

A Suboptimality Bound for 2^k Grid Path Planning

Benjamín Kramm,¹ Nicolás Rivera,² Carlos Hernández,³ Jorge A. Baier,^{1,4}

¹Departamento de Ciencia de la Computación, Pontificia Universidad Católica de Chile

²Department of Computer Science and Technology, University of Cambridge, UK

³Departamento de Ciencias de la Ingeniería, Universidad Andrés Bello, Chile

⁴IMFD Chile

Abstract

The 2^k neighborhood has been recently proposed as an alternative to optimal any-angle path planning over grids. Even though it has been observed empirically that the quality of solutions approaches the cost of an optimal any-angle path as k is increased, no theoretical bounds were known. In this paper we study the ratio between the solutions obtained by an any-angle path and the optimal path in the 2^k neighborhood. We derive a suboptimality bound, as a function of k , that generalizes previously known bounds for the 4- and 8-connected grids. We analyze two cases: when vertices of the search graph are placed (1) at the corners of grid cells, and (2) when they are located at their centers. For case (1) we obtain a suboptimality bound of $1 + \frac{1}{8k^2} + O\left(\frac{1}{k^3}\right)$, which is tight; for (2), however, worst-case suboptimality is a fixed value, for every $k \geq 3$. Our results strongly suggests that vertices need to be placed in corners in order to obtain near-optimal solutions. In an empirical analysis, we compare theoretical and experimental suboptimality.

Introduction

Grid path planning is an old AI problem whose applications range from robotics to videogames. It is an alternative to any-angle path planning, where the angles of moves are not restricted and thus the agent does not move discretely between positions in the grid.

Grid path planning is, nevertheless, a very well understood technique, being easier to implement than any-angle search. Motivated by this, Rivera, Hernández, and Baier (2017) recently introduced the 2^k -neighborhoods which generalize the traditionally used 4- and 8-connected grids, by extending them with more moves—indeed, 2^k moves, where $k \geq 2$ is a parameter. As more moves are available, optimal paths under these neighborhoods look similar to any-angle paths.

2^k path planning is practical. Indeed, many grid path planning techniques seem to be adaptable to 2^k neighborhoods. For example, Rivera, Hernández, and Baier (2017) showed that canonical orderings (Harabor and Grastien 2011; Sturtevant and Rabin 2016) can be easily adapted to 2^k -neighborhoods, allowing an implementation of an A*

planner that scales well with k and whose time performance is competitive with any-angle planners like Theta* (Daniel et al. 2010) and Anya (Harabor et al. 2016). Furthermore, Hormazábal et al. (2017) showed that the subgoal graph technique of Uras, Koenig, and Hernández (2013) can be adapted to 2^k -neighborhoods, resulting in a very fast nearly any-angle path planner.

Paths obtained with discrete neighborhoods are often sub-optimal with respect to any-angle optimal paths. The suboptimality of a neighborhood is the worst-case ratio between the cost of an optimal path found under such a neighborhood and the cost of an optimal any-angle path. It is known that suboptimality is bounded on 4- and 8-neighbor connectivity (Nash 2012; Bailey et al. 2011). Indeed, for 4-connected grids paths can be a factor of $6/(2 + \sqrt{2}) \approx 1.7573$ away from optimal, and a factor of $\sqrt{4 - 2\sqrt{2}} \approx 1.0824$ for 8-connected grids paths. Nevertheless, until now, no bounds were known for k greater than 3.

In this paper we derive suboptimality bounds for 2^k -neighborhoods. Following Bailey et al. (2011), we study two grid path planning settings: (1) when the vertices of the search graph are placed at the corners of grid cells, and (2) when the vertices are placed at the centers of the cells. For setting 1, our bound is a function of k that asymptotically decreases quadratically towards 1 as k increases. In addition, when we choose $k = 2$ or $k = 3$ we obtain the same bounds found by Bailey et al. (2011). For setting 2, we show that increasing k does not improve the tight worst-case bound of $3(\sqrt{2} - 1)$ shown by Bailey et al. (2011) to be inherent to 8-connected grids; this suggests that grid path planning over 2^k in this setting may not yield best results in practice. Furthermore, we carry out an empirical analysis in which we show that the empirical maximum and average suboptimality approaches the theoretical bound as k increases.

In the rest of the paper we first introduce some background material and present our main results. Then we prove our main results for grid without obstacles, and in a posterior section we extend those results for a general case. We finish our paper by briefly describing our empirical findings.

Preliminaries

In this section we review the basics of grid path planning and introduce the most commonly used neighborhoods.

Pairs

Throughout the paper we use ordered pairs to denote moves, points, search-graph vertices, and cells of a grid. We interpret a pair (x, y) as a two-dimensional vector and use boldface to denote them. As such, if $\mathbf{v} = (x, y)$ we say that the X-component of \mathbf{v} is x and that the Y-component of \mathbf{v} is y . Given a pair \mathbf{v} , we use the notation \mathbf{v}_x and \mathbf{v}_y , respectively, to denote the X- and Y-component of \mathbf{v} . Also $(x, y) + (x', y') = (x + x', y + y')$ and $c(x, y) = (cx, cy)$. In addition, we define $\|(x, y)\|$ as $\sqrt{x^2 + y^2}$.

Given two pairs \mathbf{u} and \mathbf{v} , we define $\det(\mathbf{u}, \mathbf{v})$ as the determinant of the matrix whose columns are \mathbf{u} and \mathbf{v} , i.e.:

$$\det(\mathbf{u}, \mathbf{v}) = \begin{vmatrix} \mathbf{u}_x & \mathbf{v}_x \\ \mathbf{u}_y & \mathbf{v}_y \end{vmatrix} = \mathbf{u}_x \mathbf{v}_y - \mathbf{u}_y \mathbf{v}_x. \quad (1)$$

It is a well-known fact that $\det(\mathbf{u}, \mathbf{v}) = \|\mathbf{u}\| \|\mathbf{v}\| \sin(\alpha)$, where α is the angle between \mathbf{u} and \mathbf{v} (starting from \mathbf{u} moving anticlockwise to \mathbf{v}). A pair \mathbf{u} is *contained* by moves \mathbf{v} and \mathbf{w} iff there exist non-negative numbers α , and β , such that $\mathbf{u} = \alpha \mathbf{v} + \beta \mathbf{w}$.

We use standard notation for closed and open real intervals; specifically, $[a, b] = \{x \in \mathbb{R} \mid a \leq x \leq b\}$ and $]a, b[= \{x \in \mathbb{R} \mid a < x < b\}$.

Grid Path Planning

An $N \times M$ grid is a tuple (C, O) , where $C = \{(i, j) \in \mathbb{N} \times \mathbb{N} \mid 0 \leq i < N, 0 \leq j < M\}$ is a set of *cells* and $O \subseteq C$ is a set of *obstacles*. A grid (C, O) is *obstacle-free* if $O = \emptyset$. Two cells are *adjacent* iff one can be obtained from the other by adding $(1, 0)$ or $(0, 1)$.

A search graph can be defined for every grid (C, O) . To define its vertices, we choose from among two options:

1. *Vertices are associated with corners of grid cells.* Then, the set of vertices of the search graph for grid (C, O) is $\{\mathbf{c} + (\ell, \ell') \mid \mathbf{c} \in C, \ell, \ell' \in \{0, 1\}\}$.
2. *Vertices are associated with the centers of grid cells.* In this case, the set of vertices for grid (C, O) is simply C .

Given a cell \mathbf{c} we define its *center*, denoted as $\text{center}(\mathbf{c})$ as $\mathbf{c} + (\frac{1}{2}, \frac{1}{2})$ if vertices are associated with corners. We state the following definitions independent of which association we use between vertices and corners/centers.

1. The *corners* of a cell \mathbf{c} is the set of points with the form $\text{center}(\mathbf{c}) + \mu(\frac{1}{2}, 0) + \nu(0, \frac{1}{2})$, where $\mu, \nu \in \{-1, 1\}$.
2. The *border* of a cell \mathbf{c} is the set of points with the form $\text{center}(\mathbf{c}) + \mu(\frac{1}{2}, 0) + \nu(0, \frac{1}{2})$, where $\mu, \nu \in [-1, 1]$, and at least one among μ, ν is equal to 1 or -1 .
3. The *interior* of a cell \mathbf{c} is the set of points with the form $\text{center}(\mathbf{c}) + \mu(\frac{1}{2}, 0) + \nu(0, \frac{1}{2})$, where $\mu, \nu \in]-1, 1[$.

A *move* \mathbf{m} is a pair. The set of points *visited* when applying a move \mathbf{m} in point \mathbf{p} are those with the form $\mathbf{p} + \lambda \mathbf{m}$, with $\lambda \in [0, 1]$. A move \mathbf{m} is *applicable* in vertex \mathbf{v} iff:

(APP1) $\mathbf{v} + \mathbf{m}$ is a vertex of the search graph, and

(APP2) the points visited when applying \mathbf{m} in \mathbf{p} do not contain:

- (a) any point in the interior of an obstacle cell.

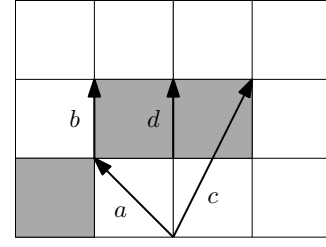


Figure 1: a and b are legal moves. c is not applicable because it violates (APP2)(a). d is not applicable because it violates (APP2)(b).

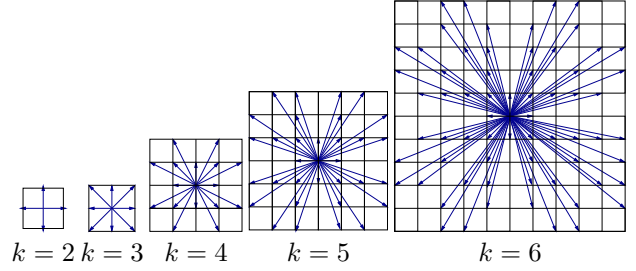


Figure 2: The 4-, 8-, 16-, 32-, and 64- neighborhoods.

- (b) any point in the intersection of the borders of two adjacent obstacle cells that is not a corner of any of them.

See Figure 1 for a graphical example of applicable and non-applicable moves.

In grid path planning moves are not arbitrary pairs: they are taken from a *neighborhood*. The 4- and 8-connected neighborhoods are:

$$\begin{aligned} \mathcal{N}_4 &= \{(i, j) \mid i, j \in \{-1, 0, 1\}, |i| + |j| = 1\}, \\ \mathcal{N}_8 &= \{(i, j) \mid i, j \in \{-1, 0, 1\}, |i| + |j| > 0\}. \end{aligned}$$

The set of successors of a vertex \mathbf{u} is defined as $\text{Succ}_{\mathcal{N}}(\mathbf{u}) = \{\mathbf{m} + \mathbf{u} \mid \mathbf{m} \in \mathcal{N} \text{ and } \mathbf{m} \text{ is applicable in } \mathbf{u}\}$. A *path over* \mathcal{N} from \mathbf{u} to \mathbf{v} is a sequence $\mathbf{v}_1 \mathbf{v}_2 \cdots \mathbf{v}_n$ such that $\mathbf{v}_1 = \mathbf{u}$, $\mathbf{v}_n = \mathbf{v}$, and for every $i \in \{1, \dots, n-1\}$ it holds that $\mathbf{v}_{i+1} \in \text{Succ}_{\mathcal{N}}(\mathbf{v}_i)$. A path $\mathbf{v}_1 \mathbf{v}_2 \cdots \mathbf{v}_n$ is *generated by applying* the sequence of moves $\mathbf{m}_1 \mathbf{m}_2 \cdots \mathbf{m}_{n-1}$ in \mathbf{v} iff $\mathbf{v} = \mathbf{v}_1$ and $\mathbf{m}_i = \mathbf{v}_{i+1} - \mathbf{v}_i$, for every $i \in \{1, \dots, n-1\}$. A path $\mathbf{v}_1 \cdots \mathbf{v}_n$ generated by $\mathbf{m}_1 \cdots \mathbf{m}_{n-1}$ penetrates a cell \mathbf{c} , iff a point in the interior of \mathbf{c} is visited when \mathbf{m}_i is applied in \mathbf{v}_i , for some $i \in \{0, \dots, n-1\}$.

The cost of a path $\sigma = \mathbf{v}_1 \cdots \mathbf{v}_n$ is $c(\sigma) = \sum_{i=1}^{n-1} \|\mathbf{v}_{i+1} - \mathbf{v}_i\|$. A path σ over \mathcal{N} from \mathbf{u} to \mathbf{v} is *optimal* if for every path σ' over \mathcal{N} from \mathbf{u} to \mathbf{v} it holds that $c(\sigma) \leq c(\sigma')$.

A grid path planning problem is a tuple $P = (G, \mathbf{u}_{start}, \mathbf{u}_{goal})$, where G is a grid, \mathbf{u}_{start} is the initial vertex, and \mathbf{u}_{goal} is the goal vertex. A *solution* (resp. *optimal solution*) for P over \mathcal{N} is a path (resp. optimal path) over \mathcal{N} from \mathbf{u}_{start} to \mathbf{u}_{goal} .

The 2^k Neighborhood

The 2^k neighborhood, denoted as \mathcal{N}_{2^k} , generalizes the 4- and 8-connected neighborhoods. \mathcal{N}_{2^k} is defined in terms of its first-quadrant moves. The first quadrant moves of \mathcal{N}_{2^k} , for $k \geq 2$, are denoted by \mathcal{Q}_{k-2} . Finally, Q_i , is defined inductively. We start off by defining the quadrant for \mathcal{N}_4 as $Q_0 = \langle (0, 1), (1, 0) \rangle$. Then, intuitively, we generate Q_{i+1} from Q_i by inserting a new move between each pair of adjacent moves of Q_i . Formally, if $Q_i = \langle \mathbf{a}_0, \dots, \mathbf{a}_n \rangle$, then $Q_{i+1} = \langle \mathbf{a}_0, \mathbf{b}_0, \mathbf{a}_1, \mathbf{b}_1, \dots, \mathbf{a}_{n-1}, \mathbf{b}_{n-1}, \mathbf{a}_n \rangle$, where $\mathbf{b}_j = \mathbf{a}_j + \mathbf{a}_{j+1}$. Thus $Q_1 = \langle (0, 1), (1, 1), (1, 0) \rangle$, $Q_2 = \langle (0, 1), (1, 2), (1, 1), (2, 1), (1, 0) \rangle$, and so on. Finally, we define $\mathcal{N}_{2^k} = \{(\ell x, \ell' y) \mid (x, y) \in \mathcal{Q}_{k-2}, \ell, \ell' \in \{-1, 1\}\}$. Figure 2 shows the first five 2^k -neighborhoods. Two different moves in $\mathbf{u}, \mathbf{w} \in \mathcal{N}$ are *adjacent* iff there is no move in \mathcal{N} different from \mathbf{u} and \mathbf{v} that is contained by them.

Given an obstacle-free grid, Rivera, Hernández, and Baier (2017) prove the following result.

Theorem 1 (Rivera, Hernández, and Baier 2017)

Let \mathbf{u} be a vertex and let \mathbf{v} and \mathbf{w} be the adjacent moves in \mathcal{N}_{2^k} that contain \mathbf{u} . Then the cost of an optimal path from origin to \mathbf{u} is $\alpha\|\mathbf{v}\| + \beta\|\mathbf{w}\|$, where α and β are such that $\mathbf{u} = \alpha\mathbf{v} + \beta\mathbf{w}$.

Henceforth we denote the cost of an optimal path from origin to \mathbf{u} as $h_{2^k}(\mathbf{u})$. Theorem 1 implies the following algorithm for computing $h_{2^k}(\mathbf{u})$. First, find two consecutive moves $\mathbf{v}, \mathbf{w} \in \mathcal{N}_{2^k}$ that contain \mathbf{u} . Then, find α and β such that $\mathbf{u} = \alpha\mathbf{v} + \beta\mathbf{w}$; this step requires solving a system of two linear equations and both solutions, α and β , are positive integers (Rivera, Hernández, and Baier 2017). Finally, return $\alpha\|\mathbf{v}\| + \beta\|\mathbf{w}\|$.

In the procedure described above, solving for α and β can be done in constant time, but we require an efficient function for finding \mathbf{v} and \mathbf{w} if we aim at computing h_{2^k} quickly. Hormazábal et al. (2017) present an iterative algorithm to compute $h_{2^k}(\mathbf{u})$, that efficiently computes \mathbf{v} and \mathbf{w} . We present this procedure here because it is relevant to one of the proofs below (Lemma 9). The main idea is that h_{2^k} can be computed by carrying out a sequence of factorizations. Thus to compute $h_{2^k}(7, 9)$ we first factorize $(7, 9)$ as $7(1, 0) + 9(0, 1)$; observe that this factorization defines an optimal path from $(0, 0)$ to $(7, 9)$ on the 4-connected neighborhood. Now, to obtain an optimal path in the 8-connected neighborhood, we re-factorize the previous expression as $7(1, 1) + 2(0, 1)$; observe that to get this factorization we take as many moves among $(1, 0)$ and $(0, 1)$, and add them together to get $(1, 1)$. Whenever we do this, one move of the previous factorization disappears from the expression, while another move in the next neighborhood appears. Furthermore, after i factorization steps, the two pairs in the expression are consecutive moves in $\mathcal{N}_{2^{i+1}}$. Therefore, we can repeat this process as many times as we want to get optimal paths over \mathcal{N}_{2^k} for increasing k . For the 16-neighborhood we obtain $5(1, 1) + 2(1, 2)$; for the 32-neighborhood, $3(1, 1) + 2(2, 3)$; for the 64-neighborhood, $(1, 1) + 2(3, 4)$; for the 128-neighborhood, $(4, 5) + (3, 4)$; for the 512-neighborhood, $(7, 9)$. Hormazábal et al. (2017)

proved this procedure is correct. Algorithm 1 shows a pseudocode.

Algorithm 1: A general distance function for \mathcal{N}_{2^k}

```

1 function distance( $x, y, k$ )
2    $\mathbf{l} \leftarrow (1, 0)$ 
3    $\mathbf{r} \leftarrow (0, 1)$ 
4   repeat  $k - 2$  times
5     if  $x > y$  then
6        $\mathbf{r} \leftarrow \mathbf{r} + \mathbf{l}$ 
7        $x \leftarrow x - y$ 
8     else
9        $\mathbf{l} \leftarrow \mathbf{r} + \mathbf{l}$ 
10       $y \leftarrow y - x$ 
11  return  $x\|\mathbf{l}\| + y\|\mathbf{r}\|$ 

```

Any-Angle Grid Path Planning

Over the any-angle neighborhood, denoted below as \mathcal{N}_{any} , any-angle moves are possible, and the agent can move through any legal point within the limits of the grid. If the grid is $N \times M$, we say that (x, y) is within the bounds of the grid if $x \in [0, N]$ and $y \in [0, M]$. We re-define when \mathbf{m} is applicable in \mathbf{v} by replacing (APP1) above by:

(APP1) $\mathbf{v} + \mathbf{m}$ is within the bounds of the grid.

The rest of the definitions remain the same. Whenever we have a grid path planning problem P , we can define an equivalent any-angle path planning problem, by simply setting the initial and final points as in P . When both the initial point and the final point correspond to the corner of a cell, optimal \mathcal{N}_{any} paths have an interesting property: they are generated by sequences of *integer moves*, that is, where each move is such that both the X- and Y-coordinates are integer. This is because these paths are *taut*, that is, they are concatenations of straight lines, where ‘breaks’ occur at the corners of obstacles (e.g., Oh and Leong 2017).

Main Results

We proceed to present the main results of this paper. For that, we must establish what we mean by a suboptimality bound. Assuming that our path planning problem is P , let σ_{any}^* be an optimal solution for P over \mathcal{N}_{any} , and let σ_k^* be an optimal solution for P over \mathcal{N}_{2^k} , our objective is to find an upper bound for $c(\sigma_k^*)/c(\sigma_{\text{any}}^*)$. For setting (1), when the vertices are placed at the corners of cells, the following holds.

Theorem 2 Let P be a path planning problem where vertices are placed at the corners of grid cells. Let σ_k^* and σ_{any}^* be optimal paths for P over, respectively, \mathcal{N}_{2^k} and \mathcal{N}_{any} . Then, $c(\sigma_k^*)/c(\sigma_{\text{any}}^*)$ is less than or equal to:

$$\Gamma_k = \sqrt{2\sqrt{(k-2)^2 + 1}(\sqrt{(k-2)^2 + 1} - (k-2))}$$

We remark two important aspects of the above theorem. First we observe the bound of Theorem 2 equals $1 + \frac{1}{8k^2} + O\left(\frac{1}{k^3}\right)$ as k tends to infinity, thus the ratio approaches to 1 with speed quadratic in k . Second, notice

that the above bound is tight in the sense that we can find problems P_n such that their ratio $c(\sigma_k^*)/c(\sigma_{any}^*)$ is as close to the bound of Theorem 2 as desired. Indeed, consider a grid without obstacles, initial state $(0, 0)$, and goal state $(n, \lfloor n\sqrt{(k-2)^2 + 1} + (k-2) \rfloor)$. Then in the \mathcal{N}_{2^k} neighborhood the optimal path is given by performing n times the move $(1, k-2)$ and $\lfloor n\sqrt{(k-2)^2 + 1} + (k-2) \rfloor - n$ times the move $(0, 1)$. It can be checked that as n grows the ratio $c(\sigma_k^*)/c(\sigma_{any}^*)$ approaches the value given in Theorem 2.

For setting (2), where the vertices are placed at the center of the cells, the following result is established.

Theorem 3 *Let P be a path planning problem where vertices are placed at the centers of grid cells. Let σ_k^* and σ_{any}^* be optimal paths for P over, respectively, \mathcal{N}_{2^k} and \mathcal{N}_{any} . Then, $c(\sigma_k^*)/c(\sigma_{any}^*)$ is less than or equal to:*

$$\Gamma_k^{center} = \begin{cases} 6/(2 + \sqrt{2}) & k = 2 \\ 3\sqrt{2} - 3 & k \geq 3 \end{cases} \quad (2)$$

For this setting, Theorem 3 is also tight. Figure 6 shows two path problems attaining the bounds of Theorem 3.

Useful Properties of \mathcal{N}_{2^k}

In this section we provide some simple yet useful properties of \mathcal{N}_{2^k} neighborhoods that feature in the proofs of Theorems 2 and 3. We begin with the following property of adjacent moves in \mathcal{N}_{2^k} .

Proposition 4 *Let \mathbf{u} and \mathbf{v} be two adjacent moves of \mathcal{Q}_i , for any $i \geq 0$. Then $|\det(\mathbf{u}, \mathbf{v})| = 1$.*

Proof: The proof is by induction on i . For the base case, $i = 2$, we have that $|\det((1, 0), (0, 1))| = 1$. For the inductive step, assume the result is true for $i \geq 2$ and let \mathbf{v}_1 and \mathbf{v}_2 be two adjacent moves in \mathcal{Q}_{i+1} . Since \mathbf{v}_1 and \mathbf{v}_2 are adjacent, at least one of them was in \mathcal{Q}_i . Assume $\mathbf{v}_1 \in \mathcal{Q}_i$ (the other case is analogous). Then we get $\mathbf{v}_2 = \mathbf{v}_1 + \mathbf{v}_3$ where $\mathbf{v}_1, \mathbf{v}_3 \in \mathcal{Q}_i$, and thus:

$$\begin{aligned} |\det(\mathbf{v}_1, \mathbf{v}_2)| &= |\det(\mathbf{v}_1, \mathbf{v}_1) + \det(\mathbf{v}_1, \mathbf{v}_3)| \\ &= |0 + \det(\mathbf{v}_1, \mathbf{v}_3)| = 1. \end{aligned}$$

■

A graphical proof of the previous proposition exists where we interpret $\det(\mathbf{u}, \mathbf{v})$ as the area of the quadrilateral defined by \mathbf{u} and \mathbf{v} and show that move $\mathbf{u} + \mathbf{v}$, along with either \mathbf{u} or \mathbf{v} , define two new quadrilaterals with the same area as the original one.

The following property is straightforward from Proposition 4.

Proposition 5 *Let α be the angle between two adjacent moves \mathbf{u} and \mathbf{v} from \mathcal{N}_{2^k} , then $\sin(\alpha) = \frac{1}{\|\mathbf{u}\|\|\mathbf{v}\|}$.*

The following result is straightforward from the definition of \mathcal{Q}_i .

Proposition 6 *For every i , if $\mathbf{u} \in \mathcal{Q}_i$, then $\|\mathbf{u}\| \geq 1$.*

The following property shows that every move that cannot be expressed as a multiple of another eventually appears in a 2^k neighborhood.

Proposition 7 *Let p and q be relative primes. Then there exists an r such that $(p, q) \in \mathcal{N}_{2^r}$.*

Proof: Let $\alpha_i \mathbf{u}_i + \beta_i \mathbf{v}_i$ be the factorization at iteration i of Hormazábal et al.'s algorithm when computing the heuristic for (p, q) . If $\alpha_i \leq \beta_i$, at the next iteration, $\alpha_{i+1} = \alpha_i$ and $\beta_{i+1} = \beta_i - \alpha_i$. When $\alpha_i > \beta_i$, symmetrical equations can be written. From this, we observe that $\max\{\alpha_i, \beta_i\} > \max\{\alpha_{i+1}, \beta_{i+1}\}$ unless $\min\{\alpha_i, \beta_i\} = 0$, in which case we have $\max\{\alpha_i, \beta_i\} = \max\{\alpha_{i+1}, \beta_{i+1}\}$. We conclude that at some iteration, the factorization stabilizes in the form $\alpha \mathbf{u}$, for some α and some \mathbf{u} . But since p and q are prime relatives, then the factorization stabilizes at (p, q) , which implies (p, q) is a move of \mathcal{N}_{2^r} , for some r . ■

Suboptimality Bound for Obstacle-Free Grids

Before finding suboptimality bounds in full generality, in this section we focus on a simpler problem. Namely, given a vertex, we find an upper bound for the ratio between the cost of an optimal \mathcal{N}_{2^k} path and an optimal \mathcal{N}_{any} path towards such a vertex over an *obstacle-free* grid.

Let $\mathbf{u} = (p, q)$ be a vertex. We want to find an upper bound for $h_{2^k}(\mathbf{u})/\|\mathbf{u}\|$. To find such a bound, we will interpret this problem as an optimization problem. Specifically, we want to *maximize* $h_{2^k}(\mathbf{u})/\|\mathbf{u}\|$. Note that this optimization problem is an integer program. Instead of solving such an integer program, we solve its relaxed version; specifically, one in which p and q are allowed to take real values. By obtaining a solution to such a (relaxed) program, we obtain an upper bound for $h_{2^k}(\mathbf{u})/\|\mathbf{u}\|$.

An important observation is that the definition of $h_{2^k}(\mathbf{u})$ depends on the two adjacent moves in \mathcal{Q}_{k-2} that contain \mathbf{u} . The following result establishes an upper bound that depends on those two moves.

Lemma 8 *Let \mathbf{u} be a vertex of a obstacle-free grid. Let \mathbf{v} and \mathbf{w} be adjacent moves in \mathcal{N}_{2^k} that contain \mathbf{u} and let α be the angle between \mathbf{v} and \mathbf{w} . Then, $h_{2^k}(\mathbf{u})/\|\mathbf{u}\| \leq 2 \sin(\alpha/2)/\sin(\alpha)$.*

Proof: Finding an upper bound is equivalent to finding a maximum for $h_{2^k}(\mathbf{u})/\|\mathbf{u}\|$, where $\mathbf{u} \in \mathbb{N}^2$. Let us consider the relaxation of such an integer program in which $\mathbf{u} \in \mathbb{R}^2$ with $\mathbf{u} = (p, q)$. The problem then is reduced to finding a maximum for *ratio* $_{\mathbf{v}, \mathbf{w}}$, defined as

$$ratio_{\mathbf{v}, \mathbf{w}}(\mathbf{u}) = (\alpha_1 \|\mathbf{v}\| + \alpha_2 \|\mathbf{w}\|)/\|\mathbf{u}\|,$$

where α_1 and α_2 are positive integers such that $\alpha_1 \mathbf{v} + \alpha_2 \mathbf{w} = \mathbf{u}$. We can express the same quantity in terms of $m = p/q$ in the following way:

$$ratio_{\mathbf{v}, \mathbf{w}}(m) = \frac{(\mathbf{w}_y - m \mathbf{w}_x)\|\mathbf{v}\| + (m \mathbf{v}_x - \mathbf{v}_y)\|\mathbf{w}\|}{\sqrt{m^2 + 1}}. \quad (3)$$

Now we find the value of m , m^+ , that maximizes Equation 3 by differentiation, obtaining

$$m^+ = \frac{\mathbf{v}_x \|\mathbf{w}\| - \mathbf{w}_x \|\mathbf{v}\|}{\mathbf{w}_y \|\mathbf{v}\| - \mathbf{v}_y \|\mathbf{w}\|}.$$

Then, replacing m^+ in Equation 3 we obtain its maximum value as

$$ratio_{\mathbf{v},\mathbf{w}}^+ = \sqrt{2} \|\mathbf{v}\| \|\mathbf{w}\| \sqrt{1 - \frac{\mathbf{v} \cdot \mathbf{w}}{\|\mathbf{v}\| \|\mathbf{w}\|}}, \quad (4)$$

where $\mathbf{v} \cdot \mathbf{w} = \mathbf{v}_x \mathbf{w}_x + \mathbf{v}_y \mathbf{w}_y$ represents the dot product between \mathbf{v} and \mathbf{w} . Using that $\mathbf{v} \cdot \mathbf{w} = \|\mathbf{v}\| \|\mathbf{w}\| \cos(\alpha)$, the trigonometric identity $1 - \cos(\alpha) = 2 \sin^2(\alpha/2)$, and Proposition 5, we obtain that

$$ratio_{\mathbf{v},\mathbf{w}}^+ = 2 \sin(\alpha/2) / \sin(\alpha). \quad (5)$$

■

Lemma 8 gives us an upper bound for $h_{2^k}(\mathbf{u})/\|\mathbf{u}\|$ that, unfortunately, depends on the moves of \mathcal{N}_{2^k} that contain \mathbf{u} . Below, in Theorem 10, we obtain a bound that is independent of such moves. Because the bound of Lemma 8, $2 \sin(\alpha/2) / \sin(\alpha)$, is non-decreasing with α on the interval $[0, \pi/2]$, we know that the ratio is maximized when \mathbf{u} is contained by the (adjacent) moves from \mathcal{N}_{2^k} that form the largest angle. A very interesting fact about \mathcal{N}_{2^k} is that, for every k , the first two (or last two) moves of a quadrant are those that form the largest angle (this can be observed in Figure 2). This is what Lemma 9 formally establishes.

Lemma 9 *For any $i \geq 0$, let $\mathbf{v}_0, \mathbf{v}_1, \dots, \mathbf{v}_n$ be moves such that $\mathcal{Q}_i = \langle \mathbf{v}_0, \mathbf{v}_1, \dots, \mathbf{v}_n \rangle$. Let α_j be the angle between \mathbf{v}_j and \mathbf{v}_{j+1} , for every $j \in \{0, \dots, n-1\}$. Then, $\alpha_0 = \max\{\alpha_0, \dots, \alpha_{n-1}\}$.*

Proof: By induction over i , we prove a stronger result. Namely that:

$$(R1) \quad \|\mathbf{v}_1\| \leq \|\mathbf{v}_k\|, \text{ for every odd } k \in \{0, \dots, n\}.$$

$$(R2) \quad \alpha_0 \geq \alpha_k \text{ for every } k \in \{0, \dots, n\}.$$

For the base case ($i = 0$), both (R1) and (R2) hold. Now we assume both (R1) and (R2) hold for \mathcal{Q}_i , and we prove they hold also for \mathcal{Q}_{i+1} . Let

$$\mathcal{Q}_{i+1} = \langle \mathbf{v}_0, \mathbf{w}_0, \mathbf{v}_1, \mathbf{w}_1, \dots, \mathbf{w}_{n-1}, \mathbf{v}_n \rangle.$$

First we prove that (R1) holds for \mathcal{Q}_{i+1} . This is equivalent to proving that $\|\mathbf{w}_0\| \leq \|\mathbf{w}_k\|$, for every $k \in \{0, \dots, n-1\}$. Because $\mathbf{w}_0 = \mathbf{v}_0 + \mathbf{v}_1$ and $\mathbf{w}_k = \mathbf{v}_k + \mathbf{v}_{k+1}$:

$$\|\mathbf{w}_0\|^2 = \|\mathbf{v}_0\|^2 + \|\mathbf{v}_1\|^2 + 2\mathbf{v}_0 \cdot \mathbf{v}_1, \quad (6)$$

$$\|\mathbf{w}_k\|^2 = \|\mathbf{v}_k\|^2 + \|\mathbf{v}_{k+1}\|^2 + 2\mathbf{v}_k \cdot \mathbf{v}_{k+1}. \quad (7)$$

In addition,

$$\mathbf{v}_0 \cdot \mathbf{v}_1 = \|\mathbf{v}_0\| \|\mathbf{v}_1\| \cos \alpha_0, \quad (8)$$

$$\mathbf{v}_k \cdot \mathbf{v}_{k+1} = \|\mathbf{v}_k\| \|\mathbf{v}_{k+1}\| \cos \alpha_k. \quad (9)$$

From the inductive hypothesis, $\alpha_0 \geq \alpha_k$, then:

$$\cos \alpha_0 \leq \cos \alpha_k. \quad (10)$$

One number among k and $k+1$ is odd. Without loss of generality, we assume that $k+1$ is odd. From the inductive hypothesis:

$$\|\mathbf{v}_1\| \leq \|\mathbf{v}_{k+1}\|. \quad (11)$$

Moreover, given that $\|\mathbf{v}_0\| = 1$ and Proposition 6, we have that:

$$\|\mathbf{v}_0\| \leq \|\mathbf{v}_k\|. \quad (12)$$

Using (6)–(12), we obtain $\|\mathbf{w}_0\| \leq \|\mathbf{w}_k\|$, for every $k \in \{0, \dots, n-1\}$, which proves that (R1) holds in \mathcal{Q}_{i+1} .

Let β_{2i} be the angle between \mathbf{v}_i and \mathbf{w}_i , and β_{2i+1} be the angle between \mathbf{w}_i and \mathbf{v}_{i+1} , for every $i \in \{0, \dots, n-1\}$. To prove that (R2) holds in \mathcal{Q}_{i+1} , we need to establish that $\beta_0 \geq \beta_k$, for every $k \in \{0, \dots, 2n-1\}$.

Assume by contradiction that for some $i \in \{0, \dots, 2n-1\}$, $\beta_0 < \beta_i$. We have two cases: i is even, and i is odd. Here we show the proof for the first case only; the proof for the second is analogous. Because the angle between two adjacent moves is on the interval $[0, \pi/2]$ and the function \sin is non-decreasing in that range, $\sin(\beta_0) < \sin(\beta_i)$. From Proposition 5:

$$\frac{1}{\|\mathbf{v}_0\| \|\mathbf{w}_0\|} < \frac{1}{\|\mathbf{v}_i\| \|\mathbf{w}_i\|}.$$

Since $\|\mathbf{v}_0\| = 1$, we obtain that $\|\mathbf{w}_0\| > \|\mathbf{v}_i\| \|\mathbf{w}_i\|$. Because of Proposition 6, $\|\mathbf{v}_i\| \geq 1$, and hence $\|\mathbf{w}_0\| > \|\mathbf{w}_i\|$, which contradicts (R1). Thus by contradiction, we have established that (R2) holds. ■

Theorem 10 *Let P be a path planning problem on an obstacle-free grid. Then the ratio between the costs of the optimal solutions over \mathcal{N}_{2^k} and \mathcal{N}_{any} is upper-bounded by:*

$$\Gamma_k = \sqrt{2\sqrt{(k-2)^2 + 1}(\sqrt{(k-2)^2 + 1} - (k-2))}$$

Proof: The bound of Lemma 8 is non-decreasing with α on the interval $[0, \pi/2]$. By Lemma 9 the maximum angle between any two consecutive moves of \mathcal{N}_{2^k} is the angle between the first and the second moves in \mathcal{Q}_{k-2} . Moreover, it is easy to see that the second move in \mathcal{Q}_{k-2} has the form $(1, k-2)$. By substituting $v = (0, 1)$ and $w = (1, k-2)$ in Equation (4) we obtain the desired result. ■

At the start of this section we focused on the case which vertices were placed at the corners of grid cells to avoid analysis over non-integer vertices. However, the suboptimality bound for the case in which vertices are placed on the centers of grid cells is the same bound that we just obtained. By adding $(1/2, 1/2)$ to every vertex, we are in the case in which the search graph has vertices are placed at the center of grid cells, and every proof still holds.

Suboptimality Bound for Grids with Obstacles

In this section we give a proof of Theorems 2 and 3 in full generality, that is, for grids with obstacles.

Vertices Placed at the Corners of Grid Cells

Let us focus on the case in which vertices are placed on the corners of the grid cells. Our proof is inspired by a technique proposed by Bailey et al. (2011) and its main idea is as follows. We show how to construct a path σ_k^f over \mathcal{N}_{2^k} which “follows” σ_{any}^* in the sense that it only penetrates the cells penetrated by σ_{any}^* (see Figure 3 for some graphical examples). Observe that σ_{any}^* is a concatenation of straight lines

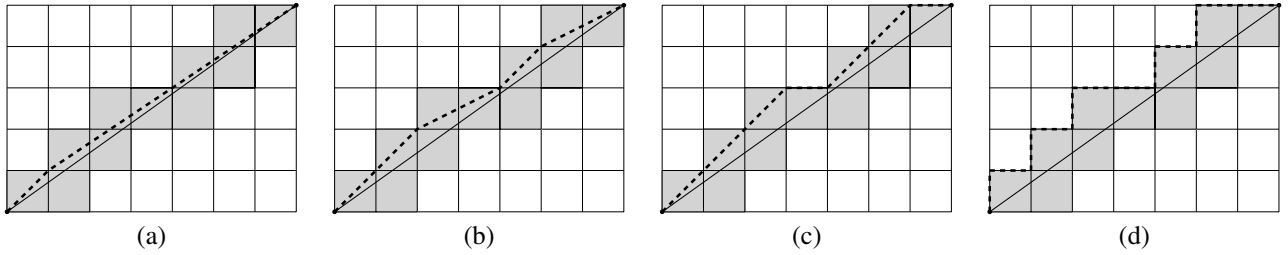


Figure 3: Optimal paths over a 32-, 16-, 8-, and 4-connected neighborhood that follow an any-angle path. Gray grid cells are those visited by the any-angle path.

which in turn are optimal paths $\sigma_{any,0}^* \sigma_{any,1}^* \dots \sigma_{any,n}^*$ over \mathcal{N}_{any} between each vertex where σ_{any}^* ‘breaks’. We will create σ_k^f generating an optimal path $\sigma_{k,i}^f$ for each $\sigma_{any,i}^*$ and then we will use Theorem 10 to get a suboptimality bound for each one of these straight lines, which finally will give us a suboptimality bound between σ_k^f and σ_{any}^* .

Lemma 13, below, is the main technical part of the proof. It shows that if we take an optimal path σ_{any}^* from (x_1, y_1) to (x_2, y_2) over \mathcal{N}_{any} where $(x_2, y_2) - (x_1, y_1) = (p, q)$ are such that (p, q) is a move of some neighborhood, we can create a path σ_k^f over \mathcal{N}_{2^k} which follows σ_{any}^* by reversing the process carried out to compute the heuristic for (p, q) . Recall that to obtain the heuristic for any vertex \mathbf{u} , one carries out a sequence of factorizations that ends with \mathbf{u} or with a multiple of a single move of \mathcal{N}_{2^k} , for some k . See Figure 3 for a graphical illustration. We conclude that by reversing the factorization (a process we decided to name as *decomposition*), we actually obtain a \mathcal{N}_{2^k} path for any value of k we aim at.

Before we formalize our result, we prove an intermediate result that says that if we take a trajectory of two adjacent \mathcal{N}_{2^k} moves \mathbf{v}_1 and \mathbf{v}_2 , we will not visit more cells than those visited by $\mathbf{v}_1 + \mathbf{v}_2$.

Lemma 11 *Let \mathbf{v}_1 and \mathbf{v}_2 be two adjacent moves of \mathcal{N}_{2^k} and \mathbf{u} be any vertex. Then each cell visited by the path generated from \mathbf{u} by applying $\mathbf{v}_1\mathbf{v}_2$ is also visited by the path generated from \mathbf{u} by applying $\mathbf{v}_1 + \mathbf{v}_2$.*

Proof: Our proof is by contradiction. Suppose \mathbf{v}_1 and \mathbf{v}_2 are adjacent moves of some neighborhood such that the property does not hold. We know that $\mathbf{v}_1 + \mathbf{v}_2$ is contained by \mathbf{v}_1 and \mathbf{v}_2 , and therefore a necessary condition for $\mathbf{v}_1 + \mathbf{v}_2$ not to visit a cell visited by \mathbf{v}_1 is that it passes below such a cell, as we can see in Figure 4 (dark gray filled cell). The existence of such a cell implies at least one vertex (p in Figure 4) between \mathbf{v}_1 and \mathbf{v}_2 . This vertex, however, cannot exist in the \mathcal{N}_{2^k} neighborhood; if it did, it would be reached by an integer combination of \mathbf{v}_1 and \mathbf{v}_2 , and thus it would not be in the region shown in the figure. The same contradiction can be obtained for \mathbf{v}_2 . ■

Now we define more precisely what we mean by a decomposition. First we need a few definitions.

We define the *level* of a move \mathbf{v} , $level(\mathbf{v})$, as k if it is in \mathcal{N}_{2^k} but not in $\mathcal{N}_{2^{k-1}}$. Intuitively, a move has level k if it appears first in \mathcal{N}_{2^k} , that is to say, it is the sum of two

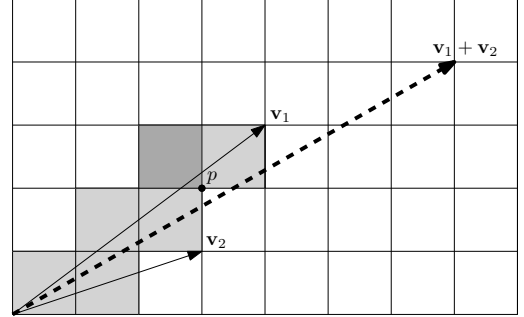


Figure 4: An illustration for the proof of Lemma 11.

adjacent moves in $\mathcal{N}_{2^{k-1}}$. An important fact to note is that given any two adjacent moves of \mathcal{N}_{2^k} , the level of one of them is k , while the level of the other is lower than k .

A decomposition tree \mathcal{T} for a move \mathbf{v} is a binary tree whose nodes are labeled with a move of some 2^k -neighborhood; in particular, the root node is labeled with \mathbf{v} . In addition, if n is a non-leaf node of \mathcal{T} labeled with move \mathbf{w} and $level(\mathbf{w}) = \ell$, then n has two children, labeled with moves \mathbf{w}_1 and \mathbf{w}_2 such that (1) \mathbf{w}_1 and \mathbf{w}_2 are adjacent in $\mathcal{N}_{2^{\ell-1}}$, and (2) $\mathbf{w} = \mathbf{w}_1 + \mathbf{w}_2$. Intuitively the children of a node n are labeled with the moves that ‘create’ the move that labels n . As such, if n is labeled with move \mathbf{w} and the labels of n ’s children are \mathbf{w}_1 and \mathbf{w}_2 , then $level(\mathbf{w}) = 1 + \max\{level(\mathbf{w}_1) + level(\mathbf{w}_2)\}$.

If \mathcal{T} is a decomposition tree then the level of \mathcal{T} , $level(\mathcal{T}) = \max_{t \in leaves(\mathcal{T})} \{level(t)\}$, is the largest level of its leaves. Given a move \mathbf{v} and a natural k , Algorithm 2 computes a decomposition tree of level k for \mathbf{v} .

Lemma 12 *Let \mathcal{T} be a decomposition tree of level k for a move \mathbf{v} . Then, the leaves of \mathcal{T} are labeled with at most two adjacent moves in \mathcal{N}_{2^k} .*

Proof: If $level(\mathbf{v}) = k$, `DecompositionTree` can not continue the decomposition because of its restriction of the maximum level of its leaves, and returns \mathcal{T} , a decomposition tree with just one leaf node, its root, which satisfies the property. Now we will prove a loop invariant: *at the end of the body of each while loop of `DecompositionTree`, \mathcal{T} ’s leaves are labeled with at most two adjacent moves in \mathcal{N}_{2^ℓ} .* Because of this loop invariant, at the end of the body of the last while loop, $\ell = k$ and \mathcal{T} satisfies the property.

Algorithm 2: Decomposition Tree for \mathbf{v} of level k

```
1 function DecompositionTree( $\mathbf{v}$ ,  $k$ )
2    $\ell \leftarrow \text{level}(\mathbf{v})$ ;
3   if  $\ell < k$  or  $k < 2$  then return failure;
4    $\mathcal{T} \leftarrow \text{Tree with a single node labeled with } \mathbf{v} \text{ as}$ 
      $\text{root}$ ;
5   while  $\ell > k$  do
6      $\mathcal{L} \leftarrow \mathcal{T}$  leaves of level  $\ell$ ;
7     foreach  $n \in \mathcal{L}$  do
8        $\mathbf{w} \leftarrow \text{label of } n$ ;
9        $\mathbf{v}_1, \mathbf{v}_2 \leftarrow \text{Adjacent moves in } \mathcal{N}_{2^{\ell-1}}$  such
         that  $\mathbf{w} = \mathbf{v}_1 + \mathbf{v}_2$ ;
10       $n_1, n_2 \leftarrow \text{Nodes labeled with moves } \mathbf{v}_1$ 
         and  $\mathbf{v}_2$ ;
11      Add  $n_1$  and  $n_2$  to  $\mathcal{T}$  as children of  $n$ ;
12     $\ell \leftarrow \ell - 1$ ;
13  return  $\mathcal{T}$ 
```

At the first iteration, $\mathcal{L} = \{n\}$ and $l = \text{level}(\mathbf{v}) > 2$. Then, \mathbf{v} is “decomposable” and there exist adjacent moves \mathbf{v}_1 and \mathbf{v}_2 in $\mathcal{N}_{2^{\ell-1}}$ such that $\mathbf{v} = \mathbf{v}_1 + \mathbf{v}_2$. DecompositionTree will be able to add n_1 and n_2 as children nodes of n , which satisfies the invariant.

Suppose the property is true at iteration i and let \mathbf{v}_1 and \mathbf{v}_2 be the moves that label the leaves of \mathcal{T} . Without loss of generality assume $\mathbf{v}_1, \mathbf{v}_2 \in \mathcal{N}_{2^m}$, $\text{level}(\mathbf{v}_1) < \text{level}(\mathbf{v}_2)$ and $\text{level}(\mathbf{v}_2) = m$, for some natural number m . Then, at iteration $i + 1$, DecompositionTree will decompose \mathbf{v}_2 into two adjacent moves in $\mathcal{N}_{2^{m-1}}$, say \mathbf{v}_3 and \mathbf{v}_4 . Because all moves in $\mathcal{N}_{2^{m-1}}$ are also in \mathcal{N}_{2^m} , $\langle \mathbf{v}_3, \mathbf{v}_2, \mathbf{v}_4 \rangle$ or $\langle \mathbf{v}_4, \mathbf{v}_2, \mathbf{v}_3 \rangle$ is a subsequence of \mathcal{Q}_{m-2} . Similarly, since \mathbf{v}_1 is an adjacent move to \mathbf{v}_2 in \mathcal{N}_{2^m} , $\langle \mathbf{v}_1, \mathbf{v}_2 \rangle$ or $\langle \mathbf{v}_2, \mathbf{v}_1 \rangle$ is a subsequence of \mathcal{Q}_{m-2} . Then, $\mathbf{v}_1 = \mathbf{v}_3$ or $\mathbf{v}_1 = \mathbf{v}_4$. Again, without loss of generality assume $\mathbf{v}_1 = \mathbf{v}_3$. DecompositionTree will decompose each leaf node labeled with \mathbf{v}_2 into two new leaf nodes labeled with \mathbf{v}_1 and \mathbf{v}_4 , letting \mathbf{v}_1 and \mathbf{v}_4 be the moves that label the leaves of \mathcal{T} . Hence, at the end of the iteration $i + 1$, the leaves of \mathcal{T} are labeled with at most two adjacent moves in $\mathcal{N}_{2^{m-1}}$.

Figure 5 shows a graphical example of the output of DecompositionTree($((7, 5), k)$), with k equal to 7, 6, 5 and 4. It can also be seen as the state of \mathcal{T} at each iteration of DecompositionTree($((7, 5), 4)$). ■

Lemma 13 *Let p and q be relative primes, and let σ_{any}^* be an optimal path from origin to (p, q) . Moreover, let \mathcal{T} be the decomposition tree of level k for (p, q) , and let $\mu = \mathbf{m}_0, \mathbf{m}_1, \dots, \mathbf{m}_n$ be the sequence of labels of \mathcal{T} from left to right. Finally, let us define $\sigma_k = \mathbf{v}_0 \mathbf{v}_1 \dots \mathbf{v}_{n+1}$, where $\mathbf{v}_0 = (0, 0)$ and $\mathbf{v}_{i+1} = \mathbf{v}_i + \mathbf{m}_i$, for every $i \in \{0, \dots, n\}$. Then σ_k is an optimal path between origin and (p, q) .*

Proof: We prove that: (1) σ_k is a path, and (2) σ_k is optimal. For (1) we know there is some r for which $(p, q) \in \mathcal{N}_{2^r}$ (Proposition 7). Take now the decomposition tree of level $r - 1$. Because all moves in the leaves of such a tree are adjacent, by Lemma 11 we know that the path generated by these

moves exist because such moves do not visit more cells than those visited by the original move. If $r - 1$ is equal to k we are done proving that σ_k is a path. Otherwise, we repeat the same argument but this time with each of the moves obtained by the previous decomposition. Observe that, in doing so, we are actually generating the decomposition tree of level k . For an illustration, see Figure 3. This proves (1), that is that σ_k is a path. Now, for (2), we know that the decomposition tree has at most two adjacent moves labeling its leaves. This means that σ_k is optimal, because of Theorem 1. ■

With the above results, we proceed to give a proof of Theorem 2.

Proof of Theorem 2 : Let σ_{any}^* be generated by moves $\mathbf{m}_0^* \mathbf{m}_1^* \dots \mathbf{m}_n^*$. Without loss of generality, we assume each of those moves is a pair of the form (p, q) , where p and q are relative primes (otherwise, if any move is a multiple of another, we split such a move into multiple ones). Now from Lemma 13, we know each one of these moves can be ‘followed’ by a sequence of applicable moves in \mathcal{N}_{2^k} . Let μ_i be the moves in \mathcal{N}_{2^k} that follow \mathbf{m}_i^* , and let us call σ_k^f the path that is generated by applying such moves. Then by Theorem 10:

$$c(\mu_i) \leq c(\mathbf{m}_i^*) \Gamma_k$$

Summing up over $i \in \{0, \dots, n\}$, we obtain $c(\sigma_k^f) \leq c(\sigma_{any}^*) \Gamma_k$. Finally, the cost of the optimal path σ_k^* over \mathcal{N}_{2^k} is lower than the cost of σ_k^f . This allows us to write $c(\sigma_k^*) \leq c(\sigma_{any}^*) \Gamma_k$, our desired result. ■

Vertices Placed at the Centers of Grid Cells

Now we focus our attention on the case where vertices are placed at the center of grid cells. As before, we have a path planning problem P and optimal solutions σ_k^* and σ_{any}^* over \mathcal{N}_{2^k} and \mathcal{N}_{any} , respectively. We look for an upper bound for $c(\sigma_k^*)/c(\sigma_{any}^*)$. Intuitively, the main difference from the previous case is that σ_k^* cannot ‘touch’ obstacles (because there are no vertices at their corners), while σ_{any}^* can.

Suboptimality bounds for 4- and 8-connected neighborhoods are known for this case (Bailey et al. 2011); respectively, $6/(2 + \sqrt{2})$ and $3\sqrt{2} - 3$. Bailey et al.’s key idea is to use the instances shown in Figure 6 as path planning problems where the bound is given, and then proves such a ratio cannot be worsened. Then, since bounds for 4- and 8-connected neighborhoods are known, we are interested to find bounds for 2^k -neighborhoods, for every $k > 3$. We will prove this bounds are all equal to $3\sqrt{2} - 3$, the 8-connected bound. This happens because, intuitively, \mathcal{N}_{2^k} “can not do better” in the Bailey et al.’s 8-connected instance.

Proof of Theorem 3 : For $k = 2$ the proof is straight forward from Bailey et al.’s results. Now, for $k \geq 3$ we will perform an induction over k . For the base case ($k = 3$), again, it is straight forward from Bailey et al.’s results. Now, assume it is true for every $k \in \{3, 4, \dots, n - 1\}$, we will prove it is true for $k = n$.

Assume by contradiction that exists a P path planning problem where vertices are placed at the centers of grid cells such that $c(\sigma_k^*)/c(\sigma_{any}^*) > \Gamma_k^{center}$. Let σ_{k-1}^* be an

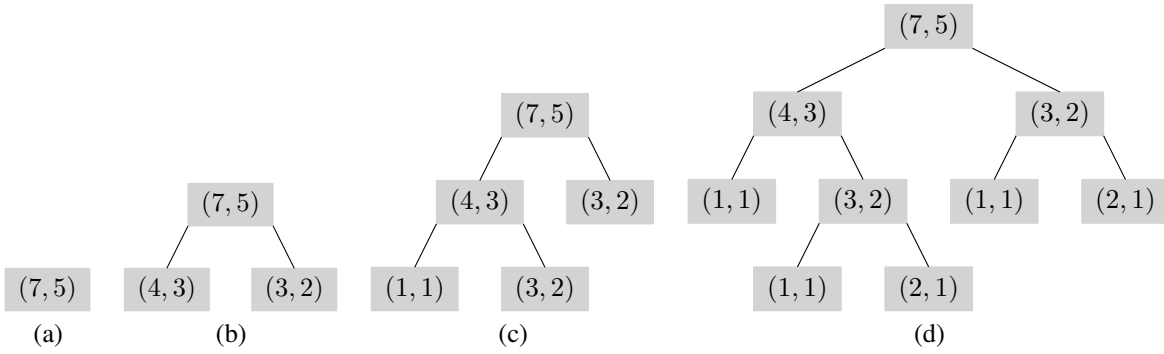


Figure 5: Decomposition Trees for the move (7, 5) of levels 7, 6, 5 and 4.

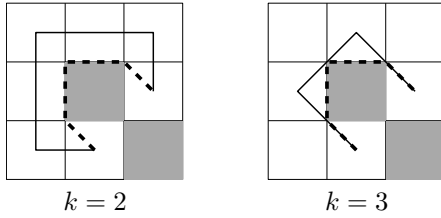


Figure 6: A worst-case path for 4- and 8-connected grids.

optimal path for P over $\mathcal{N}_{2^{k-1}}$. σ_{k-1}^* is not necessarily an optimal path for P over \mathcal{N}_{2^k} . Then, $c(\sigma_{k-1}^*) \geq c(\sigma_k^*)$ and $c(\sigma_{k-1}^*)/c(\sigma_{any}^*) \geq c(\sigma_k^*)/c(\sigma_{any}^*) > \Gamma_k^{center}$. By induction hypothesis $c(\sigma_{k-1}^*)/c(\sigma_{any}^*) \leq \Gamma_k^{center}$, leading us to a contradiction. ■

Empirical Evaluation

The objective of our empirical evaluation was to compare the average and maximum experimental suboptimality with our theoretical (worst-case) suboptimality on instances used in the literature to evaluate grid path planning algorithms. Because the most interesting setting is when vertices are placed at the corners, we focus only on such a case.

We implemented A* for 2^k -neighbor grids on top of Uras and Koenig’s code (2015). We used Anya (Harabor et al. 2016) to compute optimal any-angle solutions. The experiments were performed on a 2.20GHz Intel(R) Xeon(R) CPU machine with 128GB of RAM.

We calculated the experimental suboptimality of 6,282 instances of Baldur’s Gate maps (BG), 45,760 instances of Random maps (Random), and 39,460. Both the maps and the instances were taken from Moving AI repository (Sturtevant 2012). In Figure 7, we compare average experimental suboptimality (ratio-average) and maximum experimental suboptimality (ratio-max), for several values for k . ratio-max is very close to 1 in the three maps, in fact in BG with neighborhood 128 the ratio-max is exactly 1. In addition, we can observe that the ratio-average is more close to the theoretical bound when the neighborhood increases. We conclude that average suboptimality of 2^k is only about 3% under our theoretical subop-

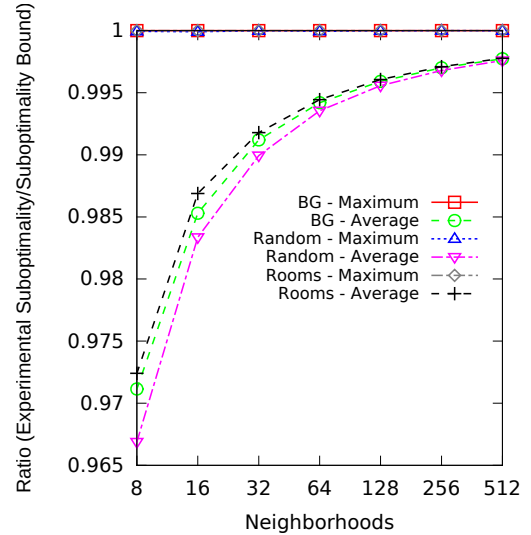


Figure 7: Experimental suboptimality compared with the suboptimality bound.

timality, which indicates that using our theoretical bound to make practical decisions about what k to use when aiming at a desired target suboptimality.

Summary and Conclusions

We presented theoretical bounds for the suboptimality of 2^k grid path planning, for any $k \geq 2$, and for two relevant cases: when search vertices are placed at the corners or at the centers of the cells. When vertices are at the corners, our bound asymptotically approaches 1 at a rate quadratic in k . When vertices are at the centers the bound does not approach to 1 as k increases, suggesting that implementations based on this scheme may not exploit the full potential of 2^k neighborhoods. In an experimental evaluation in which we use standard benchmarks, we show that the average suboptimality is less than 2% away from our theoretical bound. When given a target suboptimality, therefore, our theoretical bound can be used to determine which value of k to use along with a 2^k neighborhood.

Acknowledgements

We acknowledge support from Fondecyt via grants number 1161526.

References

- Bailey, J.; Tovey, C.; Uras, T.; Koenig, S.; and Nash, A. 2011. Path planning on grids: The effect of vertex placement on path length. In *Proceedings of the 11th Annual International AIIDE Conference (AIIDE)*. Palo Alto, California: AAAI Press.
- Daniel, K.; Nash, A.; Koenig, S.; and Felner, A. 2010. Theta*: Any-angle path planning on grids. *Journal of Artificial Intelligence Research* 39:533–579.
- Harabor, D. D., and Grastien, A. 2011. Online graph pruning for pathfinding on grid maps. In Burgard, W., and Roth, D., eds., *Proceedings of the 25th AAAI Conference on Artificial Intelligence (AAAI)*, 1114–1119. AAAI Press.
- Harabor, D. D.; Grastien, A.; Öz, D.; and Aksakalli, V. 2016. Optimal Any-Angle Pathfinding In Practice. *Journal of Artificial Intelligence Research* 56:89–118.
- Hormazábal, N.; Díaz, A.; Hernández, C.; and Baier, J. A. 2017. Fast and Almost Optimal Any-Angle Pathfinding Using the 2^k Neighborhoods. In *Proceedings of the 10th Symposium on Combinatorial Search (SoCS)*, 139–143. AAAI Press.
- Nash, A. 2012. *Any-Angle Path Planning*. PhD Thesis, University of Southern California.
- Oh, S., and Leong, H. W. 2017. Edge n-level sparse visibility graphs: Fast optimal any-angle pathfinding using hierarchical taut paths. In *Proceedings of the 10th Symposium on Combinatorial Search (SoCS)*, 64–72. AAAI Press.
- Rivera, N.; Hernández, C.; and Baier, J. A. 2017. Grid Pathfinding on the 2^k Neighborhoods. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI)*, 891–897. San Francisco, CA: AAAI Press.
- Sturtevant, N. R., and Rabin, S. 2016. Canonical orderings on grids. In Kambhampati, S., ed., *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)*, 683–689. IJCAI/AAAI Press.
- Sturtevant, N. 2012. Benchmarks for grid-based pathfinding. *Transactions on Computational Intelligence and AI in Games* 4(2):144 – 148.
- Uras, T., and Koenig, S. 2015. An empirical comparison of any-angle path-planning algorithms. In Lelis, L., and Stern, R., eds., *Proceedings of the 8th Symposium on Combinatorial Search (SoCS)*, 206–211. AAAI Press. Code available at: <http://idm-lab.org/anyangle>.
- Uras, T.; Koenig, S.; and Hernández, C. 2013. Subgoal graphs for optimal pathfinding in eight-neighbor grids. In Borrajo, D.; Kambhampati, S.; Oddi, A.; and Fratini, S., eds., *Proceedings of the Twenty-Third International Conference on Automated Planning and Scheduling, ICAPS 2013, Rome, Italy, June 10-14, 2013*. AAAI Press.