DIMACS Technical Report 2007-14
October 2007

On Approximating Four Covering/Packing Problems With Applications to Bioinformatics

by

Mary Ashley[1]
Department of Biological Sciences
University of Illinois at Chicago
Chicago, IL 60607-7053
Email: ashley@uic.edu

Tanya Berger-Wolf[1]
Department of Computer Science
University of Illinois at Chicago
Chicago, IL 60607-7053
Email: tanyabw@cs.uic.edu

Piotr Berman[2]
Department of Computer Science & Engineering
Pennsylvania State University
University Park, PA 16802
Email: berman@cse.psu.edu

Wanpracha Chaovalitwongse[1]
Department of Industrial Engineering
Rutgers University
New Brunswick, NJ 08854
Email: wchaoval@rci.rutgers.edu

Bhaskar DasGupta[3]
Department of Computer Science
University of Illinois at Chicago
Chicago, IL 60607-7053
Email: dasgupta@cs.uic.edu

Ming-Yang Kao[4]
Department of Electrical Engineering & Computer Science
Northwestern University
Evanston, IL 60208
Email: kao@cs.northwestern.edu

[1]Supported by NSF grant IIS-0612044.
[2]Supported by NSF grant CCR-0208821.
[3]Supported by NSF grants DBI-0543365, IIS-0612044 and IIS-0346973.
[4]Supported in part by NSF grant EIA-0112934.

DIMACS is a collaborative project of Rutgers University, Princeton University, AT&T Labs–Research, Bell Labs, NEC Laboratories America and Telcordia Technologies, as well as affiliate members Avaya Labs, HP Labs, IBM Research, Microsoft Research, Stevens Institute of Technology, Georgia Institute of Technology and Rensselaer Polytechnic Institute. DIMACS was founded as an NSF Science and Technology Center.

ABSTRACT

In this paper, we consider approximability of four covering/packing type problems which have important applications in computational biology. The problems considered in this paper are the *triangle packing problem*, the *full sibling reconstruction problem* under two parsimonious assumptions, the *maximum profit coverage problem* and the 2-*coverage problem*. We provide approximation algorithms and inapproximability results for various values of parameters of interest for these problems. Our inapproximability constant for the triangle packing problem improves slightly upon the best-known inapproximability constant that can be achieved from previous results [14]; this is done by directly transforming the inapproximability gap of Håstad for the problem of maximizing the number of satisfied equations for a set of equations over GF(2) [26] and is interesting in its own right. Our inapproximability results on full siblings reconstruction problems answers open questions about the computational complexities of these problems posed by Berger-Wolf *et al.* [5]. Our results on the maximum profit coverage problem provides almost matching upper and lower bounds on the approximation ratios for this problem posed by Hassin and Or [25].

1 Introduction

We consider four problems motivated by four different applications in bioinformatics. Each of them concerns with packing or covering. We start with the precise definitions of the problems and later describe their motivations.

Triangle Packing Problem (TP)

In TP we are given an undirected graph. A triangle is a cycle of 3 nodes. The goal is to find (pack) a maximum number of *node-disjoint* triangles in G.

Full Sibling Reconstruction Problems, 4-ALLELE$_{n,\ell}$ and 2-ALLELE$_{n,\ell}$

We consider the problem of reconstructing sibling relationships based on a genetic sample of individuals from the same generation. Each individual is represented by a string of ℓ pairs of symbols (each symbol represents a unique DNA string, or an allele). The objective is to reconstruct possible groups of siblings, *i.e.* groups of individuals whose genetics are consistent with having the same parents according to Mendelian laws of inheritance. Below is the formal definition of two versions of the problem.

Definition 1
(a) *An ℓ-locus individual (genetic sample) is a sequence $S = ((S_{1,1}, S_{1,2}), (S_{2,1}, S_{2,2}), \ldots, (S_{\ell,1}, S_{\ell,2})]$ of ordered pairs of symbols over an alphabet.*
(b) *A set $\mathcal{A}$ of ℓ-locus individuals satisfies the 4-allele condition [4] at a locus j if and only if $|\cup_{T\in\mathcal{A}} \{T_{j,1}, T_{j,2}\}| \leq 4$.*
(c) *A set $\mathcal{A}$ of ℓ-locus individuals satisfies 2-allele condition [4] at a locus j if and only if each $A \in \mathcal{A}$ has a permutation σ_A of $\{1, 2\}$ such that $|\cup_{A\in\mathcal{A}} \{A_{j,\sigma_A(k)}\}| \leq 2$ for $k = 1$ and $k = 2$.*
(d) *A set $\mathcal{A}$ of ℓ-locus individuals is full siblings under the k-allele condition ($k \in \{2, 4\}$) if and only for it satisfies the k-allele condition at* every *locus.*

Note that any set of two individuals is always a full sibling set under either condition. We now state the full sibling reconstruction problem based on the two parsimonious constrains as outlined in [4, 5, 15] We can simultaneously state two versions of the problem, with $k = 2$ and $k = 4$:

> **Problem name:** k-allele problem with parameter ℓ (k-ALLELE$_{n,\ell}$) with $k \in \{2, 4\}$.
>
> **Instance:** a set $\mathcal{A}$ of n ℓ-locus individuals.
>
> **Valid Solutions:** a collection $\mathcal{S}$ of sets of individuals (sibling groups) such that
>
> - $\mathcal{A} = \cup_{\mathcal{B}\in\mathcal{S}}\mathcal{B}$ ($\mathcal{S}$ covers $\mathcal{A}$);

- each $\mathcal{B} \in \mathcal{S}$ is full sibling set under the k-allele condition;

Objective: *minimize* $|\mathcal{S}|$.

Although various full sibling reconstruction methods have been investigated previously experimentally, there were no theoretical investigation of the computational complexities of these problems. A natural parameter of interest is the maximum size of any full sibling set a; we denote the corresponding problem by a-k-ALLELE$_{n,\ell}$ in the discussion below. Both 2-4-ALLELE$_{n,\ell}$ and 2-2-ALLELE$_{n,\ell}$ are trivial. Furthermore, if a is a constant, both a-4-ALLELE$_{n,\ell}$ and a-2-ALLELE$_{n,\ell}$ can be posed as a set-cover problem with a polynomially many sets with the maximum set size being a and thus have a $(1 + \ln a)$-approximations (by using standard algorithms for the set-cover problem [38]). For general a, since any two individuals can be put in the same sibling group, both a-4-ALLELE$_{n,\ell}$ and a-2-ALLELE$_{n,\ell}$ have a trivial $\frac{a}{2}$-approximation.

Maximum Profit Coverage Problem (MPC) [25]

We have family of m sets $\mathcal{S}$ over a universe $\mathcal{U}$ of n elements. For each $A \in \mathcal{S}$ we have a non-negative *cost* q_A and for each $i \in \mathcal{U}$ we have a non-negative *profit* w_i. For $\mathcal{P} \subset \mathcal{S}$ we define the profit $c(\mathcal{P}) = \sum_{i \in \cup_{S \in \mathcal{P}} S} w_i - \sum_{A \in \mathcal{P}} q_A$. The goal is to find a subcollection of sets $\mathcal{P}$ that maximizes $c(\mathcal{P})$. A natural parameter for this problem is $a = \max_{A \in \mathcal{S}} |A|$. MPC admits a PTAS in Euclidean space but otherwise its complexity was unknown.

2-Coverage Problem

Given $\mathcal{S}$ and $\mathcal{U}$ as in the MPC problem above and an integer $k > 0$, a valid solution is $\mathcal{P} \subset \mathcal{S}$ such that $|\mathcal{P}| \leq k$; the goal is to *maximize* the number of elements that occur in *at least two* of the sets from $\mathcal{P}$. Another natural parameter of interest here is the *frequency* f, *i.e.*, the maximum number of times any element occurs in various sets.

1.1 Motivation

In this section we discuss the motivations for the problems considered in this paper. We discuss one motivation in details and mention the remaining ones very briefly.

For wild populations, the growing development and application of molecular markers provides new possibilities for the investigation of many fundamental biological phenomena, including mating systems, selection and adaptation, kin selection, and dispersal patterns. The power and potential of the genotypic information obtained in these studies often rests in our ability to reconstruct genealogical relationships among individuals [21]. These relationships include parentage, full and half-sibships, and higher order aspects of pedigrees [12, 13, 29]. In this paper we are only concerned with full sibling relationships from single generation sample of microsatellite markers

Several methods for sibling reconstruction from microsatellite data have been proposed [1, 2, 11, 34–37, 39]; Most of the currently available methods use statistical likelihood models and are inappropriate for wild populations. Recently, a fully combinatorial approach [4, 5, 15] to sibling reconstruction has been introduced. Our approach uses the simple Mendelian inheritance rules to impose constraints on the genetic content possibilities of a sibling group. A formulation of the inferred combinatorial constraints under the parsimony assumption of constructing the smallest number of groups of individuals that satisfy these constraints leads to the full sibling problems discussed in the paper. More specifically, an individual is represented by an ordered sequence of ℓ positions, called *loci*, say $p = (p_1, \ldots, p_\ell)$, where p_i is multiset of two elements (alleles). An individual p can be a child of a pair of *parents*, say q and r, if for each $i \in \{1, \ldots, \ell\}$ we have $p_i \subseteq q_i \cup r_i$ and $p_i \cap s_i \neq \emptyset$ for $s \in \{r, q\}$; a set S of individuals are full siblings if one can construct a pair of parents, say q and r, such that every $p \in S$ can be a child of q and r. Both the 4-allele and the 2-allele constraints encode the above biological conditions for full siblings with varying strictness. In this paper we study of worst-case computational complexity issues of these approaches together with additional experimental results that *did not appear in* [5].

MPC has applications in clustering identification of molecules [25]. The 2-coverage problem has motivations in optimizing multiple spaced seeds for homology search (for relevant concepts, see *e.g.* [40]). For application of TP to genome rearrangement problems, see [3, 14].

1.2 Some Problems Useful for Reductions

Several additional known problems will be used for (in)approximability results. Below we list many of these problems together with the known relevant results. Recall that a a $(1+\varepsilon)$-*approximate solution* (or simply an $(1+\varepsilon)$-approximation) of a minimization (resp. maximization) problem is a solution with an objective value no larger (resp. no smaller) than $1+\varepsilon$ times (resp. $(1+\varepsilon)^{-1}$ times) the value of the optimum, and an algorithm achieving such a solution is said to have an *approximation ratio* of at most $1+\varepsilon$. A problem is r-inapproximable under a certain complexity-theoretic assumption means that the the problem does not have a r-approximation unless the complexity-theoretic assumption is false.

3-LIN-2 We are given a set of linear equations modulo 2 with 3 variables per equation. Our goal is to maximize the number of equations that are satisfied with a certain value assignment to the variables. A well-known result by Håstad [26] shows the following result: for every $\varepsilon < \frac{1}{2}$ it is NP-hard to differentiate between the instances that have at least $(1-\varepsilon)m$ satisfied equations from those that have at most $\left(\frac{1}{2}+\varepsilon\right)m$ satisfied equations.

MAX-CUT on a 3-regular graph (3-MAX-CUT) An instance is a 3-regular graph, *i.e.*, a graph $G = (V, E)$ where the degree of every vertex is exactly 3 (and thus $|E| = \frac{3}{2}|V|$). The goal is to find $V' \subseteq V$ such that $score(V') = |\{u, v\} | \{u, v\} \in E$ **and** $|\{u, v\} \cap V'| = 1\}|$ is maximized. We will need the following inapproximability result for this problem proved

in [8]. For every constant $\varepsilon > 0$, it is NP-hard to decide whether an instance G of 3-MAX-CUT with $|V| = 336n$ vertices has a valid solution with a score below $(331 - \varepsilon)n$ or above $(332 + \varepsilon)n$.

Independent set problem for a a-regular graph A set of vertices are called independent if no two of them are connected by an edge. The goal is to find an independent set of maximum cardinality when the input graph is a-regular, *i.e.*, every vertex has degree a. It is well-known that this problem is NP-hard for $a \geq 3$ and $\Omega\left(\frac{a}{\ln a}\right)$-inapproximable for general a assuming P≠NP [27].

Graph Coloring The goal is to produce an assignment of colors to vertices of a given graph $G = (V, E)$ such that no two adjacent vertices have the same color and the number of colors is *minimized*. Let $\Delta^*(G)$ denote the *maximum* number of independent vertices in a graph G and $\chi^*(G)$ denote the minimum number of colors in a coloring of G. The following inapproximability result is a straightforward extension of a hardness result known for coloring of G [19]: for any two constants $0 < \varepsilon < \delta < 1$, $\chi^*(G)$ cannot be approximated to within a factor of $|V|^{\varepsilon}$ even if the $\Delta^*(G) \leq |V|^{\delta}$ unless NP⊆ZPP.

Weighted set-packing We have a collection of sets each with a non-negative weight over an universe. Our goal is to select a collection of mutually disjoint sets of total maximum weight.

Densest Subgraph problem (DS) We are given a graph $G = (V, E)$ and a positive integer $0 < k < |V|$. The goal is to pick k vertices such that the subgraph induced by these vertices has the maximum average degree. The densest subgraph problem is $(1 + \varepsilon)$-inapproximable for some constant $\varepsilon > 0$ unless NP⊆BPTIME($2^{n^{\varepsilon}}$) [31]. A more general weighted version of DS admits a $O(m^{\frac{1}{3}-\varepsilon})$-approximation for some constant $\varepsilon > 0$ [20].

Maximum coverage problem This is the same as the 2-coverage problem except that every the number of elements that occur in at least *one* of the selected sets is *maximized*. It is known that the maximum coverage problem can be approximated to within a ratio of $1 - \left(1 - \frac{1}{k}\right)^k > 1 - (1/\mathbf{e})$ either by a greedy algorithm [32] or by LP-rounding [40] and approximation with ratio better than $1-(1/\mathbf{e})$ is not possible unless $P = NP$ [18]. Obviously, the same lower bound carries over to 2-coverage also *for arbitrary* f.

1.3 Our Results and Techniques

1.3.1 Triangle Packing (TP)

We show that TP is $\frac{154}{153} \approx 1.0065$-inapproximable assuming P$\neq NP$. This is done by a careful reduction from 3-LIN-2 that *roughly* shows that it is NP-hard to distinguish between

instances of TP that has a cost of at most $152k$ as opposed to a cost of at least $153k$ for every k.

The reduction is described in Section 2, but here we make the following two relevant comments regarding the reduction (see Section 2 for more details):

- One can have a somewhat larger construction but with the extra property that the resulting graph is 4-regular. In [9] it is shown that it is NP-hard to differentiate between 3-regular graphs that have $200n$ nodes and a maximum independent set has either (a) at least $(98-\varepsilon)n$ nodes, or (b) at most $(97-\varepsilon n)$ nodes. We can replace those graphs with their line-dual, and we will have graphs of $300n$ nodes in which we can find at least about $98n$ or at most about $97n$ triangles, hence with cost of at most about $101n$ or at least about $101.5n$, which gives a worse $\frac{204}{203}$-inapproximability. A proof of Caprara and Rizzi [14] is yet earlier and it implies a still worse inapproximability constant.

- Our construction can be improved using smaller amplifiers of Chlebík and Chlebíková [17]. More significant improvement is also possible, because amplifier property is stronger than necessary in this context and smaller graphs can be used. Back-of-the-envelope estimate would give $\frac{131}{130}$-inapproximability.

1.3.2 4-ALLELE$_{n,\ell}$ and 2-ALLELE$_{n,\ell}$

$a = 3$ This is the *smallest* non-trivial value of a. We prove $\frac{305}{304} \approx 1.0032$-inapproximability for both problems when $\ell = O(n^3)$ assuming P$\neq NP$. The reduction is from triangle packing to a generic version that covers both problems for this case; loci are introduced carefully via a "labeling" procedure to ensure a faithful reduction. Also, either problem (for any *ell*) has an easy $\left(\frac{7}{6}+\varepsilon\right)$-approximation for any constant $\varepsilon > 0$ using the results of Hurkens and Schrijver [28].

$a = 4$ We prove $\frac{6725}{6724} \approx 1.00014$-inapproximability for both problems *even with the restriction of* $\ell = 2$ assuming P$\neq$NP. The combination $a = 4$ and $\ell = 2$ represent the *second smallest* non-trivial values of them. The hardness result is obtained by reducing 3-MAX-CUT via an intermediate geometric mapping; the idea of this proof and that for the case of $a = 6$ described next are inspired by some gadgets used in [7, 8]. Also, either problem has an easy $\left(\frac{3}{2}+\varepsilon\right)$-approximation for any constant $\varepsilon > 0$ using the result of Berman and Krysta [10].

$a = 6$ **and** $\ell = O(n)$ This combination of a and ℓ represents a slightly higher value of a and a *moderate* value of ℓ. We prove $\frac{1182}{1181} \approx 1.00084$-inapproximability for 4-ALLELE$_{n,\ell}$ only *even with the restriction of* $\ell = O(n)$ assuming P$\neq$NP. The inapproximability constant is better than the correspoding one for $a = 4$ and $\ell = 2$ but worse than the corresponding one in $a = 3$ and $\ell = O(n^3)$. The result is obtained by reducing 3-MAX-CUT via a (different) intermediate geometric mapping.

$a = n^{\delta}$ This case represents *all sufficiently large* values of a. We prove n^{ε}-inapproximability for both problems assuming NP$\not\subseteq$ZPP. Here $0 < \varepsilon < \delta < 1$ are any two constants. We reduce the hard instance of the graph coloring problem as described in the previous section.

In general, additional loci are used carefully in many of the above reductions to rule out possibilities that would violate the validity of our reductions.

1.4 Maximum Profit Coverage (MPC)

For MPC we prove almost matching lower and upper bounds on approximability:

(i) MPC is NP-hard for $a \geq 3$ and $\Omega\left(\frac{a}{\ln a}\right)$-inapproximable for arbitrary a assuming P$\neq$NP even if every set has weight $a-1$, every element has weight 1 and every set contains exactly a elements. The hard instances can further be restricted such that each element is a point in some underlying metric space and each set correspond to a ball of radius α for some fixed specified α. The reduction is from the independent set problem on a-regular graphs.

(ii) MPC is polynomial-time solvable for $a \leq 2$. Otherwise, we provide, for any constant $\varepsilon > 0$, a $\left(\frac{a+1}{2} + \varepsilon\right)$-approximation for fixed a and a $(0.6454a + \varepsilon)$-approximation otherwise via the weighted set-packing problem. The $(0.6454a + \varepsilon)$-approximation for arbitrary a can be achieved via a very careful polynomial-time dynamic programming implementation of the 2-IMP approach in Berman and Krysta [10] that implicitly maintains subsets for possible candidates for improvement that cannot be explicitly enumerated due to their non-polynomial number.

1.4.1 2-coverage

For the 2-coverage problem:

- We observe that, for $f = 2$, the 2-coverage problem is $(1 + \varepsilon)$-inapproximable for some constant $\varepsilon > 0$ unless NP$\subseteq$BPTIME$(2^{n^{\varepsilon}})$ (the $1 - (1/\mathbf{e})$-inapproximability result for maximum coverage does not hold under the assumption of $f = 2$) and admits a $O(m^{\frac{1}{3}-\varepsilon'})$-approximation for some constant $\varepsilon' > 0$ by identifying with the DS problem.
- For arbitrary f, we show a $O(\sqrt{m})$-approximation.

Note that a significantly better than $O(\sqrt[3]{m})$-approximation for 2-coverage would imply a better approximation for DS than what is currently known.

2 Inapproximability Result for Triangle Packing

Note that the above theorem gives a $\frac{154}{153} \approx 1.0065$-inapproximability for TP.

Theorem 2 *For every constant $\varepsilon > 0$, it is NP-hard to decide whether an instance G of TP with $|V| = 228k$ vertices has a valid solution with a maximum number of triangles below $(76 + \varepsilon)k$ or above $(76.5 - \varepsilon)k$.*

Proof. Inapproximability bound for a maximization problem has the following form: for every $\varepsilon < \frac{1}{2}k^{-1}$ it is NP-hard to differentiate between the instances that have profit at least $(1 - \varepsilon)m$ from those that have a profit at most $(1 - k^{-1} + \varepsilon)m$; in the case of minimization problems we switch the profit with cost. As stated before, a well-known result by Håstad [26] shows that the the 3-LIN-2 problem has inapproximability parameter $k = 2$.

Berman and Karpinski [9] described a way of reducing the number of occurrences of variables in equations: replace occurrences of a variable, say x, with separate *original* variables, say, from a set V_x with m elements, add $6m$ additional *checker* variables, say, set C_x, and connect $V_x \cup C_x$ with equations of the form $x' + x'' = 1 \mod 2$, for $\{x', x''\} \in E_x$.

The graph $(V_x \cup C_c, E_x)$ is called *amplifier*, and it has the following properties: it is bipartite, each terminal has 2 neighbors and each checker has 3 neighbors, and every set of nodes A has a cut of size at least $\min(|A \cap V_x|, |A - V_x|)$.

The amplifier property allows to *normalize* assignments of values to $V_x \cup C_x$. As the graph is bipartite, with parts A_0 and A_1, we say that assigning value b to $x' \in A_i$ corresponds to assigning $b + 1 \mod 2$ to x. Let B_b be the set of variables in $V_x \cup C_x$ with assignment that corresponds to assigning b to x, and suppose that B_0 has more elements of V_x. In this case we modify the assignment such that it corresponds to assigning 0 to x; some ℓ terminal variable will change values, which changes the satisfaction of ℓ original equations. On the other hand, we have at least ℓ equations that correspond to the cut between B_0 and B_1 and we will gain satisfaction of those, hence the normalization cannot decrease the number of satisfied variables.

A technical detail is that when we may need to modify the original equations when we use *original* variables that correspond to negations of the actual value that we assign.

As a result, if we started with $2n$ equations, we had $6n$ variable occurrences, we created amplifiers with $60n$ edges/equations and all these equations are satisfied in the normal solutions, so now it is hard to tell if we can satisfy at least $(60 + 2 - \varepsilon)n$ equations or at most $(60 + 1 + \varepsilon)n$ equations, which means that the restricted version of 3-LIN-2 has inapproximability parameter of $k = 62$.

We can adapt this construction to show inapproximability of the problem of minimum triangle cover, or TCP (triangle cover with pairs): given a graph, we can cover it with sets that are either (a) triangles, or (b) have at most two nodes. We minimize the number of sets in the cover.

Suppose that we have an instance with n nodes, then a solution with $n/3 - a$ triangles has the cost of $\lceil n/3 + a/2 \rceil$. (We will skip the rounding later). We can also say that if we fail to cover a nodes with triangles, then the cost is $n/3 + a/6$. Inapproximability remains the same if we multiply the cost by 3, to be $n + a/2$ (we can talk about *small cost* and *large cost*.

We formulate an equivalent slightly more general version of the problem, TCPD: besides normal nodes we have *don't care nodes*, and if we fail to cover a normal nodes the (large)

cost is is $n+a/2$. To apparently more general version can be reduced to the ordinary version as follows: create three copies of the graph, and connect with triangles copies of each don't care node. It is easy to see that every triangle is either contained in a copy or it connects copies of the same don't care node.

A solution in the new graph can be normalized as follows. Consider a copy that has the minimal number of uncovered nodes, say a. Then the entire graph had at least $3a$ uncovered nodes and the (small) cost is at least $n+a/2$. If we restrict the solution to triangles contained in a single copy, we have at most a normal nodes uncovered, so the (large) cost is $n + a/2$.

Now we reduce 3-LIN-2 to TCPD. A variable with m occurrences is replaced with an amplifier with m terminals and $6m$ checkers, and in turn, we each node of the amplifier with a triangle, and an edge by identifying nodes of two triangles. One can see that such a graph has $11m$ nodes, of which $10m$ correspond to edges within the amplifier and m correspond to *connections* between the terminal nodes and the (original) equations with 3 variables. (We assume, without loss of generality, that m is even).

We will identify our solution with assigning 0/1 values to variable: when we use a triangle, it corresponds to value 1, and when we do not use, it corresponds to value 0. A normalized solution covers all the nodes in such a structure with the exception of the *connection nodes* that correspond to copies with value 0 assigned. If there is a minority of connection nodes that corresponds to a different value of the original variable than the majority, say, with a elements, then we have a cut of a edges of the amplifier that correspond to a nodes that are not covered with triangles and are not connection nodes. The normalization covers these at least a nodes and changes the status a connections.

What we need is a gadget for an equation with 3 variables that has the following property: when the number of the covered connections is correct (0 or 2 in case "$= 0 \mod 2$", 1 or 3 in case "$= 1 \mod 2$") then we can cover all normal nodes, otherwise we can cover all but one normal node. In this case the number of unsatisfied equation corresponds exactly to the number of uncovered nodes. Moreover, if normalize value on a connection, we loose at most one covered normal node, so normalization can be performed without a loss. Fig. 1 shows that for the case of "$= 1 \mod 2$" the gadget uses 6 nodes in addition to the connection nodes, and in the other case, 4 nodes. We can always toggle all the equations (and variable values) and have the same number of them satisfied, so we can make sure that we have no more larger gadgets than the smaller gadgets.

Summarizing, we start with $2n$ equations, of which we must either leave $(1-\varepsilon)n$ unsatisfied, or only εn, and we create a graph with $11 \times 3 \times 2n + 5 \times 2n = 76n$ nodes,of which we must either leave $(1-\varepsilon)n$ uncovered by triangles, or only εn, leading to the cost of $(76.5-\varepsilon)n$ or $(76+\varepsilon)n$, which gives inapproximability parameter of $k = 153$. ❑

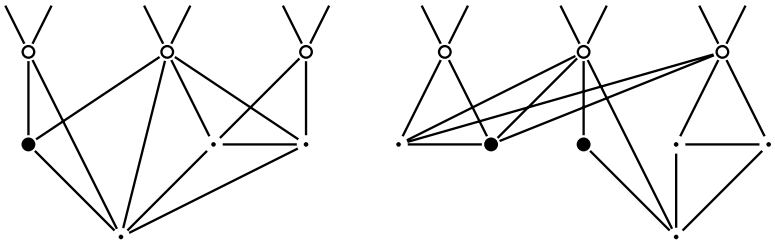


Figure 1: Equation gadgets. Connection nodes are circles, don't care node are large black nodes, normal nodes are small dots. The left gadget is for equation $x + y + z = 1 \mod 2$. If three connection nodes are covered, we can cover normal nodes with a triangle, if 1 is covered, we can cover 2 connections, 1 don't care and 3 normal with 2 triangles. Now, the cases when the equation is not satisfied. We can "pretend" that one more connection is covered, then we cover the rest, leaving 1 node uncovered. However, we cannot cover all nodes. When 0 connections are covered, we cannot cover more than one of them using triangles that do not include don't care node, and if we use on of these, we have to cover 7 nodes. When 2 connections are covered, we have to cover 4 nodes with one optional don't care, so again, we cannot cover all. The right gadget is for equation $x + y + z = 0$, so the gadget should be able to cover 1 or 3 connection nodes. The analysis is similar.

3 Inapproximability for 4-ALLELE$_{n,\ell}$ and 2-ALLELE$_{n,\ell}$ for $a = 3$

Theorem 3 *Both* 4-ALLELE$_{n,\ell}$ *and* 2-ALLELE$_{n,\ell}$ *are* $\frac{305}{304}$*-inapproximable even if* $a = 3$ *assuming P≠NP and (for any* ℓ*) have an easy* $\left(\frac{7}{6} + \varepsilon\right)$*-approximation for any constant* $\varepsilon > 0$*.*

Proof. We reduce the *Triangle Packing* (TP) problem to our problem. We will use the inapproximability result for TP as described in Section 2.

To treat both 4-ALLELE$_{n,\ell}$ and 2-ALLELE$_{n,\ell}$ in an unified framework in our reduction, it is convenient to introduce the 2-label cover problem. The inputs are the same as in 4-ALLELE$_{n,\ell}$ or 2-ALLELE$_{n,\ell}$ except that each locus has just one value (label) and a set is individuals are full siblings if on every locus they have at most 2 values. Thus, each individual can be thought of as an ordered sequence of labels. An instance of the 2-label cover problem can be translated to an instance of our problem by replacing each label in each locus in the following manner:

- for 4-ALLELE$_{n,\ell}$, the label the value v is replaced by the pair (v, v') where v' is a new symbol;
- for 2-ALLELE$_{n,\ell}$ the value v is replaced by the pair (v, v).

We will reduce an instance of TP to the 2-label cover problem by introducing an individual for every node of the graph G with n nodes and providing label sequences for each node (individual) such that:

($\star$) three individuals corresponding to a triangle of G have at most two values on every locus, and

($\star\star$) three individuals that do not correspond to a triangle of G have three values on some locus.

Note that, since any pair of individuals can be full siblings, the above properties imply that TP has a solution with t triangles if and only if the 2-label cover can be covered with $\frac{n-t}{2}$ sibling groups. Thus, Theorem 2 implies that it is NP-hard to decide on instances of $228k$ individuals whether the number of full sibling groups is above $(76-\varepsilon)k$ or below $(75.75+\varepsilon)k$, thereby giving $\frac{305}{304} \approx 1.0032$-inapproximability.

The index of a locus, which we call the coordinate, is defined by:

(a) an "origin" node a, and

(b) *optionally*, a certain edge e.

Thus, we will have at most $O(|V|\cdot|E|)$ loci. The respective label of a node v at this coordinate is the distance from a to v, assuming every edge except e has length 1 while e has length 0. Let $\mathrm{dist}(u, v)$ denote the distance between nodes u and v.

It is easy to see that Property ($\star$) holds. Consider a triangle $\{u, v, w\}$ and assume that u has the minimum label value of L, *i.e.*, it is the nearest with respect to the origin node that defined this locus. Then labels of v and w are at least L and at most $L + 1$, hence we have at most two labels.

It is a bit more involved to verify Property ($\star\star$). Consider a non-triangle $\{u, v, w\}$ in a labeling defined by u (with no edge). u has label 0 and v, w have positive labels which may be equal: if not, we are done; if yes, let $L = \mathrm{dist}(u, v) = \mathrm{dist}(u, w)$.

Consider the two shortest paths from u to v and w, respectively, such that they share a maximally long initial part; so for some node x $\mathrm{dist}(u, v) = \mathrm{dist}(u, x) + \mathrm{dist}(x, v)$, $\mathrm{dist}(u, w) = \mathrm{dist}(u, x) + \mathrm{dist}(x, w)$ and the shortest paths from x to v and w have to be disjoint. Let $\{x, y\}$ be an edge on a shortest path from x to v and now set its length to 0.

First, observe that $\mathrm{dist}(y, w) \geq \mathrm{dist}(x, w)$, since otherwise $\mathrm{dist}(y, w) \leq \mathrm{dist}(x, w) - 1$, $\mathrm{dist(u,v)} = \mathrm{dist(u,x)} + \mathrm{dist(x,y)} + \mathrm{dist(y,v)}$ and also $\mathrm{dist}(u, w) = \mathrm{dist}(u, x) + \mathrm{dist}(x, y) + \mathrm{dist}(y, w)$ and we found a longer common prefix of shortest paths from u to v and w.

Now when we shrink $e = \{x, y\}$ by setting its length to zero, the labels of u and w are unchanged and the label of v drops by 1; we have only two labels only if the labels of u, v and w are 0, 1 and 1, respectively, which implies that $\{u, v\}$ and $\{u, w\}$ are edges.

In this case we label nodes by distances from v; v gets 0, u gets 1, if w also gets 1 then we have edges $\{u, v\}$, $\{u, w\}$ and now we witnessed $\{v, w\}$, hence $\{u, v, w\}$ is a triangle.

This completes the hardness reduction.

On the algorithmic side, Hurkens and Schrijver [28] have a schema that approximates triangle packing within $1.5 + \epsilon$, which means roughly that ca. $(1/3)^{\mathrm{rd}}$ of nodes are left uncovered by triangles and their covering cost increases by 1.5 factor, so the extra cost is ca. $1/6$ of the total (plus ϵ, of course), thus the approximation ratio is $7/6 + \varepsilon$. ❑

4 Inapproximability for 4-ALLELE$_{n,\ell}$ and 2-ALLELE$_{n,\ell}$ for $a = 4$ even if $\ell = 2$ and for 4-ALLELE$_{n,\ell}$ for $a = 6$ even if $\ell = O(n)$

Theorem 4 *Assuming P≠NP, there is no $\frac{6726}{6725}$-approximation algorithm for* 2-ALLELE$_{n,\ell}$ *or* 4-ALLELE$_{n,\ell}$ *even if $a = 4$. Also, either problem (for any ℓ) has an easy $\left(\frac{3}{2} + \varepsilon\right)$-approximation for any constant $\varepsilon > 0$.*

Proof. We will prove the result for 2-ALLELE$_{n,\ell}$ only; a proof for 4-ALLELE$_{n,\ell}$ can be obtained by appropriate modification of the proof and is deferred to the full version of the paper. We will prove the result by showing that, for any constant $\varepsilon > 0$, 2-ALLELE$_{n,\ell}$ cannot be approximated to within a ratio of $\frac{6725}{6724} - \varepsilon > \frac{6726}{6725}$ unless P=NP.

We will reduce an instance $G = (V, E)$ of 3-MAX-CUT to 2-ALLELE$_{n,\ell}$ and use the previously proven result on 3-MAX-CUT as stated in Section 1.2. For notational simplicity, let $m = |E|$. We will provide a reduction from an instance $G = (V, E)$ of 3-MAX-CUT with $336n$ vertices to an instance of 4-ALLELE$_{10m,\ell}$ with $\ell = 2$. The reduction will satisfy the following properties:

(i) a solution of 3-MAX-CUT with a score of x correspond to a solution of 2-ALLELE$_{24m,2}$ with $14m - x$ sibling groups;

(ii) a solution of 2-ALLELE$_{24m,2}$ with z sibling groups can be transformed in polynomial time to another solution of 2-ALLELE$_{24m,2}$ with $14m - y \leq z$ sibling groups (for some positive integer y) such that this solution correspond to a solution of 3-MAX-CUT with a score of y.

Note that this provides the required gap in approximability. Indeed, observe that (with $m = 336 \times \frac{3}{2} \times n = 504n$) 3-MAX-CUT has a solution of score below $(331 - \varepsilon)n$ if and only if 2-ALLELE$_{24m,2}$ has a solution with at least $14 \times 504n - (331 - \varepsilon)n = (6725 + \varepsilon)n$ sibling groups and conversely 3-MAX-CUT has a solution of score above $(332 + \varepsilon)n$ if and only if 2-ALLELE$_{24m,2}$ has a solution with at most $14 \times 504n - (332 + \varepsilon)n = (6724 - \varepsilon)n$ sibling groups; thereby the inapproximability gap is $\frac{6725}{6724} - \varepsilon$.

When we look at *one locus only*, a set of full siblings can have a very limited set of values for alleles. Consider first the case in which every individual has two different elements (alleles) at this locus. We can then view each individual $\{u, v\}$ as an edge in an undirected graph with the two elements u and v representing two nodes in the graph. Three edges (individuals) can be full siblings if they form a path or a cycle; if they do not form a connected graph their union has more than 4 elements, and if they are of the form $\{u, v\}, \{u, w\}, \{u, x\}$ then also they violate the 2-allele condition. Four edges can be full siblings if they form a cycle since they must have only 4 nodes and 3 edges incident on the same node violate the 2-allele condition. The other members in a full sibling group for an individual $\{u, u\}$ can be subsets of either $\{\{u, v\}, \{v, v\}\}$ or $\{\{u, v\}, \{u, w\}, \{v, w\}\}$. In our reduction cycles of length 3 will not exist, so full siblings sets of size larger than two will be paths of 3 edges, cycles of 4 edges

and triples of the form $\{u,u\},\{u,v\},\{v,v\}$. For the purpose of the reduction, it would be more convenient to reformulate the properties **(i)** and **(ii)** of the reduction described above by the following obviously equivalent properties:

(i') a solution of 3-MAX-CUT with a score of $m-x$ correspond to a solution of 2-ALLELE$_{24m,2}$ with $13m+x$ sibling groups;

(ii') a solution of 2-ALLELE$_{24m,2}$ with z sibling groups can be transformed in polynomial time to another solution of 2-ALLELE$_{24m,2}$ with $13m+y \leq z$ sibling groups (for some positive integer y) such that this solution correspond to a solution of 3-MAX-CUT with a score of $m-y$.

We now describe our reduction. We are given a cubic graph G with $2n$ nodes (and thus with $m=3n$ edges) and we will construct an instance J of 2-ALLELE$_{24m,2}$. We replace each node u of G with a gadget G_u that consists of 36 individuals (see Figure 2). Our individuals have two loci. According to the first locus, individuals are edges in a 4-regular graph. Gadget G_u is a 3×12 grid. The rows are closed to form rings of 12 edges, and every fourth column is similarly closed to form a ring on 3 edges. This leaves 6 connected groups of 3 nodes each with 3 neighbors only (*e.g.*, the second, third and fourth node from left on the first row is one such group); these groups are connected to similar groups in other gadgets. A connection between two gadgets consists of two 2×3 grids; for each grid the two rows come from two above-mentioned groups of nodes, one from each gadget.

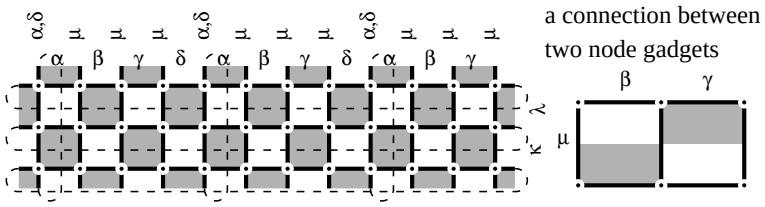


Figure 2: Node gadget G_u for a node u (left) and connections between two node gadgets (right) used in the proof of Theorem 4. The dashed lines indicate wrap-around connections between boundary nodes of the node gadget. The edge labels indicate the values (alleles) in the second locus of each edge (individual). The wrap-around horizontal edges have label δ.

We can view the second locus as labels on edges. A one-letter label a corresponds to a "pair with a repeat", *i.e.*, (a,a), and two-letter label a,b is a "normal pair" (a,b). Inside the 3×12 grid of a node gadget the labels of horizontal edges are equal if one edge is above another, and in a 12-edge ring of such edges labels repeat in a cycle of 4 (and each has one letter). We have similar situation for vertical edges inside the grid. The "wrap-around" edges (in every 4^{th} column) are labeled with proper pairs α,δ such that they intersect the labels of their neighbors. We assume that these labels are unique to every G_u (in Figure 2, these would be labels δ_u and α_u).

The edges that connect node gadgets are labeled μ where μ is the same in all node gadgets and the labels of gadget edges that take part in the connection are the same in all gadgets (thus β and γ are without implicit subscripts).

It is easy to see that every cycle of 4 edges in our new graph is indeed a full siblings set: according to the first locus they are surely so and according to the second locus we can have only two distinct labels on a cycle, *e.g.*, $\{\alpha_u, \lambda\}$ or $\{\beta, \mu\}$. Edges with a "normal pair" label α, δ do not belong to any cycle of length 4.

It is a bit more non-trivial to check that we have only two types of full sibling sets of 3 edges: subsets of 4-cycles, and sets with repeat label α, repeat label δ and normal label α, δ that include "wrap-around" edges and adjacent horizontal edges (one at each end). Basically, if we have two horizontal edges from "different columns" in a set, we cannot add any other label — with the exception we have just described. Recall that a full sibling set of 3 edges forms a path; thus combination of labels like λ, δ and κ is not full siblings.

We give each edge a *potential*. By default it is equal to 0.25. The exceptions are: an edge with the label α, δ has a potential of 0.5, an edge with label μ that is not a center of a group of three nodes in the node gadget that defined an edge connection has a potential of 0.5 and an edge with label μ that is a center of a group of three nodes in the node gadget that defined an edge connection has a potential of 0.

By previous observations, no full siblings set has a potential exceeding 1. Note also that for each node of G we distributed a potential of 19.5, so no cover with full siblings sets can use fewer than $19.5 \times 2n = 39n = 13m$ sets.

Assume that in G we have a cut with $3n - c = m - c$ edges, *i.e.*, a partition of the set of nodes into A and B such that only c edges (of $m = 3n$ edges) are inside the partitions. We will show a cover with $39n + c$ full siblings sets. First we use cycles that correspond to gray squares in every gadget G_u such that $u \in A$, and if $u \in B$ we use cycles that correspond to white square. This is 12 sets per gadgets. Next, in each gadget we use 3 triples centered on α, δ edges. Next, in a connection between A and B we have either two edges labeled β already covered, or two edges labeled γ: in the diagram, suppose that the "lower gadget" is in A, then γ is in a gray square of that gadget; and as the upper gadget is in B and in that edge the upper γ is covered by a white cycle, it is already covered. Thus we can use a cycle with two β edges and two μ's, and one μ is left out. This happens twice in a connection between two gadget, so we add two cycles and one pair of left-out μ's, a total of 3 sets.

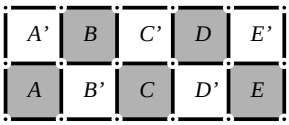
If a connection is inside A or inside B, then the uncovered edges have one β and one γ and they form a path of 5 edges, which can be covered with 2 sets, and since this happens twice, we use 4 sets.

Summarizing, we used $2n \times (12 + 3) + 3n \times 3 + c = 39n + c$ sets. This proves **(i')**.

Now, we prove **(ii')**. Suppose that we have a cover with $39n + c$ sets. We have to normalize it so it will have the form of a cover derived from a cut, without increasing the number of sets. The potential introduced above allows to make local analysis during the normalization. A set with potential $p < 1$ has a *penalty* of $1 - p$, and we have the sum of penalties equal c.

We can assign the penalty to node gadgets. If a set with a penalty is contained in some G_u than the assignment is clear. If we have a set of two edges, then we assign penalty of 0.25 to each edge with potential 0.25 and if such an edge is contained in G_u, we assign the penalty to G_u.

If G_u has a penalty of 1 or more, we remove G_u from consideration and recursively normalize the cover of the remaining gadgets. Once we make this normalization, we partition the remaining nodes into A and B. If a node u has at most one neighbor in A we insert u to A, meaning, we cover it with gray cycles etc, and we will add $19.5 + 1$ sets (an edge not covered counts as half of a set, because we can combine them in pairs).



Thus remains to normalize the cover of G_u assuming that its penalty is at most 0.75. Consider the central horizontal cycle of the grid of G_u: it has 12 edges, and no two of them can belong to the same full sibling set with more than 2 edges; moreover, the sets of at least 3 edges to which they belong are fully contained in G_u. Because G_u obtain at most 0.75 in penalties, at least 9 edges of that 12-cycle are covered by full siblings 4-cycles. Consider the longest connected fragment of such covered edges; assume that they are covered with gray cycles.

Suppose that the last two cycles in that fragment are A and B in the last diagram. We want to change the solution without increasing the number of set and use also cycle C. If C contains a set S used in the current solution, we can enlarge S (making some other sets smaller) and our fragment is extended. If C contains two edges contained in two-edge sets, we can combine the sets so the latter two are in one set, and again we can force C into our solution. So every edge of C is in a different set from the current solution and at most one of these sets is a pair.

Consider the edge on the boundary of B' and C; if it is in a set of more than 3 edges, that set is contained in C – and we excluded that case, or in B' – but only two edges of B' remain uncovered. Hence this edge is contained in a set with two edges only, and it gets a penalty of 0.25 that is delivered to G_u.

Consider the edge on the boundary of C and C'. According to our case analysis, it is contained in a set of at least 3 edges, and which has only one edge in C, so this set is contained in C'. Because A covers one edge of C', we have a set of exactly 3 edges that gets a penalty of 0.25, and thus G_u already got 0.5 of penalty.

We repeat the same reasoning at the other end of the fragment and we double the penalty to 1. The only doubt we can have is that we are counting one of the penalties twice. But this is not possible: the other end of the fragment cannot be covered by C, and it cannot be covered by D, as we use the set $C' \setminus B$ which overlaps D. If the other end of our fragment is covered with E, then we get penalties for the boundary of D and D', and for the set $D' \setminus E$ and we have no double counting. Other cases are similar.

Now an explicitly normalized node gadget has a center row covered with 12 cycles of the same color. The wrap-around edges with α, δ labels can be included in paths of 3 edges – and with potential 1; note that after we committed ourselves to 12 "central" cycles, the edges of such a path do not belong to any other set with more than two edges. Now the

uncovered edges are only in the connection gadgets and they form sets of 5 edges, with no connections between them. We have two such 5-tuples for each connection.

We split the nodes according to the colors used in their gadgets: gray cycles are in set A and white cycles are in set B. If we have a 5 tuple of an $A-B$ connection, its uncovered edges form a cycle and an edge, so we can cover it with 1.5 sets and we cannot do any better. If we have an $A-A$ or $B-B$ connections, the uncovered edges form a path of 5 edges and we much cover them with two sets.

This completes the hardness reduction.

On the algorithmic side, we can use the result of Berman and Krysta [10]. For polynomial time, we have to round the rescaled weights to small integers, so the approximation ratio should have some ϵ added. The 2-IMP with rescaled weight has an approximation ratio of βa, where for $a = 3$ $\beta = 2/3$, for $a = 4$ $\beta = 0.6514$ and for $a > 4$ $\beta = 0.6454$. We can greedily find a maximal packing with sets of size 4 and find 1/2 of the remaining sets of size 3 using 2-IMP algorithm of [10]. Easy analysis shows that that this gives an approximation ratio of 3/2. ❑

Theorem 5 *Assuming P≠NP, there is no* $\frac{1182}{1181}$*-approximation algorithm for* 4-ALLELE$_{n,\ell}$ *even if* $a = 6$ *and* $\ell = O(n)$.

Proof. We will prove the result by showing that, for any constant $\varepsilon > 0$, 4-ALLELE$_{n,\ell}$ cannot be approximated to within a ratio of $\frac{1181}{1180} - \varepsilon$ unless P=NP. Our starting point again is the 3-MAX-CUT problem and the known result about it as stated in Section 1.2. For notational simplicity, let $m = |E|$. We will provide a reduction from an instance $G = (V, E)$ of 3-MAX-CUT with $336n$ vertices to an instance of 4-ALLELE$_{10m,\ell}$ with $\ell = O(m^3)$. The reduction will satisfy the following properties:

(i) a solution of 3-MAX-CUT with a score of x correspond to a solution of 4-ALLELE$_{10m,\ell}$ with $3m - x$ sibling groups;

(ii) a solution of 4-ALLELE$_{10m,\ell}$ with z sibling groups can be transformed in polynomial time to another solution of 4-ALLELE$_{10m,\ell}$ with $3m - y \leq z$ sibling groups (for some positive integer y) such that this solution correspond to a solution of 3-MAX-CUT with a score of y.

Note that this provides the required gap in approximability. Indeed, observe that (with $m = 336 \times \frac{3}{2} \times n = 504n$) 3-MAX-CUT has a solution of score below $(331 - \varepsilon)n$ if and only if 4-ALLELE$_{10m,\ell}$ has a solution with at least $3 \times 504n - (331 - \varepsilon)n = (1181 + \varepsilon)n$ sibling groups and conversely 3-MAX-CUT has a solution of score above $(332 + \varepsilon)n$ if and only if 4-ALLELE$_{10m,\ell}$ has a solution with at most $3 \times 504n - (332 + \varepsilon)n = (1180 - \varepsilon)n$ sibling groups; thereby the inapproximability gap is $\frac{1181}{1180} - \varepsilon$.

Consider an instance $G = (V, E)$ of the 3-MAX-CUT with $336n$ vertices. For conceptual ease, our reduction is separated into two phases:

- We define "gadgets" for vertices and edges of G to obtain a new graph $G' = (V', E')$.

- We then create an instance of 4-ALLELE$_{6m,\ell}$ from G'.

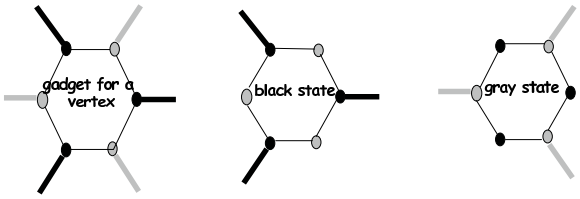


Figure 3: A vertex gadget and its states. The gadget has six internal edges, three outgoing black edges and three outgoing gray edges.

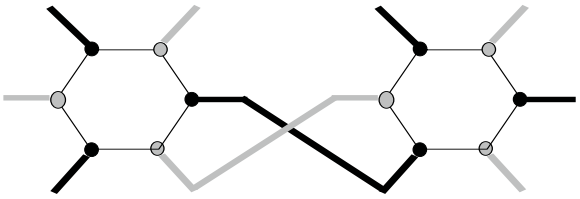


Figure 4: Connection between two adjacent vertex gadgets via black-black and gray-gray connection corresponding to an edge $e = \{u, v\} \in E$. The black-black connection will be denoted by e' and the gray-gray connection will be denoted by e''.

For every vertex $v \in V$, we have a gadget as shown in Figure 3. There are two kinds of nodes in the gadget: *black* and *gray*. The six edges coming out of the gadget are called black or gray depending on whether they are incident on a black or a gray node of the gadget, respectively; the remaining six edges are called *internal* edges. An interesting property of the gadget that we will use is that there are only two minimal ways to cover the gadget using adjacent stars[1] as shown in Figure 3. They are called the black and gray *states* of the gadget and correspond to two states of the vertex that denote in which partition of the 3-MAX-CUT the vertex belongs to. A "normal" state of a vertex gadget correspond to either a black or a gray state.

We now state how to connect two vertex gadgets corresponding to an edge in G. This is shown in Figure 4. In essence we traverse the gadget in clockwise order, select the first two free edges exactly one of which is black and black-black and gray-gray edges are fused. The intuition behind this is that if the two vertex gadgets are in *different* normal states then the both of the fused edges will be covered by the set of stars that cover each gadget, otherwise one such edge will not be covered.

For the convenience of understanding of the reader, we show in Figure 5 the transformation when G is a 3-regular graph on 4 vertices. Note however that G is not a valid input for our reduction since our input graph must have at least 336 vertices.

We now state how to create an instance of 4-ALLELE$_{\ell,10m}$ with $\ell = O(n)$ from G'. For every edge $e = \{u, v\}$ of G' we have two new individuals. We will denote them by $\widetilde{\mathbf{e}^1}$ (or $\widetilde{\{\mathbf{u}, \mathbf{v}\}^1}$) and $\widetilde{\mathbf{e}^2}$ (or $\widetilde{\{\mathbf{u}, \mathbf{v}\}^2}$). Thus, in all we have $12 \cdot |V| + 2 \cdot |E| = 10m$ individuals. We define the following.

Definition 6
(a) *An allowed triplet (of edges) of G' is a set of three edges that form a star. A forbidden triplet of edges of G' is any set of three edges of G' that is not an allowed triplet.*

[1] A star is a set of three edges with exactly one common vertex.

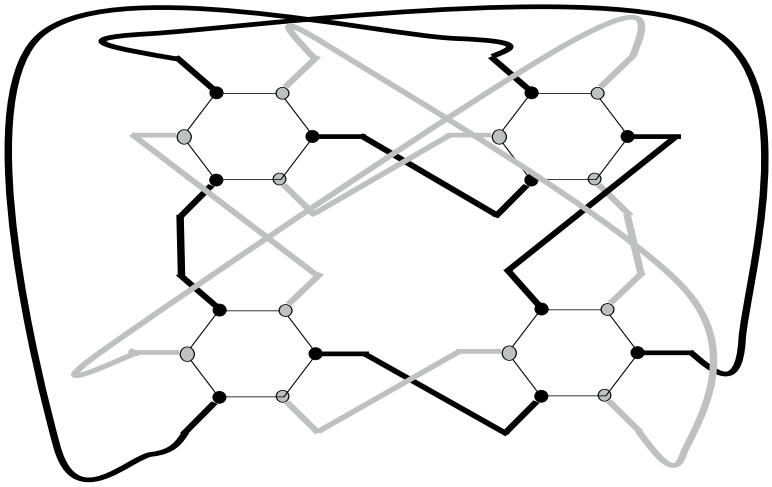


Figure 5: G' corresponding to a 3-regular graph G of 4 vertices. G is not a valid input for our reduction since our input graph must have at least 336 vertices.

(b) *By an allowed triplet (of individuals) of* 4-ALLELE$_{\ell,10m}$ *we mean a set of three individuals* $\{\widetilde{\mathbf{e}^i}, \widetilde{\mathbf{f}^j}, \widetilde{\mathbf{g}^k}\}$ *(for every* $i, j, k \in \{1, 2\}$*) such that* $\{e, f, g\}$ *is an allowed triplet of* G' *A forbidden triplet of* 4-ALLELE$_{\ell,10m}$ *is any triplet that is not an allowed triplet.*

The loci of the $10m$ individuals are now set in the following manner. We first describe the values in the first locus of every individual. Every vertex of the graph corresponds to a new symbol. An individual corresponding to an edge $\{u, v\}$ has the set of alleles $\{u, v\}$ in this first locus.

Note that, according to the first locus, the following cannot be a full siblings: a set of three or more disjoint edges or an edge and a path of two edges. Thus, the only possibilities for a full sibling group are: a cycle of 4 edges (or part of it), three edges incident on the same vertex (a star), a path of three edges, and any pair of edges. Thus, to prohibit forbidden triplets, it suffices to design a gadget for every path of 3 edges in the graph (a cycle of 4 edges is automatically prohibited since it contains at least one such path). There are $O(n)$ such paths of three edges. For every such path consisting of edges e, f and g, we need to design a gadget that will disallow the individuals $\{\widetilde{\mathbf{e}^i}, \widetilde{\mathbf{f}^j}, \widetilde{\mathbf{g}^k}\}$ to be full siblings but will allow any other combinations of three individuals to be full siblings. We will design a gadget that will disallow the individuals $\widetilde{\mathbf{e}}, \widetilde{\mathbf{f}}, \widetilde{\mathbf{g}}$ to be full siblings but will allow any other combinations of three individuals to be full siblings. Select a *new* locus, say i, five symbols, say q_1, q_2, q_3, q_4 and q_5, and put the following values in this locus for the individuals: $(\widetilde{\mathbf{e}}_{i,1}, \widetilde{\mathbf{e}}_{i,2}) = (q_1, q_2)$, $(\widetilde{\mathbf{f}}_{i,1}, \widetilde{\mathbf{f}}_{i,2}) = (q_3, q_4)$, $(\widetilde{\mathbf{g}}_{i,1}, \widetilde{\mathbf{g}}_{i,2}) = (q_5, q_5)$ and $(\widetilde{\mathbf{h}}_{i,1}, \widetilde{\mathbf{h}}_{i,2}) = (q_1, q_1)$ for all $h \in E \setminus \{e, f, g\}$. It is now easy to see if at most two of the individuals from the set of individuals $\{\widetilde{\mathbf{e}}, \widetilde{\mathbf{f}}, \widetilde{\mathbf{g}}\}$ are selected in a full sibling group, then any number of remaining individuals can be selected to be in this group without violating the full sibling condition for this locus. Finally, we need to make sure that no two individuals are identical, *i.e.*, every pair of individuals differ in at

least one locus, while still allowing any subset of individuals to be in a full sibling group. The only two individuals that can be identical are of the form $\widetilde{\mathbf{e}^1}$ and $\widetilde{\mathbf{e}^2}$ for some e. Thus, it suffices if we add a new locus, introduce two symbols a and b, and put $(\widetilde{\mathbf{e}^1}_{i,1}, \widetilde{\mathbf{e}^1}_{i,2}) = (a, a)$ and $(\widetilde{\mathbf{e}^2}_{i,1}, \widetilde{\mathbf{e}^2}_{i,2}) = (b, b)$.

Consider a solution $V' \subseteq V$ of 3-MAX-CUT with a *score* of x. For each vertex $v \in V'$, set the state of its gadget in G' to gray and set the state of all remaining vertex gadgets in G' to black. Every star $\{e, f, g\}$ in each vertex gadget in G' (with exactly two edges being the interior edge of the same gadget) correspond to a set of six individuals $\{\widetilde{\mathbf{e}^1}, \widetilde{\mathbf{e}^2}, \widetilde{\mathbf{f}^1}, \widetilde{\mathbf{f}^2}, \widetilde{\mathbf{g}^1}, \widetilde{\mathbf{g}^2}\}$ that can be full siblings. By our observation before, this covers all individuals except for one pair of individuals $\{\widetilde{\mathbf{d}^1}, \widetilde{\mathbf{d}^2}\}$ for every edge $\{u, v\} \in E$ with $|\{u, v\} \cap V'| \in \{0, 2\}$. The number of such edges in E is obviously $m - x$. We can have a new full sibling group of each such pair of individuals (we cannot cover more than one such pair together because of the forbidden triplet gadgets). This gives a solution of 4-ALLELE$_{\ell,10m}$ with exactly $3 \cdot |V| + (m - x) = 3m - x$ full sibling groups.

Conversely, suppose that we have a solution of 4-ALLELE$_{\ell,10m}$ with y full sibling groups. We first show how to "normalize" this solution to obtain another solution with $y' \leq y$ full sibling groups such that:

(a) All sibling groups are either

- **(a1)** a set of two individuals (a pair) of the form $\{\widetilde{\mathbf{d}^1}, \widetilde{\mathbf{d}^2}\}$ for some edge $d \in E'$ that is not an internal edge of a vertex gadget or,
- **(a2)** a set of six individuals $\{\widetilde{\mathbf{e}^1}, \widetilde{\mathbf{e}^2}, \widetilde{\mathbf{f}^1}, \widetilde{\mathbf{f}^2}, \widetilde{\mathbf{g}^1}, \widetilde{\mathbf{g}^2}\}$ corresponding to a star $\{e, f, g\} \subseteq E'$ with two of edges of star being internal edges of the same vertex gadget.

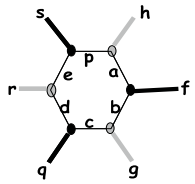
(b) When the full sibling groups of size 6 of 4-ALLELE$_{\ell,10m}$ are mapped back to the vertices of G' every vertex of G' is in normal (black or gray) state.

If this can be done, then the same argument as stated in the previous paragraph implies that $y' = 3m - x$ for some x and this solution of 4-ALLELE$_{\ell,10m}$ provides a solution of 3-MAX-CUT with a score of x.

First, we make the following trivial modification to our solution that does not increase the number of sibling groups. We replace every sibling group $\mathcal{A}$ by a sibling group $\mathcal{A}'$ with the largest number of individuals, if any, that properly includes $\mathcal{A}$. After this, because of the forbidden triplet gadgets, it is easy to see that every sibling group contain either exactly two individuals or exactly six individuals. Moreover, every sibling group of exactly six individuals corresponding to a star $\{e, f, g\} \subseteq E'$.

Suppose that our requirement **(a)** is violated. Thus, we have an edge $d \in E'$ but $\widetilde{\mathbf{d}^1}$ and $\widetilde{\mathbf{d}^2}$ appear in different sibling groups. If at least one of them belong to a group of six individuals, we can put the other one in the group also. Otherwise they belong to two different pairs, say $\{\widetilde{\mathbf{d}^1}, \widetilde{\mathbf{e}^j}\}$ and $\{\widetilde{\mathbf{d}^2}, \widetilde{\mathbf{e}^{j'}}\}$ (for some $j, j' \in \{1, 2\}$). Then, the regrouping $\{\widetilde{\mathbf{d}^1}, \widetilde{\mathbf{d}^2}\}$ and $\{\widetilde{\mathbf{e}^j}, \widetilde{\mathbf{e}^{j'}}\}$ reduces the number of such violations by one. Repeating the above

steps will finally reduce the number of such violations to zero. We then again replace every sibling group $\mathcal{A}$ by a sibling group $\mathcal{A}'$ with the largest number of individuals, if any, that properly includes $\mathcal{A}$. After this, because of the forbidden triplet gadgets, it is easy to see that every sibling group contain either exactly two individuals $\widetilde{\mathbf{d}^1}$ and $\widetilde{\mathbf{d}^2}$ that correspond to a non-internal edge $d \in E'$ or exactly six individuals that correspond to a star $\{e, f, g\} \subseteq E'$ with exactly two of the three edges being internal edges.



Now, suppose that **(b)** is not true. Since each vertex gadget of G' is covered by triplets, two such triplets, say the triplets $\{a, b, f\}$ and $\{b, c, g\}$, must exist that have one edge in common. Considering "shifting" the triplet $\{b, c, g\}$ clockwise, *i.e.*, replacing it by the triplet $\{c, d, q\}$. If the triplet $\{c, d, q\}$ already existed, we can remove one copy and cover the individuals $\{\widetilde{\mathbf{g}^1}, \widetilde{\mathbf{g}^2}\}$ corresponding to the uncovered edge g by a pair and stop. Otherwise, the triplet $\{d, e, r\}$ must exist in our collection. We now "shift" this triplet clockwise to replace it by $\{e, p, s\}$. If the triplet $\{e, p, s\}$ already existed, we can remove one copy and cover the individuals $\{\widetilde{\mathbf{r}^1}, \widetilde{\mathbf{r}^2}\}$ corresponding to the uncovered edge r by a pair and stop. Otherwise, the triplet $\{p, a, h\}$ must exist in our collection. We now "shift" this triplet clockwise to replace it by $\{a, b, f\}$. Since $\{a, b, f\}$ exists in our collection, we can remove one copy and cover the individuals $\{\widetilde{\mathbf{h}^1}, \widetilde{\mathbf{h}^2}\}$ corresponding to the uncovered edge h by a pair and stop. ❑

5 Inapproximability for 4-ALLELE$_{n,\ell}$ and 2-ALLELE$_{n,\ell}$ for $a = n^\delta$

Lemma 7 *For any two constants $0 < \varepsilon < \delta < 1$ with $a = n^\delta$, 4-ALLELE$_{n,\ell}$ and 2-ALLELE$_{n,\ell}$ are n^ε-inapproximable assuming NP$\not\subseteq$ZPP.*

Proof. For any two constants $0 < \varepsilon < \delta < 1$, consider a hard instance $G = (V, E)$ of the graph coloring problem with n vertices $[n] = \{1, 2, \ldots, n\}$ and $\Delta^*(G) \leq |V|^\delta$. As observed in the proof of Theorem 3, it will be sufficient to translate this to an instance $\mathcal{J}$ of the 2-label cover problem. We will have a individual for every vertex i. We will translate an edge $\{i, j\} \in E$ to *exactly* $n-2$ "forbidden triplets" of individuals $\{\{i, j, k\} \mid k \in [n] \setminus \{i, j\}\}$ of the 2-label cover problem such that each of these set of individuals cannot be a full sibling group. We call $\{i, j\}$ as the "anchor" of these triplets. The translation is done by by introducing a new locus and three labels a, b and c, putting a and b as the labels of individuals i and j in this locus, and putting c as the label of every other individual in this locus. Finally, we use the following distinctness gadgets, if necessary, to ensure that all the individuals are distinct. There are at most $O(n^2)$ such gadgets. The purpose of such gadgets is to make sure no two individuals are identical, *i.e.*, every pair of individuals differ in at least one locus, while still allowing any subset of individuals to be in a full sibling group. Consider a pair of individuals u and v that have the same set of loci. Select a *new* locus, two symbols, say a and b, and put a in the locus of all individuals except v and put b in the locus of v.

It suffices to show that our reduction has the following properties:

(1) A set of x vertices of G are independent if and only if the corresponding set of x individuals in $\mathcal{J}$ is a valid full sibling group.

(2) If G can be colored with k colors then $\mathcal{J}$ can be covered with k sibling groups.

(3) If $\mathcal{J}$ can be covered with k' sibling groups then G can be colored with no more than $2k$ colors.

Suppose that we have a set S of independent vertices in G. Suppose that the corresponding set of individuals in $\mathcal{J}$ cannot be a full sibling group and thus must include a forbidden triplet $\{i, j, k\}$ with $\{i, j\}$ as the anchor. Then $\{i, j\} \in E$, thus S is not an independent set. This verifies Property **(1)**.

Suppose that G can be colored with k colors. We claim that the set of individuals corresponding to the set of vertices with the same color constitute a sibling group for either problem. Indeed, since the set of vertices of G with the same color are mutually non-adjacent, they do not include a forbidden triplet. This verifies Property **(2)**.

Finally, suppose that the instance of the generated 2-label cover problem has a solution with k' sibling groups. For each sibling group, select a new color and assign it to all the individuals in the group. Now, map the color of individuals in $\mathcal{J}$ to the corresponding vertices of $G = (V, E)$. Let $E' \subseteq E$ be the set of edges which connect two vertices of the same color. Note that in the graph $G' = (V, E')$ every vertex is of degree at most one since otherwise the sibling group that contains these three individuals corresponding to the three vertices that comprise the two adjacent edges has a forbidden triplet. Thus, we can color the vertices of G' from a set C of two colors. Obviously, the graph $G'' = (V, E \setminus E')$ can be colored with colors from a set D of k' colors. Now, it is easy to see that G can be colored with at most $k \leq 2k'$ colors: assign a new color to every pair in $C \times D$ and color a vertex with the color $(c, d) \in C \times D$ where c and d are the colors that the vertex received in the coloring of G' and G'', respectively. This verifies Property **(3)**. ❑

6 Maximum Profit Coverage (MPC)

Lemma 8
(a) *MPC is NP-hard for $a \geq 3$ and $\Omega\left(\frac{a}{\ln a}\right)$-inapproximable for arbitrary a assuming P≠NP even if every set has weight $a - 1$, every element has weight 1 and every set contains exactly a elements. The hard instances can further be restricted such that each element is a point in some underlying metric space and each set correspond to a ball of radius α for some fixed specified α.*
(b) *MPC is polynomial-time solvable for $a \leq 2$. Otherwise, for any constant $\varepsilon > 0$, the MPC problem has a $\left(\frac{a+1}{2} + \varepsilon\right)$-approximation algorithm for fixed a and a $(0.6454a + \varepsilon)$-approximation algorithm otherwise.*

Proof.
(a) Consider an instance of the independent set problem on a a-regular graph $G = (V, E)$. Build the following instance of the MPC problem. The universe U is E. For every vertex $v \in V$, there is a set S_v consisting of the edges incident on v. Finally, set the weight of every element to be 1 and the weight of every set to be $a - 1$. Note that each set contains exactly a elements.

It is clear that an independent set of x vertices correspond to a solution of the MPC problem of profit x by taking the sets corresponding to the vertices in the solution. Conversely, suppose that a solution of the MPC problem contains two sets S and S' that have a non-empty intersection. Since each set contains exactly a elements, removing one of the two sets from the solution does not decrease the total profit. Thus, one may assume that every pair of sets in a solution of the MPC problem has empty intersection. Then, such a solution involving x sets of total profit x correspond to an independent set of x vertices.

If one desires, one can further restrict the instance of the MPC problem in **(a)** above to the case where each element is a point in some underlying metric space and each set correspond to a ball of radius α for some fixed specified α. All one needs to do is to use the standard trick of setting the weight of each edge in the graph to be α and define the distance between two vertices to be the length of the shortest path between them.
(b) Consider the weighted set-packing problem and let a denote the maximum size of any set. For fixed a, it is easy to use the algorithm for the weighted set-packing as a black box to design a $a/2$-approximation for the MPC problem. For each set S_i of MPC, consider all possible subsets of S_i and set the weight $w(P)$ of each subset P to be the sum of weights of its elements minus q_i. Remove any subset from consideration if its weight is negative. The collection of all the remaining subsets for all S_i's form the instance of the weighted set-packing problem.

It is clear that a solution of the weighted set-packing will never contain two sets S and S' that are subsets of some S_i since then the solution can be improved by removing the sets S and S' and adding the set $S \cup S'$ to the solution (the solution cannot contain the set $S \cup S'$ because of the disjointness of sets in the solution). Thus, at most one subset of any S_i is used the solution of the weighted set-packing. If a subset S of some S_i was used, we use the set S_i in the solution of the MPC problem; note that the elements in $S_i \setminus S$ must be covered in the solution by other sets since otherwise there is a trivial local improvement. In this way, a solution of the weighted set-packing of total weight x corresponds to a solution of the MPC problem of total profit x. Conversely, in an obvious manner a solution of the MPC problem of total profit x corresponds to solution of the weighted set-packing of total weight x.

For $a \leq 2$, weighted set-packing can be solved in polynomial time via maximum perfect matching in graphs.

For fixed $a > 2$, Berman [6] provided an approximation algorithm based on local improvements for this problem produces an approximation ratio of $\frac{a+1}{2} + \varepsilon$ for any constant $\varepsilon > 0$. An examination of the algorithm in [6] shows that the running time of the procedure for our case is $O\left(2^{(a+1)^2} m^{a+1}\right) = O(m^{a+1})$.

When a is *not* a constant, Algorithm 2-IMP of Berman and Krysta [10] can be adapted for MPC to run in polynomial time. For polynomial time, we have to round the rescaled weights to small integers, so the approximation ratio should have some ϵ added. The 2-IMP with rescaled weight has an approximation ratio of $0.6454a$ for any $a > 4$. However, we need a somewhat complicated dynamic programming procedure to implicitly maintain all the subsets for each S_i without explicit enumeration.

Here are the technical details of the adaptation. We will view sets that we can use as having *names* and elements. A name of A is a set $N(A)$ given in the problem instance, and elements form a subset $S(A) \subset N(A)$. The profit $w(S)$ is sum of weights of elements minus the cost of the naming set, $p(A) = w(S(A)) - c(N(A))$.

The algorithm attempts to insert two sets to the current packing and remove all sets that overlap them; this attempt is succesful if the sum of weights raised to power $\alpha > 1$ increases; more precisely, the increase should be larger then some δ, chosen is such a way that it is impossible to perform more than some polynomial time of succesful attempts. As a result, we can measure the weights of sets with a limited precision, so we have a polynomially many different possible weights.

When we insert set with name B that overlaps a set A currently in the solution, we have a choice: remove set A from the solution or remove $A \cap B$ from B. If we also insert a set with name C we have the same dilemma for A and C. Our choice should maximize the resulting sum of $w^\alpha(S)$ for S in the solution.

If we deal with two sets, we can define the quantities
$x_A = p(A - B)$
$x_B = p(B - A)$
$w_{AB} = w(A \cap B)$.

If we include $A \cap B$ in A, the modified profit is $(x_A + w_{AB})^\alpha + x_B^\alpha$.

If we include $A \cap B$ in B, and remove A, the modified profit is $(x_B + w_{AB})^\alpha$.

Our problem is that we know $y_1 = x_A^\alpha$ and $y_1 = w_{AB}$ but we do not know x_B, because the exact composition of B depends on many decisions. Thus we do not know of the following inequality holds for $x = x_B + x_{AB}$:

$$(y_1 + y_2)^+ (x - y_2)^\alpha \leq x^\alpha.$$

It is easy to see that the left-hand-side grows slower than the right-hand side, so once the inequality holds, it is true for all larger x. For this reason it is never optimal to split $A \cap B$ between the two sets, instead we allovate the overlap to one of them.

The situation is similar when we insert two sets. To decide how to handle each overlap of the (names of) sets that we are inserting with the sets already in the solution, it suffices to know their profits. Because we measure profits with a bounded precision, we can make every possible assumption about these two profits, make the decisions and check if the resulting profits are consistent with the assumption; if not, we ignore that assumptions. Among assumptions that we do not ignore, we select one with the largest increase of profits raised to power α. If one of them is positive, we perform the insertion.

Thus we can select a pair of insertion in polynomial time even though we have a number of candidates that is proportional to $n2^a$. Thus our algorithm runs in polynomial time even for $a >> \log n$. Therefore we can achieve the approximation ratio of 2-IMP, *i.e.*, $0.6454a + \varepsilon$, which is better than factor a offered by a greedy algorithm: keep inserting a set with maximum profit that does not overlap an already selected set. ❑

7 2-coverage problem

Lemma 9
(a) *For $f = 2$, the 2-coverage problem is $(1+\varepsilon)$-inapproximable for some constant $\varepsilon > 0$ unless NP⊆BPTIME(2^{n^ε}) and admits a $O(m^{\frac{1}{3}-\varepsilon'})$-approximation for some constant $\varepsilon' > 0$.*
(b) *For arbitrary f, the problem admits a $O(\sqrt{m})$-approximation.*

Proof.
(a) Consider an instance $< G, k >$ of the densest subgraph problem. Then, define an instance of the $(k, 2)$-coverage problem such that $U = E$, there is a set for every vertex in V that contains all the edges incident to that vertex, and we need to pick k sets. Note that for this instance $f = 2$.

For the other direction, define a vertex for every set, connect two vertices if they have a non-empty intersection with a weight equal to the number of common elements. This gives an instance of *weighted* DS whose goal is to maximize the sum of weights of edges in the induced subgraph and admits a $O(m^{\frac{1}{3}-\varepsilon})$-approximation for some constant $\varepsilon > 0$ [20].
(b) For notational convenience it will be convenient to define the (k, ℓ)-coverage problem (for $\ell \geq 1$) which is same as the 2-coverage problem with k sets to be selected except that every element must belong to at least ℓ selected sets (instead of two selected sets). We will also use the following notations. $\mathrm{OPT}(k, \ell, \mathcal{S})$ is the maximum value of the objective function for the (k, ℓ)-coverage problem on the collection of sets in $\mathcal{S}$ and $\mathrm{A}(k, \ell, \mathcal{S})$ is the value of the objective function for the (k, ℓ)-coverage problem on the collection of sets in $\mathcal{S}$ computed by our algorithm. For notational convenience, let $\wp = 1 - (1/\mathbf{e})$. We will give both an $O(k)$ and an $O(m/k)$ approximation which together gives the desired approximation.

The following gives an $O(k)$-approximation. Create a new set $T_{i,j} = S_i \cap S_j$ for every pair of indices $i \neq j$. Run the $(k/2, 1)$-coverage $\wp$-approximation algorithm on the $T_{i,j}$'s and output the elements and, for each selected $T_{i,j}$, the corresponding S_i and S_j. Note that each element is covered at least twice. One can look at all the $\binom{k}{2}$ pairwise intersections of sets in an optimal solution of $(k, 2)$-coverage on $\mathcal{S}$, consider the $k/2$ pairs that have the largest intersections and thus conclude that an optimal solution of 2-coverage on $\mathcal{S}$ covers no more than $O(k)$ times the number of elements in an optimal solution of the $(k/2, 1)$-coverage on the $T_{i,j}$'s.

To get an $O(m/k)$-approximation, first note that $\mathrm{OPT}((k/2), 1, \mathcal{S}) \geq \mathrm{OPT}(k, 2, \mathcal{S})$. Run the $\wp$-approximation algorithm to select the collection of sets $\mathcal{T} \subseteq \mathcal{S}$ to approximate $\mathrm{OPT}((k/2), 1, \mathcal{S})$. For each remaining set in $\mathcal{S} \setminus \mathcal{T}$, remove all elements that do not belong to the sets in $\mathcal{T}$ and remove all elements that are already covered twice in $\mathcal{T}$. We know

that if we were allowed to choose all of the $m-k$ remaining sets in $\mathcal{S} \setminus \mathcal{T}$ we would cover all the elements in the sets $\mathcal{T}$. But since we are allowed to choose only additional $k/2$ sets, we choose those $k/2$ sets from $\mathcal{S} \setminus \mathcal{T}$ that cover the maximum number of elements in the union of sets in $\mathcal{T}$. This involves again running the $\wp$-approximation algorithm. We will cover at least a fraction $k/(2m)$ of the maximum number of elements. ❑

8 Conclusion and Further Research

In this paper we investigated four covering/packing problems that have applications to several problems in bioinformatics. Several questions remain open on the theoretical side. For example, can stronger inapproximability results be proved for 4-ALLELE$_{n,\ell}$ and 2-ALLELE$_{n,\ell}$ intermediate values of a and ℓ that are excluded in our proofs?

References

[1] A. Almudevar. A simulated annealing algorithm for maximum likelihood pedigree reconstruction, *Theoretical Population Biology*, 63, pp. 63–75, 2003.

[2] A. Almudevar and C. Field. Estimation of single generation sibling relationships based on DNA markers, *Journal of Agricultural, Biological, and Environmental Statistics*, 4, pp. 136–165, 1999.

[3] V. Bafna and P. Pevzner. *Genome rearrangements and sorting by reversals*, SIAM. J. Computing, 25, pp. 272-289, 1996.

[4] T. Y. Berger-Wolf, B. DasGupta, W. Chaovalitwongse, and M. V. Ashley. *Combinatorial reconstruction of sibling relationships*, Proceedings of the 6th International Symposium on Computational Biology and Genome Informatics (CBGI 05), pp. 1252–1255, 2005.

[5] T. Y. Berger-Wolf, S. Sheikh, B. DasGupta, M. V. Ashley, I. Caballero, and W. Chaovalitwongse. *Reconstructing Sibling Relationships in Wild Populations*, to appear in Intelligent Systems in Molecular Biology (ISMB) 2007 and to appear in a special issue of Bioinformatics devoted to ISMB 2007.

[6] P. Berman. A $d/2$ Approximation for Maximum Weight Independent Set in d-Claw Free Graphs, Nordic Journal of Computing, 7(3):178-184, Fall 2000.

[7] P. Berman, B. DasGupta, M.-Y. Kao and J. Wang. *On Constructing An Optimal Consensus Clustering from Multiple Clusterings*, to appear in Information Processing Letters.

[8] P. Berman and M. Karpinski. *On some tighter inapproximability results*, 26th Int. Coll. on Automata, Languages, and Programming, pp. 200-209, 1999.

[9] P. Berman and M. Karpinski, *Improved Approximation Lower Bounds on Small Occurrence Optimization Problems*, ECCC TR Report 03-008, 2003, available from `http://eccc.hpi-web.de/eccc-reports/2003/TR03-008/index.html`.

[10] P. Berman and P. Krysta. *Optimizing misdirection*, Proceedings of SODA 2003, pp. 192-201.

[11] J. Beyer and B. May. *A graph-theoretic approach to the partition of individuals into full-sib families*, Molecular Ecology, 12, pp. 2243–2250, 2003.

[12] M. S. Blouin. *DNA-based methods for pedigree reconstruction and kinship analysis in natural populations*, TRENDS in Ecology and Evolution, 18 (10), pp. 503-511, 2003.

[13] K. Butler, C. Field, C. Herbinger and B. Smith. *Accuracy, efficiency and robustness of four algorithms allowing full sibship reconstruction from DNA marker data*, Molecular Ecology, 13, pp. 1589-1600, 2004.

[14] A. Caprara and R. Rizzi. *Packing Triangles in Bounded Degree Graphs*, Information Processing Letters, 84 (4), pp. 175–180, 2002.

[15] W. Chaovalitwongse, T. Y. Berger-Wolf, B. DasGupta and M. V. Ashley. *Set covering approach for reconstruction of sibling relationships*, Optimization Methods and Software, 22 (1), pp. 11-24, February 2007.

[16] J. Chlebíková and M. Chlebík. *Approximation Hardness for Small Occurrence Instances of NP-Hard Problem*, ECCC TR Report 02-073, 2003, available from `http://eccc.hpi-web.de/eccc-reports/2002/TR02-073/index.html`.

[17] J. Chlebíková and M. Chlebík. *Complexity of approximating bounded variants of optimization problems*, Theoretical Computer Science, 354 (3), April 2006, pp. 320–338.

[18] U. Feige. *A threshold for approximating set cover*, JACM, Vol. 45, 1998, pp. 634-652.

[19] U. Feige and J. Kilian. *Zero Knowledge and the Chromatic Number*, Journal of Computers & System Sciences, 57 (2), pp. 187-199, 1998.

[20] U. Feige, D. Peleg, and G. Kortsarz. The dense k-subgraph problem, Algorithmica, 29 (3), pp. 410–421, 2001.

[21] D. Garant and L. E. B. Kruuk. *How to use molecular marker data to measure evolutionary parameters in wild populations*, Molecular Ecology, 14, pp. 1843-1859, 2005.

[22] V. Guruswami, C. Pandu Rangan, M.-S. Chang, G. J. Chang, C. K. Wong. *The Vertex-Disjoint Triangles Problem*, proceedings of WG 1998, pp. 26-37.

[23] D. Gusfield. *Partition-distance: A problem and class of perfect graphs arising in clustering*, Information Processing Letters, 82 (3), pp. 159-164, 2002.

[24] R. L. Hammond, A. F. G. Bourke, and M. W. Bruford. *Mating frequency and mating system of the polygynous ant, Leptothorax acervorum*, Molecular Ecology, 10, pp. 2719–2728, 2001.

[25] R. Hassin and E. Or. A Maximum Profit Coverage Algorithm with Application to Small Molecules Cluster Identification, 5th International Workshop Experimental Algorithms (WEA), LNCS 4007, pp. 265-276, Springer-Verlag, 2006.

[26] J. Håstad. *Some Optimal Inapproximability Results*, Proceedings of STOC 1997, pp. 1-10.

[27] E. Hazan, M. Safra and O. Schwartz. On the Complexity of Approximating k-Set Packing, Computational Complexity, 15(1), pp. 20-39, 2006.

[28] C. A. Hurkens and A. Schrijver. *On the size of systems of sets every t of which have an SDR with applications to worst-case heuristics for packing problems*, SIAM J. Discr. Math, 2(1), pp. 68-72, Feb. 1989.

[29] A. G. Jones, and W. R. Ardren. *Methods of parentage analysis in natural populations*, Molecular Ecology, (12), pp. 2511-2523, 2003.

[30] V. Kann. *Maximum bounded 3-dimensional matching is MAX SNP-complete*, Information Processing Letters, 37, pp 27-35, 1991.

[31] S. Khot. Ruling Out PTAS for Graph Min-Bisection, Densest Subgraph and Bipartite Clique, Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science, pp. 136 - 145, 2004.

[32] S. Khuller, A. Moss and J. Naor. *The budgeted maximum coverage problem*, Information Processing Letters, 70 (1), pp. 39-45, 1999.

[33] K. Kichler, M. T. Holder, S. K. Davis, R. Márquez-M, and D. W. Owens, *Detection of multiple paternity in the Kemps ridley sea turtle with limited sampling*, Molecular Ecology, 8, pp. 819–830, 1999.

[34] D. A. Konovalov, C. Manning, and M. T. Henshaw, *KINGROUP: a program for pedigree relationship reconstruction and kin group assignments using genetic markers*, Molecular Ecology Notes, doi: 10.1111/j.1471-8286.2004.00796.x.

[35] I. Painter. Sibship reconstruction without parental information. *Journal of Agricultural, Biological, and Environmental Statistics*, 2, pp. 212–229, 1997.

[36] B. R. Smith, C. M. Herbinger and H. R. Merry. Accurate partition of individuals into full-sib families from genetic data without parental information. *Genetics*, 158, pp. 1329–1338, 2001.

[37] S. C. Thomas and W. G. Hill. Sibship reconstruction in hierarchical population structures using markov chain monte carlo techniques. *Genet. Res., Camb.*, 79, pp. 227–234, 2002.

[38] V. Vazirani. *Approximation Algorithms*, Springer-Verlag, July 2001.

[39] J. Wang. *Sibship reconstruction from genetic data with typing errors*, Genetics, 166, pp. 1968–1979, 2004.

[40] J. Xu, D. Brown, M. Li and B. Ma. *Optimizing multiple spaced seeds for homology search*, to appear in Journal of Computational Biology.