement schemas implied by the second item presented.

Algorithm 21

```
begin
    run ISALGO on G;
    let S′ be the obtained solution;
    S′ ← IMPROVE(S′);
end.
```

This algorithm is parametrized by algorithm ISALGO, which can be any PTAA for IS, and procedure IMPROVE, which is a procedure locally improving the solution provided by ISALGO. It is easy to see that if both ISALGO and IMPROVE are polynomial, then the overall complexity of algorithm 21 is polynomial.

In what follows, we give the two most interesting, up to now, instantiations of this schema.

"Small" improvements in law-degree graphs. The algorithm we discuss in this section (Algorithm 22) can be seen as an instantiation of the schema expressed by Algorithm 21 and has been proposed by Berman and Fürer [1994].

Before outlining this algorithm, let us define what in Berman and Fürer [1994] is called an *improvement*.

Definition 6. Let S' be an independent set of a graph $G = (V, E)$. Then:

—S^i is an improvement of S' iff $G[S^i]$ is connected, $S' \oplus S^i$ is also an independent set of G, and $|S' \oplus S^i| \geq S'$ (where operator $\oplus$ denotes the symmetric difference);

—$C(S')$ is an induced subgraph of G such that $C(S') = G[V \setminus S']$ if $\Delta = 3$; otherwise $C(S')$ is the subgraph of G induced by the vertices v of $V \setminus S'$ such that $|\Gamma(v) \cap S'| \geq 2$.

Algorithm 22 is what we have inferred from the very interesting (but difficult to read) paper of Berman and Fürer [1994]. It is recursive and calls algorithm EIS which, in linear time, finds an independent set in graphs with $\Delta = 2$ (a collection of cycles and trees). Let us remark here that the maximum degree

Algorithm 22. Algorithm SI(G, Δ, κ); every time G is updated, Δ is considered the maximum degree of the updated version of G.

begin
  $S'_\Delta \leftarrow \emptyset$;
  $\sigma = \Delta^{4\kappa} 32\kappa \log n$;
  while there exists an improvement S^i of size $\leq \sigma$ **do** $S' \leftarrow S' \oplus S^i$ **od**
  $G \leftarrow C(S')$;
  case Δ **do**
    Δ > 4: call SI(G, ($\Delta - 2$), κ);
    $\Delta \leq 4$: **call** EIS on G;
  od
  $S' \leftarrow \max_\Delta\{S_\Delta\}$;
end.

of $C(S')$ is at most $\Delta - 2$. So, for $\Delta > 4$, in order to solve IS in $C(S')$, one can continue to execute SI(G, Δ, κ) by simply replacing Δ by $\Delta - 2$; at the instance where $\Delta \leq 4$, then the maximum degree of the current $C(S')$ will become at most 2 and, in this case, IS becomes easily solvable to optimality by EIS.

In order for algorithm 22 to be polynomial, the improvements in Definition 6 have to be "small." In fact, to perform an improvement, one has first to search connected graphs of order equal to the size of the considered improvement. Denoting by σ the maximum size of such an improvement, the number of connected subgraphs of G with order at most σ is estimated to be lower than $n(4\Delta)^\sigma$ Berman and Fürer [1994]. This enforces a tradeoff between the sizes of Δ (influencing the applicability of the algorithm) and σ (influencing the magnitude of the improvement). For example, if we want σ to be of $O(\log n)$, then Δ has to be bounded above by a constant; if, however, we want Δ to be "large," say $O(n)$, then σ has to be a fixed constant.

The authors of Berman and Fürer [1994] prove, by a very elegant and fine analysis, that Algorithm 22 has approximation ratio $\rho \geq 5/(\Delta + 3) - O(1/\kappa)$ if Δ is even, whereas, in the case where Δ is odd, $\rho \geq 5/(\Delta + 3.25) - O(1/\kappa)$.

Here lies another drawback of Algorithm 22. In order for the ratio be as close as possible to $5/\Delta$, the quantity $1/\kappa$ has to tend to 0; consequently κ has to tend to ∞. But the larger the values of κ, the larger the size of σ, therefore, the larger the number of the candidate improvements (which depends on the number of connected subgraphs of G of order, at most, σ). Therefore, a large value of κ should very seriously aggravate the complexity of the algorithm.

Despite this drawback (affecting its complexity), the very interesting (from a mathematical point of view) work of Berman and Fürer [1994] has the great additional merit of being among the first to have accomplished a significant improvement of the IS-ratio.

Two improvements of an independent set. We present in the following another instantiation (Algorithm 23) of the schema expressed by Algorithm 21, the instantiation introduced by Khanna et al. [1994].

Algorithm 23. 2_OPT.

begin
  call ISALGO on G;
  let S' be the obtained IS-solution;
  while there exists a 2-improvement $\langle v_1, v_2, u\rangle$ of S' **do** $S' \leftarrow S' \oplus \{v_1, v_2, u\}$ **od**;
end.

Definition 7. Given an independent set S' of a graph G, two vertices v_1, $v_2 \in V \setminus S'$, and a vertex $u \in S'$, the triple $\langle v_1, v_2, u\rangle$ is a 2-*improvement* of S' if $S' \oplus \{v_1, v_2, u\}$ is an independent set of G.

It is easy to see that each 2-improvement augments the size of S' by 1. Moreover, performing all possible 2-improvements requires, at most, $O(n\Delta^2)$ time. When no 2-improvement is possible, then, as the authors of Khanna et al [1994] prove, $|S'| \geq [(1 + \iota(G))/(\Delta + 2)]n$, whereas in ℓ-clique-free graphs the bound for $|S'|$ is better than in general graphs, and equal to $[2/(\Delta + \ell)]n$.

If we consider the instantiation of ISALGO by Algorithm 17, then Algo-

Algorithm 24

begin
call ISALGO on G;
let S' be the obtained IS-solution;
for $k \leftarrow \ell$ to 2 **od**;
find a collection $\mathscr{C}_k$ of disjoint j-cliques;
$G \leftarrow G[V \setminus \cup_{K_k \in C_k} \{V(K_k)\}]$;
call ISALGOk in G;
let S_k be the obtained IS-solution;
od
$S' \leftarrow \max_{k=2, \ldots, \ell}\{S', S_k\}$;
end.

Algorithm 25

begin
compute a maximal matching M in G;
$X \leftarrow V \setminus T[M]$;
call algorithm 17 on G to obtain an IS-solution S';
$S' \leftarrow \max\{X, S'\}$;
end.

rithm 23 attains an approximation ratio $(2.44/\Delta) + o(1/\Delta)$ whereas, if the so-instantiated schema instantiates ISALGO in Algorithm 20 (replaces algorithm ISALGO), then the resulting algorithm attains an approximation ratio $3/(\Delta + 2)$ (but requires more execution time than Algorithm 17 which, as we have seen, yields the same ratio).

9.3.3 *Removing Cliques.* It is widely known that graphs not containing cliques contain larger independent sets than the general ones. Moreover, we could hope that finding a large independent set in clique-free graphs could be more fruitful and easier than for general graphs. On the other hand, all cliques of a fixed (not depending on n) size can be removed in polynomial time. So, if we remove cliques of a certain order from the initial graph, we could eventually apply improved IS-algorithms to the surviving subgraphs. This could be advantageous if the initial graph contains few cliques. However, if the input graph contains many such cliques, then obviously its stability number is small, and therefore many approximation algorithms would have good approximation ratios. In both cases, the overall approximation ratio of the strategy will be improved.

Algorithm 24, parametrized by a fixed constant ℓ and by a collection of IS-algorithms, ISALGO operating on general graphs and ISALGOk, $k = 2, \ldots, \ell$, algorithm ISALGOk operating in k-clique-free graphs, is a specification of a schema relying on clique removal, originally proposed in Halldórsson and Radhakrishnan [1994b]. Since searching for k-cliques can be performed in $O(\Delta^{k-1}n)$, the whole process of clique removal can be performed in $O(\Delta^{\ell}n)$, which is polynomial when $\ell = O(\log_\Delta n)$ (notice that, in order for such a process to remain polynomial for large values of Δ, ℓ has to be quite small); so, if all the called algorithms are also polynomial, the whole process is of polynomial complexity $O(\max_{k=2, \ldots, \ell}\{T(n), T_k(n_k), \Delta^{\ell}n\}$, where $T(n)$ denotes the complexity of algorithm ISALGO and $T_k(n_k)$, $k = 2, \ldots, \ell$, the ones of algorithms ISALGOk (operating on surviving graphs of order n_k, resp.).

This kind of schema for IS was originally used independently in Boppana and Halldórsson [1992] (see also Section 9.2) and in Paschos [1992] (Algorithm 25).

In the following, we will survey several instantiations of Algorithm 24.

Edge removal. Algorithm 25 considers edges as cliques (an edge can be seen, in fact, as a clique K_2) and moreover it instantiates ISALGO of Algorithm 24 by Algorithm 17. Once a maximal collection of such "cliques" has been removed, it considers graph $G[V \setminus T[M]]$ the simplest among all algorithms, the one that simply stocks $V \setminus T[M]$ (the set $V \setminus T[M]$ is the set of the exposed vertices of G with respect to M, and therefore an independent set by itself).

As mentioned in Haldórsson and

Algorithm 26. Shearer's algorithm; $f'(\cdot)$ denotes the derivative of $f(\cdot)$; μ represents the average degree of the surviving graph.

begin
  $S' \leftarrow \emptyset$;
  while $V \neq \emptyset$ **do**
    choose $v \in V$ such that $(\delta_s + 1)f_s(\mu) \leq 1 + (\mu\delta_s + \mu - 2\Sigma_{u \in \Gamma(u)}\delta_u)f'_s(\mu)$;
    $S' \leftarrow S' \cup \{v\}$;
    $G \leftarrow G[V \setminus (\Gamma(v) \cup \{v\})]$;
  od
end.

Radhakrishnan [1994a], Algorithm 25 yields approximation ratio $5/(2\mu + 4.5)$.

Triangle removal. If instead of edges one removes triangles, the surviving graph is triangle-free. For this class of graphs, Shearer [1983; 1991] has proved the following very nice result.

THEOREM 14 [Shearer 1983]. *For a triangle-free graph G, $\alpha(G) \geq f_s(\mu)n$, where $f_s(\delta) = (\delta \ln \delta - \delta + 1)/(\delta - 1)^2$, $\delta \in \mathbb{N}^+$, $f_s(0) = 1$ and $f_s(1) = 1/2$.*

Let us note that the bound of Theorem 14 has been slightly improved in Shearer [1991].

Based upon the preceding result, Shearer has devised a PTAA (Algorithm 26) yielding, in triangle-free graphs, approximation ratio $\Omega(\log\Delta/\Delta)$.

Algorithm 24 parametrized by $\ell = 3$ and instantiated by Algorithm 17 for ISALGO and by Algorithm 26 for ISALGOk yields, as mentioned in Halldórsson and Radhakrishnan [1994a], approximation ratio $(3.5/\Delta) - o(1/\Delta)$.

Removing larger cliques. Another instantiation of the clique-removal schema is proposed in Halldórsson and Radhakrishnan [1994a]. There, Algorithm 9.3.3 is instantiated by $\ell = 4$, ISALGO and ISALGO4 are instantiated by Algorithm 17, algorithm ISALGO3 by Algorithm 26, and algorithm ISALGO2 consists of taking all the surviving vertices as a candidate solution. Such an instantiation yields approximation ratio $(3.67/\Delta) - \epsilon$, for a very small fixed constant ϵ.

One of the most interesting instantiations of the clique-removal schema is, however, the following: Algorithm 24 is parametrized by a large ℓ, algorithm ISALGO is instantiated by Algorithm 23 (where the algorithm ISALGO of Algorithm 23 can be any maximal IS-algorithm, e.g., Algorithm 17), and algorithms ISALGOk are instantiated by Algorithm 20. The algorithm ISALGO called by Algorithm 20 is instantiated by Algorithm 26.

The so-devised instantiation yields the very interesting approximation ratio $(6/\Delta) - \epsilon(\ell) - \eta(\Delta)$, where $\epsilon(\ell) \to 0$ when $\ell \to \infty$, and $\eta(\Delta) \to 0$ when $\Delta \to \infty$.

This elegant instantiation has, however, a fairly serious drawback. Let us first notice that its complexity is $O(\Delta^{\ell} n)$. The asymptotic ($\Delta \to \infty$) ratio yielded is certainly, for large values of ℓ, close to $6/\Delta$, but in this case, the complexity becomes huge. This result was, up to now, the best polynomial approximation result for IS in general graphs.

Let us now consider another application of Algorithm 24, yielding ratio $O(\log\log \Delta/\Delta)$, but *working, unfortunately, only for bounded-degree graphs*.

Ajtai et al. [1981] proved the following very interesting theorem.

THEOREM 15 [Ajtai et al. 1981]. *There exists a fixed constant c such that every graph of order n without ℓ-cliques contains an independent set of cardinality greater than, or equal to, $cn[\log[(\log \mu)/\ell]]/\mu$.*

By derandomizing these parts of the proof of Theorem 15 where probabilistic existence arguments are used, Halldórsson and Radhakrishnan [1994b] have produced an elegant (and quite complicated) constructive proof giving rise to a nice approximation algorithm, called AEKS in Halldórsson and Radhakrishan [1994b], finding an independent set of size $O(n[\log[(\log \mu)/\ell]]/\mu)$.

Algorithm AEKS is strongly dependent on the constructive proof of Theorem 15, this proof taking seven pages in

Algorithm 27. Instantiation of the clique-removal schema with ratio $O(\log\log \Delta/\Delta)$ for bounded-degree graphs.

begin
  let S' be any maximal independent set of G;
  for $k \leftarrow \lfloor c \log\log \Delta \rfloor$ to 2 **od**;
    find a collection C_k of disjoint cliques;
    $G \leftarrow G[V \setminus \cup_{K_k \in C_k} \{V(K_k)\}]$;
    call AEKS in G;
    let S_k be the obtained IS-solution;
  od
  $S' \leftarrow \max_{k=2, \ldots, \ell}\{S', S_k\}$;
end.

Algorithm 28

begin
  call algorithm 2_OPT in G;
  let S' be the obtained IS-solution;
  $\tilde{V} \leftarrow \{v \in V \setminus S' : |\Gamma(v) \cap S'| \geq 2\} \cup S$;
  compute a maximal collection $\mathscr{C}_\ell$ of disjoint ℓ-cliques in $G[\tilde{V}]$;
  $G \leftarrow G[\tilde{V} \setminus \cup_{K_\ell \in \mathscr{C}_\ell} V(K_\ell)]$;
  call ISALGO on G;
  let S_ℓ the obtained IS-solution;
  $S' \leftarrow \text{argmax}\{|S'|, |S_\ell|\}$;
end.

Halldórsson and Radhakrishnan [1994b]; moreover, despite our efforts, we have not been able to simplify or, at least, to shorten this elegant proof. Since, giving the whole proof here would seriously lengthen this (already long) article and, on the other hand, giving the algorithm without the proof would seriously attain its (the article's) clarity, we prefer to omit it.

The main result of [Halldórsson and Radhakrishnan 1994b] is given by the following theorem evaluating the approximation performance of Algorithm 27, which, in fact, is an instantiation of Algorithm 9.3.3, parametrized by $c \log\log \Delta$ (c being the constant of Theorem 15) and calling any maximal IS-algorithm as instantiation of ISALGO, as well as algorithm AEKS as instantiation of algorithms ISALGOk.

THEOREM 16 [Halldórsson and Radhakrishnan 1994b]. *Algorithm* 27 *has performance ratio* $O(\log\log \Delta/\Delta)$.

The result of Theorem 16 [Halldórsson and Radhakrishnan 1994b] is the best approximation result up to now and partly answers one of the major questions about the limits of the improvement of the IS-ratio (cf. the beginning of this part).

But its impact is seriously attenuated by the fact that Algorithm 27 does not always run in polynomial time. As can be seen in Halldórsson and Radhakrishnan [1994b], its complexity is dominated by the clique removal. Therefore the overall complexity of Algorithm 27 is polynomial only for Δ bounded above by $n^{1/\log\log n}$.

Improved polynomial approximation algorithms for independent sets. In the following we present some very recent results radically improving IS-approximation ratio in polynomial time (without any constraint in the degrees of the input graph G).

Let us consider Algorithm 28 which is an instantiation of the clique removal schema (Algorithm 24). Let us denote by $T(n)$ the complexity of ISALGO.

Then the whole complexity of Algorithm 28 is $O(\max\{n\Delta^2, T(n), \Delta^{\ell-2}|E|\})$, where $O(n\Delta^2)$ expresses the complexity of algorithm 2_OPT and $O(\Delta^{\ell-2}|E|)$ the one of clique removing [Demange and Paschos 1966].

Regarding Algorithm 28, the following theorem is proved in Demange and Paschos [1996b].

THEOREM 17 [Demange and Paschos 1996b]. *If there exists an algorithm ISALGO guaranteeing, for every* $\ell \in \mathbb{N}$ *and every* ℓ*-clique-free-graph, an approximation ratio* ρ_ℓ *for IS, then, for every graph G, for every* $\epsilon > 0$*, and for every* $\lambda > 0$*, Algorithm* 28 *guarantees an approximation ratio for IS, bounded below by min* $\{\lambda, \epsilon'(1-\lambda)\rho_\ell, 2(\ell-\epsilon)(1-\lambda)/(\Delta+2)\}$, *where* ϵ' *is such that* $1/(\ell-\epsilon) = (1/\ell) + \epsilon'$.

Let us remark that in the proof of our Theorem 17 and in Algorithm 28, the constants ϵ and λ do not intervene, neither in the algorithm, nor consequently in its complexity.

Let us, for a fixed constant κ, set $\ell = \lceil(\kappa/2)\rceil + 1$, $\lambda = (\kappa + 2)/\Delta$, and $\epsilon = 1$ (which implies $\lambda \geq 2[(\ell - \epsilon)(1 - \lambda)]/(\Delta + 2)$ and $\epsilon' = 1/[\lceil\kappa/2\rceil(\lceil\kappa/2\rceil + 1)]$). Moreover, consider that algorithm ISALGO is instantiated by algorithm AEKS of Halldórsson and Radhakrishnan [1994b].

Then the application of Theorem 17 guarantees, in time bounded above by $O(n^{\lceil\kappa/2\rceil})$, an approximation ratio bounded below by $(\kappa/\Delta) - \eta$, $\eta \rightarrow 0$ and the following theorem holds [Demange and Paschos [1996b].

THEOREM 18 [Demange and Paschos 1996]. *For every fixed integer constant* κ *and for* $\ell = \lceil(\kappa/2)\rceil + 1$*, Algorithm* 28 *is parametrized by* ℓ *and* AEKS *is an approximation algorithm with time-complexity* $O(n^{\lceil\kappa/2\rceil})$ *for IS, guaranteeing an approximation ratio asymptotically* ($\Delta \rightarrow \infty$) *equal to* κ/Δ.

The result of Theorem 18 constitutes a radical improvement for the positive polynomial approximation results for IS. Up to now, Algorithm 27, providing (for $\Delta \rightarrow \infty$) an approximation ratio of $6/(\Delta + 2) - \epsilon$ was, to our knowledge, the best polynomial time approximation result. However, as we have seen, this algorithm has the drawback of being exponential in $1/\epsilon$.

Let us revisit Algorithm 28 and set $\ell = 3$. By instantiating algorithm ISALGO, by Algorithm 26 and setting $\lambda = 6/(\Delta + 8)$ and (for $\Delta \geq \exp\{19\}$) $\epsilon = 54/(\ln \Delta - 1)$, we get, applying Theorem 17, an approximation ratio of $[[6 - [108/\log(\Delta - 1)]][1 - 6/(\Delta + 8)]]/(\Delta + 2)$.

This ratio is slightly better than the one (of the same order) of Algorithm 27, since here there is no term $\epsilon(k)$ (recall that the ratio in Haldórsson and Radhakrishnan [1994b] is $(6/\Delta) - \epsilon(k) - \eta(\Delta)$, where $\epsilon(k) \rightarrow 0$ when $k \rightarrow \infty$ and $\eta(\Delta) \rightarrow 0$ when $\Delta \rightarrow \infty$). Moreover, in this case, the complexity of Algorithm 28 is of $O(n|E|)$ (recall that the construction of $\mathscr{C}_3$ takes time $O(\Delta|E|) \leq n|E|$).

Algorithm 29

begin
initialize S' by the output of algorithm ISALGO1;
compute a maximal collection $\mathscr{C}_\ell$ of disjoint ℓ-cliques in G;
$G \leftarrow G[V \setminus \cup_{K_\ell \in \mathscr{C}_\ell} V(K_\ell)]$;
call ISALGO2 on $G[X_\ell]$;
let S_ℓ be the obtained IS-solution;
$S' \leftarrow \text{argmax}\{|S'|, |S_\ell|\}$
end.

Thus the following result improves the corresponding one of Haldórsson and Radhakrishnan [1994b], from both approximation performance and complexity points of view.

THEOREM 19. *Algorithm* 28*, parametrized by* $\ell = 3$ *and by Algorithm* 26*, constitutes an* $O(n|E|)$ *approximation algorithm for IS yielding asymptotic* ($\Delta \rightarrow \infty$) *approximation ratio* $6/\Delta$.

Let us now consider another Algorithm 29 (lying in the line of the clique-removing schema), parameterized by algorithms ISALGO1 and ISALGO2, where algorithm ISALGO2 runs on ℓ-clique-free graphs.

If we denote by $T_1(n)$ the complexity of algorithm ISALGO1 and by $T_2(n)$ the complexity of algorithm ISALGO2, the total complexity of Algorithm 29 is $O(\max\{T_1(n), T_2(n), \Delta^{\ell-1}n\})$.

Then the following theorem holds Demange and Paschos [1996b].

THEOREM 20 [Demange and Paschos 1996]. *Let us consider, for every fixed integer constant* ℓ*, the simultaneous existence of an algorithm* ISALGO2 *guaranteeing, for every* $\ell \in \mathbb{N}$*, an approximation ratio* ρ_ℓ *for IS in* ℓ*-clique-free graphs and of an algorithm* ISALGO1 *constructing, for every graph G, a maximal independent set of size at least n f(G). Then Algorithm* 29 *solves IS with an approximation ratio bounded below* *by* $\min\{\epsilon' \rho_\ell, (\ell-\epsilon)f(G)\}$, *where* ϵ' *is such that* $1/(\ell - \epsilon) = (1/\ell) + \epsilon'$.

Let us instantiate algorithm ISALGO1 by Algorithm 17 (the greedy algorithm[17]) and ISALGO2 by algorithm AEKS. Then we obtain, applying Theorem 20 with $\ell = \kappa$ and $\epsilon = 1$, the following concluding theorem.

THEOREM 21 [Demanage and Paschos 1996b]. *For every fixed integer constant* κ, *Algorithm* 29, *parametrized by* κ, *where algorithm* ISALGO1 *is instantiated by Algorithm* 17 *and* ISALGO2 *by algorithm* AEKS, *constitutes an approximation algorithm with time-complexity of* $O(n^\kappa)$, *guaranteeing asymptotic* ($\mu \to \infty$) *approximation ratio bounded below by* $\min\{\kappa/\mu, [\kappa' \log(\log \Delta)]/\Delta\}$, *where* $\kappa' = c'/[\kappa(\kappa - 1)]$, *for a fixed constant* c'.

The result of Theorem 21 further improves (sometimes quite largely) the result of Theorem 18 and constitutes, to our knowledge, the best known polynomial approximation result for IS.

10. APPROXIMATING THE WEIGHTED INDEPENDENT SET

10.1 Algorithms Based on the Semiintegral Property of LP-Relaxation

Hochbaum [1983] is the first author who has proposed an efficient algorithm for IS, the approximation ratio of which is essentially better than $2/\Delta$. Based upon the semi-integral property of the LP-relaxation of WIS, she proves the following very nice theorem.

THEOREM 22 [Hochbaum 1983]. *Let* G *be a graph of order* n *with* m *edges; let* k *be an integer greater than* 1. *If it takes only* s *steps to color the vertices of* G *in* k *colors* (*so that adjacent vertices have distinct colors*), *then it takes only* $s + O(nm \log n)$ *steps to find an independent set whose weight is at least* $2/k$ *times the weight of an optimal independent set, and to find a vertex cover whose weight is at most* $[2 - (2/k)]$ *times the weight of an optimal vertex cover.*

In fact, to prove Theorem 22, one has first to define the sets P, Q, and R of vertices with feasible values 1, 1/2, and 0, respectively; next one colors the vertices of $G[Q]$ by k colors and, finally, one takes as IS-solution the union of the heavier color class of Q with the set P. So we find here once more an application of the results of [Nemhauser and Trotter 1975] for the improvement of the approximation ratio of VC.

Let us first notice that a coloring of the vertices of the graph, such that adjacent vertices have distinct colors, causes each color to induce an independent set in the graph.

It now remains to define k. To do that, a definition can be based upon the following famous Brooks' [1941] theorem.

THEOREM 23 [Brooks 1941]. *If* G *is a connected graph, and if* $G \neq K_{\Delta+1}$, *and if* G *is not an odd length cycle, in case of* $\Delta = 2$, *then it is* Δ*-colorable.*[18]

Lovász [1975a] gives an elegant constructive proof of Theorem 23, this proof providing an algorithm finding a Δ-coloring of a graph in $O(\Delta n)$ steps.

In all, Algorithm 30 is the IS-algorithm proposed in Hochbaum [1983], and Theorem 24 characterizes its performance.

THEOREM 24 [Hochbaum 1983]. *Algorithm* 30 *is an* $O(n^3 \log n)$ *worst-case approximation algorithm which, given a graph of order* n *and of maximum degree* Δ, *finds an independent set whose weight is at least* $2/\Delta$ *times the weight of an optimal independent set; Algorithm* 30 *also finds a vertex covering whose weight is at most* $[2 - (2/\Delta)]$ *times the weight of an optimal vertex covering.*

In Hochbaum [1983], the following counterexample is given to show that it is very unlikely that the approximation

[17] Recall that by Turán's [1941] theorem, it guarantees a maximal independent set of size at least $n/(\mu + 1)$.

[18] By K_t, we denote a complete graph on t vertices.

Algorithm 30. Hochbaum's IS-algorithm.

begin
solve $WIS_r(G)$;
let P be the set of vertices with feasible value 1;
let Q be the set of vertices with feasible value 1/2;
let R be the set of vertices with feasible value 0;
call the algorithm of [54] to obtain a Δ-coloring for $G[Q]$;
let S be the heaviest among the Δ color classes of Q;
take $P \cup S$ as solution for WIS;
take $V \setminus (P \cup S)$ as solution for WVC;
end.

ratio for WVC (this counterexample works also for WIS) could be improved. Consider a graph G formed by k cliques K_k and $k(k-1)$ independent sets. Each clique has one edge connecting it to one of the vertices of an independent set; $k-1$ of the stable sets are one set of vertices in a complete bipartite graph, the kth independent set being the second set of vertices (in the complete bipartite graph). For such a family of graphs, one can easily see that G is k-chromatic. One feasible k-coloring consists of each one of the independent sets colored by one of the k-colors.[19]

Then Algorithm 30 provides a vertex covering C of size $(2k-1)(k-1)$, and the optimal vertex covering C^* has cardinality equal to $k(k-1)+k-1$; consequently, $|C|/|C^*| = 2 - [[2-(3/k)]/[k-(1/k)]]$, this approximation ratio being arbitrarily close to 2 (for large values of k).

In the same vein, it is proved in Paschos [1993] that the combination of Lovász's coloring-algorithm with Algorithm 16 of Section 8.2 provides a PTAA for IS, the ratio of which is $2/\Delta + o(1/\Delta)$, slightly improving Hochbaum's result.

There is an interesting conclusion (or, rather, feeling) credited to Hochbaum's works on the approximation of hard combinatorial optimization problems. It seems that primal-dual methods of linear programming are very efficiently adaptable for the approximation of optimal solutions of weighted versions of some combinatorial problems; for example, some of the most efficient approximation algorithms for WIS or WVC are not purely combinatorial, but they use a linear programming technique. However, the use of linear programming methods make the derived algorithms more "expensive" in complexity.

10.2 Improved Approximations for the Weighted Independent Set

For many years, $2/\Delta$ remained the best approximation ratio for WIS. Recently, Halldórsson [1995], using a very beautiful partition theorem as well as a little-known partition-result of Lovász[20] [1966], improved the IS-ratio to $3/\Delta + 2$.

Let us present what we have inferred from the very interesting (but quite short) work of Halldórsson [1995]. The starting point there, is the following theorem.

THEOREM 25 [Halldórsson 1995]. *Let Π be a hereditary induced subgraph problem.*[21] *If one can:*

(1) *obtain t induced subgraphs $G_1, \ldots, G_t$ of G, such that each vertex in G is contained in at least k different G_i, and*
(2) *obtain t feasible solutions for Π (one solution for each G_i) which approxi-*

[19] Moreover, one can make this coloring unique, by adding some more edges as follows: the vertices in the cliques that connect each clique to each independent set are linked together to make a complete subgraph; the ith independent set is connected to all these vertices, except for the ith vertex.

[20] Let us remark that, definitely, the works of Lovász are at the origin of the most of the interesting approximation results on WIS.

[21] A property π of graphs is *hereditary* if, whenever it holds for a graph, it also holds for its induced subgraph; for π, the associated subgraph problem is the one of finding a subgraph of maximum weight satisfying π.

mate the t optimal ones within ratios ρ_i, $i = 1, \ldots, t$,

then Π can be approximated within ratio $(k/\Sigma_{i=1}^{t} \rho_i)$ by simply retaining the largest of the t feasible solutions obtained.

Let us note that the approximation strategy induced by Theorem 25 has been also used (for $k = 1$, informally and without any study of its impact) in Paschos [1994]; a notable and systematic application of Theorem 25 (concerning, moreover, classes of the problems treated here), can be found in Baker [1994].

The following little-known result of Lovász [1966] (Theorem 26), on the other hand, turns out to be a very interesting and quite powerful tool (the whole impact of which has not yet been exhaustively studied) for partitioning a graph into subgraphs of low maximum degree.

THEOREM 26 [Lovász 1966]. *Let $G = (V, E)$ be a multigraph with no self loops. Let $k_1, k_2, \ldots, k_t$ be nonnegative integers such that $1 + \Sigma_{i=1}^{t} (k_i - 1) = \Delta$. Then V can be partitioned into t subsets, inducing subgraphs $G_1, \ldots, G_t$ such that, if Δ_i is the maximum degree of G_i, $i = 1, \ldots, t$, then $\Delta_i \leq k_i$.*

The proof of Theorem 26 is based upon a local search, applying the following rule (Algorithm 2): if for some G_i a vertex v of G_i has $|\Gamma(v) \cap V(G_i)| > k_i$, then move it into some G_j, $j \neq i$, where $|\Gamma(v) \cap V(G_j)| \leq k_j$.

In Lovász [1966] it is proved that, by the pigeonhole principle, at least one of the subgraphs will have this property and, in addition, that this local search converges. Moreover, as is shown in Halldórsson [1995c], Procedure 2 has complexity at most $O(\Delta|E|)$.

A corollary of Theorem 26 is that one can partition a graph into, at most, $\lceil(\Delta + 1)/3\rceil$ graphs of degree at most 2 (this explains the second line of Algorithm 31). Then, applying Theorems 25 and 26, one can obtain a ratio $3/(\Delta + 2)$ for

Procedure 2. Procedure PARTITION $(G, t, (k_i)_{i=1,\ldots,t})$.

begin
  $(G_1, \ldots, G_t) \leftarrow$ an arbitrary partition of G into t induced subgraphs;
  for $i \leftarrow 1$ to t **do**
    if $v \in V(G_i) : |\Gamma(v) \cap V(G_i)| > k_i$
      then move v into G_j where $|\Gamma(v) \cap V(G_j)| \leq k_j$;
  fi
end;

Algorithm 31

begin
  call PARTITION $(G, \lceil(\Delta + 1)/3\rceil, (2)_{i=1,\ldots,\lceil(\Delta+1)/3\rceil})$;
  for each G_i do solve IS in G_i **od**
  maximilize the obtained independent sets;
  retain the heaviest so-formed independent set;
end.

WIS Halldórsson [1995c]. This ratio is the best known up to now.

PART V. NAVIGATING BETWEEN VERTEX COVERING AND INDEPENDENT SETS

One of the most interesting and challenging problems in complexity theory is related to the approximability of VC. As we have already seen in Part II, numerous polynomial time approximation algorithms have been developed for VC, all of them achieving approximation ratio 2 and, moreover, counterexamples have been developed for these algorithms, showing that their ratio cannot be considerably improved.

Consequently, there are two interesting open problems about VC's approximability:

(i) devise an algorithm with a better ratio, and
(ii) prove a lower bound for the ratios of any algorithm supposed to solve approximately VC.

Of course, the result of Arora et al. [1992], recently improved in Bellare et al. [1995], affirms the nonapproximability of VC within an approximation ratio less than 1.038 (assuming that P ≠ NP), but all the possibilities in the interval [1.038, 2] remain open.

The lack of notable progress on the improvement of the VC-ratio has a great number of researchers convinced that there is no polynomial time approximation algorithm achieving a ratio strictly smaller than $2 - \epsilon$, for a positive constant ϵ, unless P ≠ NP (see, e.g., Hochbaum [1983], where Hochbaum conjectures the nonexistence of such an approximation algorithm).

In this part, we give some other leads in order to tackle points (i) and (ii). In fact, we give two sufficient conditions, the satisfiability of the former inducing an upper bound on all ratios obtained by potential polynomial time approximation algorithms for VC and, in addition, the satisfiability of the latter permitting the conception of a polynomial time approximation algorithm, achieving a ratio strictly smaller than 2 for VC.

11. PROBLEMS S_κ

We present here a restricted class of IS-problems (Definition 8) introduced in Demange and Paschos [1995b] and discussed in Demange and Paschos [1995b, c, 1996a] the approximability of which is closely related to the approximability of VC.

Definition 8. Problem S_κ

—For every constant $\kappa > 1$, the stability problem S_κ corresponds to the restriction of IS to graphs (of order n) admitting stability number greater than or equal to n/κ.

—For problem S_{κ^∞}, an instance is a pair (G, κ), $\kappa \geq 1$, and G is a graph of order n with $\alpha(G) \geq n/\kappa$. Here also, the objective is to determine a maximum independent set of G.

Problems S_κ ($\kappa \geq 2$), in particular for small values of κ, seem to have interesting properties, since they interfere with VC (as well as with some interesting mathematical programming problems, as we show in Demange and Paschos [1995b]) and its approximability behavior, from both positive and negative points of view.

Let us now give some results (proved in Demange and Paschos [1996a]) about problems S_κ.

THEOREM 27 [Demange and Paschos 1996a].

—*The problem of deciding if a graph G of order n verifies $\alpha(G) \geq n/\kappa$ is NP-complete* $\kappa \geq 2$, $\kappa \in \mathbb{N}$.

—*For every constant ρ, there is no polynomial time approximation algorithm for S_{κ^∞} that guarantees an approximation ratio greater than or equal to ρ, unless $P = NP$.*

—*If $P \neq NP$, then* $\varepsilon > 0$ *and* κ_0 *such that,* $\kappa \geq \kappa_0$, *no algorithm polynomial in n (but, eventually, exponential in κ) for S_κ guarantees approximation ratio* $(1/\kappa)^\varepsilon$.

The second item of Theorem 27 is proved by a kind of graph-composition similar to the one of Garey and Johnson [1979] for the case of IS.

For the problems S_κ, the nonexistence of constant-ratio PTAA for S_{κ^∞} means that, *if there exists a polynomial time approximation algorithm guaranteeing, for every problem S_κ, an approximation ratio $\rho(\kappa)$, then the mapping $\kappa \leftrightarrow p(k)$* tends to 0 whenever $\kappa \leftrightarrow \infty$; then the third item of Theorem 27 specifies the convergence velocity of ρ to 0.

Unfortunately, we cannot, up to now, characterize more precisely the hardness of approximating problems S_κ for a fixed κ. The only remark we can make is that, for $\kappa < 2$, S_κ is polynomially constant-approximable.

In fact, let us consider an instance G of $S_{2-\epsilon}$ for a positive constant ϵ. Then $\alpha(G) \geq n/(2 - \epsilon) \geq [(1/2) + \epsilon']n$ for $\epsilon' > 0$. Let us denote by e the number of vertices exposed[22] with respect to a maximum matching M of cardinality m.

[22] The set of these vertices is nonempty because of the hypothesis $\alpha(G) > n/2$.

We have $n = 2m + e$ and $\alpha(G) \leq m + \epsilon$. If we choose as an approximate solution for IS the independent set formed by these vertices, the previous expressions lead to $e \geq 2\epsilon' n$ and the constant ratio $2\epsilon'$ is guaranteed for $S_{2-\epsilon}$.

12. APPROXIMABILITY BEHAVIOR OF A VERTEX COVERING

In what follows, by C and S, respectively, we denote a vertex covering and the independent set associated with C. Also, for reasons of simplicity, we fix $\kappa = 3$ (we treat thusly the case of S_3); as shown in Demange and Paschos [1995b], the arguments proving the results of this section remain valid for any value of κ.

12.1 The Nonconstant-Approximability of S_3

Theorem 28 is the "negative" facet of the influence of the approximability of S_3 on the one of VC. It is proved in Demange and Paschos [1995b] using a standard gap technique. As one can see, the potential lower bound introduced by this result is significantly better than the one of Bellare et al. [1995].

THEOREM 28 [Demange and Paschos 1995b]. *If S_3 is not constant-approximable in polynomial time, then there cannot exist a polynomial time approximation algorithm for VC guaranteeing an approximation ratio strictly smaller than* 3/2.

12.2 The Constant-Approximability of S_3

In order to study the "positive" facet of the interfering of the approximability of S_3 to the one of VC, we introduce in Demange and Paschos [1995b] a polynomial time approximation algorithm for VC (Algorithm 32). For this, we suppose that there exists a polynomial time approximation algorithm $\mathcal{A}$ for S_3 with a fixed positive constant approximation ratio ρ.

12.2.1 *An Algorithm for Vertex Covering and Its Properites.* We introduce

Algorithm 32

begin
  call the algorithm of [26] in G;
  if G is KE **then**
    let S^* be the obtained IS-solution;
    $C^* \leftarrow V \setminus S^*$;
    output C^*;
  fi
  call algorithm 6 and store the obtained VC-solution;
  if M is perfect
    then call procedure 3 and store the obtained VC-solution;
    else call procedure 4 and store the obtained VC-solution;
  fi
  choose the smallest among the candidate VC-solutions;
end.

and discuss now two procedures for finding a vertex covering in a graph G. They are then exploited in the more general Algorithm 32 presented in the sequel.

As we see, Algorithm 32 calls the two procedures presented in what follows as well as Algorithms 6 and the one of Deming [1979] (to decide if G is KE and if yes, to find a maximum independent set) and chooses the smallest solution among the produced ones. Let us note that the version of Algorithm 6 considered in this part finds a *maximum* matching M and not a *maximal* one (as in the original version of Algorithm 6 used previously in this article).

Procedures 3 and 4, as well as Algorithms 6 and 32, have as input a connected graph G and output a vertex covering for G. First Algorithm 6 is called and a VC-solution is computed starting from a maximum matching M. In the case where M is perfect, Procedure 3, a simple procedure calling the hypothetical constant-ratio approximation algorithm $\mathcal{A}$, and then taking the complement of the solution provided by $\mathcal{A}$, is executed. Finally, if M is not perfect, then Procedure 4 is executed.

Let $|M| = m$ (and suppose that M is not perfect). Let S be the independent

Procedure 3

begin
  apply $\mathscr{A}$ on G;
  let S be the obtained IS-solution;
  $C \leftarrow V \setminus S$;
end;

Procedure 4

begin
  compute a maximum matching M in G;
  $G' \leftarrow G[T[M]]$;
  call procedure 3 on G' to obtain sets C and S;
  determine M_1, M_2, X_1 and X_2;
  if $m_1 \leq \rho m/3$
    then $C \leftarrow C \cup (T[M_1] \cap S)$;
    else
      $C_2 \leftarrow T[M_2]$;
      $C_1 \leftarrow \emptyset$;
      $S_1 \leftarrow \emptyset$;
      order arbitrarily the elements of X_1;
      let $X_1 = \{x_1, \ldots, x_{|X_1|}\}$be the resulting ordering;
      for $l \leftarrow 1$ to $|X_1|$ **do**
        $V_l \leftarrow T[M_1[x_l]] \cup \{x_1\}$;
        $E_l \leftarrow E(G[V_l])$;
        find all 5-cycles $x_l - s_i - c_i - c_j - s_j - x_l$ with $\{s_i c_i, c_j s_j\} \subseteq M_1$;
        $C_1 \leftarrow C_1 \cup \{x_l, c_i, c_j\}$;
        $S_1 \leftarrow S_1 \cup \{s_i, s_j\}$;
        find all triangles $x_l - s_i - c_i - x_l$ such that $\{s_i c_i\} \in M_1$;
        $C_1 \leftarrow C_1 \cup \{x_l, c_i\}$;
        $S_1 \leftarrow S_1 \cup \{s_i\}$;
        while $(c_k \in (V_l \setminus C_1) \wedge s_i \in S_1 \cap V_l)$, $c_k s_i \in E_l$, $k \neq i$, **do**
          $C_1 \leftarrow C_1 \cup \{c_k\}$;
          $S_1 \leftarrow S_1 \cup \{m(c_k)\}$;
        od
        while $s_k c_k \in M_1$, $\{s_k, c_k\} \subseteq V_l$, $\{s_k, c_k\} \cap C_1 = \emptyset$ **do**
          $C_1 \leftarrow C_1 \cup \{s_k\}$;
          $S_1 \leftarrow S_1 \cup \{c_k\}$;
          od
        od
        $S_1 \leftarrow S_1 \cup (X \setminus C_1)$;
        $C \leftarrow C_1 \cup C_2$;
    fi
end;

set derived by Procedure 3 when applied to $G' = G[T[M]]$. Let X be the set of the exposed vertices of V with respect to M, and let $M_1 \subseteq M$ ($|M_1| = m_1$) be the edges of M having one endpoint in $S \cap \Gamma(X)$. Let $M_2 = M \setminus M_1$ ($|M_2| = m_2 = m - m_1$); also, let us assume that $T[M_1] \cap S = \{s_1, \ldots, s_{m_1}\}$ and $c_i = m(s_i)$, $i = 1, \ldots, m_1$; let $X_1 = \Gamma(T[M_1] \cap S) \cap X$ and let $X_2 = X \setminus X_1$.

For X_1 the following holds: $x \in X_1$, $c_i s_i \in M_1[x]$, $s_i x \in E$.

In fact, since $c_i s_i \in M_1$, we have $y \in X_1$, $s_i y \in E$. If $y = x$, the truth of the remark is obvious. Suppose now that $x \neq y$. Since $c_i s_i \in M_1[x]$, $\Gamma(x) \cap c_i, s_i \neq \emptyset$. But $c_i x \notin E$ because, in the opposite case, the alternating path $y - s_i - c_i - x$ would be augmenting, contradicting the maximality of M; consequently, $s_i x \in E$.

Finally, let us note that the set C (output of Procedure 4) is initialized at the fourth line of the procedure by the output of Procedure 3 called at this line, and it is completed by the execution of either the consequence **then**, or the consequence **else** of Procedure 4.

The following lemma (proved in Demange and Paschos [1995b]) brings to the fore some interesting properties of Procedure 4.

LEMMA 4 [Demange and Paschos 1995b].

(1) *Suppose that the consequence **else** of the **if** clause of Procedure 4 is executed and consider a vertex $v \in S_1 \setminus X$. Then there exists an exposed vertex $x_1 \in X_1$ and an alternating path from v to x_1 starting with $vm(v)$, all edges of this path being included in E_1.*

(2) *There is no edge $uv \in M_1$ such that there exist $\{x_i, x_j\} \subseteq X$, $x_1 \neq x_j$, with $\{ux_i, vx_j\} \subseteq E$. A particular case of this fact is that there is no edge $uv \in M_1$ such that one of its endpoints, say u, is linked to an exposed vertex $x_i \in X_1$ and the other one, say v, is linked to an exposed vertex $x_j \in X_2$.*

(3) Correctness of Procedure 4. *Procedure 4 finds in polynomial time a vertex covering C of its input graph G.*

Algorithm 32 is the claimed VC-algorithm. Let us recall that this algorithm uses the hypothetical constant-ratio approximation algorithm $\mathscr{A}$ (directly called by Procedure 3) for S_3; recall also that Algorithm 6 intermediately computes a maximum matching M.

12.2.2 *A "Positively" Conditional Result for Vertex Covering*

THEOREM 29 [Demange and Paschos 1995b]. *On the hypothesis that algorithm $\mathscr{A}$ is a ρ-approximation PTAA for S_3, for $\rho < 1$ a fixed positive constant, Algorithm* 32 *is a polynomial time approximation algorithm for VC, guaranteeing an approximation ratio smaller than* $2 - (\rho/6) < 2$.

The proof of Theorem 29 is given in detail in Demange and Paschos [1995b]. It is based upon an exhaustive study of the discrete duality gap of G, discussed in Section 8.

$f + g = 0$. In this case, as we have seen in Section 8, the IS-algorithm of Deming [1979] (called in the second line of Algorithm 32) finds a maximum independent set of G; so its complement C^* approximates VC within ratio 1.

$f + g \geq m/3$. In this case, Procedure 6 is used as an approximation algorithm for VC; its approximation ratio is proved in Demange and Paschos [1995b] to be greater than 3/2.

$0 < f + g \leq m/3$. In this case, we distinguish two subcases depending on whether M is perfect.

M is perfect. Then[23] $n = 2m$ and Procedure 3 is used to obtain an approximate solution for VC; its approximation ratio is proved to be, at most, $2 - (2\rho/3 < 2$.

M is not perfect. This is the case[24] where Procedure 4 is called to solve VC. Consider the graph $G' = G[V \setminus X]$. M is perfect for G'. Obviously, since $\alpha(G') \geq m - f \geq 2m/3 = |V(G')|/3$, G' is an instance of S_3. The call of Procedure 3 to G' (performed by Procedure 4) initializes sets C and S and allows the computation of M_1, M_2, X_1, and X_2. With respect to M_1, we consider the following two cases.

$m_1 \leq \rho m/3$. This case[25] is treated during the execution of the **then** consequence of the **if** clause in Procedure 4 and yields ratio $2 - (\rho/3) < 2$ [Demange and Paschos 1995b].

$m_1 \geq \rho m/3$. This case represents the **else** consequence of the **if** clause in Procedure 4 and, as we prove in Demange and Paschos [1995b], it yields ratio $2 - (\rho/6) < 2$.

Let us note that the negative result of Theorem 28 can be immediately transferred to the case for WVC (this problem being, a priori, harder than VC). On the other hand, the approximation preserving reduction between VC and WVC, proposed at the beginning of Section 5.2, ensures that the positive result of Theorem 29 holds also for WVC.

The following theorem summarizes the results of Sections 12.1 and 12.2.

THEOREM 30 [Demange and Paschos 1995b]. *Let $\rho < 1$ be a fixed positive constant. Under the hypothesis $P \neq NP$,*

(*i*) *the nonexistence of a ρ-approximation polynomial time algorithm for S_3 implies that no polynomial time approximation algorithm for VC and WVC can guarantee an approximation ratio strictly smaller than 3/2;*

[23] $g = |X_S| = |X_C| = |X| = 0$.

[24] $X \neq \emptyset$.

[25] We have arbitrarily chosen the constant 3 for the denominator of the fraction; in fact, Theorem 29 remains valid for all constants greater than or equal to 2 in this denominator (up to a modification of the VC-ratio's value).

(*ii*) *if, on the contrary, a ρ-approximation polynomial time algorithm exists for S_3, then there exist algorithms for both VC and WVC guaranteeing approximation ratios smaller than or equal to* $2 - (\rho/6) < 2$.

The proofs of Theorems 28 and 29 work for every fixed κ. Moreover, one can even relax the requirement of the (universally) constant-ratio approximation of S_κ by allowing this ratio to be a function of κ. Even if we suppose the existence of such a ratio, the two results always work (let us note that such a ratio remains constant with respect to n).

So, the most interesting open problem of this part is to make precise the infinum κ_0 of the values of κ for which S_κ does not admit a polynomial time approximation algorithm of constant (independent of n) ratio.

Our guess is that S_κ is not constant-approximable for any $\kappa > 2$, unless P = NP. In any case, with arguments similar to the ones used for the proof of Theorem 28, one can easily prove the following Proposition 2, supporting Hochbaum's [1983] conjecture.

PROPOSITION 2. *If S_2 is nonconstant approximable in polynomial time, then no polynomial time approximation algorithm for VC can guarantee an approximation ratio* $\rho < 2 - \epsilon$, *for a fixed positive constant* ϵ.

ACKNOWLEDGMENTS

I would like to thank Lalo Fernandez de la Vega for our long discussions concerning this article. Also, many thanks to an anonymous referee for very helpful comments and suggestions.

REFERENCES

AHO, A. V., HOPCROFT, J. E., AND ULLMAN, J. D. 1975. *The Design and Analysis of Computer Algorithms*. Addison-Wesley, Reading, MA.

AJTAI, M., ERDOS, P., KOMLÓS, J., AND SZEMERÉDI, E. 1981. On Turán's theorem for sparse graphs. *Combinatorica 1* (4), 313–317.

ARORA, S., LUND, C., MOTWANI, R., SUDAN, M., AND SZEGEDY, M. 1992. Proof verification and intractability of approximation problems. In *Proceedings FOCS'92*, 14–23.

ASPVALL, B. AND STONE, R. E. 1980. Khachiyan's linear programming algorithm. *J. Alg. 1*, 1–13.

BAKER, B. S. 1994. Approximation algorithms for NP-complete problems on planar graphs. *J. ACM 41*, 1, 153–180.

BAR-YEHUDA, R. AND EVEN, S. 1981. A linear-time approximation algorithm for the weighted vertex cover problem. *J. Alg. 2*, 198–203.

BAR-YEHUDA, R. AND EVEN, S. 1985. A local-ratio theorem for approximating the weighted vertex cover problem. *Ann. Discr. Appl. Math. 25*, 27–46.

BELLARE, M., GOLDREICH, O., AND SUDAN, M. 1995. Free bits, and non-approximability—towards tight results. Tech. Rep., preliminary version, May.

BELLARE, M. AND SUDAN, M. 1994. Improved non-approximability results. In *Proceedings STOC'94*, 184–193.

BERGE, C. 1973. *Graphs and Hypergraphs.* North Holland, Amsterdam.

BERMAN, P. AND FÜRER, M. 1994. Approximating maximum independent set in bounded degree graphs. In *Proceedings Symposium on Discrete Algorithms*, 365–371.

BERMAN, P. AND SCHNITGER, G. 1992. On the complexity of approximating the independent set problem. *Inf. Comput. 96*, 77–94.

BLOT, J., FERNANDEZ DE LA VEGA, W., PASCHOS, V. T., AND SAAD, R. 1995. Average case analysis of greedy algorithms for optimisation problems on set systems. *Theor. Comput. Sci. 147*, 267–298.

BOLLOBÁS, B. 1985. *Random Graphs.* Academic Press, London.

BOPPANA, B. B. AND HALLDÖRSSON, M. M. 1992. Approximating maximum independent sets by excluding subgraphs. *BIT 32*, 2, 180–196.

BOURJOLLY, J.-M., HAMMER, P. L., AND SIMEONE, B. 1984. Node-weighted graphs having the König-Egervary property. *Math. Prog. Study 22*, 44–63.

BROOKS, R. L. 1941. On coloring the nodes of a network. *Proc. Cambridge Philos. Soc. 37*, 194–197.

CHVÁTAL, V. 1979. A greedy-heuristic for the set covering problem. *Math. Oper. Res. 4*, 233–235.

CHVÁTAL, V. 1983. *Linear Programming.* W. H. Freeman, New York.

CRESCENZI, P. AND PANCONESI, A. 1989. Completeness in approximation classes. In *Proceedings of Fundamentals of Computation Theory, FCT'89*, LNCS, 380, Springer Verlag, 116–126.

DEMANGE, M. AND PASCHOS, V. T. 1995a. Exact and approximation results on maximum independent set and minimum vertex covering—graphs with great stability number. *Cahier du LAMSADE* 128, Université Paris-Dauphine.

DEMANGE, M. AND PASCHOS, V. T. 1995b. The approximability behaviour of some combinatorial problems with respect to the approximability of a class of maximum independent set problems. *Computational Optimization and Applications* (to appear).

DEMANGE, M. AND PASCHOS, V. T. 1995c. Valeurs extrémales d'un problème d'optimisation combinatoire et approximation polynomiale. *Mathématiques, Informatiques et Sciences Humaines* (to appear).

DEMANGE, M. AND PASCHOS, V. T. 1996a. Constructive—non-constructive approximation and maximum independent set problem. In post-conference proceedings of CCP'95, LNCS, (to appear).

DEMANGE, M. AND PASCHOS, V. T. 1996b. Improved approximations for maximum independent set via approximation chains. *Appl. Math. Lett.* (to appear).

DEMING, R. W. 1979. Independence numbers of graphs—an extension of the König-Egervary theorem. *Discrete Math. 27*, 23–33.

EVEN, S. 1979. *Graph Algorithms.* Computer Science Press, New York.

ERDÖS, P. 1959. Graph theory and probability. *Canad. J. Math. 11*, 34–38.

ERDÖS, P. AND SZEKERES, G. 1935. A combinatorial problem in geometry. *Compositio Mathematica 2*, 463–470.

FAJTLOWICZ, S. 1978. On the size of independent sets in graphs. In *Proceedings S. E. Conference on Combinatorics, Graph Theory and Computing*, 269–274, 1978.

FEIGE, U. 1995. Randomized graph products, chromatic numbers and the Lovász theta function. In *Proceedings STOC'95.*

FERNANDEZ DE LA VEGA, W. 1982. Sur la cardinalité maximum des couplages d'hypergraphes aléatoires uniformes. *Discrete Math. 40*, 315–318.

FERNANDEZ DE LA VEGA, W., PASCHOS, V. T., AND SAAD, R. 1992. Average case analysis of a greedy algorithm for the minimum hitting set problem. In *Proceedings LATIN'92*, Lecture Notes in Computer Science 583, Springer Verlag, 130–138.

GALIL, Z. 1978. A new algorithm for the maximal flow problem. In *Proceedings FOCS'78*, 231–248.

GAREY, M. R. AND JOHNSON, D. S. 1979. *Computers and Intractability. A Guide to the Theory of NP-Completeness.* W. H. Freeman, San Francisco.

GAVRIL, F. 1979. Cited in Garey, M. R. and Johnson, D. S. 1979. *Computers and Intractability. A Guide to the Theory of NP-Completeness. W. H. Freeman, San Francisco, 134.*

GOLDSMIDT, O., HOCHBAUM, D. S., AND YU, G. 1993. A modified greedy heuristic for the set covering problem with improved worst case bound. *Inf. Process. Lett. 48*, 305–310.

GRÖTSCHEL, M., LOVÁSZ, L., AND SCHRIJVER, A. 1981. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica 1*, 169–197.

HAJNAL, A. AND SZEMERÉDI, E. 1970. Proof of a conjecture of P. Erdös. In *Combinatorial Theory and its Applications, Balatonfüred*, P. Erdös, A. Rényi, and V. T. Sos, Eds., North Holland, Amsterdam, 601–623.

HALLDÖRSSON, M. M. 1995a. Approximating set cover via local improvements. JAIST Res. Rep. IS-RR-95-002F, Japan Advanced Institute of Science and Technology, Japan.

HALLDÖRSSON, M. M. 1995b. Approximating discrete collections via local improvements. In *Proceedings of the Symposium on Discrete Algorithms*, 160–169.

HALLDÖRSSON, M. M. 1995c. Approximations via partitioning. JAIST Res. Rep. IS-RR-95-0003F, Japan Advanced Institute of Science and Technology, Japan.

HALLDÖRSSON, M. M. AND RADHAKRISHNAN, J. 1994a. Greed is good: Approximating independent sets in sparse and bounded-degree graphs. In *Proceedings of STOC'94*, 439–448.

HALLDÖRSSON, M. M. AND RADHAKRISHNAN, J. 1994b. Improved approximations of independent sets in bounded-degree graphs via subgraph removal. *Nordic J. Comput. 1*, 4, 475–492.

HOCHBAUM, D. S. 1982. Approximation algorithms for the set covering and vertex cover problems. *SIAM J. Comput. 11*, 3, 555–556.

HOCHBAUM, D. S. 1983. Efficient bounds for the stable set, vertex cover and set packing problems. *Discr. Appl. Math. 6*, 243–254.

HOPKINS, G. AND STATON, W. 1982. Girth and independence ratio. *Canad. Math. Bull. 25*, 2, 179–186.

JOHNSON, D. S. 1974. Approximation algorithms for combinatorial problems. *J. Comput. Syst. Sci. 9*, 256–278.

JOHNSON, D. S. 1992. The NP-completeness column: An ongoing guide. *J. Alg. 13*, 502–524.

KANN, V. 1992. On the approximability of NP-complete optimization problems. Ph.D. Thesis, Dept. of Numerical Analysis and Computing Science, Royal Institute of Technology, Stockholm, Sweden.

KARP, R. M. 1972. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, R. E. Miller and J. W.

Thatcher, Eds., Plenum Press, New York, 85–103.

KARP, R. M. 1976. The probabilistic analysis of some combinatorial search algorithms. In *Algorithms and Complexity: New Directions and Recent Results*, J. F. Traub, Ed., Academic Press, New York, 1–19.

KHANNA, S., MOTWANI, R., SUDAN, M., AND VAZIRANI, U. 1994. On syntatic versus computational views of approximability. *Proceedings of FOCS*.

LOVÁSZ, L. 1975a. Three short proofs in graph theory. *J. Comb. Theory (B) 19*, 269–271.

LOVÁSZ, L. 1975b. On the ratio of optimal integral and fractional covers. *Discrete Math. 13*, 383–390.

LOVÁSZ, L. 1979. On the Shannon capacity of a graph. *IEEE Trans. Inf. Theory 25*, 1, 1–7.

LOVÁSZ, L. 1966. On decomposition of graphs. *Stud. Sci. Math. Hung. 1*, 237–238.

LUND, C. AND YANNAKAKIS, M. 1992. On the hardness of approximating minimization problems. AT&T Bell Laboratories, (preprint).

MONIEN, B. AND SPECKENMEYER, E. 1985. Ramsey numbers and an approximation algorithm for the vertex cover problem. *Acta Inf. 22*, 115–123.

MOTWANI, R. 1993. *Lecture Notes on Approximation Algorithms—Volume I.* Stanford University, Stanford, CA.

NEMHAUSER, G. L. AND TROTTER, L. E., JR. 1975. Vertex packings: Structural properties and algorithms. *Math. Program. 8*, 232–248.

ORPONEN, P. AND MANNILA, H. 1987. On approximation preserving reductions: Complete problems and robust measures. Tech. Rep. C-1987-28, Dept. of Computer Science, University of Helsinki, Finland.

PAPADIMITRIOU, C. H. 1994. *Computational Complexity*, Addison-Wesley, Reading, MA.

PAPADIMITRIOU, C. H. AND STEIGLITZ, K. 1981. *Combinatorial optimization: Algorithms and complexity.* Prentice-Hall, Englewood Cliffs, NJ.

PAPADIMITRIOU, C. H. AND YANNAKAKIS, M. 1988. Optimization, approximation and complexity classes. In *Proceedings to STOC'88*, 229–234.

PAPADIMITRIOU, C. H. AND YANNAKAKIS, M. 1993. The travelling salesman problem with distances one and two. *Math. Oper. Res. 18.*

PASCHOS, V. T. 1992. A $\Delta/2$-approximation algorithm for the maximum independent set problem. *Inf. Process. Lett. 44*, 11–13.

PASCHOS, V. T. 1993. *Notes de cours en optimisation combinatoire.* DEA "Méthodes Scientifiques de Gestion," Université Paris-Dauphine.

PASCHOS, V. T. 1994. A relation between the approximated versions of minimum set covering, minimum vertex covering and maximum independent set. *RAIRO Oper. Res. 28*, 4, 413–433.

PASCHOS, V. T. AND DEMANGE, M. 1994. Approximation algorithms for minimum set covering problem: A survey. *Found. Comput. Decis. Sci. 19* 3, 205–224.

PASCHOS, V. T. AND RENOTTE, L. 1995. Approximability preserving reductions for NP-complete problems. *Found. Comput. Decis. Sci. 20*, 1, 49–71.

PAZ, A. AND MORAN, S. 1981. Nondeterministic polynomial optimization problems and their approximations. *Theor. Comput. Sci. 15*, 251–277.

SAVAGE, C. 1982. Depth first search and the vertex cover problem. *Inf. Process. Lett. 14*, 233–235.

SHEARER, J. B. 1983. A note on the independence number of triangle-free graphs. *Disc. Math. 46*, 83–87.

SHEARER, J. B. 1991. A note on the independence number of triangle-free graphs. II. *J. Comb. Theory (B) 53*, 300–307.

SHMOYS, D. B. AND TARDOS, É. Computational complexity. In *The Handbook of Combinatorics*, R. L. Graham, M. Grötschel, and L. Lovász, Eds., North Holland, Amsterdam.

SIMON, H. U. 1990. On approximate solutions for combinatorial optimization problems. *SIAM J. Disc. Math. 3* 2, 294–310.

STATON, W. 1979. Some Ramsey-type numbers and the independence ratio. *Trans. Amer. Math. Soc. 256*, 353–370.

TURÁN, P. 1941. On an extremal problem in graph theory. *Mat. Fiz. Lapok 48*, 436–452 (Hungarian).

YU, G. AND GOLDSMIDT, O. 1993. On locally optimal independent sets and vertex covers. In *Proceedings of IFIP'93*, 533–542.

WIGDERSON, A. 1983. Improving the performance guarantee for approximate graph coloring. *J. ACM 30* 4, 729–735.

Received June 1994; revised July 1996; accepted December 1996