Finding Large Independent Sets in Line of Sight Networks

Pavan Sangha and Michele Zito

University of Liverpool, United Kingdom
P.Sangha@liverpool.ac.uk and M.Zito@liverpool.ac.uk

Abstract. Line of Sight (LoS) networks provide a model of wireless communication which incorporates visibility constraints. Vertices of such networks can be embedded in finite d-dimensional grids of size n, and two vertices are adjacent if they share a line of sight and are at distance less than ω. In this paper we study large independent sets in LoS networks. We prove that the computational problem of finding a largest independent set can be solved optimally in polynomial time for one dimensional LoS networks. However, for $d \geq 2$, the (decision version of) the problem becomes NP-hard for any fixed $\omega \geq 3$ and even if ω is chosen to be a function of n that is $O(n^{1-\epsilon})$ for any fixed $\epsilon > 0$. In addition we show that the problem is also NP-hard when $\omega = n$ for $d \geq 3$. This result extends earlier work which showed that the problem is solvable in polynomial time for gridline graphs when $d = 2$. Finally we describe simple algorithms that achieve constant factor approximations and present a polynomial time approximation scheme for the case where ω is constant.

1 Introduction

Geometric graphs have become a popular tool for reasoning about wireless networks. Typically wireless devices positioned in some physical space can be represented by a collection of vertices. A graph can then be constructed by representing communication between pairs of vertices by edges.

The disk intersection model is a commonly used model for representing wireless sensor networks [16]. Sensors are modelled as vertices in some topological setting and their communication ranges are represented by circles having some prescribed radius. Overlapping circles then represent communication between pairs of vertices and makes it possible to construct a graph. Unfortunately in many real world applications of wireless networks, the environments often come with a large number of obstacles which impose line of sight constrictions on vertices. These obstacles are often difficult to incorporate in the geometric models described above.

Frieze et al [7] developed the notion of a (random) line of sight network to provide a model of wireless networks which can incorporate line of sight constraints. For positive integers d and n, let $\mathbb{Z}_n^d = \{(x_1, x_2, ..., x_d) : x_i \in \{0, 1, ..., n-1\}, 1 \leq i \leq d\}$ and $\mathbb{Z}_+^d = \cup_{n=1}^{\infty} \mathbb{Z}_n^d$. In the rest of the paper the distance between points $\boldsymbol{x} = (x_1, x_2, ..., x_d)$ and $\boldsymbol{x}' = (x_1', x_2', ..., x_d')$ in $\mathbb{Z}_n^d$ is the quantity $\sum_{i=1}^{d} |x_i - x_i'|$.

We say that two distinct points $\boldsymbol{x} = (x_1, x_2, \ldots, x_d), \boldsymbol{x}' = (x'_1, x'_2, \ldots, x'_d)$ in $\mathbb{Z}_n^d$ share a line of sight if there exists a $j \in \{1, \ldots, d\}$ such that $x_i = x'_i$ for all $i \in \{1, \ldots, d\} \setminus \{j\}$, moreover in this case we say that $\boldsymbol{x}$ and $\boldsymbol{x}'$ share a j-line.

Definition 1 *A graph $G = (V, E)$ is said to be a* Line of Sight (LoS) network *with parameters d, n and ω, if there exists an embedding $f_G : V \to \mathbb{Z}_n^d$, such that $\{u, v\} \in E$ if and only if $f_G(u)$ and $f_G(v)$ share a line of sight and the distance between them is strictly less than ω.*

We denote the set of embedded vertices of $G = (V, E)$ by $f_G(V) \in \mathbb{Z}_n^d$. In what follows, because all graphs will be embedded we will abuse notation so that u will denote both a vertex in V and the corresponding embedded vertex $f_G(u) \in f_G(V)$. The parameter ω, called the network *range parameter*, can be used to model the usual proximity constraint in a wireless network. The line of sight visibility constraint is the mechanism that allows the modelling of obstacles.

Definition 2 *We say that a LoS network G with parameters d, n and ω is* d-dimensionally spanning *if for each $j \in \{1, 2, \ldots, d\}$ there exists an edge $\{u, v\} \in E$ such that $f_G(u)$ and $f_G(v)$ share a j-line.*

Let $L_{n,\omega}^d$ denote the set of all graphs G which are d-dimensionally spanning LoS networks with parameters n, d and ω. Let $L_\omega^d = \cup_{n \in \mathbb{N}} L_{n,\omega}^d$. Note that ω might be as large as n. To study the properties of LoS networks with large range parameter sometimes we are interested in the properties of $L_{n,g}^d$ (or L_g^d) the set of LoS networks with range parameter $\omega = g(n)$, if $g : \mathbb{N} \to \mathbb{N}$ is a monotone increasing function.

LoS networks generalise other well known geometric graph models. For example, a LoS network with parameter $\omega = 2$ is known as a grid graph [3], where each vertex can only share an edge with the $2d$ other vertices at distance one in $\mathbb{Z}_n^d$. On the other hand the elements of L_n^d are known as gridline graphs [15]. So far LoS networks have been studied with respect to their typical connectivity properties [4, 7].

In this paper we investigate the well known maximum independent set problem (MIS) for both 2-dimensional and higher dimensional LoS networks as the range parameter ω varies. Large independent sets in graphs have been the subject of significant study in various branches of Mathematics as they provide a measure of network dispersion and have a strong connection with other important graph measures such as vertex covers, cliques and colourings [6]. It is well known that finding a largest independent set in a graph is an NP-hard problem [8], and even good approximate solutions are hard to find [9]. We show that the problem can be solved optimally in polynomial time for $d = 1$, for any ω, using a straightforward greedy strategy. For $\omega = 2$ and any d the problem can be solved in polynomial time because the LoS network is a bipartite graph. For $d = 2$ the problem can be solved in polynomial time for the case $\omega = n$. In higher dimensions ($d \geq 3$) the problem becomes more difficult and the overall picture is less clear cut. We prove the following (here IS is the decision version of MIS)

Theorem 1. IS$(L^d_{n,\omega})$ *is* NP*-hard, for each fixed* $d \geq 2$ *and fixed integer* $\omega \geq 3$. *Additionally, for any given* $\epsilon > 0$, IS(L^d_g) *is* NP*-hard for any choice of* g *such that* $g(n) = O(n^{1-\epsilon})$.

The proof of Theorem 1 cannot be extended to cover the case of very large range parameters. However a different reduction allows us to prove the following:

Theorem 2. IS(L^d_n) *is* NP*-hard for each fixed* $d \geq 3$.

Note the statement of Theorem 2 is a generalizion to higher dimensions of a result by Peterson on 2-dimensional gridline graphs [15]. Finally, we complement these negative results by describing two heuristics that achieve constant factor approximations and an efficient polynomial time approximation scheme (EPTAS, a la [2]) for the case when ω is a fixed constant, for any d.

The layout of this paper is as follows. We start our investigation in Section 2 by studying a natural greedy heuristic for the MIS problem in LoS networks. The algorithm is optimal for $d = 1$ and represents a base-line benchmark for approximation strategies in higher dimensions. In Section 3 and 4 we study the problem complexity for $d \geq 2$. The final part of the paper is devoted to the remaining algorithmic results mentioned above.

In what follows if Π is a computational problem and $\mathcal{I}$ is a particular set of instances for it, then $\Pi(\mathcal{I})$ will denote the restriction of Π to instances belonging to $\mathcal{I}$. If Π_1 and Π_2 are decision problems, then $\Pi_1 \leq_p \Pi_2$ will denote the fact that Π_1 is polynomial-time reducible to Π_2. Unless otherwise stated we follow [6] for all our graph-theoretic notations.

2 The Corner Greedy Algorithm

We start off our investigation of the MIS problem for LoS networks by describing and analysing a very simple, natural heuristic for tackling this problem. The strategy is an adaptation of an algorithm described in [12] in the context of unit disk graphs. Given a graph $G = (V, E)$ and $u \in V$ we denote by $N(u)$ the set of neighbours of u in V. For a d-dimensional LoS network $G = (V, E)$ and a vertex $u = (u_1, u_2 \ldots, u_d) \in V$ we denote the set of vertices $v = (v_1, v_2, \ldots, v_d)$, such that u and v share a j-line and $v \in N(u)$ as $N_j(u)$ (note that if $v \in N_j(u)$ then $u \in N_j(v)$) and we say that $u <_j v$ if $u_j < v_j$. Corners in LoS networks are vertices that are extremes w.r.t. the $<_j$ relationship for all j. More precisely we give the following definition:

Definition 3 *Given a* d*-dimensional LoS network* $G = (V, E)$*, we say that* $u \in V$ *is a* corner *if for each* $j \in \{1, 2, ..., d\}$, $u <_j v$ *for all* $v \in N_j(u)$ *or* $v <_j u$ *for all* $v \in N_j(u)$.

If we lexicographically order all vertices in a LoS network w.r.t their d-tuples the first vertex in such an ordering is a corner. The corner greedy Algorithm can be described as follows:

Corner Greedy algorithm: Initially $S = \emptyset$

1. Add a corner u to S and remove the closed neighbourhood of $N(u)$ from G.
2. Repeat 1 until the graph G is empty.
3. Return S.

For $d = 1$ the process is clearly optimal and in general we can prove (here $\alpha(G)$ is the *independence number* of G, the size of the largest independent sets in G):

Theorem 3. *For any d-dimensional LoS network G the size of the set of independent vertices returned by the corner greedy algorithm is at least $\frac{\alpha(G)}{d}$.*

Proof. In each iteration the algorithm picks a corner vertex, adds it to S and then removes the vertex and all its neighbours from the graph. Let S^* be an independent set of maximum cardinality in G. Each vertex $v \in S^*$ belongs to exactly one closed neighbourhood identified in step 2. of the Corner Greedy algorithm. Furthermore since each corner vertex has neighbours in at most d different line of sights, d different elements of S^* may belong to a particular closed neighbourhood. The result follows. □

Thus, in general, the corner greedy algorithm described in this section represents a d-approximation algorithm for the MIS problem in d-dimensional LoS networks. It is then natural to ask whether this can be improved. Perhaps LoS networks are sufficiently close to grid graphs and the MIS problem can indeed be solved in polynomial time. Unfortunately in the sections that follow we will give a negative answer to such conjecture.

3 Hardness for Small Ranges

In this section we prove Theorem 1. For $d = 2$ (resp. $d \geq 3$) we describe explicit embeddings in $\mathbb{Z}_n^d$ of graphs which are subdivisions of planar graphs with maximum degree four (resp. subdivisions of bounded degree graphs). In both instances we start by embedding the given graph $G = (V, E)$ orthogonally (see Section 3.1). We then add further vertices to obtain (the embedding of) a d-dimensionally spanning LoS network. Once this is done to prove NP-hardness we show the existence of a linear relationship between the size of an independent set in G and the size of an independent set in the resulting LoS network, and since the IS problem is NP-hard for both planar graphs of maximum degree four and bounded degree graphs the result follows.

3.1 Embeddings

Graph embedding has been an active research area for quite some time (the interested reader is referred to reviews like [5], or the more recent one [13], for additional bibliographic details). Here in particular we will be interested in so called *orthogonal* embeddings of bounded degree graphs $G = (V, E)$ in $\mathbb{Z}_n^d$. Define a path in $\mathbb{Z}_n^d$ to be any sequence of distinct points in $\mathbb{Z}_n^d$ such that any two consecutive points in the sequence have distance one. An orthogonal embedding of a graph $G = (V, E)$ in $\mathbb{Z}_n^d$ denoted by $\Gamma(G)$ is an embedding where the vertices $v \in V$ are mapped to points in $\mathbb{Z}_n^d$ and the edges $\{u, v\} \in E$ to paths with end points $\Gamma(u), \Gamma(v)$. Paths representing distinct edges can only intersect at their end-points. It is well known that an orthogonal embedding of G in $\mathbb{Z}_n^d$ requires $d \geq \lceil \Delta(G)/2 \rceil$ and, for general graphs, the bound is tight except for $\Delta(G) \leq 4$. In this work we will use the following two results, dealing with the case $d = 2$ and $d \geq 3$ respectively. Moreover the embeddings described in the following two theorems can be constructed in polynomial time.

Theorem 4. [17] *Any planar graph $G = (V, E)$ with $\Delta(G) \leq 4$ and $|E| = m$, admits an orthogonal embedding $\Gamma(G)$ in $\mathbb{Z}_{3m}^2$.*

Theorem 5. [18] *Any simple graph $G = (V, E)$ with $\Delta(G) \geq 5$ admits an orthogonal embedding $\Gamma(G)$ in $\mathbb{Z}_{k|V|}^d$ where $d = \lceil \Delta(G)/2 \rceil$ and k is a positive integer constant.*

3.2 Padding

Given a bounded degree graph $G = (V, E)$ and an arbitrary function r mapping the edges of G to positive integers, let $F(G, r)$ be a new graph obtained by replacing each edge $e \in E$ by a path containing a $2r(e)$ additional vertices between it's end points. The graph $F(G, r)$ is also known as a *subdivision* of G (see [6, Chap. I]). Our reduction then uses the following well-known result.

Lemma 1. *Let $G = (V, E)$ be a graph, and r an arbitrary function mapping the edges of G to positive integers. Then $\alpha(G) \geq k$ if and only if $\alpha(F(G, r)) \geq k + \sum_{e \in E} r(e)$.*

3.3 Reduction

We now sketch the main construction in the proof of Theorem 1. Given a bounded degree graph $G = (V, E)$ we use Theorem 4 (for $d = 2$) or Theorem 5 (for $d > 2$) to define the appropriate orthogonal embedding $\Gamma(G)$ in the appropriate d-dimensional finite grid. We then *stretch* the embedding $\Gamma(G)$ to obtain a new orthogonal embedding $\Gamma'(G)$ by mapping each embedded vertex $\Gamma(u) = (u_1, \ldots, u_d) \in \Gamma(G)$ to the embedded vertex $\Gamma'(u) = (4(\omega - 1)u_1, \ldots, 4(\omega - 1)u_d) \in \Gamma'(G)$. The paths P_{uv} in $\Gamma(G)$ with end points $\Gamma(u), \Gamma(v)$ get mapped to the corresponding paths $P'_{uv} \in \Gamma'(G)$ with end points $\Gamma'(u), \Gamma'(v)$. Finally we pad the orthogonal embedding $\Gamma'(G)$ with additional *sensor* vertices to obtain a LoS network embedding of G which is also a subdivision $F(G, r)$, for a suitable choice of r. It is easy to check from the orthogonal embedding of G that the LoS network G' is d-dimensionally spanning. The additional sensor vertices are placed along the paths in $\Gamma'(G)$ corresponding to the edges of G in a way that satisfies the following constraints:

1. the total number of padding vertices placed on each path P'_{uv} is even;

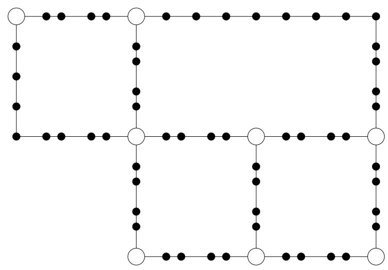

Fig. 1. A 2-dimensionally spanning LoS network $G' = F(G, r)$ with sensor vertices in black, constructed from an orthogonal embedding of the 4-planar graph G with 8 white vertices.

2. a padding vertex must be placed on every corner (bend in the path);
3. for any set of three consecutive sensor vertices $\{v_s, v'_s, v''_s\}$ placed on a path P'_{uv}, v''_s and v_s are at a distance at least ω, but v'_s is at a distance distance at most $\omega - 1$ from v_s and v''_s, maintaining a connected line of sight path structure.

Figure 1 sketches an example of the construction for $d = 2$. The correctness of the reduction follows from Lemma 1, where for each $\{u, v\} \in E$, $4 \times |\Gamma(P_{uv})|$ additional sensor vertices are added to the path P'_{uv}.

The proof of the second part of the statement of Theorem 1 is obtained by noticing that the construction above works if ω is bounded above by a function $g(n) = O(n^{1-\epsilon})$ for any fixed $\epsilon > 0$.

4 Hardness for (Very) Long Ranges

The outcome of Section 3 is a proof that for any $d \geq 2$, IS(L^d_ω) is NP-hard for small values of ω. The analysis of the problem's complexity for $\omega = n$ while different to the sublinear case is quite interesting in its own right. In this section we prove Theorem 2. Let $G = (V, E) \in L^2_n$ then for each vertex $u \in V$ in a LoS embedding of G, $v \in N(u)$ if and only if u and v share a line of sight since ω is as large as possible. A simple reduction shows that MIS(L^2_n) can be solved via a single bipartite matching computation [15].

When $d \geq 3$ things become more complicated. Each element of L^d_n can still be mapped to a d-partite graph. However the independent sets of the LoS network correspond to structures that are computationally less tractable than matchings. In what follows Max 2-Sat(3) is the problem of finding a truth assignment to the variables of a 2-CNF boolean propositional formula that maximizes the number of satisfied clauses, restricted to instances in which each variable occurs in at most three clauses [1]. The proof of Theorem 2 is completed in the case $d \geq 3$ using the following result:

Theorem 6. Max 2-Sat(3) $\leq_p$ IS(L^d_n), *for every* $d \geq 3$.

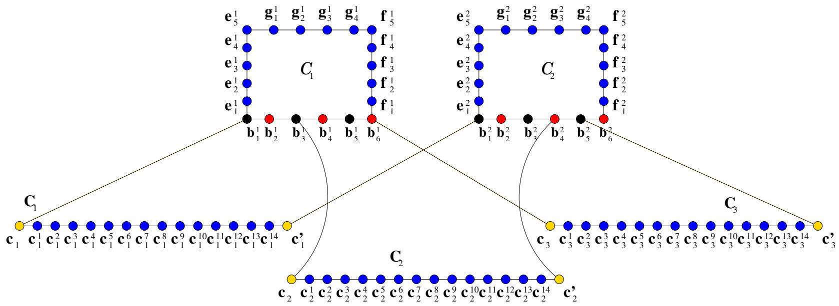

Fig. 2. The construction of the graph $\mathcal{E}^5_{2,3}(F)$ corresponding to the formula $F = (X_1 \vee X_2) \wedge (X_1 \vee \overline{X}_2) \wedge (\overline{X}_1 \vee X_2)$ and the assignment $X_1 = \text{TRUE}$, $X_2 = \text{TRUE}$ satisfying all three clauses.

The proof of this result consists of two steps. We first describe how to translate a given 2-CNF formula into a graph and then we embed such a graph into a d-dimensionally spanning LoS network where ω is maximised.

4.1 From Formulae to Graphs

Consider an instance of Max 2-Sat(3) consisting of a formula $F = C_1 \vee C_2 \vee \ldots \vee C_m$ on n variables and m clauses, each formed by two literals. We construct a graph denoted $\mathcal{E}^d_{n,m}(F)$ as follows. For each variable X_j we construct a variable gadget $\mathcal{X}_j$ which is a cycle of even length $2m + 3d - (d \bmod 2)$ in $\mathcal{E}^d_{n,m}(F)$. The vertices in $\mathcal{C}_j$ are split into four sets: a set of $2m$ *base vertices*, $\{\mathbf{b}^j_1, \mathbf{b}^j_2, \ldots, \mathbf{b}^j_{2m}\}$ and $3d - (d \bmod 2)$ additional vertices which are further split into three sets. There are vertices $\{\mathbf{e}^j_1, \mathbf{e}^j_2, \ldots, \mathbf{e}^j_{d+1-d \bmod 2}\}$, $\{\mathbf{f}^j_1, \mathbf{f}^j_2, \ldots, \mathbf{f}^j_d\}$ and $\{\mathbf{g}^j_1, \mathbf{g}^j_2, \ldots, \mathbf{g}^j_{d-1}\}$. Cycle $\mathcal{X}_j$ is then given by

$$\mathbf{b}^j_1 \ldots \mathbf{b}^j_{2m} \mathbf{f}^j_1 \ldots \mathbf{f}^j_d \mathbf{g}^j_{d-1} \ldots \mathbf{g}^j_1 \mathbf{e}^j_{d+1-d \bmod 2} \ldots \mathbf{e}^j_1 \mathbf{b}^j_1.$$

For each clause C_i in F we construct a clause gadget $\mathbf{C}_i$ containing two *clause vertices* $\mathbf{c}_i$ and $\mathbf{c}'_i$ which are connected by a path of $3d - (d \bmod 2)$ *dummy vertices*. Thus the clause gadget consists of the path $\mathbf{c}_i \mathbf{c}^1_i \mathbf{c}^2_i \ldots \mathbf{c}^{3d-(d \bmod 2)}_i \mathbf{c}'_i$. If C_i in F contains the variable X_j (resp. $\overline{X}_j$) we select odd (resp. even) parity vertex $\mathbf{b}^j_{2i-1}$ (resp. $\mathbf{b}^j_{2i}$) on the cycle $\mathcal{X}_j$ and we connect it to one of the clause vertices in $\mathbf{C}_i$. Note that because of the variable occurrence constraint, in each variable gadget at most three base vertices are selected.

Claim. For any instance of Max 2-Sat(3) with m clauses on n variables, there is an assignment that satisfies r clauses if and only if the graph $\mathcal{E}^d_{n,m}(F)$ has an independent set of size

$$r + \lfloor 3d/2 \rfloor \times m + n \times (m + (3d - (d \bmod 2))/2).$$

Each variable gadget $\mathcal{X}_j$ has an independent set of size $m + (3d - (d \bmod 2))/2$ and in fact there's always exactly two sets of this size, one using all odd indexed $\mathbf{b}_h^j$, the other using all the even indexed ones. The former of these corresponds to setting X_j to TRUE, the other one to FALSE. Also, each clause gadget $\mathbf{C}_i$ is a path of even length, the largest independent sets of these paths are of size $\lfloor 3d/2 \rfloor + 1$ and include one of $\mathbf{c}_i$ or $\mathbf{c}'_i$. If clause C_i is satisfied at least one of its literals is set to TRUE and that implies that an independent set can be picked in the corresponding variable gadget that leaves at least one of $\mathbf{c}_i$ or $\mathbf{c}'_i$ free to be added to an independent set. Conversely if C_i is not satisfied, then neither $\mathbf{c}_i$ nor $\mathbf{c}'_i$ are free and the corresponding clause gadgets will only contribute $\lfloor 3d/2 \rfloor$ vertices to the independent set. The first part of the claim follows.

For the opposite direction let $\mathcal{I}$ be an independent set in $\mathcal{E}^d_{n,m}(F)$. Two cases arise. If all the variable gadgets contain half of their vertices in $\mathcal{I}$ then the 2^n possible settings correspond to the possible ways to assign a truth value to the variables of F. In each of these cases a clause gadget will contain $\lfloor 3d/2 \rfloor + 1$ elements of $\mathcal{I}$ if and only if at least one of the clause vertices is one of them. The number r of clause gadgets containing $\lfloor 3d/2 \rfloor + 1$ elements of $\mathcal{I}$ satisfies

$$r = |\mathcal{I}| - \lfloor 3d/2 \rfloor \times m - n \times (m + (3d - (d \bmod 2))/2).$$

We call an independent set of this type *full*. If $\mathcal{I}$ is not full then wlog let $\mathcal{X}_k$ be a variable gadget such that less than half the vertices of $\mathcal{X}_k$ belong to $\mathcal{X}_k \cap \mathcal{I}$. Call $\mathbf{b}^k(x)$, $\mathbf{b}^k(y)$, $\mathbf{b}^k(z)$ the three base variables from $\mathcal{X}_k$ that are adjacent to clause gadgets $\mathbf{C}_x$, $\mathbf{C}_y$ and $\mathbf{C}_z$. We replace the set $\mathcal{X}_k \cap \mathcal{I}$ with a maximum independent set of $\mathcal{X}_k$ containing half it's vertices according to the parity of $\mathbf{b}^k(x)$, $\mathbf{b}^k(y)$, and $\mathbf{b}^k(z)$: if $\mathbf{b}^k(x)$, $\mathbf{b}^k(y)$, and $\mathbf{b}^k(z)$ all have the same, say, odd parity, we replace $\mathcal{X}_k \cap \mathcal{I}$ with the set of all even vertices in $\mathcal{X}_k$. If two of them are of the same parity we pick the opposite to the majority. During this process we may remove at most one clause vertex from $\mathcal{I}$. For instance in the case where a base vertex not in the majority and not in the independent set $\mathcal{X}_k \cap \mathcal{I}$ is added to the independent set during the process. In this case it's adjacent clause vertex must be removed once it is added. However in replacing $\mathcal{X}_k \cap \mathcal{I}$ with an independent set in $\mathcal{X}_k$ consisting of half of the vertices we add at least one extra vertex from $\mathcal{X}_k$ to our independent set, thus negating the potential loss of a clause vertex. The process terminates in at most n steps with an independent set $|\mathcal{I}'| \geq |\mathcal{I}|$.

4.2 Embedding

To complete the reduction we need to show that there exists an integer N, polynomial in n and m for fixed d, such that $\mathcal{E}^d_{n,m}(F)$ has a d-dimensionally spanning LoS network embedding in $\mathbb{Z}^d_N$ satisfying $\omega = N$, thus showing $\mathcal{E}^d_{n,m}(F) \in L^d_{N,N}$. In what follows we use $N = 7(m + d)n$.

In order to explain the embedding of $\mathcal{E}^d_{n,m}(F)$, we first discuss how to embed paths and cycles in ${\mathbb{Z}^+}^d$ satisfying LoS network constraints with the range parameter maximized. We use methods called *path rotation* and *path connection* to embed, respectively, long and short paths. Given a path $P = v_1, \ldots, v_k$

the process starts by assigning (arbitrarily or according to some rule) a d-tuple $(x_1, x_2, \dots, x_d)$ to v_1. After that, the path rotation method assigns d-tuples iteratively so that the d-tuple assigned to v_i differs from the d-tuple assigned to v_{i-1} in co-ordinate position $(i-1) \bmod d$. Furthermore the integer value assigned to the d-tuple v_i in the co-ordinate position $(i-1) \bmod d$ is one more than the integer value assigned to the d-tuple v_{i-1} in co-ordinate position $(i-1) \bmod d$. Note for two adjacent vertices v_{i-1}, v_i on the path, there remains an edge between their assigned d-tuples as these differ in exactly one co-ordinate position (i.e. share a line of sight). The path connection method is used to specifically embed a path $P' = x, c_1, c_2, \dots c_{d-1}, y$ in ${\mathbb{Z}^+}^d$ of length d where the d-tuples $x = (x_1, x_2, \dots, x_d)$ and $y = (y_1, y_2, \dots, y_d)$ are pre-assigned and distinct ($x_i \neq y_i$ for $i \in \{1, 2, \dots, d\}$). We can denote $x = c_0$ and $y = c_d$. We then embed c_i for $i \in \{1, \dots, d-1\}$ as follows. Let S_d denote the set of all permutations of the set $\{1, \dots, d\}$ and pick $\sigma \in S_d$. Then the pair (c_i, c_{i-1}) differ in co-ordinate position $\sigma(i)$ and the value assigned to c_i in this position is $y_{\sigma(i)}$.

We can use a combination of the path rotation and path connection methods to embed a cycle $v_1, \dots, v_k, v_1$ where $k \geq 3d$ in ${\mathbb{Z}^+}^d$. We embed the path $v_1, v_2, \dots, v_{k-(d-1)}$ using the path rotation method, we then embed $v_{k-(d-1)+1}$, $\dots, v_k$ using the path connection method. Note that the resulting embedding is d-dimensionally spanning. We embed each variable gadget $\mathcal{C}_i$ in $\mathcal{E}^d_{n,m}(F)$ according to the method described above, embedding the vertices $\mathbf{e}^j_{d+1-d \bmod 2}, \dots, \mathbf{e}^j_1$, $\mathbf{b}^j_1, \dots, \mathbf{b}^j_{2m}, \mathbf{f}^j_1, \dots, \mathbf{f}^j_d$ using the path rotation method initialising $\mathbf{e}^j_{d+1-d \bmod 2} = ((j-1)(B_{m,d}+1), (j-1)(B_{m,d}+1), \dots, (j-1)(B_{m,d}+1))$ where $B_{m,d} = \lceil \frac{2m+2d-1}{d} \rceil$. The vertices $\mathbf{g}^j_1, \dots, \mathbf{g}^j_{d-1}$ are then embedded using the path connection method. We next embed the clause gadgets $\mathbf{C}_j$ for $j = 1, \dots, m$. For each j we first embed the clause vertices $\mathbf{c}_j$ and $\mathbf{c}'_j$. For each clause vertex we choose the co-ordinate position that it will differ in from it's adjacent base vertex carefully. Without loss of generality assume that clause vertex $\mathbf{c}_j$ is adjacent to some base vertex denoted $\mathbf{b}^i$ (we do not distinguish between $\mathbf{b}^i = \mathbf{b}^i_{2j-1}$ and $\mathbf{b}^i = \mathbf{b}^i_{2j}$). Then we ensure that for each base vertex $\mathbf{b}^i_* \in N(\mathbf{b})$, the co-ordinate positions that the pairs of d-tuples $(\mathbf{b}^i, \mathbf{c}_j)$ and $(\mathbf{b}^i, \mathbf{b}^i_*)$ differ in are not the same. The value assigned in the co-ordinate position in which $(\mathbf{b}^i, \mathbf{c}_j)$ differ is then unique for each $\mathbf{c}_j$ ensuring we do not add unwanted edges. Once $\mathbf{c}_j$ and $\mathbf{c}'_j$ have been assigned their d-tuples the remaining dummy vertices $\mathbf{c}^1_j, \mathbf{c}^2_j, \dots, \mathbf{c}^{3d-d \bmod 2}_j$ are embedded using a combination of path rotation and path connection methods fully embedding $\mathbf{C}_j$.

5 Approximation Algorithms

In this section we further extend our understanding of the computational properties of the MIS problem for d-dimensional LoS networks with constant range parameter $\omega \in \mathbb{N}$. The hardness proofs of Section 3 and 4 will be complemented by two additional algorithmic results. First, we define a polynomial approximation scheme that works for fixed ω. Then we describe a local improvement strategy that beats the approximation guarantee of the corner greedy heuristic described in Section 2 for the extreme case $\omega = n$.

5.1 An Efficient Polynomial Time Approximation Scheme

In this section we describe an algorithm that accepts as input any d-dimensional LoS network $G = (V, E)$ with constant range parameter $\omega \in \mathbb{N}$ and returns a $(1 + \epsilon)$-approximate independent set in the given graph. The process mimics an approximation scheme proposed by Nieberg *et al.* [14] for the MIS problem in *unit disk graphs*. The algorithm in this section however has a running time that, for fixed ω, is linear in the number of vertices of the input graph (although the constant hidden in the big-Oh notation depends exponentially on ϵ^{-1}). Algorithms of this type have been referred to as *Efficient* polynomial time approximation schemes [2].

Let $\epsilon > 0$ and let $\rho = 1+\epsilon$ denote the desired approximation guarantee. Given a LoS network $G = (V, E)$ we seek to construct an independent set of size at least $\alpha(G)/\rho$. Let r be a non-negative integer and for $u \in V$ define the *bounding box* $B^r(u)$ *centered* at $u = (u_1, u_2, \ldots, u_d)$ as the region in $\mathbb{Z}_n^d$ containing the set of points

$$\{y = (y_1, y_2, \ldots, y_d) \in \mathbb{Z}_n^d \mid \max_{i=1,\ldots,d} |y_i - u_i| < r \cdot \omega\}.$$

Let $G[B^r(u)]$ denote the induced subgraph of G in the region $B^r(u)$ in the embedding of G. The proposed algorithm is an iterative process that removes vertices from G. At each iteration it picks a corner vertex $u \in G$ among the surviving vertices, builds $G[B^0(u)], G[B^1(u)], \ldots, G[B^{\hat{r}}(u)]$, computing a maximum independent set I_r of $G[B^r(u)]$ and removes $G[B^{\hat{r}+1}(u)]$ from G (note $G[B^0(u)]$ consists of just the single vertex u) . The value $\hat{r}$ is the smallest positive r for which

$$|I_{r+1}| < \rho \cdot |I_r|. \tag{1}$$

The independent set returned by the algorithm is the union of the sets $I_{\hat{r}}$ produced by each iteration of the algorithm. Note that a maximum cardinality independent set can be found in $B^r(u)$ in time $O((r \cdot \omega)^{d\ r^d \omega^{d-1}})$. It follows from (1) and by the definition of $\hat{r}$ in addition to the fact that $|I_0| = 1$ that for each $r \leq \hat{r}$,

$$|I_r| \geq (1 + \epsilon)^r. \tag{2}$$

In addition, using the pigeon hole principle it can be shown that

$$|I_r| \leq r \cdot (r \cdot \omega)^{d-1}. \tag{3}$$

Hence, for each $r \leq \hat{r}$,

$$(1 + \epsilon)^r \leq |I_r| \leq r^d \cdot \omega^{d-1}.$$

The value $\hat{r}$ is therefore upper bound by the smallest positive integer r for which $r^d \omega^{d-1} < (1 + \epsilon)^r$ and the process is indeed an EPTAS for the MIS problem for LoS networks in the case where $\omega \in \mathbb{N}$. The correctness of the algorithm is entailed by the following result, which mirrors a similar statement in [14]:

Theorem 7. *Suppose inductively that we can compute a ρ-approximate independent set $I' \subset V \setminus N^{\hat{r}+1}(v)$ for G'. Then $I \equiv I_{\hat{r}} \cup I'$ is a ρ-approximate independent set for G.*

5.2 An Improved Approximation Algorithm

In this section we show that for the case $\omega \geq n$ it is possible to improve the guarantee of the greedy heuristic provided in section 2. We provide a local improvement strategy that approximately doubles the guarantee from $\frac{1}{d}$ to $\frac{2}{d} - \epsilon$ where $\epsilon > 0$ is a fixed constant. The algorithm uses the following well known result of Hurkens and Schrijver on set systems [10].

Theorem 8. [10] *Let $E_1, \dots, E_m$ be subsets of a set T of n elements. Suppose that:*

1. *Each element of T is contained in at most $k \geq 3$ of the $E_1, \dots, E_m$ sets;*
2. *For any $p \leq t$, any p sets among $E_1, \dots, E_m$ cover at least p elements of T.*

Then:

$$\frac{m}{n} \leq \frac{k(k-1)^r - k}{2(k-1)^r - k} \quad \text{if } t = 2r-1, \qquad \frac{m}{n} \leq \frac{k(k-1)^r - 2}{2(k-1)^r - 2} \quad \text{if } t = 2r.$$

Consider a d-dimensional LoS network $G = (V, E)$ with $\omega \geq n$, the algorithm works as follows. Start with any collection of independent vertices $S = \{s_1, s_2, \dots, s_r\}$ and fix a $t \geq 1$ in G. We now perform the following iterative local search algorithm H_t which takes any set of $p \leq t$ vertices from S and replaces them with a collection of $p+1$ vertices from $V \setminus S$ so that the new collection remains pairwise independent. By repeating this algorithm it will terminate with a set of disjoint embedded vertices $S' = \{s'_1, s'_2, \dots, s'_n\}$ such that for each set of $p+1 \leq t+1$ pairwise independent vertices amoung $V \setminus S'$ they intersect at least $p+1$ vertices amoung $s'_1 \dots, s'_n$, otherwise we could run the algorithm for another step. Let $U = \{u_1, ..., u_{\alpha(G)}\}$ be an independent set in G of maximum size. Then we claim that $\frac{\alpha(G)}{|S'|}$ satisfies Theorem 9 conditions.

Theorem 9. *Let t be a positive integer, $G = (V, E)$ a LoS network with $\omega \geq n$ and S' the independent set returned by the algorithm H_t. Then*

$$\frac{\alpha(G)}{|S'|} \leq \frac{d - c/(d-1)^{\lfloor t/2 \rfloor}}{2 - c/(d-1)^{\lfloor t/2 \rfloor}}$$

where $c = d - (d-2)(1 - (t \bmod 2))$.

In our application $T = S'$. Also, if $U = \{u_1, \dots, u_{\alpha(G)}\}$ is a maximum independent set in G, we may define $E_i = \{v \in S' : \{v, u_i\} \in E(G)\}$ for each $i \in \{1, \dots, \alpha(G)\}$. Finally since U is an independent set it follows that any vertex $s'_j \in S'$ can only belong to at most d sets E_i because s'_j has exactly d different line of sights in an embedding of G. Thus $k = d$ in our application, note that condition 2. in Theorem 8 is satisfied by construction when the algorithm terminates.

References

1. Ausiello, G., Crescenzi, P., Gambosi, G., Kann, V., Marchetti-Spaccamela, A., Protasi, M. Complexity and Approximation. Combinatorial Optimization Problems and their Approximability Properties Springer-Verlag, Berlin. (1999)
2. Cesati, M., Trevisan, L. On the efficiency of polynomial time approximation schemes. Information Processing Letters 64, 165–171 (1997)
3. Clark, B. N., Colbourn, C. J., Johnson, D. S. Unit Disk Graphs, Discrete Mathematics 86, 165–177 (1990)
4. Devroye, L., Farczadi, L. Connectivity for line-of-sight-networks in higher dimensions. Discrete Mathematics and Theoretical Computer Science 15, 71–86 (2013)
5. Di Battista, G., Eades, P., Tamassia, R., Tollis, I. Algorithms for drawing graphs. An annotated bibliography. Computational Geometry: Theory and Applications, 4, 235–282 (1994)
6. Diestel, R. Graph Theory, Springer-Verlag (1999)
7. Frieze, A., Kleinberg, J., Ravi, R., Debani, W. Line of Sight Networks. Combinatorics, Probability and Computing, 18, 142–163 (2009)
8. Garey, M. R., Johnson, D. S. Computer and Intractability, a Guide to the Theory of NP-Completeness. Freeman and Company (1979)
9. Håstad, J. Clique is hard to approximate within $n^{1-\varepsilon}$. Acta Mathematica, 182, 105–142 (1999)
10. Hurkens, C. A., Schrijver, A. On the size of sets every t of which have an SDR, with an application to the worst case ratio of heuristics for packing problems. SIAM Journal of Discrete Mathematics, 2, 68–72 (1989)
11. Lovasz, L. Matching Theory. North Holland (1986)
12. Marathe, M. V., Breu, H., Hunt, H. B., Ravi, S. S., Rosenkrantz, D. J. Simple Heuristics for Unit Disk Graphs Networks 25, 59–68 (1995)
13. Mohar, B., Thomassen, C. Graphs on Surfaces, John Hopkins University Press (2001)
14. Nieberg, T., Hurink, J., Kern, W. A robust PTAS for Maximum Weight Independent Sets in Unit Disk Graphs. In Hromkovič, J., Nagl, M., Westfechtel, B. (Eds.): Graph-Theoretic Concepts in Computer Science; 30th International Workshop, WG 2004, LNCS, vol. 3353, pp. 214–221, Springer-Verlag (2004)
15. Peterson, D. Gridline graphs: a review in two dimensions and an extension to higher dimensions. Discrete Applied Mathematics, 126, 223-239 (2003)
16. Stoyan, D., Kendall, W. S., Mecke, J., Ruschendorf, L. *Stochastic geometry and its applications*, Vol. II, Wiley, Chichester (1995)
17. Valiant, L. Universality Considerations in VLSI Circuits. IEEE Transations on Computers, C-30, 135–140 (1981)
18. Wood, D. R. On higher-dimensional orthogonal graph drawing, In Australian Computer Science Communications - CATS'97 Proceedings of the Computing: the Australasian Theory Symposium. Australian Computer Science Association, Melbourne Vic Australia, 3–8 (1997)