POLITECNICO DI MILANO
Corso di Laurea Magistrale in Ingegneria Informatica
Dipartimento di Elettronica, Informazione e Bioingegneria

Online Exploration of Graphs with an Autonomous Robot: A Theoretical Analysis

AIRLab
Artificial Intelligence and Robotics Laboratory

Relatore: Prof. Francesco Amigoni
Correlatore: Dott. Alberto Quattrini Li

Tesi di Laurea Magistrale di:
Alessandro Riva, matricola 800753

Anno Accademico 2013-2014

Alla mia famiglia

Contents

List of Figures

List of Tables

Abstract

Exploration of unknown environments plays a significant role in many mobile robot applications, like map building, coverage, and searching. In literature, exploration strategies are usually defined and evaluated following two rather different approaches. On the one hand, they are defined in practical contexts of real (or realistically simulated) robots and are empirically assessed by testing them in some environments. On the other hand, exploration strategies are defined in theoretical settings (e.g., exploration of graphs) and are assessed using theoretical tools like worst-case bounds. Being inherently an online task, exploration is usually greedily addressed by letting a robot evaluate some candidate destination locations in order to choose where to go next. In this thesis, we provide a theoretical analysis of some exploration strategies. We consider a single robot exploring an initially unknown environment represented by an undirected graph. The goal of the robot is to find an exploration path (not necessarily closed) that perceives a given fraction of vertices of the graph with the minimum number of edge traversals (atomic movements from a vertex to an adjacent vertex). We assume that the robot learns of the vertices, and of the corresponding edges, within a given perception range from its current position. Our results significantly complement some of the worst-case bounds on the number of edge traversals required to explore a generic graph presented in the literature, explicitly embedding in the analysis the sensor range of the robot, and considering exploration strategies based on information gain and on combination of information gain with distance. We also provide an average-case analysis (which, to the best of our knowledge, has never appeared in the literature) of exploration strategies in classes of graphs that model realistic indoor environments. The main contribution of this thesis is a theoretical analysis that integrates and possibly better explains the experimental results obtained with real (and realistically simulated) exploring robots.

Sommario

L'esplorazione di ambienti sconosciuti è fondamentale in molte applicazioni in cui sono utilizzati robot mobili, come la costruzione della mappa di un edificio, la ricerca e i problemi di covering. In letteratura, le strategie di esplorazione sono solitamente definite e valutate in due modi differenti. Da una parte, sono definite in un contesto pratico, con robot reali (o realisticamente simulati) e valutate empiricamente attraverso test in diversi ambienti. Dall'altra parte, sono definite in un contesto teorico (e.g., esplorazione di grafi) e valutate attraverso strumenti teorici, come limiti nel caso pessimo. Essendo un'attività effettuata online, l'esplorazione è tipicamente affrontata in modo greedy, lasciando al robot la valutazione delle possibili posizioni future in cui muoversi. In questa tesi sono analizzate, da un punto di vista teorico, alcune strategie di esplorazione. Lo scenario considerato è quello in cui un singolo robot mobile esplora un ambiente inizialmente sconosciuto, rappresentato da un grafo non orientato. L'obiettivo del robot è trovare un percorso di esplorazione (non necessariamente chiuso) che percepisca una certa percentuale di vertici del grafo, con il minor numero possibile di archi attraversati (movimenti atomici da un vertice a un vertice adiacente). Il robot percepisce i vertici e i loro archi corrispondenti, all'interno di un certo raggio di percezione dalla sua posizione corrente. I risultati di questa tesi completano in modo significativo alcuni dei limiti relativi al caso peggiore (rispetto al numero di attraversamenti di archi richiesto per esplorare un generico grafo) presenti in letteratura, considerando esplicitamente nell'analisi il raggio del sensore del robot, e considerando strategie di esplorazione basate sull'information gain o sulla combinazione fra information gain e distanza. In questa tesi vengono anche analizzate, dal punto di vista del caso medio, alcune classi di grafi che modellano ambienti indoor realistici. Il principale contributo di questa tesi è un'analisi teorica che ha lo scopo di contribuire a spiegare meglio i risultati sperimentali ottenuti nell'esplorazione con robot reali o realisticamente simulati.

Ringraziamenti

Desidero ringraziare il professor Francesco Amigoni per la grande umanità e la grande disponibilità mostratami durante tutto il lavoro di tesi, nonché per gli utili consigli e il tempo speso ad aiutarmi.

Ringrazio il dottor Alberto Quattrini Li, che è stato per me un'ottima guida, dandomi sempre tutto l'aiuto di cui avevo bisogno. Non avrei potuto sperare di trovare una persona migliore.

Un ringraziamento speciale va alla mia famiglia, a cui devo questi cinque anni di soddisfazioni e fatiche, che mi hanno reso la persona che sono. Grazie per avermi sempre supportato; per essermi stati vicini nei momenti più duri così come in quelli più felici.

Ringrazio tutte le persone conosciute in questi anni, con cui ho condiviso gioie e ansie, dentro e fuori dall'università. In particolare Marco, compagno di tanti progetti e di moltissime discussioni.

Infine ringrazio i miei amici più cari, per essermi sempre stati d'aiuto con la loro presenza e le loro parole.

Chapter 1

Introduction

Exploration is a fundamental task for autonomous mobile robots that operate in a large number of applications, such as map building [Thrun, 2002], coverage [Choset, 2001], and search [Calisi et al., 2005]. A robot placed in an initially unknown environment has to discover the position of the obstacles. Specifically, the robot iteratively perceives the surrounding environment through a sensor, integrates the sensor data into a map, chooses where to move, and go to the selected destination location. The mainstream approach to exploration of initially unknown environments is greedy [Tovey and Koenig, 2003]. This Next Best View approach operates by cyclically evaluating some candidate destination locations that are usually selected on the *frontiers* between the known and the unknown portions of the environment [Yamauchi, 1997]. The evaluation is performed according to an *exploration strategy* that considers different criteria in a utility function measuring the appeal of candidate locations. A simple exploration strategy considers only the distance from the current position of the robot, and selects the closest candidate location [Yamauchi, 1997]. This simple criterion is rather popular, especially in practical applications, even if there exists strategies which perform better in the worst case. For instance, one of these strategies is the *depth-first search*, which always moves the robot from its current position to an adjacent candidate location. If such adjacent candidate location does not exist, depth-first search back-tracks to reach the next closest candidate location. It is easy to prove that the traveled distance for this kind of strategy, in the worst case, is linear with respect to the number of different candidate locations in the environment, while, greedy strategies may becomes super linear, as shown for the distance criterion in [Koenig et al., 2001]. Nevertheless, depth-first search does not have the following good properties, which, instead, characterize greedy strategies (as noted in [Koenig et al., 2001]):

- **Simple Integration into Robot Architectures:** greedy strategies are robust with respect to the errors of robot components. Moreover, if a robot has to preempt exploration to reach some known location (for instance a power outlet, to recharge the batteries), then it should be able to resume exploration from that location, instead of having to come back to the location where exploration was stopped. Greedy strategies exhibit this behavior by definition, while, depth-first search is strongly sensible to the last position reached (because of the back-tracking).

- **Prior Knowledge:** greedy strategies can benefit of prior knowledge about portions of the environment (possibly previously acquired by the robot or provided to it). The reason is that they use all of their knowledge about the environment when determining which unvisited (or unperceived) location is closest to the robot and how to get there quickly. Instead, the basic definition of depth-first search does not consider that knowledge, since it could lead to inconsistent states (how is back-tracking defined on a prior known area?).

- **Distributed Search:** greedy strategies can be performed in parallel, allowing to explore an unknown environment through several robots that share information on parts of the environment they explored individually. For depth-first search, the robot, sharing information, may have the same problems of having prior knowledge about parts of the environment.

The back-tracking, which is the strong point of depth-first search in the worst case, becomes easily a strong limitation. Thus, because of their flexibility, greedy strategies are widely adopted in real applications (see, e.g., [Thrun et al., 1998]). Other works on greedy strategies include also criteria related to the expected information gain of the candidate locations. For example, in [Stachniss and Burgard, 2003] the cost of reaching a candidate location is linearly combined with its benefit, while the method of [Gonzáles-Baños and Latombe, 2002] combines the distance and the expected information gain of the candidate location in an exponential function. The assessment of such exploration strategies (those that exploits information gain) performed in the field of robotics is mainly empirical. A number of methods are experimentally evaluated in selected settings and their performance compared [Amigoni, 2008, Julia et al., 2012]. While the computational geometry and the theoretical computer science communities have studied the problem of exploration, the derived bounds are often relative to specific, and sometimes not fully realistic, contexts (e.g., rectilinear polygonal environments [Deng et al., 1998] or closed tours for graph exploration [Kalyanasundaram and Pruhs, 1994]). To the best of our knowledge, very few works have considered more realistic settings for providing bounds on the quality of solutions produced by exploration strategies, prominently the work in [Tovey and Koenig, 2003]. In their approach, a single robot should explore all the vertices of an undirected graph, whose edges have unitary cost, with a sensor that allows to perceive the current vertex and an arbitrary number of other vertices. An upper bound independent of the sensor range is given for the number of edges that a robot has to traverse to visit all the vertices in the graph. Given that the upper bound does not consider any other perceived vertex beyond the one in which the robot currently is, it is like the robot operates under the *fixed graph scenario*, namely it learns of each vertex (and of the corresponding incident edge) adjacent to a vertex that it visits.

In this thesis, we significantly complement the analysis of Tovey and Koenig [2003] explicitly embedding in the analysis the sensor range r, and considering exploration strategies based on information gain and on combination of information gain with distance. We adopt a termination criterion based on the goal percentage p of the number of vertices to explore with respect to the total number of vertices in the environment.

To perform the analysis we first introduce a model that extends the fixed graph scenario, which allows the robot, once it reaches a vertex, to learn all the vertices (and the corresponding incident edges) within a certain distance r from the current position. The need for this extension is basically due to the nature of the information gain. To approach the information gain on a graph model, we introduce the concept of "gain" in terms of perceived vertices thanks to a more powerful sensor compared to the one employed in a fixed graph scenario.

Our contributions follow two directions: a worst-case analysis on this extended model and an average-case analysis in the fixed graph scenario, which, to the best of our knowledge, has never appeared in the literature.

We start proving some universal bounds on the number of candidate location selections, and on the traveled distance for very large values of the sensor range r. Then, we revisit the worst-case bounds found by Tovey and Koenig [2003] and Koenig et al. [2001], according to our setting, for an exploration strategy that selects the closest candidate location from the current position of the robot. We analyze how those worst-case bounds changes increasing the value of the sensor range r, pointing out the significance of the model extension. Furthermore, we provide worst-case bounds for an exploration strategy, that selects among the candidate locations, maximizing the information gain, providing, also, a universal worst-case upper bound. We consider one more strategy which combines distance and information gain, selecting the candidate location closest to the current position of the robot, and breaking ties maximizing the information gain. We complete the worst-case analysis performing some simulations of the three exploration strategies on random generated graphs.

About the average-case analysis, we consider the gain, in terms of traveled distance, provided by a tie breaker function that maximizes the information gain, over the use of a random tie breaker for a distance-based criterion, in a fixed graph scenario. We tackle this problem for certain classes of graphs that model indoor environments. In particular, we consider two classes. The first one is a tree of corridors and the second one is a simple corridor loop; in both cases some rooms are attached to the corridor(s).

In those classes, the information gain leads the exploration favoring room entrances, which have information gain higher than corridor vertices.

This thesis is structured as follows. Chapter 2 presents a survey of works about exploration, divided into those that deal with graph exploration, and those that analyzes the geometrical features of the environment. Moreover we also presents some works that experimentally evaluate information-gain based strategies. Chapter 3 formulates the problem we study and presents the model we use in this thesis. Chapter 4 shows worst-case upper and lower bounds on the number of edge traversals for three different exploration strategies: one based on the distance criterion, one on the information gain criterion, and one that combines both criteria. Chapter 5 analyzes the performance in the average case for two strategies: the first one based on the distance criterion and the second one that combines distance and information gain criteria. Finally, Chapter 6 concludes the thesis.

Chapter 2

State of the Art

Exploration strategies are usually defined and evaluated following two rather different approaches. On the one hand, they are defined in practical contexts of real (or realistically simulated) robots and are empirically assessed by testing them in some environments. Examples of this approach are reported by Amigoni [2008] and Julia et al. [2012]. On the other hand, exploration strategies are defined in theoretical settings (e.g., exploration of graphs) and are assessed using theoretical tools like worst-case bounds [Tovey and Koenig, 2003] and competitive ratio [Quattrini Li et al., 2012], which is the ratio between the cost of the solution found by an online algorithm and that of the optimal solution found by an offline algorithm.

2.1 Practical Results

On the practical side, several evaluation functions have been developed to decide, at each step, where to move next. The trivial distance criterion (namely, to choose always the nearest unscanned location) is often combined to other sophisticated policies in order to achieve the best performance (in terms of time or traveled distance), since, in robotic navigation, each useless displacement amounts to a waste of resources. Gonzáles-Baños and Latombe [2002] exploits the concept of information gain, referring to the expected amount and quality of the information that will be revealed at each new view. In the experiments this criterion seems to perform better than the plain distance criterion. The same concept has been used by Amigoni and Caglioti [2010] to develop A-C, an information-based exploration strategy for mapping, which has been shown to perform very well on real robots. Nevertheless, the evaluation of these strategies is only empirical and does not guarantee any performance bound.

Another interesting variation of the exploration problem is analyzed by Smirnov et al. [1996]. The authors consider a search problem on a partial known or un-

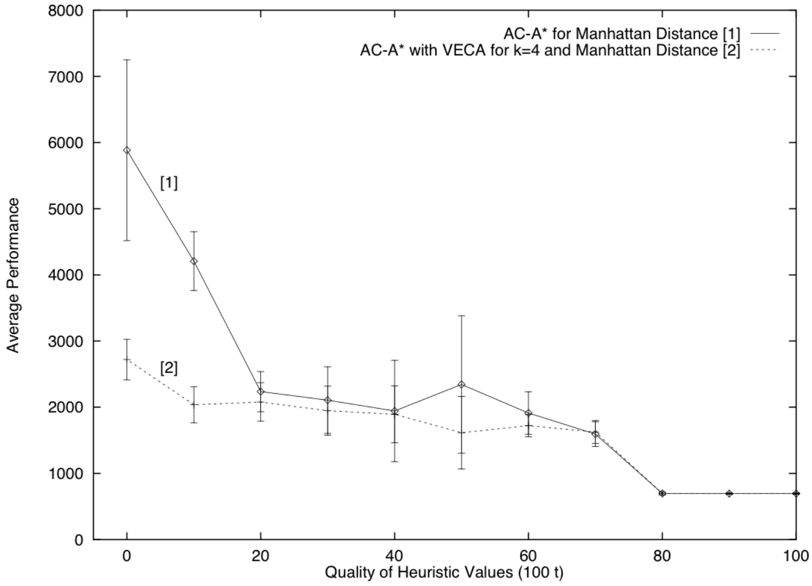

Figure 2.1: Simulations with AC-A and the framework VECA for goal-directed exploration [Smirnov et al., 1996].*

known graph, considering two strategies: the first one is pure exploration strategy, which does not know anything about the target position; the second, called Agent Centered A* (AC-A*), uses the target position as heuristic (with an approach very similar to A*). They show that both approaches have disadvantages: the first one does not utilize available knowledge to cut down the search effort, and the second one relies too much on the knowledge, even if it is misleading. Therefore, they developed a framework for goal-directed exploration, called VECA, that combines the advantages of both approaches by automatically switching from exploitation to exploration on parts of the state space where exploitation does not perform well. VECA provides better performance guarantees than previously studied heuristic-driven exploitation algorithms, and experimental evidence suggests that this guarantee does not deteriorate its average-case performance.

In [Quattrini Li et al., 2012] authors deal with a more powerful sensor range. In this work the environment is modeled as a grid and the goal is to map an initially unknown environment given a starting position. The best path is computed offline using A* over all the possible exploration states. The sensors considered are of two types: a laser range finder sensor that perceives the state of any cell whose center can be connected to the position of the robot with a straight line segment of maximum length r and crossing only free cells (without passing between occupied cells that share a vertex); a (less realistic) footprint sensor that perceives the state (free

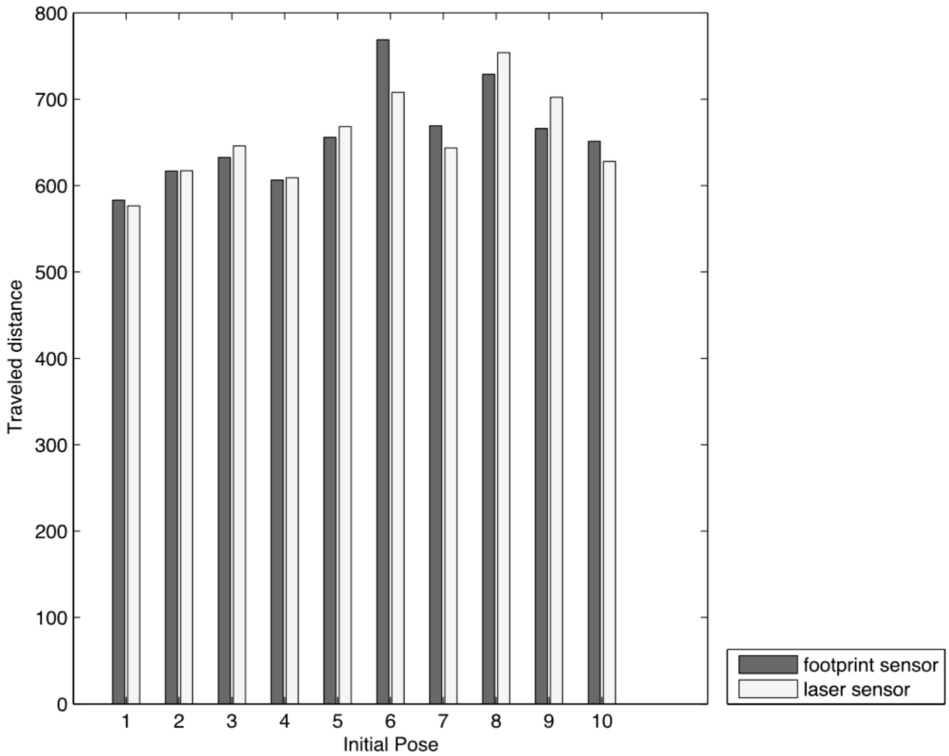

Figure 2.2: The traveled distance using a realistic laser sensor and an unrealistic footprint sensor [Quattrini Li et al., 2012].

or occupied) of any cell whose center lies within the circle centered in the robot with radius r. Some experimental results, reported in Figure 2.2, show that using a footprint sensor obtains similar results to those obtained by using real sensor.

2.2 Theoretical Results

On the theoretical side, works can be further classified according to the abstraction level at which environments are represented. The computational geometry community considers geometrically-represented environments (e.g., rectilinear polygonal environments in the work of Deng et al. [1998]). Gabriely and Rimon [2010] determined some classes of environments based on their geometrical features and, for which the competitive ratio of any exploration strategy is found. In particular three different complexity have been found: linear, quadratic, and exponential. The bounds take into account realistic and physical settings, such as the robot size. Examples of geometrical environments are given in Figure 2.3 and Figure 2.4.

Surveys such as those of Ghosh and Klein [2010] and of Isler [2001] report sev-

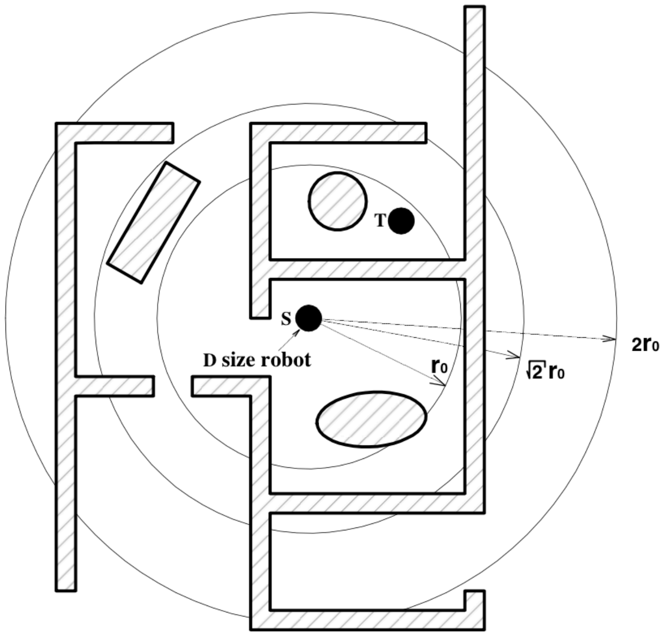

Figure 2.3: Simple geometrical enviroment [Gabriely and Rimon, 2010].

eral ad hoc strategies used to explore real polygonal environments with or without polygonal obstacles.

The theoretical computer science community, instead, usually considers more abstract graph-based representation of environments that disregards their geometry to focus on their topological and combinatorial aspects. In this field, the exploration task is usually formulated as follow: a robot has to explore an initially unknown graph $G = (V, E)$, where the vertices V correspond to the locations where it can move and the edges E represent the direct connections between these locations. During the exploration process, the robot uniquely learns of each vertex (and of the corresponding incident edge) adjacent to a vertex that it visits. This scenario is called *fixed graph scenario*.

2.2.1 Graph Edges Exploration

There are typically two variants of the exploration task studied with a graph-based representation of the environment. One is that of visiting all the edges. Deng and Papadimitriou [1999] show that in Eulerian graphs (namely those that contain cycles which use a graph edge only once, also called Eulerian cycle) the minimum competitive ratio for any exploring algorithm is 2. Furthermore, they show that, for non-Eulerian graphs, this ratio is unbounded when the deficiency of the graph (i.e.,

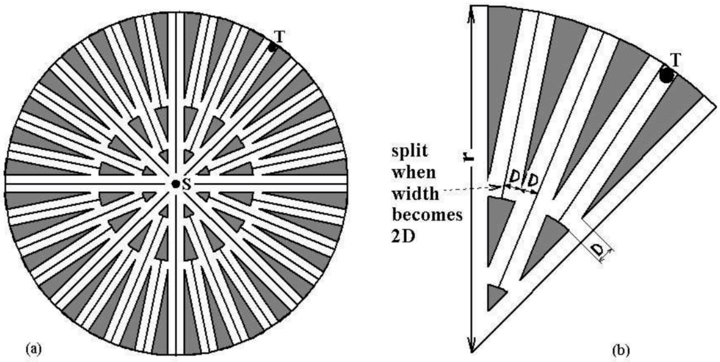

Figure 2.4: Complex geometrical enviroment. Here D is the robot size [Gabriely and Rimon, 2010].

the number of edges to be added to make the graph Eulerian) is unbounded. The authors propose an algorithm that explores a graph with deficiency d using $2^{O(d \log d)}$ edge traversals. Panaite and Pelc [2000] investigate the impact of the amount of *a priori* knowledge of the graph on the exploration performance. It is theoretically shown that the best exploration algorithm is 2-competitive (considering the number of edge traversals to explore the graph) when no *a priori* knowledge is available about the graph. In particular they consider two different kinds of knowledge: the topology of the graph and the sense of the direction. Moreover, they restrict their attention to trees exploration and they prove that it is possible, for the universe of trees, to lower the minimum feasible competitive ratio to $\frac{4}{3}$.

The work by Fraigniaud et al. [2005] considers a slightly different problem. Instead of considering the exploration time complexity, for example measured as the number of edge traversals required to explore the whole graph, the authors theoretically look at the minimum memory size the robot requires to explore a graph. They prove that the worst-case space complexity is $\Theta(D \log d)$ to explore all graphs of diameter D and maximum degree d. (The diameter of a graph is the length of the "longest shortest path" between any two graph vertices; the maximum degree refers to the maximum number of incident edges to a graph vertex.) Furthermore they show that, for any K-state robot (namely a finite state machine with K states) and any $d \geq 3$ there exists a planar graph of maximum degree d with at most $K + 1$ vertices that the robot cannot explore, or, equivalently, a robot that explores any planar graph with n vertices requires at least $\lceil log\, n \rceil$ memory bits. They leave open the problem to decide whether this latter bound is tight, or if, for any K-state robot, there exists a graph of size $o(K)$ that this kind of robot cannot explore.

2.2.2 Graph Vertices Exploration

The second variant of the graph-based exploration task is that of visiting all the vertices of a graph. Usually, the optimal solution is calculated as a minimal-length tour that visits all the vertices of an undirected graph and returns to the starting one. One of the first algorithms devised for such online Traveling Salesman Problem is called Nearest Neighbor, which always chooses to move to the closest unvisited vertex. Rosenkrantz et al. [1977] prove that such algorithm is at most $\log(|V|)$-competitive (where $|V|$ is the number of vertices in the graph), meaning that the ratio between the length of the solution it produces and that of the optimal solution is at most $\log(|V|)$.

Kalyanasundaram and Pruhs [1994] present a variant of the Depth-First Search algorithm, called ShortCut, which finds an exploration tour visiting all the vertices of an edge-weighted connected graph. The searcher operates under the fixed graph scenario and moves (traveling on known edges) to unvisited boundary vertices adjacent to at least a visited vertex. The main difference between ShortCut and DFS is the concept of blocking. At any point in time during the exploration of the graph, a boundary edge $e = (u, v)$ is said to be blocked, if there is another boundary edge $e' = (u', v')$ with u' explored and v' unexplored which is shorter than e and for which the length of any shortest known path from u to v' is at most $(1 + \delta)|e|$. Notice that δ is the key parameter of the strategy and decides when ShortCut behavior diverges from DFS. ShortCut is proved to be (at most) 18-competitive on planar graphs. In the last part of the paper, the authors explain how to adapt ShortCut to develop a competitive algorithm for visual TSP, requiring that the planar graph, used by the searcher, contains a minimum spanning tree as a subgraph. The question whether ShortCut has constant competitive ratio in general has remained open in this paper.

The case of exploration of cycles under the fixed graph scenario is analyzed by two works. The work of Miyazaki et al. [2009] shows an algorithm that builds a tour to explore the vertices of a graph and proves that its competitive ratio is $1+\sqrt{3}$ on simple cycles. In addition, they show that for unweighted graphs a standard Depth-First Search is 2-competitive. Asahiro et al. [2010] prove that the weighted Nearest Neighbor algorithm, which chooses the next vertex to visit according to a weighted distance cost, achieves a competitive ratio of 1.5 on cycles. Moreover, authors show that no exploration strategy can have a competitive ratio less than 1.25 on cycles.

The work of Megow et al. [2012] considers again the fixed graph scenario and answers the open questions left by Kalyanasundaram and Pruhs [1994]. Their main result is an involved lower bound construction that shows how the competitive ratio of ShortCut is not constant in general.

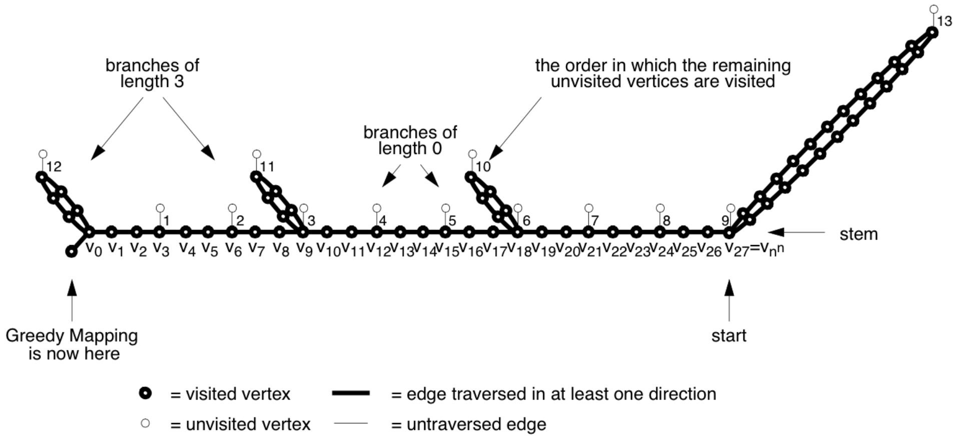

Figure 2.5: The worst case graph for Greedy Mapping with parameter $n = 3$ [Koenig et al., 2001].

Anyway, they shows that ShortCut algorithm (here called Blocking$_\delta$ algorithm, to underline the importance of the concept of blocking and the δ parameter) has a constant competitive ratio for a class of graphs wider than planar graphs. More precisely, Blocking$_\delta$ is $2(2+\delta)(1+2/\delta)(1+2g)$-competitive for graphs of genus at most g (a graph has genus at most g if it can be drawn without crossing itself on a sphere with g handles; a planar graph has genus 0). Another contribution given in this paper is the definition of a new algorithm called hierarchical depth first search (hDFS), which is $2k$-competitive on graphs with at most k distinct weights.

An important remark concerns the difference between the traditional TSP problem and an exploration problem. While in the former the graph is known from the beginning, the latter has to deal with the partial knowledge of the graph, which is obtained incrementally during the exploration. Moreover all the above results consider exploration *tours* that require the searcher to return to the starting vertex. This constraint is not usually imposed in practical robotic exploration, where the goal is to find an exploration *path*. As recognized by Kalyanasundaram and Pruhs [1994], the combinatorics of the two problems are significantly different.

One of the most notable works that deal with exploration paths is that of Tovey and Koenig [2003], which shows that a robot (basically operating in the fixed graph scenario), adopting an exploration strategy that considers the distance as criterion, namely that selects the closest unvisited vertex, has an upper bound of $O(|V| \log |V|)$ edge traversals. The same bounds hold also for other variants of the distance-based exploration strategy, choosing the closest unscanned vertex and the closest informative vertex. The first variant considers a more powerful sensor which allows the robot to perceive (scan) vertices beyond the threshold imposed

n	travel distance	$\|V\|$	$\frac{\text{travel distance}}{\|V\|}$
3	207	80	2.587500
4	2279	778	2.929306
5	31253	9612	3.251457
6	515085	144014	3.576631
7	9928271	2542528	3.904882
8	219130987	51744018	4.234905
9	5448100629	1193201300	4.565953
10	150617283953	30753086422	4.897631
...	...	...	...

Figure 2.6: The worst-case analysis for Greedy Mapping varing the parameter n [Koenig et al., 2001].

by the fixed graph scenario. The second variant considers the closest scanned vertex which has some unscanned vertices as neighbors. In both cases, the algorithm terminates when all the vertices have been scanned. This paper is the conclusion of a series of related works (Koenig [1998], Koenig et al. [2001]), where the main result achieved is the proof of the non-optimality of the Nearest Neighbor algorithm (here called Greedy Mapping). In particular, authors showed that, in the worst case, the complexity of the Greedy Mapping is $\Omega(\frac{\log |V|}{\log \log |V|}|V|)$. The proof is based on the construction of a sophisticated class of graphs, characterized by a parameter n (an example of such graph is given by Figure 2.5). It has been shown that, for those worst-case graphs, $|V|$ is $\Theta(n^n)$, while the number of edge traversals is, in the worst case, $\Omega(n^{n+1})$. This proves, after some math, the lower bound. In Figure 2.6, the results obtained computing the number of vertices and the number of edge traversals for some worst-case graphs are reported.

Note that, as all the works presented in this section consider the fixed graph scenario, we have that the timing of the perception is discrete (if we intend perception as referring to the integration of the data coming from sensors in the current map). Also, the sensor range is basically ϵ, where ϵ is a small value close to 0, as just the incident edges of the current vertex can be perceived. Note that just the work of Tovey and Koenig [2003] considers a variant in which a more powerful sensor is available. Nevertheless, the provided theoretical results are independent of r and thus as the results obtained with a sensor range ϵ.

2.2.3 Summary of Graph Exploration

Summarizing, we classify the main theoretical works described above, according to the following dimensions and report them in Table 2.1:

Paper	Goal	Solution	Optimality criterion	Graph
[Deng and Papadimitriou, 1999]	E	P	# edge traversals	directed, strongly connected
[Panaite and Pelc, 2000]	E	P	# edge traversals	undirected, connected
[Fraigniaud et al., 2005]	E	T	memory size	anonymous, undirected
[Rosenkrantz et al., 1977]	V	T	traveled distance	complete, weighted, undirected
[Kalyanasundaram and Pruhs, 1994]	V	T	traveled distance	weighted, planar
[Miyazaki et al., 2009]	V	T	traveled distance	trees and cycles
[Asahiro et al., 2010]	V	T	traveled distance	weighted undirected graph
[Megow et al., 2012]	V	T	traveled distance	undirected, connected, weighted
[Tovey and Koenig, 2003]	V	P	# edge traversals	undirected, connected, unweighted

Table 2.1: Classification of representative papers from theoretical computer science on exploration according to several dimensions.

- Goal: visit all edges (E) or all vertices (V).
- Solution: the solution can be a closed tour (T) or a path (P).
- Optimality criterion: the criterion that is optimized can be the number of visited vertices (steps) and the number of edge traversals (distance, unweighted or weighted).
- Type of graph: for example, general graphs or planar graphs.

Some other variants of the graph exploration problem include the non-ability of the robot to uniquely recognize already visited vertices [Rekleitis et al., 1999], the constrained exploration of a graph, where a robot is tethered or must return from time to time to a fixed point [Awerbuch et al., 1999, Albers et al., 2002, Duncan et al., 2006], the exploration of directed graphs [Forster and Wattenhofer, 2012], and the use of multiple explorers [Higashikawa et al., 2014, Das et al., 2007].

Chapter 3

Exploration Process Model

In this chapter we describe the model of the exploration process that we will adopt in this thesis.

The environment is represented by a graph $G = (V, E)$, where the vertices V correspond to the locations where an autonomous mobile robot can move and the edges E represent the direct connections between these locations. The graph is assumed to be undirected, connected, and finite. For sake of simplicity, we assume that edges have unitary costs (as in the work of Tovey and Koenig [2003]). Generalizing our results to weighted graphs could be an interesting future work.

The robot starts exploring in a vertex $v_0 \in V$ at a time step 0 having no *a priori* knowledge about the graph G. The robot is equipped with a sensor with a finite range $r \in \mathbb{R}_{>0}$ (also called perception radius in the rest of the thesis) that perceives all vertices within the range r. More formally, a robot in v_i at time step i, perceives the vertices in $P_i = \{v' \in V \mid d(v_i, v') \leq r\}$ and updates its knowledge about already perceived (known) vertices as $V_i = V_{i-1} \cup P_i$, where V_{i-1} is the set of known vertices at the previous step $i-1$ and $d(v_i, v')$ is a function that computes the geodesic distance between the two vertices v_i and v' in G. Note that when $i = 0$, then $V_{i-1} = \emptyset$.

Since we consider a discrete model, although r can be any real number, we will refer to r to denote $\lfloor r \rfloor$, with a slight abuse of notation.

The perception model allows the robot to acquire knowledge about the incident edges of vertices $v' \in P_i$ and to recognize whether there is an edge between two known vertices $v', v'' \in V_i$. At each time step i, we also have the set of partially perceived vertices on the *frontier*, namely the candidate vertices, $F_i = \{v' \in V \mid v' \notin V_i \wedge (\exists (v'', v') \in E \mid v'' \in V_i)\}$, (similarly to the model of Tovey and Koenig [2003]). Note that if $r = \epsilon$ (where ϵ is a small constant that tends to 0 and allows to perceive just the vertex where the robot finds itself and the incident edges), then, vertices are perceived only when physically visited by the robot, as in the fixed graph scenario. Although the perception of *all* vertices at distance up to r from the robot current vertex could be unrealistic in some scenarios (due to the presence of obstacles), this footprint model leads to interesting theoretical results that are in accordance with many results obtained experimenting with real or simulated exploring robots (as in the work of Quattrini Li et al. [2012]). Moreover, the results are valid for search problems where the perception radius is not a visual sensor (e.g., radiation detection with mobile robots).

We assume that the perception of the robot is discrete: the robot perceives the surrounding environment and updates V_i to V_{i+1} only when in the next position v_{i+1} and not continuously while moving (time-discrete perception is often assumed by online exploration algorithms, see [Amigoni et al., 2013]).

At each time step i, the robot always chooses to move to one of the candidate vertices F_i. Although it would be possible to define a motion model for which the robot can move to any vertices $v \in V_i \cup F_i$, we impose to move on frontier vertices, as usually done with real exploring robots. Since we are interested in the theoretical analysis of the online exploration strategies, we assume that the perceptions and the movements of the robot are error-free (i.e., deterministic). As a consequence, the robot perfectly knows its position in the environment.

In summary, the robot operates according to the following steps:

1. starting from an initial vertex v_0; at a generic time step i, while being in v_i,
2. it perceives the surrounding environment generating P_i;
3. it integrates the perceived data within V_i;
4. it reaches a vertex in F_i, chosen according to an exploration strategy $\mathscr{S}$, and starts again from 1.

This process continues until a percentage $p \in (0, 1]$ of the vertices of G are perceived by the robot, namely until $\frac{|V_i|}{|V|} \geq p$. Note that the exploration terminates at some finite time step k because the robot chooses vertices that provide some new information about the graph (i.e., it is $|V_{i+1}| > |V_i|$) and the graph is finite. So, in the end, the robot follows a sequence of vertices $\mathscr{P} = \langle v_0, v_1, \cdots, v_k \rangle$, called *exploration path*, where $v_{i+1} \in F_i$, with $0 \leq i < k$ (v_0 is the starting vertex).

We consider exploration strategies that evaluate a candidate vertex $v \in F_i$ from the current position v_i adopting the following criteria:

- $d_i(v_i, v)$ is the geodesic distance between v_i and v in $G'_i = (V_i \cup \{v\}, E'_i)$, which is the graph G_i induced by V_i on G, augmented with v and with the edges (in E) between v and vertices in V_i,

- $g(v, V_i)$ is the expected information gain at v, and is equal to the number of vertices the robot perceives in v minus those already known. More formally, at time step i, given a frontier vertex v, $g(v, V_i) = |P(v) \setminus V_i|$. Being $v \notin V_i$, the function $g()$ could be estimated from datasets of environments, but here we assume it as granted. In the particular case of $r = \epsilon$ ($\epsilon \to 0$) we define $g(v, V_i)$ as the number of edges incident to v that are connected to vertices not in the current map V_i at time step i.

We consider three exploration strategies:

- $\mathscr{S}_d$, which selects locations by simply minimizing the distance $d()$ (as for example in [Tovey and Koenig, 2003]),
- $\mathscr{S}_g$, which chooses candidate locations maximizing the information gain $g()$ (as, e.g., in [Amigoni, 2008]),
- $\mathscr{S}_{dg}$, based on $\mathscr{S}_d$ but breaking ties favoring vertices with larger information gain $g()$ (thus providing a more informed version of $\mathscr{S}_d$).

In all the three cases, further ties are broken randomly with uniform probability.

The problem we address is the following. Given a sensor range $r \in \mathbb{R}_{>0}$, a percentage $p \in (0, 1]$ of the environment to map, and an exploration strategy $\mathscr{S} \in \{\mathscr{S}_d, \mathscr{S}_g, \mathscr{S}_{dg}\}$, can we determine some performance bounds on the exploration path $\mathscr{P}$ in terms of number of edges traversed by the robot in any undirected, connected, and finite graph G? Moreover, given a certain class of indoor environments, can we estimate the average number of edges traversed by the robot?

Chapter 4

Worst Case Analysis

Here we provide a comparison of the three exploration strategies by presenting some bounds on their worst-case performance, measured as traveled distance (number of edge traversals). Bounds will be given according to the total number of vertices $|V|$ in the graph, and the perception radius r.

4.1 Universal Bounds

To study the upper bounds on the traveled distance of the exploration strategies, let us first derive an upper bound on the number of frontiers selected as destination locations (namely, on k, which is the cardinality of the exploration path $\mathscr{P}$ minus 1) according to the number of vertices to explore $|V|$ and the sensor range r, independently of the exploration strategy.

Theorem 1. *Given a robot sensor range $r \in \mathbb{N}$, the maximum number of frontier selections in the exploration sequence $\mathscr{P}$ is*

$$\bar{k} = \begin{cases} 2\dfrac{|V|-1}{r+1} - 1 & \textit{if } r \textit{ is odd} \\ \dfrac{2|V|}{r+2} - 1 & \textit{if } r \textit{ is even} \end{cases}$$

on any finite undirected connected graph $G = (V, E)$, where the weight of each edge is 1.

Proof. The proof develops on bounding the number of selected frontiers in the exploration sequence $\mathscr{P}$ on any spanning tree of the connected graph G.

First let us show that, given an exploration path on any connected graph, the same sequence can be obtained on one of its spanning trees. Let us consider a generic exploration path $\mathscr{P} = \langle v_0, v_1, \cdots, v_k \rangle$ on a given connected graph $G =$

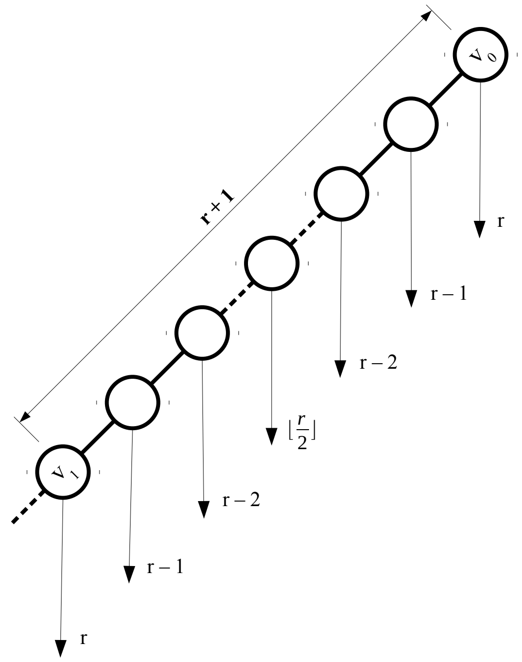

Figure 4.1: The tree exploration after the perceptions in v_0 and v_1. The arrows mean the branches for each vertex, while, the labels mean their explored depth.

(V, E). Let us build a spanning tree $T = (V, E')$ of G as follows. We choose the vertex v_0 as the root. For each step i $(0 \leq i \leq k)$ of the exploration path, the set of edges E' is built as follows: we add the edges belonging to the graph induced by P_i/V_{i-1} on G, such that the induced graph is loop-free (the edges removal can be arbitrary). Moreover, except for $i = 0$, we add (v, v_i) to E', where $v \in V_{i-1}$ which is the edge that connects the frontier v_i to a vertex v in the current known graph V_{i-1}. That edge belongs to the shortest path between v_{i-1} to v_i found in the current known graph V_{i-1}. Thus, by construction, the graph T is a tree.

Now let us prove by induction the following statement: at each step i, $v_{i+1} \in F_i$ both on G and T, namely it is possible to have the same exploration path $\mathscr{P}$ on G and on T. At step 0 (base case), $V_0 = P_0$, hence $v_1 \in F_0$ both in G and T, as the robot perceives the same vertices and by construction of T. Also, by construction, v_1 is reachable in T. At a generic step $j < i$ (the inductive step), $v_{j+1} \in F_j$ and v_{j+1} is reachable in T.
We say that V_i *coincides* on both G and T for any exploration strategy $\mathscr{S}$ if, at the step i, given the set of the perceived vertices V_i^G exploring G, and the set of the perceived vertices V_i^T exploring T, we have $V_i^T = V_i^G = V_i$. Now, let us show that at step i, V_i coincides on both G and T, if the robot is in v_i and V_{i-1} coincides in both G and T. Choosing $j = i-1$, the statement holds by the inductive hypothesis

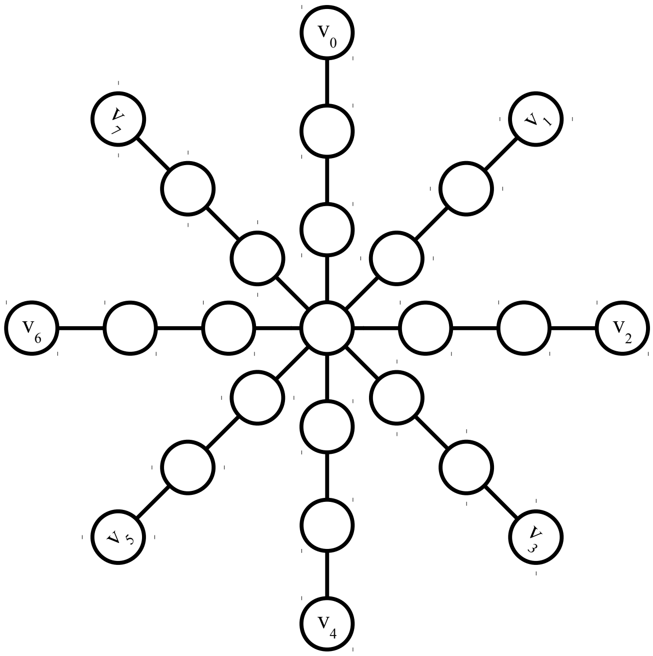

Figure 4.2: The worst case for the ratio $|V|/k$ with $|V| = 25$, $r = 5$ and $\bar{k} = 7$.

(the robot is in v_{i-1}). By construction, $\exists v \in V_i : (v, v_{i+1}) \in E'$. Thus, at step i, $v_i \in F_i$ on both G and T, therefore, on both T and G, there exists a path from v_{i-1} to v_i. This leads V_i to coincide on G and T, and so, when in v_i, $v_{i+1} \in F_i$. Hence, the statement on the fact that the same exploration path $\mathscr{P}$ can be obtained on G and T holds.

Now we bound the number of selected frontiers in $\mathscr{P}$ on any tree built as shown above, by showing the worst-case graph in terms of number of selected frontier vertices k. Let us recall that the robot can discover vertices at distance less or equal than r from v_0 (in general from any vertex v_i). Thus, frontier vertices are at least at distance $r + 1$ from a vertex v_i. Let us call *line* of v toward w all the vertices on the shortest path from v to w (which is unique because of the tree structure). Note that each perception P_i, on a frontier vertex v_i chosen from v_{i-1}, does not add, to V_{i-1}, vertices belonging to other lines of v_{i-1} towards other frontier vertices, since they are all the distance greater than r. Let us focus just on the subtree rooted in v_1 (reasonings on other subtrees are basically the same). To find this bound, we look at where to attach vertices in the graph in order to minimize the number of perceived vertices by the robot at a time step i. There are basically just two possibilities where to attach vertices: either to attach them directly to lines that start from v_0 or to subtree rooted in some other frontier vertices v_i, $i > 0$. Once the robot is in v_1, it can choose other frontier vertices discovered at time step 1 (in the subtree rooted in v_1). Those frontier vertices are at distance $r+1$ from v_1. Otherwise, it can go back to a vertex v between v_0 and v_1, and then go to another frontier vertex reachable from v. Given a line starting from v, the depth of the tree (rooted in v_0) the robot can reach depends on the distance of v from v_0 and v_1. The worst case in terms of k (namely the minimum number of vertices that should be added to have another frontier) happens when v is at distance $\lceil \frac{r}{2} \rceil$ from v_0 or v_1. In that case, the number of vertices required to reach a new frontier vertex is $\lfloor \frac{r}{2} \rfloor + 1$. This is clear in Figure 4.1. Note that having more than one line starting from v_0 is not the worst case in terms of k because $r+1$ vertices are necessary to have a new frontier vertex. Thus, for finding the worst case for k it is better to attach vertices to v so that lines starting from v to frontier vertices are composed of $\lfloor \frac{r}{2} \rfloor + 1$ vertices. This scenario can be seen as a star graph, where the length of each line that starts from v to a leaf is $\lfloor \frac{r}{2} \rfloor + 1 = \lceil \frac{r+1}{2} \rceil$. The number of such lines is the maximum number of frontier k in the exploration path $\mathscr{P}$. An example of explorations on a star graph is given in Figure 4.2.

It is easy to check that, in case of odd r, $|V| = 1 + \frac{r+1}{2}k$, where k is both the number of leaves of the star and the cardinality of the exploration path. After some math $k = 2\frac{|V|-1}{r+1}$ and because v_0 is not counted in the frontier selection $\bar{k} = 2\frac{|V|-1}{r+1} - 1$. In case of even r, similar reasonings lead to $|V| = (k+1)\frac{r+2}{2}$ and thus $\bar{k} = 2\frac{|V|}{r+2} - 1$. □

Note that $\bar{k}$ can be a rational number if the star graph is not a perfect star (namely, one ray is shorter than the others). If we are interested in the exact length we may take $\lfloor \bar{k} \rfloor$.

It is easy to see that the upper bound in Theorem 1 is tight, as we are considering the worst-case graph and the possible perceptions of the robot for the number of frontiers of the exploration path. For sake of simplicity we will take $\bar{k} = 2\frac{|V|-1}{r+1} - 1$ as upper bound, namely the value of $\bar{k}$ for r odd, which is an upper bound for r even. Note also that when $r = \epsilon$ ($\epsilon \to 0$), then we have the trivial bound on the number of selected frontiers $\bar{k} = |V| - 1$, namely, the robot has to visit all the vertices.

Now, let us show bounds on the number of edge traversals during exploration. The following result allows us to restrict our analysis to small range r, because, if r is greater than a value that depends on the size of the graph, the worst-case upper bound on the number of edge traversals is linear for any exploration strategy.

Theorem 2. *Given a robot sensor $r \geq \lfloor \frac{|V|-1}{2} \rfloor$, for any exploration strategy $\mathscr{S}$, the upper bound on the number of edge traversals is*

$$UB = \frac{3}{2}(|V| - 1)$$

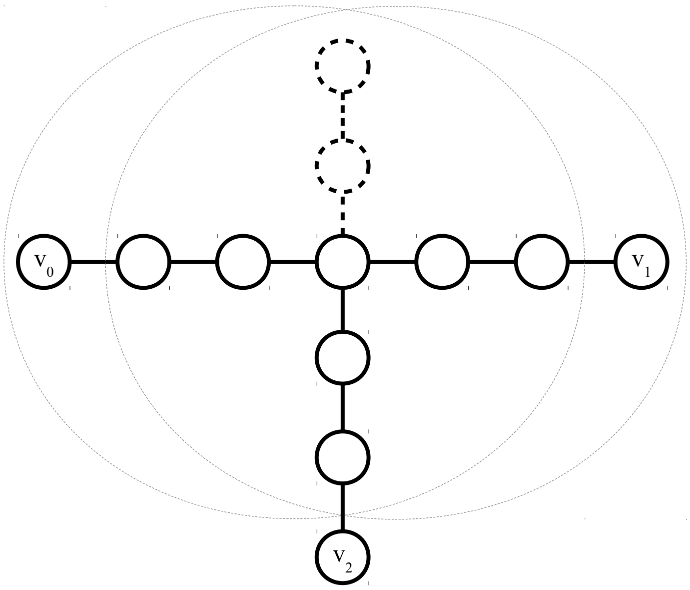

Figure 4.3: For $|V| = 12$ and $r = \frac{|V|}{2} - 1 = 5$ there are at most two frontier vertices v_1 and v_2.

on any finite undirected connected graph $G = (V, E)$, where the weight of each edge is equal to 1.

Proof. Let us start considering $r = \lfloor \frac{|V|-1}{2} \rfloor$. To prove the proposition, we incrementally build the worst-case graph, considering separately the case of even and odd $|V|$.

If the value of $|V|$ is even, $r = \frac{|V|}{2} - 1$. Thus, starting from an arbitrary v_0, the maximum number of frontier selections is 2 for any $|V| \geq 6$ because of Theorem 1 (as shown in Figure 4.3). Basically, we have two steps in the exploration path: from v_0 to a vertex v_1 and from v_1 to a vertex v_2. The distance between v_0 and v_1 is always of length $r + 1$. Thus, the first part of our worst-case graph is composed by V_0, which contains at least a line of $\frac{|V|}{2}$ vertices, plus v_1. If the remaining $\frac{|V|}{2} - 1$ vertices are attached to v_0 or v_1 there are no more frontier vertices, because they are within the range r of a perception performed from v_0 or v_1. If they are attached to a vertex on the line between v_0 and v_1, forming a new line of vertices, there could be a new frontier vertex v_2, which is not within the range r of a perception performed at v_0 and v_1. Let us show where the line of vertices containing v_2 should be attached to have the worst case. The worst case happens when it is attached at distance 1 from v_0, on the line that links v_0 and v_1. It is easy to check that this shape maximizes the distance $d(v_1, v_2)$, since $d(v_0, v_1)$ is necessarily fixed to $r+1$.

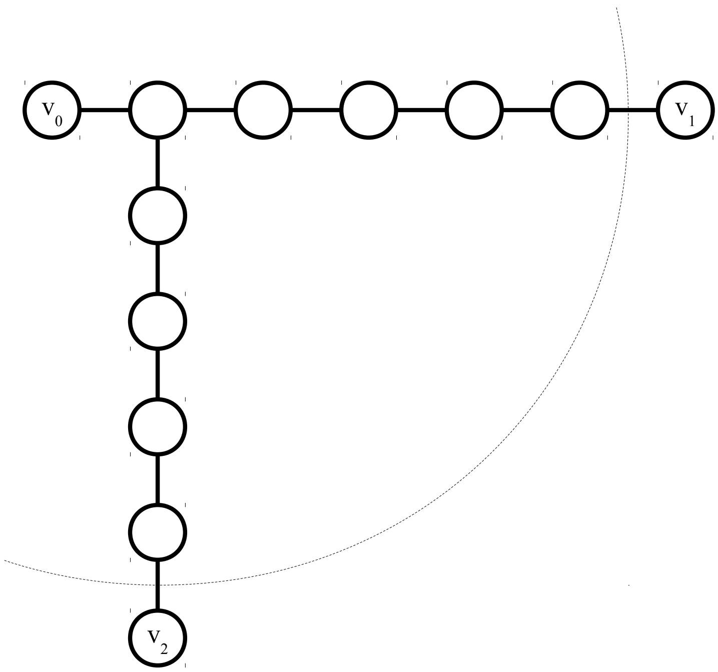

Figure 4.4: The worst case for $r = \frac{|V|}{2} - 1 = 5$ *and* $|V| = 12$.

Hence, $UB_{even} = d(v_0, v_1) + d(v_1, v_2) = (r+1) + 2r = \frac{|V|}{2} + 2(\frac{|V|}{2} - 1) = \frac{3}{2}|V| - 2$.

If $|V|$ is odd, there are still at most 2 frontier selections. Following the same reasoning, the number of vertices between v_0 and v_1 is $r = \frac{|V|-1}{2}$. Thus, the number of vertices we can use to compose our worst-case graph is $|V| - (r+2) = |V| - \frac{|V|-1}{2} - 2 = \frac{|V|-3}{2}$. If they are attached to v_0 or v_1, again, there are no more frontier vertices, but also if they are attached at distance 1 from v_0 or v_1. In case the value of $|V|$ is odd, the worst case happens when the new line, which contains v_2, is attached at distance 2 from v_0, on the line that links v_0 and v_1. Hence, $UB_{odd} = d(v_0, v_1) + d(v_1, v_2) = (r+1) + 2r - 1 = \frac{|V|-1}{2} + 2(\frac{|V|-1}{2}) = \frac{3}{2}|V| - \frac{3}{2}$. Note that we have the following relationship between the case of $|V|$ even and odd $UB_{even} \leq UB_{odd} = UB$. Also note that considering a greater r the upper bound found trivially holds, as, each increment of r implies an increment of $d(v_0, v_1)$ but an equal decrement of $d(v_1, v_2)$. After a certain radius the number of frontiers has lowered to 1 and the trivial upperbound bound becomes $|V| - 1$. If the sensro range r is greater than $|V| - 1$ the number of frontier selections is 0 as the number of edge traversals. □

4.2 Distance Criterion

Now, let us show how the upper bound on the number of edge traversals changes according to r (with $r \in \mathbb{N}$ and $r < \lfloor \frac{|V|-1}{2} \rfloor$, for Theorem 2), differently from $UB_{TK} = |V| + 2|V|\ln(|V|)$, the worst-case bound on the performance of $\mathscr{S}_d$ which has been provided by Tovey and Koenig [2003] and that is independent of r.

Theorem 3. *Given a robot sensor range $r \in \mathbb{N}_{>1}$, the worst-case upper bound on traveled distance for $\mathscr{S}_d$ is*

$$UB_{\mathscr{S}_d} = 2|V| \left(\ln \frac{2(|V|-2) + r(r-1)}{(r+1)^2} - \frac{r(r-1) + (|V|-2)}{(r+1)(|V|-2)} + 2 \right)$$

edge traversals, on any finite undirected connected graph $G = (V, E)$, where the weight of each edge is equal to 1

Proof. First, let us introduce a lemma (proved in Tovey and Koenig [2003]) that we are going to use to prove the theorem. Lemma 6.2 of Tovey and Koenig [2003] states the following.

> Define $S^t = \{v^i \in V | r^i \geq t\}$ for an orderly marking sequence $\{v^i, r^i, M^i\}$ on a given connected graph $G = (V, E)$. Then, it holds that $|S^t| \leq 2|V|/t$.

An orderly marking sequence is basically an exploration path, which includes the vertices the robot visits during the exploration. The symbols used in the above Lemma have the following correspondence with the symbols we use: $v^i = v_i$, $M^i = V_i$, and $r^i = d(v_{i-1}, v_i) - 1$, namely the radius of a circle centered on v_{i-1}, which is given by the distance between the current frontier v_i and the preceding frontier v_{i-1} minus 1 (because of the movement towards the frontier) in the exploration path. Note that, as r^i represents the radius within which, from v_{i-1}, all the vertices are in V_{i-1}. Intuitively, S^t represents all the vertices v_i that are at a distance of at least $t+1$ from the next vertex v_{i+1}. By construction, each pair of frontier vertices in the exploration path are at least at distance $r+1$.

Considering the worst case on the traveled distance, the exploration path has a number of selected frontiers equal to $\bar{k}$, according to Theorem 1. Let us define h as the number of different t-classes S^t, where S^t is defined as above. h is a positive integer with value at most $\bar{k}$.

To enumerate all the t-classes used in our exploration path, let us define a function $f()$ which orders them, starting from $f(1) =$ the smallest t-distance (which must be greater than or equal to r); until $f(h) =$ the biggest t-distance in the exploration path (which must be less than or equal to $|V| - 2$, because the maximum travelable distance between two different vertices is $|V|-1$). Let us also define $f(h+1)=|V|-1$, $f(0)=0$, $|S^{|V|-1}|=0$.

We can find the traveled distance (number of edge traversals) with the following formula (which exploits $\bar{k}$ and the radius r^i that should be summed to 1 to have the actual traveled distance):

$$\sum_{i=1}^{\bar{k}} 1+r^i=\bar{k}+\sum_{i=1}^{\bar{k}} r^i$$

(disregarding the order of the selected frontiers in the path)

$$=\bar{k}+\sum_{t=0}^{h} f(t)\cdot(|S^{f(t)}|-|S^{f(t+1)}|)$$

(by applying some math and given the fact that $|S^{|V|-1}|=0$)

$$=\bar{k}+\sum_{t=0}^{h-1}(f(t+1)-f(t))\cdot|S^{f(t+1)}|$$

(Lemma 6.2 of Tovey and Koenig [2003])

$$\leq \bar{k}+2|V|\sum_{t=0}^{h-1}\frac{f(t+1)-f(t)}{f(t+1)}. \tag{4.1}$$

We have to find the set of values of $f()$ that maximize the above sum. Because of the Theorem 1, in general, the h counterimages of $f()$ do not cover all the codomain $\{r,...,|V|-2\}$. It is easy to see that the worst case happens when all the missing counterimage values are between $r+h-1$ and $|V|-2$. Thus, when $f(1)=r$; $f(2)=r+1$; ...; $f(h-1)=\lfloor r\rfloor+h-1$; and $f(h)=|V|-2$. Hence (4.1) becomes:

$$=\bar{k}+2|V|\left(\sum_{t=0}^{h-2}\frac{f(t+1)-f(t)}{f(t+1)}+\frac{f(h)-f(h-1)}{f(h)}\right)$$

(using the considerations on the values of $f()$ and doing some math)

$$\leq \bar{k}+2|V|\left(\sum_{t=r+1}^{h+r-1}\frac{1}{t}+2-\frac{f(h-1)}{f(h)}\right)$$

(limit approximation for the sum and explicitly reporting the first and last value of $f()$)

$$\approx \bar{k}+2|V|\left(\ln\frac{h+r-1}{r+1}-\frac{h+r-1}{|V|-2}+2\right)$$

Now we have to find the value of h that maximizes the formula. Analyzing the first and second derivative with respect to $h \in \{1, \cdots, \bar{k}\}$, we find that

$$\frac{\partial}{\partial h}\left(\ln\frac{h+r-1}{r+1} - \frac{h+r-1}{|V|-2}\right) = \frac{|V|-h-r-1}{(|V|-2)(h+r-1)}$$

The only root is $h = |V| - r - 1$ which is a feasible maximum just for $r = 1$. For any $1 < r < \lfloor\frac{|V|-1}{2}\rfloor$ and a sufficiently large $|V|$, it is easy to check that the maximum value is for $h = \bar{k}$. Using $\bar{k}$ for r odd as upperbound, after some math, we have

$$2|V|\left(\ln\frac{2(|V|-2)+r(r-1)}{(r+1)^2} - \frac{r(r-1)+(|V|-2)}{(r+1)(|V|-2)} + 2\right) - \frac{2}{r+1}$$

which is slightly lower than $UB_{\mathscr{S}_d}$, thus:

$$\leq 2|V|\left(\ln\frac{2(|V|-2)+r(r-1)}{(r+1)^2} - \frac{r(r-1)+(|V|-2)}{(r+1)(|V|-2)} + 2\right)$$

□

Corollary 3.1. *Given a robot sensor range $\lceil\bar{r}\rceil$, where $\bar{r} \in \mathbb{R}_{>}1$ such that the following transcendental equation is satisfied*

$$\ln\frac{2(|V|-2)+\bar{r}(\bar{r}-1)}{(\bar{r}+1)^2} = \frac{\bar{r}(\bar{r}-1)+(|V|-2)}{(\bar{r}+1)(|V|-2)}$$

the worst-case upper bound on traveled distance for $\mathscr{S}_d$ is $UB_{\mathscr{S}_d} = 4|V|$ edge traversals, on any finite undirected connected graph $G = (V, E)$, where the weight of each edge is equal to 1.

Proof. The proof trivially derives from the result of the Theorem 3.

$$UB_{\mathscr{S}_d} = 2|V|\left(\ln\frac{2(|V|-2)+\bar{r}(\bar{r}-1)}{(\bar{r}+1)^2} - \frac{\bar{r}(\bar{r}-1)+(|V|-2)}{(\bar{r}+1)(|V|-2)} + 2\right)$$

Because of the hypotesis, we have that

$$\ln\frac{2(|V|-2)+\bar{r}(\bar{r}-1)}{(\bar{r}+1)^2} = \frac{\bar{r}(\bar{r}-1)+(|V|-2)}{(\bar{r}+1)(|V|-2)}$$

Thus, since the $UB_{\mathscr{S}_d}$ function is monotonic with respect the sensor range r, the upper bound, for $\lceil\bar{r}\rceil$ becomes

$$UB_{\mathscr{S}_d} = 4|V|$$

□

Number of vertices $\|V\|$	Sensor radius $\lceil\bar{r}\rceil$
$5 \cdot 10$	12
$5 \cdot 10^2$	70
$5 \cdot 10^3$	347
$5 \cdot 10^4$	1662
$5 \cdot 10^5$	7833
$5 \cdot 10^6$	36615

Table 4.1: Some values of $|V|$ and the corresponding $\bar{r}$, such that the worst-case upper bound on traveled distance for $\mathscr{S}_d$ is $UB_{\mathscr{S}_d} = 4n$ edge traversals

The strong implication of Theorem 3 is that linearity is achievable through perception radii less than $\lfloor\frac{|V|-1}{2}\rfloor$, as we can see in Table 4.1. The radius $\bar{r}$, such that the worst-case upper bound is linear, increases less than linearly with respect to the number of vertices to explore. Moreover, incrementing r from $\bar{r}$ to $\lfloor\frac{|V|-1}{2}\rfloor$ $\mathscr{S}_d$ gains just $\frac{|V|}{2} + 1$ edge traversals, as upper bound.

Note that, if we consider the bound independent of r that in our setting corresponds to $r = \epsilon$ ($\epsilon \to 0$)), we have a bound slightly different of that in Tovey and Koenig [2003]. However, by including in the limit approximation the first term $\frac{r}{r}$ and the last term $\frac{f(h)-f(h-1)}{f(h)}$ we considered apart from the sum $\sum_{t=0}^{h-2}\frac{f(t+1)-f(t)}{f(t+1)}$, we exactly obtain the same bound of Tovey and Koenig [2003], as it would be

$$\begin{aligned}\bar{k} + 2n\sum_{t=0}^{h-2}\frac{f(t+1)-f(t)}{f(t+1)} &\leq \bar{k} + 2n\sum_{t=r+1}^{h+r}\frac{1}{t}\\ &\approx \bar{k} + 2n\ln\frac{h+r}{r+1}\\ &\leq \bar{k} + 2n\ln\frac{\bar{k}+r}{r+1}\end{aligned}$$

With $r = \epsilon$ ($\epsilon \to 0$) and $\bar{k} = |V|$ the upper bound on the traveled distance is $|V| + 2|V|\ln|V|$. Considering instead $r = 1$ and substituting the maximum we found in the proof (namely, $h = |V| - r - 1$) we have

$$\begin{aligned}&\bar{k} + 2|V|\left(\ln\frac{h+r-1}{r+1} - \frac{h+r-1}{|V|-2} + 2\right)\\ &= \bar{k} + 2|V|\left(\ln\frac{|V|-2}{2} + 2\right)\\ &= |V| + 2|V|\left(\ln\frac{|V|-2}{2} + 2\right)\end{aligned}$$

Nevertheless for sake of simplicity we will refer to the upper bound of Theorem 3 as the general upper bound for any r.

The following theorem presents a lower bound for $\mathscr{S}_d$. Because of the complexity of the extended formula, in this case we give just the order of the bound, according to the perception radius r and the number of vertices $|V|$.

Theorem 4. *Given a robot sensor range $r \in \mathbb{N}_{>0}$, the worst-case lower bound on the traveled distance for $\mathscr{S}_d$ is*

$$LB_{\mathscr{S}_d} = \Omega\left(\frac{\log|V| - 2\log(r+1)}{\log\log|V|}|V|\right)$$

edge traversals, on any finite undirected connected graph $G = (V, E)$, where the weight of each edge is equal to 1.

Proof. The proof is structured in two parts. First we show how the worst-case graph is constructed, depending on the sensor range r. This construction is based on the worst-case graph in [Koenig, 1998]. Then, we show that, on that graph, the robot (re)traverses a certain part of the graph several times.

The worst-case graph consists of three main components. The first one, that we call *stem*, is a line graph, whose number of vertices is m^m, where $m \geq 3$ is a parameter that allows to obtain the robot behavior briefly described above. Let us call the vertices on the stem v_0, v_1, v_2, ..., v_{m^m}, where v_0 is the vertex where the robot starts the exploration. The second main component is a *loop of level* i, where $w + 1 \leq i \leq m$, and

$$w = \lceil \log_m(r+1) \rceil$$

which is a parameter that depends on the sensor range r and defines the way the loops are attached to the stem, with which loops share just one vertex. The number of loops of level i is m^{m-i}. Loops of different levels i have different length. Specifically, the number of vertices of loops of level $i = w + 1$ is $3m^w$. When $i > w + 1$, the number of vertices of each loop of level i is

$$\lceil (\frac{i-w}{2} + 1) \cdot m^w \rceil + \sum_{j=w+1}^{i-1} m^j.$$

The loops are attached to the vertices on the stem in the following way. We start to attach loops of level i in an incremental way to vertices on the stems, distanced by m^i: first all loops of level i are attached to the stem starting from v_{m^m}, next all loops of level $i + 1$ are attached to the stem starting from v_0, then loops of level $i + 2$ starting from v_{m^m}, and so on, until $i = m$. The third main component of the worst-case graph we are building is a number of lines attached either to the stem or the loops. The number of vertices of those lines is $r + 1$. Let us call the lines attached to the stem lines of level $i = w$. They are attached similarly to how

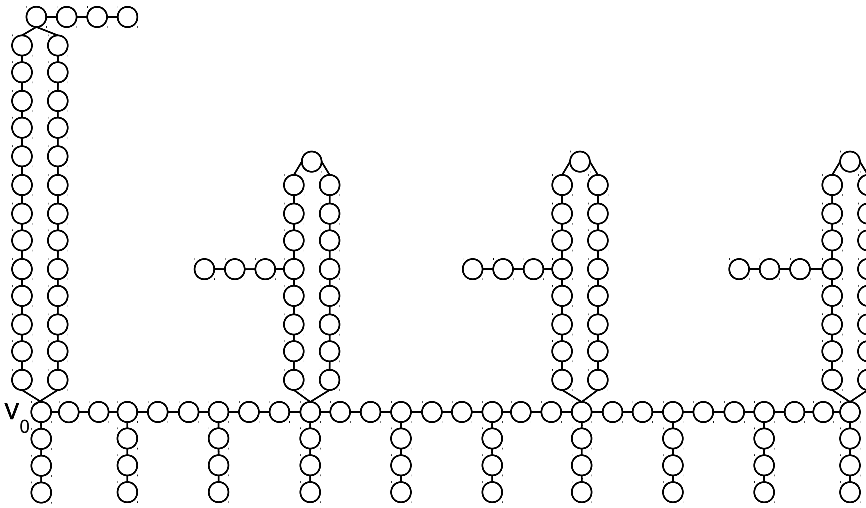

Figure 4.5: Worst-case graph for $\mathscr{S}_d$ with $m = 3$ and $r = 2$.

loops are attached to the stem, that is, starting from v_0 and distanced of m^w. Then, all the loops of level $i > w$ have a line attached to a vertex of the loop. When $i = w + 1$, the line is attached at distance $\lceil \frac{3}{2} m^w \rceil$ from the vertex on the stem. When $i > w + 1$, the lines are attached to a vertex in the loop in such a way that there are two paths of the same length from the vertex that the loop shares with the stem and the vertex that the loop shares with the line. Figure 4.5 shows an example of such graph.

Therefore, the number of vertices of the worst-case graph described above is (ignoring floors and ceils for simplicity):

$$|V| \approx m^m + \sum_{i=w+1}^{m} m^{m-i}(r + (i-w)m^w + 2\sum_{j=w}^{i-1} m^j) + \\ + 3m^{m-1} + (m^{m-w} + 1)(r + 1)$$

$$= m^m + r\frac{m^m - m^w}{m^w(m-1)} + \frac{m^{m+1} - m^{w+2} + wm^{w+1} - wm^w}{(m-1)^2} + \\ + 2\frac{m^m(m-w)}{m-1} - 2\frac{m^m - m^w}{(m-1)^2} + 3m^{m-1} + (m^{m-w} + 1)(r + 1)$$

Notice that $|V|$ is $\theta(m^m)$.

The robot, which starts in v_0 and employs $\mathscr{S}_d$ as exploration strategy, explores the graph described above as follows. First, the robot explores all the vertices of the stem and of the loops of all levels, ending up in v_{m^m}. Let us prove the reason why in this first traversal of the stem the robot does not explore the lines attached to the stem or to the loops. To get the distance of the frontiers on the lines, let us remind that, as shown in the proof of the maximum length of the sequence (see Theorem 1), ignoring all the single lines at the first traversal, the robot perceives at least a number of vertices of the lines $\lfloor \frac{r}{2} \rfloor$ and at most r. Considering that the robot starts exploring a loop of level $i = w + 1$ from its vertex shared with the stem, the distance between the frontier on the line attached to that loop and the robot position, when it explored the loop, is at least

$$3m^w + \lfloor \frac{3}{2} m^w \rfloor + \lfloor \frac{r}{2} \rfloor + 1 - (2r + 1).$$

That distance is computed by considering the path that the robot should travel by backtracking to the loop and could be easily derived by knowing that the robot started from the vertex of the loop shared with the stem, the sensor range r of the robot, and the length of the loop. Instead, the distance between the last frontier vertex of the loop and the closest frontier vertex on the stem is at most $3r+2$. (Note that the other path between the last frontier vertex of the loop and the frontier vertex on line of the same loop is greater than $3r + 2$ by construction.) To prove that the robot chooses the frontier vertex on the stem the following inequality should hold

$$3r + 2 < 3m^w + \lfloor \frac{3}{2} m^w \rfloor + \lfloor \frac{r}{2} \rfloor + 1 - (2r + 1).$$

After some math, we have

$$\frac{9}{2} r + 4 < \frac{9}{2} m^w$$

As $m^{w-1} - 1 < r \leq m^w - 1$, that inequality becomes

$$\frac{9}{2} m^w - \frac{1}{2} < \frac{9}{2} m^w$$

which is always true. For any loop of level $i > w + 1$, the inequality is still always true. Moreover, the distance from the last frontier vertex selected by the robot on the loop of level i to the nearest frontier vertex on the line of level w is always greater than or equal to the distance to the nearest frontier vertex on the stem. Thus, the frontier vertices on the lines are never chosen in the worst case at the first traversal.

Then, the robot traverses the stem from v_{m^m} to v_0 exploring the lines at level w. Next, from v_0 to v_{m^m} the robot explores the lines at level $w + 1$. This way of traveling over the vertices of the stem goes on until the lines of the last level m are explored. This behavior is caused by the fact that the distance from the current selected frontier vertex on a line of loop of level $i - 1$ to the nearest frontier vertex on a line of a loop of level $i-1$ is less or equal than the distance to the nearest frontier vertex on the lines loop of level i. The minimum difference between the two distances happens when two loops of level $i-1$ and i are attached to the same vertex of the stem. By construction, those distances are m^i and

$$\lceil(\frac{i-w}{2}+1)\cdot m^w\rceil + \sum_{j=w+1}^{i-1} m^j + d + 1$$

where d is the number of perceived vertices of the lines and can have values $\lfloor\frac{r}{2}\rfloor \leq d \leq r$, respectively. Thus, considering the maximum distance from the vertex on the stem shared by the loops at level $i-1$ and i to the frontier vertex on the line of the loop of level $i-1$ ($d_1 = r$) and considering the minimum distance to the line of loop of level i ($d_2 = \lfloor\frac{r}{2}\rfloor$) we have the following:

$$m^i + \lceil(\frac{i-w}{2}+1)\cdot m^w\rceil + \sum_{j=w+1}^{i-1} m^j + d_1 + 1 \leq$$
$$\leq \lceil(\frac{i+1-w}{2}+1)\cdot m^w\rceil + \sum_{j=w+1}^{i} m^j + d_2 + 1$$

(simplifying the equal terms)

$$d_1 - d_2 \leq \lceil\frac{i+3-w}{2}\cdot m^w\rceil - \lceil\frac{i+2-w}{2}\cdot m^w\rceil$$

(substituting d_1 and d_2 with their bounds)

$$\lceil\frac{r}{2}\rceil \leq \lceil\frac{i+3-w}{2}\cdot m^w\rceil - \lceil\frac{i+2-w}{2}\cdot m^w\rceil$$

(remind that $r \leq m^w - 1$)

$$\lceil\frac{m^w-1}{2}\rceil \leq \lceil\frac{i+3-w}{2}\cdot m^w\rceil - \lceil\frac{i+2-w}{2}\cdot m^w\rceil.$$

Notice that, if m^w is even, the difference on the right-hand side is $\frac{m^w}{2}$, thus the inequality is always true. If m^w is odd and $i+3-w$ is odd, the difference is equal to $\frac{m^w+1}{2}$, while, if $i+3-w$ is even, becomes $\frac{m^w-1}{2}$. In either case the inequality is always true. In the particular case of $i = w$, we get to the same result, as $\frac{m^w-1}{2} \leq \lceil\frac{3}{2}\cdot m^w\rceil - m^w$, which is always true. This proves that, after the first stem traversal, when traversing the stem, the robot explores all the lines of the loops at the same level i.

The number of edge traversals is:

$$LB_{\mathscr{S}_d} \geq (m-w)m^m + \sum_{i=w+1}^{m} m^{m-i}\left(r + 2(i-w)m^w + 4\sum_{j=w}^{i-1} m^j\right) + \\ + 3m^{m-1} + (m^{m-w}+1)(r+1)$$

$$= (m-w)m^m + r\frac{m^m - m^w}{m^w(m-1)} + 2\frac{m^{m+1} - m^{w+2} + wm^{w+1} - wm^w}{(m-1)^2} + \\ + 4\frac{m^m(m-w)}{m-1} - 4\frac{m^m - m^w}{(m-1)^2} + 3m^{m-1} + (m^{m-w}+1)(r+1)$$

Note that $LB_{\mathscr{S}_d}$ is $\Omega((m-w)m^m)$. Hence, as $|V|$ is $\theta(m^m)$, $w \leq \log_m(r+1)+1$, and, because $m \geq \frac{\log m^m}{\log\log m^m}$ as shown in Koenig [1998], we have that $LB_{\mathscr{S}_d}$ is

$$\Omega\left(\left(\frac{\log|V|}{\log\log|V|} - \log_m(r+1)\right)|V|\right)$$

Changing the $\log$ base we have the following result

$$\begin{aligned} \log_m(r+1) &= \frac{\log(r+1)}{\log m} \\ &\leq \frac{2\log(r+1)}{\log\log|V|} \end{aligned}$$

Since it holds that

$$\begin{aligned} \log m^2 &\geq \log(m\log m) \\ 2\log m &\geq \log\log m^m \\ \log m &\geq \frac{1}{2}\log\log m^m \end{aligned}$$

Putting all together we have as final result the lower bound

$$LB_{\mathscr{S}_d} = \Omega\left(\frac{\log|V| - 2\log(r+1)}{\log\log|V|}|V|\right)$$

□

Corollary 4.1. *Given a robot sensor range $\lceil \bar{r} \rceil$, where $\bar{r} \in \mathbb{R}_{>}0$ such that*

$$\bar{r} = \sqrt{\frac{|V|}{\log^c|V|}} - 1$$

where c is an arbitrary positive constant, the worst-case lower bound on traveled distance for $\mathscr{S}_d$ is $LB_{\mathscr{S}_d} = \Omega(|V|)$ edge traversals, on any finite undirected connected graph $G = (V, E)$, where the weight of each edge is equal to 1.

Proof. Given an arbitrary positive constant c, the lower bound is linear if $LB_{\mathscr{S}_d} = \Omega(c \cdot |V|)$. Namely, considering the lower bound found in Theorem 4 we have that the following relation must holds

$$\frac{\log |V| - 2\log(r+1)}{\log\log |V|} = c$$

Thus, after some basic math

$$\begin{aligned}\log(r+1) &= \frac{1}{2}\log|V| - \frac{c}{2}\log\log|V| \\ &= \log\sqrt{\frac{|V|}{\log^c |V|}}\end{aligned}$$

And finally we obtain the radius such that the worst-case lower bound on traveled distance for $\mathscr{S}_d$ is linear, as

$$\bar{r} = \sqrt{\frac{|V|}{\log^c |V|}} - 1$$

□

4.3 Information Gain Criterion

Now let us analyze the worst case of $\mathscr{S}_g$.

For this strategy the upper bound we found could seem not so tight, since it is the worst case upper bound for any exploration strategy. Anyway, as we will see studying the lower bound, it is not so far to be tight.

Theorem 5. *Given a robot sensor range $r \in \mathbb{N}_{>0}$, the worst-case upper bound on the traveled distance for $\mathscr{S}_g$ is*

$$UB_{\mathscr{S}_g} = \left(|V| - \frac{|V|-1}{r+1}\right)\left(2\frac{|V|-1}{r+1} - 1\right)$$

edge traversals, on any finite undirected connected graph $G = (V, E)$, where the weight of each edge is equal to 1.

Proof. The proof trivially derives from considering the following scenario: the robot at each time step i could have to traverse all the vertices perceived up to i (which possibly could be the worst case for any strategy). More formally, we have that $\sum_{i=0}^{\bar{k}-1} |V_i| + 1$. To maximize the number of traversed vertices at each time step i, as we assume that the robot can perceive just one vertex at a time, we have an additional term that takes into account all the initial perceived vertices $|P_0| = |V_0| = |V| - \bar{k}$.

In the following we derive the upper bound by doing some math.

$$\begin{aligned}\sum_{i=0}^{\bar{k}-1}|V_i|+1 &\leq \sum_{i=1}^{\bar{k}}(|V|-\bar{k}+i)\\ &= (|V|-\bar{k})\bar{k}+\sum_{i=1}^{\bar{k}} i\\ &= (|V|-\bar{k})\bar{k}+\frac{\bar{k}(\bar{k}+1)}{2}\\ &= \bar{k}\left(|V|-\frac{\bar{k}+1}{2}\right)\\ &= \left(|V|-\frac{|V|-1}{r+1}\right)\left(2\frac{|V|-1}{r+1}-1\right)\end{aligned}$$

□

Assuming $r=\epsilon$ ($\epsilon \to 0$) we have to consider $\bar{k}$ for r even. It results that the worst-case upper bound for $\mathscr{S}_g$ is $\frac{|V|(|V|-1)}{2}$, as $\bar{k}=|V|-1$, which coincides with the well-known upper bound for any exploration strategy in a fixed graph scenario $O(|V|^2)$. This is not surprising since this bound is itself an upper bound for any exploration strategy with a certain perception radius r. This is guaranteed by the first inequality used in the proof, namely

$$\sum_{i=0}^{\bar{k}-1}|V_i|+1 \leq \sum_{i=1}^{\bar{k}}(|V|-\bar{k}+i)$$

which ensure that the way to maximize the number of known vertices traversed, at each time step i, is to perceive $|V|-\bar{k}$ vertices at time step 0, and then, to perceive just 1 vertex at each time step i, $1 \leq i \leq \bar{k}$.

We are now interested to find a lower bound for $\mathscr{S}_g$. The following result allow us to compare strategies between them and to verify the goodness of the upper bound we just found. As we will see, the weakness of a purely information gain-based strategy is manifested on sparse graphs.

The following theorem presents a lower bound for $\mathscr{S}_g$.

Theorem 6. *Given a robot sensor range $r \in \mathbb{N}$, the worst-case lower bound on the traveled distance for $\mathscr{S}_g$ is*

$$LB_{\mathscr{S}_g}=\frac{r+1}{2}\left(\frac{|V|-r}{r+1}\right)\left(\frac{|V|-r}{r+1}-1\right)$$

edge traversals, on any finite undirected connected graph $G=(V,E)$, where the weight of each edge is equal to 1.

Proof. Let us consider a line graph. The robot starts from the middle of the line. In case of even number of vertices we can choose arbitrarily one of the two vertices in the middle. The proof develops from the idea that the robot can go back and forth to frontier vertices traveling over all of the already perceived vertices. Trivially the number of chosen frontiers along the exploration path is $k \geq \frac{|V|-r}{r+1} - 1$. At each time step the robot can perceive $r + 1$ vertices, thus the lower bound is

$$\sum_{i=1}^{k} (i(r+1)) = (r+1)\frac{k(k+1)}{2}$$
$$\geq \frac{r+1}{2}\left(\frac{|V|-r}{r+1}\right)\left(\frac{|V|-r}{r+1} - 1\right)$$

□

Assuming again $r = \epsilon$ ($\epsilon \to 0$), we have the worst-case lower bound for $\mathscr{S}_g$ is $\frac{|V|(|V|-1)}{2}$, which, also in this case, coincides with the well-known lower bound for any exploration in a fixed graph scenario $\Omega(|V|^2)$. Apparently, $\mathscr{S}_g$ performs fairly bad with respect to $\mathscr{S}_d$ in the wort case.

4.4 Combination of Distance and Information Gain

Now we wonder if, adopting the information gain as tie breaker it improves or worsen the performance of $\mathscr{S}_d$. The following proposition ensure that, in the worst case, the currently known bounds are equal for $\mathscr{S}_d$ and $\mathscr{S}_{dg}$.

Proposition 1. *The worst-case upper bound on the traveled distance for $\mathscr{S}_{dg}$ is $UB_{\mathscr{S}_{dg}} = UB_{\mathscr{S}_d}$ and the lower bound is $LB_{\mathscr{S}_{dg}} = LB_{\mathscr{S}_d}$ on any finite undirected connected graph $G = (V, E)$, where the weight of each edge is equal to* 1.

Proof. To prove the upper bound we can notice that at each step i, the frontier set of $\mathscr{S}_{dg}$ is a subset of the frontier set of $\mathscr{S}_d$. Thus, the set of all the possible exploration paths, on a given G, for $\mathscr{S}_{dg}$, is a subset of all the possible exploration paths, on that G, for $\mathscr{S}_d$. Thus the upper bound for $\mathscr{S}_d$ holds.

Looking at the worst-case lower bound graph for $\mathscr{S}_d$ we can notice that, the behavior of $\mathscr{S}_{dg}$ necessarily diverges from that of $\mathscr{S}_d$, when $r = m^w - 1$ for a given w ($1 \leq w \leq m$). This happens at the first stem traversal, because the robot cannot choose to enter in a loop before to explore the entire stem. In those cases it is trivial to add $O(m^m)$ single vertices on the loops, at distance $r + 1$ from the stem, to make feasible the behavior described for $\mathscr{S}_d$. □

From a worst-case point of view, S_d and S_{dg} are probably too similar to find out significative differences. Analysis of their differences in the average case is provided in the next chapter.

4.5 Comparison of Bounds

Note that, despite the derived bounds have the same asymptotic complexity of those that do not embed r in the computation of the bounds, the bounds we found have lower actual values, because we explicitly considered the sensor range r. This analysis is of interest especially when dealing with robots as their movements are physical and not just computational.

Figure 4.6 shows the trend of the worst-case bounds for $\mathscr{S}_d$ ($\mathscr{S}_{dg}$) compared to the ones found by Tovey and Koenig [2003] (UB_{TK}) and Koenig et al. [2001] (LB_{KTS}), and to the worst-case bounds of $\mathscr{S}_g$, considering $|V| = 1000$ and $r \in \{1, \cdots, \lfloor\frac{|V|-1}{2}\rfloor\}$. Note that the plot of the curves related to the lower bounds shows just the their trends and not their actual values, as they depend on the worst-case graphs built as shown in the related proofs above.

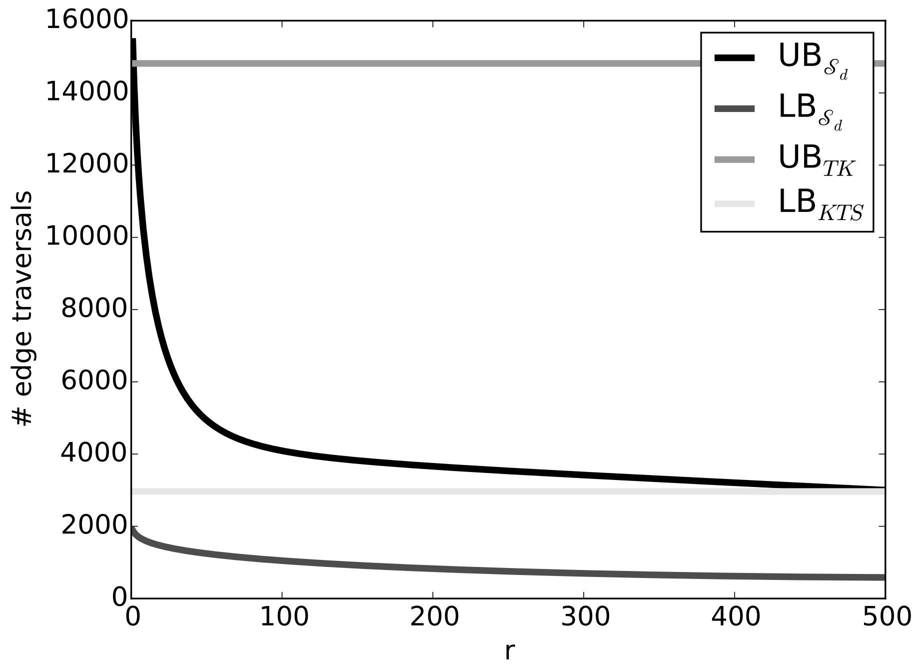

Figure 4.6: The trend of worst-case bounds of $\mathscr{S}_d$ with respect to UB_{TK} and LB_{KTS}, considering $|V| = 1000$ and $r \in \{0, \cdots, \lfloor\frac{|V|-1}{2}\rfloor\}\}$.

In Figure 4.6, we can see that UB_{TK} is higher than $UB_{\mathscr{S}_d}$ (except for the case of $r < 2$ as we discussed in the Section 4.2). This can be explained by the fact that UB_{TK} does not fully consider the more powerful sensor that allows to perceive more vertices. Similar explanation for the fact that $LB_{\mathscr{S}_d}$ is lower than LB_{KTS}. In particular, in order to generalize for any r the robot behavior in the worst case, we

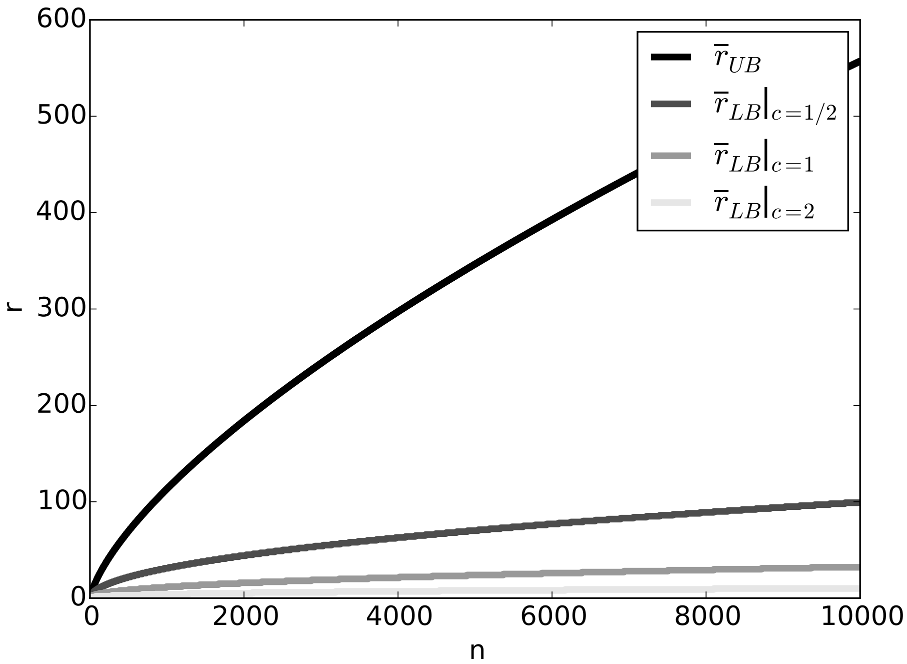

Figure 4.7: The trend of $\bar{r}$ for worst-case bounds.

added several vertices not strictly necessary in the case of $r = \epsilon$. In Figure 4.7 we plotted the trend of $\bar{r}$ (the perception radius such that the worst-case bound is linear, given a certain number of vertices to explore) for $UB_{\mathscr{S}_d}$ and $LB_{\mathscr{S}_d}$, and given $|V| = \{10, \cdots, 10000\}$. For clarity, let us call $\bar{r}_{UB}$ and $\bar{r}_{LB}$ the $\bar{r}$ of Corollary 3.1 and Corollary 4.1, respectively.

For any radius greater than $\bar{r}_{UB}$ we have that the number of edge traversals is lower than $4n$ in the worst case; for any radius lower than $\bar{r}_{LB}(c)$ we have that the number of edge traversals is not linear in the worst case. Since we gave just the order of the lower bound and because linearity holds for any positive constant c such that $\Omega(c \cdot |V|)$, three values of $\bar{r}_{LB}(c)$ are plotted, for $c = \{\frac{1}{2}, 1, 2\}$. For any radius between $\bar{r}_{UB}$ and $\bar{r}_{LB}(c)$ the number of edge traversal may be or may not be linear. From the graph we can see that the trend of both $\bar{r}_{UB}$ and $\bar{r}_{LB}$ is clearly sublinear with respect the number of vertices to explore, as we already discussed for $\bar{r}_{UB}$ and as was evident from the closed form of $\bar{r}_{LB}$ (Corollary 4.1).

Also, we can observe that for low values of r, a small increase of r leads to a large decrease of $UB_{\mathscr{S}_d}$. Instead, we have almost constant trend after a certain value of r. More generally, our analysis shows theoretically that increasing r, the exploration process is shortened, which is an intuitively evident result sound with several experimental findings (e.g., [Amigoni, 2008]), but also that it is not

$\|V\|$ \ ETG	50%	60%	70%	80%	90%
500	10	14	20	31	66
1000	12	18	26	41	90
2000	15	22	32	52	119
5000	19	29	44	73	167
10000	23	35	56	94	215
20000	27	43	70	122	278
50000	34	57	96	171	394
100000	41	70	122	223	515
200000	48	86	155	290	676

Table 4.2: Values of the perception radius r (rounded to the next integer) such that the gain on the upper bound (with respect the maximum gain, namely, for $r = \lfloor \frac{|V|-1}{2} \rfloor$, and the minimum gain corresponding to $r = 0$) is $ETG = \{50\%, 60\%, 70\%, 80\%, 90\%\}$ for some $|V|$.

convenient to increase the perception radius after a certain value, even if the sensor is perfect (e.g., the footprint sensor in [Quattrini Li et al., 2012]). A first result about this value is given by Corollary 3.1. Other numerical results are shown in Table 4.2. More precisely, assumed as $ETG = 100\%$ the gain, in terms of edge traversals, obtained on the $\mathscr{S}_d$ upper bound in correspondence of $r = \lfloor \frac{|V|-1}{2} \rfloor$ and $ETG = 0\%$ the gain for $r = 0$ (UB_{TK}), in Table 4.2 radii are computed (rounded to the next integer) such that the gain is $ETG = \{50\%, 60\%, 70\%, 80\%, 90\%\}$ for some $|V|$. Through this numerical analysis, we can note that, half of the potential gain we may have in the worst-case, increasing the perception radius, corresponds to very small r. In particular, for $|V| = 2 \cdot 10^5$ to achieve $ETG = 50\%$, we need just a sensor radius $r = 48$.

In Figure 4.8 $\mathscr{S}_g$ and $\mathscr{S}_d$ are compared through their worst-case bounds. We can see that $UB_{\mathscr{S}_g}$ is higher than $UB_{\mathscr{S}_d}$, as we expected, while $LB_{\mathscr{S}_g}$ crosses $LB_{\mathscr{S}_d}$ at $r \approx \frac{|V|}{4}$ (this trend holds also for other values of $|V|$). Anyway, this does not mean that the lower bound of $\mathscr{S}_g$ is lower than the lower bound of $\mathscr{S}_d$, since, is easy to check that the behavior shown for $\mathscr{S}_d$ on its worst-case graph, it is feasible also for $\mathscr{S}_g$ (similarly to how we proved the lower bound for $\mathscr{S}_{dg}$). Thus, the real lower bound of $\mathscr{S}_g$ is $\max(LB_{\mathscr{S}_g}, LB_{\mathscr{S}_d})$. Moreover, the upper bound and the lower bound of $\mathscr{S}_d$ are fairly close after a certain (relatively) small radius ($r \sim 50$ in Figure 4.6) and have the same trend. A last remark concerns the number of vertices to explore. So far we have considered to explore all the vertices V in the graph, but usually, in robotics, this is not required. Often it is enough to explore a certain percentage p of the environment, as, for example, in search and rescue

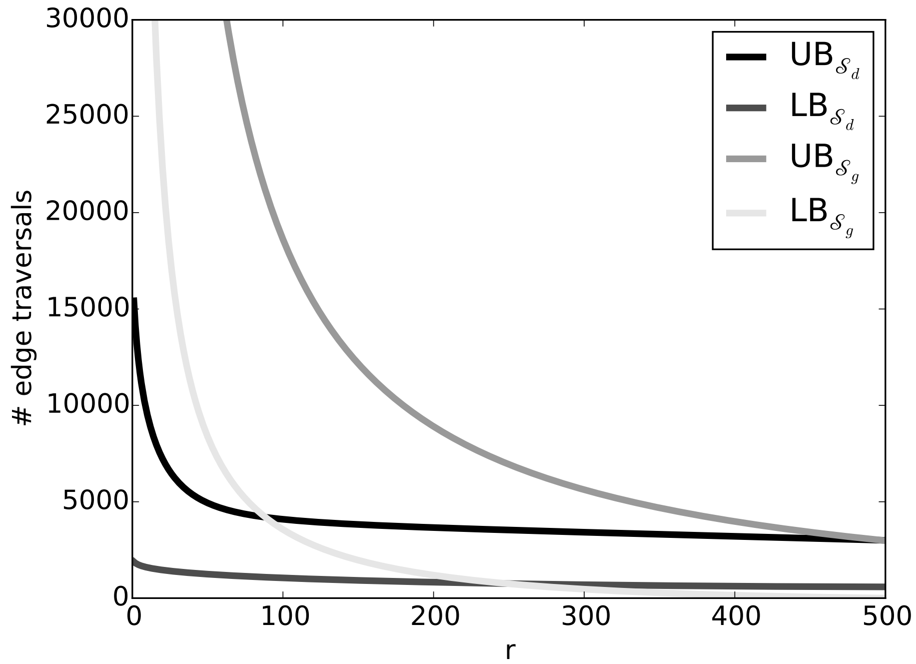

Figure 4.8: The trend of worst-case bounds of $\mathscr{S}_d$ ($\mathscr{S}_{dg}$) and $\mathscr{S}_g$ (for clarity, is a zoomed portion of the complete plot), considering $|V| = 1000$ and $r \in \{0, \cdots, \lfloor \frac{|V|-1}{2} \rfloor\}$.

scenarios. It is easy to check that we can easily change $|V|$ to $\lceil |V| \cdot p \rceil$ in each formula and have again sound bounds.

4.6 Experiments on Random Generated Graphs

In this section for completeness we present some results obtained by the strategies we consider on random generated graphs. Results are presented varying the perception radius r, the number of vertices, and the number of edges. For each setting fifty random graphs have been generated through the method of Erdős and Rényi [1964].

As discussed in the model, for $r = 0$, there is no specific reason because $\mathscr{S}_{dg}$ performs better than $\mathscr{S}_d$. While, for radii greater than 0 results are sound with respect to the experiments on real and simulated robots in other works (Gonzáles-Baños and Latombe [2002], Amigoni and Caglioti [2010]), since $\mathscr{S}_{dg}$ performs better than $\mathscr{S}_d$. Even if this kind of results was expected, it was absolutely not obvious, since this model (a graph model) is completely different to the others where information gain-based strategies have been tried. Moreover, for very low values of the sensor radius, $\mathscr{S}_g$ performs worse than $\mathscr{S}_d$ and $\mathscr{S}_{dg}$ which is sound with bounds, while, after a certain radius it performs better than $\mathscr{S}_d$ converging to $\mathscr{S}_{dg}$. Increasing again the sensor radius, the number of edge traversals for the strategies converge to 0. From experiments we can notice that, the impact of the information gain is slightly higher increasing the number of edges.

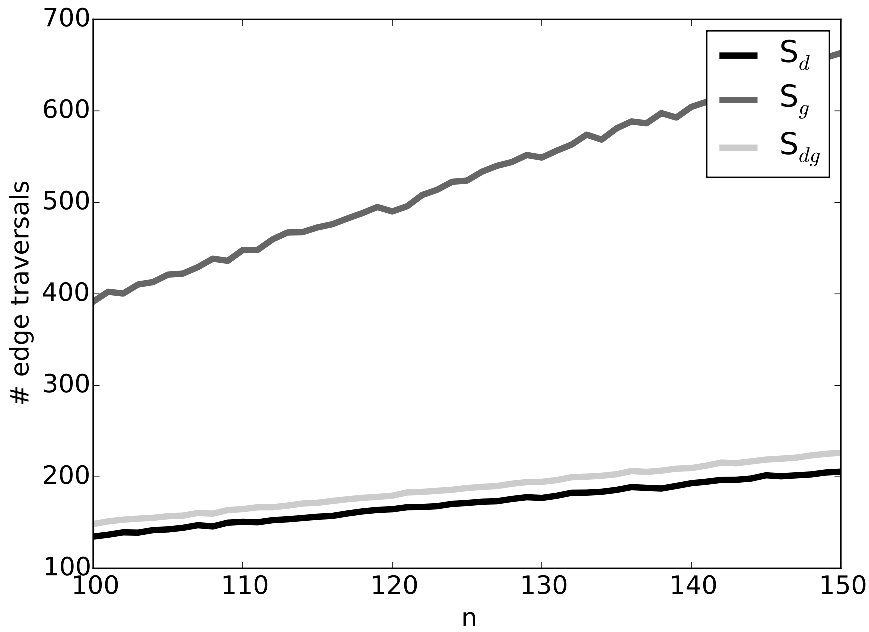

Figure 4.9: The number of edge traversals for $\mathscr{S}_d$, $\mathscr{S}_g$, and $\mathscr{S}_{dg}$, considering $|V| = 100, \cdots, 150$, $|E| = 2|V|$ and $r = 0$.

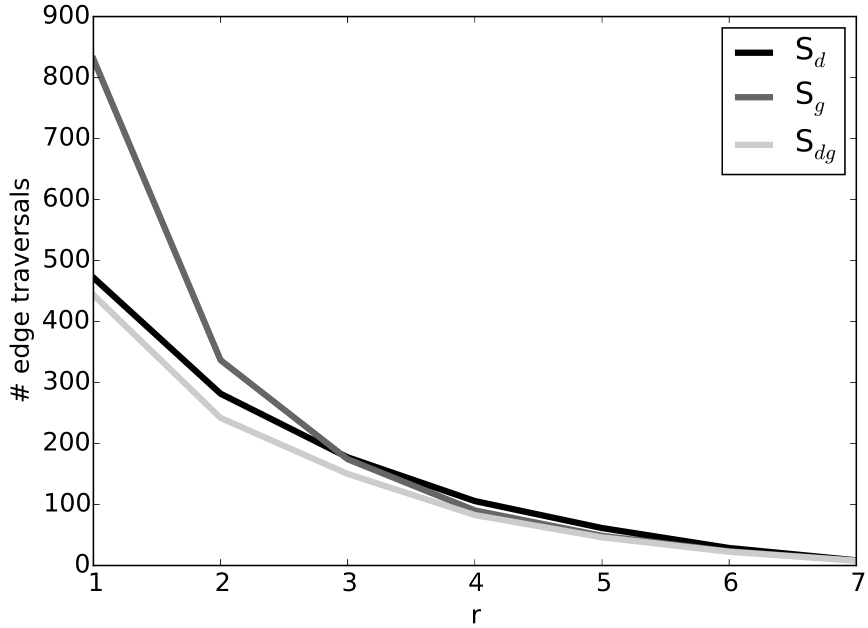

Figure 4.10: The number of edge traversals for $\mathscr{S}_d$, $\mathscr{S}_g$, *and* $\mathscr{S}_{dg}$, *considering* $|V| = 500$, $|E| = 1000$ *and* $r \in \{1, \cdots, 7\}$.

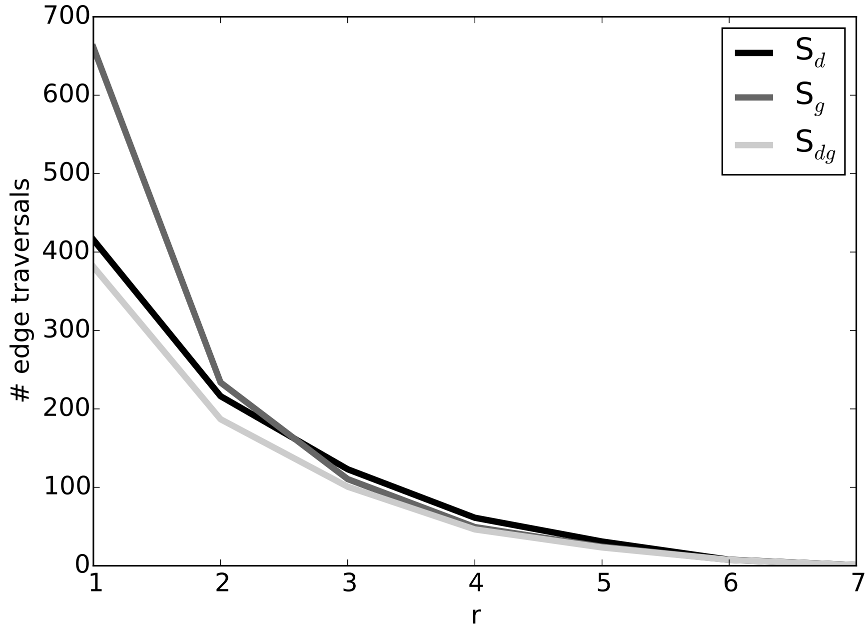

Figure 4.11: The number of edge traversals for $\mathscr{S}_d$, $\mathscr{S}_g$, *and* $\mathscr{S}_{dg}$, *considering* $|V| = 500$, $|E| = 1200$ *and* $r \in \{1, \cdots, 7\}$.

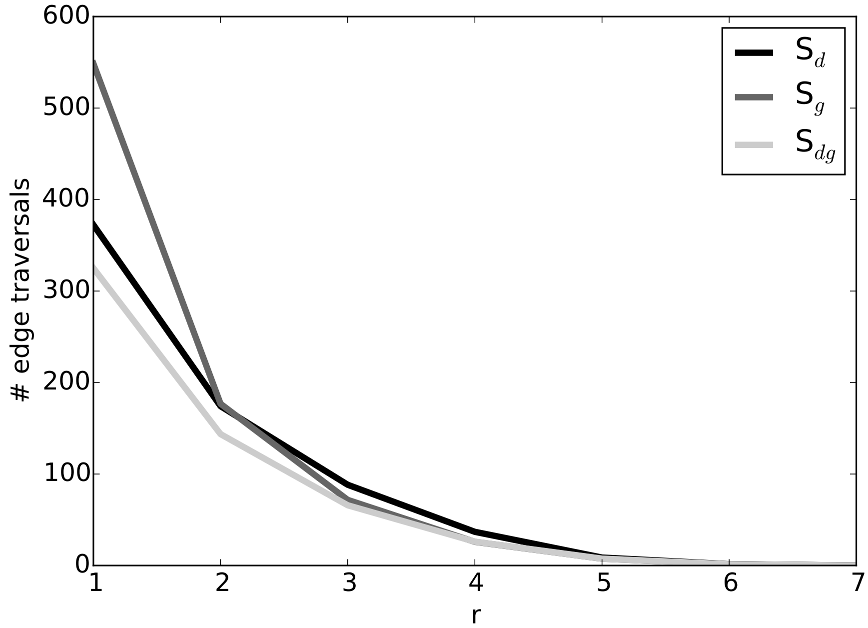

Figure 4.12: The number of edge traversals for $\mathscr{S}_d$, $\mathscr{S}_g$, *and* $\mathscr{S}_{dg}$, *considering* $|V| = 500$, $|E| = 1400$ *and* $r \in \{1, \cdots, 7\}$.

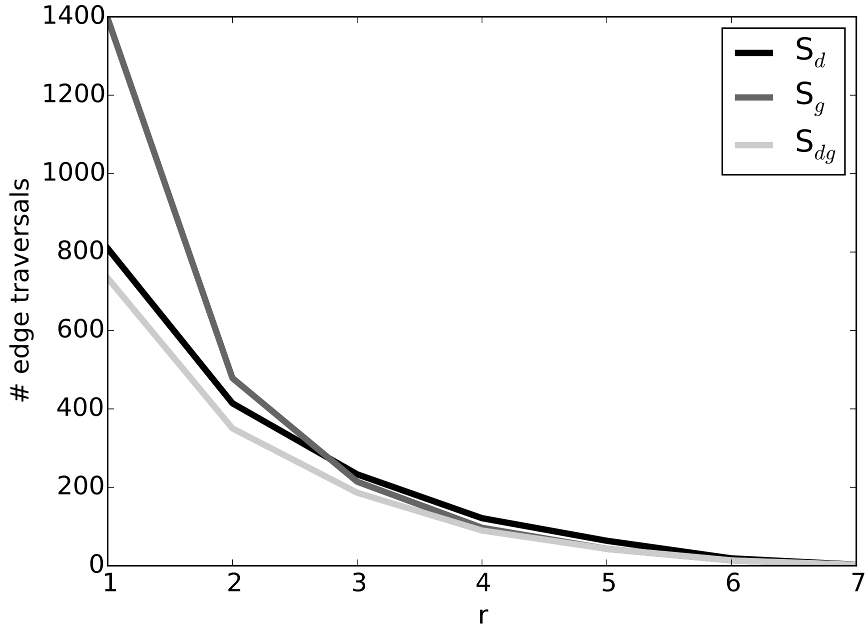

Figure 4.13: The number of edge traversals for $\mathscr{S}_d$, $\mathscr{S}_g$, *and* $\mathscr{S}_{dg}$, *considering* $|V| = 1000$, $|E| = 2500$ *and* $r \in \{1, \cdots, 7\}$.

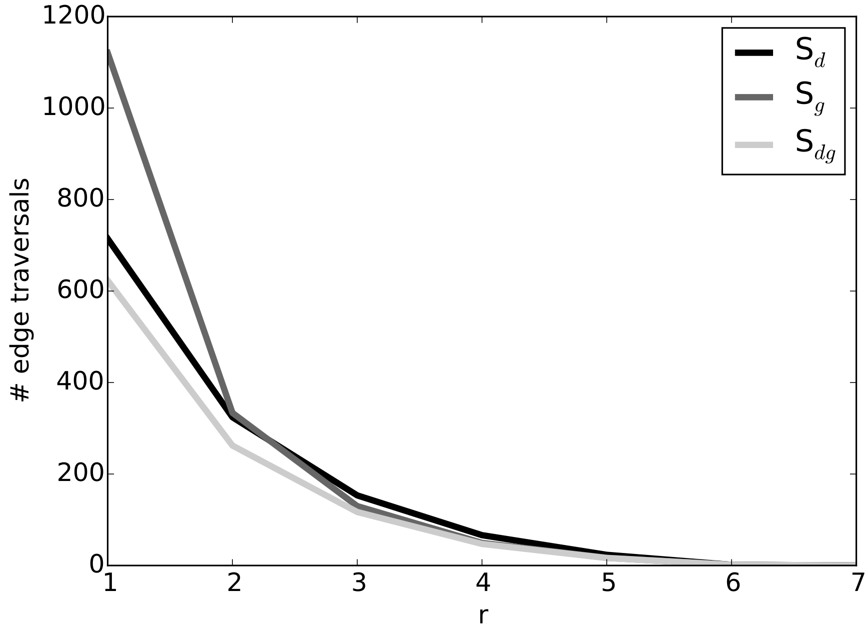

Figure 4.14: The number of edge traversals for $\mathscr{S}_d$, $\mathscr{S}_g$, and $\mathscr{S}_{dg}$, considering $|V| = 1000$, $|E| = 3000$ and $r \in \{1, \cdots, 7\}$.

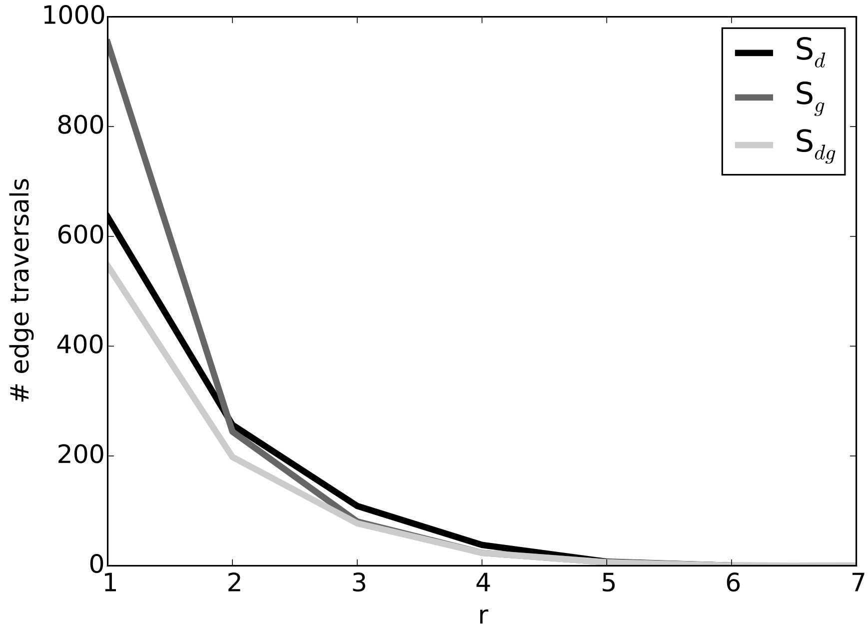

Figure 4.15: The number of edge traversals for $\mathscr{S}_d$, $\mathscr{S}_g$, and $\mathscr{S}_{dg}$, considering $|V| = 1000$, $|E| = 3500$ and $r \in \{1, \cdots, 7\}$.

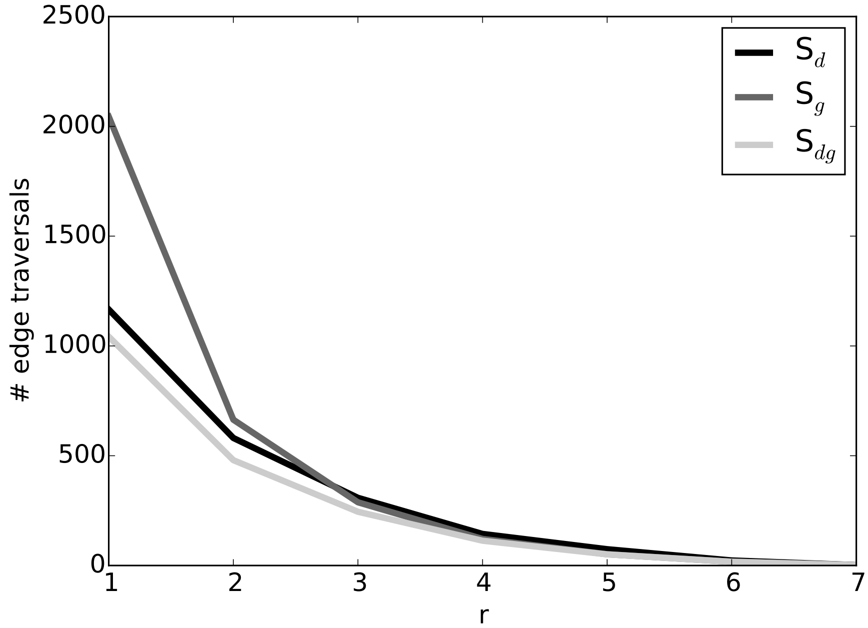

Figure 4.16: The number of edge traversals for $\mathscr{S}_d$, $\mathscr{S}_g$, and $\mathscr{S}_{dg}$, considering $|V| = 1500$, $|E| = 4000$ and $r \in \{1, \cdots, 7\}$.

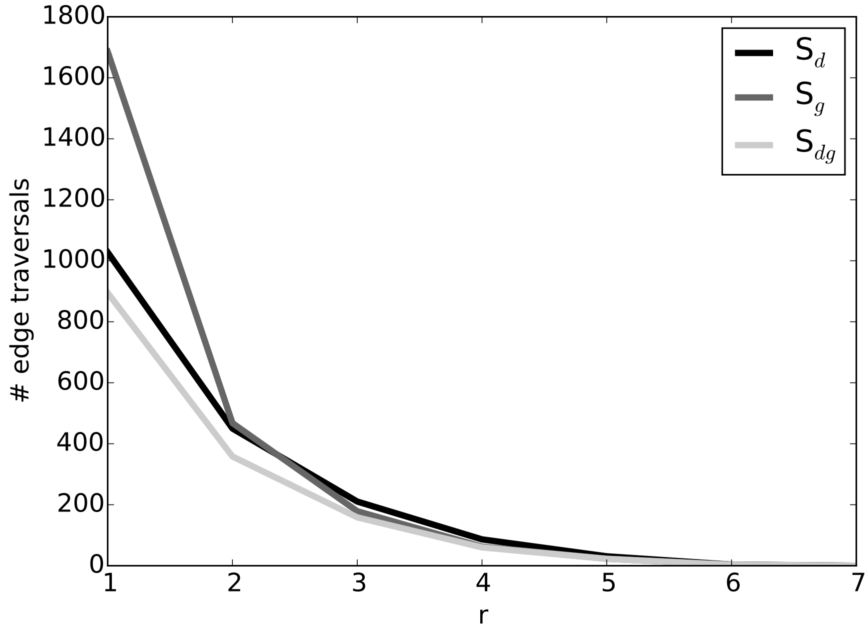

Figure 4.17: The number of edge traversals for $\mathscr{S}_d$, $\mathscr{S}_g$, and $\mathscr{S}_{dg}$, considering $|V| = 1500$, $|E| = 4750$ and $r \in \{1, \cdots, 7\}$.

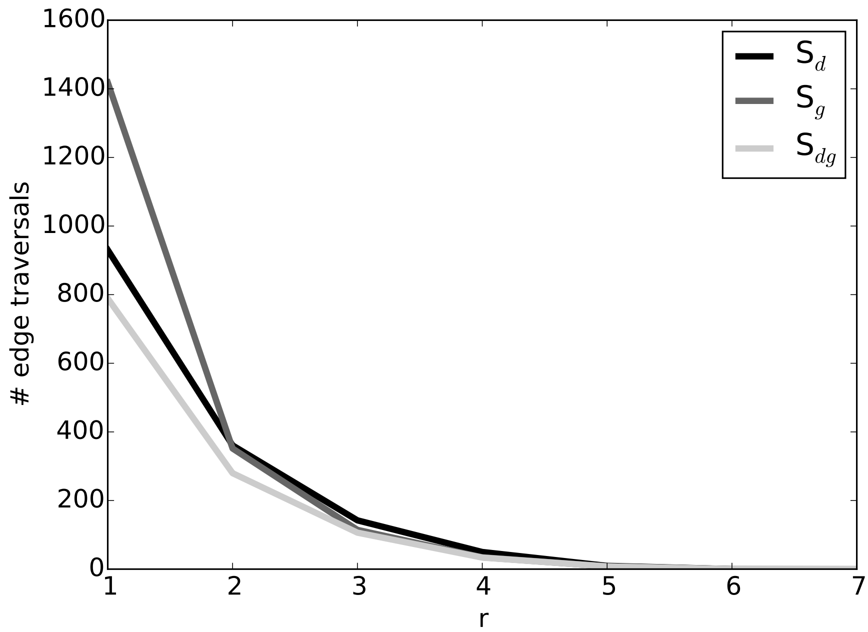

Figure 4.18: The number of edge traversals for $\mathscr{S}_d$, $\mathscr{S}_g$, and $\mathscr{S}_{dg}$, considering $|V| = 1500$, $|E| = 5500$ and $r \in \{1, \cdots, 7\}$.

Chapter 5

Average Case Analysis

In this chapter we compare $\mathscr{S}_d$ and $\mathscr{S}_{dg}$ in the average case, since, from the worst case analysis they seem more promisingly than $\mathscr{S}_g$. Because of the difficulties to evaluate the average performance in generic graphs, we consider some classes of graphs, focusing the analysis to the case of $r = \epsilon$ ($\epsilon \to 0$), and we estimate the number of edge traversals given partial information, such as, in an indoor environment, the corridors structure and the number of rooms for each corridor.

In the average-case analysis we have to deal with two different sources of randomness. The first one is due to the fact that $\mathscr{S}_d$ and $\mathscr{S}_{dg}$ break ties randomly, and thus, given a graph, several exploration paths could be possible. The second one is due to the fact that we may want to consider a certain set of different graphs (possibly all the graphs) with a certain probability, and not a single fixed graph (since the environment could be unknown, or we may want to achieve a result for an entire class of graphs with similar features).

So, we have to deal with two means. First, a mean over all the feasible paths (according to the strategy) on a given graph. Then, a mean over all the graphs that belong to a certain set (which may be the universal graphs set) with a certain probability distribution.

Before we go any further, let us introduce some notation. Given $\mathscr{S}_d$ strategy, we know that, at each step i, while the robot is in v_i, there is a subset

$$F^d_{i_{\min}} = \arg\min_{v \in F^d_i} d(v_i, v) \subseteq F^d_i$$

of vertices in the frontier set F^d_i that have the same (minimum) distance from v_i. In a similar way we define $F^{dg}_{i_{\min}}$.

So, depending on how a vertex is chosen from either $F^d_{i_{\min}}$ or $F^{dg}_{i_{\min}}$ for $\mathscr{S}_d$ and $\mathscr{S}_{dg}$, respectively, the robot could have better or worse performance. As we saw in the previous chapter, and in particular studying the lower bound for $\mathscr{S}_d$ and $\mathscr{S}_{dg}$

(Theorem 4 and Proposition 1), a wrong tie breaking could lead to a super-linear number of edge traversals. The same behavior shown in Theorem 4 would not be feasible if the robot decides to explore first all the single lines. Our conjecture is that the super linearity behavior of $\mathscr{S}_d$ is due to the wrong tie breaking, as we will see in Section 5.1.

Now we define a function that, given a graph $G = (V, E)$, and a starting vertex v_0, returns the average number of edge traversals on G. For this purpose, let us define a function $\Gamma()$ for both the strategies, $\mathscr{S}_d$ and $\mathscr{S}_{dg}$, that we will use in our analysis. For $\mathscr{S}_d$ we define this function as

$$\Gamma_d(G, v_i, V_i) = \begin{cases} \dfrac{1}{|F^d_{i_{\min}}|} \displaystyle\sum_{v \in F^d_{i_{\min}}} d(v_i, v) + \Gamma_d(G, v, V_i \cup P_{i+1}) & \text{if } \dfrac{|V_i|}{|V|} < p \\ 0 & \text{if } \dfrac{|V_i|}{|V|} \geq p \end{cases}$$

where p is the goal percentage, namely, the percentage of vertices in the environment we have to explore. Since this function only depends on $F_{i_{\min}}$ we can define the same function for $\mathscr{S}_{dg}$ as

$$\Gamma_{dg}(G, v_i, V_i) = \begin{cases} \dfrac{1}{|F^{dg}_{i_{\min}}|} \displaystyle\sum_{v \in F^{dg}_{i_{\min}}} d(v_i, v) + \Gamma_{dg}(G, v, V_i \cup P_{i+1}) & \text{if } \dfrac{|V_i|}{|V|} < p \\ 0 & \text{if } \dfrac{|V_i|}{|V|} \geq p \end{cases}$$

Intuitively, $\Gamma()$ is a recursive function that, at a given step i, for each vertex v in $F_{i_{\min}}$, weights uniformly the distance from the current position to v, plus the value of $\Gamma()$ computed in v. If the number of remaining vertices to explore is 0 (namely, the ratio between $|V_i|$ and $|V|$ is larger than the goal percentage p) the recursion stops and the value of $\Gamma()$ is 0 for any vertex. Given these functions, it is fairly natural, now, to define the average number of edge traversals on G given the starting vertex v_0 as

$$\Gamma_d(G, v_0) = \Gamma_d(G, v_0, P_0)$$

and for $\mathscr{S}_{dg}$ as

$$\Gamma_{dg}(G, v_0) = \Gamma_{dg}(G, v_0, P_0)$$

These functions allow us to deal with the first of the two sources of randomness we discussed above. We will see how to deal with the second source of randomness, defining each specific class of environments.

5.1 Average Case vs. Worst Case

From a theoretical point of view it is interesting to note that the perfect tie breaker for $\mathscr{S}_d$, at each step i, breaks as follow

$$\hat{v} = \arg\min_{v \in F^d_{i_{\min}}} d(v_i, v) + \Gamma_d(G, v, V_i \cup P_{i+1}) \tag{5.1}$$

This can be done if the environment is known and the path is computed offline. Then, be $\mathscr{S}^*_d$ an exploration strategy that minimizes the distance and breaks tie perfectly (as in (5.1)), then we formulate the following conjecture.

Conjecture 1. *The worst-case bound on the traveled distance for $\mathscr{S}^*_d$ is $\Theta(|V|)$ edge traversals, for any sensor radius r, on any finite undirected connected graph $G = (V, E)$, where the weight of each edge is equal to* 1.

We believe that tie breaking is the main issue in the $\mathscr{S}_d$ worst case. However, from the average-case viewpoint, to break ties randomly can be a smart choice. Looking at the behavior shown in Theorem 4, we can notice that the probability to break ties in such a way that the robot continuously re-traverse all the stem, is very low, and most of the times $\mathscr{S}_d$ behaves linearly on that worst-case graph.

Focusing on the worst-case graph $\bar{G}$ for $\mathscr{S}_d$ and $r = \epsilon$ ($\epsilon \to 0$) shown in [Koenig et al., 2001] and reported in Figure 2.5, we know that $|V| = \Theta(n^n)$ and, in the worst case, *LB* $= \Omega(n^{n+1})$, for any $n \geq 3$. However, consider for example the first stem traversal of the robot: at each step, the robot has to decide whether to continue on the stem or to go the frontier vertex on the line of the loops as they are at the same distance from a current robot position. Thus, the probability that the robot traverses the stem without exploring the lines is less than

$$\left(\frac{1}{2}\right)^{\alpha}$$

where α is the total number of loops plus all the n^{n-1} single vertices attached directly to the stem, namely

$$\alpha = \sum_{i=0}^{n-1} n^i = \frac{n^n - 1}{n - 1} \approx n^{n-1}$$

Thus, the weight of that specific path on $\Gamma_d(\bar{G}, v_0)$ tends to 0 increasing n, since

$$\lim_{n \to \infty} \frac{n^{n+1}}{2^{n^{n-1}}} = 0$$

This does not prove that $\Gamma_d(\bar{G}, v_0)$ is linear but, at the same time, the presence of a single super linear path does not exclude this possibility. This example clarifies the main difference between the worst case and the average case. Let us formalize this concept. First we give the following definition.

Definition 1. *Given a finite undirected connected graph $G = (V, E)$, where the weight of each edge is equal to 1, a starting vertex $v_0 \in V$, and an exploration strategy $\mathscr{S}_x$, we define $N_h^{(x)}(G, v_0)$ as the sum of the probabilities for the exploration paths to be chosen by $\mathscr{S}_x$, such that their number of edge traversals is $\Omega(h(|V|))$.*

Now we are ready to enunciate the following theorem.

Theorem 7. *Consider a finite undirected connected graph $G = (V, E)$, where the weight of each edge is equal to 1, a starting vertex $v_0 \in V$, and an exploration strategy $\mathscr{S}_x$. For any function $f()$, if there exists a function $h()$ such that*

$$N_h^{(x)}(G, v_0) \cdot h(|V|) = \Omega(f(|V|))$$

then $\Gamma_x(G, v_0) = \Omega(f(|V|))$

Proof. By definition $\Gamma_x(G, v_0)$ is the sum of the lengths L_1, L_2, ..., L_n of all the possible n paths on G, starting from v_0, and according to $\mathscr{S}_x$. Each path length in the sum is weighted with the probability for the path to be chosen (according to $\mathscr{S}_x$). Namely

$$\Gamma_x(G, v_0) = p_1 \cdot L_1 + p_2 \cdot L_2 + ... + p_n \cdot L_n$$

Since the result we want to achieve is

$$\lim_{|V| \to \infty} \frac{\Gamma_x(G, v_0)}{f(|V|)} > 0$$

let us consider the following limit

$$\lim_{|V| \to \infty} \frac{p_1 \cdot L_1 + p_2 \cdot L_2 + ... + p_n \cdot L_n}{f(|V|)}$$

for all the L_i such that $L_i = \Omega(h(|V|))$ we substitute $h(|V|)$ obtaining a limit asymptotically lower than or equal to the previous one. Let us rename as $L_1, ..., L_{n'}$ all the n' paths L_i such that $L_i \neq \Omega(h(|V|))$

$$\lim_{|V| \to \infty} \left[\frac{p_1 \cdot L_1 + ... + p_{n'} \cdot L_{n'}}{f(|V|)} + \frac{N_h^{(x)}(G, v_0) \cdot h(|V|)}{f(|V|)} \right]$$

by hypothesis we have that

$$\lim_{|V| \to \infty} \frac{N_h^{(x)}(G, v_0) \cdot h(|V|)}{f(|V|)} > 0$$

thus, because $p_i \geq 0$ and $L_i \geq 0$

$$\lim_{|V|\to\infty} \left[\frac{p_1 \cdot L_1 + ... + p_{n'} \cdot L_{n'}}{f(|V|)} + \frac{N_h^{(x)}(G, v_0) \cdot h(|V|)}{f(|V|)} \right] > 0$$

and thus, because the following limit is greater than or equal to the previous one

$$\lim_{|V|\to\infty} \frac{p_1 \cdot L_1 + p_2 \cdot L_2 + ... + p_n \cdot L_n}{f(|V|)} > 0$$

since $\Gamma_x(G, v_0) = p_1 \cdot L_1 + p_2 \cdot L_2 + ... + p_n \cdot L_n$ we proved the theorem, as

$$\lim_{|V|\to\infty} \frac{\Gamma_x(G, v_0)}{f(|V|)} > 0$$

□

Basically, this theorem asserts that the order of a lower bound for $\Gamma()$, in general, does not depend on a single path length, but, possibly, by a set of paths of length $\Omega(h(|V|))$. More precisely, in $\Gamma()$, each path length is weighted by the probability for that path to be chosen by the strategy, thus, we are interested in the product between the sum of the probabilities for all the paths of the same length, and that length. Clearly, we are not interested in the specific length of a certain path, but in the order of that path length, so we exploited the concept of $\Omega()$ to achieve this result.

Now let us consider $\mathscr{S}_{dg}$. If Conjecture 1 is true, the information gain would not be a perfect tie breaker, since, as we saw in Proposition 1, in the worst case, $\mathscr{S}_{dg}$ is $\Omega\left(\frac{\log|V|-2\log(r+1)}{\log\log|V|}|V|\right)$. Moreover, in general, considering the average number of edge traversals, we can find some environments for which $\mathscr{S}_d$ performs better than $\mathscr{S}_{dg}$ and *vice versa*. For example, in Figure 5.1, if the robot is equipped with a sensor with range $r = \epsilon$ ($\epsilon \to 0$) and starts from the leftmost vertex v, it visits one of the vertices in the cycle for both $\mathscr{S}_d$ and $\mathscr{S}_{dg}$, and then reaches the vertex v' where there is a branch. Here, $\mathscr{S}_d$ has probability $\frac{1}{2}$ to go to the left and visit the last vertex in the cycle, while $\mathscr{S}_{dg}$ has probability 1 of visiting the right portion of the graph, visiting the last vertex in the cycle at the end of the process, thus re-traversing (almost) the whole graph.

5.2 Indoor Environments

To deal with the second source of randomness discussed at the beginning of this chapter, we have to define a set of possible graphs and a probability distribution over that set. So, we decided to focus our analysis considering some restricted classes of graphs that can model realistic indoor environments, on which we will

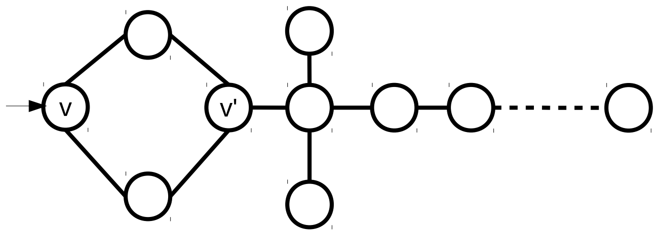

Figure 5.1: Example of graph where in the average case $\mathscr{S}_d$ performs better than $\mathscr{S}_{dg}$.

try to estimate the impact of the information gain on the exploration performance. Let us define vertex-labeled (finite undirected connected) graphs $G = (V, E)$ for which:

- some vertices $\mathscr{C} \subseteq V$ are labeled as 'corridor',
- the other vertices $\mathscr{R} \subseteq V$ are labeled as 'room' ($\mathscr{C} \cap \mathscr{R} = \emptyset$ and $\mathscr{C} \cup \mathscr{R} = V$),
- some connected sub-graphs $\mathscr{R}_i$ are defined on G, where all vertices are labeled as 'room' (in the following, with slight abuse of notation, we refer to $\mathscr{R}_i$ to indicate the set of vertices of such sub-graph; $\bigcup \mathscr{R}_i = \mathscr{R}$),
- $\mathscr{E} = \{v \in \mathscr{R} \mid \exists_{=1} w \in \mathscr{C} : (v, w) \in E\}$ are the room-type vertices that act as doorways between rooms and corridors; they are further labeled as 'entrance',
- $\neg \exists v \in \mathscr{R}, w \in \mathscr{C} : v \notin \mathscr{E} \wedge (v, w) \in \mathscr{C}$ (within a room only the entrance vertices can be attached to the corridors).

Basically, these labeled graphs are semantic maps of indoor environments that can be built autonomously by mobile robots [Mozos et al., 2005]. A realistic portion of an indoor environment and the corresponding labeled graph are shown in Figure 5.2.

Let us also define a cluster of rooms as a set of $\mathscr{R}_i$, whose entrance vertices are attached to the same corridor vertex. For example, in Figure 5.2, $\mathscr{R}_1$ and $\mathscr{R}_2$ compose a cluster.

More formally, a cluster of rooms of a given vertex $v \in \mathscr{C}$ is a set $K_v = \{\mathscr{R}_i \mid w \in \mathscr{R}_i \cap \mathscr{E}, (v, w) \in E\}$ (we call K the union of all non-empty clusters K_v). Moreover, let us define a function that we will use the next results.

Definition 2. *We call* placement function *a function $q : 2^{\mathscr{C}} \rightarrow \mathbb{N}$ such that, given a partition A of $\mathscr{C}$, then $|K| = \sum_{a \in A} q(a)$.*

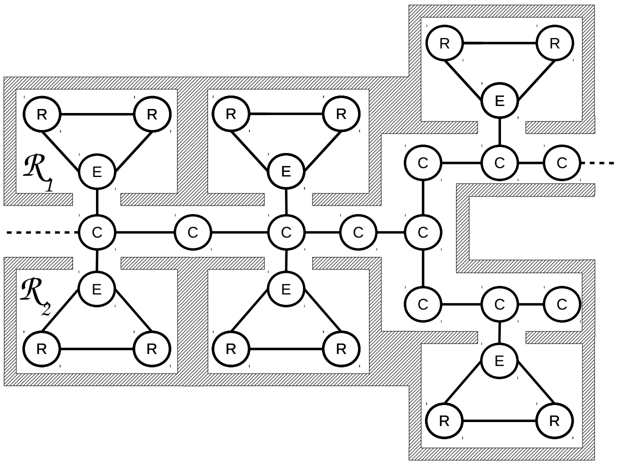

Figure 5.2: Example of graph with labels (C 'corridor'; E 'entrance'; R 'room').

Basically $q()$ is a function that, given a subset of vertices of $\mathscr{C}$, returns the number of clusters attached to vertices in that subset. In the next sections, we define some classes of graphs on the set of graphs labeled as above.

5.2.1 Tree Environments

In this section we will define and analyze environments that are basically trees of corridors with attached rooms. Each room is connected to the rest of the environment through a single entrance. More formally, we say that $G = (V, E)$ belongs to $\mathscr{G}_1$ if it satisfies the following properties:

Property 1 The subgraph on G induced by vertices in $\mathscr{C}$ is a tree, where each leaf is attached to a vertex v with $\delta(v) = 2$,

Property 2 $\forall \mathscr{R}_i : \exists_{=1} v \in \mathscr{R}_i \wedge v \in \mathscr{E}$,

Property 3 Given $\mathscr{R}_i$ and called v the entrance vertex $v \in \mathscr{R}_i \cap \mathscr{E}$, vertices $v' \in \mathscr{R}_i \setminus \mathscr{E}$ are at distance 1 from v,

Property 4 $\forall v \in \mathscr{E} : \delta(v) > 2$,

Property 5 $\forall v \in \mathscr{C} : \exists e \in \mathscr{E} : (v, e) \in E$
$\Rightarrow \forall w \in \mathscr{C} : (v, w) \in E \Rightarrow \delta(w) \leq 2$, (namely, given a corridor vertex v attached to an entrance vertex e, all the neighbor corridor vertices w should not have more than 1 other vertex, beyond v, attached to the them).

Given an arbitrary starting vertex $v_0 \in \mathscr{C}$ (the root) and the set of leafs $J = \{v \in \mathscr{C} \mid \delta(v) = 1\}$, we call $\mathscr{C}^j$ the shortest path from v_0 to the leaf $j \in J$ (the path is unique due to Property 1).

In the following result we estimate the difference between the average performance of $\mathscr{S}_d$ and $\mathscr{S}_{dg}$ on a set of possible graphs that belongs to $\mathscr{G}_1$.

Given a graph $G = (V, E) \in \mathscr{G}_1$ and a starting vertex $v_0 \in V \cap \mathscr{C}$, we define a set of graphs that have the same corridors structure (and the same corridors length) of G, the same number of clusters for each corridor $\mathscr{C}^j$, but different clusters position along the corridors, such that $q(N(v_0) \cup v0) = 0$ (where $N(v_0)$ returns the set of neighbor of v_0). We assume the probability distribution over this set as a uniform distribution. Since G could be any graph on that set, we average the difference between $\Gamma_d()$ and $\Gamma_{dg}()$ uniformly over all the clusters dispositions. Thus, we have the following proposition.

Proposition 2. *Consider a graph $G = (V, E)$ that belongs to $\mathscr{G}_1$, a starting vertex $v_0 \in V \cap \mathscr{C}$, a goal percentage $p = 1$, and a sensor range $r = \epsilon$ ($\epsilon \to 0$). An estimate for the difference between the average performance of $\mathscr{S}_d$ and $\mathscr{S}_{dg}$ on G is*

$$\sum_{j \in J} B^j \cdot (|\mathscr{C}^j| - 2)\frac{q(\mathscr{C}^j)}{q(\mathscr{C}^j) + 1} - 2\frac{|J| - q(J)}{|J|}$$

where B^j is the product of the inverse of the corridor branching factors encountered along $\mathscr{C}^j$ and $q()$ is a placement function.

Proof. Considering $\mathscr{S}_{dg}$, because of Properties 4 and 5, while the robot explores a corridor, it explores every room it encounters before continuing going along the corridor. A room, when chosen, is completely explored, due to Properties 2 and 3 and to the fact that the starting vertex $v_0 \in \mathscr{C}$. Also, Property 2 guarantees that after a room $\mathscr{R}_i$ has been explored, the robot does not directly go to another room $\mathscr{R}_l$ ($i \neq l$) without first going back to a vertex in $\mathscr{C}$. This is clearly visible in Figure 5.2: no matter which starting vertex is considered, $\mathscr{S}_{dg}$, while exploring the corridor, will explore first the encountered rooms (i.e., the probability of choosing a vertex in the corridor is 0 when there are rooms attached to the current vertex). Moreover, because of Property 3, the mean tour cost for visiting a room (starting from the corridor vertex where $v \in \mathscr{R}_i \cap \mathscr{E}$ is attached), $T(\mathscr{R}_i)$, is the same for $\mathscr{S}_{dg}$ and $\mathscr{S}_d$. Further, we have to account the difference between an exploration path that ends on the corridor or in a room. Thus, for $\mathscr{S}_{dg}$, averaging over the clusters position as a uniform distribution, we have

$$E[\Gamma_{dg}(G, v_0)] \approx \sum_{\mathscr{R}_i} T(\mathscr{R}_i) + \sum_{j \in J} B^j (|\mathscr{C}^j| + 2 \sum_{l \in J, l \neq j} |\mathscr{C}^l|) - 2\frac{q(J)}{|J|}$$

B^j is the product of the inverse of the corridor branching factors encountered along $\mathscr{C}^j$, and thus it is the probability of $\mathscr{C}^j$ to be explored at the very end of the exploration process.

Instead, for $\mathscr{S}_d$, while the robot explores a corridor, it could choose not to explore some encountered rooms. Hence, it could happen that, once the robot has reached a leaf, it should go back to explore rooms left behind (recall that corridor vertices form a tree because of Property 1). This can be seen in Figure 5.2, as $\mathscr{S}_d$, when the robot's current vertex is in the corridor and there are neighbor vertices labeled as 'room', has a probability strictly greater than 0 to choose a vertex in the corridor, possibly leaving a room as the last area to explore. The number of edge traversals to visit the rooms is the same as $\mathscr{S}_{dg}$ because of Property 3. Moreover, for the same reasons of $\mathscr{S}_{dg}$, when a room is chosen for the exploration, it is wholly explored. The only difference between $\mathscr{S}_d$ and $\mathscr{S}_{dg}$ is during the exploration of the last corridor $\mathscr{C}^j$. With $\mathscr{S}_d$ the robot could have to go back to some unexplored rooms left along the last $\mathscr{C}^j$ and could end the exploration in a room.

Defined $\mathscr{R}^j_{last}$ the unexplored room left along $\mathscr{C}^j$, which has the closest entrance vertex to the starting point v_0, let us call Δ^j the distance between $\mathscr{R}^j_{last}$ and the leaf j, and Λ^j the difference between a tour and a path to explore a room $\mathscr{R}_i$ on $\mathscr{C}^j$, on average, weighting each difference with the probability that room $\mathscr{R}_i$ is $\mathscr{R}^j_{last}$.

In summary, for $\mathscr{S}_d$, averaging over the clusters position, we have that

$$E[\Gamma_d(G, v_0)] \approx \sum_{\mathscr{R}_i} T(\mathscr{R}_i) + \sum_{j \in J} B^j(|\mathscr{C}^j| + \Delta^j - \Lambda^j + 2 \sum_{l \in J, l \neq j} |\mathscr{C}^l|)$$

There should be also the term -2 multiplied by the probability for $\mathscr{S}_d$ to end in a room, but it is neglected because is close to 0. The latter probability corresponds to the probability to explore all the cluster in $\mathscr{C}$ times the probability that a cluster is attached to a leaf j.

Note that Λ^j (the average difference between a tour and a path to explore a room attached to a corridor $\mathscr{C}^j$) is always equal to 2 because of Property 3. Having no a priori knowledge about the size of the clusters, we approximate the probability to completely explore a given cluster K_v of rooms (before to go ahead along the corridor) as a constant value p_K, which can be seen as the mean of the probabilities to completely explore each cluster K_v. Under this hypothesis, the probability of not exploring a number m of clusters K_v along corridor $\mathscr{C}^j$ is a binomial $S \sim \mathscr{B}(q(\mathscr{C}^j), 1 - p_K)$. We want to estimate Δ^j through an aleatory variable D such that $\Delta^j \approx E[D]$. D represents the distance traveled to reach $\mathscr{R}^j_{last}$ knowing that the number of clusters in $\mathscr{C}^j$ is $q(\mathscr{C}^j)$. Note that D depends on how many unexplored rooms are in $\mathscr{C}^j$, once the robot reaches j. Thus, to find $E[D]$ we are interested in $E[D \mid S]$. This conditional expectation can be approximated imagining the corridor as a continuous line and the relative position of the clusters as independent. The traveled distance to explore $S = m$ clusters left, is an aleatory variable $Z = \max(X_1, X_2, ..., X_m)$ where $X_v \sim U(0, |\mathscr{C}^j| - 2)$ (minus two because of the starting vertex constraint: $q(N(v_0) \cup v0) = 0$) is the position of the cluster K_v along $\mathscr{C}^j$. Thus, the cumulative distribution function of Z is:

$$F_Z(t) = P(Z \leq t)$$

(Because of the definition of Z)

$$= P(X_1 \leq t \wedge X_2 \leq t \wedge ... \wedge X_m \leq t)$$

(Since we assumed the uniform distributions as indipendent)

$$= P(X_1 \leq t) \cdot P(X_2 \leq t) \cdot ... \cdot P(X_m \leq t)$$

(Substituting the cumulative distribution function for a uniform distribution)

$$= \left(\frac{t}{|\mathscr{C}^j| - 2}\right)^m.$$

Now we can compute the probability density function:

$$f_Z(t) = \frac{dF_Z(t)}{dt} = m \cdot \frac{t^{m-1}}{(|\mathscr{C}^j| - 2)^m}$$

and applying the definition of the expected value for a continuous aleatory variable

$$\begin{aligned} E[D \mid S = m] = E[Z] \\ &= \int_0^{|\mathscr{C}^j|} m \cdot \frac{t^{m-1}}{(|\mathscr{C}^j| - 2)^m} \cdot t \cdot dt \\ &= \frac{m}{(|\mathscr{C}^j| - 2)^m} \int_0^{|\mathscr{C}^j|} t^m \cdot dt \\ &= \frac{m}{m+1}(|\mathscr{C}^j| - 2) \end{aligned}$$

In the hypothesis that $p_K < 1$, an estimate for the mean gain is:

$$E\left[\Gamma_d(G, v_0) - \Gamma_{dg}(G, v_0)\right] = E\left[\Gamma_d(G, v_0)\right] - E[\Gamma_{dg}(G, v_0)]$$

(Substituting the expected values)

$$\approx \sum_{j \in J} B^j \left(\Delta^j - \Lambda^j\right) + 2\frac{q(J)}{|J|}$$

(Splitting the sum)

$$= \sum_{j \in J} B^j \Delta^j - \sum_{j \in J} B^j \cdot 2 + 2\frac{q(J)}{|J|}$$

(Because $\Delta^j \approx E[D]$)

$$\approx \sum_{j \in J} B^j E\left[D\right] - 2 + 2\frac{q(J)}{|J|} \tag{5.2}$$

Thus, remembering that $\Delta^j \approx E[D] = E\left[E\left[D \mid S = m\right]\right]$ where $E\left[D \mid S = m\right]$ is the quantity found above and that $S \sim \mathscr{B}(q(\mathscr{C}^j), 1 - p_K)$, we have that (5.2) becomes

$$\approx \sum_{j \in J} B^j \left[\sum_{m=0}^{q(\mathscr{C}^j)} \binom{q(\mathscr{C}^j)}{m} \left(p_K^{q(\mathscr{C}^j)-m}(1-p_K)^m \cdot \frac{m}{m+1}(|\mathscr{C}^j| - 2)\right)\right] + \\ - 2\frac{|J| - q(J)}{|J|}$$

(Simplifying this known sum)

$$= \sum_{j \in J} B^j \left[\frac{q(\mathscr{C}^j)(1-p_K) - p_K(1-p_K^{q(\mathscr{C}^j)})}{(q(\mathscr{C}^j)+1)(1-p_K)}(|\mathscr{C}^j| - 2)\right] - 2\frac{|J| - q(J)}{|J|}$$

(Approximating $p_K(1 - p_K^{q(\mathscr{C}^j)}) \approx 0$)

$$\approx \sum_{j \in J} B^j \cdot (|\mathscr{C}^j| - 2)\frac{q(\mathscr{C}^j)}{q(\mathscr{C}^j)+1} - 2\frac{|J| - q(J)}{|J|}$$

□

Notice that the placement function can be any function that satisfies Definition 2. For instance, if the number of clusters depends only on the corridor length we can define a density parameter d_K such that $|K| \sim d_K|\mathscr{C}|$. The formula in Proposition 2 becomes

$$\sum_{j \in J} B^j \cdot (|\mathscr{C}^j| - 2)\frac{d_K(|\mathscr{C}^j| - 2)}{d_K(|\mathscr{C}^j| - 2) + 1} - 2(1 - d_K) \tag{5.3}$$

(Also in this case, the minus two are because of the starting vertex constraint: $q(N(v_0) \cup v0) = 0$)

In order to assess the validity of our estimate of Proposition 2, some simulated experiments have been conducted in randomly generated environments that

	$d_K = 0.2$		$d_K = 0.3$		$d_K = 0.4$	
$\lvert\mathscr{C}\rvert$	Gain	Error	Gain	Error	Gain	Error
50	13.8 (3.1)	-2.1 (3.8)	15.4 (1.9)	-1.0 (2.2)	16.9 (1.2)	0.1 (1.2)
100	14.7 (1.7)	0.2 (2.2)	15.9 (3.0)	-1.5 (3.3)	21.8 (1.3)	0.4 (1.3)
150	17.0 (2.9)	-3.0 (4.2)	28.8 (3.3)	0.4 (3.3)	63.3 (2.1)	0.5 (2.1)
200	66.1 (3.6)	-0.3 (3.7)	57.1 (3.6)	0.6 (3.6)	66.5 (3.8)	0.4 (3.8)
250	70.1 (4.1)	-0.1 (4.1)	76.7 (7.0)	-2.6 (7.4)	104.5 (5.5)	-2.4 (6.0)

Table 5.1: Performance on random generated $\mathscr{G}_1$ environments. The mean gain and its standard deviation, the mean error and its standard deviation with respect to 0 are reported.

belongs to $\mathscr{G}_1$. In particular we referred to (5.3), using a placement function which depends only on the corridors length, since we attached clusters randomly (and thus the probability to attach a certain cluster to a corridor is proportional to the corridor length). More precisely, each corridor tree has been generated randomly with a mean number of leaves equal to 5. Each cluster has a random number of rooms generated with a uniform probability between 1 and 4. Then the clusters have been attached to the tree randomly (preserving the properties). We varied the number of clusters $|K| \sim d_K|\mathscr{C}|$ as $d_K = \{0.2, 0.3, 0.4\}$. Since the shape of each single room does not influence the gain of $\mathscr{S}_{dg}$ over $\mathscr{S}_d$ (because of the properties) we decided to choose a full connected subgraph of three vertices each. Finally, the starting position has been chosen from those satisfying the constraint. An example of these random generated environments is given in Figure 5.3. The results of the simulations are shown in Table 5.1.

Our results (over 600 randomly generated graphs) actually suggest that $\mathscr{S}_d$ always performs worse than $\mathscr{S}_{dg}$, which is a predictable result, given the shape of the environment we considered. For example, considering $|V| = 150$ and a number of clusters $|K| \sim d_K|V|$, for $d_k = 0.4$ the difference between the mean traveled distances of $\mathscr{S}_d$ and $\mathscr{S}_{dg}$ is 62.78 (2.02 standard deviation) edge traversals. On average the gain seems to be almost independent of the number of rooms in each cluster. It seems just to depend on the way corridors are attached. This makes the estimate fairly good, since there is no assumption about the clusters size. The error between the estimate on the gain of Proposition 2 and the real one seems to be limited. We notice that the goodness of the estimate depends on the ratio between the cardinality of the set of leaves $|J|$ and that of corridor vertices $|\mathscr{C}|$. For example, in one of the experiments the error was -2.1 (3.8), over a gain of 13.8 (3.1), for $|\mathscr{C}| = 50$, $|J| = 5$, and $d_K = 0.2$. This could be explained by the fact that if $|J|/|\mathscr{C}|$ is high (in the example it is 0.1), the gain of $\mathscr{S}_{dg}$ over $\mathscr{S}_d$ is smaller and, because of the approximations in the computation of the estimate (shown in the

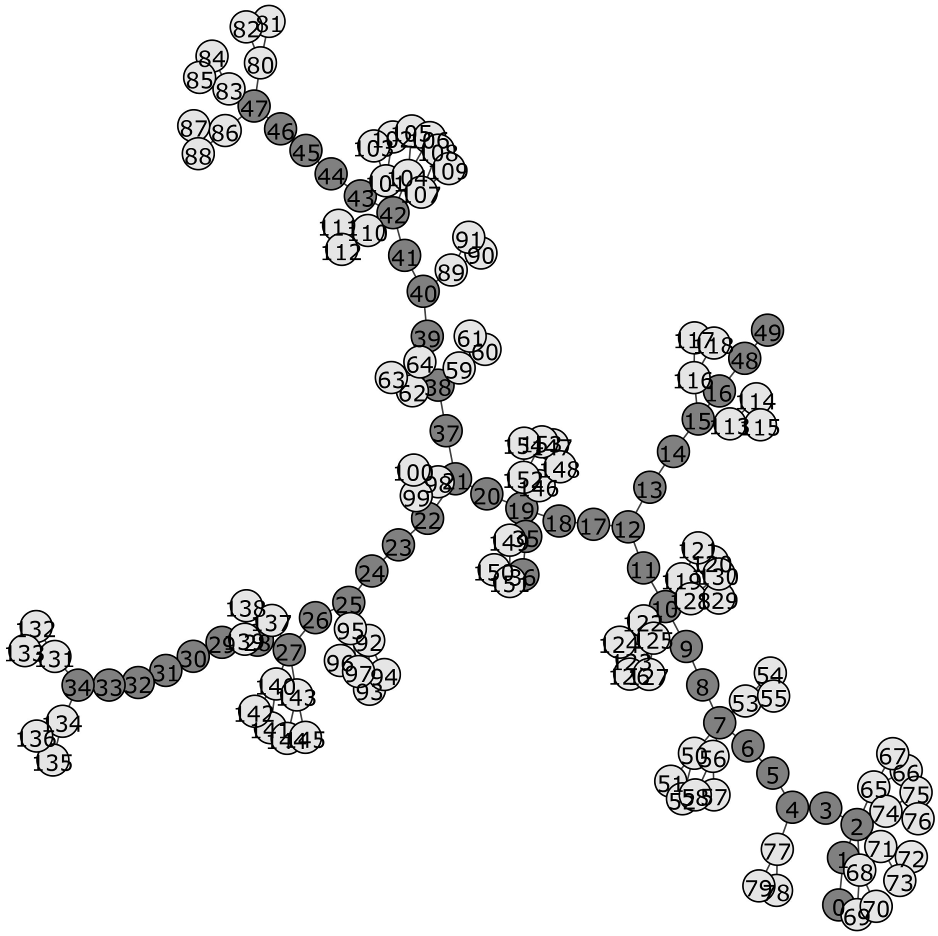

Figure 5.3: An example of random $\mathscr{G}_1$ graph with room (light grey) and corridor (dark grey) vertices and $|\mathscr{C}| = 50$, $d_K = 0.3$.

proof), the percentual error is higher. Intuitively, corridors are generally shorter when the ratio $|J|/|\mathscr{C}|$ is higher and so $\mathscr{S}_d$ has to re-traverse less vertices to visit unexplored rooms. Instead, considering $|\mathscr{C}| = 200$ (with $|J| = 6$ and $d_K = 0.2$), $|J|/|\mathscr{C}| = 0.03$ and the error percentage is smaller (namely, -0.3 (3.7) over a mean gain of 66.1 (3.6)). Moreover, we noted that there is a slight overestimate, probably due to the approximations we made in the proof which, anyway, seems not to grow with the gain nor with the number of corridor vertices.

In general, the gain estimate given in Proposition 2 can be seen as the gain, in terms of edge traversals, that $\mathscr{S}_d$ would have if, at each step i, it prefers to choose room locations than corridor locations from $F^d_{i_{min}}$. Thus, this result is valid for a wider class of functions that act as tie breaker than $g()$ (the information gain), such as those that exploit room landmarks or semantic inference, or, in general, for any tie breaker that can recognize a room frontier location with respect to a corridor frontier location (and then decides to break ties according to this knowledge, choosing first to explore the room locations). Moreover, we can notice that, on average, the $\mathscr{S}_d$ exploration path tends to be a tour over the corridor vertices. This happens on a larger class of environments than $\mathscr{G}_1$, that we call $\mathscr{G}_1'$, which is characterized by Property 1, Property 2, and Property 3. It is easy to check that Property 4 and Property 5 (which could be reasonable for certain kinds of indoor environments) are needed to guarantee that $\mathscr{S}_{dg}$ explores all the rooms as soon as possible. Thus, given a graph $G = (V, E)$ that belongs to $\mathscr{G}_1'$, for any starting vertex $v_0 \in \mathscr{C}$, the $\mathscr{S}_d$ exploration path tends to be a tour over the corridor vertices. This means that, on average, the last room that $\mathscr{S}_d$ explores is often close to the starting vertex.
To show better this fact, let us define a version of $\Gamma_d()$ function that does not take into account the edge traversals due to the rooms exploration, as

$$\Gamma_d^{\mathscr{C}}(G, v_0) = \sum_{j\in J} B^j \left(|\mathscr{C}^j| + \Delta^j + 2 \sum_{l\in J, l\neq j} |\mathscr{C}^l| \right)$$

(Including j in the sum)

$$= \sum_{j\in J} B^j \left(2 \sum_{l\in J} \left(|\mathscr{C}^l|\right) - \left(|\mathscr{C}^j| - \Delta^j\right) \right)$$

(Splitting the sum)

$$= 2 \sum_{j\in J} B^j \sum_{l\in J} \left(|\mathscr{C}^l|\right) - \sum_{j\in J} B^j \left(|\mathscr{C}^j| - \Delta^j\right)$$

(The first two sums are indipendent of each other)

$$= 2 \sum_{l\in J} \left(|\mathscr{C}^l|\right) \sum_{j\in J} B^j - \sum_{j\in J} B^j \left(|\mathscr{C}^j| - \Delta^j\right)$$

(Since the sum of all the B^j is equal to 1)

$$= 2\sum_{l\in J}\left(|\mathscr{C}^l|\right) - \sum_{j\in J} B^j \left(|\mathscr{C}^j| - \Delta^j\right)$$

where Δ^j is still the distance between $\mathscr{R}^j_{last}$ and the leaf j.
If $T(G, v_0) = 2\sum_{j\in J}|\mathscr{C}^j|$ is the length of tour over the subgraph on G induced by vertices in $\mathscr{C}$, then (averaging, again, over all the clusters positions) we have

$$\begin{aligned} E[T(G,v_0) - \Gamma_d^{\mathscr{C}}(G,v_0)] &= E[T(G,v_0)] - E[\Gamma_d^{\mathscr{C}}(G,v_0)] \\ &= \sum_{j\in J} B^j \left(|\mathscr{C}^j| - \Delta^j\right) \\ &\approx \sum_{j\in J} B^j \left(|\mathscr{C}^j| - |\mathscr{C}^j|\frac{q(\mathscr{C}^j)}{q(\mathscr{C}^j)+1}\right) \\ &= \sum_{j\in J} B^j \cdot |\mathscr{C}^j| \left(1 - \frac{q(\mathscr{C}^j)}{q(\mathscr{C}^j)+1}\right) \end{aligned}$$

As before, if we can define a d_K in the environment, we have

$$\sum_{j\in J} B^j|\mathscr{C}^j|\left(1 - \frac{d_K|\mathscr{C}^j|}{d_K|\mathscr{C}^j|+1}\right)$$

which tends to 0 for any sufficiently large d_K. This is clear from Table 5.2 which shows the number of edge traversals over the corridors in $\mathscr{G}_1$ random graphs, reporting the error and its standard deviation with respect to 0. Since we considered $d_k \geq 0.2$, the exploration path, on average, is always almost a tour over the corridor vertices. Moreover, increasing the density parameter d_K, the number of edge traversals between corridor vertices is slightly closer to $2|\mathscr{C}^j|$. The only exception is in correspondence of $|\mathscr{C}^j| = 50$, where the traveled distance is higher for $d_K = 0.2$ than for $d_K = 0.3$. We can explain this fact considering that, such distance it does not depend only on the number of clusters in the environment, but also to the corridor structures. In any case, the difference on the traveled distance from $d_K = 0.2$ and $d_K = 0.4$ is rather small (for instance, with $|\mathscr{C}^j| = 250$ the difference is just 3.7 over a traveled distance that starts from 489.9 (3.1) for $d_K = 0.2$ to 493.6 (1.1) for $d_K = 0.4$. Furthermore, we can notice that the standard deviation of the real traveled distance is lower for low values of d_K, which was predictable, since, increasing the number of cluster, the freedom on the way to attach the clusters diminishes.

5.2.2 Simple Loop Environments

Now let us consider another class of indoor environments characterized by a simple loop, which represents a corridor and to which some rooms are attached. More

	$d_K = 0.2$		$d_K = 0.3$		$d_K = 0.4$	
$\lvert\mathscr{C}\rvert$	Distance	Error	Distance	Error	Distance	Error
50	93.3 (0.9)	-1.0 (1.7)	92.2 (1.1)	-1.9 (2.3)	94.1 (0.6)	-0.2 (1.2)
100	190.1 (3.5)	-2.6 (4.3)	191.8 (2.1)	-2.2 (4.1)	194.2 (0.7)	-1.2 (1.2)
150	289.2 (5.0)	-2.1 (7.7)	293.2 (1.5)	-0.1 (3.5)	292.7 (1.2)	-0.9 (2.7)
200	390.4 (2.3)	-1.8 (2.8)	392.3 (2.7)	-2.2 (6.4)	396.2 (1.4)	1.0 (4.8)
250	489.9 (3.1)	-4.0 (6.8)	492.5 (1.7)	-2.4 (6.1)	493.6 (1.1)	0.9 (3.8)

Table 5.2: Number of edge traversals between corridor vertices on graphs belonging to $\mathscr{G}_1$. The mean distance traveled over the corridors and its standard deviation, the mean error and its standard deviation with respect to 0 are reported.

formally we say that $G = (V, E)$ belongs to $\mathscr{G}_2$ if satisfies four of the previous five properties (already defined in Section 5.2.1 and which are summarized here):

- Property 2 (There exists at most one entrance for each room),
- Property 3 (All the vertices of a given room are at most at distance 1 from the entrance),
- Property 4 (The degree of each entrance is greater than 2),
- Property 5 (Given a corridor vertex v attached to an entrance vertex e, all the neighbor corridor vertices w should not have more than 1 other vertex, beyond v, attached to them),

and the following new properties

Property 6 The subgraph on G induced by vertices in $\mathscr{C}$ is a loop,

Property 7 Each cluster is composed by at most one room.

As for $\mathscr{G}_1$, in the following result we estimate the difference between the average performance of $\mathscr{S}_d$ and $\mathscr{S}_{dg}$ on a set of possible graphs that belong to $\mathscr{G}_2$. As in Section 5.2.1, given a graph $G \in \mathscr{G}_1$ we define the set of graphs that have the same corridor structure (namely, the same loop length) of G, but different clusters position. We assume again the probability distribution over this set as a uniform distribution.

Because of Property 7 the number of clusters coincides with the number of rooms. So, for clarity, let us rename it as $R = |K|$.

Proposition 3. *Consider a graph $G = (V, E)$ that belongs to $\mathscr{G}_2$, a starting vertex $v_0 \in V \cap \mathscr{C}$, a goal percentage $p = 1$, and a sensor range $r = \epsilon$ ($\epsilon \to 0$). An* *estimate for the mean difference between the average performance of $\mathscr{S}_d$ and $\mathscr{S}_{dg}$ on G is*

$$|\mathscr{C}|\left[1-2^{-R}\left(\frac{3}{11}R^2+\frac{21}{44}R+1+\frac{e}{2}\sum_{i=3}^{R}\binom{R}{i}\frac{1}{i}\right)\right]$$

Proof. Considering $\mathscr{S}_{dg}$, again, because of Properties 4 and 5, while the robot explores the corridor loop, it explores every room it encounters before continuing going along the corridor and a room, when chosen, is completely explored, due to Properties 2 and 3 and to the fact that the starting vertex $v_0 \in \mathscr{C}$. Also, Property 2 guarantees that after a room $\mathscr{R}_i$ has been explored, the robot does not directly go to another room $\mathscr{R}_l$ ($i \neq l$) without first going back to a vertex in $\mathscr{C}$. Also here, because of Property 3, the mean tour cost for visiting a room (starting from the corridor vertex where $v \in \mathscr{R}_i \cap \mathscr{E}$ is attached), $T(\mathscr{R}_i)$, is the same for $\mathscr{S}_{dg}$ and $\mathscr{S}_d$. Thus, for $\mathscr{S}_{dg}$, averaging over the rooms positions, we have that

$$E[\Gamma_{dg}(G,v_0)] \approx \sum_{\mathscr{R}_i}[T(\mathscr{R}_i)] + |\mathscr{C}|$$

To be precise we should take account of the difference, in terms of edge traversals, between an exploration path that ends on the loop or in a room, but we neglect it (and also the minus one in $|\mathscr{C}|$, due to the fact that the last edge in the loop is not traversed). For $\mathscr{S}_d$, as we saw in the proof of Proposition 2, while the robot explores a corridor, it could choose not to explore some encountered rooms. Hence, it could happen that, once the robot has completely explored the loop, it should go back to explore rooms left behind. The number of edge traversals to visit the rooms is the same as $\mathscr{S}_{dg}$ because of Property 3. The only difference between $\mathscr{S}_d$ and $\mathscr{S}_{dg}$ is that, after the first loop traversal, $\mathscr{S}_d$ could have to re-traverse part of the loop to explore the rooms left.

We define Δ as the mean distance traveled during the second loop traversal, plus $|\mathscr{C}|$ (the distance traveled at the first loop traversal). For $\mathscr{S}_d$, averaging over the rooms positions, we have

$$E[\Gamma_d(G,v_0)] \approx \sum_{\mathscr{R}_i}[T(\mathscr{R}_i)] + \left(\frac{1}{2}\right)^R |\mathscr{C}| + \Delta$$

Also in this case we neglected the difference between an exploration path that ends on the loop or in a room and the minus one in $|\mathscr{C}|$. Notice that, because of Proposition 7, the probability to leave back a room during the first loop traversal is $\frac{1}{2}$. Similarly to the case of the tree, the probability of not exploring a number m of rooms along the corridor loop is a binomial $S \sim \mathscr{B}(m, \frac{1}{2})$. In general, Δ is rather hard to determine, even if we relax the constraints about the rooms positions. For

$m = 1$ and $m = 2$, Δ can be found enumerating all the possible rooms positions, and it is approximately $\left(\frac{1}{4}+1\right)|\mathscr{C}|$ and $(\frac{5}{11}+1)|\mathscr{C}|$, respectively. For $m \geq 3$ we proceed as follows. First we assume the loop as a continuous line and the rooms' positions as independent (as in the tree case), and then we find the mean distance from the closest unexplored room as $Z_m = \min(X_1, X_2, ..., X_m)$ where $X_i \sim U(0, \frac{|\mathscr{C}|}{2})$ is the distance, from the current position to the corridor vertex where the entrance of an unexplored room is attached. Thus, let be $l = \frac{|\mathscr{C}|}{2}$. The cumulative distribution function of Z is

$$F_Z(t) = P(Z \leq t)$$

(Basic probability theorem)

$$= 1 - P(Z > t)$$

(Because of the definition of Z)

$$= 1 - P(X_1 > t \wedge X_2 > t \wedge ... \wedge X_m > t)$$

(Since we assumed the uniform distributions as indipendent)

$$= 1 - P(X_1 > t) \cdot P(X_2 > t) \cdot ... \cdot P(X_m > t)$$

(Substituting the cumulative distribution function for a uniform distribution)

$$= 1 - \left(\frac{l-t}{l}\right)^m$$

The probability density function is

$$f_Z(t) = \frac{dF_Z(t)}{dt} = m \cdot \frac{(l-t)^{m-1}}{l^m}$$

and then, applying the definition of the expected value for a continuous aleatory variable

$$E[Z] = \int_0^l m \cdot \frac{(l-t)^{m-1}}{l^m} \cdot t \cdot dt$$
$$= \frac{m}{l^m} \int_0^l (l-t)^{m-1} \cdot t \cdot dt$$
$$= \frac{l}{m+1}$$

(Substituting l)

$$= \frac{|\mathscr{C}|}{2(m+1)}$$

Now, a rough approximation for the traveled distance is given considering

$$E[D \mid S = m] = |\mathscr{C}| + E[Z_m] + E[Z_{m-2}] + E[Z_{m-3}] + ...$$
$$\approx |\mathscr{C}| + E[Z_{m-1}] + E[Z_{m-2}] + E[Z_{m-3}] + ... \quad (5.4)$$

This sum is not valid when it is larger than $\frac{|\mathscr{C}|}{2}$, since the uniform distributions of the distance, from the current position to the corridor vertex where the entrance of an unexplored room is attached, cannot be considered between 0 and $\frac{|\mathscr{C}|}{2}$ anymore. To avoid this issue we can consider that, if the robot has covered half of the loop, the remaining distance that it has to travel, is the distance from the furthest room not explored yet. Thus, once the robot reaches the half of the loop, given α unexplored rooms, the remaining distance can be estimated as $Z' = \max(X_1, X_2, ..., X_\alpha)$, similarly to what we did in the proof of Proposition 2. We have to find when (5.4) reaches $\frac{|\mathscr{C}|}{2}$, namely, we have to find the number of rooms α such that

$$\sum_{i=\alpha}^{m-1} E[Z_i] = \frac{|\mathscr{C}|}{2}$$

(substituting $E[Z_i]$)

$$\sum_{i=\alpha}^{m-1} \frac{|\mathscr{C}|}{2(i+1)} = \frac{|\mathscr{C}|}{2}$$

(exploiting the limit approximation)

$$\frac{|\mathscr{C}|}{2} (\ln m - \ln(\alpha + 1)) = \frac{|\mathscr{C}|}{2}$$

(after some math)

$$a = \frac{m}{e} - 1$$

Thus, for $m \geq 3$

$$E[D \mid S = m] \approx \frac{|\mathscr{C}|}{2} + \frac{\alpha}{\alpha + 1} \frac{|\mathscr{C}|}{2}$$
$$= |\mathscr{C}| + \frac{|\mathscr{C}|}{2} + \frac{m - e}{m} \frac{|\mathscr{C}|}{2}$$
$$= \frac{4m - e}{2m} |\mathscr{C}|$$

The goodness of this approximation is shown in Figure 5.4 and Figure 5.5.
In summary we have that

$$E[\Gamma_d(G, v_0) - \Gamma_{dg}(G, v_0)] = E[\Gamma_d(G, v_0)] - E[\Gamma_{dg}(G, v_0)]$$

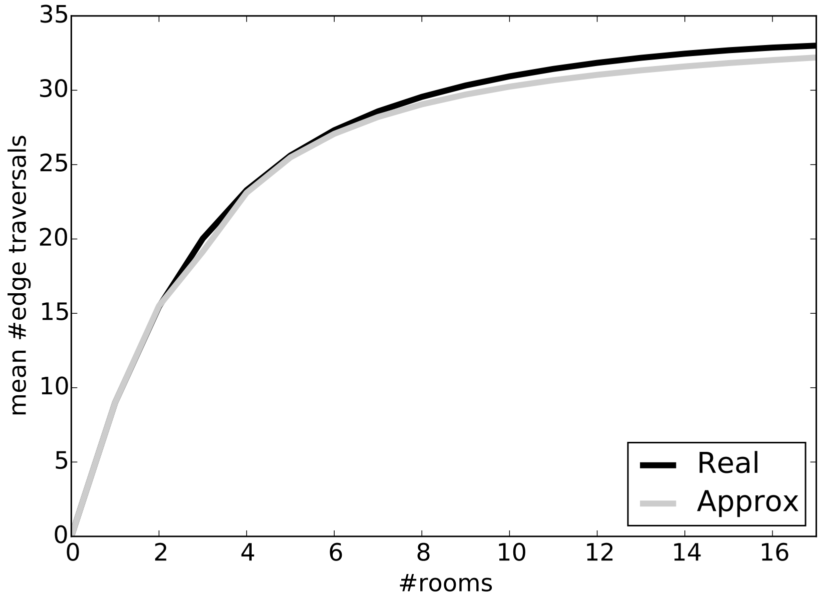

Figure 5.4: The real mean distance traveled, computed enumerating all the rooms configuration ($\{0, \cdots, 17\}$ *rooms) with* $|\mathscr{C}| = 35$

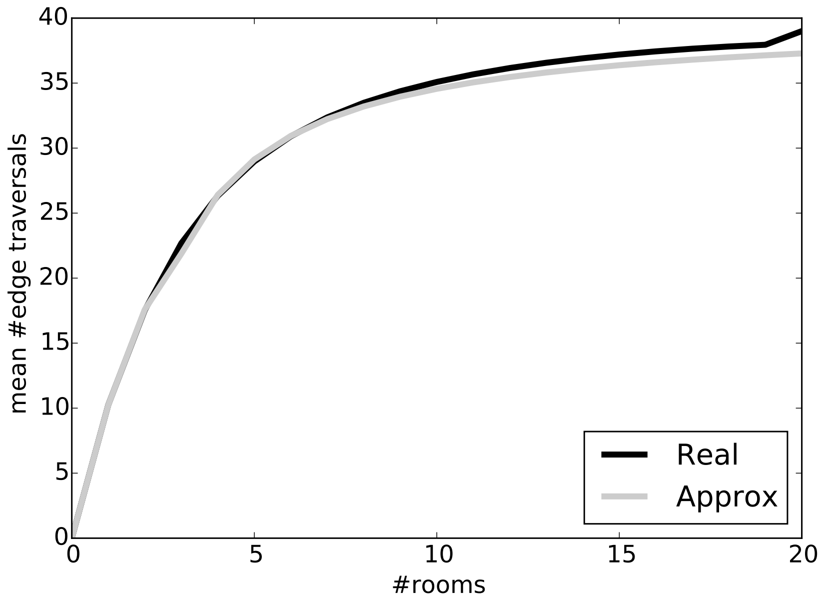

Figure 5.5: The real mean distance traveled, computed enumerating all the rooms configuration ($\{0, \cdots, 20\}$ *rooms) with* $|\mathscr{C}| = 40$

	$d_R = 0.2$		$d_R = 0.3$		$d_R = 0.4$	
$\|\mathscr{C}\|$	Gain	Error	Gain	Error	Gain	Error
50	32.1 (1.1)	-2.8 (3.1)	37.6 (1.3)	-2.6 (2.9)	40.4 (1.1)	-2.4 (2.6)
100	80.8 (2.0)	-4.7 (5.2)	87.7 (1.7)	-2.9 (3.3)	90.8 (0.9)	-2.2 (2.4)
150	132.1 (2.4)	-3.8 (4.5)	138.0 (2.1)	-2.7 (3.4)	140.8 (0.9)	-2.3 (2.5)
200	180.3 (3.3)	-5.7 (6.6)	186.9 (0.9)	-3.8 (4.0)	191.0 (2.8)	-2.2 (2.4)
250	231.4 (3.5)	-4.7 (5.8)	236.6 (2.0)	-4.2 (4.6)	242.7 (3.4)	-1.7 (2.1)

Table 5.3: Performance on random generated $\mathscr{G}_2$ environments. The mean gain and its standard deviation, the mean error and its standard deviation with respect to 0 are reported.

which is

$$= \left(\frac{1}{2}\right)^R |\mathscr{C}| + \Delta - |\mathscr{C}|$$

(Substituting $\Delta \approx E[D]$)

$$\approx E[D] - \left(1 + \left(\frac{1}{2}\right)^R\right) |\mathscr{C}|$$

(Substituting $E[D]$)

$$= |\mathscr{C}| \left(\frac{1}{2}\right)^R \left[\frac{5}{4}\binom{R}{1} + \frac{16}{11}\binom{R}{2} + \sum_{i=3}^{R} \binom{R}{i} \frac{4i - e}{2i}\right] - \left(1 + \left(\frac{1}{2}\right)^R\right) |\mathscr{C}|$$

(Simplifyng the terms)

$$= |\mathscr{C}| \left(\frac{1}{2}\right)^R \left[\frac{8}{11}R^2 + \frac{23}{44}R + 1 + \sum_{i=3}^{R} \binom{R}{i} \frac{4i - e}{2i}\right] - |\mathscr{C}|$$

(Splitting the sum)

$$= |\mathscr{C}| - |\mathscr{C}| \left(\frac{1}{2}\right)^R \left[\frac{3}{11}R^2 + \frac{21}{44}R + 1 + \frac{e}{2}\sum_{i=3}^{R} \binom{R}{i} \frac{1}{i}\right]$$

$$= |\mathscr{C}| \left[1 - \left(\frac{1}{2}\right)^R \left(\frac{3}{11}R^2 + \frac{21}{44}R + 1 + \frac{e}{2}\sum_{i=3}^{R} \binom{R}{i} \frac{1}{i}\right)\right]$$

□

Also in this case we performed some simulated experiments generating random graphs belonging to $\mathscr{G}_2$. We varied the room density parameter d_R (number

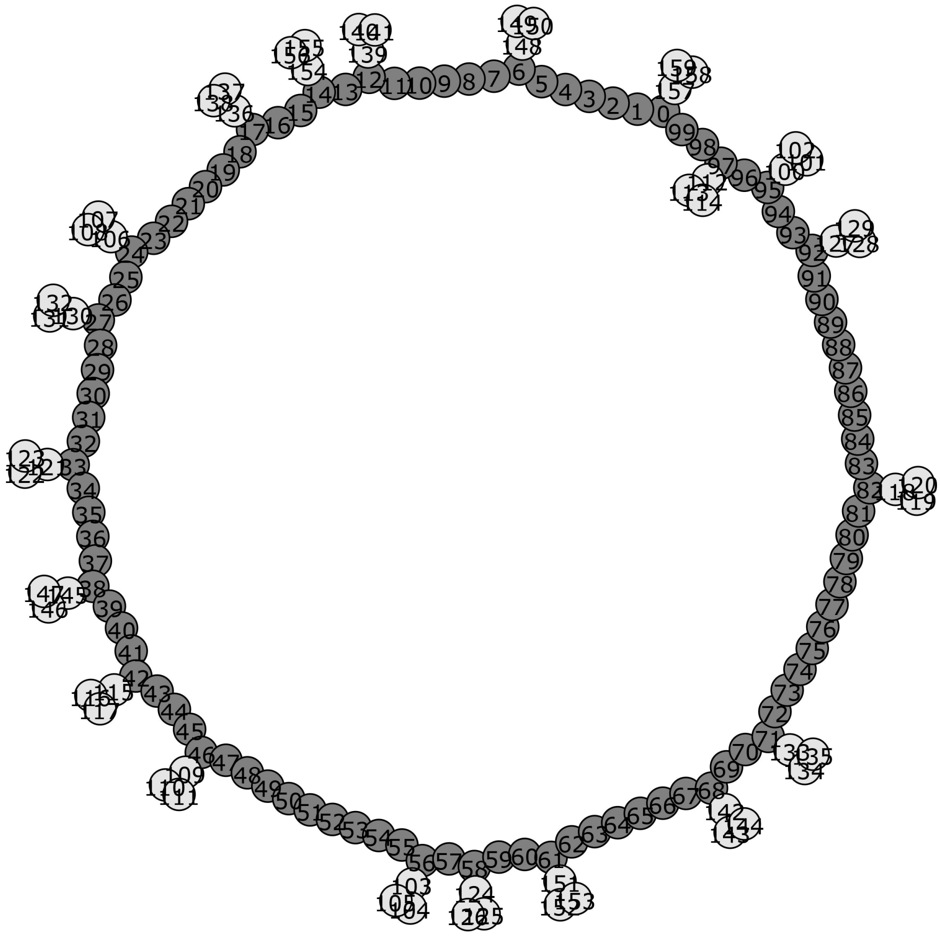

Figure 5.6: An example of random $\mathscr{G}_2$ graph with room (light grey) and corridor (dark grey) vertices and $|\mathscr{C}| = 100$, $d_K = 0.2$.

of rooms over the number of corridor vertices) and the number of corridor vertices. An example of the random generated graphs that belongs to $\mathscr{G}_2$ is given in Figure 5.6. The results are reported in Table 5.3. We can notice that, as for Proposition 2, there is an evident overestimate (and maybe a little bias) probably due to the approximations we made. Nevertheless, increasing the loop length we have that the percentage error lower significantly and the estimate seems to scale very well also in terms of standard deviation.

If we consider a larger class of environments $\mathscr{G}_2'$ characterized by Property 2, Property 3, Property 4, Property 5, and Property 6 (namely, like $\mathscr{G}_2$ but each cluster can be composed by more than one room), then, the formula of Proposition 3 is an estimate of the lower bound of the mean gain for any graph that belongs to $\mathscr{G}_2'$ (recall that $R = |K|$ is the number of clusters). To prove this fact consider the probability to leave back an unexplored cluster at the first loop traversal. This probability is trivially $\frac{1}{2}$ for any graph that belongs to $\mathscr{G}_2$. Instead, for a graph that belongs to $\mathscr{G}_2'$ this probability is greater than or equal to $\frac{1}{2}$: each cluster K_v is composed by at least one room, hence, once the robot reaches v, the probability to go ahead along the loop, leaving back at least one unexplored room, is greater than or equal to $\frac{1}{2}$. Thus, for $\mathscr{S}_d$, and for equal number of clusters, the mean number of clusters left at the first loop traversal is higher for a graph that belongs to $\mathscr{G}_2'$ than a graphs that belongs to $\mathscr{G}_2$. Since, at the second loop traversal, the mean traveled distance is a monotonic function with respect to the number of rooms left (as we can notice in the proof of Proposition 3), the formula of Proposition 3 is an estimate of the lower bound of the mean gain for any graph that belongs to $\mathscr{G}_2'$.

This result is generalizable, as for the tree environments in Section 5.2.1, for a wider class of functions that act as tie breaker than $g()$ (the information gain), namely, for any tie breaker that can distinguish a room frontier location from a corridor frontier location (and then decides to break ties according to this knowledge, choosing first to explore the room locations).

Chapter 6

Conclusions

We have proposed a model for the graph exploration problem that extends the fixed graph scenario, considering a robot equipped with a sensor able to perceive vertices within a generic range r, and that adopts a termination criterion based on the percentage p of the graph to perceive. Furthermore we introduced the concept of information gain in exploration strategies operating on graphs. This allows to approach a formal analysis of the performance of the strategies that exploit this concept, introduced and evaluated only experimentally (see, e.g., [Amigoni, 2008]) so far.

We provided worst-case bounds for three exploration strategies: $\mathscr{S}_d$, which, at a given time step, selects the frontier that minimizes its distance from the current robot position, $\mathscr{S}_g$, which, at a given time step, selects the frontier that maximizes the information gain, and $\mathscr{S}_{dg}$ which acts as $\mathscr{S}_d$ and breaks ties as $\mathscr{S}_g$. The derived bounds explicitly take into account not only the number of vertices of a graph to explore as currently done in the literature ([Tovey and Koenig, 2003] and [Koenig et al., 2001]), but also the value of the sensor range r. Moreover, the number of vertices in the bounds can be any fraction of the total vertices in the environment, according to the goal percentage p.

The main motivation of this work is that it integrates and possibly better explains the experimental results obtained with real (and realistically simulated) exploring robots. For example, according to some results obtained in specific environments, including information gain in the exploration strategies does not shorten the paths for completely exploring the environments [Julia et al., 2012, Stachniss and Burgard, 2003]. Moreover, in the literature, it is shown that, in exploration, increasing sensor range reduces the traveled distance ([Quattrini Li et al., 2012]). Our analysis confirms that finding in the worst case. Furthermore, for $\mathscr{S}_{dg}$, we have proved that the worst-case bounds are the same of those of $\mathscr{S}_d$. Practical experiments on random generated graphs have shown that, for a sensor range $r \geq 1$, using information gain as tie breaker is sufficient to improve the exploration performance, in terms of traveled distance, with respect to the plain distance criterion.

To the best of our knowledge, ours is the first work that analyzes exploration strategies in the average case. In particular, we identify two specific classes of graphs modeling indoor environments, where we calculate the average reduction in the traveled distance, that information gain can provide over considering only distance, when the sensor range is $r = \epsilon$ ($\epsilon \longrightarrow 0$).

The first class of indoor environments we considered is a tree of corridors with rooms. We defined this class of graphs through five properties. Then we removed some of them considering a larger class of environments, showing that, the exploration path of $\mathscr{S}_d$, tends to be a tour over the corridor vertices. Finally, we generalized the results to a wider class of tie breakers, namely, those that are able to distinguish a frontier room location from a frontier corridor location. The experimental results in some randomly generated graphs of the above class corroborate the average-case analysis.

The second class of indoor environments we considered is characterized by a simple loop corridor, to which some rooms are attached. We showed that the estimate found in Proposition 3, between the average performance of $\mathscr{S}_d$ and $\mathscr{S}_{dg}$, is a lower bound for a larger class of graphs. Also in this case, we performed experiments, which indicate that the approximation of the mean traveled distance we provided is very close to the real mean (computed enumerating all the feasible rooms dispositions). Then, we tested the goodness of the estimate on random generated graphs belonging to this class of loop environments. Even if there is a slight overestimate in the approximation, it is clear from the results that the error percentage diminishes, increasing the number of corridor vertices (namely, the loop length). Finally, similarly to what we did for the first class of the indoor environments, we generalized this result for a wider class of tie breakers (again, those that are able to distinguish a frontier room location from a frontier corridor location).

There are several directions of interest for future works. One could to extend the average-case analysis by considering more general class of graphs and perception radii $r \geq 0$.

About the worst-case analysis, our results could be generalized for weighted graphs by modifying edges whose weight is greater than 1 with a line graph in such a way that the weight of each edge of the line graph is less or equal than one. However, applying the worst-case bound on such modified graph is loose, so it could be interesting to find a lower worst-case bound.

Moreover, we suggest a conjecture (Section 5.1) about the perfect tie breaking of $\mathscr{S}_d$, that would be interesting to prove. The implication of that conjecture can strengthen the motivations behind the analyses on how and which heuristics could be used to break ties.

Furthermore we introduced a brief comparison between the average-case and the worst-case complexities in robotic exploration and, along this line, it is worth to deepen whether, in the fixed graph scenario, the average complexity of $\mathscr{S}_d$, on a given graph (that we measured through $\Gamma_d()$), is $\Omega(\frac{\log n}{\log \log n} n)$ as well.

Finally, it is worth to analyze the impact that uncertainty on perception, locomotion, and information gain evaluation has on the worst and average case. Moreover, it could be interesting to address other cases, like constrained exploration [Duncan et al., 2006], in which a robot should, for example, stay within a certain distance from a base station, and use of multiple robots [Brass et al., 2011].

In a broader view, the analysis done in this thesis can provide some insights for defining new exploration strategies and the corresponding bounds so that robotic exploration is improved.

Bibliography

S. Albers, K. Kursawe, and S. Schuierer. Exploring unknown environments with obstacles. *Algorithmica*, 32:123–143, 2002.

F. Amigoni. Experimental evaluation of some exploration strategies for mobile robots. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 2818–2823, 2008.

F. Amigoni and V. Caglioti. An information-based exploration strategy for environment mapping with mobile robots. *Robotics and Autonomous Systems*, 5(58): 684–699, 2010.

F. Amigoni, A. Quattrini Li, and D. Holz. Evaluating the impact of perception and decision timing on autonomous robotic exploration. In *Proceedings of the European Conference on Mobile Robots (ECMR)*, pages 68–73, 2013.

Y. Asahiro, E. Miyano, S. Miyazaki, and T. Yoshimuta. Weighted nearest neighbor algorithms for the graph exploration problem on cycles. *Information Processing Letters*, 110:93–98, 2010.

B. Awerbuch, M. Betke, R. Rivest, and M. Singh. Piecemeal graph exploration by a mobile robot. *Information and Computation*, 152(2):155–172, 1999.

P. Brass, F. Cabrera-Mora, A. Gasparri, and J. Xiao. Multirobot tree and graph exploration. *IEEE Transactions on Robotics*, 27(4):707–717, 2011.

D. Calisi, A. Farinelli, L. Iocchi, and D. Nardi. Autonomous navigation and exploration in a rescue environment. In *Proceedings of the IEEE International Workshop on Safety, Security, and Rescue Robotics (SSRR)*, pages 54–59, 2005.

H. Choset. Coverage for robotics: A survey of recent results. *Annals of Mathematics and Artificial Intelligence*, 31(1-4):113–126, 2001.

S. Das, P. Flocchini, S. Kutten, A. Nayak, and N. Santoro. Map construction of unknown graphs by multiple agents. *Theoretical Computer Science*, 385(1–3): 34–48, 2007.

X. Deng and C. Papadimitriou. Exploring an unknown graph. *Journal of Graph Theory*, 32(3):265–297, 1999.

X. Deng, T. Kameda, and C. Papadimitriou. How to learn an unknown environment I: The rectilinear case. *Journal of the ACM*, 45(2):215–245, 1998.

C. Duncan, S. Kobourov, and A. Kumar. Optimal constrained graph exploration. *ACM Transactions on Algorithms*, 2(3):380–402, 2006.

P. Erdős and A. Rényi. On the strength of connectedness of a random graph. *Acta Mathematica Academiae Scientiarum Hungarica*, 12(1-2):261–267, 1964.

K. Forster and R. Wattenhofer. Directed graph exploration. In R. Baldoni, P. Flocchini, and R. Binoy, editors, *Principles of Distributed Systems*, volume 7702 of *Lecture Notes in Computer Science*, pages 151–165. Springer, 2012.

P. Fraigniaud, D. Ilcinkas, G. Peer, A. Pelc, and D. Peleg. Graph exploration by a finite automaton. *Theoretical Computer Science*, 345(2–3):331–344, 2005.

Y. Gabriely and E. Rimon. Competitive complexity of mobile robot on-line motion planning problems. *International Journal of Computational Geometry and Applications*, 20(03):255–283, 2010.

S. Ghosh and R. Klein. Online algorithms for searching and exploration in the plane. *Computer Science Review*, 4(4):189–201, 2010.

H. Gonzáles-Baños and J.-C. Latombe. Navigation strategies for exploring indoor environments. *The International Journal of Robotics Research*, 21(10-11):829–848, 2002.

Y. Higashikawa, N. Katoh, S. Langerman, and S. Tanigawa. Online graph exploration algorithms for cycles and trees by multiple searchers. *Journal of Combinatorial Optimization*, 28(2):480–495, 2014.

V. Isler. Theoretical robot exploration. Technical report, Computer and Information Science, University of Pennsylvania, 2001.

M. Julia, A. Gil, and Ó. Reinoso. A comparison of path planning strategies for autonomous exploration and mapping of unknown environments. *Autonomous Robots*, 33(4):427–444, 2012.

B. Kalyanasundaram and K. Pruhs. Constructing competitive tours from local information. *Theoretical Computer Science*, 130:125–138, 1994.

S. Koenig. Exploring unknown environments with real-time search or reinforcement learning. In *Proceedings of Neural Information Processing Systems (NIPS)*, pages 1003–1009, 1998.

S. Koenig, C. Tovey, and W. Halliburton. Greedy mapping of terrain. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 3594–3599, 2001.

N. Megow, K. Mehlhorn, and P. Schweitzer. Online graph exploration: New results on old and new algorithms. *Theoretical Computer Science*, 463:62–72, 2012.

S. Miyazaki, N. Morimoto, and Y. Okabe. The online graph exploration problem on restricted graphs. *IEICE Transactions on Information Systems*, E92-D(9): 1620–1627, 2009.

O. Mozos, C. Stachniss, and W. Burgard. Supervised learning of places from range data using AdaBoost. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 1742–1747, 2005.

P. Panaite and A. Pelc. Impact of topographic information on graph exploration efficiency. *Networks*, 36(2):96–103, 2000.

A. Quattrini Li, F. Amigoni, and N. Basilico. Searching for optimal off-line exploration paths in grid environments for a robot with limited visibility. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 2060–2066, 2012.

I. Rekleitis, V. Dujmović, and G. Dudek. Efficient topological exploration. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 676–681, 1999.

D. Rosenkrantz, R. Stearns, and P. Lewis. An analysis of several heuristics for the traveling salesman problem. *SIAM Journal of Computation*, 6(3):563–581, 1977.

Y. Smirnov, S. Koenig, M. Veloso, and R. Simmons. Efficient goal-directed exploration. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 292–297, 1996.

C. Stachniss and W. Burgard. Exploring unknown environments with mobile robots using coverage maps. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1127–1134, 2003.

S. Thrun. Robotic mapping: A survey. In *Exploring Artificial Intelligence in the New Millenium*, pages 1–35. Morgan Kaufmann, 2002.

S. Thrun, A. Bücken, W. Burgard, D. Fox, T. Fröhlinghaus, D. Henning, T. Hofmann, M. Krell, and T. Schmidt. Map learning and high-speed navigation in RHINO. In D. Kortenkamp, R. Bonasso, and R. Murphy, editors, *AI-based Mobile Robots: Case Studies of Successful Robot Systems*. MIT Press, 1998.

C. Tovey and S. Koenig. Improved analysis of greedy mapping. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3251–3257, 2003.

B. Yamauchi. A frontier-based approach for autonomous exploration. In *Proceedings of IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA)*, pages 146–151, 1997.