Analysis of Greedy Robot-Navigation Methods

Apurva Mudgal Craig Tovey
Georgia Institute of Technology
College of Computing
801 Atlantic Drive
Atlanta, GA 30332-0280, USA
{apurva, ctovey}@cc.gatech.edu

Sven Koenig
University of Southern California
Computer Science Department
941 W 37th Street
Los Angeles, CA 90089-0781, USA
skoenig@usc.edu

Abstract

Robots often have to navigate robustly despite incomplete information about the terrain or their location in the terrain. In this case, they often use greedy methods to make planning tractable. In this paper, we analyze two such robot-navigation methods. The first method is Greedy Localization, which determines the location of a robot in known terrain by always moving it to the closest location from which it will make an observation that reduces the number of its possible locations, until it has reduced that number as much as possible. We reduce the upper bound on the number of movements of Greedy Localization from $O(n^{\frac{3}{2}})$ to $O(n \log n)$ on grid graphs and thus close to the known lower bound of $\Omega(n \log n / \log \log n)$, where n is the number of (unblocked) vertices of the graph that discretizes the terrain. The second method is Dynamic A* (D*), which is used on several prototypes of both urban reconnaissance and planetary robots. It moves a robot in initially unknown terrain from given start coordinates to given goal coordinates by always moving the robot on a shortest presumed unblocked path from its current coordinates to the goal coordinates, pretending that unknown terrain is unblocked, until it has reached the goal coordinates or there are no more presumed unblocked paths. We reduce the upper bound on the number of movements of D* from $O(n^{\frac{3}{2}})$ to $O(n \log^2 n)$ on arbitrary graphs and $O(n \log n)$ on planar graphs (including grid graphs) and thus close to the known lower bound of $\Omega(n \log n / \log \log n)$, where n is the number of (blocked and unblocked) vertices of the graph that discretizes the terrain.

1 Introduction

Robot-navigation problems with incomplete information are challenging because nondeterminism results in a large number of contingencies. These problems include localization (where the map is known but the location of the robot relative to the map is unknown and the objective is to move the robot until it has discovered its current location) and goal-directed navigation in unknown terrain (where the location of the robot relative to the map is known but the map is unknown and the objective is to move the robot to a given goal location). The sensors on-board a robot can typically sense the terrain only near its current location, and the robot thus has to interleave planning with movement to sense new parts of the terrain, either to discover more about its current location or to discover more about the map.

In this paper, we analyze two robot-navigation methods that both interleave planning with movement and use greedy (= myopic) planning approaches to make planning tractable. The first method is Greedy Localization, which moves a robot in known terrain so that it determines its initially unknown location. The second method is Dynamic A* (D*), which moves a robot in initially unknown terrain from known start to known goal coordinates. Both robot-navigation methods are simple to implement, easy to integrate into complete robot architectures, and seem to result in small travel distances in practice. We analyze their travel distances to understand whether the travel distances are indeed small in any kind of terrain or whether they were small only because of properties of the terrain used to test them experimentally. We model the terrain as graph (in practice, grid graphs are commonly used) and then analyze the worst-case number of edge traversals as a function of the size of the graph, measured by the number of its vertices n. Robots move so slowly that their task-completion times are completely dominated by their travel times. Hence this criterion is at least as important as the competitive ratio [7] because it is a guarantee on the task-completion time. We reduce the upper bound on the worst-case number of edge traversals of Greedy Localization from $O(n^{\frac{3}{2}})$ [12] to $O(n \log n)$ on grid graphs and thus close to the best known lower bound of $\Omega(n \log n / \log \log n)$ [12]. With a completely different analysis, we reduce the upper bound on the worst-case number of edge traversals of D* from $O(n^{\frac{3}{2}})$ [5, 11] to $O(n \log^2 n)$ on arbitrary graphs and $O(n \log n)$ on planar graphs (including grid graphs) and thus close to the best known lower bound of $\Omega(n \log n / \log \log n)$ [8], which holds even on grid graphs [11].

2 Greedy Localization

The localization problem on grids is defined as follows. The robot moves with no actuator uncertainty on the grid (with the usual north-east-south-west orientation). Each cell of the grid is either blocked (= untraversable) or unblocked (= traversable). The perimeter of the grid consists of blocked cells. The start cell of the robot is unblocked. The robot has an on-board compass and a map (including orientation and traversability information). However, it does not know in which cell it is located. The robot can always move from its current cell north, east, south or west to any neighboring cell. The movement succeeds if the cell is unblocked. The robot has *localized* when it has made a series of movements such that the observations made during those movements are sufficient to determine the location of the robot, or to determine that the location cannot be determined (if, for example, the robot is located within one of a pair of isomorphic connected components of the grid). After each movement, the robot is permitted to use all the observations made so far to decide in which direction to move next. The localization problem is to localize in a minimum number of movements. We assume in the following that the sensors of the robot always detect with no sensor uncertainty which of its four neighboring cells are blocked (and perhaps also the blockage status of additional cells).

Greedy Localization maintains the set of cells that the robot can be in given all observations that it has made so far. It always makes the robot execute a shortest sequence of movements so that all observations that it has made after the sequence of movements are guaranteed to reduce the number of cells that the robot can be in. It terminates once the robot cannot reduce the number any more. If the number is one after termination then the robot is localized, otherwise it cannot localize. Greedy Localization was pioneered by Nourbakhsh in robot programming classes where Nomad 150 mobile robots had to navigate mazes that were built with three-foot high and forty inch long cardboard walls [6]. Similar strategies are used in more complex environments where robots use probability distributions over locations rather than sets of locations to be able to deal with sensor noise, and the greedy localization methods then move the robots to decrease the entropy of the probability distribution rather than the cardinality of the set [3].

2.1 Worst-Case Travel Bound

Researchers often analyze localization methods using online criteria, by comparing the travel distance of a robot to the distance that would be traveled by an omniscient robot that knows its location at the outset and seeks only to verify that location. Dudek, Romanik, and Whitesides [2], for example, find a best possible online ratio of $|H|-1$, where H is computed from the start location of the robot with long range sensors in a polygonal model. We, on the other hand, analyze the worst-case travel distance of Greedy Localization because a small worst-case travel distance guarantees that it cannot perform very badly, which is more important than the "regret" measured by the competitive ratio. Our analysis makes heavy use of a lemma from [13].

We model the grid as a grid graph $G = (V, E)$ whose vertices correspond to the unblocked cells. The start vertex of the robot is x^0. The observation that the robot makes in a vertex is the same as the observation that it makes in the corresponding cell. Two vertices are connected via an edge iff they correspond to neighboring cells. The number of edge traversals of the robot on the grid graph then is the same as its travel distance on the grid.

Theorem 1 *Greedy Localization traverses at most* $|V| + 2|V|\ln|V|$ *edges on grid graphs* $G = (V, E)$.

Proof: The robot is always in a particular vertex in the graph, even though the robot does not know which one it is. For the purpose of the analysis, we follow the movements of the robot as they actually occur in the graph. During iteration i, the robot follows a shortest path from vertex x^{i-1} to vertex x^i so that the observation in x^i is guaranteed to reduce the number of vertices that the robot can be in. This implies that x^i is closest to x^{i-1} among all informative vertices, where a vertex is *informative* if the observation from the vertex is guaranteed to reduce the number of vertices that the robot can be in. Note that uninformative vertices remain uninformative. Let $d(x, x')^G$ denote the distance from vertex x to vertex x' in graph G and define $l^i = d(x^{i-1}, x^i)^G$. The main intuition behind the proof of the theorem is the fact that no vertex within a large distance of x^{i-1} can be informative if l^i is large. Thus, the number of large l^i and thus also the number of edge traversals of Greedy Localization must be small. We use marking sequences to formalize this intuition. A marking sequence on graph $G = (V, E)$ is a sequence of triples $\{v^i, r^i, M^i\}$ for $i = 1, 2, \ldots$, whose integers $r^i \geq 0$, vertices $v^i \in V$, and sets $M^i \subseteq V$ satisfy the following properties:

1. $v^i \notin M^i$,
2. $M^1 = \emptyset$ and $M^i \subset M^{i+1}$, and
3. $d(v, v^i)^G \leq r^i$ implies $v \in M^{i+1}$.

The *cost* of the marking sequence is $\sum_i (1 + r^i)$. Vertices v are considered to be *marked* at step i iff $v \in M^i$. Our key construct then is as follows: Greedy Localization forms an associated marking sequence where $r^i = l^i - 1$, $v^i = x^{i-1}$, and M^{i+1} is the set of uninformative vertices after the robot has reached and made an observation from v^i. The number of edge traversals of Greedy Localization equals the cost of the associated marking sequence since $1 + r^i = l^i$. Note that the marking sequence is less restrictive than Greedy Localization because v^i need not be at distance $1 + r^i$ from v^{i+1}. Instead, the marking sequence consists of a sequence of choices of an unmarked vertex v^i and a radius r^i. All vertices within distance r^i of v^i (and possibly additional vertices) are marked, and the marking sequence continues.

Lemma 1 *The cost of any marking sequence is no larger than $|V| + 2|V| \ln |V|$ on connected graphs $G = (V, E)$.*

Proof sketch: It follows from the triangle inequality that there exists a maximum cost marking sequence that only marks one vertex, namely v^i, per step. For if another vertex v were also marked, one could replace the step with two more expensive steps that mark only v^i and v, respectively. By viewing the marking sequence as a sequence of disjoint balls of radius r^i in the metric space of graph distances, the connectivity of the graph limits the number of radii that are at least t to $2|V|/t$. The lemma follows. Full details are given in [13].

Greedy Localization constrains the movements of the robot to be in a connected component of the graph. Hence, the lemma applies and the theorem is proved. ∎

Note that our proof is not specific to robots with short-range sensors that operate on two-dimensional grid graphs in which the potential neighbors of a cell are located only to its north, east, south, and west. Our proof does require that uninformative vertices remain uninformative. This is the case, for example, if the set of vertices that the robot believes it can be in when it is in some vertex x is always included in the set of vertices that the robot believed it could have been in when it was in the same vertex x earlier. Thus our theorem holds also, for example, for higher-dimensional or differently connected grid graphs and is completely independent of the kind of sensors used by the robot.

3 D*

The goal-acquisition problem on grid graphs is defined as follows. The robot moves again with no actuator uncertainty on a grid with blocked and unblocked cells. Its perimeter consists of blocked cells. The start cell of the robot is unblocked. The robot knows its start cell and orientation and has to move to a given goal cell. It does not know initially which cells are blocked. The robot can always move from its current cell north, east, south or west to any neighboring cell. The movement succeeds if the cell is unblocked. After each movement, the robot is permitted to use all the observations made so far (corresponding to the partial map that it has learned so far) to decide in which direction to move next. The goal-acquisition problem is move to the goal cell in a minimum number of movements. We assume in the following that the sensors of the robot always detect with no sensor uncertainty whether the cell that it attempts to move to is blocked.

D* maintains the partial map that the robot has learned so far. It always makes the robot execute a shortest sequence of movements from its current cell to the goal cell under the optimistic assumption that cells are unblocked that have not been observed to be blocked. It terminates once the robot has reached the goal cell or no such sequences of movements exist any longer (in which case the robot cannot reach the goal cell). Whenever the robot observes a blocked cell on its current path, D* needs to replan, which can be implemented efficiently [9] and easily [4]. D* has been used outdoors on an autonomous high-mobility multi-wheeled vehicle that navigated 1,410 meters to the goal location in an unknown area of flat terrain with sparse mounds of slag as well as trees, bushes, rocks, and debris [10]. As a result of this demonstration, D* is now widely used in the DARPA Unmanned Ground Vehicle (UGV) program, for example, on the UGV Demo II vehicles. D* is also being integrated into Mars Rover prototypes, tactical mobile robot prototypes and other military robot prototypes for urban reconnaissance.

3.1 Analysis

We analyze the worst-case travel distance of D* in the following. We model the grid as a grid graph whose vertices correspond to the cells. Thus, vertices can be blocked or unblocked. Two vertices are connected via an edge iff they correspond to neighboring cells. However, our analysis holds for graphs in general, not just grid graphs. We therefore generalize the problem as follows:

Consider a graph $G = (V, E)$ with blocked and unblocked vertices. We assume without loss of generality that the graph is connected (otherwise we can consider only the connected component of the graph that contains the start vertex). The start vertex v_0 of the robot is unblocked. The robot always knows its current vertex (for example, because the robot knows its start vertex) and has to move to a given goal vertex t. It does not know initially which vertices are blocked. The robot can attempt to move from its current vertex to any neighboring vertex. If the neighboring vertex is unblocked, then the robot moves to it. If the neighboring vertex is blocked then the robot remains in its current vertex. In both cases, the robot observes whether the neighboring vertex is blocked. D* always moves the robot along a shortest presumed unblocked path from its current vertex to the goal vertex. A path is *presumed unblocked* if it does not contain vertices that the robot knows to be blocked. Whenever the robot observes that a vertex is blocked, it recalculates a shortest presumed unblocked path from its current vertex to the goal vertex and repeats the process. D* terminates once the robot has reached the goal vertex or can no longer find a presumed unblocked path from its current vertex to the goal vertex.

We use the following notation: At the beginning of iteration i, the robot is in vertex v_{i-1} and E_i is the set of edges that are not incident on a vertex known to be blocked at that time. Note that $E_1 = E$. The robot then plans a shortest path P_i in graph $H^i = (V, E_i)$ from vertex v_{i-1} to t, starts to follow it, and then stops in vertex v_i either because $v_i = t$ or because the vertex b_i following v_i on P_i is blocked. In the latter case, let v'_i denote the vertex following b_i on P_i. The robot eventually stops in vertex v_k either because $v_k = t$ or because there are no longer any presumed unblocked paths from v_k to t. Thus, $k \leq |V|$. Let $d(x, x')^G$ denote the distance from vertex x to vertex x' in graph G. If x and x' are not in the same connected component of G then $d(x, x')^G = \infty$. The travel distance of the robot is

$$C = \sum_{i=1}^{k} d(v_{i-1}, v_i)^{H^i}.$$

3.2 Telescoping

Lemma 2 *D* traverses at most* $|V| + \sum_{i=1}^{k-1} d(v_i, v'_i)^{H^{i+1}}$ *edges on connected graphs* $G = (V, E)$.

Proof: Since v_i lies on the shortest path P_i from v_{i-1} to t in H^i, by the principle of optimality,

$$\begin{aligned}
C &= \sum_{i=1}^{k} d(v_{i-1}, v_i)^{H^i} \\
&= \sum_{i=1}^{k} (d(v_{i-1}, t)^{H^i} - d(v_i, t)^{H^i}) \\
&= d(v_0, t)^{H^1} - d(v_k, t)^{H^k} + \sum_{i=1}^{k-1} (d(v_i, t)^{H^{i+1}} - d(v_i, t)^{H^i}) \\
&\leq |V| + \sum_{i=1}^{k-1} (d(v_i, t)^{H^{i+1}} - d(v_i, t)^{H^i}).
\end{aligned}$$

The last inequality uses $d(v_0, t)^{H^1} < |V|$ which holds since $G = (V, E) = (V, E_1) = H^1$ is connected.

Now consider an arbitrary i with $1 \leq i < k$. The robot planned a shortest path P_i in H^i from v_{i-1} via v_i, b_i, and v'_i to t. Thus, the subpath of P_i from v'_i to t is a shortest path in H^i from

v_i' to t. Since it does not contain any edges incident on b_i, it is also a shortest path in H^{i+1} from v_i' to t and thus $d(v_i', t)^{H^{i+1}} = d(v_i', t)^{H^i}$. Thus, by the triangle inequality,

$$d(v_i, t)^{H^{i+1}} \leq d(v_i, v_i')^{H^{i+1}} + d(v_i', t)^{H^{i+1}} = d(v_i, v_i')^{H^{i+1}} + d(v_i', t)^{H^i}.$$

By the definition of v_i', we have $d(v_i, t)^{H^i} = 2 + d(v_i', t)^{H^i}$. Thus, $C \leq |V| + \sum_{i=1}^{k-1}(d(v_i, t)^{H^{i+1}} - d(v_i, t)^{H^i}) \leq |V| + \sum_{i=1}^{k-1}((d(v_i, v_i')^{H^{i+1}} + d(v_i', t)^{H^i}) - (d(v_i', t)^{H^i} + 2)) \leq |V| + \sum_{i=1}^{k-1} d(v_i, v_i')^{H^{i+1}}$.
■

3.3 Time Reversal and Weighted Edges

Consider the following function:

CYCLE-WEIGHT(T,S). Input: a tree $T = (V, E)$ and an ordered list $S = \{e_j : 1 \leq j \leq k\}$ of distinct edges from the complete graph on V such that $S \cap E = \phi$. Define the weight w_i of edge $e_i \in S$ to be the length of a shortest cycle that contains e_i in the graph $T_i = (V, E \cup \{e_j : i \leq j \leq k\})$. Output: $\sum_{i=1}^{k} w_i$.

We now show that $\sum_{i=1}^{k-1} d(v_i, v_i')^{H^{i+1}} \leq$ CYCLE-WEIGHT(T, S) for a suitably constructed tree T and $S = \{e_j = (v_j, b_j) : 1 \leq j < k\}$.

The basic idea relating the edge weights in CYCLE-WEIGHT to the $d(v_i, v_i')^{H^{i+1}}$ values can be understood by considering a special case. Assume that H^k is connected except for the isolated vertices $b_1, b_2, \ldots, b_{k-1}$. Reverse the time perspective so that the robot movements adds edges, first the edges incident on b_{k-1}, then the edges incident on b_{k-2}, and so on. Pick T to be a spanning tree of the graph $(V, E_k \cup \{(b_j, v_j') : 1 \leq j < k\})$ and S to be $\{e_j = (v_j, b_j) : 1 \leq j < k\}$. Then, $w_i \geq 2 + d(v_i, v_i')^{H^{i+1}}$ for $1 \leq i < k$ since every cycle that contains $e_i = (v_i, b_i)$ in T_i must also contain (b_i, v_i'). Consequently, $\sum_{i=1}^{k-1} d(v_i, v_i')^{H^{i+1}} \leq \sum_{i=1}^{k-1} w_i =$ CYCLE-WEIGHT(T, S).

Unfortunately this simple construction does not work in the general case since multiple connected components may be formed when the edges incident on a blocked vertex are removed. To get around this problem, we define the sequence of graphs $F_k, F_{k-1}, \ldots, F_1$ as follows:

- Let F_k be a spanning forest of H^k.
- For $1 \leq i < k$, let $C_1^i, C_2^i, \ldots, C_{k_i}^i$ be the connected components of H^{i+1} which get merged with b_i in H^i, with the restriction that $v_i' \in C_1^i$. Select a $w_j^i \in C_j^i$ for $1 \leq j \leq k_i$ such that w_j^i is a neighbor of b_i in G, with the restriction that $w_1^i = v_i'$. Let F_i result from F_{i+1} by adding the edges $\{(b_i, w_j^i) : 1 \leq j \leq k_i\}$.

The following lemma is immediate:

Lemma 3 *For* $1 \leq i \leq k$ *and all vertices* u *and* v*,* F_i *is acyclic;* $d(u,v)^{F_i} < \infty$ *iff* $d(u,v)^{H_i} < \infty$*; and* $d(u,v)^{F_i} \geq d(u,v)^{H^i}$*.*

Proof: By induction. ■

We are now ready to prove the bound.

Lemma 4 *Let* $H^1, H^2, \ldots, H^k$ *be a sequence of graphs as defined above. Let* $T = F_1$ *and* $S = \{e_i = (v_i, b_i) : 1 \leq i < k\}$*. Then* $\sum_{i=1}^{k-1} d(v_i, v_i')^{H^{i+1}} \leq$ *CYCLE-WEIGHT*(T, S)*.*

Proof: By lemma 3, F_{i+1} and H^{i+1} have the same connected components. The subgraph of F_1 induced by C_1^i is connected since C_1^i is a component of H^{i+1}. The edges e_j for $i < j < k$ are contained in C_1^i since $v_j, v_j', b_j \in C_1^i$ for all $i < j < k$. Thus, the graph obtained by contracting all vertices of C_1^i in T_{i+1} is acyclic. Since T_i is obtained from T_{i+1} by adding e_i, every cycle that contains $e_i = (v_i, b_i)$ in T_i must also contain (b_i, v_i'). Thus, w_i is equal to 2 plus the distance between v_i and v_i' in the subgraph G' of T_i induced by C_1^i. But G' is also a subgraph of H^{i+1} and hence $w_i \geq 2 + d(v_i, v_i')^{H^{i+1}}$. Consequently, $\sum_{i=1}^{k-1} d(v_i, v_i')^{H^{i+1}} \leq \sum_{i=1}^{k-1} w_i =$ CYCLE-WEIGHT(T, S). ■

3.4 An Extremal Problem on Graphs

We now bound CYCLE-WEIGHT$((V, E), S)$ in terms of $|V|$ and $|S|$. Let $E_w = \{e_i : w_i \geq w\} \subseteq S$ be the set of edges with weight at least w. Recall that the *girth* of a graph is the length of its shortest cycle. Define $\Gamma(n, w)$ [and $\Gamma_P(n, w)$] to denote the maximum number of edges in graphs [and, respectively, planar graphs] with n vertices and a girth of at least w. The following lemma relates E_w and $\Gamma(n, w)$.

Lemma 5 $|E_w| \leq \Gamma(|V|, w) - |V| + 1$ *for all* w *and CYCLE-WEIGHT*$((V, E), S)$*.*

Proof: Consider the graph $T_w = (V, E \cup E_w)$. We claim that T_w has a girth of at least w. To see this, assume that it does not and thus has a cycle C of length $w' < w$. Since (V, E) is a tree, at least one edge of C must belong to E_w. Consider the edge $e_j \in E_w \cap C$ with the smallest j. Then T_j contains C and thus $w_j \leq w' < w$. On the other hand, $w_j \geq w$ since $e_j \in E_w$, which is a contradiction. Thus, T_w has a girth of at least w. This implies that $\Gamma(|V|, w) \geq |E \cup E_w| = |E| + |E_w| = |V| - 1 + |E_w|$ and the lemma follows. ■

Corollary 1 $|E_w| \leq \Gamma_P(|V|, w) - |V| + 1$ *for all* w *and CYCLE-WEIGHT*$((V, E), S)$ *such that* $(V, E \cup S)$ *is planar.*

Proof: In the proof of lemma 5, T_w is planar because it is a subgraph of the planar graph $(V, E \cup S)$. Hence $\Gamma(|V|, w)$ may be replaced by $\Gamma_P(|V|, w)$. ■

We now bound CYCLE-WEIGHT$((V, E), S)$ by making use of bounds on $\Gamma(n, w)$ and $\Gamma_P(n, w)$, two well studied problems in extremal combinatorics. We first consider the case where the graph $(V, E \cup S)$ is planar.

Lemma 6 $\Gamma_P(n, w) \leq \frac{wn}{w-2}$ *for all* w *and* n.

Proof: Since the sum of the lengths of all faces of any planar graph $G = (V, E)$ is at most $2|E|$ and every face has a length of at least w, the number of its faces can be at most $2|E|/w$. The bound of the lemma follows from substituting this relationship in Euler's formula. ■

Note that the weight of any edge in S is at most $|V|$. Define $E_{w,2w} = \{e_i \in S : w \leq w_i < 2w\}$. Then, by corollary 1 and lemma 6,

$$\begin{aligned}
\text{CYCLE-WEIGHT}((V, E), S) &\leq \sum_{i=1}^{\log|V|} 2^{i+1}|E_{2^i,2^{i+1}}| \\
&\leq O(|S|) + \sum_{i=3}^{\log|V|} 2^{i+1}|E_{2^i}| \\
&\leq O(|S|) + \sum_{i=3}^{\log|V|} 2^{i+1}(\Gamma_P(|V|, 2^i) - |V| + 1) \\
&\leq O(|S|) + \sum_{i=3}^{\log|V|} 2^{i+1}(\frac{2^i|V|}{2^i - 2} - |V| + 1) \\
&\leq O(|S|) + \sum_{i=3}^{\log|V|} 2^{i+1}\, 4|V|/2^i \\
&= O(|S|) + \sum_{i=3}^{\log|V|} 8|V| \\
&= O(|S| + |V| \log|V|).
\end{aligned}$$

We now repeat the analysis for general graphs. In this case, we use a recent result by Alon, Hoory and Linial that states that any graph $G = (V, E)$ with average degree $d > 2$ has a girth of at most $\log_{d-1}|V|$ [1], resulting in the following lemma.

Lemma 7 $\Gamma(n, w) \leq n(n^{\frac{1}{w}} + 1)/2$ *for all* w *and* n.

Proof: Consider any graph $G = (V, E)$ with $|V| = n$, $|E| \geq |V| + 1$ and a girth of at least w. Then, its average degree is $d = 2|E|/n > 2$ and thus, by the result by Alon, Hoory and Linial, $w \leq \log_{2|E|/n-1} n$. Solving this inequality for $|E|$ yields the lemma. ■

This lemma allows us to bound CYCLE-WEIGHT$((V, E), S)$ for general graphs. Using calculus, we can show that $w(|V|(|V|^{\frac{1}{w}} - 1)) = O(|V| \log |V|)$ for $|V| \geq w > \log^2 |V|$. Using this fact with lemmata 5 and 7, we have

$$\begin{aligned}
\text{CYCLE-WEIGHT}((V, E), S) &= \sum_{i:w_i \leq \log^2 |V|} w_i + \sum_{i:w_i > \log^2 |V|} w_i \\
&\leq |S| \log^2 |V| + \sum_{i=2 \log\log |V|}^{\log |V|} 2^{i+1} |E_{2^i, 2^{i+1}}| \\
&\leq |S| \log^2 |V| + \sum_{i=2 \log\log |V|}^{\log |V|} 2^{i+1} |E_{2^i}| \\
&\leq |S| \log^2 |V| + \sum_{i=2 \log\log |V|}^{\log |V|} 2^{i+1} (\Gamma(|V|, 2^i) - |V| + 1) \\
&= |S| \log^2 |V| + \sum_{i=2 \log\log |V|}^{\log |V|} 2^{i+1} (|V|(|V|^{\frac{1}{2^i}} - 1)/2 + 1) \\
&= |S| \log^2 |V| + \sum_{i=2 \log\log |V|}^{\log |V|} O(|V| \log |V|) \\
&= O((|V| + |S|) \log^2 |V|).
\end{aligned}$$

We now state these results as lemma.

Lemma 8 *CYCLE-WEIGHT$((V, E), S) = O((|V| + |S|) \log^2 |V|)$ for all CYCLE-WEIGHT$((V, E), S)$. CYCLE-WEIGHT$((V, E), S) = O(|S| + |V| \log |V|)$ for all CYCLE-WEIGHT$((V, E), S)$ such that $(V, E \cup S)$ is planar.*

3.5 Worst-Case Travel Bound

We are now ready to prove an upper bound on the worst-case travel distance of D*.

Theorem 2 *D* traverses $O(|V| \log |V|)$ edges on connected graphs $G = (V, E)$. It traverses $O(|V| \log^2 |V|)$ edges on connected planar graphs $G = (V, E)$.*

Proof: By lemmata 2 and 4, D* traverses at most $|V| + \sum_{i=1}^{k-1} d(v_i, v_i')^{H^{i+1}} \leq |V| +$ CYCLE-WEIGHT$((V, E'), S)$ edges, where $|S| < |V|$ and $(V, E' \cup S)$ is a subgraph of G. By lemma 8, CYCLE-WEIGHT$((V, E'), S) = O((|V| + |S|) \log^2 |V|) = O(|V| \log^2 |V|)$ and, if G and thus $(V, E' \cup S)$ are planar, CYCLE-WEIGHT$((V, E'), S) = O(|S| + |V| \log |V|)$. The theorem follows. ■

4 Conclusions

The robot-navigation methods that we have analyzed in this paper, Greedy Localization and D*, are appealingly simple and easy to implement from a robotics point of view and appealingly complicated to analyze from a mathematical point of view. Our results, likewise, are satisfying in two ways. First, our tighter upper bounds on their worst-case travel distances guarantee that they cannot perform badly at all. Second, the gaps between the best known lower and upper bounds are now quite small, namely $O(\log \log n)$ for Greedy Localization on grid graphs and D* on planar graphs (including grid graphs), and $O(\log n \log \log n)$ for D* on arbitrary graphs.

References

[1] N. Alon, S. Hoory, and N. Linial. The Moore bound for irregular graphs. *Graph and Combinatorics*, 18(1):53–57, 2002.

[2] G. Dudek, K. Romanik, and S. Whitesides. Localizing a robot with minimum travel. In *Proceedings of the Annual ACM-SIAM Sympsosium on Discrete Algorithms*, pages 437–446, 1995.

[3] D. Fox, W. Burgard, and S. Thrun. Active Markov localization for mobile robots. *Robotics and Autonomous Systems*, 25:195–207, 1998.

[4] S. Koenig and M. Likhachev. Improved fast replanning for robot navigation in unknown terrain. In *Proceedings of the International Conference on Robotics and Automation*, pages 968–975, 2002.

[5] S. Koenig, C. Tovey, and Y. Smirnov. Performance bounds for planning in unknown terrain. *Artificial Intelligence*, 2002.

[6] I. Nourbakhsh. *Robot Information Packet*. Distributed at the AAAI-96 Spring Symposium on Planning with Infomplete Information for Robot Problems, 1996.

[7] D. Sleator and R. Tarjan. Amortized efficiency of list update and paging rules. *Communications of the ACM*, 28(2):202–208, 1985.

[8] Y. Smirnov. *Hybrid Algorithms for On-Line Search and Combinatorial Optimization Problems*. PhD thesis, School of Computer Science, Carnegie Mellon University, Pittsburgh (Pennsylvania), 1997. Available as Technical Report CMU-CS-97-171.

[9] A. Stentz. The focussed D* algorithm for real-time replanning. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 1652–1659, 1995.

[10] A. Stentz and M. Hebert. A complete navigation system for goal acquisition in unknown environments. *Autonomous Robots*, 2(2):127–145, 1995.

[11] C. Tovey, S. Greenberg, and S. Koenig. Improved analysis of D*. In *Proceedings of the International Conference on Robotics and Automation*, 2003.

[12] C. Tovey and S. Koenig. Gridworlds as testbeds for planning with incomplete information. In *Proceedings of the National Conference on Artificial Intelligence*, pages 819–824, 2000.

[13] C. Tovey and S. Koenig. Improved analysis of greedy mapping. In *Proceedings of the International Conference on Intelligent Robots and Systems*, 2003.