Journal of Computer and System Sciences 75 (2009) 287–302

Contents lists available at ScienceDirect

Journal of Computer and System Sciences

www.elsevier.com/locate/jcss

On approximating four covering and packing problems

Mary Ashley[a,1], Tanya Berger-Wolf[b,2], Piotr Berman[c], Wanpracha Chaovalitwongse[d,2], Bhaskar DasGupta[b,*,2], Ming-Yang Kao[e]

[a] *Department of Biological Sciences, University of Illinois at Chicago, Chicago, IL 60607-7053, United States*
[b] *Department of Computer Science, University of Illinois at Chicago, Chicago, IL 60607-7053, United States*
[c] *Department of Computer Science & Engineering, Pennsylvania State University, University Park, PA 16802, United States*
[d] *Department of Industrial Engineering, Rutgers University, New Brunswick, NJ 08854, United States*
[e] *Department of Electrical Engineering & Computer Science, Northwestern University, Evanston, IL 60208, United States*

ARTICLE INFO

Article history:
Received 16 February 2008
Received in revised form 18 August 2008
Available online 30 January 2009

Keywords:
Covering and packing problems
Sibling reconstruction in wild population
Triangle packing
Approximation hardness

ABSTRACT

In this paper, we consider approximability issues of the following four problems: *triangle packing*, *full sibling reconstruction*, *maximum profit coverage* and 2*-coverage*. All of them are generalized or specialized versions of set-cover and have applications in biology ranging from full-sibling reconstructions in wild populations to biomolecular clusterings; however, as this paper shows, their approximability properties differ considerably. Our inapproximability constant for the triangle packing problem improves upon the previous results in [A. Caprara, R. Rizzi, Packing triangles in bounded degree graphs, Inform. Process. Lett. 84 (4) (2002) 175–180; J. Chlebíková, M. Chlebík, Complexity of approximating bounded variants of optimization problems, Theoret. Comput. Sci. 354 (3) (2006) 320–338]; 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) [J. Håstad, Some optimal inapproximability results, in: Proc. of the 29th Annual ACM Symp. on Theory of Computing, 1997, pp. 1–10] and is interesting in its own right. Our approximability results on the full siblings reconstruction problems answers questions originally posed by Berger-Wolf et al. [T.Y. Berger-Wolf, B. DasGupta, W. Chaovalitwongse, M.V. Ashley, Combinatorial reconstruction of sibling relationships, in: Proc. of the 6th International Symposium on Computational Biology and Genome Informatics, 2005, pp. 1252–1255; T.Y. Berger-Wolf, S. Sheikh, B. DasGupta, M.V. Ashley, I. Caballero, W. Chaovalitwongse, S.L. Putrevu, Reconstructing sibling relationships in wild populations, Bioinformatics 23 (13) (2007) i49–i56] and our results on the maximum profit coverage problem provides almost matching upper and lower bounds on the approximation ratio, answering a question posed by Hassin and Or [R. Hassin, E. Or, A maximum profit coverage algorithm with application to small molecules cluster identification, in: 5th International Workshop Experimental Algorithms, in: Lecture Notes in Comput. Sci., vol. 4007, Springer-Verlag, 2006, pp. 265–276].

[*] Corresponding author.
E-mail addresses: ashley@uic.edu (M. Ashley), tanyabw@cs.uic.edu (T. Berger-Wolf), berman@cse.psu.edu (P. Berman), wchaoval@rci.rutgers.edu (W. Chaovalitwongse), dasgupta@cs.uic.edu (B. DasGupta), kao@cs.northwestern.edu (M.-Y. Kao).

[1] Supported by NSF grant IIS-0612044.

[2] Supported by NSF grants DBI-0543365, IIS-0612044, IIS-0346973 and DIMACS special focus on Computational and Mathematical Epidemiology.

0022-0000/$ – see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcss.2009.01.002

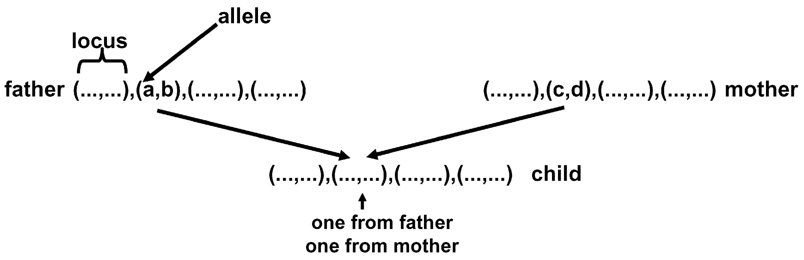

allele
locus
father (...,...),(a,b),(...,...),(...,...)
(...,...),(c,d),(...,...),(...,...) mother
(...,...),(...,...),(...,...),(...,...) child
one from father
one from mother

Fig. 1. Illustration of the Mendelian inheritance rule.

1. Introduction

We consider four combinatorial optimization problems motivated by four separate applications in computational biology. Each of them concerns with packing or covering and falls under a general framework of covering/packing as described below. In the general framework, we have a finite universe of elements and a collection of sets contained in the universe. Optional parameters can be added to the problem statement to specify problems in this framework, and in this paper we use the following (in different combinations): non-negative weights for elements, non-negative weights of sets, a limit on the number of sets that can be selected, the minimum number of selected sets that contain an element, and a family of "conflicts", pairs of sets such that at most one set from a conflict pair can be selected. Our goal is to select a sub-collection of sets that satisfies the constraints (like covering all nodes as required or not containing conflict pairs) and that optimizes an objective function which is linear in terms of the weights of the sets and elements in our selection. For example, both the minimum weight set-cover and the maximum weight coverage problem falls under the above framework. We start out with the precise definitions of our problems and later describe their motivations.

Triangle packing problem (TP) [16,22,27]. We are given an undirected graph G. 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 (k-ALLELE$_{n,\ell}$ for $k \in \{2,4\}$) [4,6,7,17,33,34]. Here the universe $\mathcal{U}$ consists of n elements. To partially motivate the problem, think of each element as an individual in a wild population. Each element p is a sequence $(p_1, p_2, \ldots, p_\ell)$ where each p_j is a genetic trait (*locus*) and is represented as an ordered pair $(p_{j,0}, p_{j,1})$ of numbers (*alleles*) inherited from its parents. We also use $\mathbf{p}_j$ to denote the set $\{p_{j,0}, p_{j,1}\}$. Certain sets of individuals can be full sibling, i.e., having the *same* pair of parents under the Mendelian inheritance rule. These sets are specified in an *implicit* manner in the following way. The Mendelian inheritance rule states that an individual $p = (p_1, p_2, \ldots, p_\ell)$ can be a child of a pair of *parents*, say father $q = (q_1, q_2, \ldots, q_\ell)$ and mother $r = (r_1, r_2, \ldots, r_\ell)$, if for each $i \in \{1, \ldots, \ell\}$ we have $p_{i,0} \in \mathbf{q}_i$ and $p_{i,1} \in \mathbf{r}_i$, or $p_{i,0} \in \mathbf{r}_i$ and $p_{i,1} \in \mathbf{q}_i$; see Fig. 1 for a pictorial illustration. This gives rise to two necessary conditions for a set $\mathcal{A}$ of elements to be full siblings.

Since each individual is generated by the same set of parents, each having at most two distinct alleles in each locus, a set $\mathcal{A}$ of elements can be full siblings if at most 4 alleles occur in each locus, i.e., $|\bigcup_{p \in \mathcal{A}} \mathbf{p}_j| \leqslant 4$ for every $j \in \{1, 2, \ldots, \ell\}$. Sets generated in this manner are said to satisfy the `4-allele condition`. Notice that the 4-allele condition is not a sufficient condition for individuals to be full siblings since it allows an individual to inherit both its alleles from the same parent which violates the Mendelian inheritance rule; nonetheless this condition is used in practice since it is easy to check.

In a more precise way, the full sibling sets can be specified via the `2-allele condition` described below. In a full sibling set, we can reorder the alleles in each locus of each individual in $\mathcal{A}$ so that the first allele always comes from the father and second one comes from the mother. Then, after such a reordering, a set $\mathcal{A}$ of elements can be full siblings if at most 2 alleles occur in each coordinate of the locus. Formally, a set $\mathcal{A} \subseteq \mathcal{U}$ of elements satisfies the 2-allele condition if and only if, for each $p \in \mathcal{A}$ and each $j \in \{1, 2, \ldots, \ell\}$, there exists a reordering $\sigma_{p,j} = (\sigma_{p,j,1}, \sigma_{p,j,2}) \in \{(p_{j,0}, p_{j,1}), (p_{j,1}, p_{j,0})\}$ such that both $|\bigcup_{p \in \mathcal{A}} \{\sigma_{p,j,1}\}| \leqslant 2$ and $|\bigcup_{p \in \mathcal{A}} \{\sigma_{p,j,2}\}| \leqslant 2$ for every $j \in \{1, 2, \ldots, \ell\}$.

With the Mendelian rules in mind, the sets in the k-ALLELE$_{n,\ell}$ problem are all possible sets of elements that satisfy the k-allele condition for $k \in \{2,4\}$. The goal is then to find a collection of sets that cover the universe and the objective is to *minimize* the number of sets selected. As an example to illustrate the k-allele condition, consider the $n = 4$ elements (with $\ell = 2$ loci) $p = (\{1,2\}, \{5,5\})$, $q = (\{3,4\}, \{5,5\})$, $r = (\{1,1\}, \{5,5\})$ and $s = (\{5,5\}, \{5,5\})$. Then, there is no set containing all of p, q, r and s in either 4-ALLELE$_{4,2}$ or 2-ALLELE$_{4,2}$ because $|\{1,2\} \cup \{3,4\} \cup \{1,1\} \cup \{5,5\}| > 4$, the set $\{p, q, r\}$ is contained in the instance of 4-ALLELE$_{4,2}$ but not in the instance of 2-ALLELE$_{4,2}$.

A natural parameter of interest in these problems is the maximum size (number of elements) a of any set; we denote the corresponding problem by a-k-ALLELE$_{n,\ell}$ in some subsequent discussions. One can make the following easy observations:

- Both 2-4-ALLELE$_{n,\ell}$ and 2-2-ALLELE$_{n,\ell}$ are trivial since any two elements always satisfy the k-allele condition for $k \in \{2,4\}$.

- 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 [37]).
- For general a, both a-4-ALLELE$_{n,\ell}$ and a-2-ALLELE$_{n,\ell}$ have a trivial $(\frac{a}{c}+\ln c)$-approximation for any constant $c>0$ obtainable in the following manner. For any integer constant $c>0$, it is trivial to find in polynomial time a set of individuals that are full siblings for both 4-ALLELE$_{n,\ell}$ and 2-ALLELE$_{n,\ell}$, if such a set exists. Thus we can assume that for every induced instance of the problem, either the maximum sibling group size is below c and we can find such a group of maximum size, or we can find a set of size c. Obviously, we can assume that if a sibling group can be used, we can use all its subsets too. Consider an optimum solution, and make it disjoint. We will distribute the cost of an actual solution between the sets of the optimum. When a set with b elements is selected, we remove each of its element and charge the sets of the optimum $1/b$ for each removal. It is easy to see that a set with a elements will get the sequence of charges with values at most $(1/c,\dots,1/c,1/(c-1),1/(c-2),\dots,1)$ and these charges add to $\frac{a}{c}-1+\sum_{i=1}^{c}\frac{1}{i}$, which in turn equals $\frac{a}{c}+\sum_{i=2}^{c}\frac{1}{i}<\frac{a}{c}+\ln c$.

Maximum Profit Coverage problem (MPC) [24]. 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. We extend costs and profits to sets: $q(\mathcal{P})=\sum_{S\in\mathcal{P}}q_{\mathcal{P}}$ and $w(\mathcal{A})=\sum_{i\in\mathcal{A}}w_i$. For $\mathcal{P}\subset\mathcal{S}$ we define the profit $c(\mathcal{P})=w(\bigcup_{A\in\mathcal{P}}A)-q(\mathcal{P})$. 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 the 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 $|\mathcal{P}|\leqslant k$; 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. These relationships include parentage, full and half-sibships, and higher order aspects of pedigrees [14,15,23,28]. In our motivation 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 [1,2,13,31,32,35,36,38]. Most of the currently available methods use statistical likelihood models and are inappropriate for wild populations. Recently, a fully combinatorial approach [4,6,7,17,33,34] to sibling reconstruction has been introduced. This 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. 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 computational complexity issues of these approaches.

MPC has applications in clustering identification of molecules [24]. The 2-coverage problem has motivations in optimizing multiple spaced seeds for homology search (for relevant concepts, see e.g. [39]). For application of TP to genome rearrangement problems, see [5,16].

2. Several useful problems for reductions

Several known problems were used for hardness results. Below we list many of these problems together with the known relevant results. Recall that 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 problem does not have an 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 [25] 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 $(\frac{1}{2}+\varepsilon)m$ 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 $|E|=\frac{3}{2}|V|$). For a subset of vertices $V'\subseteq V$, define *score*(V') to be the number of edges with exactly one endpoint in V' and the other endpoint in $V \setminus V'$. The goal is then to find $V' \subseteq V$ such that $score(V')$ is maximized. We will need the following inapproximability result for this problem proved in [9,10]. For all sufficiently small constants $\varepsilon > 0$, it is impossible to decide, modulo $\mathrm{RP} \neq \mathrm{NP}$ 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 (IS_a). 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 $a \geqslant 3$ and a^c-inapproximable for general a for some constant $0 < c < 1$ assuming $\mathrm{P} \neq \mathrm{NP}$ [3,12,26].

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 [20]: for any two constants $0 < \varepsilon < \delta < 1$, $\chi^*(G)$ cannot be approximated to within a factor of $|V|^{\varepsilon}$ even if $\Delta^*(G) \leqslant |V|^{\delta}$ unless $\mathrm{NP} \subseteq \mathrm{ZPP}$.

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. Let a denote the maximum size of any set. For $a \leqslant 2$, weighted set-packing can be solved in polynomial time via maximum perfect matching in graphs. For fixed $a > 2$, Berman [8] provided an approximation algorithm based on local improvements for this problem produces an approximation ratio of $\frac{a+1}{2} + \varepsilon$ for any constant $\varepsilon > 0$. When a is *not* a constant, Algorithm 2-IMP of Berman and Krysta [11] provides an approximation ratio of $0.6454a$ for any $a > 4$.

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 $\mathrm{NP} \not\subseteq \bigcap_{\varepsilon>0} \mathrm{BPTIME}(2^{n^{\varepsilon}})$ [29]. A more general weighted version of DS admits $O(|V|^{\frac{1}{3}-\varepsilon})$-approximation for some constant $\varepsilon > 0$ [21].

Maximum coverage problem. This is the same as the 2-coverage problem except that the number of elements that occur in at least *one* of the selected sets is *maximized*. Recall that k is the number of sets that we are supposed to select and f is the frequency, i.e., the maximum number of times any element occurs in various sets. Let $\mathbf{e}$ denote the base of natural logarithm. It is known that the maximum coverage problem can be approximated to within a ratio of $1-(1-\frac{1}{k})^k > 1-(1/\mathbf{e})$ by a simple greedy algorithm [30] and approximation with ratio better than $1 - (1/\mathbf{e})$ is not possible unless $P = NP$ [19]. Obviously, the same lower bound carries over to 2-coverage also *for arbitrary* f.

2.1. Our results and techniques

Table 1 summarizes our results.

Brief descriptions of our techniques and comparisons with relevant previous results are as follows.

Triangle packing (TP). The lower bound is shown by a careful reduction from 3-LIN-2 that *roughly* shows that, assuming $\mathrm{RP} \neq \mathrm{NP}$, it is hard to distinguish between instances of TP with profit (the number of disjoint triangles) of at most $75k$ as opposed to a profit of at least $76k$ for every k, thereby giving us an inapproximability ratio of $\frac{76}{75} \approx 1.013$. Our inapproximability constant is larger than the constant $\frac{95}{94} \approx 1.0106$ reported in [18] (assuming $\mathrm{P} \neq \mathrm{NP}$). A proof of Caprara and Rizzi [16] is yet earlier and it implies a still worse inapproximability constant.

4-ALLELE$_{n,\ell}$ and 2-ALLELE$_{n,\ell}$. The inapproximability results for the *smallest* non-trivial value of a, namely $a = 3$, and $\ell = O(n^3)$, are obtained by reducing TP to instances in which the same sets satisfy 2- and 4-allele conditions and each node of the initial graph (the TP instance) is annotated with a sequence of loci so these sets coincide with triangles. The $(\frac{7}{6} + \varepsilon)$-approximation for any ℓ and any constant $\varepsilon > 0$ is easily achieved using the results of Hurkens and Schrijver [27].

The inapproximability results for the *second smallest* non-trivial values of a and ℓ, namely $a = 4$ and $\ell = 2$, are obtained by reducing 3-MAX-CUT via an intermediate novel mapping of geometric nature. The $(\frac{3}{2} + \varepsilon)$-approximations are achieved by using the result of Berman and Krysta [11].

The inapproximability result for $a = n^{\delta}$, namely *all sufficiently large* values of a, is obtained by reducing a suitable hard instance of the graph coloring problem.

In general, for all the above reductions for 4-ALLELE$_{n,\ell}$ and 2-ALLELE$_{n,\ell}$ additional loci are used carefully to rule out possibilities that would violate the validity of our reductions.

Maximum Profit Coverage (MPC). The hardness reduction is from the IS_a and the approximation algorithms are obtained via the weighted set-packing problem. The $(0.6454a + \varepsilon)$-approximation for arbitrary a is obtained via a very careful

Table 1
Summary of results in this paper.

Problem		Lower bound (r-inapproximability)			Upper bound r-approximation for $r =$
		$r =$	Assumption	Reduction problem	
Triangle packing		$(76/75) - \varepsilon \approx 1.013$	$\mathrm{RP} \neq \mathrm{NP}$	3-LIN-2	–
{2,4}-ALLELE$_{n,\ell}$					
$a = 3$	$\ell = O(n^3)$	$(153/152) - \varepsilon \approx 1.0065$	$\mathrm{RP} \neq \mathrm{NP}$	triangle packing	–
$a = 3$	any ℓ	–	–		$(7/6) + \varepsilon \approx 1.166$
$a = 4$	$\ell = 2$	$(6725/6724) - \varepsilon \approx 1.00014$	$\mathrm{RP} \neq \mathrm{NP}$	3-MAX-CUT	–
$a = 4$	any ℓ	–	–		$(3/2) + \varepsilon$
$a = n^\delta$	$\ell = O(n^2)$	$\Omega(n^\varepsilon)$ $\forall \varepsilon < \delta$	$\mathrm{ZPP} \neq \mathrm{NP}$	graph coloring	$\varepsilon n^\delta - \ln \varepsilon$ $\forall$ constant ε
Maximum profit coverage					
$a \leqslant 2$		–	–	–	polynomial-time
$a \geqslant 3$		NP-hard	–	a-regular indep. set	–
constant a		–	–	–	$0.5a + 0.5 + \varepsilon$
any a		a^c	$\mathrm{P} \neq \mathrm{NP}$	a-regular indep. set	$0.6454a + \varepsilon$
2-Coverage					
$f = 2$		$1 + \alpha$	$\mathrm{NP} \nsubseteq \bigcap_{\varepsilon>0} \mathrm{BPTIME}(2^{n^\varepsilon})$	densest subgraph	$O(m^{\frac{1}{3}-\beta})$
any f		–	–	–	$O(\sqrt{m})$

By {2,4}-ALLELE$_{n,\ell}$ we mean that the results apply to *both* 4-ALLELE$_{n,\ell}$ and 2-ALLELE$_{n,\ell}$. $0 < \varepsilon, \delta < 1$ are *any* two constants. α, β and c are *specific* constants mentioned in [29], [21] and [26], respectively, but not explicitly calculated. The parameters a, ℓ, f and m are described in the definitions of the corresponding problems. The a^c-inapproximability result for MPC holds *even if* every set has weight $a - 1$, every element has weight 1, every set contains exactly a elements and even if we impose further restrictions such as each element is a point in some underlying metric space and each set correspond to a ball of radius β for some fixed specified β.

polynomial-time dynamic programming implementation of the 2-IMP approach in Berman and Krysta [11] that *implicitly* maintains subsets for possible candidates for improvement that *cannot be explicitly enumerated* due to their non-polynomial number.

2-coverage. The inapproximability result and approximation algorithms for $f = 2$ are obtained by identifying the problem with the DS problem. Note that the $1 - (1/\mathbf{e})$-inapproximability result for maximum coverage does not extend to 2-coverage under the assumption of $f = 2$. For arbitrary f, we show $O(\sqrt{m})$-approximation by taking the better of a direct greedy approach and another greedy approach based on the maximum coverage problem. 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.

3. Inapproximability result for triangle packing

The theorem below gives a $(76/75) - \varepsilon \approx 1.0133$-inapproximability for TP.

Theorem 1. *Assume* $\mathrm{RP} \neq \mathrm{NP}$. *If* $0 < \varepsilon < 1/2$, *there is no* RP *algorithm that for each instance of* TP *with* $228n$ *nodes and a triangle packing of size at least* $(76 - \varepsilon)n$ *returns a triangle packing of size at least* $(75 + \varepsilon)n$.

Proof. For convenience to readers, we first describe the plan of the proof, then an informal overview of the calculations and finally the details of each component of the proof.
Plan of the proof. As stated before, the following result was obtained by Håstad in [25]. Let L be any language in NP. Then, an instance x of L can be translated in polynomial time to an instance of 3-LIN-2 with $2n$ equations such that, for any constant $0 < \varepsilon < \frac{1}{2}$, the following holds:

- if $x \in L$, then we can satisfy at least $(2 - \varepsilon)n$ equations, and,
- if $x \notin L$, then we can satisfy at most $(1 + \varepsilon)n$ equations.

The above result therefore provides an $(2 - \varepsilon)$-inapproximability of 3-LIN-2 for any small constant $\varepsilon > 0$, assuming $\mathrm{P} \neq \mathrm{NP}$.

Our randomized schema to prove the desired inapproximability result modulo $\mathrm{RP} \neq \mathrm{NP}$ is as follows. Our randomized reduction uses the following polynomial-time transformations that we will devise:

(A) First, we have a randomized "instance transformation" $\mathfrak{T}_{\mathrm{ins}}$ that maps an instance S of 3-LIN-2 with $2n$ equations into a graph $G \equiv \mathfrak{T}_{\mathrm{ins}}(S)$ with $228nm_S$ nodes ($m_S < n$ is a small integer related to the size of S). The algorithm of $\mathfrak{T}_{\mathrm{ins}}$ is randomized and the output is random. *A crucial property of this transformation is that with probability* at least 1/2 *the output is* correct, *i.e., the corresponding instance graph G will satisfy the subsequent requirement in* (C) *below.*

(B) Second, we have a (deterministic) "solution transformation" $\mathfrak{T}_{\text{sol}}$ that maps a solution, say s, of the instance S of 3-LIN-2 with $2n$ equations to a solution $\mathfrak{T}_{\text{sol}}(s, G)$ of the triangle packing problem in the above-mentioned graph G. Our transformation will satisfy the following properties:
   (a) If s satisfies $2n - \ell$ equations of S then $\mathfrak{T}_{\text{sol}}(s, G)$ has $(76n - \ell)m_S$ triangles in G (and $3m_S\ell$ nodes not covered by the triangles). In particular, note that this implies that,
   - if we satisfy $(2 - \varepsilon)n$ equations of S then $\mathfrak{T}_{\text{sol}}(s, G)$ has $(76 - \varepsilon)nm_S$ triangles in G, and,
   - if we satisfy $(1 + \varepsilon)n$ equations of S then $\mathfrak{T}_{\text{sol}}(s, G)$ has $(75 + \varepsilon)nm_S$ triangles in G.

   (b) We can find s in polynomial-time if we are given $\mathfrak{T}_{\text{sol}}(s, G)$.

(C) Third, we have "solution normalization" transformation $\mathfrak{N}$ maps a triangle packing $\mathbb{P}$ in the graph G into another triangle packing $\mathfrak{N}(\mathbb{P}, G)$ in the graph G which is of the form $\mathfrak{T}_{\text{sol}}(s, G)$ for some solution s of the instance S of 3-LIN-2. If G is a "correct output" of $\mathfrak{T}_{\text{ins}}(S)$ then $|\mathfrak{N}(\mathbb{P}, G)| \geqslant |\mathbb{P}|$, i.e., normalization does not decrease the number of triangles in the solution.

Given the above transformation, the overall approach in our proof is as follows. Suppose that we have a polynomial-time randomized algorithm $\mathfrak{A}$ that with probability at least 1/2 finds triangle packing of size larger than $(75 + \varepsilon)/(76 - \varepsilon)$ times the optimum (assuming that one exists). Then, we can use $\mathfrak{A}$ to devise an RP algorithm for any language in NP in the following manner:

(a) We start with an instance x of a language $L \in$NP. Using the proof of Håstad in [25] we translate x in polynomial time to the corresponding instance of S 3-LIN-2 with $2n$ equations.
(b) We compute $G = \mathfrak{T}_{\text{ins}}(S)$.
(c) We compute the triangle packing solution $\mathbb{P} = \mathfrak{A}(G)$.
(d) We compute a new triangle packing solution $\mathbb{Q} = \mathfrak{N}(\mathbb{P})$ using the normalization transformation $\mathfrak{N}$.
(e) If $|\mathbb{Q}| < |\mathbb{P}|$ then we repeat steps (b)–(d) up to a polynomial number of times.
(f) If $|\mathbb{Q}| < |\mathbb{P}|$ in some execution of step (e) then we find the solution s of S that corresponds to $\mathbb{Q}$. If s satisfies strictly more than $(1 - \varepsilon)n$ equations then we declare $x \in L$. In all other cases we declare $x \notin L$.
   One can now see that we are always correct if $x \notin L$ and we are correct with probability at least 1/2 if $x \in L$.

An informal overview of the calculations involved in instance transformation $\mathfrak{T}_{\text{ins}}$. The transformation $\mathfrak{T}_{\text{ins}}$ from an instance S of 3-LIN-2 to an instance (graph) G of triangle packing goes through the following stages. In S we have a system of $2n$ equations modulo 2, with 3 literals per equation, and we can satisfy either at most $(\frac{1}{2} + \varepsilon)$ fraction of the equations or at least $(1 - \varepsilon)$ fraction of the equations.

First, we replicate each equation some (polynomial) m times. This is to increase the minimum number of occurrences of each variable such that the "consistency gadgets" for occurrences will be correct—the correctness of these gadgets is proved "in the limit", i.e., starting from a certain size. This *does not change* the fraction of equations in the system that can be simultaneously satisfied, which is either $1 - \varepsilon$ or $\frac{1}{2} + \varepsilon$.

Denote by $\neg x$ the negation of the variable or constant x modulo 2, i.e., $\neg x = x + 1 \pmod 2$. Then, any equation can have two "normal" forms, namely,

$$x + y + z = b \pmod 2,$$

$$\neg x + \neg y + \neg z = \neg b \pmod 2.$$

We now replace each equation with such a pair. Again, this does not change the proportion of the equations that can be simultaneously satisfied. Our reductions and instance/solution transformations will ensure that each variable $\neg x$ receives a value which is the negation of the value received by variable x. The above replications together account for the constant m_S mentioned in item (A) of the plan of the proof. In other words, after these replications, we have nm_S variables.

Now, our system of equations have some nice properties:

- roughly, for each two equations, both can be satisfied or one;
- same number of negated and non-negated literals;
- same number of equations "$= 0 \pmod 2$)" and "$= 1 \pmod 2$)";
- assured minimum number of occurrences of each variable.

Now, we show our calculation on a normal pair of equations as discussed in the replication method above.

- We have 6 occurrences of literals. We will design a "triplicate gadget" for each, in which each occurrence is represented as 3 nodes called *literal nodes*, thus we have a total of 18 literal nodes. We will design a single gadget for each "= 0" equation that has 6 other nodes, and a gadget for each "$= 1 \pmod 2$)" equation that has 4 other nodes. Thus, we have 10 extra nodes for each normal pair of equations, which makes 30 extra nodes in a "triplicate gadget".
- For each 18 literal node, we will have a part of a consistency gadget in which we have 7 triangles that make a sequence of overlaps. Together, these triangles would have 21 nodes, but one of these node is the literal node, and of the other 20,

each is shared with another triangle, so they are really 10 distinct nodes. For a pair of triplicate gadgets, we have $10 \times 18 = 180$ of the nodes of consistency gadgets.

- Thus, together, we have $(180 + 30 + 18)nm_S = 228nm_S$ nodes.
- Roughly, the two cases of triangle packing (ignoring the ε factors and so forth) are as follows. When both equations in the normal pair are satisfied, we cover them completely with 76 triangles, and when one equation fails, we will loose one triangle thereby covering with 75 triangles.

The outline of the instance translation. Given S, a system of $2n$ equations with 3 variables per equations, we proceed as follows.

1. We replicate each equation six times, three times as a simple copy, $x + y + z = b \bmod 2$ and three times as $\bar{x} + \bar{y} + \bar{z} = \bar{b} \bmod 2$. Having the same number of literals x as $\bar{x}$ helps in point 5, and having each equation copied three times helps in point 4.
2. We replicate the equations in S m times for a sufficiently (polynomially) large m such each variable occurs sufficiently (polynomially) many times. The construction in point 5 is faulty with probability $O(c^{m'})$ for some $c < 1$ when m' the number of occurrences of a variable.
3. For each literal (occurrence of a variable or its negation in an equation) we create a separate node. From now on, *literal* will mean such a node.
4. We replace three copies of equation e with an *equation gadget* B_e that contains nine literals of e (three, each in three copies) as well as other nodes.
5. For each variable x we create *consistency gadget* C_x that all the literals of x, as well as other nodes.

Constructing consistency gadget C_x. The problem of triangle packing can be mapped into the independent set problem in the following manner: starting from a graph (V, E) we create a graph (V', E'), where V' is the set of triangles in E, and $\{t, t'\} \in E'$ if triangles t and t' share a node.

If graph G' is cubic, i.e., each node has degree 3, we can have the reverse transformation: from (V', E') to (V, E); $V = E'$, and $\{e, e'\} \in E$ if e and e' are incident to the same node. In this case, a node $u \in V$ with neighbors v_i, $i = 0, 1, 2$, is transformed into nodes $\{u, v_i\}$, $i = 0, 1, 2\}$ and those three nodes for a triangle. A pair of such triangles is node-disjoint if the original nodes were not adjacent.

This point of view is not helpful in the construction of equation gadgets because we obtained smaller gadgets than those that would correspond to fragments of cubic graphs. However, our consistency gadget are obtained by such a transformation.

In particular, we will use a gadget, called an *amplifier*, introduced by Berman and Karpinski [9] in the context of maximum cut problem (see also J. Chlebíková and M. Chlebík [18]).

Assume that we construct G_x for a variable with $2k$ occurrences (k simple, k negated). The respective amplifier can be defined as the graph (V^a, E^a) where $V^a = \{u_0, \ldots, u_{14k-1}\}$, This graph is bipartite, all edges are between even nodes and odd nodes; we will refer to odd and even nodes as white and black. There are two classes of edges, the first forms a ring, $\{u_i.u_{i+1 \bmod 14k}\}$, the second forms a random matching between white (even) and black (odd) nodes whose indices *are not divisible by* 7. Nodes with indices divisible by 7 are called *contacts*, each of these nodes belongs also to an equation gadgets.

We wish a solutions—a $U \subset V$ of nodes—to be consistent within consistency gadgets. Equation gadgets "see" only the contacts. Set U is consistent within our gadget if either U contains all black contacts and none of the white ones, or vice versa. If we have an inconsistent solution, we replace it with the choice "all white" or "all black" that requires fewer changes of membership among the contacts. Here is the key property (that holds with the probability that converges to 1 as $k \to \infty$): if $U \subset V^a$ contains $i \leqslant k$ contacts of one color (the minority) and at least as many nodes of another (the majority color), then at least i edges of E^a do not belong to the cut of U.

The use of this property is that when we normalize a solution to coincide, all contacts of G_x should correspond to a single value assigned to x; we can achieve it by altering the solution to coincide the color that contains more contacts. If the normalization changes the membership of i contacts of x, we gain i units of the objective function—edges of the cut—within the gadget. Presumably the size of the cut decreases within equation gadgets, but the decrease is bounded by i, the number of contacts that changed the membership.

Now we have to translate this usage of the amplifier to the independent set problem. In a bipartite cubic graph with $14k$ nodes, an independent set S has cut $3|S|$, and if we have $3i$ edges not in the cut, then $|S| = 7k - i$. Thus the same amplifier construction can be used for independent set problem. If $U \subset V^a$ contains $i \leqslant k$ contacts of its minority color, then at least i edges of E^a are not covered by U.

Then we can translate the amplifier into a part of triangle packing as shown in Fig. 2, and the property can be rephrase by having i nodes not covered by the solution triangle packing within an equation gadget if i contacts are covered in a minority manner (if the majority of contacts covered by a solution is black, black is the D majority color and inconsistent consistent contact are black contacts that do not belong, as well as white contacts that do belong).

How equation gadget B_e works. Equations were replicated so they can be grouped into triples of identical equations. We create gadgets for equations and then, for each group of three, we connect identical gadgets by providing triangles that cover one node in each of them.

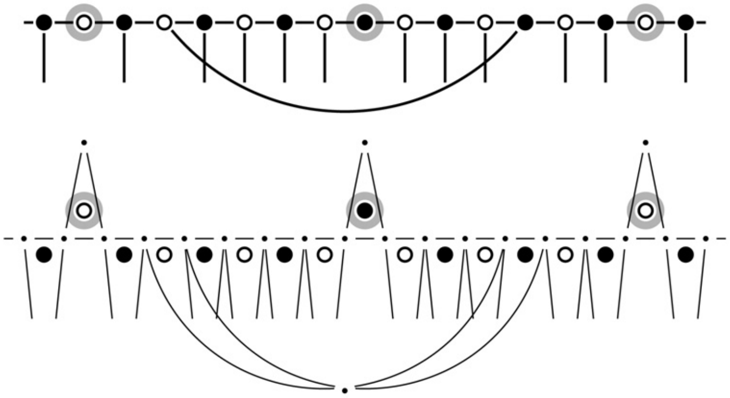


Fig. 2. A fragment of an amplifier and its translation into a fragment of our equation gadget. Contacts are indicated by a gray "halo". Note that after translation, each original contact node becomes a contact triangle. Each contact triangle contains a contact node (in the diagram, on top). If we choose white triangles, then contact nodes of the black triangles are not covered within the gadget, and vice versa when we choose all black triangles.

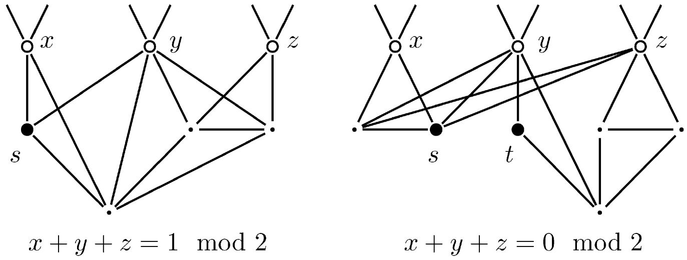


Fig. 3. Equation gadgets, used in three copies. Thick dots are nodes connected with other copies (self-sufficient), empty circles are literals, nodes shared with consistency gadgets of variables.

For such a group of copies of equation e, let B^i_e, $i=0,1,2$, be an individual gadget and B_e the combined one.

Thus we can describe a triple gadget by describing an individual gadget, $B^i_e=(V^i_e,E^i_e)$ and specifying set S^i_e of nodes that are connected to their copies in other individual gadget. From the point of view of an individual gadget, nodes in S^i_e can be covered separately.

Assume that $e \equiv x'+y'+z'=b \mod 2$ where x' is a literal of x (x or $\bar{x}$). An individual gadget contains these three literals.

The property of an individual gadget B^i_e is that V^i_e can have all nodes covered by a triangle packing and S^i_e if only if the literals are covered consistently with values that make e satisfied. For example, if $e \equiv x+y+z=0 \mod 2$, and none (or exactly two) of the three literals contained in V^i_e is covered by triangles contained in $C_x \cup C_y \cup C_z$. The property of the combined gadget is that if the literals are covered consistently (e.g. either all x' are covered by triangles contained in C_x or none), then either they are covered consistently with values that satisfy e and we can cover entire $V_e=V^0_e \cup V^1_e \cup V^2_e$, or the literals are covered consistently with values that do not satisfy e and we can cover V_e except for three nodes (one exception in each V^i_e).

Properties of gadgets imply correct normalization. So far, we described $\mathbb{Q}=\mathfrak{N}(\mathbb{P})$ only partially, namely how to select triangles contained in consistency gadget G_x (white or black, corresponding to assigning 0 or 1 to x). If the normalization change the way i contacts are covered, then within G_x we cover all nodes with the triangle, while before we did not cover i of them. Thus we can pass to each "minority case" a permission not to cover one node.

Now consider a combined equation gadget. If the majority cases satisfy the equation, after the normalization we cover all nodes of the equation gadget. Otherwise each individual gadget either contained a minority case literal and will receive a permission not to cover a node, or it had all majority cases and thus at least one uncovered node. Thus to maintain the number of covered nodes it suffices to cover the nodes in the gadget with three exceptions.

Construction of B^i_e. Consider equation $e \equiv x+y+z=0 \mod 2$. Node set V^i_e consists of three literals (one copy of x,y,z), two self-sufficient nodes $S^i_e=\{s^i,t^i\}$ and four other nodes. (See Fig. 3.)

If x,y,z are false, this is coded by a solution in which none is already covered by triangles from their consistency gadgets, we cover the nine nodes of B^i_e with three triangles. If exactly two are already covered, we cover the uncovered literal, s^i and "four other nodes" with two triangles.

If exactly one of the literals true (already covered), we would have to cover eight nodes. This could be done only with two triangles and two self-sufficient nodes; however the triangles disjoint with S_e^i all overlap, so the best we can do is to use one such triangle, one triangle that contains s^i and t^i, leaving one non-self-sufficient node uncovered.

If three literals are true, we would have to cover six nodes, this could be done only with two triangles, but there is only one triangle that does not contain literals, so the best we can do is to use this triangle, as well as S_e^i, leaving one of the "other nodes" uncovered.

Now consider equation $e \equiv x + y + z = 1 \bmod 2$. Sub-gadget B_e^i contains x^i, y^i, z^i, self-sufficient node s^i and three other nodes.

If exactly one of the literals is true, we have to cover six nodes, which we can do with two triangles. If all literals are true, we have to cover 4 nodes, which we do using a triangle that is disjoint with s^i, as well as s^i.

If no literal is true, we would have to cover 7 nodes, this could be done only with two triangles and s^i, but all triangles that do not contain s^i overlap. If two literals are true, we would have to cover 5 nodes, impossible. But if we pretend that one more literal is covered we can cover all other nodes, so when the equation is false we leave one non-self-sufficient node uncovered.

The property of the combined gadget. It is easy to see that when the literals are consistent we can cover each individual gadget in the same way, so when any nodes remain uncovered they form triples of corresponding self-sufficient nodes and thus they are covered by the triangles that connected individual gadgets. □

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

Theorem 2. *Both* 4-ALLELE$_{n,\ell}$ *and* 2-ALLELE$_{n,\ell}$ *are* $((153/152) - \varepsilon)$*-inapproximable even if* $a = 3$ *assuming* RP $\neq$ NP *and (for any* ℓ*) admit* $((7/6) + \varepsilon)$*-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 3.

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 of 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 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 1 implies that it is NP-hard to decide on instances of $228k$ individuals whether the number of full sibling groups is above $(228 - 76 + \varepsilon)k/2$ or below $(228 - 75 - \varepsilon)k/2$, thereby giving $(153/152) - \varepsilon \approx (1.0064 - \varepsilon)$-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 $\operatorname{dist}(u, v)$ 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 $L = \operatorname{dist}(u, v) = \operatorname{dist}(u, w)$.

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 $\operatorname{dist}(u, v) = \operatorname{dist}(u, x) + \operatorname{dist}(x, v)$, $\operatorname{dist}(u, w) = \operatorname{dist}(u, x) + \operatorname{dist}(x, w)$ 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 $\text{dist}(y, w) \geqslant \text{dist}(x, w)$, since otherwise $\text{dist}(y, w) \leqslant \text{dist}(x, w) - 1$, $\text{dist}(u, v) = \text{dist}(u, x) + \text{dist}(x, y) + \text{dist}(y, v)$ and also $\text{dist}(u, w) = \text{dist}(u, x) + \text{dist}(x, y) + \text{dist}(y, w)$ 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, suppose that an optimal solution for either version of the sibling problem on n individuals involve a triples and b pairs of individuals (and, thus $3a + 2b$). Hurkens and Schrijver [27] have a schema that approximates triangle packing within a ratio of $1.5 + \varepsilon$ for any constant $\varepsilon > 0$. We can use this algorithm to get at least $(2a/3) - \varepsilon$ triples. We can cover the remaining $n - (2a - 3\varepsilon) = a + 2b + 3\varepsilon$ elements by pairs. Thus, we use at most $(2a/3) - \varepsilon + (a/2) + b + (3/2)\varepsilon = (7a/6) + b + (\varepsilon/2)$ which is within a factor of $(7/6) + \varepsilon$ of $a + b$. □

5. Approximability of 4-ALLELE$_{n,\ell}$ and 2-ALLELE$_{n,\ell}$ for $a = 4$

Theorem 3. *For $a = 4$, both* 4-ALLELE$_{n,\ell}$ *and* 2-ALLELE$_{n,\ell}$ *are $((6725/6724) - \varepsilon)$-inapproximable even if $\ell = 2$ assuming* RP $\neq$ NP *and (for any ℓ) admit $((3/2) + \varepsilon)$-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 an easy modification of the above proof. 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 $\frac{6725}{6724} - \varepsilon$ unless RP = NP.

We will reduce an instance $G = (V, E)$ of 3-MAX-CUT to 2-ALLELE$_{n,\ell}$ and use the previously proved result on 3-MAX-CUT as stated in Section 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 $14m - y \leqslant z$ 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 $13m + y \leqslant z$ 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 Fig. 4). 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

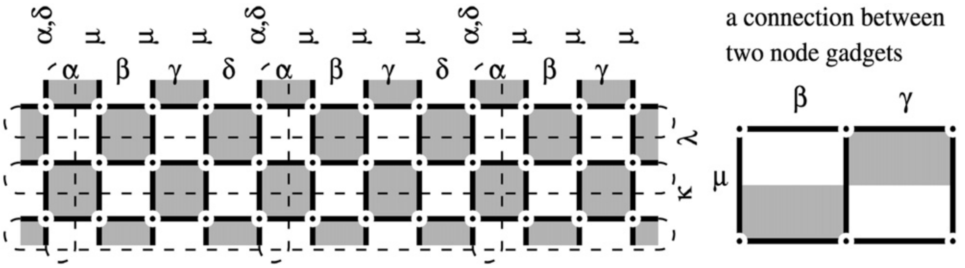


Fig. 4. Node gadget G_u for a node u (left) and connections between two node gadgets (right) used in the proof of Theorem 3. The dashed lines indicate wrap-around connections between boundary nodes of the node gadget. The edge labels indicate the values (alleles) in the second locus of each edge (individual). The wrap-around horizontal edges have label δ.

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.

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 4th 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 Fig. 4, 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.

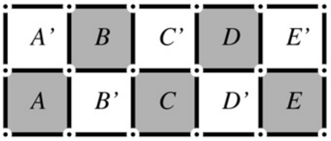
If a connection is inside A or inside B, then the uncovered edges have one β and one γ and they form a path of 5 edges, which can be covered with 2 sets, and since this happens twice, we use 4 sets.

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 [11]. 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 [11]. Easy analysis shows that this gives an approximation ratio of 3/2. □

Remark 1. Using a reduction again from 3-MAX-CUT that is similar in flavor to the above proof (but with different gadgets, different covering components and simpler case analysis) one can prove that, assuming RP $\neq$ NP, there is no $((1182/1181) - \varepsilon)$-approximation algorithm for 4-ALLELE $_{n,\ell}$ even if $a = 6$ and $\ell = O(n)$ for any constant $\varepsilon > 0$.

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

Lemma 4. *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 $\Delta^*(G) \leqslant |V|^\delta$. As observed in the proof of Theorem 2, 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 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 $x \geqslant 3$ 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. Conversely, suppose that the set of individuals $\mathcal{J}$ is be a full sibling group. Then, they cannot include a forbidden triplet. 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 $k \leqslant 2k'$ 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). □

7. Approximating maximum profit coverage (MPC)

Lemma 5.

(a) MPC *is* NP-*hard for* $a \geqslant 3$ *and* a^c-*inapproximable for arbitrary* a *and some constant* $0 < c < 1$ *assuming* $\mathrm{P} \neq \mathrm{NP}$ *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* $a \leqslant 2$. *Otherwise, for any constant* $\varepsilon > 0$, MPC *admits* $(0.5a + 0.5 + \varepsilon)$-*approximation for fixed* a *and* $(0.6454a + \varepsilon)$-*approximation 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 an $a/2$-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 $a \leqslant 2$, weighted set-packing can be solved in polynomial time via maximum perfect matching in graphs.

For fixed $a > 2$, Berman [8] 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 [8] shows that the running time of the procedure for our case is $O(2^{(a+1)^2} m^{a+1}) = O(m^{a+1})$.

When a is *not* a constant, Algorithm 2-IMP of Berman and Krysta [11] 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 successful 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 successful 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 if the following inequality holds for $x = x_B + x_{AB}$:

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

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 allocate 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 $a \gg \log n$. 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. □

8. Approximating 2-coverage

Lemma 6.

(a) *For $f = 2$, 2-coverage is $(1+\varepsilon)$-inapproximable for some constant $\varepsilon > 0$ unless* NP $\not\subseteq \bigcap_{\varepsilon>0}$ BPTIME(2^{n^ε}) *and admits $O(m^{\frac{1}{3}-\varepsilon'})$-approximation for some constant $\varepsilon' > 0$.*

(b) *For arbitrary f, 2-coverage admits $O(\sqrt{m})$-approximation.*

Proof. (a) Consider an instance $\langle G, k\rangle$ 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 $O(m^{\frac{1}{3}-\varepsilon})$-approximation for some constant $\varepsilon > 0$ [21].

(b) For notational convenience it will be convenient to define the (k,ℓ)-coverage problem (for $\ell \geqslant 1$) 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 $\mathrm{OPT}((k/2),1,\mathcal{S}) \geqslant \mathrm{OPT}(k,2,\mathcal{S})$. 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. □

9. 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?

Acknowledgments

We would like to thank the anonymous reviewers for their helpful comments that led to significant improvements in the presentation of the paper.

References

[1] A. Almudevar, A simulated annealing algorithm for maximum likelihood pedigree reconstruction, Theoret. Population Biol. 63 (2003) 63–75.

[2] A. Almudevar, C. Field, Estimation of single generation sibling relationships based on DNA markers, J. Agric. Biol. Environ. Stat. 4 (1999) 136–165.

[3] N. Alon, U. Fiege, A. Wigderson, D. Zuckerman, Derandomized graph products, Comput. Complexity 5 (1995) 60–75.

[4] M.V. Ashley, I.C. Caballero, W. Chaovalitwongse, B. DasGupta, P. Govindan, S. Sheikh, T.Y. Berger-Wolf, KINALYZER, A Computer Program for Reconstructing Sibling Groups, Molecular Ecology Resources, in press.

[5] V. Bafna, P. Pevzner, Genome rearrangements and sorting by reversals, SIAM J. Comput. 25 (1996) 272–289.

[6] T.Y. Berger-Wolf, B. DasGupta, W. Chaovalitwongse, M.V. Ashley, Combinatorial reconstruction of sibling relationships, in: Proc. of the 6th International Symposium on Computational Biology and Genome Informatics, 2005, pp. 1252–1255.

[7] T.Y. Berger-Wolf, S. Sheikh, B. DasGupta, M.V. Ashley, I. Caballero, W. Chaovalitwongse, S.L. Putrevu, Reconstructing sibling relationships in wild populations, Bioinformatics 23 (13) (2007) i49–i56.

[8] P. Berman, A $d/2$ approximation for maximum weight independent set in d-claw free graphs, Nordic J. Comput. 7 (3) (2000) 178–184.

[9] P. Berman, M. Karpinski, On some tighter inapproximability results, in: Proc. of the 26th Int. Coll. on Automata, Languages, and Programming, 1999, pp. 200–209.

[10] 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.

[11] P. Berman, P. Krysta, Optimizing misdirection, in: Proc. of the 14th ACM-SIAM Symp. on Discrete Algorithms, 2003, pp. 192–201.

[12] P. Berman, G. Schnitger, On the complexity of approximating the independent set problem, Inform. and Comput. 96 (1992) 77–94.

[13] J. Beyer, B. May, A graph-theoretic approach to the partition of individuals into full-sib families, Molecular Ecol. 12 (2003) 2243–2250.

[14] M.S. Blouin, DNA-based methods for pedigree reconstruction and kinship analysis in natural populations, Trends Ecol. Evol. 18 (10) (2003) 503–511.

[15] K. Butler, C. Field, C. Herbinger, B. Smith, Accuracy, efficiency and robustness of four algorithms allowing full sibship reconstruction from DNA marker data, Molecular Ecol. 13 (2004) 1589–1600.

[16] A. Caprara, R. Rizzi, Packing triangles in bounded degree graphs, Inform. Process. Lett. 84 (4) (2002) 175–180.

[17] W. Chaovalitwongse, T.Y. Berger-Wolf, B. DasGupta, M.V. Ashley, Set covering approach for reconstruction of sibling relationships, Optim. Methods Softw. 22 (1) (2007) 11–24.

[18] J. Chlebíková, M. Chlebík, Complexity of approximating bounded variants of optimization problems, Theoret. Comput. Sci. 354 (3) (2006) 320–338.

[19] U. Feige, A threshold for approximating set cover, J. ACM 45 (1998) 634–652.

[20] U. Feige, J. Kilian, Zero knowledge and the chromatic number, J. Comput. System Sci. 57 (2) (1998) 187–199.

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

[22] V. Guruswami, C. Pandu Rangan, M.-S. Chang, G.J. Chang, C.K. Wong, The vertex-disjoint triangles problem, in: Proc. of the 24th International Workshop on Graph-Theoretic Concepts in Computer Science, 1998, pp. 26–37.

[23] R.L. Hammond, A.F.G. Bourke, M.W. Bruford, Mating frequency and mating system of the polygynous ant, Leptothorax acervorum, Molecular Ecol. 10 (2001) 2719–2728.

[24] R. Hassin, E. Or, A maximum profit coverage algorithm with application to small molecules cluster identification, in: 5th International Workshop Experimental Algorithms, in: Lecture Notes in Comput. Sci., vol. 4007, Springer-Verlag, 2006, pp. 265–276.

[25] J. Håstad, Some optimal inapproximability results, in: Proc. of the 29th Annual ACM Symp. on Theory of Computing, 1997, pp. 1–10.

[26] E. Hazan, M. Safra, O. Schwartz, On the complexity of approximating k-set packing, Comput. Complexity 15 (1) (2006) 20–39.

[27] C.A. Hurkens, 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. Discrete Math. 2 (1) (1989) 68–72.

[28] A.G. Jones, W.R. Ardren, Methods of parentage analysis in natural populations, Molecular Ecol. 12 (2003) 2511–2523.

[29] S. Khot, Ruling out PTAs for graph min-bisection, densest subgraph and bipartite clique, in: Proc. of the 45th Annual IEEE Symp. on Foundations of Computer Science, 2004, pp. 136–145.

[30] S. Khuller, A. Moss, J. Naor, The budgeted maximum coverage problem, Inform. Process. Lett. 70 (1) (1999) 39–45.

[31] D.A. Konovalov, C. Manning, M.T. Henshaw, KINGROUP: A program for pedigree relationship reconstruction and kin group assignments using genetic markers, Molecular Ecol. Notes 4 (2004) 779–782.

[32] I. Painter, Sibship reconstruction without parental information, J. Agric. Biol. Environ. Stat. 2 (1997) 212–229.

[33] S.I. Sheikh, T.Y. Berger-Wolf, M.V. Ashley, I.C. Caballero, W. Chaovalitwongse, B. DasGupta, Error tolerant sibship reconstruction in wild populations, in: P. Markstein, Y. Xu (Eds.), Computational Systems Bioinformatics, 7th Annual International Conference on Computational Systems Bioinformatics, 26–29 August, 2008, World Scientific Publishers, 2008, pp. 273–284.

[34] S. I. Sheikh, T. Y. Berger-Wolf, A. A. Khokhar, B. DasGupta, Consensus methods for reconstruction of sibling relationships from genetic data, in: 4th Multidisciplinary Workshop on Advances in Preference Handling, Chicago, IL, 2008.

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

[36] S.C. Thomas, W.G. Hill, Sibship reconstruction in hierarchical population structures using Markov chain Monte Carlo techniques, Genet. Res. 79 (2002) 227–234.

[37] V. Vazirani, Approximation Algorithms, Springer-Verlag, 2001.

[38] J. Wang, Sibship reconstruction from genetic data with typing errors, Genetics 166 (2004) 1968–1979.

[39] J. Xu, D. Brown, M. Li, B. Ma, Optimizing multiple spaced seeds for homology search, J. Comput. Biol. 13 (7) (2006) 1355–1368.