Approximating k-Set Cover and Complementary Graph Coloring

Magnús M. Halldórsson*

Science Institute, University of Iceland, IS-107 Reykjavik, Iceland.

Abstract

We consider instances of the Set Cover problem where each set is of small size. For collections of sets of size at most three, we obtain improved performance ratios of $1.4 + \epsilon$, for any constant $\epsilon > 0$. Similar improvements hold also for collections of larger sets. A corollary of this result is an improved performance ratio of 4/3 for the problem of minimizing the unused colors in a graph coloring.

1 Introduction

A *set system*, or a *hypergraph*, consists of a finite base set and a collection of subsets of the base set. A *cover* of a set system is a sub-collection whose elementwise union equals the base set. The SET COVER problem is given an arbitrary set system, find a minimum cardinality cover. We consider instances where each set in the system is of size at most k. The k-SET COVER (k-SC) problem is then to find a cover of a a system of *k-sets*.

The set cover problem is of fundamental importance in combinatorial optimization, with innumerable applications in operations research and other fields. The bounded set cover problem is an important special case; for instance, Goldschmidt, Hochbaum and Yu [5] cite an application in manufacturing.

Even when the maximum set size k is three, the k-SET COVER problem is known to be NP-hard, and approximating it within $1 + \delta$ is NP-hard, for some fixed $\delta > 0$ [3]. On the other hand, when maximum set size is two, the problem becomes a graph problem known as EDGE COVER, which can be solved optimally via a straightforward transformation to MAXIMUM MATCHING. It is therefore interesting to study how well we can solve the boundary case, 3-SET COVER.

We are interested in efficient heuristics that always find solutions close to optimal. Efficiency dictates that the algorithm run at least in polynomial time, and effectiveness is measured by the *performance ratio* of the algorithm, which is the maximum ratio of the size of the obtained cover to that of the optimal cover. An algorithm is said to be *ρ-approximate* if its performance ratio is at most ρ.

---

*`mmh@rhi.hi.is`. Work done in part at Japan Advanced Institute of Science and Technology - Hokuriku, Japan.

The approach considered in this paper is local search. Given a particular solution, we search for small subsets that can we swapped in and out of the solution and thereby in some sense improving it. Once no further improvement operations can be found, the solution will be moderately large in comparison with some abstract optimal solution. We shall be looking for several different types of improvements.

The premier form of a local improvement is *t-change*, where we have t sets in the solution that can be covered by $t-1$ sets outside the solution.

A second form of an improvement is an *augmenting path*. Here we have a sequence of 2-sets alternately from outside and inside the solution that connect two 1-sets from the solution. This improvement operation may involve arbitrarily many sets and thus may not be "local" in the usual sense, but what matters most is that it can be found, or its non-existence confirmed, in reasonable amount of time.

Both of the above improvement types decrease the cardinality of the solution. We also use two types of improvements that do not affect the solution size but have the effect of increasing the number of 3-sets in the solution.

Our main result is that local search of the form described approximates 3-SET COVER within a factor of $7/5+\epsilon$, in time polynomial in $\log \epsilon^{-1}$. It extends to a performance ratio of $\mathcal{H}_k - 1/3$ for k-set covers. When local search is used with t-change improvements only, we get a ratio of $3/2+\epsilon$, which follows from a result of Hurkens and Schrijver [8]. Previous published ratios for this problem are 11/6 [9, 10], 10/6 [5], and 11/7 [6], which are also based on local improvements of greedy initial solutions.

One application of approximations of covers of small sets is in the coloring problem. A non-standard measure of a coloring is the number of colors saved over the trivial allocation of a different color for each vertex. Using the set cover algorithm, we obtain a ratio of 4/3, improving previous ratios of 2 [4], 1.5 [7] and 1.4 [6].

2 The problem and the notation

The k-SET COVER problem is defined by: given a collection $\mathcal{C}$ of sets of at most k elements each drawn from a finite domain S, find a subcollection $\mathcal{C}' \subseteq \mathcal{C}$ of minimum cardinality such that $\bigcup \mathcal{C}' \stackrel{def}{=} \cup_{X \in \mathcal{C}'} X$ equals S.

In order to simplify much of the presentation, we shall be working with a slightly different problem, which we name *minimum exact cover of a monotone collection*. We require that the solution be a partition of the domain S, i.e. that the sets be disjoint in addition to forming a cover. The input collection, however, is now assumed to be monotone, in that all subsets of a set in the collection are also in the collection.

A monotone collection can be produced from the original input by adding all subsets. If explicitly represented this may increase the size of the input by a factor of at most $2^k - 1$, but it may also be possible to represent this implicitly. Given that maximum set size k is for us a small constant, this causes at most a constant factor overhead. A solution in the monotone system can be easily mapped back to a solution in the original system with the same cardinality.

Intersection hypergraph Our analysis proceeds by comparing the solution found $\mathcal{A}$ to an arbitrary cover $\mathcal{B}$. The interaction between the two solutions is often well illustrated by the *intersection hypergraph* $G(\mathcal{B}, \mathcal{A})$ of the two set collections. The sets in $\mathcal{B}$ form the vertices of the hypergraph, and the edges are given by $\{B_j \in \mathcal{B} \mid A_i \cap B_j \neq \emptyset\}$, for each A_i in $\mathcal{A}$. Since both $\mathcal{A}$ and $\mathcal{B}$ are covers with 3-sets, G has the properties that each vertex is incident on at most three edges and each edge is incident on at most three vertices. The ratio between the solution then equals the density or average degree of the graph.

Notation The following notation is used throughout.

$\mathcal{A}$ refers to the cover output by the algorithm in question, $\mathcal{B}$ is any other cover. We partition $\mathcal{A}$ into $\mathcal{A}_i$, $i = 1, 2, 3$, of the sets in $\mathcal{A}$ of size i. Partition $\mathcal{B}$ into $\mathcal{B}_1$ and $\mathcal{B}_0$, where $\mathcal{B}_1$ consists of the sets in $\mathcal{B}$ with non-empty intersections with sets in $\mathcal{A}_1$. That is, $\mathcal{B}_1 = \{B \in \mathcal{B} \mid B \cap \bigcup \mathcal{A}_1 \neq \emptyset\}$, and $\mathcal{B}_0$ contains the remaining sets $\mathcal{B} - \mathcal{B}_1$.

We define $\mathcal{A}_0$ to be the restriction of $\mathcal{A}$ to $\bigcup \mathcal{B}_0$, or $\mathcal{A}_0 = \{A_i \cap \bigcup \mathcal{B}_0 \mid A_i \in \mathcal{A}\} - \emptyset$. We shall motivate this definition when we get to the analysis.

The cardinalities of the above and other collections are denoted by the respective lower case letter, e.g. a, b. The number of elements of the base set, or the vertices of the hypergraph, is denoted by n (but also by s), and the number of sets in the input, or the edges of the hypergraph, by m.

We casually use "singleton", "doubleton", and "triplet" to refer to sets of size one, two and three, respectively, and also call them 2-sets, 3-sets etc.

An example set system Figure 1 contains an example set system. The elements of the five singleton sets in $\mathcal{A}_1$ are all marked with "o" and the region is marked off with a bold line for clarity. The doubleton sets are marked by x and y, while the triplets from $\mathcal{A}_3$ are marked by A, B, C and D. In addition to these 11 sets, the columns form the sets of the optimal solution $\mathcal{B}$. The first five columns are $\mathcal{B}_1$, as those sets contain the singleton sets of $\mathcal{A}_1$. $\mathcal{A}_0$ contains six single element sets, of the elements in the last two columns.

3 Local Search Algorithms

This paper focuses on a local improvement strategy for producing small 3-set covers. The strategy is to start with an arbitrary initial cover and iteratively search for a solution of smaller cardinality that differs only in few elements.

Definition 1 *A* t-change improvement *of a cover* $\mathcal{A}$ *is formed by sets* D_1, D_2, $\ldots$, D_t *in* $\mathcal{A}$ *and sets* $E_1, E_2, \ldots E_{t-1}$ *in* $\mathcal{C}$ *(normally outside of* $\mathcal{A}$*) such that the symmetric difference*

$$\mathcal{A}' = \mathcal{A} \oplus \{D_1, \ldots D_t, E_1, \ldots E_{t-1}\} = (\mathcal{A} - \{D_1, \ldots, D_t\}) \cup \{E_1, \ldots E_{t-1}\}$$

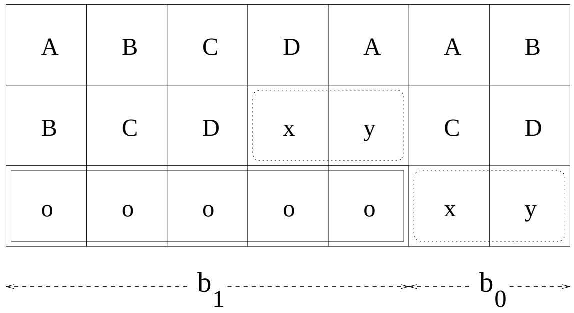

Figure 1: An example of a set system and its covers.

is also a cover.

Given any solution, finding an improvement or verifying local optimality can be done in polynomial time for any fixed value of t.

Definition 2 *An* augmenting path *with respect to a set cover solution* $\mathcal{A}$ *is a set of edges* (v_0), (v_0, v_1), $(v_1, v_2) \ldots, (v_{2k}, v_{2k+1})$, (v_{2k+1}) *in* $\mathcal{C}$ *where* $(v_0), (v_{2k+1})$ *and* (v_{2i-1}, v_{2i}) *are in* $\mathcal{A}$, *for* $1 \leq i \leq k$.

Both t-change and augmenting path improvements decrease the size of the cover, primarily by reducing the number of singleton sets. The following types do not reduce the count, but have other beneficial effects. Namely, they give preference to 3-sets, and also increase the singletons at the cost of doubletons. This will then, intuitively, make further reductions more likely.

Definition 3 *A* lasso *with respect to a set cover solution* $\mathcal{A}$ *is a set of edges forming a path* $(v_0, v_1), (v_1, v_2), \ldots (v_{2k}, v_{2k+1})$, $k \geq 1$, *a triplet* (v_1, v_2, v_{2k+1}) *and a singleton* v_0 *such that* (v_{2i}, v_{2i+1}) *is in* $\mathcal{A}$, *for* $0 \leq i \leq k$.

Definition 4 *A* fat singleton *with respect to a set cover solution* $\mathcal{A}$ *is a set of edges* (v_0), (v_1, u_1), (v_2, u_2), (v_0, v_1, v_2), (u_1), (u_2) *where the first three are contained in* $\mathcal{A}$.

Figure 2 pictures an arbitrary lasso. On the left is the set representation, with dotted lines connecting elements in the same set outside of $\mathcal{A}$. On the right is the hypergraph of $\mathcal{A}$ onto the cover given by the dotted edges.

If T is a lasso or a fat singleton with respect to $\mathcal{A}$, then the symmetric difference $\mathcal{A} \oplus T$ satisfies:

1. The cardinality of $\mathcal{A} \oplus T$ equals the cardinality of $\mathcal{A}$, and

2. The number of 3-sets in $\mathcal{A} \oplus T$ is one greater than the number of 3-sets in $\mathcal{A}$.

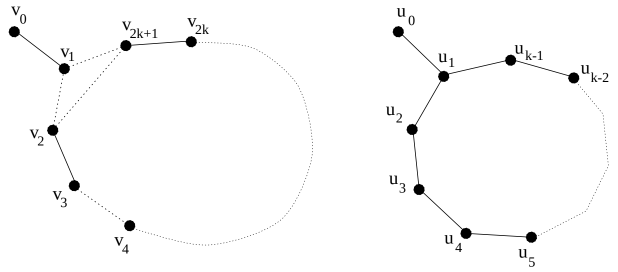

Figure 2: A lasso, as a set system, and its intersection hypergraph.

Finding the improvements Fat singletons are easily found by inspection, by marking the elements of S according to membership in $\mathcal{A}_i$, $i = 1,2,3$, and inspecting each 3-set in $\mathcal{C}$. We now indicate how the other types of improvements can be discovered efficiently.

Form a directed bipartite graph $G = (U, V, E)$ where U and V are formed by the elements of $\bigcup(\mathcal{A}_1 \cup \mathcal{A}_2)$. For an element x, let x_U (x_V) denote the corresponding vertex in U (V), respectively. The edges are given by

$$E = \{\langle x_U, y_V\rangle \mid (x, y) \in \mathcal{A}_2\} \cup \{\langle y_V, z_U\rangle \mid (y, z) \in \mathcal{C} - \mathcal{A}_2\}$$

(Notice that $\mathcal{A}_2$ contains unsorted sets, giving rise to pairs of edges in E.) Edges in one direction correspond to doubletons in the solution $\mathcal{A}$, while edges in the other direction correspond to 2-sets outside of $\mathcal{A}$. We have then a correspondence between (directed) paths in G and alternating paths in the set graph.

There is a lasso with respect to $\mathcal{A}$ iff for some set (x, y, z) in $\mathcal{C} - \mathcal{A}$ and contained in $\bigcup \mathcal{A}_2$, there is a path from x_U to y_V in G. (Hint: View x, y, z as v_2, v_{2k+1}, v_1 of Figure 2.)

There is an augmenting path iff for some x, y in $\bigcup \mathcal{A}_1$, there is a path from x_V to y_U in G.

The existence of a lasso improvement can be discovered in time proportional to finding the transitive closure of G, or $O(|E(G)||V(G)|)$. An augmenting path can be found in time $O(|E(G)|)$, via a single breadth-first search.

Algorithm Our main algorithm, LI_t, short for "local improvement", is indexed by a parameter t indicating the depth of the t-change local search.

```
LI_t(S,C):
   Start with a Greedy initial solution
   Find optima under augmenting paths
   Find optima under t-change
   Find optima under lassos and fat singletons
   Repeat search as necessary
end
```

Lassos and fat singletons preserve the solution size and increase the number of 3-sets. Augmenting paths and t-change decrease the solution size while increasing the number of 3-sets by at most $\lfloor (t-4)/2 \rfloor$. Hence, fewer than $tn/2$ iterations are ever performed.

4 Analysis

For the analysis of the algorithm, we derive a sequence of equations on the relative size of the various partitions of the optimal and heuristic solutions. Linear combinations of these inequalities combine to give the desired performance ratio.

Basic properties of set collections Let us consider some basic bounds on these set collections.

1. The sum of the cardinalities of partition classes equals the cardinality of the whole collection.

$$a_1 + a_2 + a_3 = a, \qquad b_1 + b_0 = b.$$

2. The total number of elements in a cover is s.

$$a_1 + 2a_2 + 3a_3 = s. \tag{1}$$

3. Each set trivially contains at most three elements. Thus,

$$s \leq 3b, \tag{2}$$

which is tight only when $\mathcal{B}$ is an exact cover.

Properties of small improvements Systems without small improvements obey additional bounds.

1. Suppose a set in $\mathcal{B}_1$ intersects more than one set in $\mathcal{A}_1$. Then, the three sets form a 2-improvement. Since each set in $\mathcal{B}_1$ does by definition intersect a set in $\mathcal{A}_1$, the following holds for 2-optimal covers:

$$a_1 = b_1 \tag{3}$$

2. Suppose a set in $\mathcal{A}_2$ or $\mathcal{A}_3$ contains all its elements within $\mathcal{B}_1$ (and thus none within $\mathcal{B}_0$). Then, there exists a 3- or 4-improvement. Thus, each such set in a 4-optimal solution contains a non-empty portion in $\bigcup \mathcal{B}_0$, forming a set in $\mathcal{A}_0$:

$$a_2 + a_3 = a_0. \tag{4}$$

Relating larger improvements The set system $\mathcal{A}_0$ was defined to be the assemblage of the parts of the sets in $\mathcal{A}$ that were contained in $\bigcup \mathcal{B}_0$. In other words, those parts not contained in $\bigcup \mathcal{B}_1$. The motivation for that definition is the following observation. Suppose that a t-change local search picks up a set $B \in \mathcal{B}_1$. It has then already covered a set in $\mathcal{A}$ (namely, the singleton set from $\mathcal{A}_1$ intersecting B) and thereby "paid for itself". Further, it additionally covers the other elements in the set, which are then taken care of. Thus, suppose we have obtained an improvement in $\mathcal{A}_0$, i.e. covered the $\mathcal{B}_0$-portion of several sets in $\mathcal{A}$, we can extend it to a covering of the whole of these sets by including the appropriate sets from $\mathcal{B}_1$ while maintaining the balance between the sets to be added and the sets to be deleted.

This is formalized in the following lemma, which shows that an improvement in the derived collection $(\mathcal{A}_0, \mathcal{B}_0)$ implies the existence of an improvement, albeit larger one, in the original collection $(\mathcal{A}, \mathcal{B})$.

Lemma 4.1 *If there is an r-change improvement in $(\mathcal{A}_0, \mathcal{B}_0)$, then there is a $(2r+2)$-change improvement in $(\mathcal{A}, \mathcal{B})$.*

Proof. Consider an improvement in $(\mathcal{A}_0, \mathcal{B}_0)$ of minimal size. That is, we have sets $D_1, \ldots, D_h$ in $\mathcal{A}_0$ which intersect exactly the sets $E_1, \ldots, E_{h-1}$ in $\mathcal{B}_0$, where $h \leq r$. Let d_q, $q = 1, 2, 3$ be the number of sets D_j of size q.

Consider the intersection hypergraph with vertices E_j, $j = 1, \ldots h-1$, and hyperedges $\{E_j \mid D_i \cap E_j \neq \emptyset\}$, for $i = 1, \ldots, h$. By minimality, this graph is connected. Each hyperedge of size q can reduce the number of components by at most $q-1$, while together they reduce the number of components from $h-1$ of the empty graph to 1 in the full connected graph. Thus,

$$d_2 + 2d_3 \ \geq \ h - 2.$$

Since $h = d_1 + d_2 + d_3$, the total number of elements in the D_j's is at least $2h-2$:

$$\sum_{i=1}^{s} |D_i| \ = \ d_1 + 2d_2 + 3d_3 \ \geq \ 2h - 2. \tag{5}$$

Now let us map this improvement from $\mathcal{A}_0$ to $\mathcal{A}$. Let D'_i be the[2] set in $\mathcal{A}$ that is a superset of D_i, for $i = 1, \ldots h$, and E'_j be the superset of E_j in $\mathcal{B}$, for $j = 1, \ldots h-1$. There are at most $3h$ elements contained in these sets combined. By (5), at least $2h-2$ of those elements are covered by $E'_1, \ldots, E'_{h-1}$. Each of the remaining $h' \leq h+2$ elements is outside $\mathcal{A}_0$ and $\mathcal{B}_0$ and thus, by definition, contained in a unique set in $\mathcal{B}_1$. This set in $\mathcal{B}_1$ also intersects a (single) set in $\mathcal{A}_1$. Our improvement is a union of these sets: $D'_1, \ldots D'_h$ and the h' sets in $\mathcal{A}_1$ on one side; $E'_1, \ldots, E'_{h-1}$ and the h' sets in $\mathcal{B}_1$ on the other side. ∎

[2] This set is unique, under our assumption that a cover contain only disjoint sets and the input collection be monotone.

Bounds for t-change improvements t-change improvements turn out to be surprisingly powerful, even when applied to an arbitrary starting solution. The following is a simple application of a result of Hurkens and Schrijver [8]. We can obtain the same $3/2+\epsilon$ ratio directly from Lemma 4.1 but with a worse dependence on ϵ.

Theorem 4.2 *Let $\mathcal{A} = \{A_1, A_2, \ldots A_a\}$ be a 3-set cover with no t-change improvement, and let $\mathcal{B} = \{B_1, B_2, \ldots B_b\}$ be any other 3-set cover. Then,*

$$\frac{a}{b} \;\leq\; \alpha_t \;\stackrel{def}{=}\; \frac{3}{2} + \begin{cases} \frac{1}{2^{p+1}/3-2}, & \text{if } t = 2p+1; \\ \frac{1}{2^p-2}, & \text{if } t = 2p+2. \end{cases} \tag{6}$$

The first few values of α_t are 3, 2, 9/5, 5/3, 21/13 and 11/7.

Proof. Consider the intersection hypergraph $(V, \mathcal{E})$:

$$V = \{B_1, \ldots, B_b\} \quad \text{and} \quad E_i = \{B_j | B_j \cap A_i \neq 0\} \ \text{ for } i = 1, \ldots, a.$$

We have, by t-change local optimality, that:

> For any $h \leq t$, any h of the sets in $\mathcal{E}$ cover at least h elements of V.

By the König-Hall Theorem, this can be restated as:

> Any collection of at most t sets in $\mathcal{E}$ has a system of distinct representatives.

Thus, $(V, \mathcal{E})$ satisfies the *t-SDR* property of [8], and the theorem thus follows from Theorem 1 of [8]. ∎

Hurkens and Schrijver [8] also gave a construction that yields a matching lower bound on the performance of this local improvement algorithm. The same bounds were also obtained in [6] in the context of similar local improvement algorithms for set packing problems.

We can also obtain stronger bounds for hypergraphs of bounded degree, via the ideas of Berman and Fürer [1], as in the case of the Set Packing problem [6].

Theorem 4.3 *There is a polynomial time algorithm for approximating* 3-Set Cover *in hypergraphs of degree Δ within a factor of $4/3+\epsilon$ in time $n^{poly(\Delta, 1/\epsilon)}$.*

Properties of other improvement types

Lemma 4.4 *Suppose $\mathcal{A}$ is a solution that has no lasso, no augmenting path, and no fat singleton. Then,*

$$a_1 + a_2 \;\leq\; b \tag{7}$$

Proof. Form the intersection graph H of $\mathcal{A}_1 \cup \mathcal{A}_2$ onto $\mathcal{B}$. This is a multigraph whose vertices are the sets in $\mathcal{B}$. It has an edge for each set in $\mathcal{A}_2$ connecting the corresponding sets in $\mathcal{B}$ it intersects, and a self-loop for each set in $\mathcal{A}_1$. The number of vertices of H is b and the number of edges is $a_1 + a_2$.

Suppose there exists a connected component in H with more edges than vertices. The component must include at least two loops, either as circuits or self-loops. Thus, at least one of the three following cases must occur:

Two self-loops. Then there is an augmenting path with respect to $\mathcal{A}$.

A vertex with a self-loop adjacent to two other vertices Then, there is a fat singleton with respect to $\mathcal{A}$.

A simple cycle with one vertex of degree 3. Let $(u_1, u_2, \ldots u_k = u_1)$ be a cycle with u_1 additionally adjacent to u_0. The edges map to sets in $\mathcal{A}_2$; (u_i, u_{i+1}) is the set (v_{2i}, v_{2i+1}). The vertices map to sets in $\mathcal{B}$; if we look at the subsets spanned by $v_0, \ldots v_{2k}$, we have u_0 corresponding to v_0, u_1 corresponding to (v_0, v_1, v_{2k}), and u_i, $2 \leq i \leq k$, corresponding to (v_{2i-1}, v_{2i}). Thus, we have identified a lasso with respect to $\mathcal{A}$.

We have thus proved the converse of the statement of the lemma. ∎

Main result

Theorem 4.5 *Let $\mathcal{A}$ be an output of LI_{2r+2} and $\mathcal{B}$ be a cover. Then,*

$$a \;\leq\; \frac{(\alpha_r - 1)s + (2\alpha_r - 1)b}{3\alpha_r - 2}. \tag{8}$$

Proof. Since $(\mathcal{A}, \mathcal{B})$ contains no $2r+2$-improvement, $(\mathcal{A}_0, \mathcal{B}_0)$ contains no r-improvement, by Lemma 4.1. Applying Theorem 4.2 to $(\mathcal{A}_0, \mathcal{B}_0)$, we have:

$$a_0 \;\leq\; \alpha_r b_0. \tag{9}$$

Combine (3), (4), and (9):

$$a \;\leq\; b_1 + \alpha_r b_0, \tag{10}$$

and (3), (7), and (1):

$$3a \;\leq\; b_1 + s + b. \tag{11}$$

Now add (10) and $(\alpha_r - 1)$ times (11) to yield the theorem. ∎

Recalling that $s \leq 3b$ and substituting for α_r in the statement of the previous theorem yields our main claim.

Corollary 4.6 *The performance ratio of LI_{2r+2} is at most*

$$\frac{5\alpha_r - 4}{3\alpha_r - 2} = \frac{7}{5} + O(2^{-r/2}).$$

For $r = 1, 2, 3$, the ratios are 11/7, 3/2, and 25/17, respectively.

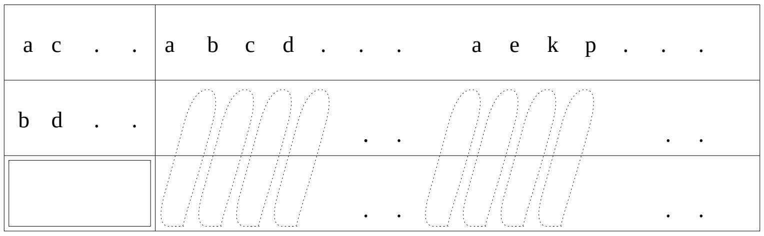

Figure 3: A construction for a ratio 1.4 for any LI_t algorithm

Limitations Figure 1 shows that that the 11/7 bound for LI_4 is tight. Rather than constructing examples for each value of t, we focus on the asymptotic case.

Figure 3 illustrates a construction that contains no lassos, fat singletons, augmenting paths, and no t-change improvements, for any fixed value of t. $\mathcal{B}_1$ is one-fifth of $\mathcal{B}$ and contains one element from each set in $\mathcal{A}_3$ as well as the singleton sets. $\mathcal{B}_0$ contains b_0 doubletons, and two elements from each triplet, properly arranged.

The intersection hypergraph of $\mathcal{A}_0$ on $\mathcal{B}_0$ in this example is a 3-regular simple graph with a high girth g; there do exist such graphs with girth $\log n - 1$ [2]. Any improvement to $\mathcal{A}$ must contain an improvement of $\mathcal{A}_0$, since the singletons are the only sets fully contained in $\mathcal{B}_1$. Any improvement must contain a cycle in the intersection hypergraph, and thus must be of cardinality at least g.

Complexity The naive bound for LI_t would give complexity of $n^t m^{t-1}$ per iteration; showing $O(n^t)$ is not too difficult. We sketch briefly how LI_{2r+2} can be implemented in time $O(n^r)$ per iteration for $r \geq 2$, and $O(mn)$ for $r = 1$.

Observe from our analysis, particularly Lemma 4.1, that we need only a special type of $2r+2$-change improvement: r sets in $\mathcal{A}_2 \cup \mathcal{A}_3$, along with up to $r+2$ sets in $\mathcal{A}_1$ (and appropriate sets from $\mathcal{B}$). We examine all r different sets in $\mathcal{A}_2 \cup \mathcal{A}_3$. These contain only $O(r)$ elements, so we can try all possible partitions to see if they form sets in $\mathcal{C} - \mathcal{A}$, and to match parts of these partition classes with elements in $\mathcal{A}_1$. For the latter, we will have precomputed for each singleton or doubleton, the number of elements in $\mathcal{A}_1$ that combine to form sets in $\mathcal{C} - \mathcal{A}_1$.

LI_4 can be implemented using Greedy, followed by augmenting paths improvements, followed by restricted 4-change improvements. It can also be done in time $O(m)$ per iteration (or $O(n^2)$, if the instance is dense), for a total complexity of $O(mn)$.

k-Set Cover The best studied heuristic for k-Set Cover is the Greedy algorithm, which has a performance ratio of $\mathcal{H}_k$ [9, 10], or 11/6 for 3-SC. The algorithm operates by iteratively adding a set X to the solution that covers the most number of (yet uncovered) elements and then eliminating those sets that intersect X from further consideration. Observe that the size of the selected sets is monotone non-increasing; thus, when the first set of size k is selected, the remaining collection has a maximum set size of k and the problem of covering the remaining elements is a k-SC problems. We can therefore partition the solution into layers consisting of sets covering the same number of elements, and after peeling off one layer the remainder is a set cover problem with a smaller maximum set size.

Our results generalize to the k-SET COVER problem, using the Greedy choices of sets of size 4 and more, and optimizing the smaller sets with the preceding technique.

Theorem 4.7 *The performance ratio of* LI_6 *for* k-SET COVER *is at most* $\mathcal{H}_k - 1/3$.

Proof. When $k = 3$, the statement is trivial, so assume $k \geq 4$.

Recall that the Greedy algorithm chooses sets of non-increasing size, resulting in a sequence of subproblems i-SC, where i goes from k down to 2. Let N_i, $3 \leq i \leq k$, denote the number of elements to be covered in the i-SC subproblem. Then, $N_i - N_{i-1}$ represents the number of elements covered by sets of size i, and $N_k = s$. Further, let X_i denote the number of sets in the optimal solution (for k-SET COVER) that contain i or more elements. Then,

$$N_i \;\leq\; X_1 + X_2 + \ldots + X_i.$$

Notice that the size of the optimal solution for i-SC is at most b, the size of the optimal solution for k-SC, since Greedy leaves no set of size $i+1$ or more.

The number of sets used by LI_6 on the 3-SC subproblem is bounded by $(N_3 + 3b)/4$, by Theorem 4.5.

The number of sets that the combined algorithm uses is bounded by:

$$\begin{aligned}
a &\leq \frac{N_k - N_{k-1}}{k} + \frac{N_{k-1} - N_{k-2}}{k-1} + \cdots + \frac{N_4 - N_3}{4} + \frac{N_3}{4} + \frac{3b}{4} \\
&= \frac{N_k}{k} + \sum_{i=4}^{k-1} \frac{N_i}{(i+1)i} + \frac{3b}{4} \\
&\leq \frac{N_k}{k} + \sum_{i=4}^{k-1} \frac{1}{(i+1)i} \sum_{j=1}^{i} X_j + \frac{3b}{4} \\
&= \frac{N_k}{k} + \sum_{j=4}^{k-1} X_j \sum_{i=j}^{k-1} \frac{1}{(i+1)i} + (X_1 + X_2 + X_3) \sum_{i=4}^{k-1} \frac{1}{(i+1)i} + \frac{3b}{4} \\
&= \frac{N_k}{k} + \sum_{j=4}^{k-1} X_j (\frac{1}{j} - \frac{1}{k}) + (X_1 + X_2 + X_3)(\frac{1}{4} - \frac{1}{k}) + \frac{3b}{4} \\
&\leq \sum_{j=4}^{k} \frac{X_j}{j} + \frac{X_1 + X_2 + X_3}{4} + \frac{3b}{4}.
\end{aligned}$$

We have obtained a bound that is stronger than what the lemma claims, which improves on the corresponding bound of [9, 10, 5]. Since $X_i \leq b$,

$$a \;\leq\; (\mathcal{H}_k - 1/3)b.$$

∎

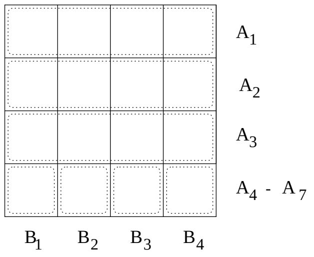

Figure 4: A construction with a ratio of 7/4 on 4-set systems.

This bound is tight for the 4-set system on sixteen elements that appears in Figure 4. The construction generalizes easily to a matching bound for $k > 4$.

5 Applications to Graph Coloring

The *graph coloring* problem is that of assigning a discrete value, or color, to each vertex such that adjacent vertices get different colors. Typically, the measure is the number of colors used $HEU(G)$, for which the objective is to minimize. Demange, Grisoni, and Paschos [4] considered finding a coloring with the objective of maximizing the number of *unused* colors, or $|V(G)| - HEU(G)$. The *complementary performance ratio* of a coloring algorithm is defined to be

$$\max_G \frac{|V(G)| - OPT(G)}{|V(G)| - HEU(G)}.$$

A fairly complicated algorithm with a complementary performance ratio of 2 was given in [4]. Improved results were obtained independently by Hassin and Lahav [7] and ourselves [6] using very similar ideas. They include a trivial algorithm with ratio of 2, and an algorithm with a ratio of 1.5. A connection of this problem to set cover was made explicit in [6], and the improved 3-set cover algorithm obtained there used to improve the coloring ratio to 1.4. We take it one step further to a ratio of 4/3 via the set cover algorithm of previous section.

Lemma 5.1 *Suppose there is an algorithm that finds a 3-set cover with at most $(1-z)s + zb$ sets, where $0 < z \leq 3/4$. Then, there is a coloring algorithm with a complementary performance ratio of at most $1/z$.*

Proof. First find a maximal collection of disjoint 4-independent sets in the graph, and color each with a different color. From the remaining graph G', form the set system consisting of all independent sets, which are necessarily of size at most three. Apply the set cover algorithm, and color each of the sets in the cover with a different color.

Let HEU_4 denote the number of 4-sets found, and let $s = |V(G')| = |V(G)| - 4HEU_4$ be the remaining elements. The number of color classes is at most

$$HEU \leq HEU_4 + (1-z)(|V(G)| - 4HEU_4) + zOPT.$$

Thus,

$$|V(G)| - HEU \geq z(|V(G)| - OPT) + HEU_4(3 - 4z)$$

which is at least $z(|V(G)| - OPT)$ whenever $z \leq 3/4$. ∎

Applying the above lemma to (8), we obtain an improved complementary performance ratio for Graph Coloring.

Theorem 5.2 *Using* LI_6 *on the set system formed by the independent sets of size at most 4 yields a coloring with a complementary performance ratio of* $4/3$.

6 Discussion

The same bounds can be obtained for the Set Multi-Cover problem, where elements are to be covered possibly more than once. Also recall the immediate applications via standard reductions to the Dominating Set problem when maximum degree is bounded by $k-1$ (or Total Dominating Set of degree k), as well as the Hitting Set problem when each element occurs at most k times.

An important open problem is to determine the approximability of these or similar algorithms on the weighted version of the Set Cover problem.

Acknowledgments

I would like to thank Martin Fürer and Rafi Hassin for helpful comments.

References

[1] P. Berman and M. Fürer. Approximating maximum independent set in bounded degree graphs. In *Proc. Fifth Ann. ACM-SIAM Symp. on Discrete Algorithms*, pages 365–371, Jan. 1994.

[2] B. Bollobás. *Extremal Graph Theory.* Academic Press, 1978.

[3] P. Crescenzi and V. Kann. A compendium of NP optimization problems. Dynamic on-line survey available at `nada.kth.se`, 1995.

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

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

[6] M. M. Halldórsson. Approximating discrete collections via local improvements. In *Proc. Sixth ACM-SIAM Symp. on Discrete Algorithms*, pages 160–169, Jan. 1995.

[7] R. Hassin and S. Lahav. Maximizing the number of unused colors in the vertex coloring problem. *Inform. Process. Lett.*, 52:87–90, 1994.

[8] 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, Feb. 1989.

[9] D. S. Johnson. Approximation algorithms for combinatorial problems. *J. Comput. Syst. Sci.*, 9:256–278, 1974.

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