Journal of Graph Algorithms and Applications

http://www.cs.brown.edu/publications/jgaa/

vol. 4, no. 1, pp. 1–16 (2000)

Approximations of Weighted Independent Set and Hereditary Subset Problems

Magnús M. Halldórsson

Science Institute
University of Iceland
IS-107 Reykjavik, Iceland
http://www.hi.is/~mmh
mmh@hi.is

Abstract

The focus of this study is to clarify the approximability of weighted versions of the maximum independent set problem. In particular, we report improved performance ratios in bounded-degree graphs, inductive graphs, and general graphs, as well as for the unweighted problem in sparse graphs. Where possible, the techniques are applied to related hereditary subgraph and subset problem, obtaining ratios better than previously reported for e.g. Weighted Set Packing, Longest Common Subsequence, and Independent Set in hypergraphs.

Communicated by S. Khuller: submitted August 1999; revised April 2000.

Earlier version appears in COCOON '99 [12]. Work done in part at School of Informatics, Kyoto University, Japan.

1 Introduction

An *independent set*, or a stable set, in a graph is a set of mutually nonadjacent vertices. The problem of finding a maximum independent set in a graph, IndSet, is one of most fundamental combinatorial NP-hard problem. It serves also as the primary representative for the family of subgraph problems that are hereditary under vertex deletions. We are interested in finding approximation algorithms that yield good performance ratios, or guarantees on the quality of the solution they find vis-a-vis the optimal solution.

The focus of this paper is to present improved performance ratios for three major versions of the independent set problem: in weighted graphs, bounded-degree graphs and sparse graphs. We also apply some of the methods to a number of related (or not-so related) problems that obey certain hereditariness property, most of which had not been approximated before.

A considerable amount of research has been done on the approximability of IndSet in the last decade. It has been shown to be hard to approximate through advances in the study of interactive proof systems. In particular, Håstad [19] showed it hard to approximate within $n^{1-\epsilon}$, for any $\epsilon > 0$, unless NP-hard problems have randomized polynomial algorithms. The best performance ratio known is $O(n/\log^2 n)$, due to Boppana and Halldórsson [4].

For bounded-degree graphs, Halldórsson and Radhakrishnan [17] gave the first asymptotic improvement over maximal solutions, obtaining a ratio of $O(\Delta/\log\log\Delta)$. For small values of Δ, an algorithm of Berman and Fujito [3] attains the best bound known of $(\Delta+3)/5$. See the survey [13] for a more complete description of earlier results. The best asymptotic bound known is $O(\Delta\log\log\Delta/\log\Delta)$ due to Vishwanathan [31] (first recorded in [13]), combining two results on semi-definite programming due to Karger, Motwani and Sudan [22] and Alon and Kahale [2].

The current paper is divided into four independent section, each of which treats a different technique for finding independent sets. They are ordered both in chronological order of inquiry, as well as the depth of the solution technique. We first study in Section 2 an elementary general partitioning technique that yields nontrivial performance ratio for a large class of problems satisfying a property that we call *semi-heredity*. All results holds for for weighted versions of the problems. We obtain a $O(n/\log n)$ approximation for Independent Set in Hypergraphs, Longest Common Subsequence, Max Satisfying Linear Subsystem, and Max Independent Sequence. We strengthen the ratio for problems that do not contain a forbidden clique, obtaining a $O(n(\log\log n/\log n)^2)$ performance ratio for IndSet and Max Hereditary Subgraph. (All problems are defined in their respective sections.)

In Section 3, we consider another elementary strategy, partitioning the vertices into weight classes. It easily yields that weighted versions of semi-hereditary problems on any class of graphs are approximable within $O(\log n)$ of the respective unweighted case. However, this overhead factor reduces to a constant in the case of ratios in the currently achievable range, giving a $O(n/\log^2 n)$ ratio for WIS.

We consider in Section 3.1 the approximation of the weighted set packing problem (WSP), in terms of m, the number of base elements. We match the best ratio known for the unweighted case of $O(\sqrt{m})$. We also describe a simplified argument of Lehmann [23] with a better constant factor.

In Section 4, we consider approximations based on semi-definite programming (SDP) relaxations. We generalize the result of Vishwanathan [31] in two ways. First, we apply it to the weighted case, obtaining a $O(\Delta \log\log\Delta/\log\Delta)$ ratio for WIS. This improves on the previous best ratio of $(\Delta+2)/3$ due to Halldórsson and Lau [15]. Halperin [18] has independently obtained the same ratio, using different techniques. Our ratio also holds in terms of another parameter, $\delta(G)$, the inductiveness of the graph, giving a $O(\delta\log\log\delta/\log\delta)$ approximation of WIS. This improves on the previous best ratio known of $(\delta+1)/2$ due to Hochbaum [20]. For the other direction, we apply the technique to sparse unweighted graphs, obtaining a ratio of $O(\overline{d}\log\log\overline{d}/\log\overline{d})$, the first asymptotic improvement on Turán's bound [6, 20].

Notation Let $G=(V,E)$ be a graph, let n denote its number of vertices and let Δ ($\overline{d}$) denote its maximum (average) degree. WIS takes as input instance (G,w), where G is a graph and $w : V \mapsto \mathbf{R}$ is a vector of vertex weights, and asks for a set of independent vertices whose sum of weights is maximized. The maximum weight of an independent set in instance (G,w) denoted by $\alpha(G,w)$, or $\alpha(G)$ on unweighted graphs. Let $|S|$ denote the cardinality of a set S, and let $w(S)$ denote the sum of the weights of the elements of S. Let $w(G)$ denote $w(V(G))$.

We say that a problem is *approximable within* $f(n)$, if there is a polynomial time algorithm which on any instance with n distinguished elements returns a feasible solution within a $f(n)$ factor from optimal. We let OPT denote some optimal solution of the given problem instance and HEU the output of the algorithm under study on that same instance. We also overload those term to refer to the weight of those solutions.

2 Partitioning into easy subproblems

We consider a collection of problems that involve finding a feasible subset of the input of maximum weight. The input contains a collection of n *distinguished elements*, each carrying an associated nonnegative rational weight. Each set of distinguished elements uniquely induces a candidate for a *solution*, which we assume is efficiently computable from the set. The weight of a solution is the sum of the weights of the distinguished elements in the solution.

A property is said to be *hereditary* if whenever a set S of distinguished elements corresponds to a feasible solution, any subset of S also corresponds to a feasible solution. A property is *semi-hereditary* if under the same circumstances, any subset S' of S uniquely induces a feasible solution, possibly corresponding to a superset of S'.

To illustrate the concept of semi-hereditarity, consider the problem Maximum Common Subtree [1]. Given is a collection of n free trees, and we are to find a tree that is isomorphic to a subtree (i.e. connected induced subgraph) of each input tree. Verifying if a particular tree is isomorphic to a subtree of another tree is polynomial solvable. Consider the vertices of the first input tree as the distinguished elements. A given subset of these vertices is not necessarily a proper solution, but it uniquely induces a tree that minimally connects the vertices of the subset. Thus, the additional power of the semi-hereditary property is necessary to capture this problem.

Hereditary graph properties are special cases of these definitions. A property of graphs is *hereditary* if whenever it holds for a graph it also holds for its induced subgraphs. For a hereditary graph property, the associated subgraph problem is that of finding a subgraph of maximum vertex-weight satisfying the property. Here, the vertices form the distinguished elements.

Our key tool is a simple partitioning idea, that has been used in various contexts before.

Proposition 2.1 *Let Π be a semi-hereditary subset property. Suppose that given an instance I, we can produce t instances $I_1, I_2, \ldots, I_t$ that cover the set of distinguished elements (i.e. each distinguished element is contained in at least one I_i). Further, suppose we can solve exactly the maximum Π-subset problem on each I_i. Then, the largest of these t solutions yields an approximation of the maximum Π-subset of I within t.*

In the remainder of this section we describe applications of this approach to a number of particular problems.

2.1 Partition into small subsets

Proposition 2.2 *Let Π be a semi-hereditary property for which feasibility can be decided in time at most polynomial in the size of the input and at most simply exponential in the number of distinguished elements. Then, the maximum weighted Π-subgraph can be approximated within $n/\log n$.*

We achieve this by arbitrarily partitioning the set of distinguished elements into $n/\log n$ sets each with $\log n$ elements. For each subset of each set, obtain the candidate solution for this subset and determine feasibility. By our assumptions, each step can be done in polynomial time, and in total at most $2^{\log n} \cdot n/\log n = n^2/\log n$ sets are generated and tested. By this procedure, we find optimal solutions within each of the $n/\log n$ sets. Since the optimal solution of the whole is divided among these sets, the performance ratio is at most $n/\log n$.

Surprisingly, this $n/\log n$-approximation appears to be the best that is known for most such problems. A property is *nontrivial* if it holds for some graphs and fails for others. It is known that, the subgraph problem for any nontrivial hereditary property cannot be approximated within any constant unless $P = NP$, and stronger results hold for properties that fail for some clique or some independent set [25].

We apply Proposition 2.2 to several problems featured in the compendium on optimization problems [5]:

Weighted Independent Sets in Hypergraphs Given a hypergraph, or a set system, $(S, \mathcal{C})$ where S is a set of weighted base elements (vertices) and $\mathcal{C} = \{C_1, C_2, \ldots, C_n\}$ is a collection of subsets of S, find a maximum weight subset S' of vertices such that no subset C_i is fully contained in S'.

Hofmeister and Lefmann [21] analyzed a Ramsey-theoretic algorithm generalizing that of [4], and showed its performance ratio to be $O(n/(\log^{(r-1)} n))$ for the case of r-uniform hypergraphs. It is straightforward to verify the heredity thus a $O(n/\log n)$ performance ratio holds by Proposition 2.1.

Longest Common Subsequence Given a finite set R of strings from a finite alphabet Σ, find a longest possible string w that is a subsequence of each string x in R. The problem is clearly hereditary, and feasibility can be tested for each string x in R separately via dynamic programming. Hence, by applying Proposition 2.2, partitioning the smallest string in the input, we obtain a performance ratio of $O(m/\log m)$, where m is the size of the smallest string.

Max Satisfying Linear Subsystem Given a system $Ax = b$ of linear equations, with A an integer $m \times n$ matrix and b an integer m vector, find a rational vector $x \in \mathcal{Q}^n$ that satisfies the maximum number of equations.

This problem is clearly hereditary, since any subset of a feasible collection of equations is also feasible. Feasibility of a given system can be solved in polynomial time via linear programming. Hence, $O(m/\log m)$ approximation follows from Proposition 2.2. This holds equally if equality is replaced by inequalities $(>, \geq)$. It also holds if a particular set of constraints/equations are required to be satisfied by a solution.

Max Independent Sequence Given a graph, find a maximum length sequence $v_1, v_2, \ldots, v_m$ of independent vertices such that, for all $i < m$, a vertex v'_i exists which is adjacent to v_{i+1} but is not adjacent to any v_j for $j \leq i$. This problem was introduced by Blundo (see [5]).

First observe that solutions to the problem are hereditary: if $v_1, v_2, \ldots, v_m$ is an independent sequence, then so is any subsequence $v_{a_1}, v_{a_2}, \ldots, v_{a_x}$. This is because, for all $i < x$, there exists a node v'_i that is adjacent to $v_{a_{i+1}}$ but not adjacent to any v_j for $j < a_{i+1}$ and hence not to any v_{a_j} for $j \leq i$. Feasibility of a solution can be tested in time polynomial in the size of the input. Independence is easily tested by testing all pairs in the proposed solution. A valid set can be turned into a valid sequence by inductively finding the element adjacent to a vertex outside the set that is adjacent to no other unselected vertex.

Thus, we obtain an $O(n/\log n)$ approximation via Proposition 2.2. We can also argue strong approximation hardness bounds.

Proposition 2.3 MAX INDEPENDENT SEQUENCE *is no easier than* INDSET*, within 2. Thus, it is hard to approximate within* $n^{1-\epsilon}$*, for any* $\epsilon > 0$*, unless* $NP = ZPP$*.*

Proof. Given a graph G on vertices $v_1, v_2, \ldots, v_n$, the graph H_G consists of G and n additional vertices $\{w_1, w_2, \ldots, w_n\}$ connected into a clique, with $(v_i, w_j) \in E(H_G)$ iff $i \geq j$. Then, any independent set in G corresponds to an independent sequence in H_G. The converse is also true, with the possible exclusion of one w_i vertex; in that case, we can replace that w_i vertex with some v_j vertex that must exist and be independent of the other v-vertices in the set. Hence, we get a size-preserving reduction. The new graph contains twice as many vertices, thus the performance ratio lower bound is weaker for MAX INDEPENDENT SEQUENCE by a factor of 2. The hardness now follows from the result of Håstad [19] on INDSET. □

Theorem 2.4 *Weighted versions of* INDSET IN HYPERGRAPHS*,* MAX HEREDITARY SUBGRAPH *and* MAX INDEPENDENT SEQUENCE *can be approximated within* $O(n/\log n)$*.*

2.2 Weighted Independent Sets and Other Hereditary Graph Properties

A theorem of Erdős and Szekeres [7] on Ramsey numbers yields an efficient algorithm [4] for finding either cliques or independent sets of nontrivial size.

Fact 2.5 (Erdős, Szekeres) *Any graph on n vertices contains a clique on s vertices or an independent set on t vertices such that* $\binom{s+t-2}{s-1} \geq n$*.*

We use this theorem to approximate a large class of hereditary subgraph problems.

Theorem 2.6 MAX WEIGHTED HEREDITARY SUBGRAPH *can be approximated within* $O(n(\log\log n/\log n)^2)$*, for properties that fail for some cliques or some independent set.*

Proof. Let n denote here the size of the input graph G to the MAX WEIGHTED HEREDITARY SUBGRAPH problem. We say that a graph is *amenable* if it is either an independent set or consists of at most $\log n/\log\log n$ disjoint cliques. Theorem 2.5 implies that we can find in G either an independent set of size at least $\log^2 n$, or a clique of size at least $\log n/2\log\log n$. Thus we can find an amenable subgraph of size $X = \log^2 n/3\log\log n$, by at most $\log n$ applications of Theorem 2.5.

We then pull these amenable subgraphs one by one from G, obtaining a partition of G into amenable subgraphs. The number of subgraphs in the partition will be at most $3n/X$. Namely, at most $n/(\log^2(n/X)/3\log\log n) = n/X(1 + o(1))$ subgraphs are found before the size of G drops below n/X and the remainder is at most another n/X.

We can solve WIS on an amenable subgraph by exhaustively checking all $(\log n/\log\log n)^{\log n/\log\log n} = O(n)$ possible combinations of selecting up to one vertex from each clique. More generally, assume without loss of generality that our hereditary subgraph property fails for cliques of size s. We can solve it optimally on an amenable subgraph by exhaustively checking all combinations of selecting at most $s-1$ vertices from each clique. That number is still at most $(\log n/\log\log n)^{s\log n/\log\log n}$, which is $poly(n)$ for fixed s. In the case that the property fails for some independent set, we exchange the roles of independent sets and cliques in our partitioning routine with no change in the results. □

Examples of such properties include: bipartite, k-colorable, k-clique free, planar.

2.3 Limitations of partitioning

The wide applicability of this partitioning technique might offer a glimmer of hope for approximating the independent set problem in general graphs within $n^{1-\epsilon}$, for some $\epsilon > 0$. The following observation casts a shade on that proposal.

For a property Π, the Π-chromatic number of a graph is the minimum number of classes that the vertex set can be partitioned into such that the graph induced by each class satisfies Π. Scheinerman [29] has shown that for any nontrivial hereditary property Π, the Π-chromatic number of a random graph approaches $\theta(n/\log n)$. This indicates that our results are essentially the best possible.

3 Partitioning into weight classes

We now consider a simple general strategy for obtaining approximations to weighted subgraph problems, that always comes within a $\log n$ factor from the unweighted case and often within less.

Theorem 3.1 *Let Π be a hereditary subgraph problem. Suppose Π can be approximated within ρ on unweighted graphs (or on a subclass thereof). Then, the vertex-weighted version can be approximated within $O(\rho \cdot \log n)$.*

Proof. Consider the following strategy. Let W be the maximum vertex weight. Delete all vertices of weight at most W/n. Let V_i be the set of vertices whose weight lies in $(W/2^i, W/2^{i-1}]$, for $i = 1, 2, \ldots, \lg n$. Run the ρ-approximate algorithm on the V_i, ignoring the weights. Output the maximum weight solution, denoted by HEU.

We claim that the performance ratio of this method is at most $2\rho\lg n + 1$. First, note that the set of vertices of small weight adds up to at most W, or less than that of HEU. Second, if G' is the graph induced by vertices of weight more than W/n,

$$OPT(G') \leq \sum_{i=1}^{\lg n} OPT(V_i) \leq \sum_{i=1}^{\lg n} 2\rho\, HEU(V_i) = 2\rho\, HEU(G),$$

where the additional factor of 2 comes from the rounding of the weights. □

We note that the logarithmic loss in approximation is caused by a logarithmic decrease in subgraph sizes. However, when the performance function is close to linear, as is the case today, decrease in subgraph size affects performance only slightly. We illustrate this with WIS, matching the known approximation for unweighted graphs.

Theorem 3.2 *If a hereditary subgraph problem can be approximated within $g(n) = n^{1-\Omega(1/\log\log n)}$, then its weighted version can also be approximated within $O(g(n))$. In particular,* WIS *can be approximated within $O(n/\log^2 n)$.*

Proof. Let G be a graph partitioned into subgraphs $V_1, \ldots, V_{\log n}$ as in Theorem 3.1, let OPT be an optimal solution and HEU the heuristic solution found. Observe that the function g satisfies $g(N) = O(g(n) \cdot N/n)$ when $N \geq n/\lg n$, and $g(N) = O(g(n)/\log n)$ when $N \leq n/\lg n$,

Let L be the set of indices ℓ that satisfy

$$w(V_\ell \cap OPT) \geq w(OPT)/2\lg n, \tag{1}$$

and note that $\sum_{i \in L} w(V_i \cap OPT) \geq w(OPT)/2$.

Suppose that for some $\ell' \in L$, $|V_{\ell'}| < n/\lg n$. By (1), $w(V_i \cap OPT) \leq w(OPT) \leq (2\lg n) w(V_{\ell'} \cap OPT)$, for all i. Thus,

$$\rho \leq \frac{w(OPT)}{w(HEU)} \leq 4\lg n \frac{w(V_{\ell'} \cap OPT)}{w(HEU)} \leq 4\lg n \cdot g(|V_{\ell'}|) = O(g(n)).$$

Otherwise, $g(|V_\ell|) = O(g(n) \cdot |V_\ell|/n)$ for all $\ell \in L$. Then,

$$\begin{aligned} \rho &\leq \sum_i \frac{w(V_i \cap OPT)}{w(HEU)} \leq 2\frac{\sum_{\ell \in L} w(V_\ell \cap OPT)}{w(HEU)} \\ &\leq 2\sum_i g(|V_i|) = \frac{g(n)}{n} \sum_{\ell \in L} O(|V_\ell|) = O(g(n)). \end{aligned}$$

The $O(n/\log^2 n)$ ratio for WIS now follows from the result of [4] for the unweighted case. □

3.1 Weighted Set Packing

The WSP problem is as follows. Given a set S of m base elements, and a collection $\mathcal{C} = \{C_1, C_2, \ldots, C_n\}$ of weighted subsets of S, find a subcollection $\mathcal{C}' \subseteq \mathcal{C}$ of disjoint sets of maximum total weight $\sum_{C'_i \in \mathcal{C}'} w(C'_i)$. A variety of applications of this problem to practical optimization problems is surveyed in [30]. It has recently been used to model multi-unit combinatorial auctions [27, 10] and and in the formation of coalitions in multiagent systems [28].

By forming the intersection graph of the given hypergraph (with a vertex for each set, and two vertices being adjacent if the corresponding sets intersect), a weighted set packing instance can be transformed to a weighted independent set instance on n vertices. Hence, approximations of WIS — as a function of n — carry over to WSP.

For approximations of unweighted set packing *as a function of* m $(= |S|)$, Halldórsson, Kratochvíl, and Telle [14] gave a simple $\sqrt{m}$-approximate greedy algorithm, and noted that $m^{1/2-\epsilon}$-approximation is hard via [19]. We observe that the positive results hold also for the weighted case, by a simple variant of the greedy method.

Theorem 3.3 WSP *can be approximated within* $2\sqrt{m}$ *in time proportional to the time it takes to sort the weights.*

Proof. The algorithm initially removes all sets of cardinality $\sqrt{m}$ or more. It then greedily selects sets of maximum weight that are disjoint from the previously selected sets.

```
SetPackingApprox(S,C)
  Max ← the set in C of maximum weight
  C ← {C ∈ C : |C| ≤ √m}
  Output the larger of GreedySP(S,C) and Max
end
GreedySP(S,C)
  t ← 0, C_t ← C
  repeat
    t ← t + 1
    X_t ← C ∈ C_{t−1} of maximum weight
    Z_t ← {C ∈ C_{t−1} : X ∩ C ≠ ∅}
    C_t ← C_{t−1} − Z_t
  until C = ∅
  return {X_1, X_2, ..., X_t}
end
```

Figure 1: Greedy set packing algorithm

Consider Z_t, the sets eliminated in some iteration i. Observe that the optimal solution contains at most $\sqrt{m}$ sets from Z_t (since sets in Z_t have an element in common with X_t which is of cardinality at most $\sqrt{m}$), all of which are of weight at most that of X_t, the set chosen by the algorithm. Hence, in every iteration, the contribution added to the algorithm's solution is at least $\sqrt{m}$-th fraction of what the optimal solution could get.

Also, the optimal solution contains at most $\sqrt{m}$ sets among those eliminated in the second line of SetPackingApprox, since each of them is of cardinality at least $\sqrt{m}$. Since the algorithm contains at least the weight of the maximum weight set, this is at most $\sqrt{m}$ times the algorithm's solution. Combined, the optimal solution is of weight at most $2\sqrt{m}$ times the algorithm's solution. $\square$

We now describe an improvement due to Lehmann [23] that shows that the greedy algorithm can be modified to give a slightly better ratio of $\sqrt{m}$ by itself. The modification to GreedySP is to change line 4 to

$$X_t \leftarrow C \in \mathcal{C}_{t-1} \text{ that maximizes } w(C)/\sqrt{|C|}.$$

Let OPT be some optimal set packing solution. Consider any iteration t of the algorithm, and let OPT_t be the sets in $OPT \cap Z_t$. Note first, that for any set $C \in \mathcal{C}_{t-1}$,

$$w(C) \leq \sqrt{|C|}\frac{w(X_t)}{\sqrt{|X_t|}},$$

because of how X_t was chosen. Thus,

$$w(OPT_t) = \sum_{C \in OPT_t} w(C) \leq \frac{w(X_t)}{\sqrt{|X_t|}} \sum_{C \in OPT_t} \sqrt{|C|}.$$

Since the sets in OPT_t must be disjoint and of total cardinality at most m, the sum on the right hand side is maximized when all the sets are of equal size. This gives

$$w(OPT_t) \leq \frac{w(X_t)}{\sqrt{|X_t|}}\sqrt{|OPT_t| \cdot m}.$$

Note that OPT_t contains at most one set for each element of X_t, so $|OPT_t| \leq |X_t|$. Hence, $w(OPT_t) \leq \sqrt{m}\, w(X_t)$. Since this holds for each iteration, a ratio of $\sqrt{m}$ follows. Gonen and Lehmann [10] show that no greedy algorithm can obtain a better ratio.

One can also observe that the constant factor can be arbitrarily improved, if one can afford a commensurate increase in the polynomial complexity. Modify SetPackingApprox to set Max as the maximum weight set packing in $(S, \mathcal{C})$ containing at most s sets. Also, change the upper bound on the cardinality of sets to be included in $\mathcal{C}$ from $\sqrt{m}$ to $q = \sqrt{m/s}$. To analyze this, let us split the optimal packing into a packing of sets of size greater than q and that of sets at most q. A packing of the former can contain at most $m/q = \sqrt{sm}$ sets, hence Max approximates it within $\sqrt{m/s}$ factor. Also, we know that GreedySP approximates the latter within the same factor. The better of the two solutions now yields a $2\sqrt{m/s}$ approximation.

4 Semi-definite programming

A fascinating polynomial-time computable function $\vartheta(G)$ introduced by Lovász [24] has the remarkable "sandwiching" property that it always lies between two NP-hard functions, $\alpha(G) \leq \vartheta(G) \leq \overline{\chi}(G)$. This property suggests that it may be particularly suited for obtaining good approximations to either function. While some of those hopes have been dashed [8], a number of fruitful applications have been found and it remains the most promising candidate for obtaining improved approximations [9].

Karger, Motwani and Sudan [22] gave improved approximations for k-colorable graphs via the theta function, followed by Alon and Kahale [2] that obtained improved approximations for IndSet in the case of linear-sized independent sets. Mahajan and Ramesh [26] showed how these and related algorithms can be derandomized. Vishwanathan [31] observed that an improved performance ratio for the independent set problem of $O(\Delta \log\log\Delta/\log\Delta)$ could be obtained by combining together theorems of [22, 2].

We illustrate here how the improved $O(\Delta \log\log\Delta/\log\Delta)$ also applies to weighted independent sets. For this purpose, we give straightforward generalizations of the results of [22] and [2]. Halperin [18] has independently obtained the same bound, using a different rounding procedure.

We actually prove a stronger bound. A graph is said to be δ*-inductive* if, there is a linear ordering of the vertices such that each vertex has at most δ neighbors ordered after itself. We obtain a $O(\delta\log\log\delta/\log\delta)$ approximation of WIS, improving on the previous best $(\delta+1)/2$ [20].

The Lovász number $\vartheta(G)$ of a graph G is the least number k such that there exists a representation of unit vectors v_i to each vertex $i \in V$, such that for any two nonadjacent vertices i and j the dot product of their vectors satisfies the equality

$$(v_i \cdot v_j) = -\frac{1}{k}.$$

Such a representation can be computed in polynomial time, or more precisely, one can obtain a representation that is arbitrarily close. We shall use several other definitions of this measure later, that were introduced in the original paper of Lovász [24]; the current one was defined in [22] as the *strict vector chromatic number*.

We first give a weighted version of a result of [22]. For our purposes, it suffices to use the simpler method of "rounding by hyperplanes", as the constant in the exponent is not important.

Proposition 4.1 *Let G be a weighted δ-inductive graph satisfying $\vartheta(\overline{G}) \leq k$. Then an independent set in G of weight $\Omega(w(G)/\delta^{1-1/2k})$ can be constructed with high probability in polynomial time.*

Proof. The inductiveness of the graph implies that its edges can be directed so that each vertex has outdegree at most d, and thus we say it has at most d *out-neighbors*. We assume such a direction on the edges.

¿From the bound on $\vartheta(\overline{G})$, we can represent the vertices as Euclidean vectors, such that for *adjacent* vertices i and j the corresponding vectors v_i and v_j satisfy $(v_i \cdot v_j) = -\frac{1}{k}$. Given such a representation, the algorithm selects r hyperplanes at random (by choosing uniformly random vectors on the unit sphere around the origin as normals), dividing $\mathbf{R}^n$ into 2^r partitions. The algorithm examines each of the partitions, collects the set of vertices with no out-neighbors in the same partition, and outputs the set of maximum weight. We give a lower bound on the expected weight of this set.

Let $q = \frac{1}{2} + \frac{1}{\pi k}$ and let $r = 2 + \lceil \log_{1/(1-q)} \delta \rceil$. Note that $(1-q)^r \le 1/4\delta$. Also, it can be shown that $1/\lg 1/(1-q) \le 1 - 1/2k$, using that $1/(1-q) = 2(1 + 1/(\pi k/2 - 1))$ and that $\ln(1+x) \ge x/(1+x)$. That means that

$$2^r \le 8\delta^{1/\lg 1/(1-q)} = O(\delta^{1-1/2k}).$$

The probability that a random hyperplane separates the vectors associated with two vertices is ϕ/π, where ϕ is the angle between the vectors. When the vertices are adjacent, this probability is

$$\frac{arccos(-1/k)}{\pi} \le \frac{1}{2} + \frac{1}{\pi k} = q,$$

where the inequality is obtained from the Taylor expansion of $arccos(x)$. Hence, the probability that none of the r hyperplanes cuts a given edge is at most

$$p = (1-q)^r \le \frac{1}{4\delta}.$$

The probability, for a given vertex v, that all of v's outedges are cut is then at least $1 - \delta p \ge 1 - 1/4$. Consider, for a given partition P, the independent set I of vertices with no out-neighbors (i.e. outdegree zero) within its partition. If $w(P)$ denotes the weight of a given partition P, the expected weight of I is at least $w(P)(1 - 1/4)$. Averaging over the 2^r partitions, the expected weight of the set output is at least

$$\frac{w(G)(1-1/4)}{2^r} = \Omega(w(G)/\delta^{1-1/2k}).$$

□

We now consider a generalization of the Lovász number to weighted graphs. An *orthonormal representation* of a graph $G = (V, E)$ is an assignment of a unit vector b_v in Euclidean space to each vertex v of G, such that $b_u \cdot b_v = 0$ if $u \ne v$ and $(u, v) \notin E$. The (weighted) theta function $\vartheta(G, w)$ [11] equals the minimum over all unit vectors d and all orthonormal labelings b_v of

$$\max_{v \in V} \frac{w(v)}{(d \cdot b_v)^2}.$$

An equivalent dual characterization is to define it as the maximum over all unit vectors d and all orthonormal representations b_v of the complement graph $\overline{G}$ of $\sum_{v \in V} (d \cdot b_v)^2 w(v)$. The Lovász number $\vartheta(G)$ is $\vartheta(G, \mathbf{1})$, the theta function on the unit-weighted graph.

Proposition 4.2 *If $\vartheta(G, w) \ge 2w(G)/k$ (e.g. if $\alpha(G, w) \ge 2w(G)/k$), then we can find an induced subgraph K in G such that $\vartheta(\overline{K}) \le k$ and $w(K) \ge w(G)/k$.*

Proof. We emulate [2]. Let d be a unit vector and b_v a representation such that $\vartheta(G,w) = \sum_{v \in V}(d \cdot b_v)^2 w(v)$. For each v, let a_v denote $(d \cdot b_v)^2$. We can then split the sum for $\vartheta(G,w)$ into two parts: those where a_v is small or at most $1/k$ for some breakpoint k, and those where a_v is large. Thus,

$$\vartheta(G,w) = \sum_{a_v \le 1/k} a_v w(v) + \sum_{a_v \ge 1/k} a_v w(v) \le w(G)/k + \sum_{a_v \ge 1/k} a_v w(v).$$

Let K be the subgraph induced by vertices v with $a_v \ge 1/k$. If $\vartheta(G,w) \ge 2w(G)/k$, we have that since $a_v \le 1$ for each vertex v,

$$w(K) = \sum_{v \in V(K)} w(v) \ge \sum_{a_v \ge 1/k} a_v w(v) \ge \vartheta(G,w) - w(G)/k \ge w(G)/k.$$

Also,

$$\max_{v \in K} \frac{1}{(d \cdot b_v)^2} \le k,$$

hence the Lovász number of K is at most k by its definition. □

Theorem 4.3 WIS *can be approximated within* $O(\delta \log\log \delta / \log \delta)$.

Proof. Let (G,w) be an instance with $\alpha(G,w) = 2w(G)/k$, for some k. We find via Proposition 4.2 a subgraph K_k with $\vartheta(\overline{K_k}) \le k$ and $w(K_k) \ge w(G)/k$. We then find via Proposition 4.1 an independent set in $K_k \subset G$ of weight at least $\frac{w(G)/k}{\delta^{1-1/2k}}$. The approximation ratio is then at most $2\delta^{1-1/2k}$.

Alternatively, δ-inductive graphs are well-known to be $\delta+1$-colorable. Thus, the heaviest color class is an independent set of weight at least $w(G)/(\delta+1)$, for a $2(\delta+1)/k$ approximation. Observe that first ratio is increasing with k and the latter decreasing, with breakpoint achieved when $k = \frac{1}{2}\log\delta/\log\log\delta$, in which case both ratios are $O(\delta \log\log \delta / \log \delta)$. □

4.1 Sparse graphs

Inductive graphs can be thought of as being "everywhere sparse". For reasons of padding, it is not possible to get similar ratios for WIS on all sparse graphs. However, we can obtain this for the unweighted IndSet problem, improving on the previous best ratio known of $(2\overline{d}+3)/5$ [16].

Theorem 4.4 IS *can be approximated within* $O(\overline{d} \log\log \overline{d} / \log \overline{d})$.

Proof. For a graph G of average degree $\overline{d}$, let t denote $n/\alpha(G)$, i.e. $\alpha(G) = n/t$. Consider the subgraph H induced by vertices of degree at most $2t\overline{d}$. Then, $\Delta(H) \le 2t\overline{d}(G)$. At least $t\overline{d}(n - |V(H)|)$ edges are removed, while G contained only $\frac{1}{2}\overline{d}n$ edges. Hence, at most $\frac{1}{2t}n$ vertices are removed, and thus $\alpha(H) \ge \alpha(G)/2$.

Apply Propositions 4.1 and 4.2 on H to obtain a subgraph $K \subset H \subseteq G$ with $\vartheta(\overline{K}) \leq k = 4t$ and at least n/k vertices, We then obtain an independent set in K with at least

$$\Omega(\frac{|V(H)|/k}{\Delta(H)^{1-1/2k}}) = \Omega(\frac{n/t}{(2t\overline{d}(G))^{1-1/8t}}) = \Omega(\frac{n/t^2}{\overline{d}(G)^{1-1/8t}}$$

vertices, for a performance ratio of $O(\overline{d}(G)^{1-1/8t}t^2)$. Recall that a minimum-degree greedy algorithm attains the Turán bound of $n/(\overline{d}+1)$ [6] for a $O(\overline{d}/t)$ approximation [16]. The two functions cross when t is about $\frac{1}{24}\log\overline{d}/\log\log\overline{d}$, for the desired ratio. □

References

[1] T. Akutsu and M. M. Halldórsson. On the approximation of largest common point sets and largest common subtrees. *Theoretical Comput. Sci.*, 233:33–50, Dec. 1999.

[2] N. Alon and N. Kahale. Approximating the independence number via the θ function. *Math. Programming*, 80:253–264, 1998.

[3] P. Berman and T. Fujito. On the approximation properties of independent set problem in degree 3 graphs. In *Proc. Fourth Workshop on Algorithms and Data Structures*, pages 449–460. Springer LNCS #955, Aug. 1995.

[4] R. B. Boppana and M. M. Halldórsson. Approximating maximum independent sets by excluding subgraphs. *BIT*, 32(2):180–196, June 1992.

[5] P. Crescenzi and V. Kann. A compendium of NP optimization problems. Dynamic online survey at `http://www.nada.kth.se/theory/problemlist.html`, 1999.

[6] P. Erdős. On the graph theorem of Turán (in Hungarian). *Mat. Lapok*, 21:249–251, 1970.

[7] P. Erdős and G. Szekeres. A combinatorial problem in geometry. *Compositio Math.*, 2:463–470, 1935.

[8] U. Feige. Randomized graph products, chromatic numbers, and the Lovász ϑ-function. *Combinatorica*, 17(1):79–90, 1997.

[9] U. Feige and J. Kilian. Heuristics for finding large independent sets, with applications to coloring semi-random graphs. In *Proc. 39th Ann. IEEE Symp. on Found. of Comp. Sci.*, 1998.

[10] R. Gonen and D. Lehmann. Optimal solutions for multi-unit combinatorial auctions: Branch and bound heuristics. Unpublished manuscript, May 2000.

[11] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization.* Springer-Verlag, 1988.

[12] M. M. Halldórsson. Approximations of weighted independent set and hereditary subset problems. In *Computing and Combinatorics, Proceedings of the 5th Annual International Conference (COCOON)*, volume 1627 of *Springer Lecture Notes in Computer Science*, pages 261–270, Tokyo, Japan, July 1998.

[13] M. M. Halldórsson. A survey on independent set approximations. In *Proc. 1st Intl. Wkshop on Approxim. Algor. (APPROX)*, volume 1444 of *Springer Lecture Notes in Computer Science*, pages 1–14, Aalborg, Denmark, July 1998.

[14] M. M. Halldórsson, J. Kratochvíl, and J. A. Telle. Independent sets with domination constraints. *Disc. Appl. Math.*, 99:39–54, Dec. 1999. URL: http://www.elsevier.nl/locate/jnlnr/05267.

[15] M. M. Halldórsson and H. C. Lau. Low-degree graph partitioning via local search with applications to constraint satisfaction, max cut, and 3-coloring. *Journal of Graph Algorithms and Applications*, 1(3):1–13, 1997.

[16] M. M. Halldórsson and J. Radhakrishnan. Greed is good: Approximating independent sets in sparse and bounded-degree graphs. *Algorithmica*, 18:145–163, 1997.

[17] M. M. Halldórsson and J. Radhakrishnan. Improved approximations of independent sets in bounded-degree graphs via subgraph removal. *Nordic J. Computing*, 1(4):275–292, Winter 1994.

[18] E. Halperin. Improved approximation algorithms for the vertex cover problem in graphs and hypergraphs. In *Proc. Eleventh ACM-SIAM Symp. on Discrete Algorithms*, pages 329–337, 2000.

[19] J. Håstad. Clique is hard to approximate within $n^{1-\epsilon}$. *Acta Mathematica*, 182:105–142, 1999.

[20] D. S. Hochbaum. Efficient bounds for the stable set, vertex cover, and set packing problems. *Disc. Applied Math.*, 6:243–254, 1983.

[21] T. Hofmeister and H. Lefmann. Approximating maximum independent sets in uniform hypergraphs. In *Proc. 23rd Intl. Symp. Math. Found. of Comp. Sci. (MFCS)*, volume 1450 of *Springer Lecture Notes in Computer Science*, pages 562–570, Brno, Czech Republic, Aug. 1998.

[22] D. Karger, R. Motwani, and M. Sudan. Approximate graph coloring by semi-definite programming. *J. ACM*, 45(2):246–265, Mar. 1998.

[23] D. Lehmann. Personal communication, May 1999.

[24] L. Lovász. On the Shannon capacity of a graph. *IEEE Trans. Inform. Theory*, IT-25(1):1–7, Jan. 1979.

[25] C. Lund and M. Yannakakis. The approximation of maximum subgraph problems. In *Proceedings of the 20th International Conference on Automata, Languages, and Programming (ICALP)*, Springer Lecture Notes in Computer Science, 1993.

[26] S. Mahajan and H. Ramesh. Derandomizing semidefinite programming based approximation algorithms. *SIAM J. Comput.*, 28(5):1641–1663, 1999.

[27] T. Sandholm. An algorithm for optimal winner determination in combinatorial auctions. In *Proc. International Joint Conference on Artificial Intelligence (IJCAI)*, Stockholm, Sweden, 1999.

[28] T. Sandholm, K. Larson, M. Andersson, O. Shehory, and F. Tohmé. Coalition structure generation with worst case guarantees. *Artificial Intelligence*, 111(1-2):209–238, 1999.

[29] E. R. Scheinerman. Generalized chromatic numbers of random graphs. *SIAM J. Disc. Math.*, 5(1):74–80, Feb. 1992.

[30] R. R. Vemuganti. Applications of set covering, set packing and set partitioning models: A survey. In D.-Z. Du and P. M. P. (Eds.), editors, *Handbook of Combinatorial Optimization*, volume 1, pages 573–746. Kluwer Academic Publishers, 1998.

[31] S. Vishwanathan. Personal communication, 1996.