Chapter 18
Approximating Discrete Collections via Local Improvements

Magnús M. Halldórsson*

Abstract

We consider an old optimization technique and apply it to the approximation of collections of discrete items. The technique is local improvement search: attempt to extend a collection by adding some items while removing others.

We obtain a $(k+1)/2$ performance ratio for k-DM, k-SET PACKING, and INDEPENDENT SET in $k+1$-claw-free graphs by an $\mathcal{NC}$ algorithm or in nearly-linear sequential time. It can be extended to $k/2+\epsilon$ (when $k \geq 4$), and to $(k+1)/3+\epsilon$ when maximum degree is bounded.

We also obtain improved performance ratios for induced subgraph problems, INDEPENDENT SET in bounded-degree graphs, VERTEX COVER in claw-free graphs, SET COVER, and GRAPH COLORING under a complementary objective function,

1 Introduction

Local search is a frequently used tool to solve hard combinatorial optimization problems. It comes in many flavors including greedy search or hillclimbing, simulated annealing or randomized hillclimbing, augmenting paths, and local changes. However, in spite of considerable success on practical instances, there are few worst-case results of a positive nature.

A common belief is that local search is largely useless theoretically. This is formalized in the notion of $\mathcal{PLS}$-completeness [14], or roughly speaking, "as hard as the hardest local optimization problems". The list of $\mathcal{PLS}$-complete problems and neighborhoods pairs covers most of the popular local search heuristics: t-opt and Lin-Kernighan for the TRAVELING SALESMAN problem; Kernighan-Lin and Fiduccia-Matheyses for GRAPH PARTITIONING; and FLIP for MAX CUT. It is an open problem whether local search for $\mathcal{PLS}$-complete problems can converge on locally optimal solutions in worst-case polynomial time. Exponential worst-case time complexity of many common heuristics has been established. In some cases, much stronger hardness results can be shown: for given instances and starting solutions, there is *no* sequence of improvements ending with a locally optimal solution that is less than exponential length [28]. This is in the case of weighted problems; for unweighted $\mathcal{PLS}$-complete problems, obtaining local optimality is $\mathcal{P}$-complete. (See [26, 14, 28].)

What really brings home the point about the weakness of local search under worst-case criteria are the proven weaknesses of several popular heuristics with good average-case behavior on relatively easy problems. For instance, the Metropolis algorithm, the heart of Simulated Annealing, not only cannot find large cliques [12] (which is hard), but also fails badly for MAXIMUM MATCHING [25] (which is not). And for METRIC TSP, which is probably the most popular testbed for heuristics, there are instances where the popular t-opt heuristics obtain solutions that are $\Omega(n^{1/(2t)})$ longer than optimal [3].

The purpose of the current paper is to present several results of a positive nature on the quality of locally optimal solutions. The goal is two-fold: on one hand, to improve the best performance ratios known for various optimization problems; on the other hand, to shed some light on the effectiveness of local-improvement heuristics as approximation algorithms. We consider a host of problems whose common theme is that they involve collections of discrete items and their objective is to minimize or maximize the size of the collection, usually defined as the number of items. The approach used is the natural one: start with a maximal/minimal collection, and expand/shrink the collection until no better solution can be found by modifying only few of the items.

The *performance ratio* of an approximate algorithm is the worst-case ratio of the size of the optimal solution to the size of the approximate solution (for minimization problem, we use the inverse, so the ratio is always at least one).

We present the following improved performance ratios:

1. $k/2+\epsilon$ for INDEPENDENT SET problem in $k+1$-claw free graphs (when $k \geq 4$), and 5/3 (for $k=3$), by an $\mathcal{NC}$ algorithm. This also applies to k-SET PACKING, k-DIMENSIONAL MATCHING, H-

*Japan Advanced Institute of Science and Technology, Tatsunokuchi, Ishikawa 923-12, Japan. magnus@jaist.ac.jp

MATCHING ($k = |V(H)|$), and k-CLIQUE PACKING.

2. $(k+1)/2$ for k-SET PACKING when the objective is to maximize the sum of the sizes of the sets in the collection.
3. $(k+1)/3 + \epsilon$ for the above problem when the maximum degree is constant.
4. $(\Delta+2+1/3)/4+\epsilon$ for INDEPENDENT SET in graphs of maximum degree Δ.
5. Same bounds as above for a class of HEREDITARY INDUCED SUBGRAPH problems, with improved ratios for the INDUCED λ-COLORABLE SUBGRAPH and VERTEX ARBORICITY problems.
6. $2 - 2f_s(k)$ for VERTEX COVER in $k+1$-claw free graphs, $k \geq 6$, where $f_s(k) = (k \ln k - k + 1)/(k-1)^2 = 2 - (\log k)/k(1 + o(1))$. For $k = 4, 5$, a ratio of 1.5 holds.
7. $11/7 \approx 1.57$ for 3-SET COVER (SET COVER restricted to sets of size at most 3), which generalizes to a $\mathcal{H}_k - 11/42$ ratio for k-SET COVER.
8. 1.4 for GRAPH COLORING when the objective is to maximize $|V| - c$, where c is the number of colors used.

In all cases, the analysis is tight for the heuristics analyzed. These problems are all MAX SNP-hard [23, 1] and thus cannot be approximated within $1+\epsilon$ for all $\epsilon > 0$ (unless $\mathcal{P} = \mathcal{NP}$). For INDEPENDENT SET and related problems, approximation hardness grows polynomially with maximum degree or claw size [6].

1.1 Related work. The local optimization technique we study is not new. It is perhaps most familiar when placed in the setting of a polynomial solvable problem: MAXIMUM MATCHING. In this setting it reduces to the fundamental concept of finding small *augmenting paths.*

Many of the ideas in the analysis reported here come from the work of Berman and Fürer [2] which obtained greatly improved ratios for the INDEPENDENT SET problem in bounded-degree graphs. Halldórsson and Radhakrishnan [9] gave some variations on that theme.

As we have recently learned, Hurkens and Schrijver [11] have previously obtained the same ratios for k-SET PACKING for the local improvement heuristic. Our analysis is, however, of independent interest: we identify a restricted class of improvements that suffices to obtain these ratios and may lead to more efficient improvement search; we address complexity and other algorithmic issues; we obtain efficient parallel algorithms; and, our analysis, we believe, is relatively short and simple. Yu and Goldschmidt [29, 30] have also concurrently obtained similar results on this problem, and Khanna et al. [16] have obtained the first ratio in the sequence, or $(k+1)/2$.

One special class of claw-free graphs is that of *unit disk graphs.* These are graphs whose vertices can be mapped to equisized circles in the plane such that two vertices are adjacent iff the corresponding circles intersect. A circle can intersect at most 5 mutually non-intersecting circles, hence unit-disk graphs are 6-claw-free. Marathe et al. [20] give approximation algorithms for various problems on unit-disk graphs. We improve their ratio of 3 for INDEPENDENT SET in unit-disk graphs to $2.5+\epsilon$, and generalize their ratio of 1.5 for VERTEX COVER to arbitrary 6-claw-free graphs.

Organization. Section 2 introduces the problems under study and their relationships, describes the local improvement paradigm, and presents notation common to the paper. Section 3 covers packing problems, including SET PACKING, the several INDEPENDENT SET (sometimes called VERTEX PACKING) problems, and VERTEX COVER. Section 4 has results on induced subgraph problems (other than INDEPENDENT SET). Covering and partitioning problems are the topic of Section 5, or more precisely, SET COVER and GRAPH COLORING.

2 Discrete Collections and Local Improvements

We first define formally the problems under study, and illustrate the not-so-obvious relationships and reductions between the problems. We then describe the basic strategy for the local improvement search, and list the notation common to the various sections.

2.1 The problems and their relationships.

Let $G = (V, E)$ be an unweighted graph. A *set system* (or, *hypergraph*) (S, C) consists of a base set S and a collection C of subsets (or, *hyperedges*) of S. A *k-set system* is a set system where each set in C is of size at most k.

Consider the following collection of problems:

3-DM Given sets W, X, Y and a set $M \subseteq W \times X \times Y$, find a maximum cardinality matching, i.e. a subset $M' \subseteq M$ such that no two elements of M' agree in any coordinate.

k-Set Packing (k-SP) Given a k-set system (S, C), find a maximum cardinality collection of disjoint sets in C.

Independent Set Given a graph G, find a maximum cardinality subset of mutually non-adjacent vertices; i.e. a subset $V' \subseteq V$ such that $v_i, v_j \in V'$ implies $(v_i, v_j) \notin E$.

Vertex Cover Given a graph G, find a minimum cardinality subset $V' \subseteq V$ such that every edge has at least one endpoint in V'.

k-SET PACKING is a generalization of MAXIMUM MATCHING, from sets of size two (i.e. edges) to sets of size $1, 2, \ldots, k$. 3-DM is a generalization of a related problem, that of BIPARTITE MATCHING.

The *intersection graph* $H(S, C)$ of a hypergraph (S, C) has a vertex for each hyperedge with two hyperedges adjacent iff they intersect (as sets). An INDEPENDENT SET in $H(S, C)$ corresponds to a k-SET PACKING in (S, C). The intersection graph of a k-set system C contains no $k+1$-*claw*, i.e. no $k+1$-independent set in the neighborhood of any vertex.

The k-*independent set system* (k-IS system, for short) of a graph is the collection of all sets of up to k independent vertices in G.

Now consider the following covering problems:

k-Set Cover (k-SC) Given a k-set system (S, C), find a minimum *cover* of S, i.e. a subset $C' \subseteq C$ of minimum cardinality such that every element of S belongs to at least one member of C'.

Graph Coloring Given a graph G, find an assignment of minimum number of colors to the vertices such that adjacent vertices are of different colors.

An optimal SET COVER of the independent set system of a graph corresponds to an optimal GRAPH COLORING. In general, however, this may result in an exponential blowup in the size of the instance.

2.2 The Local Improvement Framework.

Consider the problem of maximizing a discrete collection satisfying a certain property, i.e. is *feasible*. A solution is *maximal* if the addition of any item outside the current solution destroys feasibility. A non-maximal solution can then be *extended* by merely adding an item.

Consider a feasible solution A. Suppose there is a set x, y, z where $x \in A$ and $y, z \notin A$ such that $A' = (A - \{x\}) \cup \{y, z\}$ is also a solution. Then, A' is an *extension* of A, and $\{x, y, z\}$ a 2-*improvement* of A. More generally, an *improvement* of A is a set of items I such that $A \oplus I$ (the symmetric union of A and I) is a solution and $|A \oplus I| > |A|$. A t-*improvement* adds s new items and removes at most $s-1$ items, for some $s \leq t$. Without loss of generality, an improvement adds one item more than it removes. A solution is said to be t-*locally-optimal* if it has no t-improvement.

Local search attempts to augment an initial solution by a succession of improvements until no improvement of small size can be found. This is summarized in the following schema.

t-opt($\mathcal{I}$)
  $A \leftarrow$ maximal solution($\mathcal{I}$)
  repeat
    $I \leftarrow s$-improvement of A, where $s \leq t$
    $A \leftarrow A \oplus I$
  until no further improvement exist
  return A

For minimization problems, we proceed in much the same fashion, but in opposite direction. We start with a minimal solution, and attempt to improve it by *shrinking* steps. A t-improvement of A is then a set I of t items in A and $t-1$ items outside of A such that $A \oplus I$ is a feasible solution.

2.3 Notation.

We shall be dealing with collections of items, where an item corresponds either to a set of a set system or to a vertex in a graph. Let C denote the collection of items (i.e. sets, vertices) of the input, and let n denote their number.

A pair of items may have conflicts, which correspond to sets intersecting or vertices being adjacent. Let m denote the number of pairwise conflicts in the input collection (or the number of edges in the conflict graph). The degree of an item is the number of conflicts involving that item; let Δ be the maximum degree of an item.

Let A denote a locally optimal collection, and B any other feasible collection (e.g. the optimal one). Let D denote the intersection between A and B, and let B_i, $i = 1, 2, 3$, be collections of the items in B conflicting with one, two, and three or more items in A, respectively. Let A_1 denote the items of A that conflict with items in B_1, and let $A_0 = A - A_1 - D$. Also, $B_0 = B - B_1 - D$.

The cardinality, or *size*, of a collection C is denoted by $|C|$. The sizes of the above mentioned collections are denoted by their respective lower case letter.

3 Packing problems

For packing problems, two items conflict iff the two sets intersect (in the set problems) or the two vertices are adjacent (in the graph problems). A collection is feasible if the items are mutually non-conflicting. This fully specifies t-OPT for these problems.

Let
$$\nu_s = \left(\sum_{i=0}^{s-1} (k-1)^i\right)^{-1} = \frac{k-2}{(k-1)^s - 1}.$$

THEOREM 3.1. *$(2s)$-locally-optimal solutions for* INDEPENDENT SET *in $k+1$-claw free-graphs and k-*SET PACKING *achieve performance ratios of $k/2 + \nu_s$ in time $O(\Delta^{2s-2} n^2)$, $k \geq 4$. When $k = 3$, the ratio is $5/3$ when $s = 2$.*

COROLLARY 3.1. *The above problems can be approximated within $k/2 + \epsilon$ in time $O(n^{\log_k 1/\epsilon})$, for $k \geq 4$.*

For graphs of bounded-degree, we can obtain stronger performance ratios via results of [2], albeit with significant increase in time complexity.

THEOREM 3.2. *The above problems can be approximated within a factor of* $(k+2)/3$ *in time* $n^{8\log\Delta+O(1)}$.

Weaker search. We obtain our main result, Theorem 3.1, via local improvement search that is less general than t-opt. Our analysis works by bounding the number of items of the optimal solution that conflict with only 1 or 2 items of our (locally optimal) solution. Thus, we only need to consider those items that conflict with at most 2 items in the current solution. We can represent this in the form of a graph.

Let A be the current solution. We construct the multigraph $H(A)$ whose vertices correspond to the items in A. An item outside of A conflicting with exactly two items in A forms an edge in $H(A)$ between the corresponding vertices. An item conflicting with exactly one item in A implies a self-loop on the corresponding vertex.

One form of an *improvement* of A consists of an induced subgraph in $H(A)$ with more edges than vertices, such that the items corresponding to those edges also do not conflict with each other. For the purposes of Theorem 3.1, we need only search for a highly restrictive type of an improvement. It corresponds to the subgraph, that we call a *t-ear*, which is a (non-simple) path of length t on at most $t-1$ vertices whose starting edge is a self-loop. If the corresponding items are non-conflicting, we have a *t-ear-improvement.*

Complexity. The number of iterations is trivially bounded by n, the maximum size of any solution. The main issue is the complexity of finding an improvement.

Let us first evaluate the complexity of finding a 2-ear improvement. Consider the collection of items that conflict with exactly one vertex in A. Any pair of non-conflicting items that conflict with the same vertex in A constitute an improvement. We inspect at most every pair of items for $O(n^2)$ time complexity; if the conflict graph is given, each edge is inspected at most once, for $O(n+m)$ complexity.

For set versions of the problem, an improvement can be found faster. For the 2-opt case for 3-SET PACKING, consider the sets adjacent only to the same set of A. Throw away the incident element of each of them to get a collection of 2-sets, in which we seek a disjoint pair. We traverse through the collection, comparing each set to a short list of at most three of the previous sets. The list contains either a star (up to three sets with one item in common) or a triangle (all three pairs from a three element set); any additional set is either disjoint from one of the stored sets, or contains the fixed element of the star. This process can thus be done with a simple scan through the list, in $O(n)$ time. Thus, an iteration of 2-opt can be found in time linear in the size of the input.

To search for a t-ear improvement, we inspect every simple path of length $t-2$ in H along with the possible self-loops on either end. The number of such paths is bounded by the sizes of *distance-t* neighborhoods in H, or Δ^{t-1}. The time complexity of an improvement search is therefore bounded by $O(m\Delta^{t-2})$. For set packing problems, this can be improved to $O(nk^k\Delta^{t-2})$.

3.1 Analysis. The analysis depends only on the local optimality of a solution A with respect to t-ear improvements. We count the number of conflicts between items in A and items in another hypothetical solution B. Every item outside A must conflict with (i.e. be adjacent to) some item of A since A is maximal. The core of the argument is that few items of B can conflict with only one element of A, unless many conflict with three or more (which would also be fine).

Degree bound. Each item in $A-D$ conflicts with at most k items in B. Summing up over all items yields:

$$b_1 + 2b_2 + 3b_3 \leq k(a-d). \tag{3.1}$$

2-opt. If there are two items in B adjacent only to the same single item of A, then a 2-ear improvement exists. Hence, the number of items in B adjacent to only one item in A is at most the number of items in A. More precisely,

$$b_1 = a_1. \tag{3.2}$$

We now add (3.2) and (3.1), along with the identity $d=d$, for:

$$2b \leq (k+1)a, \tag{3.3}$$

or, a performance ratio of $(k+1)/2$. This is the result of [29, 16].

2s-opt. We consider only the case $k \geq 4$. Consider the subgraph $H_B(A)$ of $H(A)$, whose edges correspond to the items of B conflicting with one or two items of A. Since any combination of edges in $H_B(A)$ corresponds to a conflict-free collection of items, any t-ear in $H_B(A)$ would correspond to an improvement of A.

Consider a breadth-first-search on $H_B(A)$ with $X_1 = A_1$ as a starting set. Let X_i be the set of vertices at level i of the BFS dag. Observe that each vertex in X_i is adjacent to a unique vertex in X_{i-1}, for $2 \leq i \leq s$. Otherwise, any two paths back to X_1 form a $2i$-ear. Furthermore, vertices at level i are not adjacent, for $2 \leq i \leq s-1$, for the same reason.

Now, count the non-self-loop edges in $H_B(A)$; this number corresponds to b_2.

$$b_2 \leq x_2 + x_3 + \ldots + x_s + ((k-1)/2)x_s \tag{3.4}$$

Also, count the vertices, corresponding to a.

$$a = x_1 + x_2 + \ldots + x_s \tag{3.5}$$

The degree of any vertex in $H_B(A)$ is at most k. Each vertex in X_i is adjacent to at most $k-1$ vertices in X_{i+1}, since it is adjacent to exactly one vertex in X_{i-1} (or, in the case $i=1$, to itself).

$$x_{i+1} \leq (k-1)x_i, \quad 1 \leq i \leq s-1 \tag{3.6}$$

First verify that Z, given in the following expression, is at most 0, by applying (3.6) for $i = s-1, s-2, \ldots, 1$.

$$Z \doteq (1/2 - 3\nu_s)x_s - \sum_{i=1}^{s-1}((k-2)/2 + 3\nu_s)x_i + x_1$$

We add (3.4) and twice (3.2), and apply (3.5), obtaining:

$$\begin{aligned} 2b_1 + b_2 &\leq 2x_1 + x_2 + x_2 + \ldots + ((k+1)/2)x_s \\ &= [(k+3\nu_s)a + Z]/2 \end{aligned}$$

Thus, we have established:

$$2b_1 + b_2 \leq (k+3\nu_s)a/2. \tag{3.7}$$

Now add (3.7) and (3.1), and divide by 3, for:

$$b \leq (k+\nu_s)a/2 \tag{3.8}$$

establishing Theorem 3.1.

When $k=3$, this mode of analysis cannot extend past $s=2$. In fact, one can construct an instance with $b_1 = a$, $b_2 = 0$, and $b_3 = 2a/3$; then $b/a = 5/3$, for any t-ear-improvement search. The general t-opt achieves a $3/2+\epsilon$ ratio, indicating a limitation of the weaker search. It is perhaps surprising that this is the only instance where the weaker search falls behind.

$O(\log n)$-opt. From Lemma 3.3 of [2], $4h\log n + 2$-opt (not the weak search) guarantees

$$b_2 \leq (1+1/h)a_0 \tag{3.9}$$

This is shown by converting the adjacencies of items in B_2 into edges on the vertex set A_0. By 2-opt, the adjacencies can not be both outside A_0 (and thus inside A_1); those with only one endpoint in A_0 are modelled as a self-loop on that node. If the number of edges in the resulting multigraph exceeds the above bound, it can be shown that there exists a subgraph with at most $4h\log n - 1$ nodes containing a strictly greater number of edges. Since the graph may be assumed to be connected, it contains at most two self-loops. This then maps to a set of at most $4h\log n$ items in B_2, two items in B_1, and a corresponding set of items in A, that together constitute an improvement.

Now add (3.9) with $h=1$, (3.2), and (3.1), for:

$$3a \leq (k+2)b$$

using $4\log n + 2$-opt, establishing Theorem 3.2. The complexity of this approach is $(4\Delta)^{4\log n}n \approx n^{8\log\Delta}$, which is polynomial when the degree is bounded by a constant. In fact, the problems we study are all MAX SNP-complete in the constant-degree case [15].

Using larger values of h, we can obtain a ratio of $(k+1+\epsilon)/3$ (see [9] for analysis for the independent set problem in bounded-degree graphs).

3.1.1 Approximating with non-locally-optimal solutions. Finding an improvement requires exhausting all alternatives. Within the same time frame, we may find multiple non-conflicting improvements $I_1, I_2, \ldots, I_q$, which yield the solution $A \oplus (I_1 \cup I_2 \cup \cdots \cup I_q)$. This both speeds up the process and suggests that a parallel algorithm might be possible. An improvement I conflicts with an improvement J if J is not an improvement of $A \oplus I$. A set of improvements is *independent* if no two conflict.

We find that we can indeed progress non-trivially in each iteration towards the performance bound that we seek. We focus on 2-OPT, since it yields the greatest returns. We shall show that $O(k^2\log n)$ iterations suffice to obtain a solution with the same quality as guaranteed by 2-locally optimal solutions.

LEMMA 3.1. *Suppose $b_1 + d = a + p$. Then, there exist at least $\lceil p/(k-1)\rceil$ independent 2-improvements.*

LEMMA 3.2. *Any 2-improvement conflicts with at most $2k+1$ independent 2-improvements.*

THEOREM 3.3. *$2k^2\log n$ number of iterations suffice to approximate k-SET PACKING within a factor of $(k+1)/2$.*

Proof. Suppose $2b = (k+1)a + p$. Then, using (3.1),

$$p \leq b_1 - a - kd.$$

By Lemmas 3.1, 3.2, we find at least $p/((k-1)(2k+1))$ independent improvements in each iteration. Initially $p \leq n$, so after $(k-1)(2k+1)\log n$ iterations (actually $O(p/k)$ iterations suffice), p is at most 0. ∎

Further, in $O(\epsilon^{-1}k^2)$ iterations, we will have found a solution within $(1+\epsilon)(k+1)/2$ of optimal.

Parallel implementation. For each pair of items outside of A that are adjacent to the same single item in A assign a processor. If the two items do not conflict, they form a potential improvement. Construct a graph G' whose vertex set are the possible improvements, with edges between conflicting improvements. Output a maximal independent set in this graph; this corresponds to a maximal collection of independent improvements of the current solution.

The only step that is not straightforward to implement in parallel, is finding the maximal independent set. Using the algorithm of Goldberg and Spencer [7], this can be done in $O(\log^3 n)$ time using number of processors linear in the size of the graph, or $O((n+m)^2)$.

3.2 Approximating the sizes of sets. We now consider the case of k-SET PACKING when the input sets are of varying sizes, and the measure of a solution is the sum of the sizes of the sets. This is similar to "Set Covering II" of [13], while we are here concerned with a packing, a collection of disjoint sets.

For a collection A of sets, let $\|A\|$ denote the sum of the sizes of the sets in A, or

$$\|A\| = \sum_{X \in A} |X|.$$

We modify the definition of an improvement I to mean that

$$\|A \oplus I\| > \|A\|.$$

t-OPT now searches for any such improvement.

THEOREM 3.4. *2-locally-optimal solutions for k-SET PACKING II attain a performance ratio of $(k+1)/2$.*

Proof. Assign to each element e of a set X in A a value $d(e)$ representing the ratio of the size of the set Y in B containing e to the number of elements of Y intersecting sets in A. Observe that

$$\|B\| = \sum_{X \in A} \sum_{e \in X} d(e).$$

Note that $d(e)$ is always at most $|X|$. Also observe that for at most one e in X is $d(e) \geq |X|/2$. Thus,

$$\|B\| \leq \sum_{X \in A} \left((k-1)\frac{|X|}{2} + |X| \right) = \frac{k+1}{2}\|A\|.$$

∎

Similarly one can show that t-opt attains a ratio as in Theorem 3.1.

3.3 Vertex Cover. We remark on some implications for the approximation of VERTEX COVER in claw-free graphs.

Consider the following schema for approximating VERTEX COVER. Find a maximal collection of disjoint 3-cliques in the graph; delete those vertices from the graph, and run an approximate algorithm on the remaining triangle-free graph. The approximate cover is the union of the vertices of the 3-cliques and the approximate solution on the remaining graph. An optimal solution must contain at least two vertices of each 3-clique we added; hence the performance ratio of this approach is the maximum of $3/2$ and the ratio obtained on the 3-clique-free graph [22].

Observe that a $k+1$-claw-free graph without triangle is of maximum degree at most k. Shearer [27] gave an effective and efficient algorithm that finds an independent set of size $f_s(\Delta) = (\Delta \ln \Delta - \Delta + 1)/(\Delta - 1)^2$, or roughly $(\log \Delta)/\Delta n$, in triangle-free graphs. A preprocessing technique of Hochbaum [10] then yields a VERTEX COVER performance ratio of $2 - 2f_s(\Delta) \leq 2 - 2f_s(k)$. We obtain the following conclusion.

OBSERVATION 3.1. *The* VERTEX COVER *problem in $k+1$-claw-free graphs can be approximated within a factor of $2 - 2f_s(k)$, for $k \geq 6$, and factor of 1.5, for $k = 4, 5$.*

3.4 Independent Sets in Bounded Degree Graph. The INDEPENDENT SET problem remains $\mathcal{NP}$-complete even for graphs with maximum degree 3. The bounded-degree case of the independent set problem can be thought of as a restriction of the claw-free case. However, it also allows for some additional tricks, which result in considerably better performance ratios. Berman and Fürer [2] introduced a local improvement method that attains a performance ratio of $(\Delta+3)/5+\epsilon$, but with explosive time complexity. In the search for a feasible method, a weaker search was shown to attain a $(\Delta+3)/4$ ratio in linear time for Δ constant [9]. We further improve that here to $(\Delta+2)/4+\epsilon$ ($(\Delta+2+1/3)/4+\epsilon$) for Δ even (odd).

The modification to the algorithm is to apply it recursively to the graph induced by vertices adjacent to at least two vertices in our current solution. This subgraph will have a maximum degree that is two less, thus making approximation somewhat easier. We retain this result if it proves to be larger than our previous solution, and repeat this and the previous local improvement process until our solution is optimal under both of these criterias. This allows us to claim that:

$$b_0 \leq \rho^{\Delta-2} a \tag{3.10}$$

where ρ^d is the performance ratio of the algorithm on graphs with maximum degree d.

We also obtain a variation of (3.8), using a $2s\Delta$ neighborhood:

$$b_0 \leq \frac{\Delta + \nu_s}{2} a_0. \tag{3.11}$$

The reason is that a t-improvement of a_0 has t vertices in B_0 adjacent to at most $t\Delta$ vertices in A. Of those, only $t-1$ are in A_0, by assumption, thus the rest are in A_1 which is in one-to-one correspondence with the same number of vertices in B_1.

We now add $2/\Delta$ times (3.11), $(\Delta - 2)/\Delta$ times (3.10), and once (3.2) for the recurrence:

$$\rho^{\Delta} \leq 1 + 2\nu_t \cdot 2/\Delta + \frac{\Delta - 2}{\Delta}\rho^{\Delta-2}$$

With $\rho^1 = \rho^2 = 1$, this can be bounded by:

$$\rho^{\Delta} \leq \begin{cases} (\Delta+2)/4 + \nu_s/2, & \Delta \text{ even} \\ (\Delta+2+1/3)/4 + \nu_s/2, & \Delta \text{ odd.} \end{cases}$$

Thus, we have the following generalization of [9]:

THEOREM 3.5. INDEPENDENT SET *can be approximated within a factor of* $(\Delta+2)/4+\epsilon$ $((\Delta+2+1/3)/4+\epsilon)$ *for* Δ *even (odd) using* $f(\Delta, \epsilon)n = \Delta^{O(\Delta \log 1/\epsilon)}n$ *time.*

4 Induced Subgraph Problems

Our results of the previous section carry over to various problems that involve finding a maximum induced subgraph with property π. The property needs only be closed under two operations: node deletion, and disjoint union. The former stipulates that any node induced subgraph of a graph with property π also has the property, i.e. the property is *hereditary*. The second means that if two graphs have property π, taking the disjoint union of the vertex sets and edge sets results in a graph that also has the property.

We generalize our notion of $k+1$-claw-free-ness to that of *k-neighborhood-bounded with respect to property* π. Thus the characteristic of input graphs becomes that the maximum size of a π-subgraph in the neighborhood of any node is bounded by k. The results then extend to the complements of the previously mention properties; i.e. properties $\overline{\pi}$ that holds on the complement graph $\overline{G}$ iff π holds on G.

This class includes many important properties, including all those that are monotone with respect to edge deletion. Examples include planar, outerplanar, bipartite, λ-colorable, acyclic, degree-constrained, interval, circular-arc, circle graph, chordal, comparability, permutation, perfect, line graph. It does not include all hereditary properties, e.g. that of being *complete bipartite.* That problem is also among properties with a "forbidden independent set/clique" for which strong non-approximability results are known [18]. On the other hand, our class includes various types of perfect graphs that do not have a forbidden clique or independent set.

The improved approximations for bounded-degree graphs of the previous subsection hold also here. We obtain the following theorem.

THEOREM 4.1. *The results of Theorems 3.1,3.5 hold also for approximating* INDUCED SUBGRAPH WITH PROPERTY Π, *where* Π *is hereditary and closed under disjoint union.*

Further improvements can be obtained for particular problems. 2-opt attains a $(k/\lambda + 1)/2$ ratio for INDUCED λ-COLORABLE SUBGRAPH problem, and $(k+2)/3$ ratio for VERTEX ARBORICITY. The latter can be extended to $\Delta/6+1$, via the technique of Section 3.4.

5 Covering and Partitioning Problems

5.1 Set Cover. We now consider the k-SET COVER problem. Let s denote the number of elements in the base set S. As before, A will be the cover output by our algorithm, B will be any other cover.

For ease of presentation, we shall assume that the input set collection C is *monotone*, i.e. that whenever $X \in C$, so is every subset X' of X. We then also assume that a set cover is a partition of the base set S into sets in C. Clearly, replacing sets by an appropriate superset in the original input does not increase the measure of the set cover solution, i.e. the number of sets.

A simple algorithm with a good performance is the Greedy heuristic, which yields a performance ratio of $\mathcal{H}_k = \sum_{i=1}^{k} 1/i$ for k-SC [13, 17]. Research on the hardness of approximation using interactive proof techniques show that this is best possible within a constant factor [19]. A non-standard view of this algorithm is that it finds a sequence of maximal solutions to set packing problems, starting with k-SP and ending with the trivial 1-SP.

The algorithm ApproxSetCover described below uses an approximate SET PACKING algorithm to produce a set cover solution. If the set packing solutions are merely maximal, the resulting set cover algorithm is Greedy. For a collection C and a set S', let the *restriction* of C to S', denoted by $C_{|S'}$, be the collections of subsets of S' whose supersets are contained in C.

ApproxSetCover(S, C, k)
{ C is a k-set system }
  if $k = 1$ return C
  $C_k \leftarrow$ The sets in C of size k
  $A_k \leftarrow$ ApproxSetPacking(C_k, k)
  $S_k \leftarrow S - (\cup_{c \in A_k} c)$
  return $A_k \cup$ ApproxSetCover($S_k, C_{|S_k}, k-1$)

Goldschmidt, Hochbaum, and Yu [8] obtained an improved bound of $\mathcal{H}_k - 1/6$ for k-SC, by using an *exact* 2-set packing algorithm (and thus obtaining an exact 2-set cover). This is a maximum matching problem, which can be solved in time $O(|E|\sqrt{|V|})$ [21]. We observe that 2-opt on 2-SP suffices to obtain the same ratio.

Let A_i (C_i), $i = 1, 2, 3$, denote the sets in A (C) of size i, respectively. Let S_i denote the elements contained in sets in A_i.

LEMMA 5.1. ApproxSetCover, *using* 2-opt *on the* 2-

SP *subproblem, finds an approximate* 3-SET COVER *with at most* $(s+2b)/3$ *sets.*

Proof. Consider the restriction $B_{|S_2}$ of B to the elements contained in $A_1 \cup A_2$. Let B_1' (B_2') be the sets in $B_{|S_2}$ with 1 (2) elements, respectively. Recall that we assume that the cover is a partition of S. The size of a set or collection is denoted by the corresponding lower case letter.

Note that $s_2 = s - 3a3$, so $b_1' + 2b_2' + 3a_3 = s$. Thus, the number of 2-sets in $B_{|S_2}$ is at least

$$b_2' = s - 3a_3 - (b_1' + b_2') \geq s - 3a_3 - b.$$

Since our 2-set packing is 3/2-optimal, we find at least two-thirds of these sets. Thus, our solution consists of a_3 3-sets, and at least $2(s - 3a_3 - b)/3$ 2-sets, with the remainder being 1-sets. The cost of our solution is at most

$$a_3 + \frac{2}{3}[s - 3a_3 - b] + [s - 3a_3 - \frac{4}{3}(s - 3a_3 - b)] = \frac{s+2b}{3}$$

∎

Lemma 5.1 implies a ratio of 5/3 since N is at most $3OPT$, and extends to a $\mathcal{H}_k - 1/6$ ratio for k-SET COVER (proof omitted).

2-opt runs in linear time on MATCHING, because a single iteration of a maximal set of improvements suffices. Hence, the improvement does not increase the complexity over Greedy.

Direct application of local improvement. We now consider applying the local improvement method directly on the Set Cover solution. We focus on the 3-SET COVER problem.

Given a collection C and a cover A, the following situation suggests that a local "shrinking" improvement would be possible:

Sets $x_1, x_2, \ldots x_k, x_{k+1} \in A$, $y_1, y_2, \ldots, y_k \in C$
such that $A - \{x_i\} \cup \{y_i\}$ is a cover.

This would replace $k+1$ sets in A by only k sets, while still remaining a cover.

We shall focus on a particular type of a shrinking improvement: we try to "merge" a set with some unit-size sets, producing a number of size-two sets. We seek: a) a set in A_3 and three sets in A_1 for which there are three sets in C_2 covering the same six elements, or b) a set in A_2 and two sets in A_1 for which there are two sets in C_2 covering the same four elements.

Our 3-set cover algorithm is informally as follows:

Find a maximal 3-Set Packing.
Find an optimal 2-Set Cover.
Repeat shrinking steps until locally minimum.

We can find a maximal collection of independent shrinking improvements in linear time as follows. Let C_2' be the collection of sets in C_2 which have one element in S_1. Mark each element in $S_2 \cup S_3$ with at most three sets in C_2' containing that element. For each set X in A_2, A_3, test if we can choose a group of $|X|$ disjoint sets, one from the group of sets marked to each element. If so, we have found an improvement, and we update S_i, C_2' accordingly to ensure the improvements will be independent.

Observe that A_1 is monotone decreasing under these shrinking improvements. As a result, once a maximal collection of improvements has been implemented, no further improvements are possible. Hence, the whole shrinking procedure runs in linear time.

This can also be implemented in parallel, by finding a maximal independent set in the graph of all possible improvements.

LEMMA 5.2. *The above algorithm finds a cover with at most* $(2s+5b)/7$ *sets.*

Proof. For two set collections A, A', let $A \wr A'$ denote $\{X \in A : X \cap (\cup_{X' \in A'} X') \neq \emptyset\}$ or the collection of sets in A containing elements in some set in A'.

Partition B into $B_1 = B \wr A_1$, $B_2 = (B - B_1) \wr A_2$, and $B_3 = B - B_1 - B_2$. Let b_1, b_2, b_3 denote the sizes of these collections, respectively.

By optimality of our 2-SET PACKING algorithm,

$$a_1 \leq b_1, \quad \text{and} \tag{5.12}$$
$$a_1 + a_2 \leq b_1 + b_2. \tag{5.13}$$

The local minimality of our solution ensures that at least one element of each set in A_2 must belong to a set in either B_2 or B_3. Each set in B_2 contains by definition an element in A_2, thus there are at most 2 slots remaining in such a set for elements from A_3. Thus,

$$a_3 \leq 2b_2 + 3b_3. \tag{5.14}$$

Finally, the total number of elements in our solution is precisely s.

$$a_1 + 2a_2 + 3a_3 = s. \tag{5.15}$$

Now add twice (5.12), three times (5.13), once (5.14), and twice (5.15), to obtain

$$7(a_1 + a_2 + a_3) \leq 2s + 5(b_1 + b_2 + b_3) - 2b_3,$$

as desired. ∎

Since $s \leq 3b$, this implies a performance ratio of $11/7 = \mathcal{H}_3 - 11/42 \approx 1.57$. A matching lower bound can be constructed, where $a_1 = b_1 = 5/7b$, $a_2 = b_2 = 2/7b$, and $a_3 = 2b_2 = 2a_1 - a_2 = 4/7b$.

A more careful case analysis shows that 2-opt is sufficient for the 2-SET PACKING subproblem. Then, the algorithm runs in linear time and in poly-logarithmic parallel time. We omit the proof.

We omit the proof of the following generalization.

THEOREM 5.1. *There is a polynomial time algorithm for k-*SET COVER *attaining a ratio of* $\mathcal{H}_k - 11/42$, $k \geq 3$.

Improvements are possible for larger values of k, by applying local search to the larger SET PACKING problems. For instance, a $2+\epsilon$-approximate solution to 4-SC, results in a $\mathcal{H}_4 - 25/84 + \epsilon/3 \approx 1.786$ performance ratio. Such optimization can be continued through alternate levels from 4 through k.

Further improvements are also possible using more expensive shrinking improvements; the precise analysis awaits further study.

5.2 Complementary objective functions and a graph coloring problem.

We consider here covering problems, with the objective of maximizing $s-a$, the number of sets our cover saves over the worst possible cover. The *complementary performance ratio* is defined by:

$$\overline{\rho} = \max \frac{s-b}{s-a}.$$

This complementation corresponds to the relationship between the INDEPENDENT SET and VERTEX COVER problem: the vertices not belonging to a given maximum independent set form a minimum vertex cover. Another example is in computing a short *superstring* of a set of strings: we may either want to minimize the length of the resulting string, or maximize the compression [4].

A graph coloring problem. Demange, Grisoni, and Paschos [5] considered finding an approximate coloring of the vertices of a graph under the complementary objective function. They gave a coloring algorithm with a complementary performance ratio of 2.

Let χ denote the chromatic number of of the input graph, or the minimum number of colors required. A coloring is said to be *non-trivial* if no two color classes can be merged together, i.e. that the subgraph induced by the union of any two color classes is not independent. The *achromatic* number $\overline{\chi}$ is the maximum number of colors in any non-trivial coloring. We find that the chromatic and achromatic numbers cannot differ by much under the new measure – any non-trivial coloring achieves a complementary performance ratio of 2.

Let A be any non-trivial coloring, and B any valid coloring. Note that $a \leq \overline{\chi}$ and $b \geq \chi$.

OBSERVATION 5.1. $(s-\chi)/(s-\overline{\chi}) \leq 2$

Proof. Partition A into A_1, consisting of color classes with only one vertex, and A_2, the remaining classes with two or more vertices each. Since no two color classes can be merged together, the vertices in A_1 must form a clique, and thus $a_1 \leq b$. Hence,

$$a \leq (N-a_1)/2 + a_1 \leq (N+b)/2,$$

and $N - \overline{\chi} \geq (N-\chi)/2$. Note that it suffices that merely the single-vertex color classes be non-mergeable. ∎

This is tight on the m by m bipartite graph that is complete minus $m-1$ independent edges. Another example is the path on 4 vertices. This latter example shows that the Greedy set cover algorithm has no better complementary performance ratio, even if applied to the full independent set system of the graph.

We apply our 4-SET COVER algorithms on the 4-independent set system of the graph, and show that the result maps to an identical complementary approximation of the coloring problem. The algorithms are to find a maximal 4-set packing, followed by the algorithms of Lemmas 5.1, 5.2.

THEOREM 5.2. *Applying the approximation algorithms for* 4-SC *to* GRAPH COLORING *yields complementary performance ratios of* 1.5 *and* 1.4, *respectively.*

Proof. Let a_4 denote the number of 4-sets in A. Let G' denote the graph on the remaining $s - 4a_4$ vertices. The size $\mathrm{SC3}(G')$ of the minimum cover of G' by independent sets of size at most 3 is at most the chromatic number of G', which again is at most the chromatic number of G.

Then, for the former 4-SC algorithm, by Lemma 5.1,

$$a \leq a_4 + (s - 4a_4 + 2 \cdot \mathrm{SC3}(G'))/3 \leq (s+2b)/3 - a_4/3$$

and, thus

$$\overline{\rho} \leq \frac{s-b}{s-a} \leq \frac{s-b}{2(s-b)/3} = 3/2.$$

For the latter algorithm, by Lemma 5.2,

$$\begin{aligned} a &\leq a_4 + (2(s-4a_4) + 5 \cdot \mathrm{SC3}(G'))/7 \\ &\leq (2s+5b)/7 - a_4/7, \end{aligned}$$

and thus, $(s-b)/(s-a) \leq (s-b)/(5s/7 - 5b/7) = 7/5$. ∎

This analysis is tight, if we convert the hard instances for the set covering algorithms into a graph coloring instance. All of these results apply equally to the related problems of PARTITION INTO CLIQUES and COVERING EDGES BY CLIQUES.

This problem is also MAX-SNP hard, by a reduction from 3-DM, even in graphs with no 4-independent sets. Each element of the 3-DM instance corresponds to a vertex, with two vertices non-adjacent iff they are not contained in the same triangle in the 3-DM instance. Independent vertices must all be in different partitions, thus the graph contains no independent set on 4 vertices. Because it is hard to approximate a perfect 3-DM matching [24], it is also hard to approximate an $n/3$-coloring.

6 Discussion

The set of problems that can benefit from the local search paradigm is in no way limited to the problems discussed here. Two promising problems are MIN INDEPENDENT DOMINATING SET and in bounded-degree graphs and INDEPENDENT SET in bounded-degree hypergraphs.

A structural description of the effectiveness and range of local search approximations would be interesting.

Acknowledgments

I would like to thank Jaikumar Radhakrishnan and Barun Chandra for helpful comments.

References

[1] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and hardness of approximation problems. *FOCS '92*, 14–23.

[2] P. Berman and M. Fürer. Approximating maximum independent set in bounded degree graphs. *SODA '94*, 365–371.

[3] B. Chandra, H. Karloff, and C. Tovey. New results on the old k-opt algorithm for the traveling salesman problem. *SODA '94*, 150–159.

[4] A. Czumaj, L. Gąsieniec, M. Piotrów, and W. Rytter. Parallel and sequential approximation of shortest superstrings. *SWAT '94*, 95–106.

[5] M. Demange, P. Grisoni, and V. T. Paschos. Approximation results for the minimum graph coloring problem. *IPL*, 50:19–23, 1994.

[6] U. Feige. Private communication, 1994.

[7] M. Goldberg and T. Spencer. Constructing a maximal independent set in parallel. *SIAM J. Disc. Math.*, 2(3):322–328, 1989.

[8] O. Goldschmidt, D. S. Hochbaum, and G. Yu. A modified greedy heuristic for the set covering problem with improved worst case bound. *IPL*, 48:305-310, 1993.

[9] M. M. Halldórsson and J. Radhakrishnan. Improved approximations of independent sets in bounded-degree graphs. *SWAT '94*, 195–206.

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

[11] C. A. J. Hurkens and A. Schrijver. On the size of systems of sets every t of which have an SDR, with an application to the worst-case ratio of heuristics for packing problems. *SIAM J. Disc. Math.*, 2(1):68–72, 1989.

[12] M. Jerrum. Large cliques elude the Metropolis process. *Random Structures and Alg.*, 3(4):347–359, 1992.

[13] D. S. Johnson. Approximation algorithms for combinatorial problems. *JCSS*, 9:256–278, 1974.

[14] D. S. Johnson, C. H. Papadimitriou, and M. Yannakakis. How easy is local search? *JCSS*, 37:79–100, 1988.

[15] V. Kann. *On the Approximability of NP-complete Optimization Problems.* PhD thesis, Royal Inst. Tech, Sweden, 1992.

[16] S. Khanna, R. Motwani, M. Sudan, and U. Vazirani. On syntactic versus computational views of approximability. *FOCS '94*.

[17] L. Lovász. On the ratio of optimal integral and fractional covers. *Disc. Math.*, 13:383–390, 1975.

[18] C. Lund and M. Yannakakis. The approximation of maximum subgraph problems. *ICALP '93*.

[19] C. Lund and M. Yannakakis. On the hardness of approximating minimization problems. *STOC '93*, 286–293.

[20] M. V. Marathe, H. Breu, H. B. Hunt III, S. S. Ravi, and D. J. Rosenkrantz. Simple heuristics for unit disk graphs. *Networks*, to appear. (Earlier version: *CCCG '92*, 244–249.).

[21] S. Micali and V. V. Vazirani. An $O(\sqrt{|V|}|E|)$ algorithm for maximum matching in general graphs. *FOCS '80*.

[22] B. Monien and E. Speckenmeyer. Some further approximation algorithms for the vertex cover problem. *CAAP '83*, 341–349.

[23] C. H. Papadimitriou and M. Yannakakis. Optimization, approximation, and complexity classes. *JCSS*, 43:425–440, 1991.

[24] E. Petrank. The hardness of approximation: Gap location. *ISTCS '93*, 275–284.

[25] G. H. Sasaki and B. Hajek. The time complexity of maximum matching by simulated annealing. *JACM*, 35(2):387-403, 1988.

[26] A. A. Schäffer and M. Yannakakis. Simple local search problems that are hard to solve. *SIAM J. Comput.*, 20(1):56–87, 1991.

[27] J. B. Shearer. A note on the independence number of triangle-free graphs. *Disc. Math.*, 46:83–87, 1983.

[28] M. Yannakakis. The analysis of local search problems and their heuristics. *STACS '90*, 298–311.

[29] G. Yu and O. Goldschmidt. On locally optimal independent sets and vertex covers. Manuscript, 1993.

[30] G. Yu and O. Goldschmidt. Local optimality and its application on independent sets for k-claw free graphs. Manuscript, 1994.