Journal of Graph Algorithms and Applications
http://www.cs.brown.edu/publications/jgaa/
vol. 1, no. 3, pp. 1–13 (1997)

Low-degree Graph Partitioning via Local Search with Applications to Constraint Satisfaction, Max Cut, and Coloring

Magnús M. Halldórsson

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

Hoong Chuin Lau

Information Technology Institute
11 Science Park Road
Singapore 117685
hclau@iti.gov.sg

Abstract

We present practical algorithms for constructing partitions of graphs into a fixed number of vertex-disjoint subgraphs that satisfy particular degree constraints. We use this in particular to find k-cuts of graphs of maximum degree Δ that cut at least a $\frac{k-1}{k}(1 + \frac{1}{2\Delta+k-1})$ fraction of the edges, improving previous bounds known. The partitions also apply to constraint networks, for which we give a tight analysis of natural local search heuristics for the maximum constraint satisfaction problem.

These partitions also imply efficient approximations for several problems on weighted bounded-degree graphs. In particular, we improve the best performance ratio for the weighted independent set problem to $\frac{3}{\Delta+2}$, and obtain an efficient algorithm for coloring 3-colorable graphs with at most $\frac{3\Delta+2}{4}$ colors.

Communicated by M. Fürer: submitted February 1996; revised March 1997.

1 Introduction

Graph partitioning is a common theme in combinatorial optimization. We consider in this paper partitions of the vertices into a fixed number of induced subgraphs so that the *indegrees* of the vertices, or their degrees within their assigned subgraph, be within pre-specified upper bounds. The simplest objective is to limit the maximum indegree, while more generally we have bounds for each vertex which depend on the degree of that vertex. Since it is NP-complete in general to determine if a partition exists, we seek instead weak restrictions on the indegree upper bounds that guarantee existence.

An immediate area of application is the Max k-Cut problem, which is to partition the vertices of a graph into k parts so as to maximize the number of the edges going between subgraphs. This is the dual problem of minimizing the average indegree.

Edge cutting can be generalized beyond pure graphs to constraint systems: each edge is a binary relation whose satisfaction depends on the assignment of the incident vertices. The maximum constraint satisfaction problem, Max-Csp, is to find a k-partition of the vertices that maximizes the number of satisfied edges. Constraint satisfaction is a recurring theme in Artificial Intelligence with a variety of applications, e.g. in machine vision, temporal reasoning and scheduling. It generalizes other important combinatorial problems including Satisfiability. The Max-Csp problem naturally involves a parameter known as *consistency*: an instance is r-consistent, if for any constraint and any value of one incident vertex, there are r choices for the other vertex that satisfy the constraint. Note that Max k-Cut is a special case of $(k-1)$-consistent Max-Csp.

Our treatment of Max k-Cut and Max-Csp is characterized by two attributes. First, we seek to analyze the performance of simple and natural local search algorithms, which was our initial motivation in the current study. Second, the analysis focuses on the *absolute ratio* of the algorithms, which is the fraction of the edges (or constraints) that are cut (or satisfied). This is contrasted with the *relative ratio*, better known as the *performance ratio*, which is the ratio of the number of edges satisfied by the algorithm to the size of the optimal solution.

When the maximum indegree is fixed while the number of subgraphs is allowed to vary, we obtain a form of a coloring problem. In the standard Graph Coloring problem, indegree is fixed to be zero. The partitions we obtain also apply to these coloring problems.

Our results After preliminary definitions in Section 2, we consider plain local search for Max-Csp in Section 3. We show that it produces a k-partition of r-consistent instances such that the number of satisfied constraints incident on a vertex v of degree $d(v)$ is at least $\lceil \frac{r \cdot d(v)}{k} \rceil$. This gives an absolute ratio of $\frac{r}{k}$, which is tight. Slightly better bounds hold for special cases.

We next give, in Section 4, a method, also based on local search, that produces tighter partitions. It obtains a partition of a graph where, for a nonconstraint graph, each vertex v assigned to the t-th subgraph is of indegree at most $\lfloor \frac{d(v)+t}{k} - 1 \rfloor$. We use this to obtain an absolute ratio of $\frac{k-1}{k}(1+\frac{1}{2\Delta+k-1})$ for Max k-Cut, improving the best previous known ratio of $\frac{k-1}{k}(1+\frac{1}{n})$. This can be implemented in $O(\Delta n)$ time. We, however, find that this approach cannot improve the ratio for Max-Csp.

Finally, we derive, in Section 5, improved performance ratios for Graph Coloring and a family of induced subgraph problems in weighted bounded-degree graphs. These use a simplification of our partitioning that reduces to a result of Lovász, where the bounds for all the vertices are identical. We obtain a performance ratio of $\frac{3}{\Delta+2}$ for the weighted independent set problem, and $1/\lceil(\Delta+1)/3\rceil$ for all hereditary induced subgraph problems on weighted graphs in linear time. Also, we show how to color 4-clique-free graphs, which include 3-colorable graphs, using at most $\frac{3\Delta+2}{4}$ colors in linear time.

Previous results A number of approximation results are known for Max k-Cut. Let n and m denote the number of vertices and edges in the input graph, respectively. For $k = 2$, there are absolute ratios of $\frac{1}{2} + \frac{1}{2n}$ [10], $\frac{1}{2} + \frac{1}{2(n-1)}$ [12], and $\frac{1}{2} + \frac{n-1}{4m}$ [20], while the relative ratio has recently been improved to about 0.878 by Goemans and Williamson [9]. For $k > 2$, the best absolute ratio is $\frac{k-1}{k}(1 + \frac{1}{n})$ [22], while Frieze and Jerrum [7] generalized the results of [9], achieving a relative ratio of $\frac{k-1}{k} + \Theta(\frac{\ln k}{k^2})$.

For Max-Csp, the only published absolute ratio we are aware of is the $\frac{1}{2}+\frac{n-1}{4m}$ ratio of Poljak and Turzík [20] for domain size $k = 2$ and consistency $r = 1$. An absolute ratio of $\frac{r}{k}$ can be observed for a *greedy* algorithm, which would also be the derandomized version of the randomized schema similar to that used by Yannakakis [23] and Goemans and Williamson [8] used to approximate maximum satisfiability.

As for relative ratios, Khanna *et al.* [14] considered weighted Max-Csp with domain size 2. They achieved a ratio of $\frac{1}{4}$ via a sophisticated local search algorithm. More recently, this ratio was improved to $\frac{1}{2}$ by Trevisan [21] using randomized rounding of linear programs. The current best ratio is 0.859, due to Feige and Goemans [6] via randomized rounding of semidefinite programs. Lau and Watanabe [16] have proved a ratio of 0.408 for Max-Csp with domain size 3, and in general a ratio of $\frac{1}{k}$ for domain k.

2 Preliminaries

Let G denote an unweighted, undirected, not necessarily simple, graph, with vertex set V and edge set E. Let n denote the number of vertices, m the number of edges, and $d(v)$ the degree of vertex v. Let k be a positive integer.

Graph partitioning The *maximum k-cut problem* (Max k-Cut) is defined as follows:

> **INSTANCE**: Graph $G = (V, E)$, and positive integer k.
> **SOLUTION**: A partition of V into k subsets such that the number of edges across subsets is maximized.

Constraint Satisfaction The *domain* of a vertex v is the set of assignable values for v. For our purpose, these domains are all $\{1, \ldots, k\}$. An *assignment* is a mapping of vertices to values in their respective domains. A *constraint* $R(u, v)$ between vertices u and v, is a binary relation on $\{1, \ldots, k\} \times \{1, \ldots, k\}$ which defines the pairs of values that can be assigned to u and v simultaneously. A constraint $R(u, v)$ is said to be *satisfied* by an assignment f iff $(f(u), f(v))$ is in $R(u, v)$.

The maximum constraint satisfaction problem (Max-Csp) is defined as follows:

> **INSTANCE**: Graph $G = (V, E)$, with constraint relation $R(u, v)$ associated with each edge $(u, v) \in E$, and positive integer k.
> **SOLUTION**: An assignment $f : V \to \{1, \ldots, k\}$ such that the number of satisfied constraints is maximized.

An assignment can be viewed as a partition of vertices into k classes, or subsets. We shall use "edge" and "constraint" interchangably.

Consistency A constraint $R(u, v)$ is said to be *r-consistent* $(1 \leq r \leq k)$ iff, for every value x, $1 \leq x \leq k$, there exist at least r consistent values y such that $(x, y) \in R(u, v)$, and vice versa. A Max-Csp instance is r-consistent iff all its constraints are r-consistent. Let Max-Csp(k, r) represent the class of Max-Csp instances which have domain size k and are r-consistent. Observe that Max k-Cut is equivalent to Max-Csp$(k, k-1)$ with all constraints being the "not-equal" constraint $((x, y) \in R(u, v)$ iff $x \neq y)$.

Approximation Let **A** be an algorithm for a maximization problem. We say that *A approximates the problem within a relative ratio ρ* $(0 < \rho \leq 1)$ iff on all instances, A returns a solution whose value is at least ρ times the value of the optimal solution in time polynomial in the size of the input. We say that **A** *approximates the problem within an absolute ratio ϵ* iff it returns a solution whose value is at least ϵ times the largest possible solution on each given input. Clearly, an absolute ratio implies a no smaller relative ratio but not vice versa.

Satisfiable instances A Max-Csp instance $G = (V, E)$ is said to be *satisfiable* if all its constraints can be simultaneously satisfied. It is known that ratios that hold for the class of 1-consistent instances also hold for the class of satisfiable instances.

Observation 1 *If 1-consistent* Max-Csp *can be approximated within absolute ratio ρ, then satisfiable* Max-Csp *can be approximated within absolute ratio ρ.*

Proof. Let G be an instance of satisfiable Max-Csp of m constraints. Then we can construct a 1-consistent CSP instance $G' = (V, E')$ in time $O(nk^2)$ [5] such that: (1) each vertex in V has domain size at most k; (2) $E' \subseteq E$ (though the underlying relations may not be the same); and (3) for any assignment f, a constraint in E' is satisfied iff it is satisfied in E, while all constraints in $E - E'$ are always satisfied.

Suppose f is an assignment which satisfies at least $\epsilon m'$ constraints in G'. The same f then satisfies at least $\epsilon m' + (m - m') \geq \epsilon m$ constraints in G. ∎

3 Simple Local Search

In this section, we consider the performance of a simple local search procedure for Max-Csp.

The objective value of an assignment is the number of constraints that it satisfies. Two assignments f and f' are said to be *neighbors* if their values differ on exactly one vertex. An assignment is a *local optimum* iff its objective value is at least that of all its neighbors. Let LS be the following simple local search, or hillclimbing, procedure:

```
Start with any arbitrary initial assignment $f$.
while (there is a neighbor $f'$ of $f$ with a higher objective value) do
    $f \leftarrow f'$
output $f$
```

The following lemma is the key to our analysis:

Lemma 2 *Let f be any locally optimal solution computed by* LS *from an instance of* Max-Csp*(k, r). Then, for all vertices v, the number of constraints incident on v that are satisfied by f is at least $\lceil \frac{r \cdot d(v)}{k} \rceil$.*

Proof. Let v be a vertex, and evaluate the number of satisfied constraints as we examine all k possible values for v. Only the constraints incident on v are affected, while consistency ensures that each of them is satisfied at least r times, independent of the value of the other incident vertices. Thus, the locally optimal value for v must satisfy at least $\lceil \frac{rd(v)}{k} \rceil$ of the incident constraints. ∎

We obtain a bound on the performance of simple local search.

Theorem 3 LS *approximates* Max-Csp*(k, r) within an absolute ratio of $\frac{r}{k}$.*

Proof. Termination of the search is guaranteed by the fact that the objective function is monotone increasing with a maximum of m. Then, summing up over all the vertices, at least $\frac{1}{2} \sum_v \frac{rd(v)}{k} = \frac{rm}{k}$ constraints are satisfied, by Lemma 2. ∎

This bound is tight in that there are instances, even satisfiable ones, where the heuristic satisfies no more than $\frac{r}{k}$ constraints. We present these in Section 4.1. This result can, however, be slightly strengthened when the degrees of the vertices satisfy a certain congruence relation.

Corollary 4 *Suppose there is an s, $1 \leq s \leq k-1$, such that, for each constraint, the residue $(k-r)d(v) \bmod k$ is at least s. Then,* LS *satisfies at least $m\frac{r}{k} + n\frac{s}{2k}$ constraints.*

Proof. The number of satisfied constraints incident on a given vertex v is at least:

$$\lceil \frac{rd(v)}{k} \rceil \;=\; d(v) - \lceil \frac{k-r}{k} d(v) \rceil + 1 \;=\; \frac{rd(v)+s}{k}.$$

Summing over all vertices, the total number of satisfied constraints is at least: $\frac{1}{2}\sum_v (d(v)\frac{r}{k} + \frac{s}{k}) \;=\; m\frac{r}{k} + n\frac{s}{2k}$. ∎

4 Modified Local Search

We present in this section a lemma on low-degree partitioning of graphs, and its application to improved approximations of Max k-Cut. The bounds on the degrees in the resulting partition are specified in terms of a matrix, which we define as follows.

Definition 1 *Let $G = (V, E)$ be a graph on n vertices, k be an integer, and A be a $n \times k$ integer matrix. A is a* degree partitioning matrix (DPM) *of G if,*

$$\sum_{j=1}^{k} A[v,j] \geq d(v) - k + 1, \qquad \textit{for each vertex } v \in V.$$

A DPM suggests a partition f where, for each vertex v, the number $indeg(v)$ of neighbors within its subgraph is bounded from above by the corresponding entry $A[v, f(v)]$. We shall argue the existence of such a partition, as a corollary of the termination of a simple local improvement algorithm.

Given a k-partition f, let $deg_j(v)$ denote the number of vertices in subset j that are adjacent to v. Namely, $deg_j(v) = |\{w : (v,w) \in E \text{ and } f(w) = j\}|$. Let $indeg_f(v)$ be a shorthand for $deg_{f(v)}(v)$. We omit f when implicit.

We consider a local search algorithm that evaluates the following local condition:

$$A[v,j] - deg_j(v) \text{ has a maximum at } j = f(v). \tag{1}$$

The local search rule is simply to change the assignment of a vertex to one satisfying (1). The termination condition is that the local condition be satisfied at each vertex.

Observe that the local condition can always be fulfilled with a non-negative value $A[v,j] - deg_j(v)$, given the defining property of the DPM.

We find that the local search always terminates, and does so relatively quickly, especially given a starting assignment of rudimentary quality.

Lemma 5 *Consider a graph G, an associated DPM A, and a starting assignment f. Then, at most*

$$\sum_{v \in G} \left[(\max_j A[v,j]) - A[v, f(v)] + \frac{1}{2} indeg_f(v) \right] \tag{2}$$

iterations are performed until a locally optimal assignment is obtained under condition (1).

Before we give the proof, we note that this shows that in most cases the number of iterations is $O(m)$. This holds for instance when given a greedy initial assignment or the trivial assignment that maximizes the A-value for each vertex independently. It also holds when the improvements are scheduled so that all vertices are tested for improvement in the first n iterations, e.g. by using a worklist approach. Finally, one would expect in all applications that all entries $A[v, j]$ would be bounded by the degree $d(v)$ of the vertex. In all of these cases, the number of iterations is $O(m)$. It, however, appears plausible that there exist an initial assignment and a sequence of improvements whose number is asymptotically greater than m.

Proof. Consider the potential function $\Psi(f) = \sum_v [2A[v, f(v)] - indeg(v)]$, which measures the progress towards a locally optimal solution. In an iteration of the algorithm, a single vertex v is moved from partition i to partition j. This changes the potential only for the part contributed by v on one hand, and by the neighbors of v in subsets i and j on the other hand. The resulting change in potential is

$$\begin{aligned}\Delta\Psi &= (2A[v, j] - deg_j(v)) - (2A[v, i] - deg_i(v)) + (deg_i(v) - deg_j(v)) \\ &= 2\,[(A[v, j] - deg_j(v)) - (A[v, i] - deg_i(v))]\end{aligned}$$

The local search rule ensures that $A[v, j] - deg_j(v)$ is strictly greater than $A[v, i] - deg_i(v)$ when v is moved from partition i to j. Hence $\Delta\Psi \geq 2$.

The difference in the potential of the final, locally optimal solution f' and the initial solution f is

$$\sum_v [2(A[v, f'(v)] - A[v, f(v)]) + indeg_f(v) - indeg_{f'}(v)].$$

Hence, half this number of iterations suffices. ∎

We now use this partitioning to approximate Max k-Cut. For this, we need a slightly stronger local improvement search.

Theorem 6 Max k-Cut *can be approximated within an absolute ratio of* $\frac{k-1}{k}(1+\frac{1}{2\Delta+k-1})$.

Proof. The k-partition is obtained in two steps. First, we find a partition that is locally optimal w.r.t. (1). We then apply standard local search, optimizing the number of cut edges, until local optima is achieved.

For the former, we use an evenly split DPM, with $A[v, t] = \lfloor \frac{d(v)+t}{k} \rfloor - 1$, for each vertex v and each subgraph t. For such a balanced DPM, the application of standard local search preserves optimality w.r.t. (1). Namely, if $deg_j(v) < deg_i(v)$, then $A[v, j] - deg_j(v) \geq A[v, i] - deg_i(v)$, since $A[v, i]$ and $A[v, j]$ differ by at most one. Hence, we obtain an assignment f that is locally optimal under both criterias.

We now focus on bounding the quality of the resulting assignment. Local optimality w.r.t. (1) ensures that

$$indeg(v) \;\leq\; \lfloor \frac{d(v)+f(v)}{k} \rfloor - 1. \tag{3}$$

Let Unsat denote the number of edges that are not satisfied, i.e. have both endpoints within the same subgraph. Let V_i denote the set of vertices in the i-th subgraph.

Applying (3) and simplifying by ignoring the floor, we obtain

$$\text{Unsat} \;=\; \frac{1}{2}\sum_t \sum_{v \in V_t} indeg(v) \;\leq\; \frac{m}{k} - \frac{n}{2} + \frac{1}{2k}\sum_{t=1}^{k} t \cdot |V_t|. \tag{4}$$

Replace the last term of the sum using $|V_k| = n - \sum_{t=1}^{k-1} |V_t|$, to get

$$\text{Unsat} \;\leq\; \frac{m}{k} - \frac{1}{2k}\sum_{t=1}^{k-1} (k-t) \cdot |V_t|. \tag{5}$$

By standard local optimality, each vertex v in the graph is adjacent to at least $indeg(v)$ vertices in V_t, while each vertex in V_t can contribute to at most Δ of these adjacencies. Hence, the number of vertices in V_t is bounded by

$$|V_t| \;\geq\; \sum_{v \in G} \frac{indeg(v)}{\Delta} \;=\; 2\,\frac{\text{Unsat}}{\Delta}, \qquad t = 1, \ldots, k-1.$$

Plug this into (5) to obtain

$$\text{Unsat} \;\leq\; \frac{m}{k} - \frac{(k-1)\text{Unsat}}{2\Delta}. \tag{6}$$

Thus,

$$\text{Unsat} \;\leq\; \frac{m}{k}/(1 + \frac{k-1}{2\Delta}) \;=\; \frac{m}{k}(1 - \frac{k-1}{2\Delta + k - 1}).$$

Hence, at least

$$m\frac{k-1}{k}(1 + \frac{1}{2\Delta + k - 1})$$

edges are satisfied, yielding the theorem. ∎

Time complexity In the case of a balanced DPM as above, the difference in the initial and final potential (as in Lemma 5) is at most n plus the difference in the number of satisfied edges. By using a Greedy initial assignment, the initial number of unsatisfied edges is at most $\frac{m}{k}$. Thus, $n + \frac{m}{k}$ iterations suffice.

The local search algorithms can be implemented to require only amortized $O(\Delta)$ time per iteration. We pre-compute the degrees of the vertices into each subgraph in the starting partition, and then only $O(\Delta)$ updates are needed in each iteration. Hence, the total complexity is at most $O(\Delta \frac{m}{k})$.

If we are content with the performance bounds, and do not seek local optima, the local search can be terminated prematurely when sufficiently many edges are satisfied. Starting with a greedy assignment, we only need to satisfy $\frac{m}{2\Delta} < n$ additional edges. From the potential function, we see that this is accomplished in at most $2n$ iterations, or in time $O(\Delta n)$.

4.1 Extension to Constraint Satisfaction

The concept of a degree partitioning matrix and the corresponding partition can also be extended to constraint systems. A matrix A is said to be a DPM of an r-consistent constraint graph G if,

$$\sum_{i=1}^{k} A[v,j] \geq (k-r)d(v) - k + 1, \qquad \text{for each vertex } v.$$

Define $deg_j(v)$ to be the number of constraints incident on v that are not satisfied if the assignment of v is changed to j (assuming a given assignment f). As before, $indeg(v) = deg_{f(v)}(v)$. The algorithm and its proofs of optimality and time complexity remain the same.

It is tempting to try to prove a similar result for Max-Csp(k,r). Unfortunately, this is not possible, since instances can be constructed where a locally optimal solution satisfies at most an $\frac{r}{k}$ fraction of the constraints. In fact, even the relative ratio of the algorithm is tight.

Theorem 7 *Local search under (1) does not approximate* Max-Csp *within better than $\frac{r}{k}$, neither absolute nor relative. Namely, there is an infinite sequence of satisfiable instances where some locally optimal solutions satisfy only an $\frac{r}{k}$ fraction of the constraints.*

Proof. Given k and r, we construct the following constraint graph G. G contains $3k$ vertices $v_{i,j}$, for $i = 0, \ldots, k-1$ and $j = 0, 1, 2$, each of degree $2k$. There are constraints between $v_{i,j}$ and $v_{i',j'}$ whenever $j \neq j'$, given by:

$$R(v_{i,j}, v_{i',j+1 \bmod 3}) \;=\; \{\,(x,y) : [(i'-i)+(y-x)] \bmod k \text{ lies between } 0 \text{ and } r-1\,\}.$$

Notice that the constraints are not symmetric.

An optimal solution assigns each vertex $v_{i,j}$ to subset i, yielding a totally satisfied solution. Suppose we have an initial assignment where all vertices are assigned to subset $k-1$. Then each vertex $v_{i,j}$ is consistent with $2r$ of the adjacent vertices; i.e. r vertices of the form $v_{i',j+1 \bmod 3}$ and r of the form $v_{i',j-1 \bmod 3}$. Hence, $indeg(v_{i,j}) \leq 2k-2 = A[v_{i,j}, k-1]$. Furthermore, one can verify that moving a single vertex to a different subgraph leaves the number of incident satisfied constraints unchanged. Hence, we have a local optima with an absolute and relative ratio of $\frac{r}{k}$. Note that this proof can be extended to graphs of any number of layers greater than 2. ∎

It is also easy to construct multigraphs where at most an $\frac{r}{k}$ fraction of the constraints can be simultaneously satisfied. In the case of simple constraint graphs, however, it appears that a slightly larger fraction is satisfiable, while it does not exceed the $\frac{r}{k}(1 + \Theta(\frac{1}{\Delta}))$ bound of Max k-Cut.

5 Applications to induced subgraph problems and graph coloring

An important special case of the graph partitioning in the previous section is when we only seek to bound the maximum indegree from above. In this case, Lemma 5 reduces to the following result of Lovász.

Lemma 8 (Lovász [18]) *Let $G = (V, E)$ be a multigraph without self loops. Let $t_1, t_2, \ldots, t_k$ be non-negative integers such that $\sum_i (t_i + 1) - 1 = \Delta(G)$. Then, V can be partitioned into k subsets inducing subgraphs $G_1, G_2, \ldots G_k$ such that $\Delta(G_i) \leq t_i$, for $i = 1, 2, \ldots k$.*

By (2), we achieve this in $O(\frac{m}{k})$ iterations, or $O(\Delta \frac{m}{k})$ time.

This lemma has several elegant applications to the approximation of induced subgraph and vertex partitioning problems. A property of graphs is said to be *hereditary* if, whenever it holds for a graph it also holds for any induced subgraph.

Theorem 9 *Hereditary weighted induced subgraph problems can be approximated within relative ratio of* $1/\lceil (\Delta + 1)/3 \rceil$ *in linear time.*

Proof. We partition the graph into at most $s = \lceil \frac{\Delta+1}{3} \rceil$ graphs of degree at most 2, in linear time, using Lemma 8. Such graphs consist of disjoint paths and cycles, and allow for a linear time solution of hereditary induced subgraph problems via dynamic programming. Any property π holds either for every path, or for all paths of length up to q, where q is a fixed constant, and the same dichotomy holds for cycles.

Our approximate solution will be the largest of the (exact) solutions from these s subgraphs. The optimal solution of the whole graph can contain at most as many vertices from each subgraph as the optimal solution on that subgraph. Hence, the optimal solution is at most s times as large as the approximate solution. ∎

One such problem is Max Compatible Constraint Satisfaction [4, MS11], This is problem on a CSP instance, where the objective is to find an assignment to a subset of the vertices that satisfies all the induced constraints.

Corollary 10 Max Compatible Constraint Satisfaction *can be approximated within a relative ratio of* $1/\lceil \frac{\Delta+1}{3} \rceil$ *in linear time.*

The previous best approximation for the weighted independent set problem is $\frac{2}{\Delta}$ due to Hochbaum [11]. We can use her approximation for $\Delta = 3$ to improve our ratio when Δ is a multiple of 3.

Proposition 11 *The weighted independent set problem can be approximated within a relative ratio of $\frac{3}{\Delta+2}$. When $\Delta \bmod 3 = 2$, the ratio is $\frac{3}{\Delta+1}$, while when $\Delta \bmod 3 = 0$, the ratio is $\frac{3}{\Delta+3/2}$.*

Proof. When $\Delta \bmod 3$ is congruent to 1 or 2, the claim follows from Theorem 9. When $\Delta \bmod 3 = 0$, partition the vertices into $\frac{\Delta}{3}$ classes, where all but possibly the last have maximum degree 2. Find a $\frac{2}{3}$-approximate weighted independent set in the last class, compute optimal solutions of the other classes, and let output the largest of all of these. Let W denote the weight of our solution.

The weight of the optimal solution is at most the sum of the weights of the optimal solutions on the $\frac{\Delta}{3}$ subgraphs. The weight of the optimal solution of the last subgraph is at most $\frac{3}{2}W$, but at most W for the other subgraph. Hence, the weight of the optimal solution is at most $[\frac{3}{2} + (\frac{\Delta}{3} - 1)]W = [\frac{\Delta}{3} + \frac{1}{2}]W$. ∎

By applying a preprocessing method championed by Hochbaum [11], we can obtain improved approximations of weighted vertex cover, for $\Delta \geq 5$. The relative approximation of minimization problems is defined identically to that of maximization problems, except that the ratio is necessarily greater than 1.

Corollary 12 *The weighted vertex cover problem can be approximated within $2 - \frac{3}{\Delta+2}$ in time $O(\Delta n^{3/2})$.*

5.1 Coloring

Lovász's lemma also has implications for the coloring of bounded-degree graphs, as observed previously by Catlin [3], Borodin and Kostochka [2] and Lawrence [17]. The constructive nature of the lemma has apparently not been made explicit before. Our implementation yields an efficient coloring algorithm.

Proposition 13 *Graphs without 4-cliques can be colored with $\frac{3\Delta+2}{4}$ colors in linear time.*

Proof. Partition the input graph into subgraphs of degree 3 or 4 via Lemma 8, with $(\Delta + 2) \bmod 3$ subgraphs of degree 4 and the remaining ones of degree 3. Assuming the graph contains no clique on 4 vertices, each subgraph G_i can be colored with $\Delta(G_i)$ colors by the algorithm that follows from Lovász' constructive proof of Brooks' theorem [19]. ∎

This can be generalized to $\frac{(t-1)\Delta+(t-2)}{t}$ colors for graphs without t-cliques.

Karger, Motwani and Sudan [13] have recently obtained a $\Delta^{1-\Omega(1)} \log n$ approximation for 3-coloring. The advantage of our approach, however, is speed, simplicity, ease of implementation, and better bounds for all constant (or slightly superconstant) values of Δ.

Acknowledgement

We would like to thank Osamu Watanabe for his helpful comments, and the referees for suggestions for improved presentation. A portion of this paper appeared in preliminary form in [15].

References

[1] Paola Alimonti. New local search approximation techniques for maximum generalized satisfiability problems. *Inf. Proc. Let.*, 57(3):151–158, 12 Feb 1996.

[2] O. V. Borodin and A. V. Kostochka. On an upper bound of a graph's chromatic number, depending on the graph's degree and density. *J. Combin. Theory Ser. B*, 23:247–250, 1977.

[3] P. A. Catlin. A bound on the chromatic number of a graph. *Discrete Math.*, 22:81–83, 1978.

[4] P. Crescenzi and V. Kann. A compendium of NP optimization problems. Dynamic on-line survey available at `ftp.nada.kth.se`, also in `http://www.nada.kth.se/~viggo/problemlist/compendium.html`, Oct. 1995.

[5] Rina Dechter and Judea Pearl. Network-based heuristics for constraint satisfaction. *Artif. Intell.*, 34:1–38, 1988.

[6] Uriel Feige and Michel Goemans. Approximating the Value of Two-Prover Proof Systems, with Applications to MAX 2SAT and MAX DICUT. In *Proc. Israel Symp. on Theoretical Computer Science*, 182–189, 1995.

[7] Alan Frieze and Mark Jerrum. Improved approximation algorithms for MAX k-CUT and MAX BISECTION. *Algorithmica*, 18(1):67–81, May 1997.

[8] Michel X. Goemans and David P. Williamson. New 3/4 Approximation Algorithms for the Maximum Satisfiability Problem. *SIAM J. Disc. Math*, 7(4):656–666, 1994.

[9] Michel X. Goemans and David P. Williamson. Improved Approximation Algorithms for Maximum Cut and Satisfiability Problems Using Semidefinite Programming. *J. ACM*, 42:1115-1145, 1995.

[10] D. J. Haglin and S. M. Venkatesan. Approximation and intractability results for the maximum cut problem and its variants. *IEEE Transactions on Computers*, 40:110–113, 1991.

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

[12] Y. Kajitani, J. D. Cho, and M. Sarrafzadeh. New approximation results on graph matching and related problems. In *Graph-Theoretic Concepts in Computer Science, International Workshop (WG)*, pages 343–358. Springer Verlag LNCS (903), 1994.

[13] D. Karger, R. Motwani and M. Sudan. Approximate Graph Coloring by Semidefinite Programming. In *Proc. 35th IEEE Symp. on Found. of Comp. Sci.*, pages 2–13, 1994.

[14] S. Khanna, R. Motwani, M. Sudan, and U. Vazirani. On syntactic versus computational views of approximability. In *Proc. 35th IEEE Symp. on Found. of Comp. Sci.*, pages 819–830, 1994.

[15] H. C. Lau. Approximation of constraint satisfaction via local search. In *Proc. 4th Wksp. on Algorithms and Data Structures (WADS)*, pages 461–472. Springer LNCS (955), 1995.

[16] H. C. Lau and O. Watanabe. Randomized Approximations of the Constraint Satisfaction Problem. *Nordic J. Computing*, 3:405–424, 1996.

[17] J. Lawrence. Covering the vertex set of a graph with subgraphs of smaller degree. *Discrete Math.*, 21:61–68, 1978.

[18] L. Lovász. On decomposition of graphs. *Stud. Sci. Math. Hung.*, 1:237–238, 1966.

[19] L. Lovász. Three short proofs in graph theory. *J. Combin. Theory Ser. B*, 19:269–271, 1975.

[20] Svatopluk Poljak and Daniel Turzík. A polynomial time algorithm for constructing a large bipartite subgraph, with an application to a satisfiability problem. *Can. J. Math*, 34(3):519–524, 1982.

[21] Luca Trevisan. Positive Linear Programming, Parallel Approximation and PCPs. In *Proc. European Symp. on Algorithms (ESA)*. Springer LNCS (1136), 1996.

[22] P. M. Vitányi. How well can a graph be n-colored. *Discrete Math*, 34:69–80, 1981.

[23] M. Yannakakis. On the Approximation of Maximum Satisfiability. *J. Algo*, 17:475–502, 1994.