Shifting Is Optimal Under Gap-ETH: A Lower Bound Framework for Geometric Approximation Schemes

Manuel Cáceres
Department of Computer Science, Aalto University, Finland
Department of Mathematics and Computer Science, University of Southern Denmark, Odense, Denmark

Sándor Kisfaludi-Bak
Department of Computer Science, Aalto University, Espoo, Finland

Saeed Odak
Department of Computer Science, Aalto University, Espoo, Finland

Abstract

The shifting technique of Hochbaum and Maass [J.ACM'85] produces PTASes with the fastest known running times $n^{O(1/\varepsilon^{d-1})}$ for several d dimensional geometric problems. However, it is only known, due to Marx [FOCS'07], that these algorithms are indeed optimal for dimension $d = 2$. We show that these running times are optimal under Gap-ETH for every constant dimension. More precisely, we develop a framework that enables us to prove the conditional optimality of the shifting algorithms for several problems on unit ball graphs, such as maximum independent set, maximum induced forest, and others, as well as for the problem of piercing unit balls. Our framework is built using the cube wiring theorem of De Berg et al. [SICOMP'20] and the reduction steps of Marx and Sidiropoulos [SoCG'14] to create a convenient maximization version of geometric CSP that can be used as a basis for reductions.

2012 ACM Subject Classification Theory of computation → Approximation algorithms analysis; Theory of computation → Problems, reductions and completeness; Theory of computation → Computational geometry; Theory of computation → Packing and covering problems

Keywords and phrases Gap-ETH Lower Bounds, Geometric PTASes, Geometric CSP, Shifting Technique, Maximum Induced Forest, Constant Dimension

Digital Object Identifier 10.4230/LIPIcs.ESA.2026.38

Related Version *Full Version*: https://arxiv.org/abs/2607.06069

Funding *Manuel Cáceres*: Supported by the Helsinki Institute for Information Technology HIIT.
Sándor Kisfaludi-Bak: Supported by the Research Council of Finland, Grant 363444.

Acknowledgements We appreciate the insightful comments provided by the anonymous reviewers.

1 Introduction

One of the great successes in the area of geometric algorithms is the existence of polynomial-time approximation schemes (PTASes) in low-dimensional settings. The resulting running times are usually of the form $2^{f(1/\varepsilon)}\mathrm{poly}(n)$ or $n^{f(1/\varepsilon)}$. The former are called *efficient* polynomial time approximation schemes, or EPTAS for short. Once a PTAS or EPTAS running time is achieved for a problem, the research focuses on optimizing the dependence on n and $1/\varepsilon$ until a matching (conditional) lower bound is proven.

The first technique for proving EPTAS and PTAS lower bounds was introduced by Marx [25]. Under the Exponential Time Hypothesis (ETH) [18], Marx provided almost matching lower bounds for several problems on planar graphs and on intersection graphs

34th Annual European Symposium on Algorithms (ESA 2026).
Editors: Philip Bille, Seth Pettie, and Sabine Storandt; Article No. 38; pp. 38:1–38:17
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

of unit disks: for maximum independent set, the running times $2^{O(1/\varepsilon)}\mathrm{poly}(n)$ in planar graphs and $n^{O(1/\varepsilon)}$ in unit disk graphs are essentially optimal. Using the stronger Gap-ETH hypothesis [14, 23], Marx's techniques give *tight* lower bounds: they match up to constant factors in the exponent. However, the technique does not extend to higher-dimensional geometric problems.

In dimension 3 and above, a lower bound framework was devised by De Berg *et al.* [12] for exact algorithms. These results can be extended by using their Cube Wiring Theorem (see also [19]) and a careful analysis of gaps, to obtain lower bounds of the form $2^{\Omega(1/\varepsilon^{d-1})}\mathrm{poly}(n)$ under Gap-ETH, in particular for the Euclidean traveling salesman problem [21]. The same technique has been used recently for proving EPTAS lower bounds for clustering problems [8].

This higher-dimensional lower bound technique only produces lower bounds of the form $2^{\Omega(1/\varepsilon^{d-1})}\mathrm{poly}(n)$. Currently, there is no technique that yields lower bounds of the form $n^{\Omega(1/\varepsilon^{d-1})}$. On the other hand, the shifting technique of Hochbaum and Maass [17] is a common source of such running times. Shifting is a simple meta-algorithm that can be sketched as follows. Consider a randomly shifted grid consisting of hypercube cells of side length $O(1/\varepsilon)$. Obtain a naive solution near the boundary of the cells. Due to the random shift, any given constant size ball has a probability of at most $O_d(\varepsilon)$ of being intersected by a cell boundary, thus in expectation the naive solution introduces an error of at most $O_d(\varepsilon \cdot \mathrm{OPT})$. Additionally, the remaining instances in the interiors of the cells are pairwise disjoint and do not interfere, thus they can be solved separately. Each such instance has a volume of $O(1/\varepsilon^d)$ and has geometric separators of size $O(1/\varepsilon^{d-1})$, so one can solve all cell instances exactly in $n^{O(1/\varepsilon^{d-1})}$ time using techniques from parameterized exact algorithms [9]. These algorithms can be derandomized with a polynomial overhead by simply checking all shifts. Problems that have such PTASes include maximum independent set [7] and minimum dominating set [10] in intersection graphs of unit balls. This raises a natural question.

► **Question 1.** *Is the $n^{O(1/\varepsilon^{d-1})}$-time shifting algorithm for* Maximum Independent Set *in unit ball graphs optimal for every fixed $d \geq 3$?*

We note that existing parameterized lower bounds rule out an EPTAS for Maximum Independent Set. Under ETH, the problem has no $f(k)n^{o(k^{1-1/d})}$-time algorithm for any computable f [24, 26] parameterized by the solution size k. As such, if an EPTAS exists, setting $\varepsilon = 1/(2k)$ derives an $f(1/\varepsilon)\mathrm{poly}(n) = f(k)\mathrm{poly}(n)$-time algorithm, contradicting ETH. However, it is unclear how to use this result to obtain a tight lower bound.

There are two natural approaches to tackle Question 1. One is to attempt to generalize Marx's technique [25]. This technique [25], based on the so-called Matrix Tiling problem. Matrix Tiling is defined as follows. An instance of Matrix Tiling consists of integers k, D, and k^2 nonempty sets $C_{i,j} \subseteq [D] \times [D]$ for $1 \leq i, j \leq k$. The task is to select in each cell (i, j) of a $k \times k$ matrix some element $(a, b) \in C_{i,j}$, such that the first coordinates of the selected elements are equal in horizontally neighboring cells and the second coordinates are equal in vertical neighbors. The objective is to maximize the number of such equalities.[1] See Figure 1. Unfortunately, the natural extension of matrix tiling to 3 dimensions, where we require neighboring cells to agree on the corresponding coordinate, seems to behave differently: in order to find a $(1 - \varepsilon)$-approximate solution of a satisfiable instance, we can follow a simple shifting strategy. We ignore cells that intersect a randomly shifted grid of side length $1/\varepsilon$ (these will contribute to the error). The remaining cells inside each $1/\varepsilon \times 1/\varepsilon \times 1/\varepsilon$

[1] We note that the original definition of Matrix Tiling in [25] is slightly different, but equivalent to ours in terms of approximability.

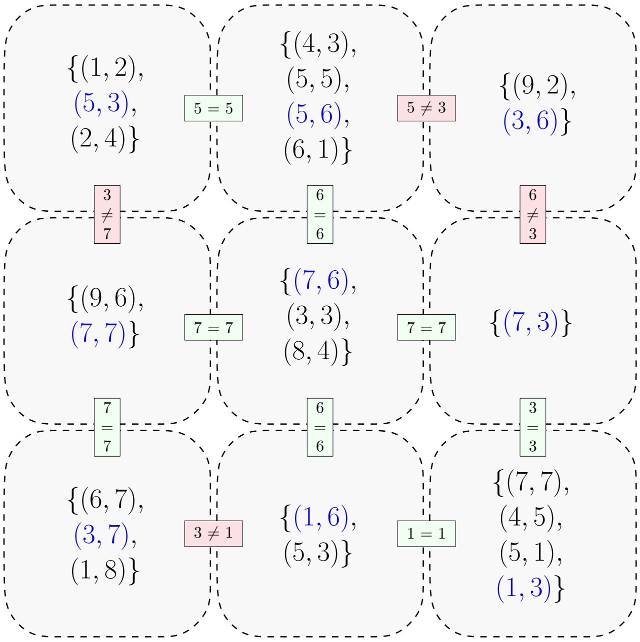

Figure 1 An instance of Matrix Tiling: the cell assignments are shown in blue; the equality and conflicting assignment on the corresponding coordinates of adjacent cells are shown in green and red respectively.

portion can be solved exactly in only $n^{O(1/\varepsilon)}$ time: indeed, one can simply try all choices of values on the $O(1/\varepsilon)$ axis-aligned slabs of cells that share one coordinate. As a result, the complexity of this problem in dimension d is likely different from $n^{\Theta(1/\varepsilon^{d-1})}$, and it is not suitable for our purposes.

The second approach requires taking a step back. Matrix Tiling is the optimization variant of the Grid Tiling problem from the exact parameterized algorithms literature, which is convenient for proving lower bounds of the form $n^{\Omega(\sqrt{k})}$ for geometric problems in the plane [9]. A higher-dimensional generalization of this exact parameterized lower bound framework was proposed by Marx and Sidiropoulos [26]; the optimization variants of these problems could serve as the starting point for the PTAS lower bound framework. Marx and Sidiropoulos defined the following variants of constraint satisfaction (CSP) problems.

Constraint Satisfaction Problems. An instance I of a *binary constraint satisfaction problem* (CSP) is a triple (V, D, C), where:

- V is a finite set of *variables*,
- D is a *domain* of values, and
- $\mathcal{C}$ is a set of *constraints*, where it contains a unary constraint $C_v \subseteq D$ for each $v \in V$ and some binary constraints $C_{u,v} \subseteq D \times D$ for distinct $u, v \in V$.

An *assignment* of I is a function $f : V \to D$. A unary constraint C_a is *satisfied* by f if $f(a) \in C_a$, and a binary constraint $C_{a,b}$ is satisfied if $(f(a), f(b)) \in C_{a,b}$. An assignment is a *solution* (or *satisfying assignment*) if it satisfies every constraint in $\mathcal{C}$. The *primal graph* of a binary CSP instance $I = (V, D, C)$ is the undirected simple graph $G = (V, E)$ where distinct vertices $u, v \in V$ satisfy $\{u, v\} \in E$ if and only if there exists a binary constraint $C_{u,v} \in \mathcal{C}$. We call a pair of distinct variables $u, v \in V$ *adjacent* if there is a binary constraint $C_{v,u} \in \mathcal{C}$.

A *d-dimensional geometric CSP* (GEOMETRIC CSP) instance $I = (V, D, C)$ is a binary CSP instance whose primal graph is isomorphic to an induced subgraph of the d-dimensional grid hypercube G_k^d for some positive integer length k, via a subgraph isomorphism $\mathsf{loc} : V \to V(G_k^d)$. Note that the constraint set $\mathcal{C}$ includes a unary constraint $C_v \subseteq D$ for each $v \in V$ and a binary constraint $C_{u,v} \subseteq D \times D$ for each edge $\{u, v\}$ in the primal graph.

A *d-dimensional tiling CSP* (TILING CSP) is a GEOMETRIC CSP instance where the domain is $D = [\Lambda]^d$ for some positive integer Λ. Similarly to GEOMETRIC CSP, the constraint set $\mathcal{C}$ includes a unary constraint $C_v \subseteq D$ for each $v \in V$. However, the binary constraints are defined *implicitly*: for every pair of adjacent variables $a, b \in V$ with $\mathsf{loc}(b) = \mathsf{loc}(a) + \vec{e}_i$ ($\vec{e}_i$ is the standard basis vector), the constraint $C_{a,b} \subseteq D \times D$ contains exactly the pairs $((x_1, \dots, x_d), (y_1, \dots, y_d)) \in [\Lambda]^d \times [\Lambda]^d$ satisfying $x_i = y_i$. A *d-dimensional $\leq$-CSP* ($\leq$-CSP) is identical to TILING CSP, except the binary constraints are of the form $x_i \leq y_i$ instead of $x_i = y_i$. A key difference from MATRIX TILING is that the primal graph in TILING CSP (or GEOMETRIC CSP) can be any induced subgraph of the grid.

In the optimization versions of all these CSPs (denoted with the prefix "MAX-"), the goal is to find an assignment that maximizes the number of satisfied constraints, or equivalently, minimizes the number of violated constraints. Marx and Sidiropoulos [26] show that TILING CSP, $\leq$-CSP and INDEPENDENT SET in ball graphs has no $f(k)n^{O(k^{1-1/d})}$-time algorithm for any computable f under ETH. It is therefore natural to ask if the maximization variants of these problems (maximizing the number of satisfied constraints) is a good basis for PTAS lower bounds.

1.1 Our contribution

We provide a framework that derives tight conditional lower bounds for geometric problems solved with the shifting technique in any constant dimension $d \geq 3$. More precisely, we prove that the maximization variants of TILING CSP and $\leq$-CSP have the desired lower bounds.

▶ **Theorem 1** (Main theorem). *Assuming Gap-ETH, for every integer $d \geq 2$ there exists a constant $\gamma > 0$ such that there is no algorithm with running time $N^{\gamma/\varepsilon^{d-1}}$ that, given any $\varepsilon > 0$ and instance $I = (V, D, \mathcal{C})$ of d-dimensional TILING CSP or $\leq$-CSP of input size $N = \mathrm{poly}(|\mathcal{C}|, |D|)$, can distinguish between the following two cases:*

(i) *there exists an assignment satisfying all constraints, and*

(ii) *every assignment violates at least an ε-fraction of the constraints.*

We obtain the above theorem using a chain of reductions from the MAX 3-SAT problem. The first step of the reduction chain in [26] involves a complicated embedding of the primal graph of the 3-SAT formula into a grid, where maintaining the gap for partially satisfied formulas is difficult. Instead, we use a much simpler embedding via the Cube Wiring Theorem of De Berg *et al.* [12]. We map variables and clauses of a formula ϕ to the d-dimensional grid graph and use a strong form of the Cube Wiring Theorem to find internally vertex-disjoint paths (wires) mimicking the incidence graph of ϕ. Intuitively, these wires will "carry" the corresponding boolean value of an assignment of ϕ. To ensure this behavior, we partition the grid graph into small cubes and assign to each cube a CSP variable whose value encodes the boolean values of the wires passing through that cube. We then use unary constraints to enforce consistency of literals and satisfaction of clauses, and binary constraints to enforce consistency of wire values across adjacent cubes. This reduction has the desired locality structure to ensure that a constant fraction of error in ϕ will propagate as a proportional fraction of error in the corresponding CSP instance.

For the rest of the steps in the reduction chain, we proceed analogously to [26], but we need to create stronger gap-preserving reduction steps. This needs some work, but can be completed after observing important locality properties of the constructions at each step.

To prove the usefulness of our main theorem, we show that the shifting-based algorithms for a wide range of problems are Gap-ETH-tight. In particular, we consider several problems in intersection graphs of unit balls. Given a set of balls in d-dimensional space, we are looking for the maximum independent set (a maximum set of pairwise disjoint balls), a minimum dominating set (a minimum subset of balls where the union of their closed graph neighborhoods is all balls), a maximum induced matching (a maximum set of balls whose induced subgraph is a matching), or a maximum induced forest (a maximum set of balls whose induced subgraph has no cycle). We also consider the piercing set problem: given a set of balls, find the minimum set of points such that each ball contains a point. The following theorem demonstrates several applications of our framework, and gives an affirmative answer to Question 1 assuming Gap-ETH.

▶ **Theorem 2** (Applications). *For any constant $d \geq 2$, Maximum Independent Set, Minimum Dominating Set, Maximum Induced Forest, Maximum Induced Matching, Minimum Piercing Set have PTASes with running time $n^{O(1/\varepsilon^{d-1})}$ on unit ball graphs in $\mathbb{R}^d$. Under Gap-ETH, for any constant $d \geq 2$ there exists some $\gamma > 0$ such that none of these problems have an $n^{\gamma/\varepsilon^{d-1}}$-time PTAS.*

The upper bounds of this theorem are either known or follow the standard shifting technique [17], paired with divide-and-conquer algorithms in the hypercubes of side length $O(1/\varepsilon)$. For the lower bounds, we use our framework: we reduce either from Max-Tiling CSP or Max-$\leq$-CSP. In case of Maximum Independent Set and Minimum Dominating Set we are able to use existing gadgetry [24, 26] without changes. For Minimum Piercing Set we rely on an intricate gadgetry introduced recently for the k-center problem by Blank *et al.* [4]. For Maximum Induced Forest and Maximum Induced Matching we introduce new gadgetry that may be of independent interest.

1.2 Further related work

The connection of approximation lower bounds and parameterized complexity has been studied before [16, 22]. More recently, Gap-ETH-based lower bounds have been an important tool in parameterized approximation algorithms, see [5, 3, 6] and the survey [15].

Shifting as an algorithmic paradigm has several variants. In the setting of planar graphs, it was first introduced by Baker [2]. A similar source of approximation schemes is the bidimensionality theory of Demaine and Hajiaghayi [13]. In the Euclidean setting, sometimes it is important to perform shifting on a hierarchical decomposition rather than a fixed grid, leading to approximation schemes that tackle objects/distances at different scales simultaneously. Typically one shifts a so-called quadtree [1, 7]. Our lower bound framework does provide lower bounds for some quadtree-based shifting algorithms, in particular we obtain a matching lower bound for Chan's algorithm [7] for independent set in intersection graphs of balls (of arbitrary radius) in Euclidean space.

2 Preliminaries

For a positive integer n, we write $[n] = \{1, 2, \ldots, n\}$ and $[n]_\circ = \{0, 1, \ldots, n-1\}$. For a point $x \in \mathbb{R}^d$, we denote its i-th coordinate by $x(i)$. The standard basis vectors of $\mathbb{R}^d$ are $\vec{e}_1, \vec{e}_2, \ldots, \vec{e}_d$. We write $\|x\|_1 = \sum_{i=1}^d |x(i)|$ for the ℓ_1-norm and $\|x\| = \sqrt{\sum_{i=1}^d x(i)^2}$ for the Euclidean norm.

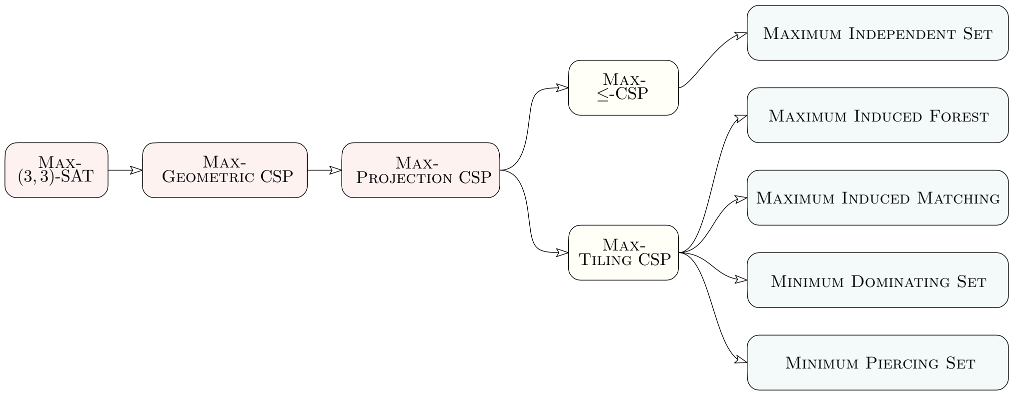

Figure 2 Reduction chain used in the proof of Theorem 1 (yellow boxes). The green boxes contain the problems for which we prove hardness in Theorem 2.

For integers $k, d > 0$, the *d-dimensional grid graph* of side length k, denoted G_k^d, is the graph with vertex set $[k]^d$ in which two vertices $a, b \in V(G_k^d)$ are adjacent if and only if $\|a - b\|_1 = 1$. A *unit ball* in $\mathbb{R}^d$ is an open ball of diameter 1. Two unit balls *intersect* if their interiors overlap; equivalently, if their centers are at Euclidean distance strictly less than 1. A *unit box* in $\mathbb{R}^d$ is an open axis-aligned hypercube of side length 1. The *intersection graph* of a collection $\mathcal{B}$ of geometric objects has one vertex per object, with two vertices adjacent whenever the corresponding objects intersect.

An *induced forest* of a graph G is a subset $S \subseteq V(G)$ such that the subgraph of G induced by S is acyclic. The Maximum Induced Forest problem asks for the largest induced forest in a given graph. In the geometric setting, the input is a collection $\mathcal{B}$ of geometric objects (such as unit balls or unit boxes in $\mathbb{R}^d$), and the goal is to find the Maximum Induced Forest of the intersection graph of $\mathcal{B}$.

A *polynomial-time approximation scheme* (PTAS) for a maximization (resp. minimization) problem is a family of algorithms parameterized by $\varepsilon > 0$ that, for every $\varepsilon > 0$, runs in time polynomial in the input size and returns a solution of value at least $(1-\varepsilon)\text{OPT}$ (resp. at most $(1+\varepsilon)\text{OPT}$).

Our hardness results rely on the following hypothesis.

▶ **Gap Exponential Time Hypothesis (Gap-ETH)** (Dinur [14], Manurangsi and Raghavendra [23]). *There exist constants $\delta, \gamma > 0$ such that there is no $2^{\gamma m}$-time algorithm which, given a 3-CNF formula ϕ on m clauses, can distinguish between the cases where*

(i) *ϕ is satisfiable, or*
(ii) *all assignments violate at least δm clauses.*

A Boolean formula ϕ in CNF is a (a, b)-CNF formula if each variable occurs at most a times and each clause has at most b literals. We consider the Max-$(3,3)$-SAT problem, in which the objective is to maximize the number of satisfied clauses in a $(3,3)$-CNF formula. Note that for $(3,3)$-CNF formulas, the numbers of variables and clauses are within constant factors of each other. Papadimitriou [27] gives an L-reduction from Max-3-SAT to Max-$(3,3)$-SAT, which implies the following:

▶ **Lemma 3** (Corollary of [27]). *There exist constants $\delta, \gamma > 0$ such that, unless Gap-ETH fails, there is no algorithm with running time $2^{\gamma n}$ that, given a $(3,3)$-CNF formula ϕ with n variables and m clauses, can distinguish between the following two cases:*

(i) *ϕ is satisfiable; and*
(ii) *for every assignment at least δm clauses of ϕ are violated.*

3 From (3,3)-SAT to Geometric CSP

In this section, we prove a hardness result for Max-Geometric CSP on d-dimensional grids ($d \geq 2$) via an approximation-preserving reduction from Max-$(3,3)$-SAT. To bridge the gap between the combinatorial structure of a $(3,3)$-CNF formula ϕ and the geometry of a d-dimensional grid in Geometric CSP, we utilize a strong form of the Cube Wiring Theorem [19] to embed the incidence graph of ϕ on a d-dimensional grid for $d \geq 3$.

▶ **Theorem 4** (Strong Cube Wiring Theorem [19]). *Let $d \geq 3$ and let $G = (A \cup B, E)$ be a bipartite graph on n vertices of maximum degree at most $2d$. Let $n_1 = \Theta\left(n^{1/(d-1)}\right)$ be an integer. Then there exists a constant $c > 0$ and injective mappings*

$$\mathsf{Emb}_A : A \to [cn_1]^{d-1} \times \{1\} \subseteq V(G^d_{cn_1}), \qquad \mathsf{Emb}_B : B \to [cn_1]^{d-1} \times \{cn_1\} \subseteq V(G^d_{cn_1}),$$

such that the following holds. There exists a collection of $|E|$ internally vertex-disjoint paths $\mathcal{W} = \{W_e\}_{e \in E}$ in the d-dimensional grid graph $G^d_{cn_1}$ with the property that for every edge $e = \{a, b\} \in E$ with $a \in A$ and $b \in B$, the path W_e connects $\mathsf{Emb}_A(a)$ to $\mathsf{Emb}_B(b)$. We refer to these paths as wires. Moreover, given G and d the embedding can be constructed in polynomial time.

For $d = 2$, we replace vertex-disjoint wires in Theorem 4 with an *edge-disjoint* wiring in a 2-dimensional grid (allowing crossings, but no shared edges) as in [19]. See the proof of the following lemma in the full version.

▶ **Lemma 5** (2-dimensional Grid Wiring). *Let $G = (A \cup B, E)$ be a bipartite graph on n vertices with maximum degree at most 3 and $|E| = m$. Then there exists an integer $n_1 = \Theta(n + m)$, a constant $c > 0$, injective mappings*

$$\mathsf{Emb}_A : A \to [cn_1] \times \{1\} \subseteq V(G^2_{cn_1}), \qquad \mathsf{Emb}_B : B \to [cn_1] \times \{cn_1\} \subseteq V(G^2_{cn_1}),$$

and a collection $\mathcal{W} = \{W_e\}_{e \in E}$ of edge-disjoint *grid paths in $G^2_{cn_1}$ such that each W_e connects $\mathsf{Emb}_A(a)$ to $\mathsf{Emb}_B(b)$ for any edge $e = \{a, b\} \in E$ and every grid vertex belongs to at most three wires; if two wires meet at a grid vertex then they do not share any grid edge (they only cross). Moreover, given G, the embedding can be constructed in polynomial time.*

The main result of this section is the following hardness theorem where we present an approximation-preserving reduction from Max-$(3,3)$-SAT to Max-Geometric CSP on d-dimensional grids, for every $d \geq 2$.

▶ **Theorem 6.** *Assuming Gap-ETH, for every fixed integer $d \geq 2$ there exists a constant $\gamma > 0$ such that there is no algorithm with running time $N^{\gamma/\varepsilon^{d-1}}$ that, given any $\varepsilon > 0$ and instance $I = (V, D, \mathcal{C})$ of d-dimensional Geometric CSP of input size $N = \mathrm{poly}(|\mathcal{C}|, |D|)$, can distinguish between the following two cases:*

(i) *there exists an assignment satisfying all constraints in I, and*
(ii) *every assignment violates at least $\varepsilon|\mathcal{C}|$ constraints.*

Proof. Fix a positive integer $d \geq 2$. Let ϕ be a $(3,3)$-CNF formula. We construct an instance of Geometric CSP whose variables correspond to blocks of a d-dimensional grid, and whose domain encodes Boolean values propagated along vertex-disjoint paths (edge-disjoint paths for the case $d = 2$). The key ingredients of the reduction are as follows.

- We consider the incidence graph of ϕ, viewed as a bipartite graph between variables and clauses. Using the Strong Cube Wiring Theorem (Theorem 4) for $d \geq 3$ and 2-dimensional Edge-Disjoint Wiring (Lemma 5) for $d = 2$, we embed this bipartite graph into a d-dimensional grid so that each variable and clause incidence is represented by a disjoint path (a *wire*).

- We partition the grid into small cubes and assign to each cube a variable whose value encodes the Boolean values of the wires passing through that cube. Unary constraints enforce consistency of literals belonging to the same variable and satisfaction of clauses, while binary constraints enforce consistency of wire values across adjacent cubes.

We show that a satisfiable formula ϕ yields fully satisfiable Geometric CSP instances, whereas assignments violating a constant fraction of clauses in ϕ correspond to assignments violating a proportional fraction of constraints in the Geometric CSP instance.

Embedding the incidence graph of ϕ

Assume that the $(3,3)$-CNF formula ϕ has variable set Var_ϕ and clause set Cla_ϕ. We associate with ϕ its *incidence graph* $G = (A \cup B, E)$, where $A = \mathsf{Var}_\phi$, $B = \mathsf{Cla}_\phi$, and $\{\mathsf{va}, \mathsf{cl}\} \in E$ if and only if the variable $\mathsf{va} \in \mathsf{Var}_\phi$ appears in the clause $\mathsf{cl} \in \mathsf{Cla}_\phi$. Since ϕ is a $(3,3)$-CNF formula, each variable appears in at most three clauses and each clause contains at most three literals. Consequently, the maximum degree of G is at most 3.

For any fixed $d \geq 3$, we apply the Strong Cube Wiring Theorem (Theorem 4) and for $d = 2$, we apply the 2-dimensional Edge-Disjoint Wiring (Lemma 5) to embed G into a d-dimensional grid. See Figure 3. Let $n_1 = \Theta\big(n^{1/(d-1)}\big)$ be the grid side length parameter guaranteed by Theorem 4 (or Lemma 5). Moreover, it provides injective mappings

$$\mathsf{Emb}_A : A \to [cn_1]^{d-1} \times \{1\} \qquad \text{and} \qquad \mathsf{Emb}_B : B \to [cn_1]^{d-1} \times \{cn_1\},$$

for a constant $c > 0$, together with a collection of vertex-disjoint paths (edge-disjoint for $d = 2$) $\mathcal{W} = \{W_e\}_{e \in E}$ in the grid graph $G^d_{cn_1}$.

Note that the size of the grid is $\Theta\big(n^{d/(d-1)}\big)$. Each path $W_e \in \mathcal{W}$ connects $\mathsf{Emb}_A(\mathsf{va})$ to $\mathsf{Emb}_B(\mathsf{cl})$ for the corresponding incidence $e = \{\mathsf{va}, \mathsf{cl}\} \in E$. Recall that we refer to these paths as *wires*. Intuitively, each wire is responsible for transmitting the Boolean value assigned to a literal from its variable vertex to the corresponding clause vertex.

Constructing an instance of Geometric CSP

We first define the variables and domain of the Geometric CSP instance. Let s be a positive integer parameter. We define an *s-partition* of $G^d_{cn_1}$ by grouping vertices into axis-aligned d-dimensional cubes of side length s. Formally, define a mapping $\psi : V(G^d_{cn_1}) \to V(G^d_{n_2})$, by

$$\psi(x_1, \ldots, x_d) = \big(\lceil x_1/s \rceil, \ldots, \lceil x_d/s \rceil\big),$$

where $n_2 = \left\lceil \frac{cn_1}{s} \right\rceil$. For each $v \in V(G^d_{n_2})$, the preimage $\psi^{-1}(v)$ induces a connected subgraph of $G^d_{cn_1}$ isomorphic to a d-dimensional grid of side length at most s. We refer to such a subgraph as a *small cube* and we denote the set of all small cubes by Σ. The variables V of the Geometric CSP instance correspond to the small cubes in Σ, i.e., $V = \{v_\sigma : \sigma \in \Sigma\}$. Naturally, we define the subgraph isomorphism $\mathsf{loc} : V \to V(G^d_{n_2})$ by $\mathsf{loc}(v_\sigma) := \psi(z)$ for an arbitrary vertex z of σ. We say two small cubes σ and σ' are adjacent if $\mathsf{loc}(v_\sigma)$ and $\mathsf{loc}(v_{\sigma'})$ are adjacent in $G^d_{n_2}$. See Figure 3.

Recall that the collection $\mathcal{W}$ of wires consists of internally vertex-disjoint paths in $G^d_{cn_1}$ (edge-disjoint paths for $d = 2$). For a small cube $\sigma \in \Sigma$, let $\mathcal{W}_\sigma \subseteq \mathcal{W}$ denote the set of wires that intersect σ. We bound the number of such wires as follows. Each wire enters or exits a small cube through one of its $2d$ facets, each facet containing at most s^{d-1} vertices. Since the wires are internally vertex-disjoint, at most three wires can occupy each boundary vertex (for $d > 2$, only when it is an extreme vertex of a wire otherwise it is at most two, and for $d = 2$, Lemma 5 ensures this property also holds for the edge-disjoint paths). Consequently, $|\mathcal{W}_\sigma| \leq 6d \cdot s^{d-1}$.

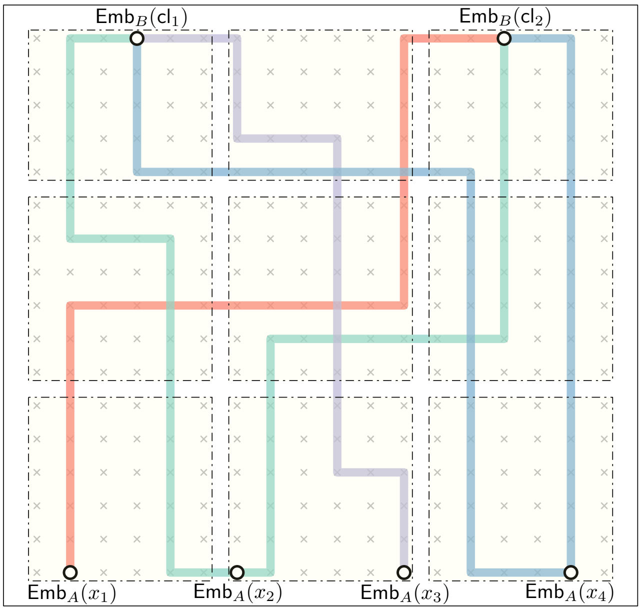

Figure 3 An s-partition of a 2-dimensional embedding of the incidence graph of the formula $\phi = (\bar{x}_2 \vee x_3 \vee x_4) \wedge (\bar{x}_1 \vee x_2 \vee \bar{x}_4)$ for $s = 6$. The wires corresponding to the same variable share the same color.

As a shorthand, we define $\kappa := 6d \cdot s^{d-1}$. The domain of the Geometric CSP instance is defined as $D := \{0,1\}^\kappa$. Intuitively, each coordinate of a domain element corresponds to the Boolean value carried by one of the wires passing through the corresponding small cube. For each small cube σ, we fix an arbitrary injective mapping $f_\sigma : \mathcal{W}_\sigma \to [\kappa]$, which assigns to each wire intersecting σ a unique coordinate of the domain vector.

Finally, we define the set of constraints $\mathcal{C}$ of the Geometric CSP instance. The constraints are of three types:

(i) Variable consistency constraints. Let σ be a small cube that contains the embedded vertex $\mathsf{Emb}_A(\mathsf{va})$ corresponding to a variable $\mathsf{va} \in \mathsf{Var}_\phi$. Let $L(\mathsf{va})$ denote the set of literals of ϕ in which va participates. For the variable $v_\sigma \in V$, we introduce a unary constraint requiring that all wires corresponding to occurrences of va encode a consistent Boolean assignment.

Formally, the allowed domain values at $v_\sigma \in V$ are those vectors $x \in \{0,1\}^\kappa$ such that for all literals $l, l' \in L(\mathsf{va})$ with corresponding wires W and W',

$$x(f_\sigma(W)) = \begin{cases} x(f_\sigma(W')) & \text{if } l \text{ and } l' \text{ have the same sign,} \\ 1 - x(f_\sigma(W')) & \text{otherwise.} \end{cases}$$

(ii) Clause satisfaction constraints. Let σ be a small cube that contains the embedded vertex $\mathsf{Emb}_B(\mathsf{cl})$ corresponding to a clause $\mathsf{cl} \in \mathsf{Cla}_\phi$. Let $L(\mathsf{cl}) = \{l_1, l_2, l_3\}$ be the (at most three) literals appearing in the clause cl. For the variable v_σ, we introduce a unary constraint requiring that at least one of these literals evaluates to true.

Formally, the allowed domain values at $v_\sigma \in V$ are those vectors $x \in \{0,1\}^\kappa$ such that if W_1, W_2, and W_3 are the corresponding wires of the literals l_1, l_2, and l_3 respectively, then

$$\big(x(f_\sigma(W_1)), x(f_\sigma(W_2)), x(f_\sigma(W_3))\big) \neq (0,0,0).$$

(iii) Wire consistency constraints. Finally, consider two adjacent small cubes $\sigma, \sigma' \in \Sigma$. For every wire $W \in \mathcal{W}_\sigma \cap \mathcal{W}_{\sigma'}$, the Boolean value carried by W must be consistent across both cubes. We introduce a binary constraint between v_σ and $v_{\sigma'}$ requiring equality on all shared wire coordinates.

Formally, the allowed pairs $(x, y) \in \{0,1\}^\kappa \times \{0,1\}^\kappa$ at the binary constraint between v_σ and $v_{\sigma'}$ satisfy

$$x(f_\sigma(W)) = y(f_{\sigma'}(W)) \qquad \text{for all } W \in \mathcal{W}_\sigma \cap \mathcal{W}_{\sigma'}.$$

This completes the definition of the constraint set $\mathcal{C}$.

Exact Equivalence and Error analysis

We show that if the Boolean formula ϕ is satisfiable, then the constructed Geometric CSP instance admits an assignment that satisfies all constraints. Let $\alpha : \mathsf{Var}_\phi \to \{0,1\}$ be a solution of ϕ. We define an assignment $\mu : V \to \{0,1\}^\kappa$ to the Geometric CSP variables as follows. Each literal l of ϕ corresponds to a unique wire $W_l \in \mathcal{W}$ in the grid embedding. For each small cube σ and each wire $W_l \in \mathcal{W}_\sigma$, where l is a literal over variable va, we set

$$\mu(v_\sigma)\big(f_\sigma(W_l)\big) = \begin{cases} \alpha(\mathsf{va}) & \text{if } l \text{ is a positive occurrence of the variable } \mathsf{va}, \\ 1 - \alpha(\mathsf{va}) & \text{if } l \text{ is a negated occurrence of the variable } \mathsf{va}. \end{cases}$$

All remaining coordinates of $\mu(v_\sigma)$ (corresponding to wires not intersecting σ) may be assigned arbitrarily. By construction, μ satisfies all constraints of the Geometric CSP instance.

Now consider an arbitrary assignment $\lambda : V \to \{0,1\}^\kappa$ to the Geometric CSP instance. We define a Boolean assignment $\alpha : \mathsf{Var}_\phi \to \{0,1\}$ for the formula ϕ as follows. For each variable $\mathsf{va} \in \mathsf{Var}_\phi$, let $\sigma(\mathsf{va})$ be the unique small cube containing the embedded vertex $\mathsf{Emb}_A(\mathsf{va})$. If the variable-consistency constraint at $\sigma(\mathsf{va})$ is satisfied by λ, then all wires corresponding to occurrences of va encode a consistent value; we set $\alpha(\mathsf{va})$ to that value. Otherwise, we assign $\alpha(\mathsf{va})$ arbitrarily. Again by construction, if λ is a solution of the Geometric CSP instance, then α is a solution of ϕ.

$\triangleright$ Claim 7. Each violated constraint in the assignment λ of the Geometric CSP can cause at most 6κ violated clauses in the corresponding assignment α for the formula ϕ.

Proof. Any violated constraint (unary or binary) involves at most two small cubes (a pair of adjacent small cubes in case of binary constraint) whose value assignment may corrupt the Boolean values carried by all wires intersecting the cube(s). Each such cube intersects at most κ wires, and each wire corresponds to a literal occurrence that appears in at most three clauses. Therefore, each violated constraint can affect at most 6κ clauses of ϕ. $\triangleleft$

The number of variables, hence the number of constraints up to a constant factor, is

$$|\mathcal{C}| = \Theta(n_2^d) = \Theta\left(\left(\frac{n_1}{s}\right)^d\right).$$

Moreover, for $(3,3)$-formulas we have $m = \Theta(n)$ and $|E| = \Theta(n)$. Suppose that at most $\varepsilon|\mathcal{C}|$ constraints are violated by the arbitrary Geometric CSP assignment λ. Then by Claim 7 the number of clauses violated in the induced Boolean assignment α is at most

$$6\kappa \cdot \varepsilon|\mathcal{C}| = \Theta(s^{d-1}) \cdot \varepsilon \cdot \Theta\left(\left(\frac{n_1}{s}\right)^d\right) = \varepsilon \cdot \Theta\left(\frac{n_1^d}{s}\right) = \varepsilon \cdot \Theta\left(\frac{n^{d/(d-1)}}{s}\right).$$

Assume that Gap-ETH holds and γ, δ be the two constants from Lemma 3. By Claim 7, setting $\varepsilon := \delta\, s\, c_d / n^{1/(d-1)}$ for an appropriate constant $c_d > 0$ yields that if every assignment of ϕ violates at least δm clauses, every assignment to the GEOMETRIC CSP instance violates at least $\varepsilon|\mathcal{C}|$ constraints.

The constructed GEOMETRIC CSP instance has $|\mathcal{C}| = \Theta\left(n^{d/(d-1)}/s^d\right)$ and domain size $|D| = 2^\kappa = 2^{\Theta(s^{d-1})}$ and it is obtained from ϕ in polynomial time in the instance size $N = \text{poly}(|\mathcal{C}|, |D|)$. Suppose there exists an algorithm that, for every $\gamma_0 > 0$, distinguishes satisfiable GEOMETRIC CSP instances from those in which every assignment violates at least $\varepsilon|\mathcal{C}|$ constraints in time $N^{\gamma_0/\varepsilon^{d-1}}$. Applying this algorithm to the instance produced by the reduction for values of $s = \Omega(\log^{1/(d-1)} n)$ yields an algorithm that distinguishes satisfiable $(3,3)$-CNF formulas from those in which every assignment violates at least δm clauses in time

$$\begin{aligned}
\text{poly}\big(|\mathcal{C}|, |D|\big)^{\Theta(\gamma_0 n/s^{d-1})} &\le \left(2^{\Theta(s^{d-1})} \cdot \frac{n^{d/(d-1)}}{s^d}\right)^{\Theta(\gamma_0 n/s^{d-1})} \\
&= 2^{\Theta(s^{d-1})\cdot\Theta(\gamma_0 n/s^{d-1})} \cdot \left(\frac{n^{d/(d-1)}}{s^d}\right)^{\Theta(\gamma_0 n/s^{d-1})} \\
&= 2^{\gamma_0 \Theta(n)} \cdot 2^{\Theta\left(\frac{\gamma_0 n \log(n/s)}{s^{d-1}}\right)} \\
&= 2^{\gamma_0 c'_d n},
\end{aligned}$$

for some constant $c'_d > 0$. Choosing $\gamma_0 > 0$ sufficiently small contradicts Lemma 3. ◀

Following the reduction chain described in Figure 2, we obtain the proof of Theorem 1 by showing that the reductions in Proposition 2.18 and Proposition 2.19 of [26, Section 2.3] are approximation preserving. See the full version for the details.

4 Hardness of Approximation via Tiling CSP and $\le$-CSP

As applications of Theorem 1, we derive running-time lower bounds for approximation schemes for MAXIMUM INDEPENDENT SET, MAXIMUM INDUCED FOREST, MAXIMUM INDUCED MATCHING, MINIMUM DOMINATING SET, and MINIMUM PIERCING SET on unit balls (and on unit cubes) in $\mathbb{R}^d$. Our framework is based on reductions from either MAX-TILING CSP or MAX-$\le$-CSP. For MAXIMUM INDEPENDENT SET and MINIMUM DOMINATING SET, we can directly employ existing gadget constructions [24, 26], whereas MINIMUM PIERCING SET requires the more involved gadgetry recently developed for the k-center problem by Blank *et al.* [4]. Here, we give a sketch of the reduction for MAXIMUM INDUCED FOREST in Theorem 8, and refer to the full version for complete proofs of all results. As a matching upper bound, in the full version, we discuss shifting based algorithms for these problems as well.

▶ **Theorem 8.** *Let $d \ge 2$ be an integer. There is a real $\gamma > 0$ such that if the* MAXIMUM INDUCED FOREST *of n unit balls in $\mathbb{R}^d$ admits a PTAS with running time $n^{\gamma/\varepsilon^{d-1}}$ then Gap-ETH fails.*

Proof sketch. We adapt the gadget construction of Marx [25] given for MAXIMUM INDEPENDENT SET on unit disks in the plane to the MAXIMUM INDUCED FOREST problem on unit balls in $\mathbb{R}^d$, and reduce from TILING CSP (Theorem 1). We first recall the gadgetry of Marx's construction, then describe our construction for the MAXIMUM INDUCED FOREST instance for $d = 2$, and finally explain how the cross-polytope structure of the box boundary allows an extension of our instance to any dimension $d \ge 3$.

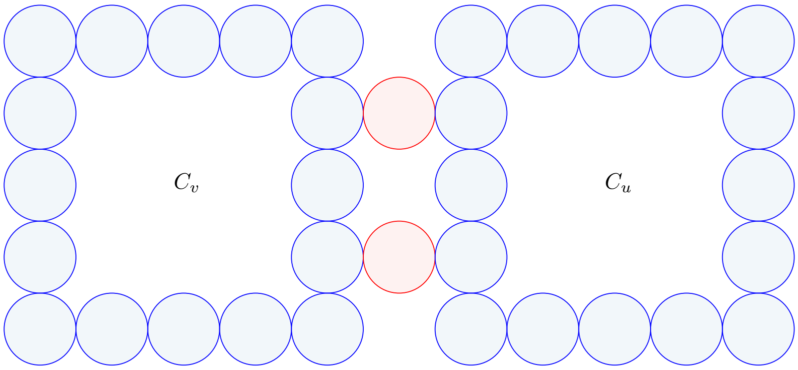

Figure 4 Marx's gadget [25] for Maximum Independent Set of unit disks via Matrix Tiling. The blue and red disks represent clique positions along the variable gadget and the connector cliques, respectively.

Marx's construction for Maximum Independent Set on unit disks. In [25], Marx reduces Matrix Tiling to Maximum Independent Set on unit disks in the plane. Each unary constraint C_v of the Matrix Tiling instance is implemented by a *variable gadget* consisting of 16 cliques (the blue disks in Figure 4) of unit disks arranged around the boundary of an axis-aligned square. For each value $(x, y) \in C_{i,j}$ (encoded as a single index $r = \iota(x, y)$ via a fixed bijection ι), the gadget places one unit disk at each of the 16 clique positions, with small perturbations along the boundary depending on r. The perturbations are chosen so that, for two consecutive clique positions p_j and p_{j+1}, the disks placed for values r and r' intersect if and only if $r > r'$. Selecting independent disks from each of the 16 positions therefore forces a cyclic chain $r_1 \leq r_2 \leq \cdots \leq r_{16} \leq r_1$, which collapses to a common value r: the *value represented by the gadget*.

The gadgets are placed on a grid, and gadgets corresponding to adjacent variables are tied together by *two* short sequences of *connection disks* (the red disks in Figure 4). The connection disks are positioned so that they create two cliques between the adjacent variable gadgets. A connection admits independent disks only when the two incident gadgets agree on the relevant coordinate: one connection clique enforces the $\leq$-relation, while the other enforces the $\geq$-relation. Together, they ensure equality: horizontal connections enforce agreement on the first coordinate, and vertical connections on the second coordinate. An independent set of size $16|V| + 2|E|$, induces a consistent assignment of values to variables, and thus a solution of the Matrix Tiling instance. Conversely, every Matrix Tiling solution yields an independent set of size $16|V| + 2|E|$ by selecting, in each gadget, the 16 disks indexed by its assigned value together with two compatible disks from each connection. Cliques prevent picking two disks at the same clique position, and pairwise disjointness across the gadget cycle and across connections enforce consistency and coordinate-matching.

Our construction for Maximum Induced Forest on unit balls, case $d = 2$. Our construction is inspired by [25] and [11]. It reduces from Tiling CSP. Let $I = (V, \mathcal{D}, \mathcal{C})$ be the input Tiling CSP instance with the primal graph G. Fix a spanning tree T of G.[2] We obtain a collection $\mathcal{B}$ of unit balls from I. We follow a similar global structure as seen for Maximum

[2] We assume that the primal graph G is connected. Otherwise, the same construction can be applied to each connected component separately.

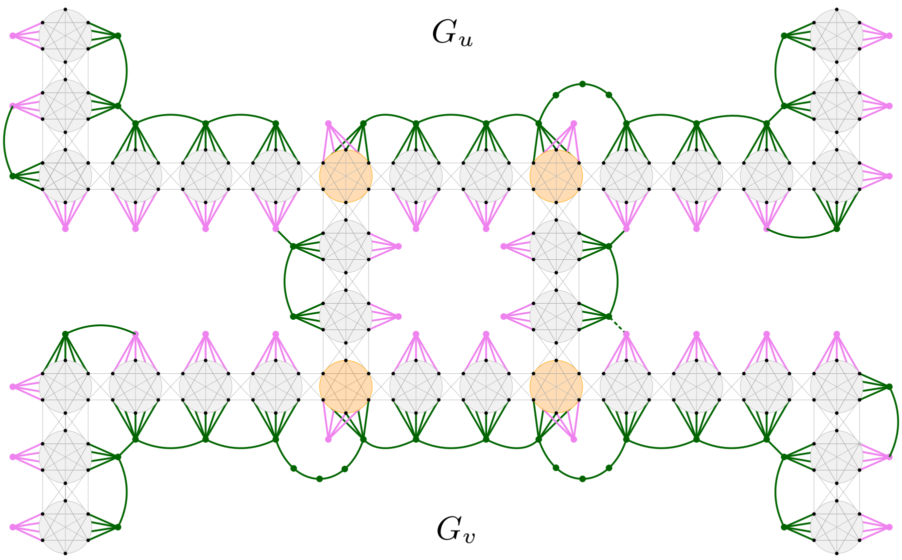

Figure 5 The vertices corresponding to clique balls (gray regions), forest ball (depicted in green), private balls (depicted in pink) are represented in the intersection graph of the instance at the neighborhood of two vertically adjacent variables gadgets G_u and G_v. The dashed edge in this figure represents a potential edge of spanning tree T in order to obtain the global forest.

Independent Set above, by replacing the 16 clique positions with $\tau = 44$ positions around an axis-aligned square Q_v of side length 11 and replacing the connector cliques with a chain of two adjacent cliques. See Figure 6. Our goal is to preserve the two structural properties in Marx's reduction: The *independent selection* along the cycle of the variable gadget and the *coordinate consistency* across connectors .

In the induced forest setting, neither property is enforced automatically: a forest can contain two balls from the same clique (as long as no triangle is created). We therefore need additional gadgetry that (i) limits each clique position to at most one selected ball, and (ii) penalizes intersections between selected balls at consecutive positions by turning them into cycles. We enforce both properties using two auxiliary ball types per clique position, as follows. For each clique position p_j along the boundary of a variable gadget, we place a *private ball* $h_j(v)$ slightly exterior to Q_v, intersecting exactly the clique balls and nothing else; and a *forest ball* $t_j(v)$ slightly interior to Q_v, also intersecting exactly the clique balls at p_j. A similar pair of private and forest balls is considered for each clique position along the connectors. The forest balls form a global forest structure that helps to enforce independent selections in the cliques. See Figure 5.

The private ball (or the forest ball) at p_j implements property (i): any forest containing $h_j(v)$ and two clique balls at p_j would contain a triangle which contradicts the forest property, so at most one clique ball at p_j can be selected together with $h_j(v)$. Note that for a given clique position p_j, selecting a forest ball $t_j(v)$, a private ball $h_j(v)$, and one ball from the clique at position p_j contributes 3 balls to an induced forest. In contrast, excluding $t_j(v)$ and $h_j(v)$ allows for the selection of at most 2 balls from the clique at position p_j.

▷ Claim 9. Let F be an induced forest of $\mathcal{B}$ that contains a private ball $h_j(v)$ or a forest ball $t_j(v)$ at position p_j. Then F can contain at most one clique ball at position p_j.

To implement property (ii), we connect the forest balls across the whole instance into a single *global forest* $\mathcal{F}$. Once all forest balls are included in the forest $\mathcal{F}$, two intersecting selected balls at adjacent clique positions would close a cycle through the forest ball path

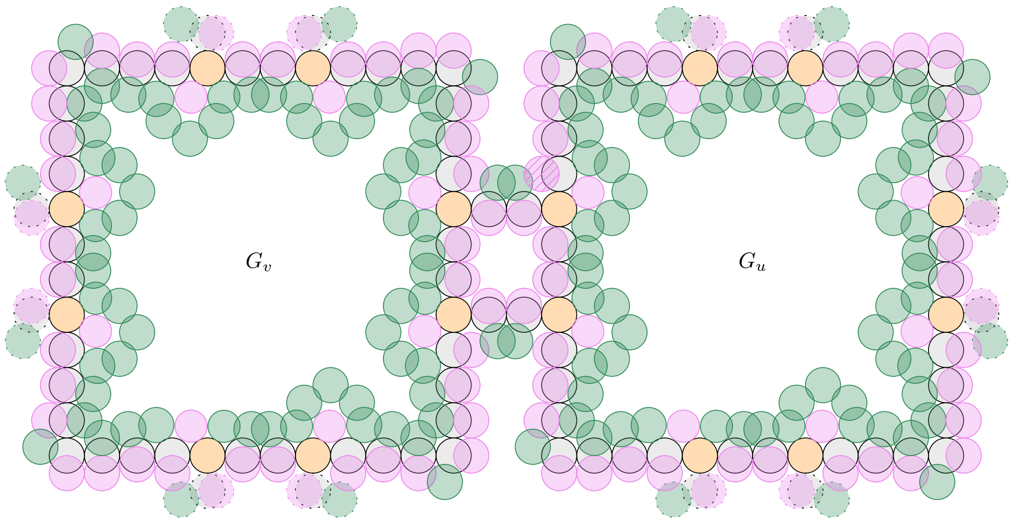

Figure 6 An instance of unit disks obtained from two adjacent variables of a 2-dimensional Tiling CSP where their corresponding edge appears in the spanning tree T. The green, violet and orange colors represent the forest balls, private balls, port positions, respectively. The violet colored disk with parallel stripes represents the private ball in G_u that ensures the connectivity of the global forest.

connecting their positions in $\mathcal{F}$. Hence selection at consecutive clique positions must be disjoint, and this yields exactly the cyclic-ordering and coordinate-agreement conditions needed to recover an assignment as in the reduction to Maximum Independent Set.

▷ Claim 10. Let F be an induced forest of $\mathcal{B}$ containing all forest balls of $\mathcal{F}$. Then clique balls in distinct clique positions are disjoint. Moreover, if F is maximal, all private balls are in F.

We obtain this global forest $\mathcal{F}$ as follows: within each gadget, the forest balls are linked into a tree T_v using $O(1)$ additional balls routed through the interior of Q_v. Across connectors, the forest balls of the two intermediate cliques form a short path and globally, the trees of gadgets are stitched together using the fixed spanning tree T of the primal graph G. For each edge $\{a, b\} \in E(T)$, one of the two connector chains is designated as a *bridge* whose forest ball path connects T_a and T_b through neighboring private balls, while the other chain is attached as a pendant subtree to T_a only. For edges of the primal graph not in T, both chains are attached as pendant subtrees to T_a.

Figure 6 illustrates the placement of the different types of balls for two neighboring variable gadgets and their connectors. The clique positions are placed at unit spacing along the boundary of Q_v, with 12 positions per side and corners shared between adjacent sides. Among these, 8 are designated as *ports* (two per side) that serve as endpoints of the two connector chains to each neighboring gadget. Unlike the other positions, which sit exactly on the boundary of Q_v, each port position is shifted slightly inward along the inward normal. This inward shift creates just enough room to place the private ball of a port position *inside* Q_v without having it intersect any clique ball of an adjacent position. At each clique position, for every $s \in C_v$, we place a clique ball $B_j(v, s)$ whose center is perturbed along the boundary direction by an amount depending on s; at port positions, an additional normal-direction perturbation encodes the coordinate of s relevant to the incident connector. Forest balls at corner positions are placed slightly outside Q_v (rather than inside, as in non-corner positions) due to space constraints, and are reconnected to the global forest $\mathcal{F}$ through the private ball of the adjacent non-corner position.

Extension to $d \geq 3$. For fixed a dimension at least 3, the variable gadget is built around an axis-aligned box Q_v of side length 30 with $2d$ facets. The clique positions are arranged along a directed Hamiltonian cycle H_v near the boundary of Q_v that visits the facets in the order given by a Hamiltonian cycle of the *cross-polytope* dual to Q_v, similarly to [20]: consecutive facets in the cycle have orthogonal outward normals. On each facet F_i, the cycle enters, traces a rectilinear path of unit-spaced adjacent points that includes a designated port (branching) positions for two connector chains. The clique, private, and forest balls are defined with the same intersection pattern as before.

Let g denote the number of forest balls per gadget and recall that τ is the total number clique position in H_v. Set $\Gamma = (g + 2\tau)|V| + 12|E|$. Using Claims 9 and 10, one shows that I admits a satisfying assignment if and only if $\mathcal{B}$ has an induced forest of size Γ. Moreover, whenever every assignment to I violates a δ-fraction of constraints, a standard accounting of variables whose gadget or incident connectors are not consistent (i.e., the forest is not locally maximal), yields $\textsc{mif}(\mathcal{B}) \leq (1 - c'_d\,\delta)\,\Gamma$, for a constant $c'_d > 0$ depending only on d. Therefore, a PTAS for Maximum Induced Forest on unit balls in $\mathbb{R}^d$ with running time $n^{\gamma_0/\varepsilon^{d-1}}$, invoked on $\mathcal{B}$ with $\varepsilon = c'_d\,\delta/2$, would distinguish the satisfiable case from the case where every assignment violates δ-fraction of the constraints. Thus, for γ_0 small enough, this contradicts Theorem 1. ◀

References

1 Sanjeev Arora. Polynomial Time Approximation Schemes for Euclidean TSP and Other Geometric Problems. *Journal of the ACM*, 45(5):753–782, 1998. doi:10.1145/290179.290180.

2 Brenda S. Baker. Approximation algorithms for NP-complete problems on planar graphs. *Journal of the ACM*, 41(1):153–180, 1994. doi:10.1145/174644.174650.

3 Arnab Bhattacharyya, Suprovat Ghoshal, Karthik C. S., and Pasin Manurangsi. Parameterized Intractability of Even Set and Shortest Vector Problem from Gap-ETH. In *45th International Colloquium on Automata, Languages, and Programming (ICALP)*, volume 107 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 17:1–17:15, 2018. doi:10.4230/LIPIcs.ICALP.2018.17.

4 Lotte Blank, Karl Bringmann, Parinya Chalermsook, Karthik C. S., Benedikt Kolbe, Hung Le, and Geert van Wordragen. Fine-grained complexity of continuous euclidean k-center. In Aditya Bhaskara and Artur Czumaj, editors, *Proceedings of the 58th Annual ACM Symposium on Theory of Computing, STOC 2026, Salt Lake City, UT, USA, June 22-26, 2026*, pages 330–341. ACM, 2026. doi:10.1145/3798129.3800753.

5 Édouard Bonnet, Bruno Escoffier, Eun Jung Kim, and Vangelis Th. Paschos. On subexponential and FPT-time inapproximability. *Algorithmica*, 71(3):541–565, 2015. doi:10.1007/s00453-014-9881-x.

6 Parinya Chalermsook, Marek Cygan, Guy Kortsarz, Bundit Laekhanukit, Pasin Manurangsi, Danupon Nanongkai, and Luca Trevisan. From gap-exponential time hypothesis to fixed parameter tractable inapproximability: Clique, dominating set, and more. *SIAM J. Comput.*, 49(4):772–810, 2020. doi:10.1137/18M1166869.

7 Timothy M. Chan. Polynomial-time approximation schemes for packing and piercing fat objects. *J. Algorithms*, 46(2):178–189, 2003. doi:10.1016/S0196-6774(02)00294-8.

8 Vincent Cohen-Addad, Karthik C. S., David Saulpic, and Chris Schwiegelshohn. Almost-optimal upper and lower bounds for clustering in low dimensional euclidean spaces, 2026. doi:10.48550/arXiv.2603.09846.

9 Marek Cygan, Fedor V. Fomin, Łukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, Cham, 2015. doi:10.1007/978-3-319-21275-3.

10 Minati De, Gautam K. Das, Paz Carmi, and Subhas C. Nandy. Approximation Algorithms for a Variant of discrete Piercing Set Problem for Unit Disks. *Int. J. Comput. Geom. Appl.*, 23(6):461, 2013. doi:10.1142/S021819591350009X.

11 Mark de Berg, Hans L. Bodlaender, and Sándor Kisfaludi-Bak. The homogeneous broadcast problem in narrow and wide strips II: lower bounds. *Algorithmica*, 81(7):2963–2990, 2019. doi:10.1007/S00453-019-00561-0.

12 Mark de Berg, Hans L. Bodlaender, Sándor Kisfaludi-Bak, Dániel Marx, and Tom C. van der Zanden. A framework for exponential-time-hypothesis-tight algorithms and lower bounds in geometric intersection graphs. *SIAM Journal on Computing*, 49(6):1291–1331, 2020. doi:10.1137/18M1167733.

13 Erik D. Demaine and MohammadTaghi Hajiaghayi. Bidimensionality: New connections between FPT algorithms and PTASs. In *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 590–601. SIAM, 2005. URL: https://dl.acm.org/citation.cfm?id=1070432.1070514.

14 Irit Dinur. Mildly exponential reduction from gap-3SAT to polynomial-gap label-cover. *Electronic Colloquium on Computational Complexity (ECCC)*, 23:128, 2016. URL: https://eccc.weizmann.ac.il/report/2016/128/.

15 Andreas Emil Feldmann, Karthik C. S., Euiwoong Lee, and Pasin Manurangsi. A survey on approximation in parameterized complexity: Hardness and algorithms. *Algorithms*, 13(6):146, 2020. doi:10.3390/a13060146.

16 Naveen Garg and Amit Kumar. Minimizing average flow-time: Upper and lower bounds. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 603–613. IEEE Computer Society, 2007. doi:10.1109/FOCS.2007.52.

17 Dorit S. Hochbaum and Wolfgang Maass. Approximation schemes for covering and packing problems in image processing and VLSI. *J. ACM*, 32(1):130–136, 1985. doi:10.1145/2455.214106.

18 Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *Journal of Computer and System Sciences*, 63(4):512–530, 2001. doi:10.1006/jcss.2001.1774.

19 Sándor Kisfaludi-Bak. *ETH-tight algorithms for geometric network problems.* Phd thesis, Technische Universiteit Eindhoven, Department of Mathematics and Computer Science, Eindhoven, The Netherlands, 2019. Available at https://research.tue.nl/en/publications/eth-tight-algorithms-for-geometric-network-problems.

20 Sándor Kisfaludi-Bak, Dániel Marx, and Tom C. van der Zanden. How does object fatness impact the complexity of packing in d dimensions? In Pinyan Lu and Guochuan Zhang, editors, *30th International Symposium on Algorithms and Computation, ISAAC*, LIPIcs, pages 36:1–36:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPIcs.ISAAC.2019.36.

21 Sándor Kisfaludi-Bak, Jesper Nederlof, and Karol Węgrzycki. A gap-ETH-tight approximation scheme for Euclidean TSP. *Journal of the ACM*, 72(6):42:1–42:48, 2025. doi:10.1145/3723180.

22 Ariel Kulik and Hadas Shachnai. There is no EPTAS for two-dimensional knapsack. *Information Processing Letters*, 110(16):707–710, 2010. doi:10.1016/j.ipl.2010.05.031.

23 Pasin Manurangsi and Prasad Raghavendra. A Birthday Repetition Theorem and Complexity of Approximating Dense CSPs. In *44th International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 78:1–78:15, 2017. doi:10.4230/LIPIcs.ICALP.2017.78.

24 Dániel Marx. Parameterized Complexity of Independence and Domination on Geometric Graphs. In Hans L. Bodlaender and Michael A. Langston, editors, *Parameterized and Exact Computation, Second International Workshop, IWPEC, Zürich, Switzerland, September 13-15, 2006, Proceedings*, Lecture Notes in Computer Science, pages 154–165. Springer, 2006. doi:10.1007/11847250_14.

25 Dániel Marx. On the optimality of planar and geometric approximation schemes. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 338–348. IEEE Computer Society, 2007. doi:10.1109/FOCS.2007.26.

26 Dániel Marx and Anastasios Sidiropoulos. The limited blessing of low dimensionality: When $1-1/d$ is the best possible exponent for d-dimensional geometric problems. In *Proceedings of the 30th Annual Symposium on Computational Geometry (SoCG)*, pages 67:1–67:10. Association for Computing Machinery, 2014. doi:10.1145/2582112.2582124.

27 Christos H. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.