SAND99-1978C

Die Backside FIB Preparation for Identification and Characterization of Metal Voids

Ann. N. Campbell and William F. Filter
Sandia National Laboratories, Albuquerque, NM

Nicholas Antoniou
Micrion Corp., Peabody, MA

Abstract

Both the increased complexity of integrated circuits, resulting in six or more levels of integration, and the increasing use of flip-chip packaging have driven the development of integrated circuit (IC) failure analysis tools that can be applied to the backside of the chip. Among these new approaches are focused ion beam (FIB) tools and processes for performing chip edits/repairs from the die backside. This paper describes the use of backside FIB for a failure analysis application rather than for chip repair. Specifically, we used FIB technology to prepare an IC for inspection of voided metal interconnects ("lines") and vias. Conventional FIB milling was combined with a super-enhanced gas assisted milling process that uses XeF_2 for rapid removal of large volumes of bulk silicon. This combined approach allowed removal of the TiW underlayer from a large number of M1 lines simultaneously, enabling rapid localization and plan view imaging of voids in lines and vias with backscattered electron (BSE) imaging in a scanning electron microscope (SEM). Sequential cross sections of individual voided vias enabled us to develop a 3-d reconstruction of these voids. This information clarified how the voids were formed, helping us identify the IC process steps that needed to be changed.

Introduction

Performing FIB circuit modification from the backside of a chip has become routine for flip-chip packaged ICs, where backside access to the chip is required if electrical operation of the IC is to be preserved in the original package [1-4]. Backside FIB modification is also being considered for making repairs at lower-lying nodes (e.g., M1) in multi-layer non-flip-chip ICs that would otherwise require elaborate circuit modification from the front side. Beyond this, backside FIB sample preparation techniques offer a powerful new capability which failure analysts are just beginning to exploit. This paper describes one such failure analysis application: the use of backside FIB milling combined with conventional FIB milling, imaging, and cross-sectioning to enable the investigation of voiding and etch defects in a 2-level metal (aluminum) CMOS technology.

The goal of our study was to identify and image voiding at the M1 level and in the M1 to M2 vias without performing any deprocessing steps that would introduce additional voiding or alter existing voids. Backscattered electron (BSE) imaging performed at the highest primary beam energies (say, 30-40kV) of most scanning electron microscopes (SEMs) can often be used to characterize voiding from the front side of the chip [5], but with limited resolution and only in the uppermost interconnect layers. The resolution of BSE void images suffers even for the top level metal due to the thickness and, for unplanarized technologies, the roughness of the passivation layer covering it. Obtaining crisp images of voids in vias or deeper levels of metal from the front side is problematic because of the thickness of overlying materials and the nonplanar topography in some IC technologies. In our case, the M1 to M2 vias were formed from aluminum as part of the M2 deposition, resulting in a highly nonplanar topography that produced marked shadowing in the SEM images of the vias. More important, however, the presence of a TiW layer at the bottom of M1 prevented imaging of the voids in M1 and in the vias. The strong backscattered signal from the TiW overwhelms the signal from the aluminum interconnect, obscuring any evidence of voiding. While FIB cross-sections through vias or buried metal layers can be used to image voids, only one location or via at a time can be studied, and this technique obviously cannot be used to locate the voids in the first place. Thus, it was necessary to remove the TiW underlayer in order to successfully perform BSE imaging of the voids in M1 and in the M1 to M2 vias. The backside FIB technique described in this paper exposes long lengths of interconnects and

DISCLAIMER

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

On the red-blue set cover problem

ROBERT D. CARR [1] SRINIVAS DODDI [2] GORAN KONJEVOD [3] MADHAV MARATHE [2]

July 15, 1999

Abstract

Given a ground set $S = \{s_1, \ldots, s_k\}$ and a family of its subsets $\mathcal{S} = \{S_1, \ldots, S_m\} \subseteq 2^S$, the classical set cover problem consists of finding the minimum number of sets $F \subseteq 2^S$ that cover all the elements in the ground set S.

Given a finite set of "red" elements R, a finite set of "blue" elements B and a family $\mathcal{S} \subseteq 2^{R \cup B}$, the *red-blue set cover* problem is to find a subfamily $\mathcal{C} \subseteq \mathcal{S}$ which covers all blue elements, but which covers the minimum possible number of red elements.

We note that RED-BLUE SET COVER is closely related to several combinatorial optimization problems studied earlier. These include GROUP STEINER TREE, DIRECTED STEINER TREE, MINIMUM COLOR PATH, MINIMUM MONOTONE SATISFYING ASSIGNMENT, SYMMETRIC LABEL COVER, etc.

We further show that unless P=NP, even the restriction of RED-BLUE SET COVER where every set contains only one blue element can not be approximated to within $O(2^{\log^{1-\delta} n})$, where $\delta = 1/\log\log^c n$, for any constant $c < 1/2$ (where $n = \mathcal{S}$). This extends results of Dinur and Safra and answers an open question of Motwani and Goldwasser, placing LABEL COVER in the class MMSA_3,

We give integer programming formulations of the problem and use them to obtain a $2\sqrt{n}$ approximation algorithm for the restricted case of RED-BLUE SET COVER in which every set contains only one blue element.

Keywords: Approximation Algorithms, Computational Complexity, Set Cover, Linear Programming, Multi-criteria Problems.

AMS 1991 Subject Classification: 68Q25, 68Q45, 90B06, 68R10

---

[1]Email: bobcarr@cs.sandia.gov Applied Mathematics Department, Sandia National Laboratory, P.O. Box 5800, Albuquerque, NM 87185. Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under Contract DE-AC04-94AL85000.

[2]Email: {srinu,marathe}@lanl.gov Los Alamos National Laboratory, P.O. Box 1663, MS B265, Los Alamos, NM 87545. Research supported by the Department of Energy under Contract W-7405-ENG-36.

[3]Email: konjevod@andrew.cmu.edu| Dept. of Mathematical Sciences, Carnegie Mellon University, Pittsburgh, PA 15213-3890 and Los Alamos National Laboratory. Supported by an NSF CAREER grant CCR-9625297 and DOE Contract W-7405-ENG-36.

1 Introduction and summary of results

Let $R = \{r_1, \ldots, r_\rho\}$ and $B = \{b_1, \ldots, b_\beta\}$ be two (disjoint) finite sets, and let $\mathcal{S} \subseteq 2^{R \cup B}$ be a family of subsets of $R \cup B$. We call the elements of B *blue elements* and the elements of R *red elements.* The RED-BLUE SET COVER problem is to find a subfamily $\mathcal{C} = \{S_{i_1}, \ldots, S_{i_m}\} \subseteq \mathcal{S}$ which covers all blue elements, and which covers the minimum possible number of red elements. If the total number of red elements contained in the sets of this subfamily is denoted by ℓ, then

$$\ell = \min \left\{ \left| R \cap (\cup_j S_{i_j}) \right| \mid \cup_j S_{i_j} \supseteq B \right\}.$$

In the rest of the paper, we denote by n the size of the family $\mathcal{S}$,

$$n = |\mathcal{S}|.$$

Note that every instance of RED-BLUE SET COVER may be transformed into an equivalent one where every set contains only one blue element. Namely, given a set $S \in \mathcal{S}$, make a copy of S, denoted by S^b, for each blue element $b \in S$, and remove all other blue elements from S^b. We will use this observation later. The RED-BLUE SET COVER problem arose naturally in considering certain data mining problems. We describe some of these considerations in section 6. However, our reasons for investigating the red-blue set cover problem come from complexity and approximation algorithm theory.

We relate RED-BLUE SET COVER to several problems investigated in the literature. These include the group Steiner tree, minimum monotone satisfying assignment and minimum color path problems. Most of these are special cases of RED-BLUE SET COVER and these reductions supply our hardness results. Two of our reductions place the label cover problem in the class MMSA_3. This extends the recent results of Goldwasser and Motwani [13] and Dinur and Safra [7], who showed that LABEL COVER was either in MMSA_3 or in MMSA_4. The strongest hardness result we obtain shows that unless P=NP the special case of RED-BLUE SET COVER where every set contains only one blue element cannot be approximated to within $O(2^{\log^{1-\delta} n})$, where $\delta = 1/\log\log^c n$, for any constant $c < 1/2$. The result is obtained via a polynomial-time approximation-preserving reduction from SYMMETRIC LABEL COVER.

On the other hand, we provide a polynomial-time $2\sqrt{n}$ approximation algorithm for the same restriction of RED-BLUE SET COVER. The approximation algorithm is based on combining a simple greedy algorithm and rounding a linear relaxation of the problem. The ideas can be extended to obtain a $2\sqrt{k \cdot n}$-approximation algorithm when $|S_i \cap B| \leq k$. A $O(n^{1-1/k} \log n)$-approximation algorithm for the RED-BLUE SET COVER problem restricted $|S_i \cap R| \leq k$ is also presented. We also consider a number of other natural linear programming formulations and discuss their merit in the context of obtaining polynomial time approximation algorithms. All our positive results (linear formulations, approximation algorithm) apply as well to the weighted version of RED-BLUE SET COVER, where each red element has a nonnegative *weight* associated with it, and the goal is to minimize the total weight of red elements in a cover of B by the sets from $\mathcal{S}$.

2 Related problems

RED-BLUE SET COVER contains the classical set cover and several known generalizations of this problem as its special cases.

Set cover. The SET COVER problem (Johnson [16]) can be viewed as the RED-BLUE SET COVER problem where each set $S \in \mathcal{S}$ contains exactly one red element, and no red element is contained in more than one set. The goal is to cover all blue elements using the minimum possible number of

subsets from $\mathcal{S}$. This reduction shows that the RED-BLUE SET COVER problem is at least as "hard" as the classical SET COVER problem.

Group Steiner and directed Steiner problems. Another special case of RED-BLUE SET COVER is the (rooted, unweighted) group Steiner problem for trees studied by Garg, Konjevod and Ravi [12]. Let a tree $G = (V, E)$ be given together with a family of subsets of vertices $\mathcal{G} = \{g_1, \ldots, g_m\}$, called *groups*. Let one vertex $v \in V$ be marked as the *root*. We call all vertices belonging to some group *terminals*. The objective of GROUP STEINER TREE is to find a subtree of G that contains the root and at least one vertex of each group, such that it has the minimum possible number of edges. This problem can be modeled by RED-BLUE SET COVER. Let $B = \mathcal{G}$, let $R = E$, and for each path P from the root to a vertex v of some group g let S_P be the set formed by the edges of P and by g. We define $\mathcal{S}$ to be the family of all sets of the form S_P for a path from the root to one of the terminals. The resulting RED-BLUE SET COVER instance is equivalent to the original GROUP STEINER TREE instance.

A similar reduction can be used to transform any instance of DIRECTED STEINER TREE (see for example Charikar et al. [6]) into an instance of RED-BLUE SET COVER. However, in general, the RED-BLUE SET COVER instance will have exponentially many sets (since there may be exponentially many paths from the root to the terminals in a directed graph). Thus our reduction is only polynomial in the case where there are only polynomially many root-terminal paths in the directed Steiner instance.

Minimum color path. Krumke and Wirth [17] discuss approximability of the following problem: given a graph $G = (V, E)$ with a function $c : E \to C$, where C is a finite set of colors, find a spanning tree of G whose edges are labeled with minimum possible number of colors. They show that the problem is as hard to approximate as set cover, and complement this result by an greedy algorithm whose approximation guarantee $(2 \log |V|)$ essentially matches that of set cover.

This problem can be naturally generalized to the case where a feasible solution is a more general subgraph, for instance a forest satisfying connectivity requirements between given pairs of vertices. However, this problem (min-color generalized forest) is at least as difficult to approximate as RED-BLUE SET COVER. Even the simplest case where only one pair of vertices $\{s, t\} \subseteq V$ is given and the goal is to find an s-t path that uses fewest colors contains RED-BLUE SET COVER as a special case. This problem was first considered in [15].

The reduction is as follows: given an instance of RED-BLUE SET COVER we construct an equivalent instance of MINIMUM COLOR PATH. First we create one color for each red element. Then we create some vertices of G. Let there be a vertex v_i for each blue element b_i, and another vertex b_0. We denote b_0 by s and $b_{|B|}$ by t. For each set S containing b_i, we add a path between v_{i-1} and v_i whose length is equal to the number of red elements in S. We arbitrarily assign to each of the edges in this path one of the red elements in S and color the edge with the corresponding color. Clearly, every s-t path in this graph defines a collection of sets (the paths chosen between pairs of vertices v_{i-1} and v_i) and these sets form a red-blue cover. The cost of the cover is equal to the number of different colors used in the s-t path. Similarly, every red-blue cover defines an s-t path of equal cost.

3 Hardness of approximation

We continue our discussion on relating RED-BLUE SET COVER to other combinatorial optimization problems and as corollaries obtain stronger non-approximability results for RED-BLUE SET COVER and related problems.

Let us use $\Pi_1 \leq_A \Pi_2$ to denote that the problem Π_1 can be polynomial time reduced in an approximation-preserving way to Π_2. We also use $\Pi_1 \equiv_A \Pi_2$ to denote $\Pi_1 \leq_A \Pi_2$ and $\Pi_2 \leq_A \Pi_1$.

Symmetric label cover. The *label cover problem* (see, for example, Arora and Lund [4]) was originally defined in order to make easier the use of the PCP theorem in proving approximation hardness of SET COVER and related problems. One variant of this problem is described in Dodis and Khanna [9], where it is called *symmetric label cover.*

A complete bipartite graph $G = (U, W)$ (where U and W are the two parts of the bipartition of G of n vertices each) is given, together with two finite sets A and B (called the *label sets*), and for each edge $uw \in U \times W$, a (non-empty) relation $R_{uw} \subseteq A \times B$. A feasible solution is a pair of mappings (*label assignments*) (ϕ, ψ), $\phi : U \to 2^A$, $\psi : W \to 2^B$ such that each edge uw is *consistent*, that is there exists a pair $(a, b) \in \phi(u) \times \psi(w)$ such that $(a, b) \in R_{uw}$. The objective is to minimize $\sum_{u \in U} |\phi(u)| + \sum_{w \in W} |\psi(w)|$.

As discussed in [9] unless $\mathbf{P} = \mathbf{NP}$, even SYMMETRIC LABEL COVER cannot be approximated to within $O(2^{\log^{1-\delta} n})$, where $\delta = 1/\log\log^c n$, for any constant $c < 1/2$, where $n = \max\{|A|, |B|, |U|\}$. We use this to obtain the following result:

Theorem 3.1. SYMMETRIC LABEL COVER $\leq_A$ RED-BLUE SET COVER. *Thus, unless P=NP,* RED-BLUE SET COVER *cannot be approximated to within a factor* $O(2^{\log^{1-\delta} n})$ *, where* $\delta = 1/\log\log^c n$, *for any constant* $c < 1/2$, *and where* $n = |\mathcal{S}|$.

Proof. Let an instance of SYMMETRIC LABEL COVER be given using the notation above. We construct an instance of RED-BLUE SET COVER such that **(1)** each feasible red-blue cover corresponds to a label assignment of equal cost and **(2)** each feasible label assignment corresponds to a red-blue cover of no greater cost.

For each edge $uw \in U \times W$, define a blue element b_{uw}. For each pair $(u, a) \in U \times A$ define a red element r_{ua}, and for each pair $(w, b) \in W \times B$ a red element r_{wb}. Finally, for each quadruple $(u, w, a, b) \in U \times W \times A \times B$ such that $(a, b) \in R_{uw}$, define a set $S_{uwab} = \{b_{uw}, r_{ua}, r_{uw}\}$.

Consider now a feasible red-blue cover $\mathcal{C}$. We define a label assignment (ϕ, ψ) in the following way:

$$\phi(u) = \{a \in A \mid r_{ua} \in \cup_{S \in \mathcal{C}} S\}, \quad \psi(w) = \{b \in B \mid r_{wb} \in \cup_{S \in \mathcal{C}} S\}.$$

Each blue element b_{uw} is covered by a set in $\mathcal{C}$. This means that in the cover, there is a set of the form S_{uwab}, in which case, $(a, b) \in R_{uw}$. This implies consistency of the edge uw because S_{uwab} contains r_{ua} and r_{wb}, and hence $(a, b) \in \phi(u) \times \psi(w)$. Since

$$\sum_{u \in U} |\phi(u)| = \sum_{u \in U} |\{a \in A \mid r_{ua} \in \cup_{S \in \mathcal{C}} S\}|,$$

the total number of labels used for elements of U is equal to the number of red elements in the cover that correspond to elements of U. Similarly, the number of labels used for W is equal to the number of red elements corresponding to W, and thus the cost of the label assignment is equal to the cost of the red-blue cover.

On the other hand, from a feasible label assignment we can construct a red-blue cover $\mathcal{C}$ of no greater cost. Given a label assignment, we include in $\mathcal{C}$ exactly those sets S_{uwab} such that $(a, b) \in \phi(u) \times \psi(w)$. Consider a blue element b_{uw}. Since uw is consistent, there will be at least one pair $(a, b) \in \phi(u) \times \psi(w)$ such that $(a, b) \in R_{uw}$, which means that b_{uw} gets covered by the set $S_{uwab} \in \mathcal{C}$.

Let $r_{ua} \in \bigcup_{S \in \mathcal{C}} S$ (respectively, $r_{wb} \in \bigcup_{S \in \mathcal{C}} S$). Then clearly $a \in \phi(u)$ (resp. $b \in \psi(w)$). Hence

$$|\{r \in R \mid \exists S \in \mathcal{C}, r \in S\}| \leq \sum_{u \in U} |\phi(u)| + \sum_{w \in W} |\psi(w)|.$$

Finally, note that in the above reduction $|\mathcal{S}| \leq m^{1/4}$, where $m = \max\{|A|, |B|, |U|\}$. This implies the theorem since a polynomial increase in the problem size can be neutralized by adjusting δ. □

An alternate way of showing inapproximability of RED-BLUE SET COVER is to use two-prover proof systems and was pointed out to us by Feige [10].

Minimum monotone satisfying assignment (MMSA). This problem was independently introduced by Alekhnovich, Buss, Moran and Pitassi [3] and by Goldwasser and Motwani [13] (under the name AND-OR scheduling). The problem MMSA_k is specified by a monotone formula (that is, a formula that uses no negations) of depth k (the formula has k levels of alternating AND and OR gates, where the top level consists of an AND gate). The goal is to find a satisfying assignment that minimizes the number of variables set to TRUE. General MMSA places no restriction on the formula depth. Note that MMSA_2 is the problem of determining the minimum satisfying assignment for a monotone formula in conjunctive normal form and is equivalent to SET COVER. The variables in the instance of MMSA_2 are in one-to-one correspondence with the sets in the SET COVER instance. The topmost AND gate corresponds to requiring that every element of the ground set must be covered, and the OR gates underneath to allowing an element to be covered by any of the sets that contain it.

Analogously, MMSA_3 is equivalent to RED-BLUE SET COVER. Variables correspond to red elements. There is an OR gate for each blue element, and AND gates below an OR gate correspond to the sets that cover the corresponding blue element. Finally, a lowest-level conjunction contains a variable iff the red element corresponding to the variable appears in the set corresponding to the conjunction. Now the topmost AND gate requires that every blue element must be covered, the layer of OR gates says that a blue element may be covered by any of the sets that contain it, and the final layer of AND gates allows a set to be included in the cover only if all the red elements that it contains have been counted by the objective function.

Alekhnovich et. al. [3] show that (unrestricted) MMSA is at least as hard to approximate as LABEL COVER. Goldwasser and Motwani [13] reduce label cover to MMSA_4, and Dinur and Safra [7] further show a reduction from MMSA_3 to LABEL COVER. Denote RED-BLUE SET COVER by RBSC, LABEL COVER by LC and SYMMETRIC LABEL COVER by SLC. Then we have

$$\text{MMSA}_3 \leq_A \text{LC} \leq_A \text{MMSA}_4.$$

In contrast, combining the reduction of Theorem 3.1 and the above correspondence between RED-BLUE SET COVER and MINIMUM MONOTONE SATISFYING ASSIGNMENT with the results in [7, 13], we have

$$\text{MMSA}_3 \leq_A \text{LC} \leq_A \text{SLC} \leq_A \text{RBSC} \equiv_A \text{MMSA}_3,$$

showing that $\text{LC} \equiv_A \text{MMSA}_3$ and thus precisely placing LABEL COVER in the MMSA hierarchy.

Corollary 3.2. $\text{MMSA}_3 \equiv_A \text{LC}$.

4 Linear programming relaxations

Consider the following natural formulation of the RED-BLUE SET COVER problem.

$$
\begin{aligned}
&\min \sum_r y_r \\
&y_r \geq x_S \quad \forall (r,S) : r \in S \\
&\sum_{S \ni b} x_S \geq 1 \quad \forall b \in B \\
&x \in \{0,1\}^{|\mathcal{S}|},
\end{aligned}
\tag{1}
$$

This formulation is not very useful to us because the gap between the optimal integral and fractional solutions may be huge. As the following example shows, the objective function value of the linear relaxation of IP (1) can be as much as $\Omega(n)$ times smaller than the value of the optimal solution. Consider $R = \{r\}$, $B = \{b_1, \ldots, b_n\}$ and the family $\mathcal{S} = \{S_1, \ldots, S_n\}$, where $S_i = \{r\} \cup B \setminus \{b_i\}$. All feasible integral solutions must cover the red element r, and so have cost 1, but the linear program may assign the value $1/(n-1)$ to each set and so the objective function value is only $1/(n-1)$.

We can improve the linear relaxation of IP (1) by restricting the set of instances considered to those where each set contains only one blue element (as described earlier). If this assumption is satisfied, we may use the following inequality for each pair $(r,b) \in R \times B$:

$$
y_r \geq \sum_{S \supseteq \{r,b\}} x_S.
$$

Note that these inequalities are not valid in general. There may be feasible integral solutions that violate some of these inequalities (namely, those where more than one set covers some blue element). However, in this case, all but one of the sets covering b can be discarded without increasing the cost of the solution. In this way one obtains a "minimal" solution of no greater cost which satisfies these inequalities. We refer to this improved integer program as IP (1)'.

The new inequalities help eliminate some bad examples like the one described above. However, before discussing this improved formulation further, let us consider a different relaxation of the red-blue set cover problem. To give some intuition, we turn to the integer program used for approximating the rooted group Steiner problem by Garg, Konjevod and Ravi [12].

For a set of vertices $S \subset V$, let $\delta(S)$ denote the set of all edges with exactly one endpoint in S.

$$
\begin{aligned}
&\min \sum_{e \in E} c_e x_e \\
&\sum_{e \in \delta(S)} x_e \geq 1 \quad \forall S \in r : \exists g \in \mathcal{G} \quad g \cap S = \emptyset \\
&x \in \{0,1\}^{|E|}.
\end{aligned}
\tag{2}
$$

4.1 Improved Formulations

IP (2) requires that each cut separating some group from the root vertex be covered by at least one edge. In view of the transformation from the group Steiner problem on a tree to the red-blue set cover, this has a clear interpretation. We need to cover each blue element g. A cut around the group corresponds to a set of red elements hitting all the sets that contain g.

To simplify notation, we write $\mathcal{S}_b$ for the family of all sets that contain b, and $\mathcal{H}_b$ for the family of all sets of red elements which form a hitting set for $\mathcal{S}_b$. Since one of the sets in $\mathcal{S}_b$ must be chosen,

we must pay for at least one red element from each hitting set for $\mathcal{S}_b$. Defining $\mathcal{H} = \bigcup_{b \in B} \mathcal{H}_b$ yields the following formulation.

$$\begin{aligned} &\min \sum_r y_r \\ &\sum_{r \in H} y_r \geq 1 \quad \forall H \in \mathcal{H} \\ &y \in \{0,1\}^{|R|}. \end{aligned} \tag{3}$$

Without variables corresponding to sets, we need to describe how to interpret a solution to IP (3), namely how to determine which sets in $\mathcal{S}$ should be included in the cover given a feasible 0-1 solution y^*. We include in the cover all those sets S such that $y^*_r = 1$ for all $r \in S$. To show that this indeed produces a feasible red-blue cover, suppose that some blue element b is left uncovered. That means that for every $S \ni b$ there is a $r_S \in S$ such that $y^*_{r_S} = 0$. But these red elements form a hitting set $H \in \mathcal{H}_b$ for $\mathcal{S}_b$. Since none of the terms in the inequality corresponding to H has value 1, this inequality is violated by the given solution y^*.

The linear relaxation of IP (3) is a hard problem itself. There are exponentially many constraints, and the separation oracle (see Grötschel, Lovász and Schrijver [14]) needs to be able to solve a hitting set problem. (The question that a separation oracle needs to answer is: Does there exist a blue element $b \in B$ and a hitting set for $\mathcal{S}_b$ whose cost, as defined by the given fractional solution y^*, is less than 1?) Since HITTING SET is SET COVER on the dual set-system, we can only provide an approximate separation oracle with a factor of $\log t$, where t is the maximum number of sets containing any pair of one red and one blue element (a simple upper bound on t is $n = |\mathcal{S}|$.) However, this allows us to find a feasible solution to the linear program which is at most $\log t$ times more expensive than the optimal one.

Lemma 4.1. *An α-approximate separation oracle $\mathcal{O}$ can be used to find in polynomial time an α-approximate solution to the linear relaxation of a 0-1 covering problem $Ax \geq 1$.*

Proof. Use $\mathcal{O}$ as if it were an exact separation oracle and apply the ellipsoid algorithm. The algorithm will finish when no more hitting sets of size less than 1 can be found. Let x^* be the resulting candidate solution to the covering problem. Let

$$x'_i = \min\{x^*_i \cdot \alpha, 1\}, \text{for all } i.$$

Then x' is a feasible solution to the covering linear program. (Otherwise, there exists an inequality whose left-hand side adds up to less than 1. Since no term in this inequality was rounded to 1, all of them were originally smaller than $1/\alpha$, and further, the sum of all of these terms was smaller than $1/\alpha$. However, if there existed a hitting set of cost less than $1/\alpha$, the approximation algorithm would have found a hitting set of size less than 1.) Finally, the cost of x' is at most α times the cost of x^*. □

We next describe an extension of IP (3). Instead of constraints based on hitting sets (covers), we can use multiple hitting sets (multicovers). For instance, we can form constraints corresponding to sets of red elements which contain at least two elements from each set in $\mathcal{S}_b$ for some b. We may set the right-hand side of such a constraint to 2, because every feasible solution must contain at least two red elements from such a set.

In the same way we can form k-hitting set constraints for a blue element b, where k is any positive integer no greater than the minimum number of red elements contained in a set that covers b. (In

fact, the minimum number of red elements contained in a set is not the critical obstacle to deriving valid inequalities. We defer the details to the full version of the paper.)

The separation algorithm this time needs to solve a multicover problem, that is find a minimum-cost family of sets such that each element of the ground set is covered at least k times by the family. Efficient approximation algorithms for multicover problems were described by Dobson [8] and by Rajagopalan and Vazirani [18]. The approximation guarantees achieved are $\log kn$ and $\log n$, respectively, where n is the number of elements in the ground set.

In order to write the constraints for this IP, we denote the family of all k-hitting sets by $\mathcal{H}_k$.

$$\begin{aligned} &\min \sum_r y_r \\ &\sum_{r\in H} y_r \geq k \quad \forall H \in \mathcal{H}_k \\ &y \in \{0,1\}^{|R|}. \end{aligned} \tag{4}$$

These inequalities are a strict superset of the inequalities used in IP (3). For example, let $S_1 = \{r_1, r_2, b\}$, $S_2 = \{r_2, r_3, b\}$, $S_3 = \{r_3, r_1, b\}$. Then IP (3) has 3 inequalities,

$$\begin{aligned} y_1 + y_2 \geq 1 \\ y_2 + y_3 \geq 1 \\ y_3 + y_1 \geq 1, \end{aligned}$$

and so the solution $y_1 = y_2 = y_3 = 1/2$ is feasible. However, IP (4) also includes the inequality

$$y_1 + y_2 + y_3 \geq 2,$$

because the set $R = \{r_1, r_2, r_3\}$ hits every cover for b twice.

However, in some special cases, such as group Steiner problem on trees, the two formulations are equivalent.

Lemma 4.2. *For an instance of the group Steiner problem on a tree, every inequality of IP (4) can be written as a convex combination of inequalities of IP (3).*

4.2 Hitting-set LP versus improved simple LP

We relate LP (1)′ and LP (3) by showing that every fractional solution y of LP (1)′ also satisfies all inequalities of LP (3). This implies that the improved simple LP (1)′ is a better formulation than the hitting set LP (3), despite its conciseness.

Let y be a fractional solution to LP (1)′. Let H be a hitting set of red elements for all sets containing a fixed $b \in B$. Then for every $r \in H$, $y_r \geq \sum_{S \supseteq \{r,b\}} x_S$. Thus,

$$\sum_{r\in H} y_r \geq \sum_{r\in H} \sum_{S\supseteq\{r,b\}} x_S \geq \sum_{S\ni b} x_S \geq 1.$$

The next-to-last inequality is satisfied because every set $S \ni b$ contains a red element in H and so the variable x_S appears in the double sum.

Even though we cannot prove any "good" bounds on the integrality gap of either of the two linear formulations, it is easily seen that LP (1)′ sometimes gives an objective function value at least a factor $\log n$ higher (thus, better) than the hitting set LP (3), unless NP$\subseteq$TIME($n^{O(\log\log n)}$). (As

before, $n = |\mathcal{S}|$.) For suppose that the values of the two LPs were always within a logarithmic factor of each other. Then, since LP (1)$'$ always has a higher objective value than LP (3), we could use the optimum value of LP (1)$'$ as an approximation to the optimal solution of LP (3). However, since this optimum is hard to approximate, there must be an instance where the two linear relaxations' optimal values differ by at least a logarithmic factor.

4.3 Multi-element inequalities and set cover

The inequalities described above are special cases of a class that completely specifies the optimal value of an instance of RED-BLUE SET COVER. Obviously, the inequalities we will now describe can only be of restricted use since they are hard to separate, but some of their special cases may be useful in practice.

Consider two blue elements, b_1 and b_2. We may change our problem by replacing b_1 and b_2 by a single blue element b_{12}. We say that b_{12} is covered by any set that contains both b_1 and b_2, and add a new set $S_{ij} = S_i \cup S_j$ for each pair of sets S_i and S_j such that $b_1 \in S_i$ and $b_2 \in S_j$. This change creates a new instance of the red-blue set cover. Since both b_1 and b_2 will be covered in any feasible solution to the original instance, multi-hitting inequalities for the new instance are valid for the original instance of RED-BLUE SET COVER.

We will describe these inequalities in more detail in the full paper, but let us consider here the last inequality generated by this procedure in the special case of a set cover problem, where, every set contains a unique red element. First, all blue elements are merged into one. Then, for every subfamily $\mathcal{C} \subseteq \mathcal{S}$ that covers all blue elements we add a new set $S_{\mathcal{C}}$. Let us denote the cost of the optimal set cover by z^*. The inequality produced has all the variables corresponding to red elements on its left-hand side, and since every cover contains at least z^* red elements, the right-hand side of the inequality is z^*. Hence this is a k-hitting set constraint where $k = z^*$. Thus, the cost of the optimal solution to this linear program is equal to the cost of the optimal integer solution of the set cover problem.

5 Approximation Algorithms.

We begin the section with some simple propositions that yield approximation algorithms for restricted cases of the RED-BLUE SET COVERproblem.

Proposition 5.1. *The following statements hold:*

(1.) The RED-BLUE SET COVER *problem when restricted to instances in which* $\forall i, |S_i| = 2$, *has a* $\log n$ *approximation algorithm.*

(2.) The RED-BLUE SET COVER *problem when* $|S_i \cap R| \leq k$ *has a* $O(n^{1-1/k} \log n)$*-approximation algorithm.*

Proof. Proof of Part (1) follows by direct arguments from SET COVER. We sketch proof of Part (2) for $k = 2$. First by preprocessing we ensure that $\forall i, S_i \cap R$ are not identical. Then, for each $S_i = \{r_i^1, r_i^2, b_i^1, \ldots, b_i^{j_i}\}$, we treat as if $\{r_i^1, r_i^2\}$ covers $\{b_i^1, \ldots, b_i^{j_i}\}$ and perform a greedy set cover. There is a loss of $\lg n$ factor from the optimum. The optimum, however, can presumably do much better by exploiting the overlap between r_i's. Letting $\mathcal{K}$ denote the optimal value of the modified instance, we know that the standard greedy heuristic for set cover will out a solution of value no more than $\mathcal{K} \lg n$. We claim that the optimum for the original RED-BLUE SET COVER instance is $\geq \sqrt{\mathcal{K}}$. This is because the maximum overlap it can achieve (because of the preprocessing) is precisely $\mathcal{K}'$ s.t. $\binom{\mathcal{K}'}{2} = \mathcal{K}$, which makes $\mathcal{K}' = O(\sqrt{\mathcal{K}})$. This implies a $O(\sqrt{n} \lg n)$-approximation algorithm. $\square$

Next we give a polynomial time approximation algorithm for the restriction of RED-BLUE SET COVER where each set contains only one blue element, with guarantee $2\sqrt{n}$. The approximation algorithm is based on the linear relaxation of IP (1)′ discussed earlier. We denote by n the number of sets $S \in \mathcal{S}$. Since every instance of RED-BLUE SET COVER may be reduced to one where every set contains only one blue element, the algorithm is applicable to any instance of RED-BLUE SET COVER. However, since the reduction (Section 1) may increase the number of sets, the performance of our algorithm is $2\sqrt{kn}$ (where $|S_i \cap B| \leq k$); and hence in general is not $o(n)$.

The algorithm first solves the linear relaxation of IP (1)′, producing the optimal solution (x^*, y^*). Let us call the blue elements $b \in B$ that appear in more than $\sqrt{n}$ sets $S \in \mathcal{S}$, *bad*, and all the other blue elements, *good*.

The algorithm proceeds in two phases. In the first phase, we multiply y^* by $\sqrt{n}$, and then round down to 0 or 1. Denote the resulting vector of red-element values by y' We include in the cover all the sets S such that $y'_r = 1$ for all $r \in S$. The cost of this step is at most $\sqrt{n}$ times the cost of the optimal red-blue cover.

We claim that the first phase covers all the blue elements. Fix a $b \in B$ and consider a hitting set H for the sets containing b. Since b is contained in fewer than $\sqrt{n}$ sets, at least one of the variables y_r, $r \in H$ must have value at least $1/\sqrt{n}$. This variable will be rounded to 1, hence the rounded y' will satisfy this inequality. Given a 0-1 solution to LP (3), we have seen how to construct a cover of no larger cost, so we apply that procedure to cover the good elements. Since each set $S \in \mathcal{S}$ contains only one blue element, there can be no more than $\sqrt{n}$ bad blue elements.

In the second phase, we include in the cover the set of smallest cost that covers b, for every bad blue element b. Since every blue element is covered by the optimal solution, every set added to the cover in the second phase costs no more than the optimal solution, and thus the cost of the second phase is no more than $\sqrt{n}$.

Our algorithm does not provide an upper bound on the integrality gap of any of the linear relaxations we have proposed so far. However, it is easy to amend LP (1)′ to provide such a bound.

We add to IP (1)′ another inequality for each $b \in B$. Let $\mu_b = \min_{S \ni b} |S \cap R|$. Then the inequality

$$\sum_{r \in R} y_r \geq \mu_b$$

is valid for all $b \in B$. Denote the resulting improvement of IP (1)′ by IP (1)″. These added inequalities guarantee that the second phase of the algorithm costs no more than $\sqrt{n}$ times the optimal fractional solution, and thus the integrality gap of IP (1)″ is no more than $2\sqrt{n}$. We thus have

Theorem 5.2. *The* RED-BLUE SET COVER *problem restricted to instances where each set contains only one blue element, ($|S_i \cap B| = 1$) can be approximated within a factor of* $2\sqrt{n}$.

6 Practical motivation

We conclude the paper with a few practical motivations for the RED-BLUE SET COVER problem and more generally its relation to classification problems.

6.1 Fraud and Anomaly Detection

Our original motivation for considering RED-BLUE SET COVER arose in the context of a data mining project. The goal of the project was to detect possible fraud/anomaly in Medicare/Medicaid data (claims). The project was undertaken at the Los Alamos National Laboratory and was sponsored by the HCFA (Health Care and Finance Agency) and the New York State University research Foundation

[1, 2]. More details about this and related anomaly detection projects at the laboratory can be found at `http://www.c3.lanl.gov`. There is a large data base of records (claims). Each record[4] is a vector in feature space, with real/Boolean values correspond to individual features. The goal is to use the information that is available from known fraudulent records to identify possible anomaly in records for which no apriori information is available. Automating this process is complicated by the fact that misclassifying a good record (false positives) often amounts to possible law suits and other legal problems and thus needs to be avoided at all costs. Using standard terminology from classification literature, we wish to build a model for determining the validity of a claim, that uses known information. As training data we have a small subset of claims that are labeled as valid (red) claims or fraudulent (blue) claims[5]. A number of classification methods were used to design classification of the red and the blue points. These included Cluster analysis, classification and regression tree (CART), logistic regression. Each of the classification scheme yields subsets containing both red and blue elements. The goal is to design a new classifier that is complete (meaning that all blue points are covered) and minimally inconsistent (i.e. misclassifies minimum number of red elements).

6.2 Information Retrieval

This example is a slight variation of the example discussed in [11]. A typical problem in this area consists if classifying a large corpus of documents. Such a classification might be related to grouping articles found during a web search (e.g. classifying news articles based on subject topic). We view each word in our vocabulary as a feature and represent an article as a 0-1 vector in this feature space. Typically, the number of features is huge (on the order of 10^5) and it is intuitively clear that most documents have only a small number of relevant features. Thus a natural way to overcome the computational bottleneck of classifying in such a large feature space is to combine the results of algorithms that classify documents based on very small subsets of feature space. In this example, we have elements of different colors (corresponding to the subjects) and our goal is construct a combined classifier.

6.3 Relationship to General Classification and Learning

RED-BLUE SET COVER can be thought of as a restricted form of a more general machine learning/classification problem [19]. In this setting sets R and B can be regarded as a set of elements, each of which has an m-dimensional vector of Boolean values. Component i of the vector for element x equals true iff $x \in S_i$. Component i can be considered a Boolean variable x_i. The goal is to construct a Boolean function $\mathcal{H}$ of m variables, so that $\mathcal{H}$ applied to the vector for each blue element is true, and the number of red elements whose vector maps to true is minimized. What makes the problem nontrivial is that the space of Boolean functions under consideration is very limited. So, let G be a collection of m-variable Boolean functions. Let us say that a given Boolean function $\mathcal{H}$ *covers* an element $x \in R \cup B$ iff $\mathcal{H}$ applied to the vector for element x is true. The G-Boolean-Classification Problem is to find a function $\mathcal{H} \in G$ that satisfies the constraint that $\mathcal{H}$ covers all blue elements, and the number of red elements covered is minimized.

RED-BLUE SET COVER can also be viewed as a problem of classifying (or predicting) by combining the classification results (or predictions) of subordinate classification algorithms also referred to as "experts" in the learning theory literature [11, 5].

[4]Each record is a line item claim

[5]Such a data is obtained in a variety of ways including past history, sample checks, etc.

7 Acknowledgments

We thank Kevin Buescher for explaining the particular application that led to the formulation of this problem. We thank Riko Jacob, Uri Feige, Sven Krumke, Ravi Kumar, David Peleg, Dan Rosenkrantz, R. Ravi, Santosh Vempala and Hans-Christoph Wirth for several useful and stimulating discussions during all phases of our research.

References

[1] Advanced methods for medicare fraud waste and abuse detection project. Los Alamos National Laboratory, Internal Report, June 1998. Prepared for the Health care and Finance Agency as Phase III report.

[2] NYS-LANL medicaid FWA detection project scoping paper detection project. Los Alamos National Laboratory, Internal Report, Dec 1997.

[3] M. Alekhnovich, S. Buss, S. Moran, and T. Pitassi. Minimum propositional proof length is np-hard to linearly approximate. manuscript, 1998.

[4] S. Arora and C. Lund. Hardness of approximations. In *Approximation algorithms for NP-hard problems*, pages 399–446. PWS, 1995.

[5] N. Cesa-Bianci, Y. Freund, D. Helmbold, D. Haussler, R. Schapire, and M. Warmuth. How to use expert advice. In *Proceedings of the 25thAnnual ACM Symposium on Theory of Computing*, pages 382–391, 1993.

[6] M. Charikar, C. Chekuri, T. yat Cheung, Z. Dai, A. Goel, S. Guha, and M. Li. Approximation algorithms for directed steiner problems. In *Proceedings of the 9thAnnual ACM-SIAM Symposium on Discrete Algorithms*, pages 192–200, 1998.

[7] I. Dinur and S. Safra. On the hardness of approximating label cover. ECCC Report 15, 1999.

[8] G. Dobson. Worst-case analysis of greedy heuristics for integer programming with nonnegative data. *Math. Oper. Res.*, 7:515–531, 1982.

[9] Y. Dodis and S. Khanna. Designing networks with bounded pairwise distance. In *Proceedings of the 31stAnnual ACM Symposium on Theory of Computing*, pages 750–759, 1999.

[10] U. Feige. private communication.

[11] Y. Freund, R. Schapire, Y. Singer, and M. Warmuth. Using and combining predictors that specialize. In *Proceedings of the 29thAnnual ACM Symposium on Theory of Computing*, pages 334–344, 1999.

[12] N. Garg, G. Konjevod, and R. Ravi. A polylogarithmic approximation algorithm for the group steiner tree problem. In *Proceedings of the 9thAnnual ACM-SIAM Symposium on Discrete Algorithms*, pages 253–259, 1998.

[13] M. Goldwasser and R. Motwani. Intractability of assembly sequencing: unit disks in the plane. In *Algorithms and Data Structures, 5th International Workshop*, Lecture Notes in Computer Science, 1997.

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

[15] R. Jacob, G. Konjevod, S. Krumke, M. Marathe, R. Ravi, and H. Wirth. The minimum label path problem. Unpublished manuscript, Los Alamos National Laboratory, 1999.

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

[17] S. O. Krumke and H.-C. Wirth. On the minimum label spanning tree problem. *Information Processing Letters*, 66(2):81–85, 1998.

[18] S. Rajagopalan and V. Vazirani. Primal-dual RNC approximation algorithms for set covering and covering integer programs. *SIAM J. Comput.*, 28(2):525–540, 1998.

[19] D. Rosenkrantz. private communication.