Value-of-Information based Data Collection in Underwater Sensor Networks

2019

Fahad Khan
University of Central Florida

VALUE-OF-INFORMATION BASED DATA COLLECTION IN UNDERWATER SENSOR NETWORKS

by

FAHAD AHMAD KHAN
MSc Electrical Engineering, University of Engineering & Technology Lahore, 2014
BSc Electrical Engineering, University of Engineering & Technology, Lahore, 2007

A dissertation submitted in partial fulfilment of the requirements
for the degree of Doctor of Philosophy
in the Department of Electrical and Computer Engineering
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Spring Term
2019

Major Professor: Damla Turgut

ABSTRACT

Underwater sensor networks are deployed in marine environments, presenting specific challenges compared to sensor networks deployed in terrestrial settings. Among the major issues that underwater sensor networks face is communication medium limitations that result in low bandwidth and long latency. This creates problems when these networks need to transmit large amounts of data over long distances. A possible solution to address this issue is to use mobile sinks such as autonomous underwater vehicles (AUVs) to offload these large quantities of data. Such mobile sinks are called data mules. Often it is the case that a sensor network is deployed to report events that require immediate attention. Delays in reporting such events can have catastrophic consequences. In this dissertation, we present path planning algorithms that help in prioritizing data retrieval from sensor nodes in such a manner that nodes that require more immediate attention would be dealt with at the earliest. In other words, the goal is to improve the Quality of Information (QoI) retrieved. The path planning algorithms proposed in this dissertation are based on heuristics meant to improve the Value of Information (VoI) retrieved from a system. Value of information is a construct that helps in encoding the valuation of an information segment i.e. it is the price an optimal player would pay to obtain a segment of information in a game theoretic setting. Quality of information and value of information are complementary concepts. In this thesis, we formulate a value of information model for sensor networks and then consider the constraints that arise in underwater settings. On the basis of this, we develop a VoI-based path planning problem statement and propose heuristics that solve the path planning problem. We show through simulation studies that the proposed strategies improve the value, and hence, quality of the information retrieved. It is important to note that these path planning strategies can be applied equally well in terrestrial settings that deploy mobile sinks for data collection.

In the name of the One through whom I am, I seek, and derive meaning for this gift of life!

I dedicate this to my family, friends, mentors, guides and to this beautiful universe and all the gifts that it has given to me. I am blessed!

To all those who have loved me, groomed me, guided me, endured me, and suffered at my behest; I am forever in your debt. I wish you all the most beautiful and best!

May the search for the Truth live on forever and provide meaning for life. May it provide meaning to the ones in pain. May the ones who seek it be fulfilled. May me and my fellow seekers be together forever in bliss and wonder!

TABLE OF CONTENTS

LIST OF FIGURES xii

LIST OF TABLES xvi

CHAPTER 1: INTRODUCTION 1

1.1 Our Contributions 2

1.1.1 Employing Value of Information in Underwater Sensor Networks 3

1.1.1.1 Temporality of VoI - Infotentials 3

1.1.1.2 VoI Model for Sensor Networks 4

1.1.2 VoI based Path Planning Algorithms for Data Mules (AUVs) 4

1.1.2.1 Path Planning for a Single AUV without Resurfacing 5

1.1.2.2 Determining Resurfacing Schedules 6

1.1.2.3 Path Planning with Multiple AUVs 7

1.2 Dissertation Sequence 8

CHAPTER 2: LITERATURE SURVEY 9

2.1 Relationship between QoS, QoI, VoI & Routing 9

2.2 Applications of IQ & VoI in Sensor Networks 10

2.3 Using Mobile Sinks in Underwater Sensor Networks 11

2.4 AUV Path Planning 11

CHAPTER 3: VALUE OF INFORMATION MODELING FOR UNDERWATER SENSOR NETWORKS 13

3.1 Value of Information 13

3.2 VoI Temporality - Infotentials 14

3.2.1 Infotentials - Exponential Models 16

3.3 UWSN Deployment Scenario and Infotential Application 18

3.3.1 Using Multiple AUVs 22

3.3.2 Resurfacing Locations 24

3.3.3 An Alternate Scenario - Node Valuation instead Data Report Valuation . . 24

3.4 Constructing VoI Model and Path Planning Problem 25

3.4.1 VoI maximization 25

3.4.1.1 Data Report Valuation 26

Single AUV: 26

Multiple AUVs: 27

Single AUV with Resurfacing: 28

3.4.1.2 Node Valuation 29

Single AUV: 30

Multiple AUVs: 30

Single AUV with Resurfacing: 30

3.4.2 Role of τ_f in Infotentials 31

3.4.3 The Path Planning Problem 31

3.4.4 Path Planning Performance - VoI Accumulated versus VoI Lost 33

CHAPTER 4: PATH PLANNING HEURISTICS 35

4.1 Path Planning with Infotentials 35

4.1.1 Effect of Valuation and Decay Coefficients 36

4.1.1.1 **Scenario I-a:** 38

4.1.1.2 **Scenario I-b:** 39

4.1.1.3 **Scenario I-c:** 40

4.1.1.4 **Scenario I-d:** 41

4.1.1.5 **Scenario I-e:** 42

4.1.2 Effect of Tour Time 44

4.1.2.1 **Scenario II:** 45

4.1.3 Navigating Tours 46

4.1.3.1 **Scenario III:** 48

4.1.3.2 Selection of Intermediate Nodes 49

4.2 Heuristics for VoI based Path Planning 50

4.2.1 Next Node Visit based on VoI Maximization - H_{MaxVoI} 50

4.2.2 Minimize Tour Time for Path Traversal - H_{ShPath} 51

4.2.3 Visit Intermediate Nodes - $H_{IntVisit}$ 51

4.2.4 Load Balancing in terms of Nodes Visited - $H_{NodeBal}$ 53

4.2.5 Balanced Distribution of Nodes in terms of VoI - H_{VoIBal} 54

4.2.6 Partitioning Map on basis of Node Proximity - $H_{MapPart}$ 55

CHAPTER 5: SINGLE AUV PATH PLANNING 57

5.1 VoI Maximization and Path Planning Problem 57

5.2 Path Planning Algorithms 58

5.2.1 Lawn-Mower Path Planner - *LPP* 58

5.2.2 Greedy Path Planner - *GPP* 61

5.2.3 Greedy Path Planner with Intermediate-Node-Visitation - *GIPP* 62

5.2.4 Hybrid Path Planner - *HPP* 65

5.2.5 Hybrid Path Planner with Intermediate-Node-Visitation - *HIPP* 66

5.2.6 Random Path Planner - *RPP* 67

5.3 Performance Measures, Simulation Setup and Results 67

5.3.1 Performance Measures and Experiments 67

5.3.2 Simulation Setup 70

5.3.3 Results 71

5.3.3.1 Valuation Ratio 71

5.3.3.2 Justification of Heuristics 73

5.3.3.3 Comparative Analysis 76

VoI Accumulated and VoI Lost 76

Tour Time 77

Efficiency 77

5.3.3.4 Scalability with an increase in Hot-Spots 81

5.3.3.5 Response to Emergency 82

5.4 Remarks 85

CHAPTER 6: DETERMINING RESURFACING SCHEDULES 87

6.1 VoI Maximization and Path Planning Problem 87

6.1.1 Scheduling AUV Resurfacing 88

6.2 Genetic Algorithms for Resurfacing Schedules 91

6.2.1 H_{PR} - Periodic Resurfacing Heuristic 91

6.2.2 Genetic Algorithms 93

6.2.2.1 G_{PR}:A for Optimal Periodic Resurfacing Schedule 93

6.2.2.2 G_{Opt}:A for Optimal Resurfacing Schedule 93

6.3 Simulation Setup & Results 94

6.3.1 Studying the effect of Deployment Depth on AUV Resurfacing 95

6.3.2 Performance Analysis of G_{PR} & G_{Opt} Heuristics 96

6.4 Remarks 98

CHAPTER 7: MULTIPLE AUV PATH PLANNING 99

7.1 VoI Maximization and Path Planning Problem 99

7.2 Multiple AUVs - A Physical Strategy to Improve VoI 101

7.3 Various Heuristic Combinations for Path Planning Algorithms 102

7.4 Multiple AUV Path Planning Algorithms 104

7.4.1 Random Path Planner - *RPP* 105

7.4.2 Zig-Zag Path Planner - *ZPP* 105

7.4.3 Lawn-Mower Path Planner - *LPP* 105

7.4.4 Greedy Path Planner with Node Balancing - GPP_N 107

7.4.5 Greedy Path Planner with Node Balancing and Intermediate Node Visitation - $GIPP_N$ 107

7.4.6 Greedy Path Planner with VoI Balancing - GPP_{VB} 111

7.4.7 Greedy Path Planner with VoI Balancing and Intermediate Node Visitation - $GIPP_{VB}$ 112

7.4.8 Map Partitioned Greedy Path Planner - GPP_{MP} 112

7.4.9 Map Partitioned Greedy Path Planner with Intermediate Node Visitation - $GIPP_{MP}$ 114

7.5 Simulation Setup & Results 115

7.5.1 VoI Accumulated in various Spatial Distributions of Hot-Spots 117

7.5.2 AUV Tour Time with increasing number of AUVs 120

7.5.3 Emergency Response - Time to hit First Hot-Spot 122

7.6 Remarks 124

CHAPTER 8: CONCLUSION 125

LIST OF REFERENCES 127

LIST OF FIGURES

Figure 3.1: Examples of Infotentials 15

Figure 3.2: The figure is an example of exponentially decaying infotentials that we use for modelling and analyzing high-priority and low-priority events. 18

Figure 3.3: An example of a UWSN which uses AUVs to offload data from sensor nodes. 20

Figure 3.4: The figure shows various parameters and measures that we use for comparative analysis of path planning algorithms. 33

Figure 4.1: Traversals are the routes that the mobile sink *m* will use to navigate through the map to collect data from the sensor nodes *x*, *y* & *z*.
(a) The configuration on the left side will be used to study valuation and decay coefficients.
(b) The configuration on the right side will be used to understand the effect of tour time. 37

Figure 4.2: Alternate source to destination routes with approximately equal lengths. . . . 47

Figure 4.3: Heuristic H_{MaxVoI} - VoI Maximization 50

Figure 4.4: Heuristic H_{ShPath} - Shortest Path 51

Figure 4.5: Heuristic $H_{IntVisit}$ (Intermediate Node Visitation) 52

Figure 4.6: Heuristic $H_{NodeBal}$ (Node Balancing) 53

Figure 4.7: Heuristic H_{VoIBal} - VoI Balancing 54

Figure 4.8: Heuristic $H_{MapPart}$ - Map Partitioning 55

Figure 5.1: The UWSN mesh setting used for the simulation study. The same arrangement will be used for simulation and experimentation in the results section. The blue colored path is of the AUV that uses *LPP*, red of the AUV that uses *GPP* and green of the AUV that uses *GIPP*. The *GPP* visits one hot-spot after another. Notice how *LPP* stumbles upon a hot-spot, i.e., discovers it by chance. Also, observe how *GIPP* visits intermediate nodes in-between hot-spot visits. 59

Figure 5.2: The top two graphs (colored in blue) are for Υ_{Acc} while the bottom two are for Υ_L while the graphs on the left are for the valuation ratio range $[10^0 : 10^9]$ while the graphs on the right are a magnification between $[10^4 : 10^5]$: (a) The top-left graph is for Υ^n_{Acc} which is charted in the range $[10^0 : 10^9]$; (b) The top-right graph is for Υ^n_{Acc} which is charted in the range $[10^4 : 10^5]$; (c) The bottom-left graph is for Υ^n_L which is charted in the range $[10^0 : 10^9]$; (d) The bottom-right graph is for Υ^n_L which is charted in the range $[10^4 : 10^5]$. . . . 72

Figure 5.3: Bar graphs for justifying the use of various heuristics : (a) Greedy vs. Lawn-Mower; (b) Inter-Node Traversal Advantage; (c) Hybrid Advantage; (d) Hybrid vs. Greedy counterparts. 75

Figure 5.4: Comparative analysis for Υ_{ACC} & Υ_L : (a) The top stacked bar graph is for VoI accumulated Υ_{ACC}; (b) The bottom stacked bar graph is for VoI lost Υ_L. . 78

Figure 5.5: Comparative analysis for T_{tour} & Ω : (a) The top stacked bar graph is for time to complete the tour T_{tour}; (b) The bottom stacked bar graph is for the efficiency measure Ω. 79

Figure 5.6: Scaling of response w.r.t. number of hot-spots N_{HS} : (a) Percentage improvement in VoI accumulated Υ_{ACC}; (b) Percentage reduction in VoI lost Υ_L; (c) Percentage reduction in tour time T_{Tour}; (d) Normalized measure of efficiency Ω. 80

Figure 5.7: Emergency Response w.r.t. number of hot-spots N_{HS} : (a) Percentage improvement in VoI accumulated from first hot-spot $\upsilon_{HS_{ACC}}$; (b) Percentage reduction in VoI lost from first hot-spot υ_{HS_L}; (c) Percentage reduction in time to reach first hot-spot τ_{HS}; (d) Normalized urgency score Ψ. 83

Figure 6.1: Side view of the underwater sensor network showing inter-node distance S_I and node depth from sea surface S_D. 89

Figure 6.2: The average number of times the AUV resurfaces as a function of the ratio $R_{DI} = S_D/S_I$. The results are averaged over 50 simulation runs. 96

Figure 6.3: The VoI accumulated (Υ_S) by different schedulers as a function of the number of sensor nodes. The results are averaged over 50 simulation runs and have been normalized w.r.t. R_{End}. 97

Figure 7.1: Physical strategies to improve VoI accumulated by improving speed of a mobile sink (MS) or by using multiple mobile sinks. 100

Figure 7.2: An example of UWSN deployed in the Strait of Gibraltar. The inter-node distance is approximately equal to 1 *km* and is drawn to-scale in the figure. . . 116

Figure 7.3: VoI accumulated by various path planners with two AUVs. 118

Figure 7.4: VoI accumulated by various path planners with five AUVs. 119

Figure 7.5: Average distance travelled by each AUV with increasing number of AUVs. . 121

Figure 7.6: Time taken to hit the first hot-spot given the hot-spots are randomly distributed.123

LIST OF TABLES

Table 6.1: Simulation Parameters 94

Table 7.1: A listing of heuristics employed by various algorithms 103

CHAPTER 1: INTRODUCTION

Traditional wireless sensor networks used for monitoring environmental conditions such as temperature or humidity level [1] require only low data bandwidth and do not usually pose real-time communication challenges. On the other hand, if the sensor network is used to monitor for intruder tracking or catastrophic events, it might require larger quantities of data, such as images or video streams.

Monitoring in underwater sensor networks (UWSNs) has emerged as a subject of active research with applications such as maritime security operations, infrastructure surveillance and maintenance, sea-life monitoring and sea pollutant & contaminant measurements. Routing techniques in UWSNs differ from traditional sensor networks. The physical deployment and maintenance of the nodes are difficult and underwater sensing abilities can be hindered by cloudy water and debris. Radio waves travel only very short distances underwater (as water tends to absorb a large part of the electromagnetic spectrum). This limitation can be mitigated by using an acoustic communication medium. Acoustic signals can travel long distances underwater but they have relatively high latency and low bandwidth. Such a bandwidth may be sufficient for transmitting compressed data [2] but is not appropriate for greater volumes of data such video recordings.

As an alternative, the research community proposed the use of mobile sinks to gather and offload the data in wireless multimedia networks [3, 4]. In the case of UWSN, an autonomous underwater vehicle (AUV) can be used as a mobile sink. The total energy consumption of the network significantly drops by the use of an AUV to transport data. However, the physical transport of the data creates a long latency during the path taken by the AUV. A potential solution is to use short distance optical communication between the sensor node and a nearby AUV, and acoustic communication for signaling of events for data offloading.

An important issue for data offloading in UWSNs is the path planning for an AUV, i.e. the sequence in which the AUV visits the nodes for data offloading. An efficient planned path can reduce data delivery latency and increase the collected information over a given period of time. Let us assume a surveillance application that must report an important event. In a traditional sensor network, the information can be sent to the sink node instantaneously by initiating the data transfer from either the sink or the source nodes. But for a UWSN with a mobile sink, the sensor node must signal the event to the AUV and the AUV must travel to the required node for data collection. The time when the information is transferred to the customer depends on the speed of the AUV (which depends on weather and oceanic currents), the current location of the AUV, its schedule of diving and resurfacing. As the AUV needs to handle all the nodes of the network, the time until the information about a given event is picked up can range from a couple of minutes to a few hours depending on the choices made by the path planning algorithm.

1.1 Our Contributions

In this dissertation our primary contribution is developing techniques that will allow us to retrieve data from a UWSN in an event prioritization fashion, thereby, allowing us to cater to emergency situations at the earliest. We make use of Value of Information (VoI) for developing heuristics that will be incorporated in path planning algorithms for the AUV that will traverse the sensor nodes in the UWSN. Our contributions have two aspects: one is formulating the use of VoI as a temporal construct, developing the VoI optimization problem statement for sensor networks and identifying constraints for VoI based data retrieval in UWSNs; the other one is developing path planning algorithms for AUVs that traverse the UWSN in a manner that it optimizes VoI retrieved under constraints such as emergency situation requirements, resurfacing rules, and number of AUVs.

1.1.1 Employing Value of Information in Underwater Sensor Networks

The first step is to determine the manner in which we want to employ the concept of VoI. For this, we come up with a temporal definition of VoI; namely infotentials. Afterward, we use this definition of infotentials to develop a VoI model in a sensor network. We apply this model to a UWSN that uses AUVs for data retrieval and then determine a VoI optimization relation.

1.1.1.1 Temporality of VoI - Infotentials

VoI has been discussed in the literature in more of a non-temporal sense. In a game theoretic setting, it is defined as the price of an optimal player would pay to acquire a segment of information. We term the subject of an information segment as ‘asset’. The value of an information segment is directly proportional to the valuation of the asset. Also, the value of an information segment is dependent upon factors such as its reliability, accuracy, repetitiveness, uniqueness, etc.

We introduce VoI as a temporal entity; one that is decaying in time. We argue that the value of an information segment decreases across time due to the depreciation of the asset the information segment is reporting about. Specifically, if this depreciation is triggered due to a catastrophic event, we can see a rapid decline in the valuation of the asset. This depreciation in time can be captured by time decaying functions which we term as ‘infotential(s)’, which is short for Information-Potential. An infotential is a function that captures the valuation of a piece of information about an event that will depreciate the value of an asset.

Other factors that affect VoI such as reliability, accuracy, repetitiveness, uniqueness, etc. can be categorized under the concept of Quality of Information. We have developed an equation that captures the VoI in terms of asset valuation, event-based depreciation, and QoI. This equation is what we term as the infotential.

1.1.1.2 VoI Model for Sensor Networks

We develop a VoI model for a single sensor node. From this, we develop a VoI model for a UWSN which uses AUVs for data retrieval form sensor nodes. The model incorporates in it the geophysical location of the sensor nodes, the information segments specifics at the sensor nodes such as information class and VoI assignments and aspects of the AUV such as velocity and resurfacing constraints. We developed a simulator that analyzes the performance of various path planners in terms of VoI accumulated in the aforementioned UWSN setting. The AUV traverses the UWSN and retrieves information segments from the sensor nodes in the simulation. Afterward, various metrics are used and an analysis is run to gauge the performance of the path planners.

From this model, we derive the equation for the VoI optimization relation. The optimization problem is to maximize the amount of VoI accumulated from the UWSN in one complete tour of the AUV. The goal of each path planner would be to maximize the value of this optimization relation.

1.1.2 VoI based Path Planning Algorithms for Data Mules (AUVs)

We have developed path planning algorithms for scheduling the sequence in which the sensor nodes will be visited by the AUV in a UWSN. We identify two basic scenarios for which we propose the path planning algorithms. In the first scenario, the AUV needs to resurface occasionally during its tour for delivering data so that certain VoI objectives can be met. In the second scenario, the AUV can meet the required objectives without the need to resurface. The decay of VoI of an information segment stops at the time instant when the information segment is processed for certain decision-making purposes. If the AUV is not equipped with the required decision-making ability then the AUV needs to resurface and transmit to a remote sink node for the required decision. Alternately, if it is equipped with the required ability then it does not need to resurface and can take the required decisions on its own. For both these scenarios, we propose heuristics and path planning algorithms that solve the VoI optimization problem. The proposed algorithms involve a single AUV. Multiple AUVs can be used to achieve a higher aggregate of VoI. We also propose heuristics and algorithms for scheduling multiple AUVs. We have, therefore, proposed path planning algorithms for the following scenarios: path planning for a single AUV without resurfacing, determining resurfacing schedules in the case of resurfacing and path planning with multiple AUVs.

1.1.2.1 Path Planning for a Single AUV without Resurfacing

In this project, we design heuristics for path planning algorithms in a UWSN setting where resurfacing is not required. We propose that visiting nodes with higher VoI first and minimizing AUV travel time can serve as good heuristics for the path planning algorithms. We developed several algorithms based on these heuristics.

The first algorithm, Lawn-Mower, is based on the time minimization heuristic. The heuristic minimizes the overall tour time and is approximates a solution to the traveling salesman problem in a mesh setting of vertices (sensor nodes). The second one is the Greedy path planner and it is based on choosing the next node to be visited as the one which has the highest amount of VoI to offer. It is based on maximizing the VoI accumulated heuristic. The third path planner is Greedy with Inter-Node Traversal that modifies Greedy with a subtle addition: while moving towards the next destination as determined by the highest VoI offer, it visits any sensor node that it encounters along the path and offloads data from them too, thereby, minimizing tour time. Hence, it uses VoI maximization in conjunction with time minimization.

In our simulations, we consider two situations: with and without hot-spots. A hot-spot is a region where a sensor node has identified a catastrophic event. Reporting of such an event carries high priority and thus, the VoI retained at sensor nodes reporting hot-spots is higher than usual. In our simulations, we find out that the Lawn-Mower performs best in situations where there are no hot-spots while the Greedy path planners perform better in the case of hot-spots. Keeping these results in view we have developed two more algorithms; Hybrid & Hybrid with Inter-Node Traversal which is a spin on Greedy & Greedy with Inter-Node Traversal respectively. The Hybrid path planner variants behave as a Lawn-Mower when there are no hot-spots and as a Greedy path planner when there are one or more hot-spots. In this way, these Hybrid algorithms achieve the best of both worlds.

To have a detailed evaluation of the efficacy of the path planners, we introduced two metrics. The first performance metric is the reciprocal of the product of VoI lost and tour time. The other one is a measure of efficiency for the visitation schedule of hot-spots and is named as urgency ‘ efficiency’.

1.1.2.2 Determining Resurfacing Schedules

In the previous project, we worked with the assumption that the AUV has certain decision-making abilities due to which it does not need to resurface to report the data for actuation purposes. In this project, we consider that resurfacing is required so that data can be transmitted to a remote node for further action. Therefore, the question we ask is that what would be a resurfacing schedule that would result in a higher amount of VoI accumulated.

The way we go about developing a path planning algorithm for this scenario is that we break it down into two steps. First, we figure out a path using the strategies in the previous project i.e. assuming no need to resurface we determine a path that hypothetically maximizes VoI. In the second step, we plug this path with the required resurfacing locations assuming that the AUV needs to resurface if VoI of information segments is to be stopped from further decaying.

For plugging the path with resurfacing locations we first determine the effect of the ratio of sensor node deployment depth to intermediate node distance. We have determined that if the ratio is too large then it makes more sense to visit all nodes before resurfacing. In contrast, if the ratio is too small then resurfacing and transmitting at every node is more appropriate. However, there is a range somewhere between these two extremes where determining a resurfacing schedule will result in a higher VoI return. We have developed algorithms that, based on this range, determine the appropriate resurfacing strategy, and then compute the optimum resurfacing schedule.

We propose a periodic resurfacing heuristic. On the basis of this, we developed an algorithm that gives us the best periodic resurfacing schedule. We also developed a genetic algorithm for finding the optimal resurfacing schedule. The choice between these algorithms is a trade-off between time complexity or a more optimal VoI aggregate.

1.1.2.3 Path Planning with Multiple AUVs

The goal for this project is to develop path planners that assign the sensor nodes to different AUVs for data offloading. We designed three more heuristics in addition to the heuristics proposed in the aforementioned project; thereby, giving us a repertoire of six heuristics. Path planners for multiple AUVs will use a combination from among these six heuristics to perform the scheduling task.

The first heuristic that we propose is balancing the number of nodes to be distributed among the AUVs. This will in effect help in balancing tour time, and hence, can be considered a time minimization metric. The second heuristic assigns sensor nodes to AUVs in a manner that balances the VoI distribution; the next node with highest VoI at offer should be assigned to the AUVs in a round-robin fashion. Such round-robin assignment by default fulfills the first heuristic of balancing nodes. This heuristic is essentially an attempt to maximize VoI by reducing time to higher priority nodes by assigning them into schedules as early as possible. The third heuristic is to partition the map into as many equal spaces as the number of AUVs. This will further reduce travel time. This heuristic also inherently fulfills the first heuristic i.e. balancing the number of nodes.

Based on various combinations of these six heuristics we develop a couple of algorithms. We establish the efficacy of these algorithms and heuristics through simulation and analysis. The two defining strategies are VoI balancing versus map partitioning. We find out that algorithms with the map partitioning heuristic perform the best.

1.2 Dissertation Sequence

In Chapter 2 we present a review of the literature that serves as the motivation for this dissertation. In Chapter 3 we define the concept of VoI and Infotentials in the UWSN scenario. In this chapter, we also formulate the problem statements that we will be dealing with in this dissertation. In Chapter 4 we present an analysis of Infotentials with the help of which we determine what can be useful heuristics for VoI based path planning algorithms. In Chapter 5 we propose heuristics and algorithms for the single AUV path planning problem. Using simulations, we discuss the results for the case of a single AUV which does not resurface. In Chapter 6 we describe heuristics and algorithms and carry out an analysis for the scenario of a single AUV that requires occasional resurfacing. While Chapter 5 and 6 deal with single AUVs, Chapter 7 deals with multiple AUVs. In Chapter 7 we discuss heuristics and path planning algorithms and their respective performance for the case of multiple AUVs in the non-resurfacing scenario. We conclude in Chapter 8.

CHAPTER 2: LITERATURE SURVEY

2.1 Relationship between QoS, QoI, VoI & Routing

The over-arching goal of this dissertation is to improve the QoI (quality of information) that can be obtained from a UWSN. In terms of the performance objectives, QoI is to sensor networks what QoS (quality of service) is to conventional computer and wireless networks [5, 6, 7]; although, technically speaking, in terms of implementation, QoI is built upon a QoS layer in sensor networks [8, 9, 10]. It is shown in [5, 7] that the construct of QoI is intrinsically related to VoI, therefore, we use VoI for path planning purposes to improve QoI garnered from the system.

Design of good routing algorithms is one of the fundamental aspects that dictates QoS and QoI in conventional networks [11], ad hoc and sensor networks [12, 13, 14, 10, 15] and networks with mobile elements [16, 17, 18, 19, 20, 21]. Routing in networks is tailored to meet certain application-level or operational objectives [22]. Wang et al. [23] introduces m-limited forwarding algorithm to reduce the power consumption of the nodes and improve the routing performance through forwarding packets to the limited set of nodes. Rahmatizadeh et al. [24, 25] proposed a directional virtual coordinate routing towards a mobile sink using virtual coordinates in a wireless sensor network.

With the advent of IoT [26] and Fog-computing [27] paradigms, there continues to remain a keen interest in the development of routing algorithms to meet the new emergent objectives and constraints [28, 29, 30]. In this dissertation, we employ a mobile sink (AUV) as part of the routing effort to deliver a better measure of VoI and hence QoI.

2.2 Applications of IQ & VoI in Sensor Networks

Information quality (IQ) aware routing schemes make data routing decisions based on the threshold of information required by application or mission objectives [31, 32, 1]. The data is fused en-route to the sink node and routed on paths which can potentially satisfy the aggregated IQ threshold or constraints. Data is kept fusing incrementally on each of the next forwarding node until it reaches the IQ threshold. When the threshold is met, the information is sent to the main fusion center or sink node.

We use VoI in an event driven context. An event-based information quality aware routing (IQAR) has been proposed in [33]. The unique aspect about IQAR is its use of event-based data fusion whereas the previous IQ aware routing algorithms would initiate from the fusion centers. In the initial phase of the IQAR method, an aggregation tree is constructed that spans the whole network. In the case of an event, the sensors in vicinity forward the data using the preexisting links as established during the tree initialization. After data reaches the fusion center, the network uses a greedy approach to prune the initial tree in order to maximize the information retrieved and to minimize the energy consumption of the nodes on the data forwarding links.

One of the applications where IQ and VoI have been used in sensor networks is for tracking [32, 34]. In [34] a VoI based approach has been proposed for intruder tracking objectives. The sensor network, based on predictive measures centered around VoI, makes data routing decisions in the network for more efficient target tracking. The network is able to prioritize among high and low-value targets. Tracking a higher value target should result in a higher amount of VoI accumulated. In [35], the authors further proposed IVE which is an improved version of the VoI based intruder tracking system. The network is able to control the demand of data packets while balancing trade-offs between network energy consumption and required VoI.

Concepts of quality and utility have been frequently applied to a variety of scheduling activities in sensor networks. There are some examples of scheduling algorithms and protocols [36, 37, 38, 39, 40, 41, 42, 43] which employ concepts of quality based on priority and utility metrics.

2.3 Using Mobile Sinks in Underwater Sensor Networks

Mobile sinks have been used as data mules in both sensor networks [4, 44, 45, 46] and underwater sensor networks [3, 47, 48, 49, 50]. Mobile sinks in UWSNs have been used in the form of AUVs [48, 49] as well as dolphins [51]. In [51] it has been proposed to use a DDD (delay-tolerant data dolphin) to harvest the data from sensor nodes in the region of interest. The data collection event can be triggered by both the DDD or the sensor node. The DDD uses a bi-directional acoustic communication medium. As the movement of DDDs is random, therefore, there is a randomness associated with the event data collection, and this inhibits optimal performance.

VoI based transmission scheduling of sensor nodes via acoustic links to the sink has been explored in [52, 17, 16, 18]. VoI based path planning of an AUV to collect data from sensor nodes has also been explored in [53, 54, 55, 56, 57, 58].

2.4 AUV Path Planning

Path planning is a diverse subject and there is a lot of literature on it specifically in the artificial intelligence community [59, 60]. In terms of route discovery, a path planner is an algorithm that provides a sequence of steps which give a valid route between two points on a map (usually represented by a graph or a grid). Among the sub-goals or constraints of path planning algorithms is to find the most optimal route in terms of shortest distance, minimal time, optimal fuel expenditure, and so on. They essentially convert a set of high-level goal specifications into a sequence of low-level instructions, thus breaking down the problem solution into simpler steps.

Depending upon the nature of the environment path planners can be static or dynamic based on how the information is being updated. Path planners, therefore, can also be classified as pre-computing algorithms or reactive algorithms [60]. If the path is planned before the mobile agent starts the course then the algorithm is pre-computing. If the plan is updated during the course, in reaction to changes in objectives, constraints or new obstacles, then the algorithm is deemed reactive. Maps represented by graphs are usually solved by planners that are variants of Dijkstra's Algorithm. One of the well-known variants is A^* path planning which uses admissible heuristics to accelerate the path planning. The path planned in [56] can be seen as a two-tiered approach; the higher level algorithms, such LPP or GPP, determine the sequence of node visitation on the map, while A^* provides the detailed sequence of steps required to travel from a source to a destination node.

AUVs are autonomous devices and hence require path planning techniques to help them navigate beneath seas and oceans. In terms of navigation, they have certain issues that hamper them more as compared to dry-land autonomous agents, which include communication limitations, limited sensing, and power issues. AUV path planning has been around for a while now. One of the earlier path planning techniques used for AUVs is case-based reasoning [61]. One of the first efforts to use $A*$ for path planning for AUVs is in [62]. FM^* in [63] gives a continuous path based on a discrete representation of the environment and also takes water currents into account. The path planning approach in [64] uses methods based on potential field strategies. This approach has been further improved on in [65]. Genetic Algorithms [66, 67, 68] and evolutionary algorithms [69] have also been employed for AUV path planning. Path planning for data mules has also been explored from the perspective of sensor networks in [45, 46]. We have proposed path planning techniques for AUVs in UWSNs [55, 56, 57, 58].

CHAPTER 3: VALUE OF INFORMATION MODELING FOR UNDERWATER SENSOR NETWORKS

3.1 Value of Information

Information is knowledge that helps in developing and updating a model for monitoring an interest. It is based on this model that certain steps are taken to keep the situation in check. Value of Information (VoI) is the valuation of an information segment relevant to and can be added to this model.

Let us consider a marine environment that is being monitored by an underwater sensor network for reporting any catastrophic event. An example could be an offshore oil-rig and pipeline system that is under surveillance for events such as potential oil spills. The sensor nodes have video cameras which record information that can be processed to ascertain aging, rust, accidental damage or oil leaks. The higher the increase in the risk of an oil spill, as concluded from a model after adding an information segment, proportional is the increase in valuation of the information segment that leads to that conclusion.

From the perspective of multi-agent systems, VoI is the price a player would pay to acquire a piece of information in a game theoretic setting. This description is consistent with the usage of the term VoI by Howard in [70].

To have a more abstract view, let us consider a classic control theory scenario where, based on feedback, an agent has a certain degree of observability and controllability over a system. The actions taken are such that some measure of fiscal profit or loss is incurred. These profits or losses are assessed from the current and future states of the system. The observations are part of the information based on which the future discourse of the steps is decided by the agent. Therefore, information is a data report that can aid in building a more accurate model of a system such that the newer model aids in alleviating for instance fiscal consequences.

3.2 VoI Temporality - Infotentials

Infotential is a term that we have coined for information potential. By potential, we imply the potential of impact an event has on an asset, and therefore, the report of such an event can be deemed to have a certain information potential. Infotentials are functions that encode the variability of VoI. VoI of a report is subject to various factors such as the valuation of the asset being reported; the impact of the event being reported on the asset; timeliness of the report; reliability, precision, and accuracy of the report; and the fact that whether this report is a repetition or not. Asset valuation dictates VoI but as there is a temporal dynamism to this valuation, therefore, VoI should correlate with this variability across time. We quote two factors that result in asset valuation decay; the first is that assets naturally depreciate with time; the other is that events of catastrophic nature can result in a sharper decline in the valuation of an asset. As these fiscal attributes are decaying in nature, therefore, VoI should be a monotonically decreasing function.

In the context of this dissertation, we define a function that encodes the temporal variability of VoI. We consider this variability to be monotonically decreasing in time; we assume that at any point in time the valuation of the asset does not increase due to any other factor. These functions can be designed in a variety of different ways so as to fit the needs of an application scenario. We model depreciation with a decaying exponential so as to maintain generality. Infotentials don't necessarily need to be in exponential form. In the physical world, however, most models and descriptions of decay are exponential in nature. For example, both physical laws governing radio-active decay and models of asset valuation depreciation in economics have exponential decay formulations.

Therefore, we justify our use of decaying exponentials as a general model for infotentials. On the other hand, functions such as descending staircases or any other complex combination of exponentials can also be used. Ideally, the function should be constructed based on practical statistical data or any realistic model that charts out the valuation accurately. Examples of various types of infotentials are given in Figure 3.1.

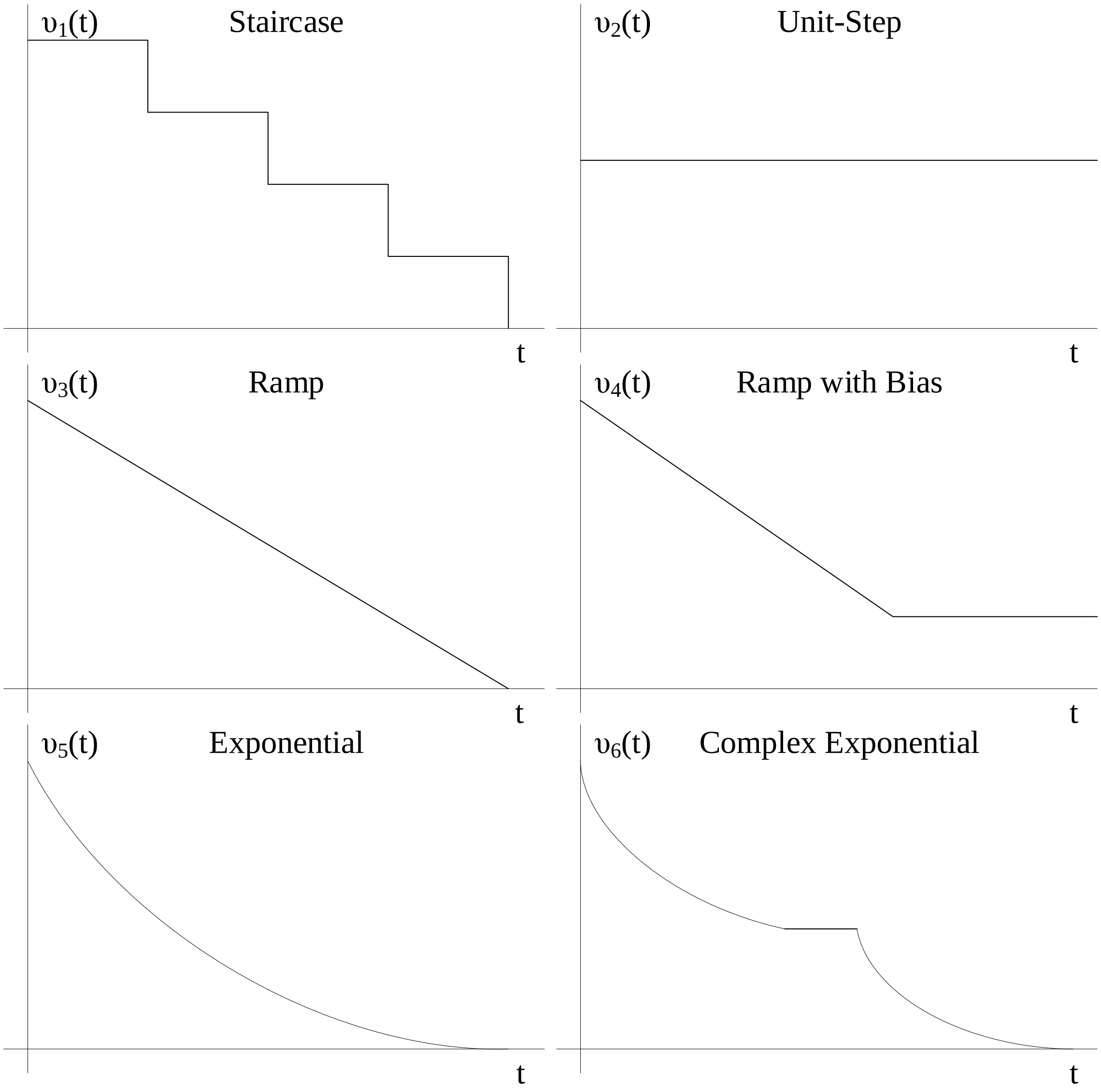


Figure 3.1: Examples of Infotentials

As discussed earlier, these functions can be designed in a variety of different ways so as to suit one's narrative of an application scenario. We understand that these functions should generally have two types of parameters that can control the shape of the monotonically decreasing curves. These parameters respectively control the significance of the event (valuation of the information at $t = 0$, i.e. when event is first observed) and decay of information valuation with time. Figure 3.1 shows different types of monotonically decreasing functions which we denote by $\upsilon(t)$. A staircase function can help in modelling valuations in discrete steps. A unit-step assumes no decay in valuation; it is like assigning a packet to a certain priority class in networked systems that use class based priority to meet QoS needs, where no packet is ever discarded due to absence of any hard-real-time constraints. It is understood that, if an information valuation goes to zero then there in no point in delivering that piece of information. Such a situation can be modeled using ramp functions. This strategy is useful in situations with hard-real-time or soft-real-time-constraints. But what if information still need to be delivered for historical records, although no further degeneration of an asset is taking place due to an event. In that case, it would be useful to add a bias infotential as in the case of the ramp with bias function (unit-step added to ramp). We can also have complex exponentials, which are a combination of multiple exponential functions, or whose exponents are a combination of multiple parameters. One reason to use exponential is that they are always approaching a limiting value, e.g. always approaching zero but not having the value zero itself. This is a very effective way of encoding time precedence in various information segments that have similar significance and decay rates but are never deemed to have zero valuation.

3.2.1 Infotentials - Exponential Models

In this study, we employ exponential functions of the form

$$\upsilon(t) = Ae^{-B(t-\tau_o)} \tag{3.1}$$

Examples of the infotential $\upsilon(t)$ are shown in Figure 3.2. The parameters that control the function are A and B and τ_o. The parameter τ_o is the time at which the event is reported. Parameters A and B scale the VoI across the domain and range of the function. Parameter A represents the valuation of information at $t = \tau_o$ while parameter B represents the decay in information valuation A for $t > \tau_o$, i.e. after the onset of the event being reported at τ_o. The two exponential functions in Figure 3.2 could be understood as representing two different classes of information in an application scenario. The more the valuation of an asset is, the higher the number A would be. Likewise, higher the B value is, greater is the rate of damage to an asset.

Let there be two events E_H and E_L with significance and damage rate $\{S_H, D_H\}$ and $\{S_L, D_L\}$ respectively. The subscript H corresponds to a high-priority event while the subscript L indicates a low-priority event. We denote the corresponding VoI functions as,

$$\begin{aligned} \upsilon_H(t) &= A_H e^{-B_H(t-\tau_{oH})} \\ \upsilon_L(t) &= A_L e^{-B_L(t-\tau_{oL})} \end{aligned} \tag{3.2}$$

The relationship between significance and valuation, and between damage rate and decay, is given as,

$$\begin{aligned} S_H > S_L &\implies A_H > A_L \\ D_H > D_L &\implies B_H > B_L \end{aligned} \tag{3.3}$$

These inequalities are general guidelines. The actual values of parameters A_x and B_x need to be configured with the help of a system expert or should be based on statistical data.

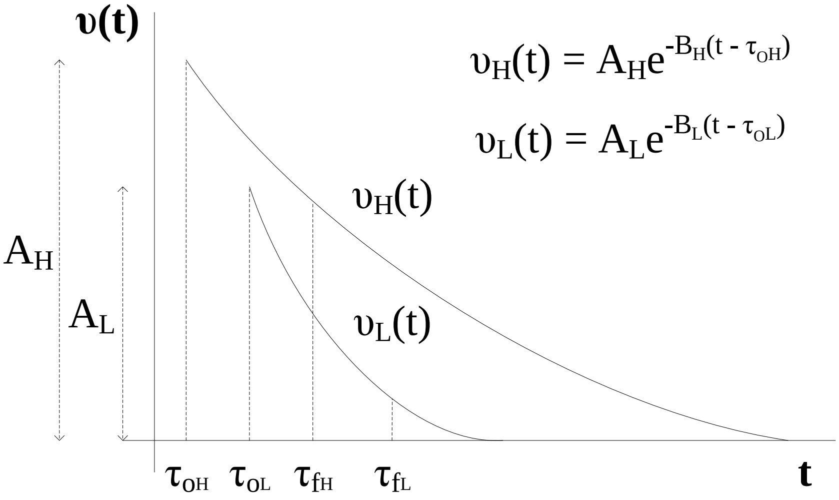


Figure 3.2: The figure is an example of exponentially decaying infotentials that we use for modelling and analyzing high-priority and low-priority events.

3.3 UWSN Deployment Scenario and Infotential Application

Let us assume that the UWSN has been deployed to monitor disasters such as oil spills and leaks from vessels or pipelines. Sensor nodes in the UWSN have the ability to detect and classify such disasters. Therefore, we have two classes of information reported; one is normal routine data while the other is data pertaining to disasters. Both classes of information are mapped to separate infotentials. The information for high-priority events such as oil leaks is mapped to the function $\upsilon_H(t)$ while normal routine information is classified as low-priority and is mapped to $\upsilon_L(t)$. The UWSN considered in this dissertation is illustrated in Figure 3.3. We assume a UWSN with n sensor nodes

$$S = \{s_1, s_2, \ldots, s_j, \ldots, s_n\} \tag{3.4}$$

These nodes are equipped with sensors that collect high-quality video data which makes their transmission over acoustic channels unfeasible. An AUV is deployed to gather data from these sensor nodes. The sensor nodes have acoustic communication modules for long distance communication such as transmitting signaling and configuration messages while optical communication modules are used for short distance transmissions such as offloading the data from the sensor nodes to the AUVs. The sensor nodes have the ability to classify data into q different information classes

$$C = \{c_1, c_2, \ldots, c_p, \ldots, c_q\} \tag{3.5}$$

Each information class is characterized by a unique infotential. The VoI functions that we use, as discussed earlier, are of the form

$$\upsilon(t) = Ae^{-B(t-\tau_o)} \tag{3.6}$$

In this dissertation, we assume a binary class model for the information i.e. high-priority versus low-priority. The infotentials for the high-priority and low-priority information classes are as follows respectively

$$\begin{aligned} \upsilon_H^{report}(t) &= A_H e^{-B_H(t-\tau_{oH})} \\ \upsilon_L^{report}(t) &= A_L e^{-B_L(t-\tau_{oL})} \end{aligned} \tag{3.7}$$

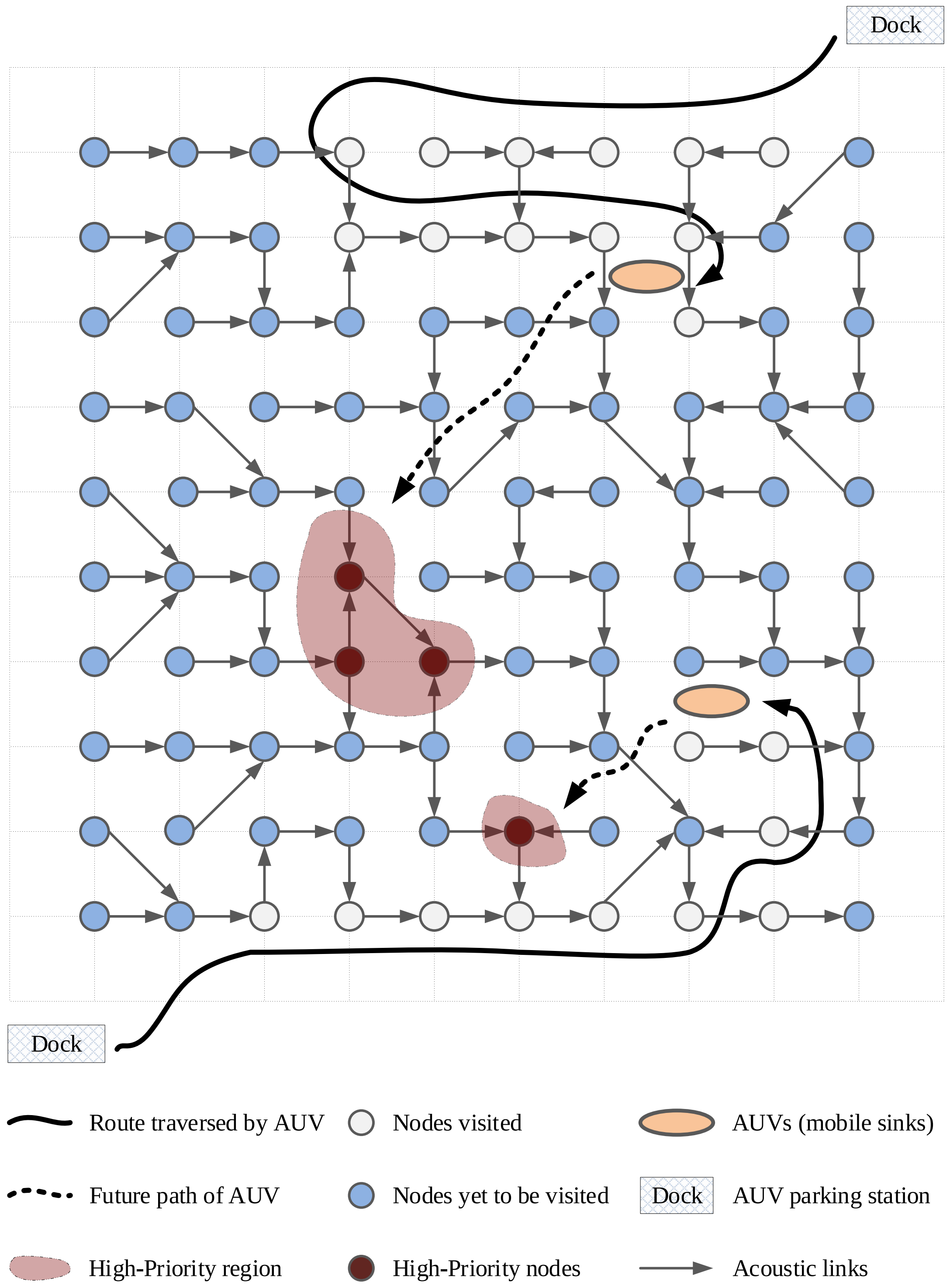


Figure 3.3: An example of a UWSN which uses AUVs to offload data from sensor nodes.

The sensor nodes process and save the data in the form of data reports. After time t the j_{th} sensor node has d data reports

$$D_i = \{d_{j1}, d_{j2}, \ldots, d_{jk}, \ldots, d_{jd}\} \tag{3.8}$$

Data reports are mapped to specific information classes such that the mapping is surjective. This means that a classification function α^{data} (a computational procedure in the sensor node) may assign more than one data report to the same information class

$$\alpha^{data} : D \rightarrow C \tag{3.9}$$

In terms of the j_{th} sensor node and k_{th} data report, the infotentials are denoted with subscripts as

$$\upsilon^{r}_{jk}(t) = A_{jk}e^{-B_{jk}(t-\tau_{jk})} \tag{3.10}$$

The information class identity is stored in a tag associated with each data report. This tag contains the necessary information required to reconstruct the infotential at the remote user's end. The tag in our case is completely characterized by constants A, B and time stamp τ_o where τ_o is the time at which the data report was recorded by the sensor node. A tag is 3-tuple entity. The tag for the k_{th} data report at the j_{th} sensor node is

$$\lambda_{jk} = (A_{jk}, B_{jk}, \tau_{o_{jk}}) \tag{3.11}$$

Because the data reports are large in size, it is not possible to transmit them over the acoustic channel as stated earlier. The acoustic channel will be used for broadcasting the λ_{jk} tags residing at a sensor node. The tags will be transmitted to a sink node. On the other hand, the optical channel will be used for transmitting the actual data reports to the AUV during its tour. The packets transmitted over the acoustic communication channel by the j_{th} sensor node are composed of the payload λ_{jk} (the VoI tag) and δ_{jk} (protocol, header information etc.). These packets are intended for delivery to a remote agent (sink node) and on the basis of this data, i.e. λ_{jk}, the sink node will plan a schedule of node visitation for the AUV. The packets transmitted on this channel are

$$\Gamma_j^{Acoustic} = \{(\lambda_{jk}, \delta_{jk})_1, \ldots, (\lambda_{jk}, \delta_{jk})_x\} \tag{3.12}$$

The packets transmitted over the optical communication channel by the i_{th} sensor node are composed of the d_{jk} (the data report), λ_{jk} (the VoI tag) and δ_{jk} (protocol, header information etc.). These packets have the actual recorded data of the events. The data is offloaded from the sensor nodes onto the AUV through this optical channel. The packets transmitted on this channel are

$$\Gamma_j^{Optical} = \{(d_{jk}, \lambda_{jk}, \delta_{jk})_1, \ldots, (d_{jk}, \lambda_{jk}, \delta_{jk})_y\} \tag{3.13}$$

As soon as the data is received and processed at the remote user's end, another time stamp $\tau_{f_{jk}}$ is assigned to the data report which helps in determining it's current VoI from the infotential. This substitution yields

$$\upsilon_{jk}^r = A_{jk} e^{-B_{jk}(\tau_{f_{jk}} - \tau_{o_{jk}})} \tag{3.14}$$

3.3.1 Using Multiple AUVs

The aforementioned formulation is for modeling a scenario which assumes there to be a single AUV. For the case where there are multiple AUVs, the sensor nodes in S are visited by AUVs in A

which are

$$A = \{a_1, a_2, \ldots, a_i, \ldots, a_a\} \tag{3.15}$$

The sensor nodes are distributed into a subsets and each of these subsets will subsequently be assigned to an AUV in A

$$\left(S_{All} = \{S_1, S_2, \ldots, S_i, \ldots, S_a\}\right) \wedge \left(S_i | S_i \subseteq S\right) \tag{3.16}$$

In this dissertation, in the chapter that addresses multiple AUVs, we divide the nodes into disjoint subsets such that there is no sensor node that has not been assigned to a subset, and hence, every node is uniquely assigned to an AUV such that

$$S_i \in S_{All} \left| \left(\bigcup_{i=1}^{a} S_i = S \right) \wedge \left(\left(S_x \cap S_y = \phi\right) \forall \left((x \neq y) \wedge (x, y \in \{1, 2, \ldots, a\})\right) \right) \right. \tag{3.17}$$

A node distribution function/algorithm β assigns each subset of nodes to an AUV in A in a bijective manner, i.e. it is a one-to-one correspondence

$$\beta : S_{All} \rightarrow A \tag{3.18}$$

In terms of the j_{th} sensor node and k_{th} data report, the infotentials are denoted with subscripts as

$$\upsilon^r_{jk}(t) = A_{jk} e^{-B_{jk}(t - \tau_{o_{jk}})} \tag{3.19}$$

3.3.2 Resurfacing Locations

From an information fusion perspective, either the AUV has the ability to gather reports and fuse them so as to make an informed decision for an actuation response, or, it might need to resurface frequently so as to transmit to a remote sink node which would perform the information fusion and actuation response initiation activity. If the latter is the case, then we have the following g resurfacing locations

$$R = \{r_1, r_2, \ldots, r_h, \ldots, r_g\} \tag{3.20}$$

It is on these resurfacing locations that the AUV will tranmsit data to the remote sink node.

3.3.3 An Alternate Scenario - Node Valuation instead Data Report Valuation

In the aforementioned model, the valuation is tied to data reports. A valuation can also be assigned to sensor nodes instead of data reports. In such a case it would the nodes that would be deemed as high-priority or low-priority. A mapping function α^{node} will assign the sensor nodes to the information classes.

$$\alpha^{node} : S \rightarrow C \tag{3.21}$$

The high-priority and low-priority infotentials to describe the nodes are

$$\begin{aligned} \upsilon_H^{sensor}(t) &= A_H e^{-B_H(t-\tau_{oH})} \\ \upsilon_L^{sensor}(t) &= A_L e^{-B_L(t-\tau_{oL})} \end{aligned} \tag{3.22}$$

In terms of the j_{th} sensor node, the infotential is described as

$$\upsilon_j^s(t) = A_j e^{-B_j(t-\tau_{o_j})} \tag{3.23}$$

3.4 Constructing VoI Model and Path Planning Problem

The goal is to extract the maximum amount of VoI and to minimize system and asset losses. To maximize the VoI collected, the path planners will need to determine a route that is efficient in terms of accumulating VoI from the system. In this section, we first define the VoI aggregation and maximization relations. Then we describe the role of final time stamp τ_f in the resurfacing versus non-resurfacing scenario. Then we give the path planning problem statements for the cases of single AUV, single AUV with resurfacing, and Multiple AUVs scenarios. Finally, we describe measures of VoI accumulated and VoI Lost and their role in determining path planning efficacy.

3.4.1 VoI maximization

Here we give the VoI accumulation and maximization relations. The two basic scenarios that we cover are the data report valuation and the node valuation. In this dissertation, we use the data report valuation model in the simulation but all the discussion and algorithms in this dissertation, without much modification, can be directly applied to the node valuation model.

3.4.1.1 Data Report Valuation

The VoI for the k_{th} data report at the j_{th} node is given as

$$\upsilon^{r}_{jk} = A_{jk} e^{-B_{jk}(\tau_{f_{jk}} - \tau_{o_{jk}})} \tag{3.24}$$

Single AUV:

The VoI collected from the j_{th} sensor node is the combined VoI of all data reports residing on the sensor node and is given as

$$\Upsilon_j = \sum_{k=1}^{d} \upsilon^{r}_{jk} = \sum_{k=1}^{d} A_{jk} e^{-B_{jk}(\tau_{f_j} - \tau_{o_{jk}})} \tag{3.25}$$

Here τ_{f_j} is the time at which the AUV visits the j_{th} sensor node. The total VoI accumulated from the UWSN by the AUV after it has visited all the sensor nodes based on a planned tour can be calculated as

$$\begin{gathered} \Upsilon_{Acc} = \sum_{j=1}^{n} \Upsilon_j . v_j^{visit} = \sum_{j=1}^{n} \sum_{k=1}^{d} \upsilon^{r}_{jk} . v_j^{visit} \\ \implies \Upsilon_{Acc} = \sum_{j=1}^{n} \sum_{k=1}^{d} A_{jk} e^{-B_{jk}(\tau_{f_j} - \tau_{o_{jk}})} . v_j^{visit} \end{gathered} \tag{3.26}$$

where,

$$v_j^{visit} = \begin{cases} 1 & \text{if } j_{th} \text{ node visited by AUV} \\ 0 & \text{otherwise} \end{cases}$$

Therefore, maximizing VoI in the case of single AUV is defined as

$$\Upsilon^{Single}_{Maximize} \rightarrow \max \sum_{j=1}^{n}\sum_{k=1}^{d} A_{jk} e^{-B_{jk}(\tau_{f_j}-\tau_{o_{jk}})}.v_j^{visit} \tag{3.27}$$

Multiple AUVs:

The VoI collected from the j_{th} sensor node by the i_{th} AUV is the combined VoI of all data reports residing on the sensor node and is given as

$$\Upsilon_{ij} = \sum_{k=1}^{d} \upsilon^{r}_{ijk} = \sum_{k=1}^{d} A_{jk} e^{-B_{jk}(\tau_{f_{ij}}-\tau_{o_{jk}})} \tag{3.28}$$

Here $\tau_{f_{ij}}$ is the time at which the i_{th} AUV visits the j_{th} sensor node. The VoI collected by the i_{th} AUV is

$$\Upsilon_{i} = \sum_{j=1}^{n} \Upsilon_{ij}.v_{ij}^{visit} = \sum_{j=1}^{n}\sum_{k=1}^{d} \upsilon^{r}_{ijk}.v_{ij}^{visit} \tag{3.29}$$

where,

$$v_{ij}^{visit} = \begin{cases} 1 & \text{if } j_{th} \text{ node visited by } i_{th} \text{ AUV} \\ 0 & \text{otherwise} \end{cases}$$

The total VoI accumulated from the UWSN by all the the AUVs after they have visited all the sensor nodes based on a planned tour can be calculated as

$$\begin{aligned} \Upsilon_{Acc} &= \sum_{i=1}^{a} \Upsilon_{i} = \sum_{i=1}^{a}\sum_{j=1}^{n} \Upsilon_{ij}.v_{ij}^{visit} \\ \implies \Upsilon_{Acc} &= \sum_{i=1}^{a}\sum_{j=1}^{n}\sum_{k=1}^{d} \upsilon^{r}_{ijk}.v_{ij}^{visit} \end{aligned}$$

$$\implies \Upsilon_{Acc} = \sum_{i=1}^{a}\sum_{j=1}^{n}\sum_{k=1}^{d} A_{jk}e^{-B_{jk}(\tau_{f_{ij}}-\tau_{o_{jk}})}.v_{ij}^{visit} \tag{3.30}$$

Therefore, maximizing VoI in the case of multiple AUVs is defined as

$$\Upsilon_{Maximize}^{Multiple} \rightarrow \max \sum_{i=1}^{a}\sum_{j=1}^{n}\sum_{k=1}^{d} A_{jk}e^{-B_{jk}(\tau_{f_{ij}}-\tau_{o_{jk}})}.v_{ij}^{visit} \tag{3.31}$$

Single AUV with Resurfacing:

The VoI collected by the AUV from the j_{th} sensor node and then transmitted to the sink node by the AUV at the h_{th} resurfacing location is given as

$$\Upsilon_{hj} = \sum_{k=1}^{d} \upsilon_{hjk}^{r} = \sum_{k=1}^{d} A_{jk}e^{-B_{jk}(\tau_{f_h}-\tau_{o_{jk}})} \tag{3.32}$$

Note the change in subscript from τ_{f_j} to τ_{f_h}. The time-stamp τ_{f_h} is the instant at which the AUV resurfaces at the h_{th} resurfacing location to transmit the data. The VoI delivered at the h_{th} resurfacing location is determined as

$$\Upsilon_h = \sum_{j=1}^{n} \Upsilon_{hj}.v_j^{visit} = \sum_{j=1}^{n}\sum_{k=1}^{d} \upsilon_{hjk}^{r} \tag{3.33}$$

where,

$$v_j^{visit} = \begin{cases} 1 & \text{if } j_{th} \text{ node visited by AUV} \\ 0 & \text{otherwise} \end{cases}$$

The total VoI accumulated from the UWSN by the AUV, after it has visited all the sensor nodes and resurfaced on certain locations based on a planned tour, can be calculated as

$$\begin{aligned}
\Upsilon_{Acc} &= \sum_{h=1}^{g} \Upsilon_{h}.l_{hj}^{visit} = \sum_{h=1}^{g}\sum_{j=1}^{n} \Upsilon_{hj}.v_{j}^{visit}.l_{hj}^{visit} \\
\Longrightarrow \Upsilon_{Acc} &= \sum_{h=1}^{g}\sum_{j=1}^{n}\sum_{k=1}^{d} \upsilon_{hjk}^{r}.v_{j}^{visit}.l_{hj}^{visit} \\
\Longrightarrow \Upsilon_{Acc} &= \sum_{h=1}^{g}\sum_{j=1}^{n}\sum_{k=1}^{d} A_{jk}e^{-B_{jk}(\tau_{f_h}-\tau_{o_{jk}})}.v_{j}^{visit}.l_{hj}^{visit} \qquad (3.34)
\end{aligned}$$

where,

$$l_{hj}^{visit} = \begin{cases} 1 & \text{if data for } j_{th} \text{ node is transmitted at } h_{th} \text{ resurfacing location} \\ 0 & \text{otherwise} \end{cases}$$

Therefore, maximizing VoI in the case of single AUV that resurfaces for transmitting data is defined as

$$\Upsilon_{Maximize}^{Resurface} \to \max \sum_{h=1}^{g}\sum_{j=1}^{n}\sum_{k=1}^{d} A_{jk}e^{-B_{jk}(\tau_{f_h}-\tau_{o_{jk}})}.v_{j}^{visit}.l_{hj}^{visit} \qquad (3.35)$$

3.4.1.2 Node Valuation

VoI in terms of node valuation for the j_{th} node is given as

$$\upsilon_{j}^{s} = A_{j}e^{-B_{j}(\tau_{f_j}-\tau_{o_j})} \qquad (3.36)$$

Single AUV:

For the scenario where valuation is made on sensor nodes instead of data reports, the VoI accumulation and VoI maximization definitions for a single AUV are

$$\begin{aligned} \Upsilon_{Acc} &= \sum_{j=1}^{n} \upsilon_{j}^{s}.v_{j}^{visit} \\ \Upsilon_{Maximize}^{Single} &\rightarrow \max \sum_{j=1}^{n} \upsilon_{j}^{s}.v_{j}^{visit} \end{aligned} \tag{3.37}$$

Multiple AUVs:

For multiple AUVs it translates into

$$\begin{aligned} \Upsilon_{Acc} &= \sum_{i=1}^{a} \sum_{j=1}^{n} \upsilon_{ij}^{s}.v_{ij}^{visit} \\ \Upsilon_{Maximize}^{Multiple} &\rightarrow \sum_{i=1}^{a} \sum_{j=1}^{n} \upsilon_{ij}^{s}.v_{ij}^{visit} \end{aligned} \tag{3.38}$$

Single AUV with Resurfacing:

For a single AUV that resurfaces the definitions are

$$\begin{aligned} \Upsilon_{Acc} &= \sum_{h=1}^{g} \sum_{j=1}^{n} \upsilon_{hj}^{s}.v_{j}^{visit}.l_{hj}^{visit} \\ \Upsilon_{Maximize}^{Resurface} &\rightarrow \sum_{h=1}^{g} \sum_{j=1}^{n} \upsilon_{hj}^{s}.v_{j}^{visit}.l_{hj}^{visit} \end{aligned} \tag{3.39}$$

3.4.2 Role of τ_f in Infotentials

The VoI for the k_{th} data report evaluated at the sink node, which can either be a remote sink node or the AUV itself, is given as

$$\upsilon_k^r = A_k e^{-B_k(\tau_{f_k} - \tau_{o_k})} \tag{3.40}$$

The time-stamp τ_f determines the VoI extracted from an information segment based on it's infotential. Therefore, VoI of a data report varies according to the time instant at which the information is gathered by an end-processing agent. In the given UWSN scenario, we have two different final time-stamp τ_f definitions. The definitions are based on who the end-processing agent is; the one who is responsible for triggering an actuation response after processing the data. If the AUV is equipped with the ability to process the data such that it can initiate an actuation response then τ_f is the time at which the AUV collects the data from the respective sensor node. But if the end-processing agent is above the sea surface, then τ_f is determined when the information is received by the end-processing agent (for which the AUV will have to resurface). The time stamp τ_f definition can crucially impact the design of AUV path planning algorithms for VoI maximization. In this dissertation, we assume the AUV to be the end-processing agent which implies that τ_f will be determined by instant at which the information segment was offloaded from the sensor node on to the AUV.

3.4.3 The Path Planning Problem

The problem definition for AUV path planning is to devise an algorithm that attempts to maximize VoI accumulated from the UWSN. More formally, given sensor nodes S and VoI profile $\Upsilon_{Acc}(t)$ of data reports D; what is the sequence of node visitation P_S in S that will result in the accumulation of VoI $\Upsilon_{Acc}^{Alg^{PP}}$

$$\left[\Upsilon_{Acc}^{Alg^{PP}} \leftarrow P_S\right] \leftarrow Alg^{PP}\left[S, D, \Upsilon_{Acc}(t)\right] \tag{3.41}$$

For multiple AUVs it translates into $P_{(S,A)}$ which is a mapping of nodes in S to AUVs in A i.e. which nodes s_j in S and in what sequence should mapped to an AUV a_i in A

$$\left[\Upsilon_{Acc}^{Alg^{PP}} \leftarrow P_{(S,A)}\right] \leftarrow Alg^{PP}\left[A, S, D, \Upsilon_{Acc}(t)\right] \tag{3.42}$$

For a single AUV that need to resurface, as described earlier, we introduce the set of resurfacing location R to the problem statement. The visitation schedule then becomes a sequence of nodes and resurfacing locations intertwined with other. The problem statement for this case is

$$\left[\Upsilon_{Acc}^{Alg^{PP}} \leftarrow P_{S+R}\right] \leftarrow Alg^{PP}\left[S, R, D, \Upsilon_{Acc}(t)\right] \tag{3.43}$$

where,

$\Upsilon_{Acc}^{Alg^{PP}}$ is the VoI accumulated from the sensor nodes S by employing the traversal sequence P_S,

$P_{(S,A)}$ is the set of all node visitation sequences for various AUVs determined by Alg^{PP} in 3.42,

P_S is the node visitation sequence determined by Alg^{PP} in 3.41,

P_{S+R} is the node visitation sequence intertwined with the resurfacing locations in Alg^{PP} in 3.43,

Alg^{PP} is a path planning algorithm that generates path P_S or $P_{(S,A)}$ such that $\Upsilon_{Acc}^{Alg^{PP}}$ is accumulated,

A is the set of all AUVs,

S is the set of all sensor nodes,

R is the set of all resurfacing locations,

D is the set of all data reports,

$\Upsilon_{Acc}(t)$ is the function total VoI accumulated.

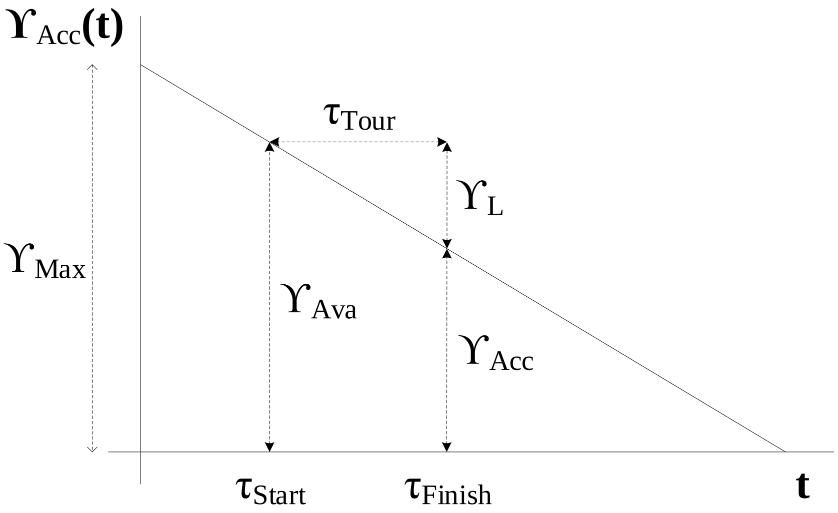


Figure 3.4: The figure shows various parameters and measures that we use for comparative analysis of path planning algorithms.

3.4.4 Path Planning Performance - VoI Accumulated versus VoI Lost

To determine the effect of a path planner on VoI we need to use a measure of VoI. The basic measures of performance that we use are VoI accumulated 'Υ_{Acc}' and VoI lost 'Υ_L'. To discuss what these measures are we refer to Figure 3.4. $\Upsilon_{Acc}(t)$ is the decaying VoI profile in the system and is represented in abstract terms by a straight line. It is a general statement on the depreciation of the valuation, whereas, in an actual situation the dynamics of this depreciation i.e. the actual shape of the curve, will be governed by system variables and type of information recorded. This chart assumes that no measurements are recorded after $t = 0$, hence, the chart only shows a monotonic decay in the valuation after $t = 0$. τ_{start} and τ_{finish} are the start and end times for the complete AUV tour. In our case, a tour is a visitation sequence that is a permutation on the set of all sensor nodes i.e. the tour is a 'simple path' on a graph that has the sensor nodes as vertices. Υ_{Ava} is the VoI available in the system at the start of the tour while the Υ_{Acc} is the VoI accumulated by the

AUV by the end of its tour. Υ_L, for example in the case of multiple AUVs, can be determined as

$$\Upsilon_L = \Upsilon_{Ava} - \Upsilon_{Acc} \tag{3.44}$$

$$\Upsilon_L = \sum_{i=1}^{a}\sum_{j=1}^{n}\sum_{k=1}^{d}\left[A_{jk}e^{-B_{jk}(\tau_{Start}-\tau_{o_{jk}})} - A_{jk}e^{-B_{jk}(\tau_{f_{ij}}-\tau_{o_{jk}})}\right] \tag{3.45}$$

The loss in Υ_{Acc} is a result of a combination of factors that can be attributed to physical system limitations such as the AUV speed, delay in starting time of the tour τ_{Start}, and inefficiencies resulting from the planned path. It is the path planning part of this problem that we want to explore in this dissertation. To filter out effects of a delay in τ_{Start} we use Υ_L as it is a measure of loss between the range $t = [\tau_{Start} : \tau_{Finish}]$. Any loss other than Υ_L is not a result of path planning inefficiencies. Υ_L gives a more clear picture in terms of comparative performance of various path planning techniques as compared to Υ_{Acc}.

CHAPTER 4: PATH PLANNING HEURISTICS

In this dissertation, path planning algorithms will be responsible for both allocating the sensor nodes in the UWSN to AUVs. Not only will they determine which AUV visits which sensor node, but they will also schedule the sequence in which these nodes will be visited. In this chapter, we propose path planning heuristics on the basis of which we design the path planning algorithms for the AUVs. The algorithms are formulated around one or more of the following heuristics: next node visit based on maximum VoI, H_{MaxVoI}; minimize tour time, H_{ShPath}; visit intermediate nodes, $H_{IntVisit}$; balance node distribution, $H_{NodeBal}$; balance VoI distribution, H_{VoIBal}; & partition the map, $H_{MapPart}$. Heuristics $H_{NodeBal}$, H_{VoIBal} & $H_{MapPart}$ specifically address the scheduling of multiple AUVs while heuristics H_{MaxVoI}, H_{ShPath} & $H_{IntVisit}$ are more generic and foundational in nature as they are part of the single and the multiple AUV path planning algorithms.

4.1 Path Planning with Infotentials

In this section, we propose various scenarios to understand the ramifications of the exponential infotential model in relation to a planned path for VoI collection. We use the same infotential model as in Equation 3.1

$$\upsilon(t) = Ae^{-B(t-\tau_o)}$$

Of the two aspects that we want to study, one is a comparative magnitude and decay-rate of infotentials, and the other is time. To study comparative magnitude and decay-rates, we use coefficients $\{A, B\}$ in the exponential infotential. To figure out the role of time we use t in the infotential. The analysis for valuation and decay coefficients $\{A, B\}$ is given in Section 4.1.1 and analysis for the role of time t in accumulating VoI is given in Sections 4.1.2 & 4.1.3.

4.1.1 Effect of Valuation and Decay Coefficients

Coefficients $\{A,B\}$ control the shape of the infotential. The goal is to study their effect on VoI collected through certain paths given that the paths are similar in construction in terms of inter-node distance. This will help us in isolating the role of time in terms of distance traveled in this study. Consider the scenario shown in Figure 4.1.1 (a) on the left-hand side. Let the VoI for sensor node x be $\upsilon_x(t)$ and the VoI for sensor node y be $\upsilon_y(t)$

$$\begin{aligned} \upsilon_x(t) &= A_x e^{-B_x(t-\tau_{o_x})} \\ \upsilon_y(t) &= A_y e^{-B_y(t-\tau_{o_y})} \end{aligned} \tag{4.1}$$

The VoI that can be accumulated from these nodes is calculated as

$$\begin{aligned} \Upsilon_{xy}(t_x,t_y) &= \upsilon_x(t_x)+\upsilon_y(t_y) \\ \implies \Upsilon_{xy}(t_x,t_y) &= A_x e^{-B_x(t_x-\tau_{o_x})} + A_y e^{-B_y(t_y-\tau_{o_y})} \end{aligned} \tag{4.2}$$

The mobile sink can traverse the nodes in two different ways. One is from node x to y and we denote the path as $m \to x \to y$ and the corresponding VoI gained from this path as

$$\Upsilon_{xy}^{m\to x\to y} = \upsilon_x(t_x=\tau_{mx}) + \upsilon_y(t_y=\tau_{mx}+\tau_{xy}) \tag{4.3}$$

The other path is one in which the mobile sink first visits node y and afterwards node x. This path is denoted in the superscript as $m \to y \to x$. The VoI accumulated in this case is

$$\Upsilon_{xy}^{m\to y\to x} = \upsilon_y(t_y = \tau_{my}) + \upsilon_x(t_x = \tau_{my} + \tau_{yx}) \tag{4.4}$$

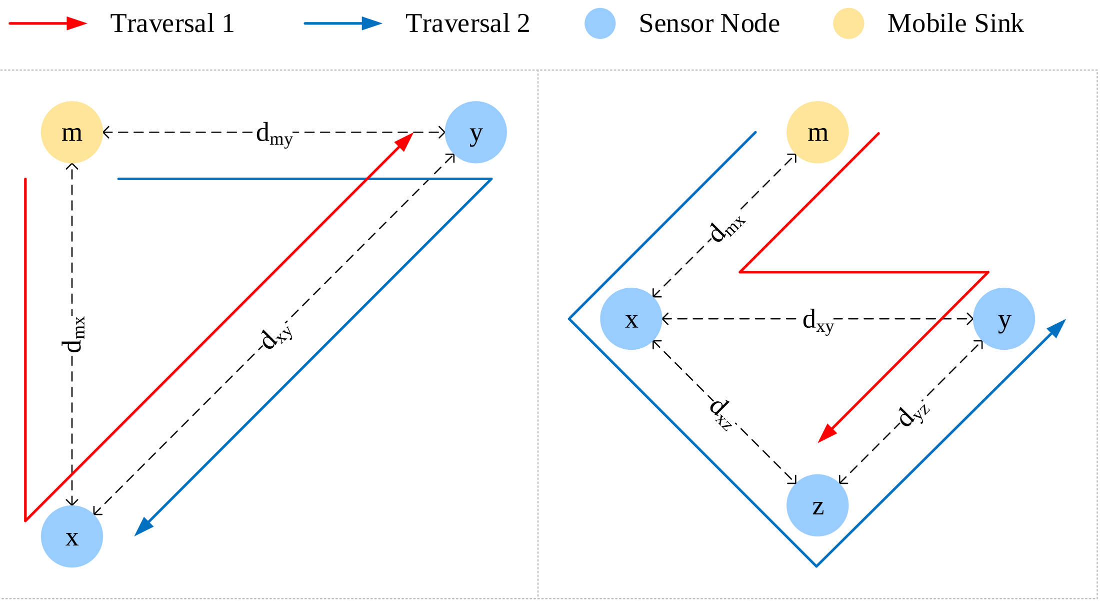


Figure 4.1: Traversals are the routes that the mobile sink m will use to navigate through the map to collect data from the sensor nodes x, y & z.
(a) The configuration on the left side will be used to study valuation and decay coefficients.
(b) The configuration on the right side will be used to understand the effect of tour time.

The values τ_{mx}, τ_{my}, τ_{xy} & τ_{yx} give the time that is required to traverse between the positions as marked in the subscripts. For example, τ_{mx} corresponds to the time taken to travel from the current opposition of the mobile sink m to the sensor node x as shown in Figure 4.1.1 (a). Similarly, τ_{xy} and τ_{yx} give inter-node travel time required by the mobile sink to move from node x to y.

In terms of the valuation coefficients and decay coefficients $\{(A_x, B_x), (A_y, B_y)\}$ we will discuss four different equality/inequality scenarios. These scenarios have been developed to show the effect of $\{(A_x, B_x), (A_y, B_y)\}$ on the path planning sequences $m \rightarrow x \rightarrow y$ and $m \rightarrow y \rightarrow x$. Therefore, we keep the travelling distances equal, and this gives $\tau_{mx} = \tau_{my}$, $\tau_{xy} = \tau_{yx}$ and $\tau_{o_x} = \tau_{o_y} = 0$ in these scenarios.

4.1.1.1 ***Scenario I-a:***

Let us assume the following scenario in which the infotentials are exactly the same in their construction

$$\begin{aligned}
A_x &= A_y = A \\
B_x &= B_y = B \\
\tau_{mx} &= \tau_{my} = \tau_1 > 0 \\
\tau_{xy} &= \tau_{yx} = \tau_2 > 0 \\
\tau_{o_x} &= \tau_{o_y} = 0
\end{aligned} \tag{4.5}$$

Under these conditions the VoI accumulated by vising the nodes in any order should be the same

$$\Upsilon_{xy}^{m\rightarrow x\rightarrow y} = \Upsilon_{xy}^{m\rightarrow y\rightarrow x} \tag{4.6}$$

We can verify by plugging in the values

$$\begin{aligned}
&\Upsilon_{xy}^{m\rightarrow x\rightarrow y} = \Upsilon_{xy}^{m\rightarrow y\rightarrow x} \\
\implies &\upsilon_x(t = \tau_{mx}) + \upsilon_y(t = \tau_{mx} + \tau_{xy}) = \upsilon_y(t = \tau_{my}) + \upsilon_x(t = \tau_{my} + \tau_{yx}) \\
\implies &Ae^{-B(\tau_{mx})} + Ae^{-B(\tau_{mx}+\tau_{xy})} = Ae^{-B(\tau_{my})} + Ae^{-B(\tau_{my}+\tau_{yx})} \\
\implies &Ae^{-B(\tau_1)} + Ae^{-B(\tau_1+\tau_2)} = Ae^{-B(\tau_1)} + Ae^{-B(\tau_1+\tau_2)}
\end{aligned} \tag{4.7}$$

Hence, if the infotentials are identical, then their is no priority in choosing which node to visit first in the given setting.

*4.1.1.2 **Scenario I-b:***

Here we assess the effect of difference in valuation co-efficient i.e. $A_x > A_y$. Let the constraints be the following in this case

$$\begin{aligned}
&A_x > A_y \\
&B_x = B_y = B \\
&\tau_{mx} = \tau_{my} = \tau_1 > 0 \\
&\tau_{xy} = \tau_{yx} = \tau_2 > 0 \\
&\tau_{o_x} = \tau_{o_y} = 0
\end{aligned} \tag{4.8}$$

Under these constraints we hypothesize that visiting node x before node y will result in a higher VoI accumulated. The following hold the following inequality to be true

$$\Upsilon_{xy}^{m\to x\to y} > \Upsilon_{xy}^{m\to y\to x} \tag{4.9}$$

We can verify this by substituting values to solve this inequality

$$\begin{aligned}
\Upsilon_{xy}^{m\to x\to y} &> \Upsilon_{xy}^{m\to y\to x} \\
\implies \upsilon_x(t_x = \tau_{mx}) + \upsilon_y(t_y = \tau_{mx} + \tau_{xy}) &> \upsilon_y(t_y = \tau_{my}) + \upsilon_x(t_x = \tau_{my} + \tau_{yx}) \\
\implies A_x e^{-B(\tau_{mx})} + A_y e^{-B(\tau_{mx}+\tau_{xy})} &> A_y e^{-B(\tau_{my})} + A_x e^{-B(\tau_{my}+\tau_{yx})} \\
\implies A_x e^{-B(\tau_1)} + A_y e^{-B(\tau_1+\tau_2)} &> A_y e^{-B(\tau_1)} + A_x e^{-B(\tau_1+\tau_2)}
\end{aligned}$$

$$\implies A_x(e^{-B(\tau_1)} - e^{-B(\tau_1+\tau_2)}) > A_y(e^{-B(\tau_1)} - e^{-B(\tau_1+\tau_2)})$$
$$\implies A_x > A_y \tag{4.10}$$

Therefore, the inequality holds true. This result can also be easily mapped to the scenario where $A_y > A_x$, thereby implying that node y should be visited first to accumulate higher VoI.

4.1.1.3 ***Scenario I-c:***

Here we assess the effect of difference in decay co-efficient i.e. $B_x > B_y$. Let the constraints be the following in this case

$$\begin{aligned} &A_x = A_y = A \\ &B_x > B_y \\ &\tau_{mx} = \tau_{my} = \tau_1 > 0 \\ &\tau_{xy} = \tau_{yx} = \tau_2 > 0 \\ &\tau_{o_x} = \tau_{o_y} = 0 \end{aligned} \tag{4.11}$$

Under these constraints we hypothesize that visiting node x before node y will result in a higher VoI accumulated. The following hold the following inequality to be true,

$$\Upsilon_{xy}^{m\to x\to y} > \Upsilon_{xy}^{m\to y\to x} \tag{4.12}$$

We verify this by solving this inequality

$$\Upsilon_{xy}^{m\to x\to y} > \Upsilon_{xy}^{m\to y\to x}$$
$$\implies \upsilon_x(t_x = \tau_{mx}) + \upsilon_y(t_y = \tau_{mx} + \tau_{xy}) > \upsilon_y(t_y = \tau_{my}) + \upsilon_x(t_x = \tau_{my} + \tau_{yx})$$

$$\begin{aligned}\implies Ae^{-B_x(\tau_{mx})} + Ae^{-B_y(\tau_{mx}+\tau_{xy})} &> Ae^{-B_y(\tau_{my})} + Ae^{-B_x(\tau_{my}+\tau_{yx})}\\ \implies e^{-B_x(\tau_1)} + e^{-B_y(\tau_1+\tau_2)} &> e^{-B_y(\tau_1)} + e^{-B_x(\tau_1+\tau_2)}\\ \implies e^{-B_x(\tau_1)} - e^{-B_y(\tau_1)} &> e^{-B_x(\tau_1+\tau_2)} - e^{-B_y(\tau_1+\tau_2)}\end{aligned}\tag{4.13}$$

Because $\tau_1 < (\tau_1 + \tau_2)$, therefore, we can conclude that the inequality holds true. Another way to understand is through the following rearrangement of terms

$$\begin{aligned}e^{-B_x(\tau_1)} - e^{-B_x(\tau_1+\tau_2)} &> e^{-B_y(\tau_1)} - e^{-B_y(\tau_1+\tau_2)}\\ \Delta_x(\tau_1,\tau_2) &> \Delta_y(\tau_1,\tau_2)\end{aligned}\tag{4.14}$$

Because $B_x > B_y$, hence, it contributes contributes to a faster decaying exponential which leads to a larger $\Delta(\tau_1,\tau_2)$. The inequality, therefore, holds true and validates the hypothesis in this scenario.

4.1.1.4 ***Scenario I-d:***

Here we assess the effect of both the valuation and decay co-efficient i.e. $A_x > A_y$ and $B_x > B_y$. Let the constraints be the following in this case

$$\begin{aligned}&A_x > A_y\\ &B_x > B_y\\ &\tau_{mx} = \tau_{my} = \tau_1 > 0\\ &\tau_{xy} = \tau_{yx} = \tau_2 > 0\\ &\tau_{o_x} = \tau_{o_y} = 0\end{aligned}\tag{4.15}$$

These constraints lead to us hypothesize that node y should be visited after node x has been visited.

$$\Upsilon_{xy}^{m\to x\to y} > \Upsilon_{xy}^{m\to y\to x} \tag{4.16}$$

Verifying the inequality yields

$$\begin{aligned}
&\Upsilon_{xy}^{m\to x\to y} > \Upsilon_{xy}^{m\to y\to x}\\
\implies &\upsilon_x(t_x = \tau_{mx}) + \upsilon_y(t_y = \tau_{mx} + \tau_{xy}) > \upsilon_y(t_y = \tau_{my}) + \upsilon_x(t_x = \tau_{my} + \tau_{yx})\\
\implies &A_x e^{-B_x(\tau_{mx})} + A_y e^{-B_y(\tau_{mx}+\tau_{xy})} > A_y e^{-B_y(\tau_{my})} + A_x e^{-B_x(\tau_{my}+\tau_{yx})}\\
\implies &A_x e^{-B_x(\tau_{mx})} - A_x e^{-B_x(\tau_{my}+\tau_{yx})} > A_y e^{-B_y(\tau_{mx}+\tau_{xy})} - A_y e^{-B_y(\tau_{my})}\\
\implies &A_x(e^{-B_x(\tau_1)} - e^{-B_x(\tau_1+\tau_2)}) > A_y(e^{-B_y(\tau_1)} - e^{-B_y(\tau_1+\tau_2)})\\
\implies &A_x\Delta_x(\tau_1, \tau_2) > A_y\Delta_y(\tau_1, \tau_2)
\end{aligned} \tag{4.17}$$

As inequality 4.10 and inequality 4.13 hold true, therefore, inequality 4.17 automatically holds true. We can show this by multiplying both sides of the inequalities with each other respectively to yield inequality 4.17.

Up till now, in all the aforementioned scenarios, the path that gives the higher VoI can be ascertained by just evaluating the relationship between the valuation and decay coefficients, i.e. $\{(A_x, B_x), (A_y, B_y)\}$, but the forthcoming scenario is slightly different in this regard.

4.1.1.5 ***Scenario I-e:***

Here we again assess the effect of both the valuation and decay co-efficient but in a way that one of the inequalities is reversed e.g. $A_x > A_y$ and $B_x < B_y$. Let the constraints be the following in this case

$$\begin{aligned}
&A_x > A_y \\
&B_x < B_y \\
&\tau_{mx} = \tau_{my} = \tau_1 > 0 \\
&\tau_{xy} = \tau_{yx} = \tau_2 > 0 \\
&\tau_{o_x} = \tau_{o_y} = 0
\end{aligned} \tag{4.18}$$

Under such constraints there is no conclusive hypothesis that which path is better. The reason is that infotential curves for $\Upsilon_{xy}^{m \to x \to y}$ and $\Upsilon_{xy}^{m \to y \to x}$ should intersect each other at a certain time τ_c. The point τ_c depends upon the coefficients $\{(A_x, B_x), (A_y, B_y)\}$. Still, the following can be ascertained with guarantee

$$\begin{aligned}
\tau_1 \geq \tau_c &\implies \Upsilon_{xy}^{m \to x \to y} < \Upsilon_{xy}^{m \to y \to x} \\
\tau_1 + \tau_2 \leq \tau_c &\implies \Upsilon_{xy}^{m \to x \to y} > \Upsilon_{xy}^{m \to y \to x}
\end{aligned} \tag{4.19}$$

But for $\tau_1 < \tau_c < \tau_1 + \tau_2$ we cannot determine the better path by just looking at the inter-coefficient $\{(A_x, B_x), (A_y, B_y)\}$ relationships. Instead, we will have to plug in the actual values of τ_1 and τ_2 to figure out which node would be better to visit first for accumulating higher VoI.

These are pretty basic scenarios that shed light on the behavior of infotentials in regards to path planning and VoI collected. This gives us the ability to derive intuitive heuristics for VoI based path planning. For example, we can say that creating a VoI-aware path planner, e.g. one that greedily chooses which node to visit based on the time-based VoI offer at the sensor nodes, should yield a higher amount of VoI as compared to a path planner that is not aware of the time-based VoI being offered.

4.1.2 Effect of Tour Time

To study time we will use identical infotentials but will vary the inter-node distance. In this way, we will be able to see the effect of time in terms of distance traveled on VoI collected. Consider the scenario shown in Figure 4.1.1 (b). Let the VoI for sensor node x be $\upsilon_x(t)$, the VoI for sensor node y be $\upsilon_y(t)$ and the VoI for sensor node z be $\upsilon_z(t)$

$$\begin{aligned}
\upsilon_x(t) &= A_x e^{-B_x(t-\tau_{ox})} \\
\upsilon_y(t) &= A_y e^{-B_y(t-\tau_{oy})} \\
\upsilon_z(t) &= A_z e^{-B_z(t-\tau_{oz})}
\end{aligned} \tag{4.20}$$

The VoI that can be accumulated from these nodes is calculated as

$$\begin{aligned}
&\Upsilon_{xyz}(t_x,t_y,t_z) = \upsilon_x(t_x) + \upsilon_y(t_y) + \upsilon_z(t_z) \\
\Longrightarrow\ &\Upsilon_{xyz}(t_x,t_y,t_z) = A_x e^{-B_x(t_x-\tau_{ox})} + A_y e^{-B_y(t_y-\tau_{oy})} + A_z e^{-B_z(t_z-\tau_{oz})}
\end{aligned} \tag{4.21}$$

Let us consider only two ways in which the mobile sink can traverse the nodes. The first path is denoted as $m \rightarrow x \rightarrow y \rightarrow z$ from which the VoI accumulated is

$$\Upsilon_{xyz}^{m\rightarrow x\rightarrow y\rightarrow z} = \upsilon_x(t_x = \tau_{mx}) + \upsilon_y(t_y = \tau_{mx} + \tau_{xy}) + \upsilon_z(t_z = \tau_{mx} + \tau_{xy} + \tau_{yz}) \tag{4.22}$$

The other path is denoted in the superscript as $m \rightarrow x \rightarrow z \rightarrow y$ and the VoI accumulated from traversing this path is

$$\Upsilon_{xyz}^{m\rightarrow x\rightarrow z\rightarrow y} = \upsilon_x(t_x = \tau_{mx}) + \upsilon_z(t_z = \tau_{mx} + \tau_{xz}) + \upsilon_y(t_y = \tau_{mx} + \tau_{xz} + \tau_{zy}) \tag{4.23}$$

In this scenario we wish to study the effect of tour time, therefore, we set τ_{xy}, τ_{yz}, τ_{zy} and τ_{xz} in such a way that one tour is shorter than the other.

4.1.2.1 ***Scenario II:***

Let us have the following set of constraints where the valuation and decay coefficients are same while $\tau_{xy} > \tau_{xz} = \tau_{yz}$

$$\begin{aligned} A_x = A_y = A_z = A \\ B_x = B_y = B_z = B, \\ \tau_{mx} = \tau_{xz} = \tau_{zy} = \tau_{yz} = \tau_1 > 0 \\ \tau_{xy} = \tau_2 > \tau_1 > 0 \\ \tau_{o_x} = \tau_{o_y} = \tau_{o_z} = 0 \end{aligned} \tag{4.24}$$

$m \rightarrow x \rightarrow z \rightarrow y$ is shorter in terms of distance as compared to $m \rightarrow x \rightarrow y \rightarrow z$. We hypothesize that the shorter tour $m \rightarrow x \rightarrow z \rightarrow y$ should yield more VoI than the tour $m \rightarrow x \rightarrow y \rightarrow z$

$$\Upsilon_{xyz}^{m \rightarrow x \rightarrow z \rightarrow y} > \Upsilon_{xyz}^{m \rightarrow x \rightarrow y \rightarrow z} \tag{4.25}$$

We verify by substitution

$$\begin{aligned} \Upsilon_{xyz}^{m \rightarrow x \rightarrow z \rightarrow y} > \Upsilon_{xyz}^{m \rightarrow x \rightarrow y \rightarrow z} \\ \implies \upsilon_x(t_x = \tau_1) + \upsilon_z(t_z = 2\tau_1) + \upsilon_y(t_y = 3\tau_1) > \upsilon_x(t_x = \tau_1) + \upsilon_y(t_y = \tau_1 + \tau_2) + \upsilon_z(t_z = 2\tau_1 + \tau_2) \\ \implies Ae^{-B(\tau_1)} + Ae^{-B(2\tau_1)} + Ae^{-B(3\tau_1)} > Ae^{-B(\tau_1)} + Ae^{-B(\tau_1+\tau_2)} + Ae^{-B(2\tau_1+\tau_2)} \\ \implies e^{-B(2\tau_1)} > e^{-B(\tau_1+\tau_2)} \end{aligned} \tag{4.26}$$

As $\tau_2 > \tau_1$, therefore, the inequality 4.26 holds true. This observation implies that if the infotentials are identical, then a path planning algorithms based on a shortest distance or minimal time heuristic would give the highest amount of VoI collected. Note that although the heuristic is not explicitly aware of the VoI on the sensor nodes, yet it should perform well because of its influence on time, which is a variable in the infotential.

4.1.3 Navigating Tours

The goal is to show that there can be smart choices one can take will developing path planning algorithms. The analysis in this section is similar to the minimal time analysis in Section 4.1.2, but it slightly differs in the aspect that we will not use identical infotentials. This will help us in developing an intermediate node visitation heuristic based on in which we can develop algorithms that are primarily VoI greedy (Section 4.1.1), but can still make an attempt to minimize tour time (Section 4.1.2), thereby, improving VoI collected.

Consider the scenario in shown in Figure 4.2. The source and destination in this case are s_2 and s_{23} respectively. We denote the nodes s_2, s_9, s_{16}, s_{23} by w, x, y, z respectively. Similarly, let the VoI for sensor nodes s_2, s_9, s_{16}, s_{23} be $\upsilon_w(t), \upsilon_x(t), \upsilon_y(t), \upsilon_z(t)$ respectively

$$\begin{aligned}
\upsilon_w(t) &= A_w e^{-B_w(t-\tau_{ow})} \\
\upsilon_x(t) &= A_x e^{-B_x(t-\tau_{ox})} \\
\upsilon_y(t) &= A_y e^{-B_y(t-\tau_{oy})} \\
\upsilon_z(t) &= A_z e^{-B_z(t-\tau_{oz})} \qquad (4.27)
\end{aligned}$$

The VoI that can be accumulated from these nodes is calculated as

$$\begin{aligned}
&\Upsilon_{wxyz}(t_w,t_x,t_y,t_z) = \upsilon_w(t_w) + \upsilon_x(t_x) + \upsilon_y(t_y) + \upsilon_z(t_z) \\
\implies &\Upsilon_{wxyz}(t_w,t_x,t_y,t_z) = A_w e^{-B_w(t_w-\tau_{ow})} + A_x e^{-B_x(t_x-\tau_{ox})} + A_y e^{-B_y(t_y-\tau_{oy})} + A_z e^{-B_z(t_z-\tau_{oz})} \qquad (4.28)
\end{aligned}$$

The inter-node distance among all the diagonal nodes is τ_d. Let us consider three ways in which the mobile sink can traverse the nodes. The paths are denoted as $m \to w \to x \to y \to z$, $m \to w \to z \to y \to x$ and $m \to w \to z \to x \to y$. The VoI accumulated from these paths is determined as

$$\begin{aligned}
\Upsilon_{wxyz}^{m\to w\to x\to y\to z} &= \upsilon_w(t_w = \tau_d) + \upsilon_x(t_x = 2\tau_d) + \upsilon_y(t_y = 3\tau_d) + \upsilon_z(t_z = 4\tau_d) \\
\Upsilon_{wxyz}^{m\to w\to z\to y\to x} &= \upsilon_w(t_w = \tau_d) + \upsilon_z(t_z = 4\tau_d) + \upsilon_y(t_y = 5\tau_d) + \upsilon_x(t_x = 6\tau_d) \\
\Upsilon_{wxyz}^{m\to w\to z\to x\to y} &= \upsilon_w(t_w = \tau_d) + \upsilon_z(t_z = 4\tau_d) + \upsilon_x(t_x = 6\tau_d) + \upsilon_y(t_y = 7\tau_d) \qquad (4.29)
\end{aligned}$$

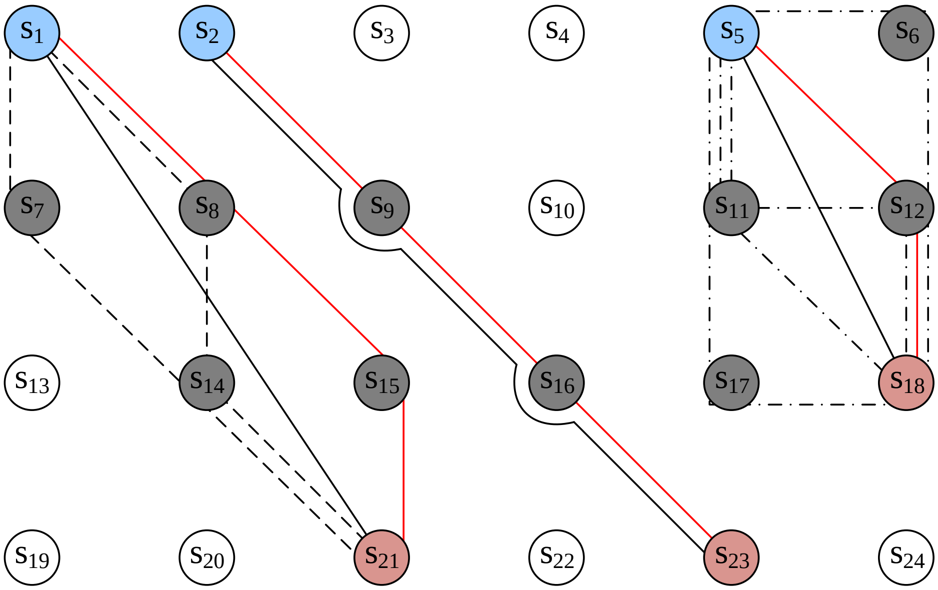


Figure 4.2: Alternate source to destination routes with approximately equal lengths.

4.1.3.1 ***Scenario III:***

Let us have the following set of constraints where $\tau_{mw} = \tau_{wx} = \tau_{xy} = \tau_{yz} = \tau_{zy} = \tau_{yx}$

$$\begin{aligned} \tau_{mw} = \tau_{wx} = \tau_{xy} = \tau_{yz} = \tau_{zy} = \tau_{yx} = \tau_d > 0 \\ \tau_{o_w} = \tau_{o_x} = \tau_{o_y} = \tau_{o_z} = 0 \end{aligned} \tag{4.30}$$

We hypothesize that

$$\Upsilon_{wxyz}^{m\to w\to x\to y\to z} > \Upsilon_{wxyz}^{m\to w\to z\to y\to x} > \Upsilon_{wxyz}^{m\to w\to z\to x\to y} \tag{4.31}$$

We separately verify two segments of inequality 4.31 by substitution

$$\begin{aligned} \Upsilon_{wxyz}^{m\to w\to x\to y\to z} &> \Upsilon_{wxyz}^{m\to w\to z\to y\to x} \\ \implies \upsilon_x(t_x = 2\tau_d) + \upsilon_y(t_y = 3\tau_d) &> \upsilon_y(t_y = 5\tau_d) + \upsilon_x(t_x = 6\tau_d) \\ \implies A_x e^{-2B_x\tau_d} + A_y e^{-3B_y\tau_d} &> A_x e^{-6B_x\tau_d} + A_y e^{-5B_y\tau_d} \end{aligned} \tag{4.32}$$

$$\begin{aligned} \Upsilon_{wxyz}^{m\to w\to z\to y\to x} &> \Upsilon_{wxyz}^{m\to w\to z\to x\to y} \\ \implies \upsilon_x(t_x = 5\tau_d) &> \upsilon_x(t_x = 7\tau_d) \\ \implies A_x e^{-5B_x\tau_d} &> A_x e^{-7B_x\tau_d} \end{aligned} \tag{4.33}$$

As $A_x e^{-2B_x\tau_d} > A_x e^{-6B_x\tau_d}$ and $A_y e^{-3B_y\tau_d} > A_y e^{-5B_y\tau_d}$, hence, the inequality 4.32 holds true. Inequality 4.32 also holds true.

Just as in the Section 4.1.2, this is a tour time and distance traveled analysis from which we can again infer that developing time minimization heuristics could aid in improving the amount of VoI collected.

4.1.3.2 Selection of Intermediate Nodes

The source and destination nodes in the aforementioned scenario were s_2 and s_{23} as shown in Figure 4.2. The intermediate node selection for this route is relatively straightforward as the nodes chosen for this purpose, s_9 and s_{16}, are directly on the straight line route between s_2 and s_{23}. Visiting nodes s_9 and s_{16} does not change the travelling distance between s_2 and s_{23}. Alternatively, consider the scenario where the source and destination are s_5 and s_{18}. There are no nodes directly on the path from s_5 to s_{18}. We can use a distance based metric to determine the intermediate nodes.

For thus, we first define a neighborhood, and then use a distance based metric to select a node from this neighborhood. We use a Moore neighborhood definition for the given mesh arrangement, e.g. the neighborhood for s_8 is $N_{s_8} = \{s_1, s_2, s_3, s_7, s_9, s_{13}, s_{14}, s_{15}\}$ and for s_5 is $N_{s_5} = \{s_4, s_6, s_{10}, s_{11}, s_{12}\}$. Even though we use a mesh arrangement of nodes to illustrate the concept, this concept can be generalized to other arrangements as well. The general idea is to choose the intermediate node from a subset of nodes, i.e. the neighborhood, based on a distance metric.

Let s_r, s_d and s_i be the source, destination and the intermediate node respectively. Their inter-node distances are defined as d_{rd}, d_{ri} and d_{id}. One metric can be to choose the next node based on $\{s_i | (s_i \in N_{s_r}) \wedge (d_{id} < d_{rd})\}$. This means that the intermediate node chosen should be one whose distance from the destination node (d_{id}) should be smaller than the distance between the source node and destination node (d_{rd}). All dotted lines and solid red lines in Figure 4.2 fulfill this metric.

Another metric can be $\{s_i | (s_i \in N_{s_r}) \wedge (d_{id} = min\{d_{id} | d_{id} < d_{rd}\})\}$ which implies that the intermediate node chosen from the neighborhood should have the smallest distance to the destination node. The solid red lines in Figure 4.2 represent all such paths. We use this definition for intermediate node visitation in this dissertation. Note that for the source destination pair s_1 and s_{21}, the three intermediate paths shown have the same length but only the red one fits the second definition.

4.2 Heuristics for VoI based Path Planning

4.2.1 *Next Node Visit based on VoI Maximization - H_{MaxVoI}*

This heuristic suggests that the mobile sink should visit nodes in a prioritized sequence based upon the amount of VoI being offered at the sensor nodes. It is a greedy heuristic and is based on observations from the discussion in Section 4.1.1. This heuristic attempts to maximize the amount of value collected at every node visit. It does it by scheduling the next visit to the node that has the highest VoI to offer. In Figure 4.3 we have nodes with priority marked as P_x, such that in terms of VoI they have the precedence $\Upsilon_{P_1} > \Upsilon_{P_2} > \Upsilon_{P_3} > \Upsilon_{P_4} > \Upsilon_{P_5}$. The blue colored path in the right hand side of the Figure 4.3 maintains this precedence by following the sequence $P_1 \rightarrow P_2 \rightarrow P_3 \rightarrow P_4 \rightarrow P_5$. The path on the left hand side violates this precedence by traversing the nodes in the sequence $P_3 \rightarrow P_5 \rightarrow P_4 \rightarrow P_1 \rightarrow P_2$.

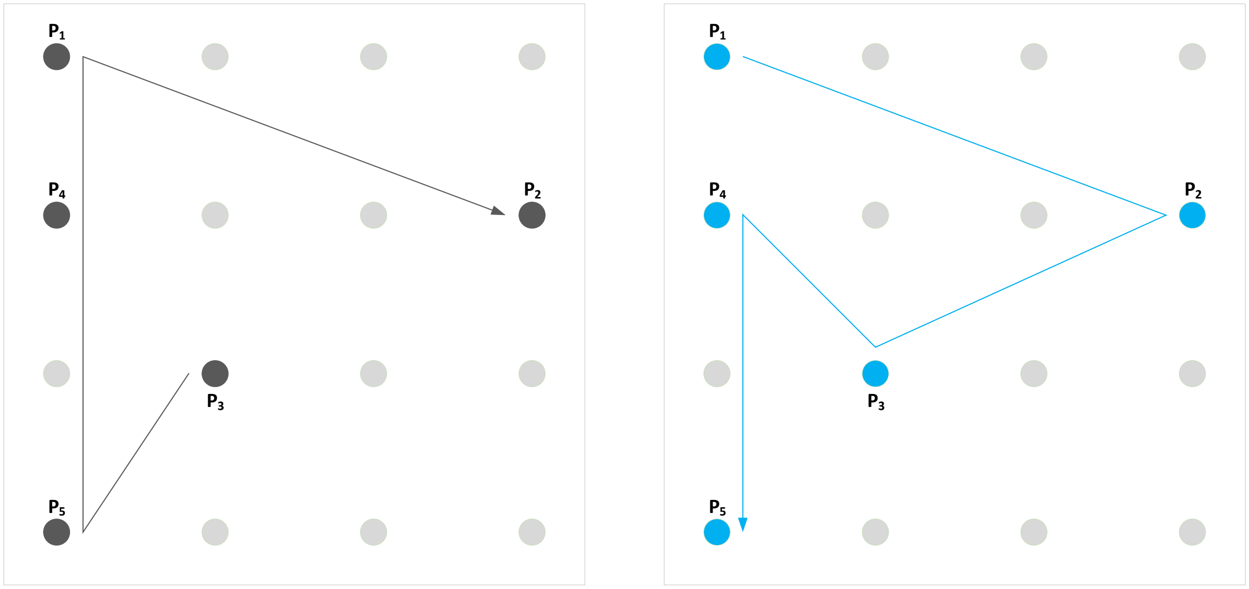


Figure 4.3: Heuristic H_{MaxVoI} - VoI Maximization

4.2.2 *Minimize Tour Time for Path Traversal - H_{ShPath}*

The proposal in this heuristic is to minimize the tour time of the AUV because time has been shown to be a factor that affects VoI collected. Based on the discussion in Section 4.1.2, we propose the heuristic H_{ShPath} which is to find the shortest traveling path given a set of sensor nodes. An example of this is shown on the right-hand side of Figure 4.4. On the contrary, a non-optimal path in terms of time is shown on the left-hand side of Figure 4.4.

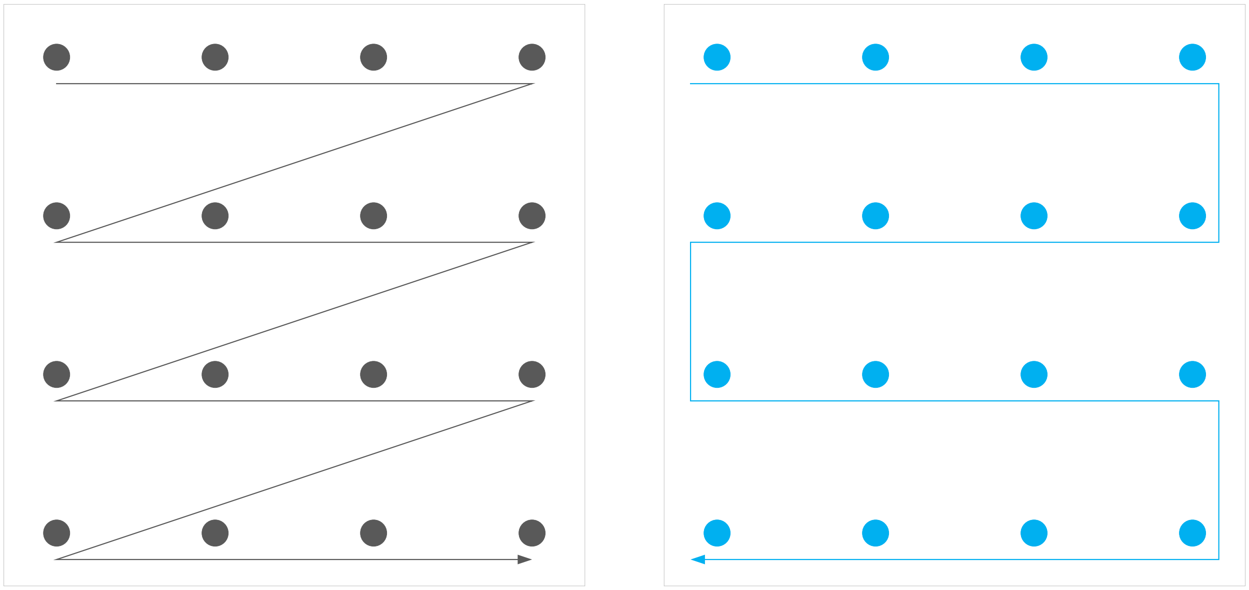


Figure 4.4: Heuristic H_{ShPath} - Shortest Path

4.2.3 *Visit Intermediate Nodes - $H_{IntVisit}$*

This heuristic is based on the discussion in Section 4.1.3. According to this heuristic, tours should be planned in a manner that nodes lying near the path of a source-destination pair should be visited also along the tour. Let us imagine a scenario where nodes are visited in-order of the VoI they offer just as in H_{MaxVoI}. It might be useful that we couple it with a time heuristic as suggested by e.g. H_{ShPath} to further improve the VoI collected. This will help in minimizing the tour time and, hence, VoI loss by avoiding delayed visits to nodes whose visitation at an earlier point in time would have had been less taxing in terms of VoI collected. This also helps in reducing fuel expenditure. In the aforementioned scenario, $H_{IntVisit}$ can be seen as being applied as a combination H_{MaxVoI} followed by H_{ShPath}. The definition of nodes encountered on the path can be determined, for example, by choosing the intermediate node which is in the neighborhood of current source node and is also closer to the destination. This definition is applied recursively by setting the current intermediate node to be the next source node and then choosing the next intermediate node from the neighborhood of the current intermediate node.

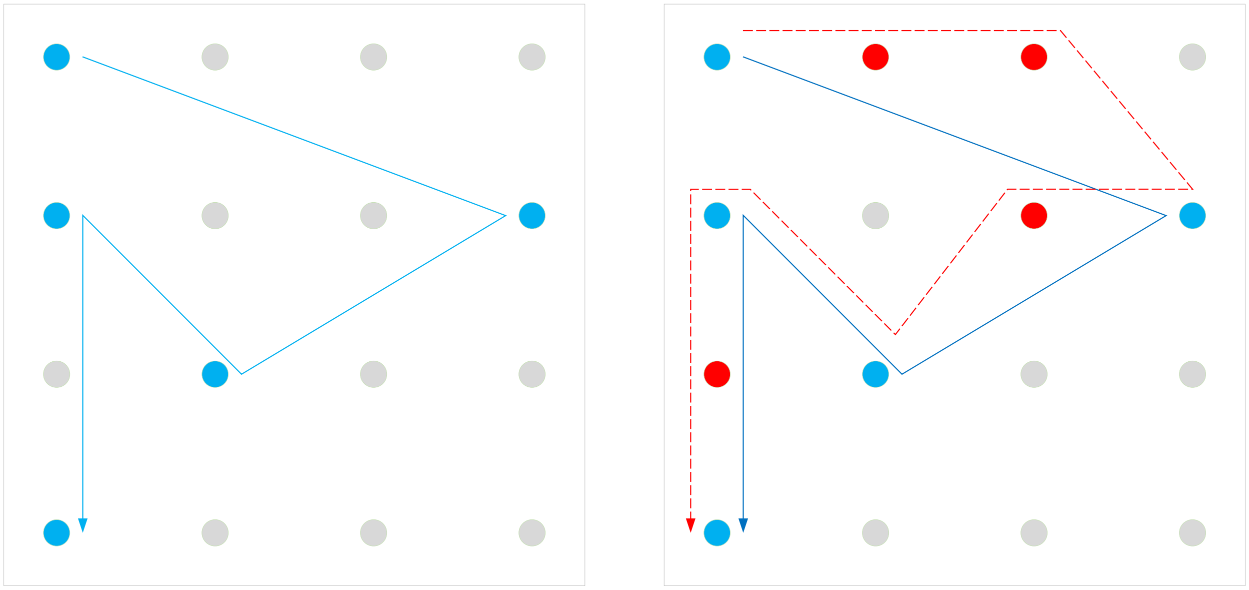


Figure 4.5: Heuristic $H_{IntVisit}$ (Intermediate Node Visitation)

4.2.4 Load Balancing in terms of Nodes Visited - $H_{NodeBal}$

This heuristic is for multiple AUV path planning. It enforces an identical allocation in terms of the number of nodes assigned to each AUV for their respective tours. The intuition behind this heuristic is that it can help in improving the chances of minimizing tour time for data collection. We argue that this is because the chance of some AUVs collecting data while others sitting idle due to finished with their visitation tasks is minimized because of a more balanced distribution of nodes. On the left-hand side in the Figure 4.6 the blue path has lesser nodes as compared to the grey path, i.e. 5 nodes are covered by the blue path while 11 nodes visited by the grey path. Note how the blue path is much shorter than the grey path. While on the right-hand side of Figure 4.6, the blue dotted line shows additional nodes assigned to the blue path, by off-loading them from the grey path, in order to balance the tours, thereby, giving both tours eight nodes each.

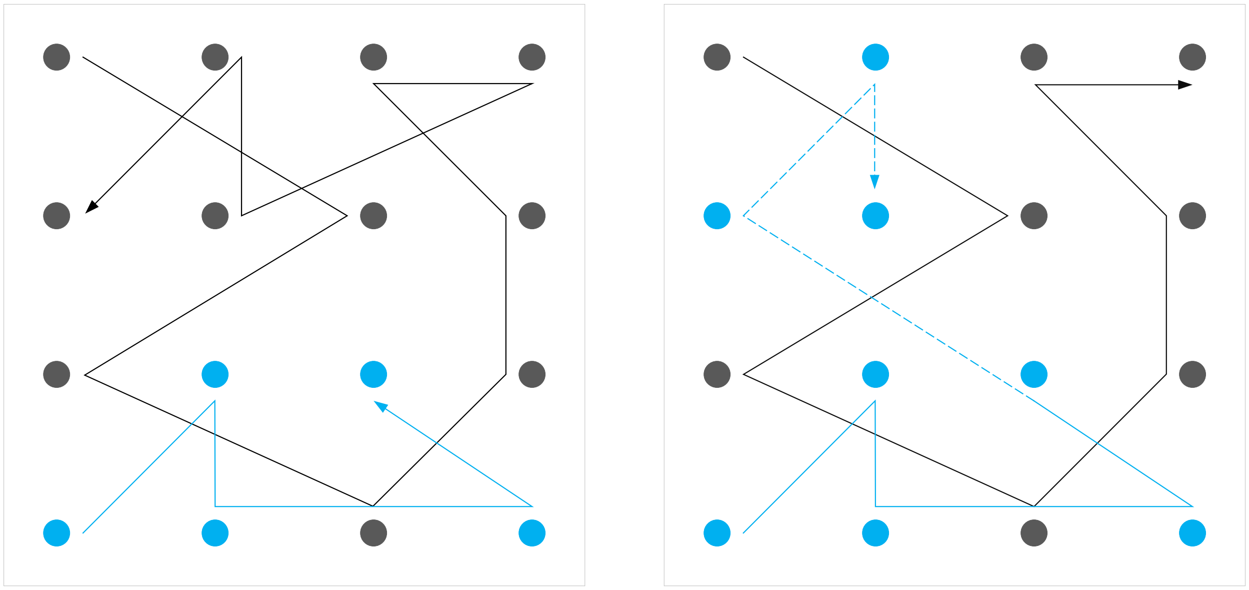


Figure 4.6: Heuristic $H_{NodeBal}$ (Node Balancing)

4.2.5 Balanced Distribution of Nodes in terms of VoI - H_{VoIBal}

This heuristic suggests that the nodes should be assigned to AUVs in a round-robin fashion such that at each iteration the node with the highest VoI at offer is chosen to be assigned to the next AUV. In this way, no single AUV will be scheduled to visit a disproportionate number of high priority nodes, i.e. nodes that have a higher VoI than usual. This, in a certain manner, imitates the H_{MaxVoI} heuristic by ensuring that nodes with higher priority will be scheduled to be visited earlier, thereby, collecting higher VoI. In Figure 4.7, we have labeled three nodes as C_H colored in red and indicating a higher priority, while three are colored in blue and marked as C_L so as to indicate a lower priority. On the left-hand side of Figure 4.7, we have two paths uniquely assigned to the C_H and C_L nodes, thereby, violating VoI balancing. While on the right-hand side in Figure 4.7, we have two paths that have a more balanced distribution of high and low priority nodes.

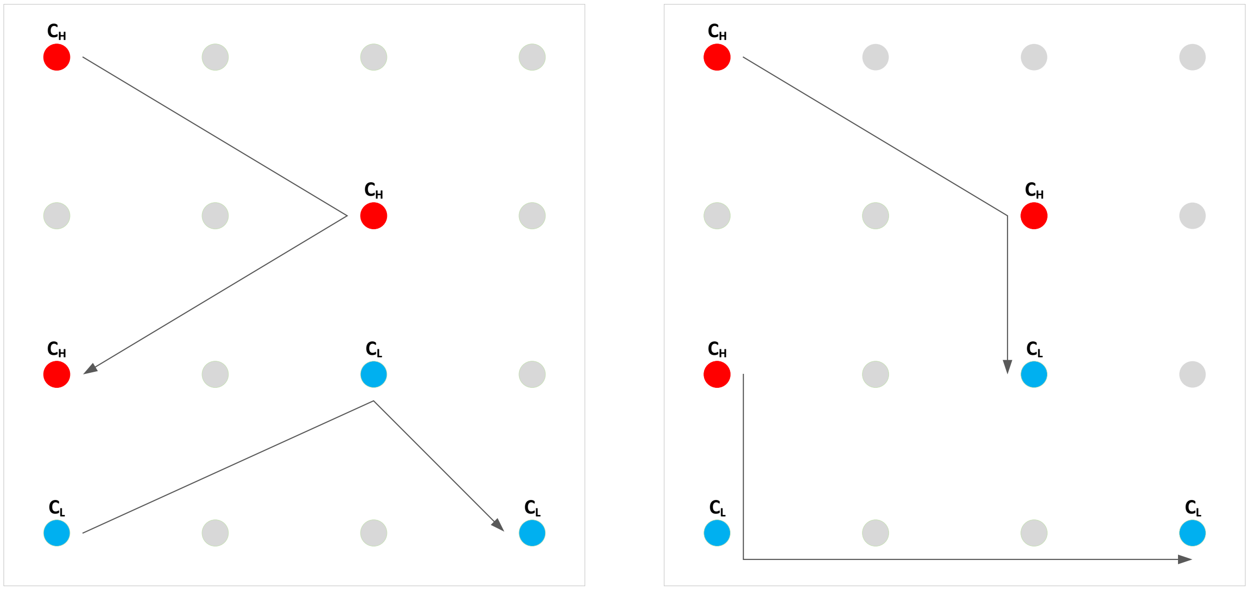


Figure 4.7: Heuristic H_{VoIBal} - VoI Balancing

4.2.6 Partitioning Map on basis of Node Proximity - $H_{MapPart}$

The goal with $H_{MapPart}$ is to reduce the average traveling time of each AUV by creating map partitions. This reduction in travel time should improve the overall VoI collection. This heuristic proposes to partition the map into non-overlapping segments such that each segment is uniquely assigned to an AUV, i.e. there is a one-to-one mapping between the partitioned segments and the AUVs. The partitions in the mesh arrangement are such that nodes in a partition are reachable to one another by recursively traversing one-hop through their neighbors.

This heuristic should reduce the map traversal time by reducing the travel time for each AUV on average. This reduction in travel time is because each AUV now has a lesser area to cover because of the partitioned regions that are smaller in size as compared to the whole map.

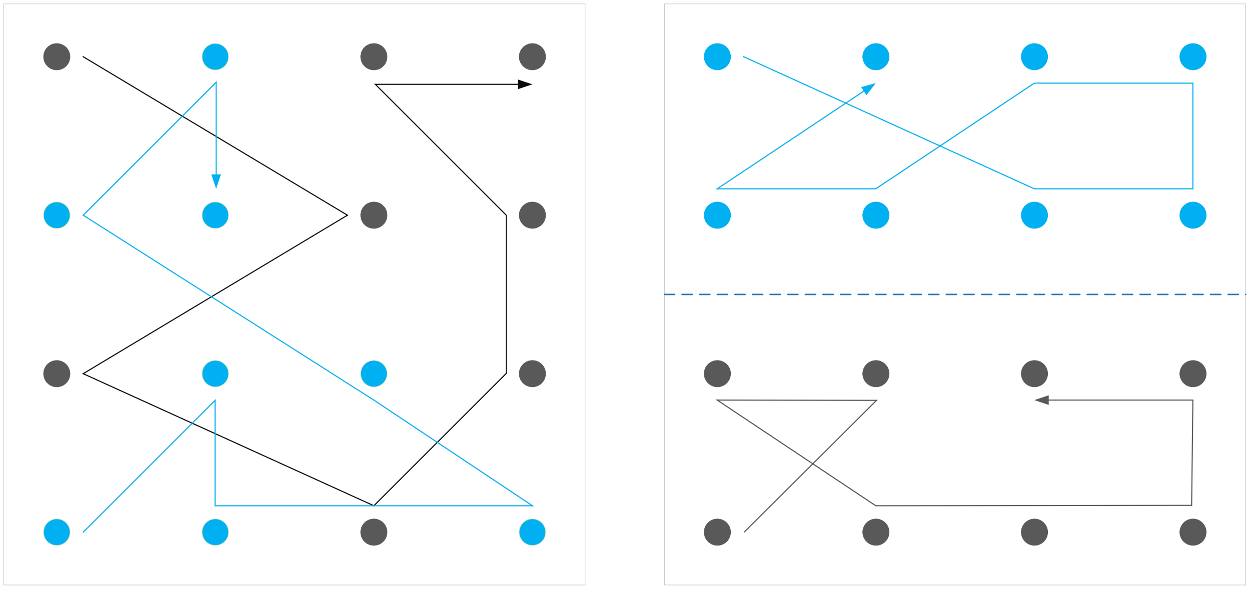


Figure 4.8: Heuristic $H_{MapPart}$ - Map Partitioning

Another way to understand this is that if *N* number of nodes are concentrated in a smaller region, then it is more likely that the AUV will travel a shorter distance to traverse all of these nodes. This is illustrated by the paths shown in Figure 4.8, all of which are assigned eights nodes each, but the paths on the left-hand side are longer then the ones on the right-hand side.

Partitioned maps leads to shorter travel distances for each AUV and, therefore, on a collective basis, the map is traversed completely in a shorter amount of time by all the AUVs.

CHAPTER 5: SINGLE AUV PATH PLANNING

5.1 VoI Maximization and Path Planning Problem

In this chapter we will develop path planning algorithms that will address the following VoI accumulation and maximization definitions, as given in Equation 3.26 & 3.27

$$\Upsilon_{Acc} = \sum_{j=1}^{n}\sum_{k=1}^{d} A_{jk} e^{-B_{jk}(\tau_{f_j} - \tau_{o_{jk}})} . v_j^{visit}$$

$$\Upsilon_{Maximize}^{Single} \rightarrow \max \sum_{j=1}^{n}\sum_{k=1}^{d} A_{jk} e^{-B_{jk}(\tau_{f_j} - \tau_{o_{jk}})} . v_j^{visit}$$

The path planning problem statement we will solve in this chapter is described by Definition 3.41

$$\left[\Upsilon_{Acc}^{Alg^{PP}} \leftarrow P_S\right] \leftarrow Alg^{PP}\left[S, D, \Upsilon_{Acc}(t)\right]$$

Where,

$\Upsilon_{Acc}^{Alg^{PP}}$ is the VoI accumulated from the sensor nodes S by employing the traversal sequence P_S,

P_S is the node visitation sequence determined by Alg^{PP},

Alg^{PP} is a path planning algorithm that generates path P_S or $P_{(S,A)}$ such that $\Upsilon_{Acc}^{Alg^{PP}}$ is accumulated,

S is the set of all sensor nodes,

D is the set of all data reports,

$\Upsilon_{Acc}(t)$ is the function total VoI accumulated.

5.2 Path Planning Algorithms

The path planning algorithms provide us with the sequence of nodes which the AUV will traverse for data collection. As suggested earlier, the sequence of node visitation by the AUV will affect the overall accumulated VoI. We propose a few different path planning algorithms for experimentation and analysis. The first algorithm is the Lawn-Mower path planner (*LPP*) based on minimizing total tour time. The Greedy path planner (*GPP*) is based on accumulating VoI in a greedy fashion. Greedy with Intermediate-Node-Visitation (*GIPP*) is a variant which is based on greedily accumulating VoI while aiming to minimize the tour time. Hybrid path planner (*HPP*) is a combination of *LPP* and *GPP*. Hybrid with Intermediate-Node-Visitation (*HIPP*) is a combination of *LPP* and *GIPP*. The Random path planner (*RPP*) is used as a baseline algorithm for the analysis purposes.

5.2.1 Lawn-Mower Path Planner - LPP

The *LPP* algorithm is based on a pre-computing strategy. It determines the route before-hand and it does not take into account the VoI profiles of the sensor nodes for its path planning decisions. The crux of this algorithm is to find the most optimal tour in terms of time traveled. This, in essence, is like solving the traveling salesman problem (TSP). The title Lawn-Mower is motivated by the analogy that the AUV will cover the area in parallel tracks in the same fashion as the grass is mowed down on a lawn using a lawn-mowing engine. A potential path planned using the *LPP* algorithm is shown with the blue colored trail in Figure 5.1.

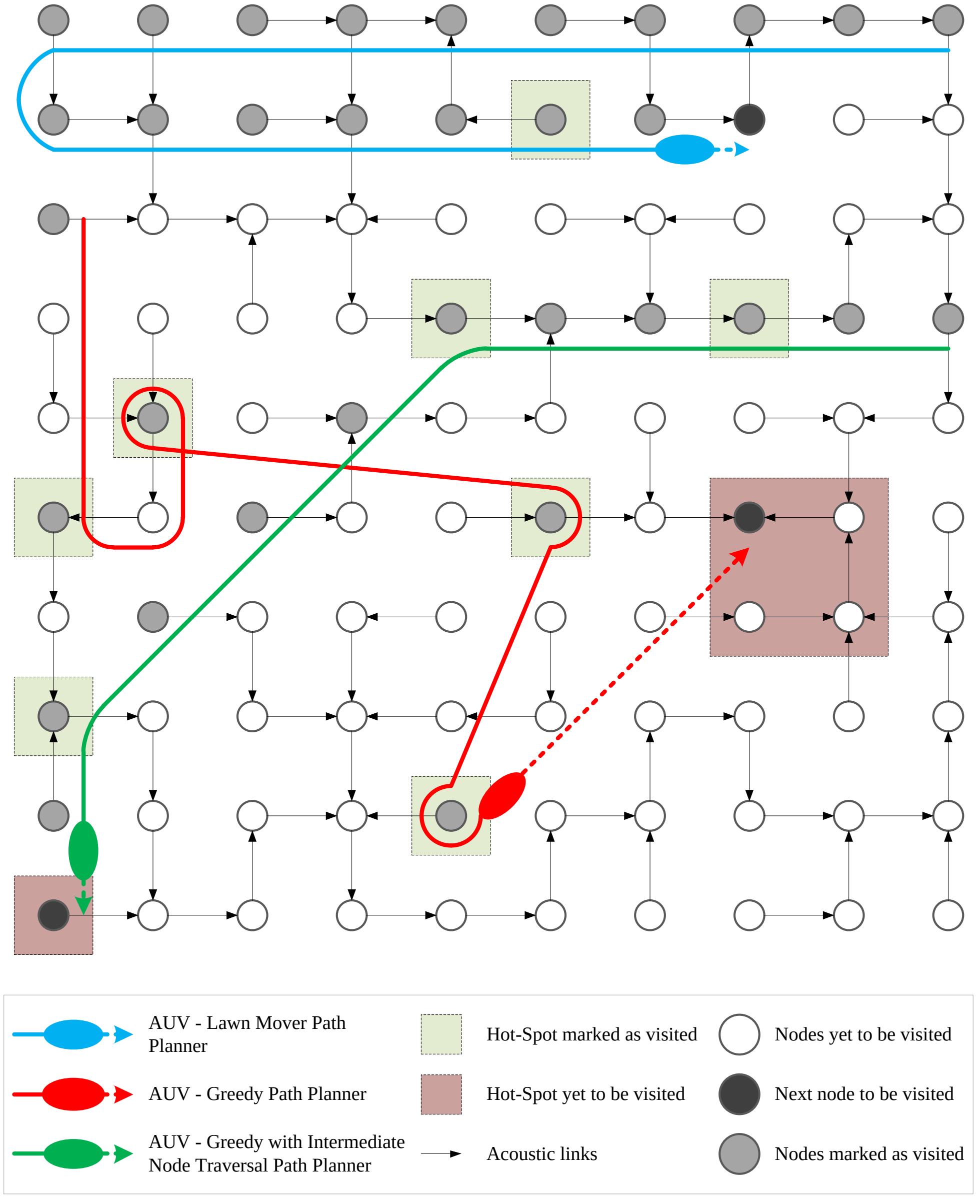

AUV - Lawn Mower Path Planner
AUV - Greedy Path Planner
AUV - Greedy with Intermediate Node Traversal Path Planner
Hot-Spot marked as visited
Hot-Spot yet to be visited
Acoustic links
Nodes yet to be visited
Next node to be visited
Nodes marked as visited

Figure 5.1: The UWSN mesh setting used for the simulation study. The same arrangement will be used for simulation and experimentation in the results section. The blue colored path is of the AUV that uses *LPP*, red of the AUV that uses *GPP* and green of the AUV that uses *GIPP*. The *GPP* visits one hot-spot after another. Notice how *LPP* stumbles upon a hot-spot, i.e., discovers it by chance. Also, observe how *GIPP* visits intermediate nodes in-between hot-spot visits.

Algorithm 1 Shortest Path Lawn-Mower Path Planner – *LPP*

```
procedure LPP(S)
    $S \leftarrow \{s_1, s_2, ..., s_n\}$                ▷ Set of sensor nodes
    $V \leftarrow \emptyset$                              ▷ Visitation sequence
    $P_D \leftarrow \{(East, West, South, North)\}$       ▷ Direction priority list
    $i \leftarrow s_1$                                    ▷ Tour starting node
    while $S \neq \emptyset$ do
        $N \leftarrow$ NEIGHBORHOOD$(i, S)$
        $j \leftarrow s_x$ from $N$ in the direction given by $P_D$
        TOURECULIDEAN$(i, j)$
        $V \leftarrow V + j$
        $S \leftarrow S - j$
        $i \leftarrow j$
    end while
    return $V$
end procedure
```

Please note that this path planner does not explicitly take VoI into account but it should give effective results as it essentially takes into consideration the minimization of time t, which is a parameter in determining VoI as suggested by Equations (3.14). Therefore, this algorithm implicitly improves VoI by taking care of overall visitation time.

Algorithm 1 highlights the steps involved in the Lawn-Mower path planner. The AUV maintains a history of nodes already visited and the ones not yet visited. Besides this, it also maintains a direction priority list. This list contains elements related to all the sensor nodes such that each element contains the information about which node among the neighbors of a sensor node needs to be visited next. For our implementation, we prioritize the selection of the next neighbor that is located on the East of the current node, then West, followed by South and finally North. The complexity of *LPP* is $\mathcal{O}(n)$ where n is the number of sensor nodes.

A more generic implementation of this algorithm, for any topology other than a mesh, would be to use an algorithm that solves the TSP. However, for a mesh, instead of using a computationally expensive TSP algorithm, one can just simply use the *LPP*.

5.2.2 Greedy Path Planner - GPP

GPP and its variants are reactive path planning algorithms. This path planner provides a visitation sequence of nodes in descending order of their VoI aggregate profiles, i.e., the AUV will visit the node with maximum VoI available first and then other nodes in decreasing order of VoI. The VoI is determined as the value available at the time instant when the AUV visits the sensor node. This is a greedy approach as the algorithm tries to maximize the VoI accumulated by determining which node has the highest amount to offer and then visit it. A potential path planned using the *GPP* algorithm is shown with the red colored trail in Figure 5.1.

VoI accumulation is dependent on the final time-stamps τ_f as suggested by Equation (3.14). The lesser the value of τ_f, the higher the VoI accumulated, i.e., the earlier the data is collected from a node, the more quickly the value decay is locked down, hence, resulting in greater value dividends. It would, therefore, make more sense to lock down VoI at nodes that have higher values to offer. Otherwise, more VoI would be potentially lost.

If there is a scenario where high-priority events are being reported because of a catastrophe, then it is imperative that it be reported at the earliest. The more the delay in reporting catastrophic data, the more losses would be incurred. Moreover, as the UWSN has been deployed to report of catastrophic events, therefore, it is necessary that the AUV visits the nodes reporting the catastrophe first.

Algorithm 2 details the steps for the greedy approach. The algorithm maintains a history of nodes already visited and the ones not yet visited. From the nodes not visited yet, it selects the ones that give it the maximum VoI at its time of arrival. Once it selects a node as its next destination, it takes a direct Euclidean route to it. The complexity of *GPP* is $\mathcal{O}(n^2)$ where n is the number of sensor nodes.

5.2.3 Greedy Path Planner with Intermediate-Node-Visitation - GIPP

It is a similar algorithm to *GPP* but with the added notion that while the AUV is on its way to the next node with the highest VoI offer, it can visit intermediate nodes that (by some definition e.g., distance) lie on the prescribed path. It is a reasonable idea to visit intermediate nodes (i.e. nodes that lie on the path to the destination node), because this may help in minimizing tour time and, therefore, can improve VoI accumulated.

Algorithm 3 lays out the steps for *GIPP*. It is similar to *GPP* except for the intermediate node visitation which is a slight detour. For the intermediate node visitation, the algorithm selects the next node to be visited from the neighbors of the current node in a manner that the distance between the neighbor node and the node with the maximum VoI (destination node) is minimum among all the neighboring nodes. This algorithm is inspired by concepts behind both the *GPP* and *LPP*, i.e., visit sequence is greedy but tour time is minimized by visiting nodes that lie across the path. However, when it comes to reporting catastrophes at the earliest, it is a tad slower than *GPP* because of the intermediate node visitations. A potential path planned using the *GIPP* algorithm is depicted by the green colored trail in Figure 5.1.

The complexity of *GIPP* is $\mathcal{O}(n^2 + n \times d \times m)$ where n is the number of sensor nodes, d is the count of nodes in the longest intermediate tour, and m is the maximum number of nodes in a neighborhood definition. Because $d < n$ and also $m < n$, therefore, it follows that the complexity of *GIPP* is $\mathcal{O}(n^3)$.

Algorithm 2 Greedy Path Planner – *GPP*

```
procedure GPP($S$, $s_o$)
    $S \leftarrow \{s_1, s_2, ..., s_n\}$                                  ▷ Set of sensor nodes
    $V \leftarrow \emptyset$                                               ▷ Visitation sequence
    $i \leftarrow s_o$                                                     ▷ Tour starting node
    while $S \neq \emptyset$ do
        $j \leftarrow$ GETNODETHATHASMAXVOI($S$)
        TOURECULIDEAN($i, j$)
        $V \leftarrow V + j$
        $S \leftarrow S - j$
    end while
    return $V$
end procedure
procedure GETNODETHATHASMAXVOI($S_r$)
    $\forall s_x \in S_r$ determine $\Upsilon_{s_x}$ using DETERMINENODEVOI($D_{s_x}$, $t$)
    $k \leftarrow s_x \in S_r$ such that $\Upsilon_{s_x}$ is max $\sum A e^{-B(t-\tau_o)}$
    return $k$
end procedure
procedure DETERMINENODEVOI($D, t$)
    $D \leftarrow \{d_1, d_2, ..., d_k\}$                                  ▷ Data reports at node
    $\Upsilon \leftarrow 0$
    $\tau_f \leftarrow t$                                                  ▷ AUV arrival time at node
    while $D \neq \emptyset$ do
        $j \leftarrow$ GETNEXTDATAREPORT($D$)
        $\alpha \leftarrow$ GETA$_x$($j$)
        $\beta \leftarrow$ GETB$_x$($j$)
        $\tau_o \leftarrow$ GET$\tau_{o_x}$($j$)
        $\Upsilon \mathrel{+}= \alpha e^{-\beta(\tau_f - \tau_o)}$
        $D \leftarrow D - j$
    end while
end procedure
```

Algorithm 3 Greedy Path Planner with Intermediate-Node-Visitation – *GIPP*

```
procedure GIPP($S$, $s_o$)
    $S \leftarrow \{s_1, s_2, ..., s_n\}$                          ▷ Set of sensor nodes
    $V \leftarrow \emptyset$                                        ▷ Visitation sequence
    $i \leftarrow s_o$                                              ▷ Tour starting node
    while $S \neq \emptyset$ do
        $j \leftarrow$ GetNodeThatHasMaxVoI($S$)
        $T \leftarrow \emptyset$
        $T \leftarrow$ TourIntermediate($i, j, S$)
        $V \leftarrow V + T$
        $S \leftarrow S - T$
        $i \leftarrow j$
    end while
    return $V$
end procedure
procedure TourIntermediate($source, destination, S_r$)
    $p \leftarrow source$
    $q \leftarrow destination$
    $T_I \leftarrow \emptyset$                                      ▷ Intermediate Visitation Sequence
    while $p \neq q$ do
        $N \leftarrow$ GetNeighborhood($p, S_r$)
        $i \leftarrow s_x \in N$ such that EculideanDistance($s_x, q$) is minimized
        TourEculidean($p, i$)
        $p \leftarrow i$
        $T_I \leftarrow p$
    end while
    return $T_I$
end procedure
```

Algorithm 4 Hybrid Path planner – *HPP*

```
procedure $HPP(S, s_o)$
    $S \leftarrow \{s_1, s_2, ..., s_n\}$                              ▷ Set of sensor nodes
    $V \leftarrow \emptyset$                                             ▷ Visitation sequence
    $S_{HP} \leftarrow \{s_m \mid s_m \in S \wedge s_m$ is high-priority$\}$      ▷ Set of high-priority sensor nodes
    $S_{LP} \leftarrow S - S_{HP}$                                       ▷ Set of low-priority sensor nodes
    $V \leftarrow GPP(S_{HP}, s_o)$
    $V \leftarrow V + LPP(S_{LP})$
    return $V$
end procedure
```

5.2.4 Hybrid Path Planner - HPP

The algorithms for path planners *HPP* and *HIPP* were devised after studying results from our simulation experiments. What we observed was that the *LPP* would perform the best, in terms of VoI accumulation, when there where no catastrophes, i.e., VoI profiles were similar across all the sensor nodes. On the other hand, *GPP* and *GIPP* performed better in the scenario where the sensor nodes were reporting catastrophes.

So the thought process behind this path planner is to use a greedy algorithm (*GPP* or GIPP) for collecting data from nodes reporting catastrophes, while, using TSP like algorithms (*LPP*) for collecting data from the rest of the nodes. This implies switching a greedy and a shortest path algorithm and so the name Hybrid Path Planner. *HPP* switches between *GPP* and *LPP*.

The algorithm for *HPP* is given as Algorithm 4. The complexity of this algorithm is $\mathcal{O}(h^2 + l)$ where h is the number of high-priority sensor nodes while l is the count of low-priority sensor nodes. $\mathcal{O}(h^2)$ is the complexity of *GPP* being used within *HPP* while $\mathcal{O}(l)$ is the complexity of *LPP*. As $h \leq n$ and $l \leq n$, therefore, the complexity of *HPP* is determined as $\mathcal{O}(n^2)$.

Algorithm 5 Hybrid Path Planner with Intermediate-Node-Visitation – *HIPP*

```
procedure HIPP($S$, $s_o$)
    $S \leftarrow \{s_1, s_2, ..., s_n\}$                ▷ Set of sensor nodes
    $V \leftarrow \emptyset$                             ▷ Visitation sequence
    $S_{HP} \leftarrow \{s_m \mid s_m \in S \wedge s_m$ is high-priority $\}$    ▷ Set of high-priority sensor nodes
    $i \leftarrow s_o$                                   ▷ Sensor node to start tour from
    while $S_{HP} \neq \emptyset$ do
        $j \leftarrow$ GETNODEMAXVOI($S_{HP}$)
        $T \leftarrow \emptyset$
        $T \leftarrow$ TOURINTERMEDIATE($i, j, S$)
        $V \leftarrow V + T$
        $S_{HP} \leftarrow S_{HP} - T$
        $i \leftarrow j$
    end while
    $S_{LP} \leftarrow S - V$                            ▷ Set low-priority sensor nodes
    $V \leftarrow$ V + $LPP(S_{LP})$
    return $V$
end procedure
```

5.2.5 Hybrid Path Planner with Intermediate-Node-Visitation - HIPP

HIPP has the same logic behind it as *HPP* but the only difference is that it uses *GIPP* instead of *GPP*. The algorithm first discovers the hot-spots, i.e., catastrophe reporting nodes and then it finds the intermediate nodes that can be visited. Afterwards, these nodes are visited using *GIPP* while the rest of the nodes are visited using *LPP*.

The algorithm for *HIPP* is given as Algorithm 5. The complexity of this algorithm is $\mathcal{O}(h^3 + l)$ where h is the number of high-priority sensor nodes while l is the count of low-priority sensor nodes. Similarly, as in the case of *HPP*, $h \leq n$ and $l \leq n$, therefore, the complexity of *HIPP* is determined as $\mathcal{O}(n^3)$.

5.2.6 *Random Path Planner - RPP*

For evaluation purposes, we also implemented a random path planner (*RPP*) where the AUV randomly chooses the next sensor node for data collection. The route towards the selected node employs Euclidean shortest path. This path planner can be thought of as a planner which does not take into account VoI or time while scheduling visits to nodes, i.e., AUV schedules its visitation activity irrespective of the critical nature of the VoI in the UWSN.

5.3 Performance Measures, Simulation Setup and Results

5.3.1 *Performance Measures and Experiments*

The basic measures of performance used are VoI accumulated ’Υ_{Acc}’ and VoI lost ’Υ_L’. To discuss what these measures are, we refer to Figure 3.4. $\Upsilon_{Acc}(t)$ is the decaying VoI profile in the system and represented in abstract terms by a straight line. It is a general statement on the depreciation of the valuation, whereas, in an actual situation the dynamics of this depreciation, i.e., the actual shape of the curve, will be governed by system variables and type of information recorded. This chart assumes that no measurements are recorded after $t = 0$, hence, the chart only shows a monotonic decay in the valuation after $t = 0$. τ_{start} and τ_{finish} are the start and end times for the complete AUV tour. In our case, a tour is a visitation sequence that is a permutation on the set of all sensor nodes, i.e., the tour is a simple path on a graph that has the sensor nodes as vertices. Υ_{Ava} is the VoI available in the system at the start of the tour while the Υ_{Acc} is the VoI accumulated by the AUV by the end of its tour. Υ_L is determined as

$$\Upsilon_L = \Upsilon_{Ava} - \Upsilon_{Acc}$$

The loss in Υ_{Acc} is a result of a combination of factors that can be attributed to physical system limitations such as the AUV speed, delay in starting time of the tour τ_{Start}, and inefficiencies resulting from the planned path. It is the path planning part of this problem that we explore in this chapter. To filter out effects of a delay in τ_{Start}, we use Υ_L as it is a measure of loss between the range $t = [\tau_{Start} : \tau_{Finish}]$. Any loss other than Υ_L is not a result of path planning inefficiencies. Υ_L gives a more clear picture in terms of comparative performance as compared to Υ_{Acc}.

Other than VoI based performance markers, we can also use time as a metric to determine certain aspects of performance. Time taken by an AUV for a complete traversal of the set of sensor nodes can be employed for this purpose. The earlier an AUV completes the tour of a UWSN, the more readily it is available for a new tour of this or another neighboring UWSN. Time can be useful in understanding energy E_{AUV} consumed by the AUV to complete its tour. Energy consumed is proportional to the distance it travels D_{tour} which is proportional to the time taken by the AUV to complete the tour T_{tour}. Shorter tours also result in less wear and tear of the AUVs.

$$\Delta\tau = \tau_{finish} - \tau_{start} = T_{tour} \propto D_{tour} \propto E_{AUV} \tag{5.1}$$

A measure of the efficiency Ω of a planner can be determined by combining Υ_L and T_{tour}. Ω is inversely proportional to both Υ_L and T_{tour}, therefore,

$$\Omega \propto \frac{1}{\Upsilon_L} \cdot \frac{1}{T_{tour}} \implies \Omega = \frac{p}{q\Upsilon_L \cdot rT_{tour}} = \frac{k}{\Upsilon_L \cdot T_{tour}} \tag{5.2}$$

where, $k = \frac{p}{q \cdot r}$ is a constant and is set to 1 in this experimental study. Other than the measures of Υ_{Acc}, Υ_L, T_{tour} & Ω, we also need to have some specific measures for 'response to emergency situations'. These would be VoI accumulated from first hot-spot $\upsilon_{HS_{Acc}}$, VoI lost from first hot-spot υ_{HS_L}, time taken to arrive at first hot-spot τ_{HS} and a measure of urgency Ψ. We define Ψ as the ratio of the score of the path planner schedule for visiting hot-spots S_I to the score of an expected perfect schedule to visit hot-spots S_P. Let there be n sensor nodes and m hot-spots. Then, the score S_I & S_P are determined as

$$S_I = \sum_{i=0}^{m-1} n - seq\#_{s_i} \, , S_P = \sum_{i=1}^{m-1} n - i \tag{5.3}$$

where $seq\#_{s_i}$ is the number at which a node is visited in the schedule of visitation given by a path planner. Ψ is, therefore, determined as

$$\Psi = \frac{S_I}{S_P} = \frac{\sum_{i=0}^{m-1} n - seq\#_{s_i}}{\sum_{i=1}^{m-1} n - i} \tag{5.4}$$

The intuition behind urgency score Ψ can be understood by the following example. Assume that there is a sensor network with 8 sensor nodes $\{s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8\}$ and 3 hot-spots $\{s_3, s_6, s_7\}$. Here $n = 8$ & $m = 3$. The hot-spots in sequence of their precedence are $[s_6, s_3, s_7]$, i.e., it is most urgent to visit s_6 on priority, then s_3 and then s_7. The perfect visitation sequence should be $[s_6, s_3, s_7, \ldots]$. Let there be a path-planner *PP* such that it gives us the following schedule of visitation $[s_1, s_3, s_2, s_6, s_5, s_7, s_8, s_4]$. In this visitation sequence s_6 has a $seq\#_{s_6} = 3$, s_3 has $seq\#_{s_3} = 1$ while s_7 has $seq\#_{s_7} = 5$. In a perfect visitation sequence s_6 should have $seq\#_{s_6} = 0$, s_3 should have $seq\#_{s_3} = 1$ while s_7 should have $seq\#_{s_7} = 2$. We can now calculate S_I and S_P as $S_I = (8-3) + (8-1) + (8-5) = 15$ and $S_P = (8-0) + (8-1) + (8-2) = 21$. The smaller a $seq\#_{s_i}$ is, the higher the difference $n - seq\#_{s_i}$ will be and hence the greater the score. The urgency score for *PP* is $\Psi = S_I / S_P = 15/21 = 0.714$.

Therefore, the complete list of measures that we use in this study for adjudication are Υ_{ACC}, Υ_L, T_{tour}, Ω, υ_{HS_L}, τ_{HS} & Ψ.

5.3.2 Simulation Setup

To investigate our various hypotheses regarding VoI based path planning we have used a simulation approach. We assume a scenario with two types of classes for the events; normal routine events while the other events require an emergency response. To monitor these events a UWSN has been deployed. The nodes in the UWSN collect multimedia information through cameras. Due to a limited finite capacity, data needs to be offloaded from these nodes by an AUV. The nodes communicate infotential data using the acoustic communication medium. A path planning agent, based on the infotential data received, schedules a visitation sequence of the nodes for the AUV. During its tour, the AUV off-loads multimedia data from the nodes using the optical communication medium. A node that has recorded an event which can be classified as an emergency is marked as a hot-spot.

The simulation has 100 nodes arranged in a 10×10 mesh/grid. The horizontal/vertical inter-node distance is 1000 m while the diagonal distance is 1414.2 m. The AUV traverses the UWSN at a constant speed of 10 m/s. We assume that before the AUV embarks on its tour, the UWSN has been recording data for 24 hours. Each node has video data reports of length 15 minutes each. Therefore, the reports have been recorded starting at intervals in the multiples of 15 minutes and with no recording overlap. The event coverage of the nodes is such that they have minimum or no-overlap, and hence, the recorded events are unique. Each node records data reports and classifies them either as a high-priority or a low-priority event. A $\{valuation, decay\}$ tuple, corresponding to significance and damage rate, is appended to each data report. If it is a routine low-priority event, then the $\{valuation, decay\}$ tuple $\{A_L, B_L\}$ are appended to the data report. Alternatively, if it is an emergency event, then the tuple $\{A_H, B_H\}$ are attached to it. The UWSN nodes communicate with the remote path-planning agent over the acoustic channel and transmit to it the $\{valuation, decay\}$ and time-stamp details of the data reports they have recorded. Based on these details, the path-planning agent determines the visitation sequence for the AUV.

5.3.3 Results

In this section, we attempt to make a comparative analysis of the path planning algorithms. We ascertain the performance of algorithms that take into consideration VoI or time or both, versus those algorithms, that do not. The regions where a high-priority event takes place are deemed as hot-spots. The effect of parameters such as valuation ratio A_H/A_L and number of hot-spots N_{HS} is assessed.

We make a performance comparison among six path planners; *RPP*, *LPP*, *GPP*, *HPP*, *GIPP*, and *HIPP*. At various points in this section, we normalize results. The normalization is with respect to *RPP* as it is our base-case. Except for in anomalous cases, *RPP* is the worst performing among all path planning algorithms. This is expected as it does not take into account time or information valuation for generating the node visitation sequence. Each reading in the results has been averaged over 100 different VoI profiles across the same UWSN map.

5.3.3.1 Valuation Ratio

We first investigate the valuation ratio. The valuation ratio is the ratio between the valuation of information of a high-priority event versus valuation of information of a low-priority event at $t = \tau_o$, i.e., the ratio A_H/A_L. In hindsight of the results for $N_{HS} = 0$, we found that if the VoI profile in a UWSN is similar or homogeneous, i.e. all the sensor nodes have almost a similar amount of VoI to offer, then the shortest path algorithm proved quite effective. This effectiveness is a result of distance minimization which results in minimization of time, and this time minimization helps to lock-in decaying VoI profiles at the earliest. However, so should be the case if $A_H \approx A_L$, because, even though there are hot-spots, yet the VoI profile of the system is as if $N_{HS} = 0$. In such a case, there would be no real advantage of using a greedy planner. Therefore, there should be a particular

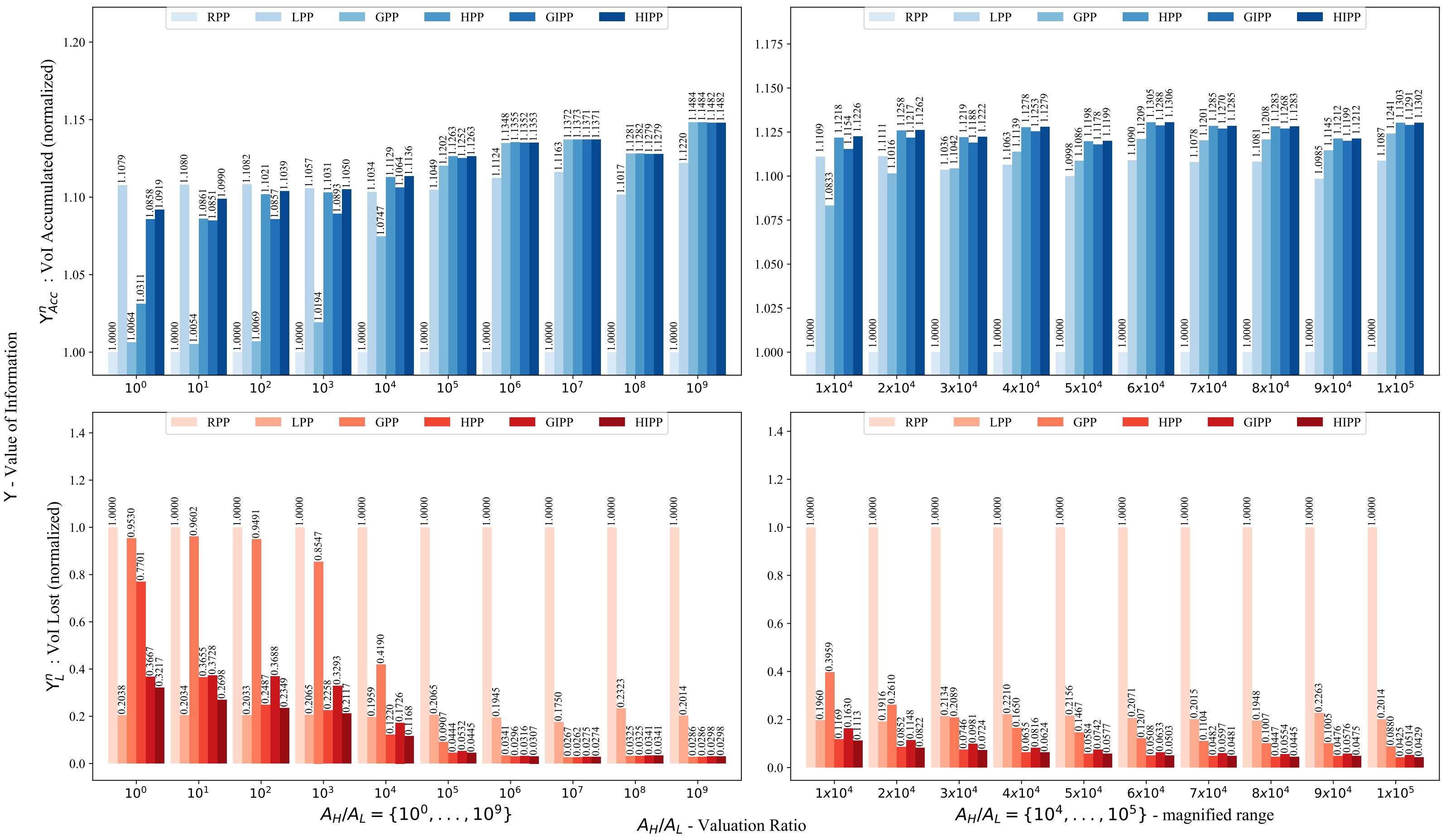


Figure 5.2: The top two graphs (colored in blue) are for Υ_{Acc} while the bottom two are for Υ_L while the graphs on the left are for the valuation ratio range $[10^0 : 10^9]$ while the graphs on the right are a magnification between $[10^4 : 10^5]$: (a) The top-left graph is for Υ^n_{Acc} which is charted in the range $[10^0 : 10^9]$; (b) The top-right graph is for Υ^n_{Acc} which is charted in the range $[10^4 : 10^5]$; (c) The bottom-left graph is for Υ^n_L which is charted in the range $[10^0 : 10^9]$; (d) The bottom-right graph is for Υ^n_L which is charted in the range $[10^4 : 10^5]$.

range after which we can observe a performance gain for the greedy planners as compared to the shortest path planners. In this experiment, we determine the effective performance range for the lawn-mower, greedy and hybrid algorithms. The results for Υ^n_{Acc} and Υ^n_L versus A_H/A_L are shown in Figure 5.2. The superscript n implies that the results have been normalized with respect to *RPP*.

$$\Upsilon^n_{Acc} = \frac{\Upsilon^{PP}_{Acc}}{\Upsilon^{RPP}_{Acc}}\ , \Upsilon^n_L = \frac{\Upsilon^{PP}_L}{\Upsilon^{RPP}_L} \tag{5.5}$$

In Figure 5.2, the left two graphs are for the range $A_H/A_L = [10^0{:}10^9]$. The right two graphs are a magnification of the results between the range $A_H/A_L = [10^4{:}10^5]$. Also, the top two graphs are for Υ^n_{Acc} (colored in tones of blue), while the bottom two graphs are for Υ^n_L (colored in tones of red). We can observe that *LPP* performs better early on but once the ratio A_H/A_L becomes considerably large, the greedy algorithms start giving better performance. It is somewhere between a valuation ratio of 10^3 and 10^4 units that *LPP* loses its top spot on performance. At 10^4 *GIPP*, *HPP*, and *HIPP* start performing better. While at 10^5 *GPP* also starts performing better than *LPP*. *LPP* clearly performs best up to 10^3, while afterward, the greedy or hybrid algorithms start performing better. If we magnify the range between $A_H/A_L = [10^4{:}10^5]$, we find that the switch in performance takes place at $A_H/A_L = 3 \times 10^4$.

It is based on these results that we use a setting of $A_H/A_L = 5 \times 10^4$ for the rest of this chapter. This is a reasonable number for practical situations as, for example, it may imply a \$1 versus \$50,000 valuation. However, it is worth to note that the hybrid or intermediate node visitation algorithms perform better as early as a fiscal value of \$10,000; as suggested by the performance improvement at $A_H/A_L = 10^4$.

5.3.3.2 *Justification of Heuristics*

The algorithms proposed in this chapter are based on different heuristics. In this section, we verify whether our intuitions behind those heuristics are valid or not. We use Υ^n_{ACC} (values normalized with respect to RPP) and study the performance of the path planners for the case where the number of hot-spots $N_{HS} = \{0, 1, 10\}$.

Our first hypothesis was that using VoI aware algorithms may help in accumulating a higher amount of VoI. From the results in Figure 5.3 we can see that all path planners have a value greater than 1.0, i.e., they are all better than *RPP*, and hence, suggesting that the hypothesis is correct.

Our second hypothesis was that greedy path planners amass a higher VoI when there are hot-spots. However, when there are no hot-spots, a time minimization planner like the Lawn-Mower accumulates a higher VoI. This hypothesis is validated through the results in Figure 5.3 (a). When $N_{HS} = 0$, *LPP* performs best, but in the case of hot-spots, the greedy approach accumulates more VoI.

Our third hypothesis was that inter-node traversal helps in minimizing time while still maintaining its VoI greedy character, thereby, improving the VoI accumulated. *GIPP*, for instance, visits hot-spots on priority but also visits other nodes that lie along the path. Figure 5.3 (b) confirms this intuition where *GIPP* accumulates more VoI than *GPP*.

Our fourth hypothesis was that hybrid path planners can provide the best of both worlds, i.e., when $N_{HS} = 0$, they behave like time minimization path planners and when there are hot-spots they use greedy techniques for scheduling purposes (until all hot-spots have been visited). To understand this, observe Figure 5.3 (c) & (d). When there are no hot-spots *HPP* performs as good as the *LPP* (because their algorithmic construction, in this case, is the same) and in case of hot-spots, they accumulate more VoI than their greedy counterparts, i.e., *GPP* and *GIPP*. Figure 5.3 (c) shows that the hybrid algorithm is better than both *LPP* and *GPP* while Figure 5.3 (d) shows that hybrid algorithms are better than their greedy counterparts in both cases, i.e., with or without intermediate node visitation.

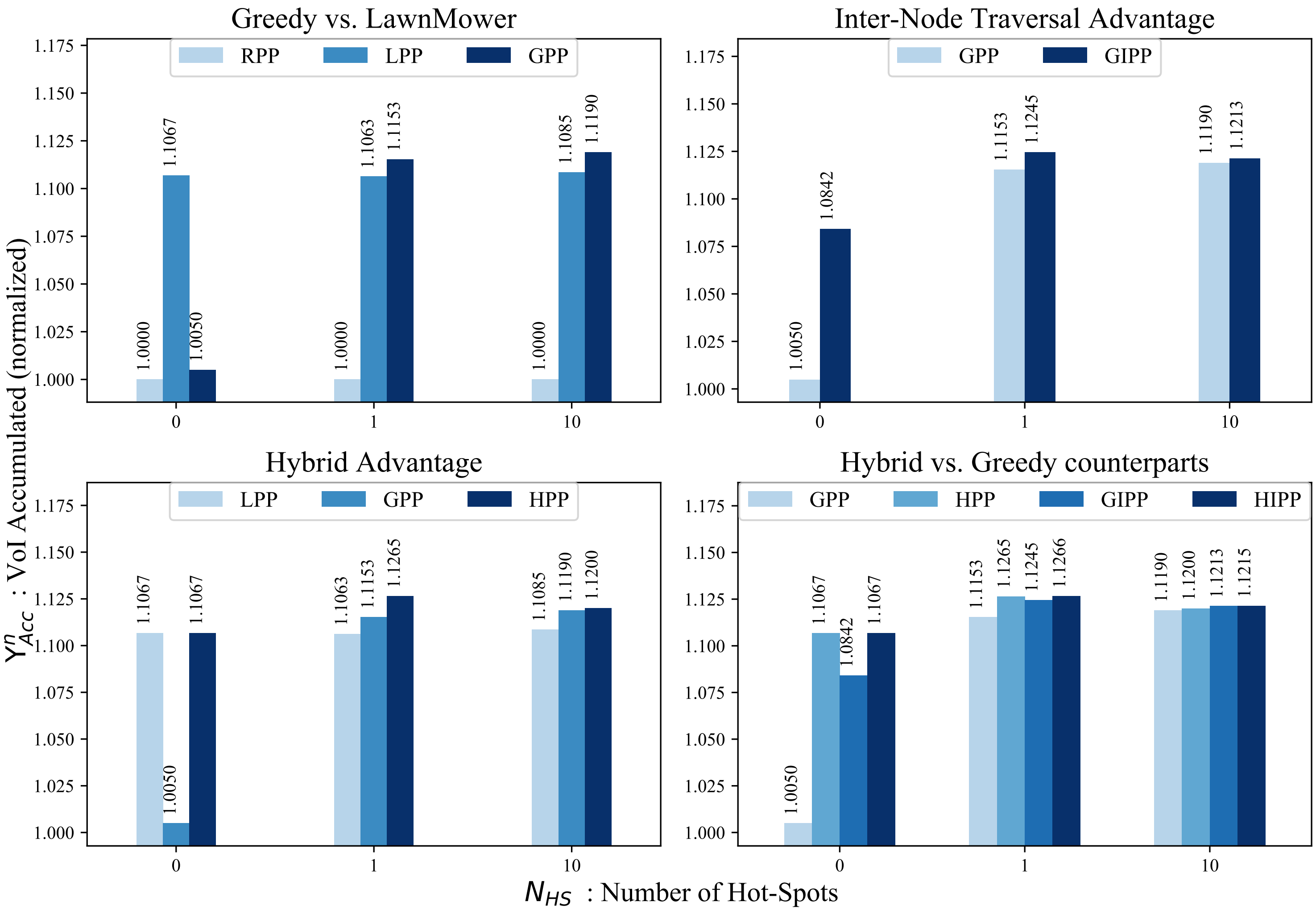


Figure 5.3: Bar graphs for justifying the use of various heuristics : (a) Greedy vs. Lawn-Mower; (b) Inter-Node Traversal Advantage; (c) Hybrid Advantage; (d) Hybrid vs. Greedy counterparts.

5.3.3.3 Comparative Analysis

In this section, we do a thorough comparative analysis to have a better understanding of the performance dynamics of the various path planners. The parameters we explore for this purpose are Υ_{ACC}, Υ_L, T_{tour} & Ω^n. We vary the number of hot-spots N_{HS} from 0 to 10 and see the effect on the path planners' performance.

For this section, we have used stacked bar-plots (Figures 5.4 & 5.5). The results have not been normalized to give a more clear picture of the actual values obtained from the simulation. Each reading for a {Planner, Hot-Spot(s)} tuple is averaged over 100 different VoI profiles and is represented by a block in the stacked bar-plot. The exact value of a block is given in a corresponding cell in the table below. The changing color gradient of the blocks matches to the index column N_{HS} in the table. The changing color tones of the table cells correspond to a change in intensity of values. The better the performance, the darker the color tone is in the table. Color tones for comparative performance can only be compared across a row, i.e., values can be only compared across a particular N_{HS} value.

VoI Accumulated and VoI Lost

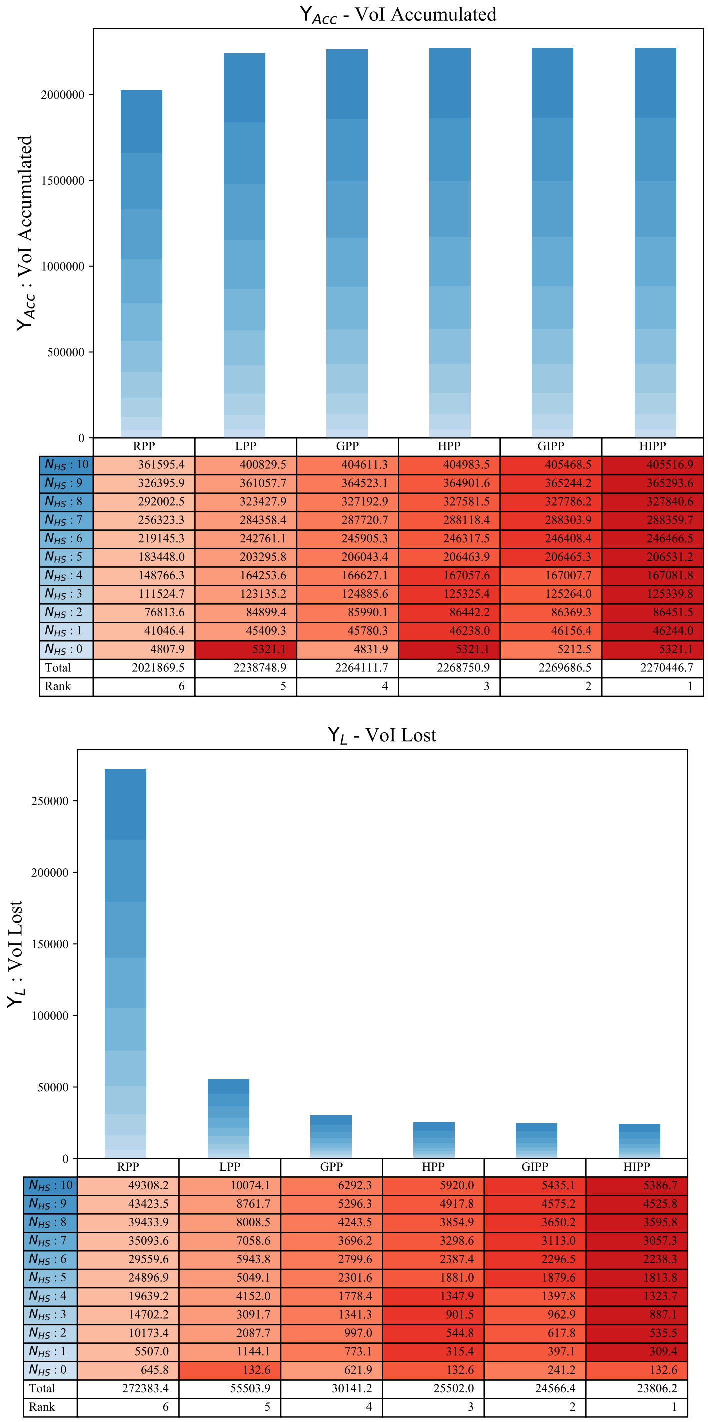
This discussion refers to graphs in Figure 5.4. The bar graph in Figure 5.4 (a) is for VoI accumulated Υ_{ACC} by various path planners, while Figure 5.4 (b) pertains to VoI lost Υ_L. They have the same conclusions in terms of performance but the performance results for Υ_L are more pronounced as compared to Υ_{ACC} and this is because, as stated earlier, they remove the bias due to loss before $t = \tau_{Start}$. *LPP* performs best when there are no hot-spots. In this case, the performance of *HPP* and *HIPP* is same as *LPP*. With $N_{HS} > 0$ the greedy and hybrid approaches start to perform better. The hybrid algorithms always perform better than their greedy counterparts. Also, *GIPP* always performs better than *GPP*. An interesting thing to note is that initially *HPP* has better performance than *GIPP* but after $N_{HS} = 4$ the *GIPP* path planner starts outperforming HPP; which is due to the fact that the AUV using *GIPP* might start encountering lesser valued hot-spots more frequently while on its way towards a higher valued hot-spot; a benefit in hindsight of using intermediate node visitation. The cumulative performance rank for $N_{HS} = [0:10]$ is : $HIPP > GIPP > HPP > GPP > LPP > RPP$.

Tour Time

The best performing algorithm in terms of time is shortest path algorithm *LPP* and this can be seen from the results in Figure 5.5 (a). *HIPP* and *HPP* come in close in terms of minimizing time, but as the number of hot-spots increase, their performance gap to *LPP* also widens. *GIPP* performs better than *GPP* because of using the intermediate node visitation heuristic. *GPP* is as worse as *RPP* in terms of time, which is understandable as the VoI profile is distributed randomly across the UWSN map. It is important to note that when $N_{HS} = 0$ then $LPP = HPP = HIPP$. The performance rank precedence across $N_{HS} = [0:10]$ is : $LPP > HIPP > HPP > GIPP > GPP \approx RPP$.

Efficiency

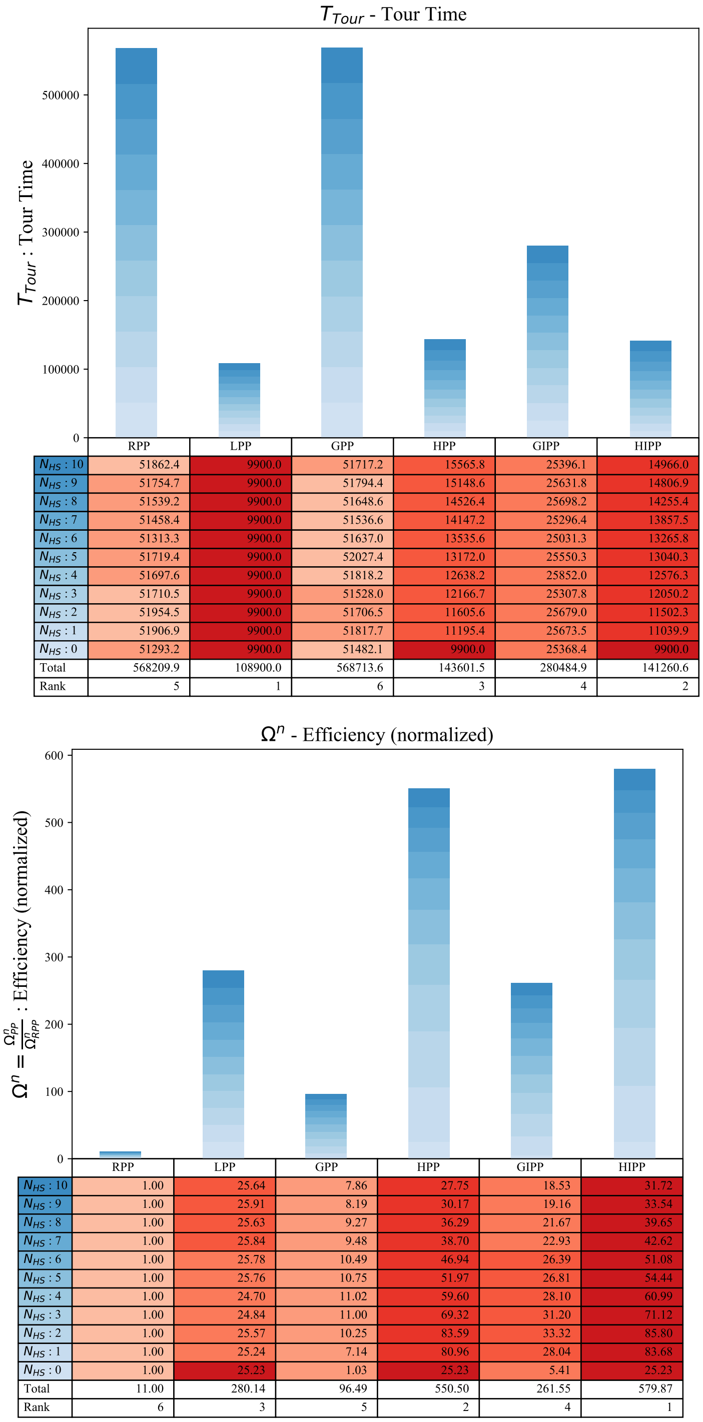
This is a useful metric as it takes into account both the VoI lost and tour time. The results are shown in Figure 5.5 (b). The hybrid algorithms turn out to be the best in terms of efficiency. The conclusion is not a surprise as they combine the best of both the shortest path and greedy algorithms. The cumulative efficiency rank is: $HIPP > HPP > LPP > GPP > GIPP > RPP$.



	RPP	LPP	GPP	HPP	GIPP	HIPP
N_{HS} : 10	361595.4	400829.5	404611.3	404983.5	405468.5	405516.9
N_{HS} : 9	326395.9	361057.7	364523.1	364901.6	365244.2	365293.6
N_{HS} : 8	292002.5	323427.9	327192.9	327581.5	327786.2	327840.6
N_{HS} : 7	256323.3	284358.4	287720.7	288118.4	288303.9	288359.7
N_{HS} : 6	219145.3	242761.1	245905.3	246317.5	246408.4	246466.5
N_{HS} : 5	183448.0	203295.8	206043.4	206463.9	206465.3	206531.2
N_{HS} : 4	148766.3	164253.6	166627.1	167057.6	167007.7	167081.8
N_{HS} : 3	111524.7	123135.2	124885.6	125325.4	125264.0	125339.8
N_{HS} : 2	76813.6	84899.4	85990.1	86442.2	86369.3	86451.5
N_{HS} : 1	41046.4	45409.3	45780.3	46238.0	46156.4	46244.0
N_{HS} : 0	4807.9	5321.1	4831.9	5321.1	5212.5	5321.1
Total	2021869.5	2238748.9	2264111.7	2268750.9	2269686.5	2270446.7
Rank	6	5	4	3	2	1

	RPP	LPP	GPP	HPP	GIPP	HIPP
N_{HS} : 10	49308.2	10074.1	6292.3	5920.0	5435.1	5386.7
N_{HS} : 9	43423.5	8761.7	5296.3	4917.8	4575.2	4525.8
N_{HS} : 8	39433.9	8008.5	4243.5	3854.9	3650.2	3595.8
N_{HS} : 7	35093.6	7058.6	3696.2	3298.6	3113.0	3057.3
N_{HS} : 6	29559.6	5943.8	2799.6	2387.4	2296.5	2238.3
N_{HS} : 5	24896.9	5049.1	2301.6	1881.0	1879.6	1813.8
N_{HS} : 4	19639.2	4152.0	1778.4	1347.9	1397.8	1323.7
N_{HS} : 3	14702.2	3091.7	1341.3	901.5	962.9	887.1
N_{HS} : 2	10173.4	2087.7	997.0	544.8	617.8	535.5
N_{HS} : 1	5507.0	1144.1	773.1	315.4	397.1	309.4
N_{HS} : 0	645.8	132.6	621.9	132.6	241.2	132.6
Total	272383.4	55503.9	30141.2	25502.0	24566.4	23806.2
Rank	6	5	4	3	2	1

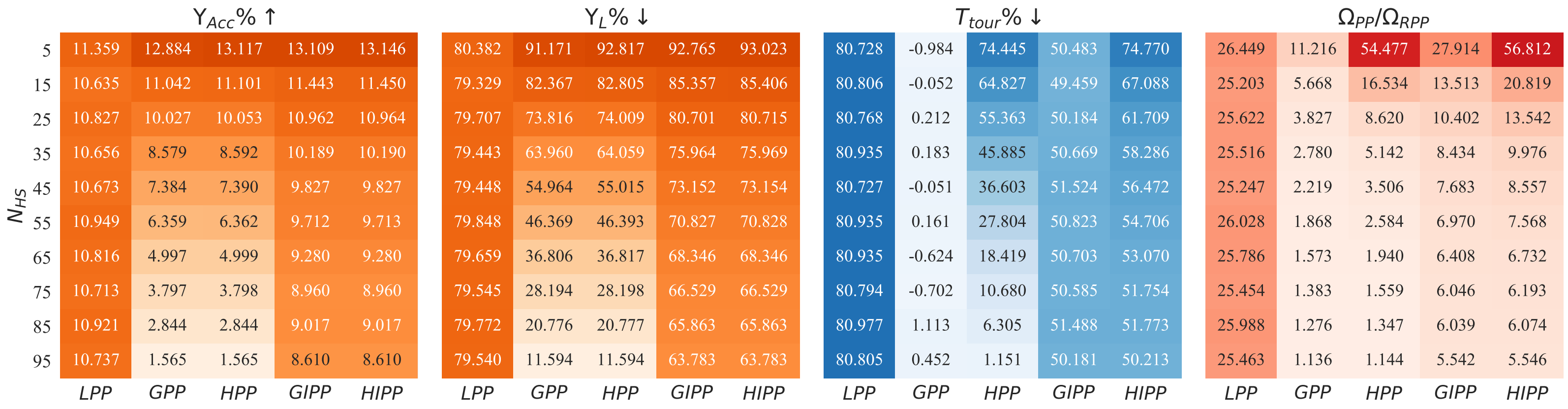
Figure 5.4: Comparative analysis for Υ_{ACC} & Υ_L : (a) The top stacked bar graph is for VoI accumulated Υ_{ACC}; (b) The bottom stacked bar graph is for VoI lost Υ_L.



	RPP	LPP	GPP	HPP	GIPP	HIPP
N_{HS} : 10	51862.4	9900.0	51717.2	15565.8	25396.1	14966.0
N_{HS} : 9	51754.7	9900.0	51794.4	15148.6	25631.8	14806.9
N_{HS} : 8	51539.2	9900.0	51648.6	14526.4	25698.2	14255.4
N_{HS} : 7	51458.4	9900.0	51536.6	14147.2	25296.4	13857.5
N_{HS} : 6	51313.3	9900.0	51637.0	13535.6	25031.3	13265.8
N_{HS} : 5	51719.4	9900.0	52027.4	13172.0	25550.3	13040.3
N_{HS} : 4	51697.6	9900.0	51818.2	12638.2	25852.0	12576.3
N_{HS} : 3	51710.5	9900.0	51528.0	12166.7	25307.8	12050.2
N_{HS} : 2	51954.5	9900.0	51706.5	11605.6	25679.0	11502.3
N_{HS} : 1	51906.9	9900.0	51817.7	11195.4	25673.5	11039.9
N_{HS} : 0	51293.2	9900.0	51482.1	9900.0	25368.4	9900.0
Total	568209.9	108900.0	568713.6	143601.5	280484.9	141260.6
Rank	5	1	6	3	4	2

	RPP	LPP	GPP	HPP	GIPP	HIPP
N_{HS} : 10	1.00	25.64	7.86	27.75	18.53	31.72
N_{HS} : 9	1.00	25.91	8.19	30.17	19.16	33.54
N_{HS} : 8	1.00	25.63	9.27	36.29	21.67	39.65
N_{HS} : 7	1.00	25.84	9.48	38.70	22.93	42.62
N_{HS} : 6	1.00	25.78	10.49	46.94	26.39	51.08
N_{HS} : 5	1.00	25.76	10.75	51.97	26.81	54.44
N_{HS} : 4	1.00	24.70	11.02	59.60	28.10	60.99
N_{HS} : 3	1.00	24.84	11.00	69.32	31.20	71.12
N_{HS} : 2	1.00	25.57	10.25	83.59	33.32	85.80
N_{HS} : 1	1.00	25.24	7.14	80.96	28.04	83.68
N_{HS} : 0	1.00	25.23	1.03	25.23	5.41	25.23
Total	11.00	280.14	96.49	550.50	261.55	579.87
Rank	6	3	5	2	4	1

Figure 5.5: Comparative analysis for T_{tour} & Ω : (a) The top stacked bar graph is for time to complete the tour T_{tour}; (b) The bottom stacked bar graph is for the efficiency measure Ω.



$\Upsilon_{Acc}\%\uparrow$

N_{HS}	LPP	GPP	HPP	GIPP	HIPP
5	11.359	12.884	13.117	13.109	13.146
15	10.635	11.042	11.101	11.443	11.450
25	10.827	10.027	10.053	10.962	10.964
35	10.656	8.579	8.592	10.189	10.190
45	10.673	7.384	7.390	9.827	9.827
55	10.949	6.359	6.362	9.712	9.713
65	10.816	4.997	4.999	9.280	9.280
75	10.713	3.797	3.798	8.960	8.960
85	10.921	2.844	2.844	9.017	9.017
95	10.737	1.565	1.565	8.610	8.610

$\Upsilon_L\%\downarrow$

N_{HS}	LPP	GPP	HPP	GIPP	HIPP
5	80.382	91.171	92.817	92.765	93.023
15	79.329	82.367	82.805	85.357	85.406
25	79.707	73.816	74.009	80.701	80.715
35	79.443	63.960	64.059	75.964	75.969
45	79.448	54.964	55.015	73.152	73.154
55	79.848	46.369	46.393	70.827	70.828
65	79.659	36.806	36.817	68.346	68.346
75	79.545	28.194	28.198	66.529	66.529
85	79.772	20.776	20.777	65.863	65.863
95	79.540	11.594	11.594	63.783	63.783

$T_{tour}\%\downarrow$

N_{HS}	LPP	GPP	HPP	GIPP	HIPP
5	80.728	-0.984	74.445	50.483	74.770
15	80.806	-0.052	64.827	49.459	67.088
25	80.768	0.212	55.363	50.184	61.709
35	80.935	0.183	45.885	50.669	58.286
45	80.727	-0.051	36.603	51.524	56.472
55	80.935	0.161	27.804	50.823	54.706
65	80.935	-0.624	18.419	50.703	53.070
75	80.794	-0.702	10.680	50.585	51.754
85	80.977	1.113	6.305	51.488	51.773
95	80.805	0.452	1.151	50.181	50.213

Ω_{PP}/Ω_{RPP}

N_{HS}	LPP	GPP	HPP	GIPP	HIPP
5	26.449	11.216	54.477	27.914	56.812
15	25.203	5.668	16.534	13.513	20.819
25	25.622	3.827	8.620	10.402	13.542
35	25.516	2.780	5.142	8.434	9.976
45	25.247	2.219	3.506	7.683	8.557
55	26.028	1.868	2.584	6.970	7.568
65	25.786	1.573	1.940	6.408	6.732
75	25.454	1.383	1.559	6.046	6.193
85	25.988	1.276	1.347	6.039	6.074
95	25.463	1.136	1.144	5.542	5.546

Figure 5.6: Scaling of response w.r.t. number of hot-spots N_{HS} : (a) Percentage improvement in VoI accumulated Υ_{ACC}; (b) Percentage reduction in VoI lost Υ_L; (c) Percentage reduction in tour time T_{Tour}; (d) Normalized measure of efficiency Ω.

5.3.3.4 Scalability with an increase in Hot-Spots

In the context of this discussion, a hot-spot is an anomalous event of probably catastrophic proportions and needs to be taken care of as quickly as possible. Such an anomalous event should be a rare occurrence and it would be a bit unlikely to find more than a few at the same time. It can also be argued that the whole system is compromised if such a situation arises. Our scalability study is based on this hypothetical assumption that there might be a drastic increase in the number of hot-spots. Our goal is to see how the path planners behave when we vary the number of hot-spots (N_{HS}) across the range of nodes.

The results are shown as heat-maps in Figure 5.6. Darker color tones imply better comparative performance in the heat-maps. We start from 5 hot-spots and ramp up to 95 hot-spots. We record percentage improvement in VoI accumulated ($\Upsilon_{ACC}\% \uparrow$), percentage reduction in VoI lost ($\Upsilon_L\% \downarrow$), percentage reduction in tour time ($T_{Tour}\% \downarrow$) and a normalized measure of efficiency ($\frac{\Omega_{PP}}{\Omega_{RPP}}$). The percentages are calculated with respect to the *RPP* path planner performance:

$$\Upsilon_{ACC}\% \uparrow = \frac{\Upsilon_{ACC}}{\Upsilon_{ACC}^{RPP}} \times 100 \tag{5.6}$$

$$\Upsilon_L\% \downarrow = \frac{\Upsilon_L}{\Upsilon_L^{RPP}} \times 100 \tag{5.7}$$

$$T_{Tour}\% \downarrow = \frac{T_{Tour}}{T_{Tour}^{RPP}} \times 100 \tag{5.8}$$

$$\frac{\Omega_{PP}}{\Omega_{RPP}} = \frac{\Upsilon_L^{RPP} \times T_{Tour}^{RPP}}{\Upsilon_L \times T_{Tour}} \tag{5.9}$$

For Υ_{ACC} and Υ_L, as the number of hot-spots increase, the performance of *HPP* and *LPP* decreases drastically. *GIPP* and *HIPP* also experience a degradation in performance but the change is not that drastic. The performance of *LPP* remains the same at average. The reason for this is that an increase in hot-spots results in a map that is increasingly homogeneous in terms of VoI. When there are no hot-spots, there is a homogeneity in terms of VoI in the sensor map. In such a case, as discussed earlier, *LPP* performs the best. With one and up till a few hot-spots, greedy path-planners perform better. However, at a certain tipping point in terms of N_{HS}, the performance falls below than that of *LPP*. With a large N_{HS}, the sensor map is similar to the one with no hot-spots. This explains the better performance of *LPP* in such a scenario.

In terms of T_{tour}, there is no effect in the performance of *LPP*, *GPP* or *GIPP* with an increase in N_{HS}. However, the performance of *HPP* deteriorates to the level of *GPP* eventually and *HIPP* degrades to *GIPP*. The reason lies in the algorithmic construction of the hybrid path planners. They are designed in such a way that they switch to *LPP* once all the hot-spots have been visited. The time performance of hybrid algorithms is good because they incorporate *LPP*. Therefore, if there is a large N_{HS}, there will be a lesser involvement of *LPP* in the path-planning process and this leads to an increase in the T_{tour} for hybrid path-planners.

The measure of efficiency, normalized to *RPP*, has a very high value for hybrid algorithms at 5 hot-spots. However, onward 15 hot-spots, we see a very sharp decline in this performance measure. *LPP* is not affected in this regard. The reason is that Ω depends upon Υ_L and T_{tour} and *LPP* is not affected by an increase in N_{HS}.

5.3.3.5 Response to Emergency

The basic reason to employ VoI was to have a mechanism to distinguish between higher and lower priority situations in an organic fashion, thus enabling a more appropriate response to the situation. The greedy and hybrid algorithms were designed to address high-priority situations such as emergencies. Here, we look at some measures that shed light on how the various path planners perform under emergency. Again, all results are described with respect to *RPP* in terms of percentage improvement in performance. The results are shown in Figure 5.7. We vary N_{HS} from 1 to 10

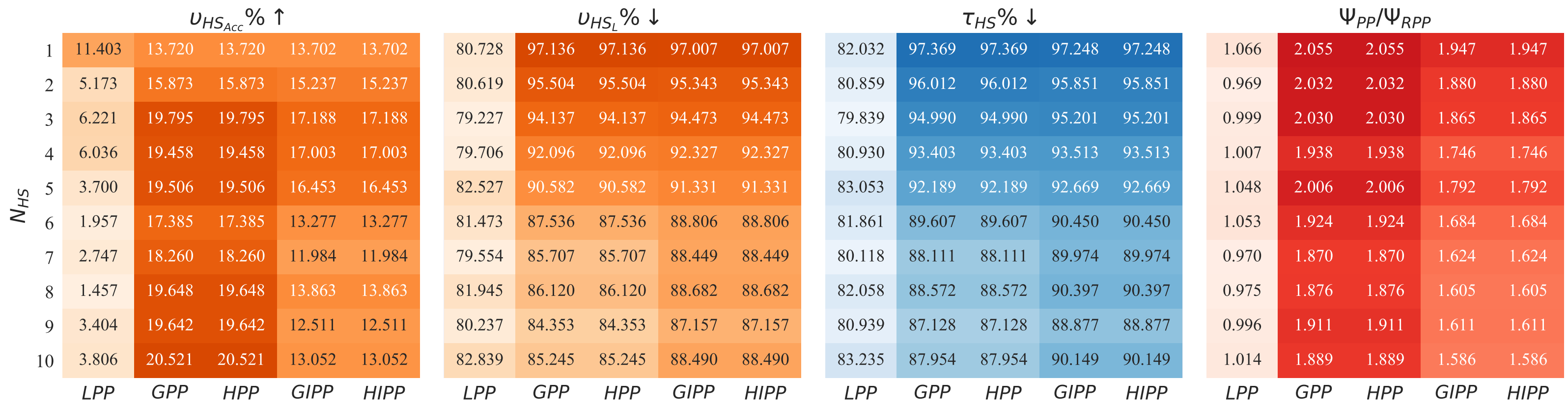


Figure 5.7: Emergency Response w.r.t. number of hot-spots N_{HS} : (a) Percentage improvement in VoI accumulated from first hot-spot $\upsilon_{HS_{ACC}}$; (b) Percentage reduction in VoI lost from first hot-spot υ_{HS_L}; (c) Percentage reduction in time to reach first hot-spot τ_{HS}; (d) Normalized urgency score Ψ.

hot-spots. We record percentage improvement in VoI accumulated from first hot-spot ($\upsilon_{HS_{ACC}}\% \uparrow$), percentage reduction in VoI lost at first hot-spot ($\upsilon_{HS_L}\% \downarrow$), percentage reduction in tour time to first hot-spot ($\tau_{HS}\% \downarrow$) and the normalized urgency score ($\frac{\Psi_{PP}}{\Psi_{RPP}}$). The percentages are calculated with respect to the *RPP* path planner performance:

$$\upsilon_{HS_{ACC}}\% \uparrow = \frac{\upsilon_{HS_{ACC}}}{\upsilon^{RPP}_{HS_{ACC}}} \times 100 \tag{5.10}$$

$$\upsilon_{HS_L}\% \downarrow = \frac{\upsilon_{HS_L}}{\upsilon^{RPP}_{HS_L}} \times 100 \tag{5.11}$$

$$\tau_{HS}\% \downarrow = \frac{\tau_{HS}}{\tau^{RPP}_{HS}} \times 100 \tag{5.12}$$

$$\frac{\Psi_{PP}}{\Psi_{RPP}} = \frac{S_I}{S_P} \times \frac{S_P^{RPP}}{S_I^{RPP}} \tag{5.13}$$

GPP and *HPP* are best when it comes to accumulating VoI from the first hot-spot. *GIPP* and *HIPP* follow closely in terms of performance but the gap widens with increasing N_{HS}. The reason, as discussed earlier, is that the intermediate path planners start to hit lesser valued hot-spots on their way to the highest valued hot-spot. As an intermediate lesser valued hot-spot maybe encountered first by the AUV, therefore, $\upsilon_{HS_{ACC}}$ performance should decrease.

In terms of υ_{HS_L}, *GPP* and *HPP* are better for N_{HS} = 1 or 2. For $N_{HS} \geq 3$, *GIPP* and *HIPP* start performing better, i.e., they avoid a higher loss in terms of VoI from the first hot-spot encountered. The reason is the same as stated above, i.e., they encounter other hot-spots on the tour while traveling towards the highest-priority one. Because, hot-spots are encountered earlier, therefore, the respective VoI loss υ_{HS_L} at that node should be lower. This phenomenon of encountering hot-spots earlier than planned can be verified from the time to arrive at the first hot-spot τ_{HS} results.

These results for τ_{HS} follow exactly υ_{HS_L} in character. Again, *GPP* and *HPP* perform better for N_{HS} = 1 or 2, while for $N_{HS} \geq 3$, *GIPP* and *HIPP* are better. This thus corroborates the speculation that hot-spots are being encountered earlier by path planners that are based on the intermediate node visitation strategy.

The urgency measure Ψ sheds light on response to an emergency. This is because it generates a score based on the sequence of visitation to the hot-spots. It encodes, that how much priority was maintained while visiting the nodes. *GPP* and *HPP* have the highest urgency score. This is because they visit the nodes in the exact order of descending priority as dictated by VoI advertised. The intermediate node visitation algorithms, *GIPP* and *HIPP*, come in second. This is expected as intermediate nodes are being attended to en-route to the highest-priority node. The performance gap widens between algorithms with or without intermediate node visitation with an increasing N_{HS}. The performance of *LPP* is as worse as *RPP* throughout the N_{HS} range. This is inferred from normalized result value of $\Psi_{LPP}/\Psi_{RPP} \approx 1$.

This shows that *GPP* and *HPP* are best for addressing emergencies as they directly go to the highest priority node first. Close in second are *GIPP* and *HIPP*. They lose out marginally because of visiting intermediate nodes. *LPP* has no capacity for dealing with emergencies.

5.4 Remarks

In this chapter, we have used VoI in the form of infotentials for solving data off-loading precedence issues in UWSNs. We have used a VoI model for UWSNs and developed various path planning algorithms based on heuristics. We have also proposed measures and metrics to evaluate system performance in such a scenario. A relationship has already been identified between the quality of information, QoI, and value of information, VoI, in [5, 7]. Based on this relationship and the extensive experiments that we have performed, it is reasonable to conclude that employing VoI for path planning algorithms improves the quality of information gained from a UWSN.

The path planner performance depends on the context of the situation. If the VoI profile in the system is homogeneous, i.e., there are no hot-spots or the valuation ratio A_H/A_L is small, then the shortest path algorithm like *LPP* should be used. *LPP* is also fuel optimal. In the case of hot-spots, given the valuation ratio A_H/A_L is considerably large, greedy algorithms perform better. Hybrid algorithms offer the best strategy by combining greedy and shortest path algorithms.

Intermediate node visitation improves VoI by saving tour time and, therefore, *GIPP* and *HIPP* perform better than their *GPP* and *HPP* counterparts respectively. However, if an emergency is classified as serious such that its priority should not be marginalized, then *GPP* or *HPP* should be used as they directly visit the node of concern.

CHAPTER 6: DETERMINING RESURFACING SCHEDULES

6.1 VoI Maximization and Path Planning Problem

In this chapter, we develop path planning algorithms to address the following VoI accumulation and maximization definitions, as given in Equation 3.34 & 3.35

$$\Upsilon_{Acc} = \sum_{h=1}^{g}\sum_{j=1}^{n}\sum_{k=1}^{d} A_{jk}e^{-B_{jk}(\tau_{f_h}-\tau_{o_{jk}})}.v_j^{visit}.l_{hj}^{visit}$$

$$\Upsilon_{Maximize}^{Resurface} \rightarrow \max \sum_{h=1}^{g}\sum_{j=1}^{n}\sum_{k=1}^{d} A_{jk}e^{-B_{jk}(\tau_{f_h}-\tau_{o_{jk}})}.v_j^{visit}.l_{hj}^{visit}$$

The path planning problem statement we solve in this chapter is described by Definition 3.43

$$\left[\Upsilon_{Acc}^{Alg^{PP}} \leftarrow P_{S+R}\right] \leftarrow Alg^{PP}\left[S,R,D,\Upsilon_{Acc}(t)\right]$$

where,

$\Upsilon_{Acc}^{Alg^{PP}}$ is the VoI accumulated from the sensor nodes S by employing the traversal sequence P_S,

P_{S+R} is the node visitation sequence intertwined with the resurfacing locations in Alg^{PP},

Alg^{PP} is a path planning algorithm that generates path P_S or $P_{(S,A)}$ such that $\Upsilon_{Acc}^{Alg^{PP}}$ is accumulated,

S is the set of all sensor nodes,

R is the set of all resurfacing locations,

D is the set of all data reports,

$\Upsilon_{Acc}(t)$ is the function total VoI accumulated.

6.1.1 Scheduling AUV Resurfacing

In this chapter, we consider the end-processing agent, i.e. the sink node, to be above the sea surface. Therefore, the AUV has to resurface to transmit data to a base station which serves as a part of the end-processing agent system. Three of the many possible tours (paths) involving resurfacing schedules are shown in Figure 6.1. The instant at which the AUV resurfaces and transmits data to the base station serve as our final time stamp in determining the VoI gained. This resurfacing affects the VoI gathered. A balance is required in terms of the number of times an AUV resurfaces because resurfacing at each node visit or resurfacing after visiting all nodes may not be the most optimal option. We present two contrasting scenarios to draw an intuitive inference. We work with two sensor nodes to illustrate the concept in a similar setting as described in Figure 6.1. Both sensor nodes have solitary information segments that were recorded at the same time and have the same VoI profile e^{-Bt}. The AUV can take two routes, the visitation sequences of which are:

- $P_1 : s_1 \rightarrow r_1 \rightarrow s_2 \rightarrow r_2$
- $P_2 : s_1 \rightarrow s_2 \rightarrow r_2$

where s_1 and s_2 are sensor nodes while r_1 and r_2 are the resurfacing points above them.

Consider the case when the sensor nodes are very near to the sea surface such that the distance of the sensor nodes to the sea surface (S_D) is almost zero and the inter-node distance (S_I) is much greater than this i.e. $S_D \cong 0$ and $S_I \gg S_D$. Let us calculate the VoI accumulated by the two paths

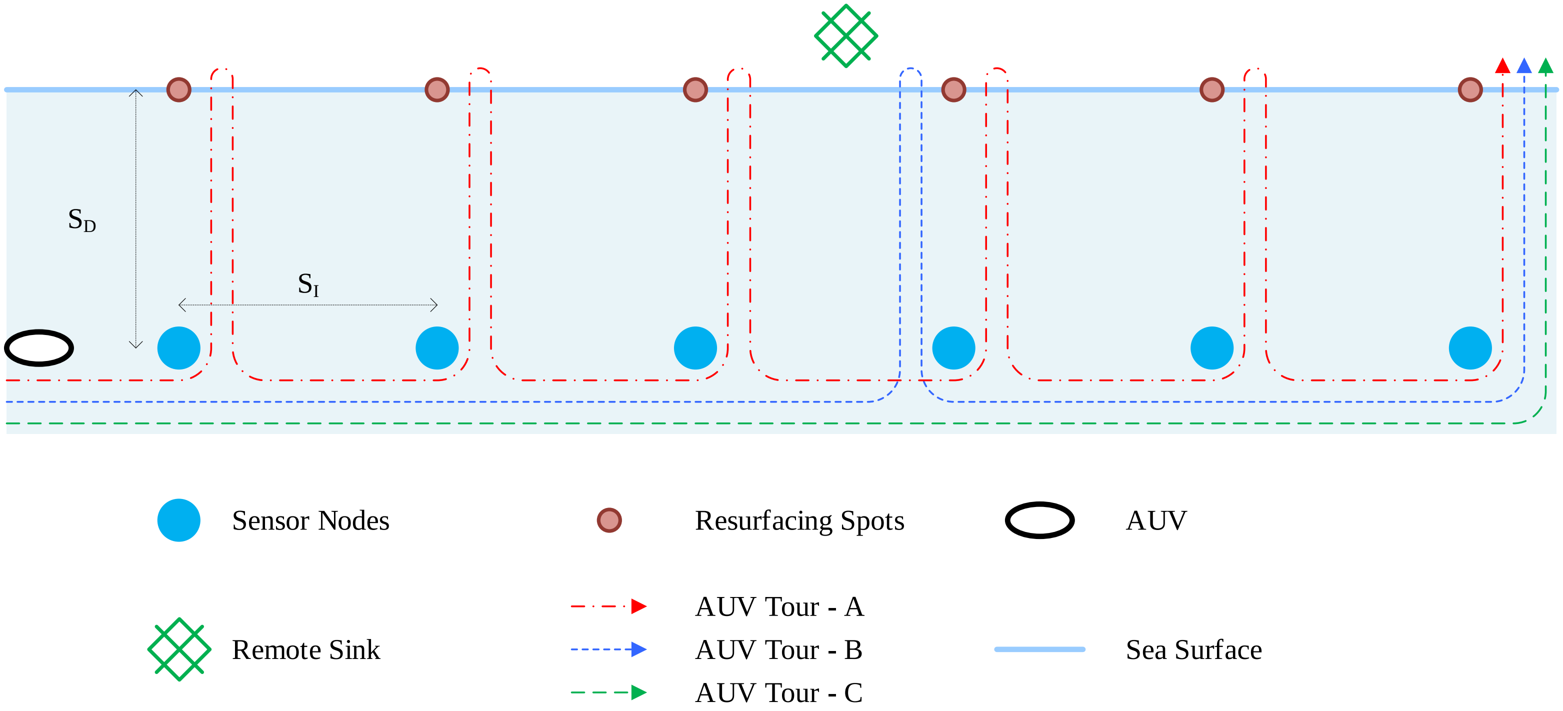


Figure 6.1: Side view of the underwater sensor network showing inter-node distance S_I and node depth from sea surface S_D.

assuming $S_I \neq 0$ and $S_D = 0$:

$$\begin{aligned}
\Upsilon^{P_1} &= e^{-Bt_1} + e^{-Bt_2} \\
&= e^{-B(t_{S_I}+t_{S_D})} + e^{-B(t_1+t_{S_D}+t_{S_I}+t_{S_D})} \\
&= e^{-Bt_{S_I}} + e^{-B(t_1+t_{S_I})} \\
&= e^{-Bt_{S_I}} + e^{-2Bt_{S_I}} \\
\Upsilon^{P_2} &= e^{-Bt_2} + e^{-Bt_2} \\
&= 2e^{-B(t_{S_I}+t_{S_I}+t_{S_D})} \\
&= 2e^{-2Bt_{S_I}}
\end{aligned}$$

Clearly $\Upsilon^{P_1} > \Upsilon^{P_2}$ which implies that it is better to resurface and transmit data at each node in the given distance (S_I, S_D) settings.

In contrast, consider the case when the sensor nodes are deep in the sea such that inter-node distance is much smaller than the distance to the surface of the sea i.e. $S_D \gg S_I$. Let us calculate the VoI accumulated by the two paths assuming $S_D \neq 0$ and $S_I = 0$:

$$\begin{aligned}
\Upsilon^{P_1} &= e^{-Bt_1} + e^{-Bt_2} \\
&= e^{-B(t_{S_I}+t_{S_D})} + e^{-B(t_1+t_{S_D}+t_{S_I}+t_{S_D})} \\
&= e^{-Bt_{S_D}} + e^{-B(t_1+2t_{S_D})} \\
&= e^{-Bt_{S_D}} + e^{-3Bt_{S_D}} \\
\Upsilon^{P_2} &= e^{-Bt_2} + e^{-Bt_2} \\
&= 2e^{-B(t_{S_I}+t_{S_I}+t_{S_D})} \\
&= 2e^{-Bt_{S_D}}
\end{aligned}$$

Clearly, $\Upsilon^{P_2} > \Upsilon^{P_1}$ which implies that it is better to resurface and transmit data after both nodes have been visited.

From this we can infer that there should be a ratio of S_D to S_I at which $\Upsilon^{P_2} = \Upsilon^{P_1}$ and an increase or decrease in this ratio will lead to one path performing better than the other in terms of accumulating VoI.

6.2 Genetic Algorithms for Resurfacing Schedules

We propose two alternative genetic algorithms, namely, G_{PR} and G_{Opt}, the former reaching quick convergence while latter offers a more optimal solution.

G_{PR} is modeled with an intuitive heuristic H_{PR} - a periodic resurfacing template. The algorithm has a low computational (running) cost as compared to G_{Opt} as it explores only a subset of the range of all the possible solutions. Due to the reduced search space, the algorithm might not yield the most optimal solution.

G_{Opt} has higher time complexity as it searches in the full domain of the possible solution set. Hence, it leads to a more optimal solution in comparison to G_{PR}. To improve the convergence time of G_{Opt}, and hence, the algorithmic runtime, we provide the algorithm with good seeds (based on H_{PR}) as a part of the initial generation of the population.

6.2.1 H_{PR} - Periodic Resurfacing Heuristic

From the derivations in Section 6.1.1, we infer that the choice of intermediate resurfacing points may improve or degrade the accumulated VoI. Moreover, shifting a resurfacing point in the schedule changes the potential VoI accumulated up to that point. VoI functions are time-dependent and a reconfiguration in resurfacing points affects the distance traveled by AUV, thereby affecting the final time stamp of a batch of information chunks.

Let S be a UWSN with n sensor nodes as defined in Chapter 3. Let the AUV resurface after every p sensor nodes, implying, that after visiting p nodes the AUV will resurface and transmit the batch of the information to the base station. The AUV will then visit the next p nodes before resurfacing to transmit and will keep on doing this until all n nodes have been visited. Period P can take on values

$$P = \{1, 2, \dots, p, \dots, n\}$$

As the number of the periods is n, a basic linear search based on H_{PR} will be of the order of $O(n) * O(V)$, where $O(V)$ is the complexity of the VoI evaluation procedure. In contrast, a basic linear search to find the most optimum schedule (maybe periodic or not) will have a complexity of the order of $O(2^n) * O(V)$, hence, advocating our use of H_{PR} for reducing complexity, albeit sacrificing optimality.

The number of resurfacing iterations for an AUV would be

$$\alpha = \lceil n/p \rceil$$

One anomaly to this periodic visitation is that in the last iteration, the AUV might not be able to visit p nodes as n may not be exactly divisible by p. Besides this anomaly, the rest of the schedule will have a periodic resurfacing pattern. In this last iteration, the number of nodes the AUV will visit will be

$$n - \lfloor n/p \rfloor \times p$$

6.2.2 Genetic Algorithms

Both of the proposed genetic algorithms, G_{PR} & G_{Opt}, use the same fitness function for evaluating the chromosomes. The fitness function is based on Equation 3.34,

$$F_C = \Upsilon_{Acc} = \sum_{h=1}^{g} \sum_{j=1}^{n} \sum_{k=1}^{d} A_{jk} e^{-B_{jk}(\tau_{f_h} - \tau_{o_{jk}})} . v_j^{visit} . l_{hj}^{visit} \tag{6.1}$$

6.2.2.1 G_{PR}:A for Optimal Periodic Resurfacing Schedule

This GA is based on the heuristic H_{PR} to find the optimal period for a resurfacing schedule that maximizes the VoI accumulated. The optimal period is an integer that varies between 0 and n (number of sensor nodes), therefore, the chromosome is simply a binary string where each gene can take on a binary value (0 or 1). G_{PR} employs the *uniform crossover operator* and *tournament selection* for evolving the population. It also uses *elitism* to retain the best solution after each generation during evolution.

6.2.2.2 G_{Opt}:A for Optimal Resurfacing Schedule

The chromosome for this optimal resurfacing schedule is a strand of genes where each gene represents a unique resurfacing location. The number of genes in each chromosome is equal to the number of resurfacing locations. Each gene can take on two values encoded to represent whether the corresponding resurfacing location should be visited or not. G_{Opt} construction is similar to G_{PR}. The crossover operator is *uniform* and the selection methodology is *tournament selection*. *Elitism* is employed to retain the best combinatorial solution while evolving through the various population generations.

Table 6.1: Simulation Parameters

Parameter	**Values**	
UWSN Deployment Parameters		
Deployment area	10 x 10 km^2	
Node deployment	Uniform Grid	
Inter-node distance - S_I	1 km	
Network deployment depth - S_D	Ratio * S_I	
Ratio - R_{DI} - S_D/S_I	0 - 1000	
Number of sensor nodes	100	
Transmission range	120 - 140 m	
Sensing range	70m	
Mobile sink speed	2 m/s	
Experimental Parameters		
Genetic algorithms	G_{PR}, G_{Opt}, R_{End}, R_{All}, R_{Rand}	
Runs per experiment	50	
Genetic Algorithm Parameters		
Genetic algorithm	G_{PR}	G_{Opt}
Generations (iterations)	20	100
Population size	25	50
Selection mechanism	*Tournament Selection*	
Tournament size	5	5
Elitism	*Yes*	
Crossover operator	*Uniform Crossover*	
Crossover rate	0.5	0.5
Mutation rate	0.1	0.15

The initial population is supplemented with good seeds, i.e. chromosomes with high fitness score that will yield good solutions. These seeds are obtained from the top best solutions generated by G_{PR}. This small variation could lead to a fast convergence time towards the optimal solution.

6.3 Simulation Setup & Results

The simulation parameters are given in Table 6.1. It is a 100 node UWSN deployed in a uniform grid over a 10 km x 10 km area. An AUV moving with a speed of 2 m/s is used to collect the data.

The AUV can offload data from sensor nodes using 10 Mbps optical links.

6.3.1 Studying the effect of Deployment Depth on AUV Resurfacing

This study is to validate inferences made in Sec 6.1.1 by shedding light on the relationship between the number of times an AUV resurfaces and the ratio R_{DI}.

$$R_{DI} = S_D / S_I$$

where S_D is the network deployment depth and S_I is the inter-node distance. The tour size for these experiments is 25 sensor nodes. The results in Figure 6.2 are averaged over 50 simulation runs per experiment.

Section 6.1.1 hypothesizes that if nodes are nearer to the water surface then more frequent resurfacing is required as compared to when the nodes are deeper in the sea. The results in Figure 6.2 can validate this argument. As R_{DI} increases, i.e. the nodes are placed deeper into the sea, the number of times the AUV resurfaces reduces. Note that there is a range of R_{DI} for which there is a significant change in the number of times an AUV resurfaces. Below that range, the AUV almost always resurfaces after every single node visit and above that range, the AUV rarely resurfaces before the end of the tour. In this experiment (at least) up to $R_{DI} = 0.5$ the AUV resurfaces 25 times in its 25-node tour and after $R_{DI} = 25$ it starts to taper to 1 (a single resurfacing event at the end of the tour). This implies that the scheduling algorithms for our setting of VoI functions (setting parameters A_{jk} and B_{jk}) are only effective within this range. Outside this range a deterministic approach such as R_{Every} and R_{End} would suffice.

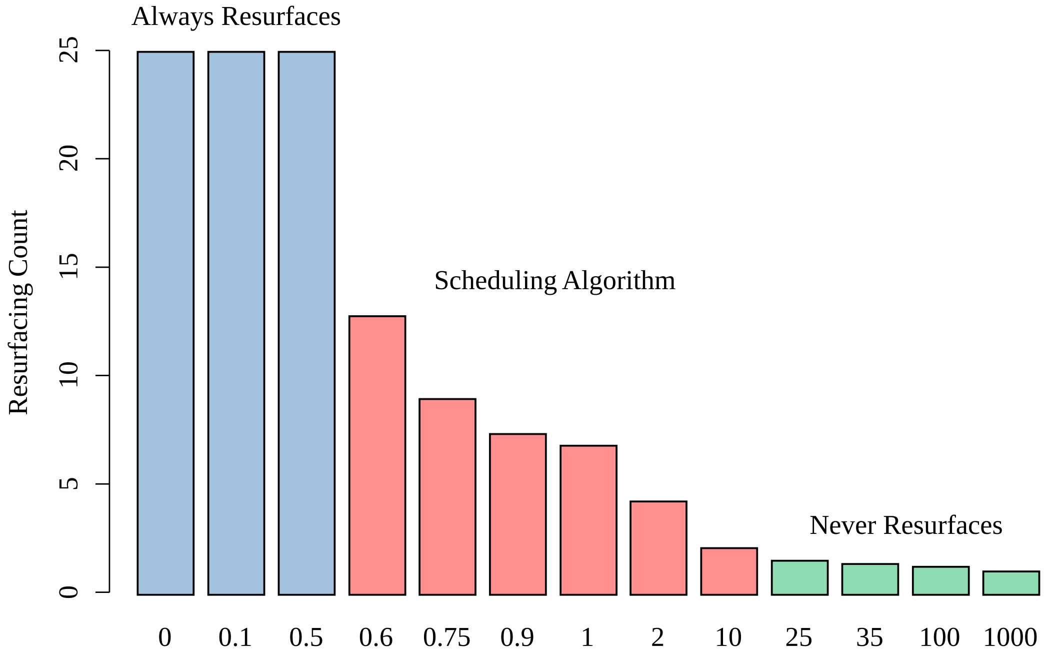


Figure 6.2: The average number of times the AUV resurfaces as a function of the ratio $R_{DI} = S_D/S_I$. The results are averaged over 50 simulation runs.

6.3.2 Performance Analysis of G_{PR} & G_{Opt} Heuristics

We use R_{Every}, R_{End} & R_{Rand} as scheduling procedures that serve as a baseline for comparison with the GA schedulers G_{PR} & G_{Opt}. These schedulers are described below:

- R_{Every} - AUV resurfaces after every node visit.
- R_{End} - AUV resurfaces at the end of the tour i.e. after visiting all of the sensor nodes in the tour.
- R_{Rand} - AUV randomly chooses the number of times it resurfaces during a tour and also after which node visit should it resurface.

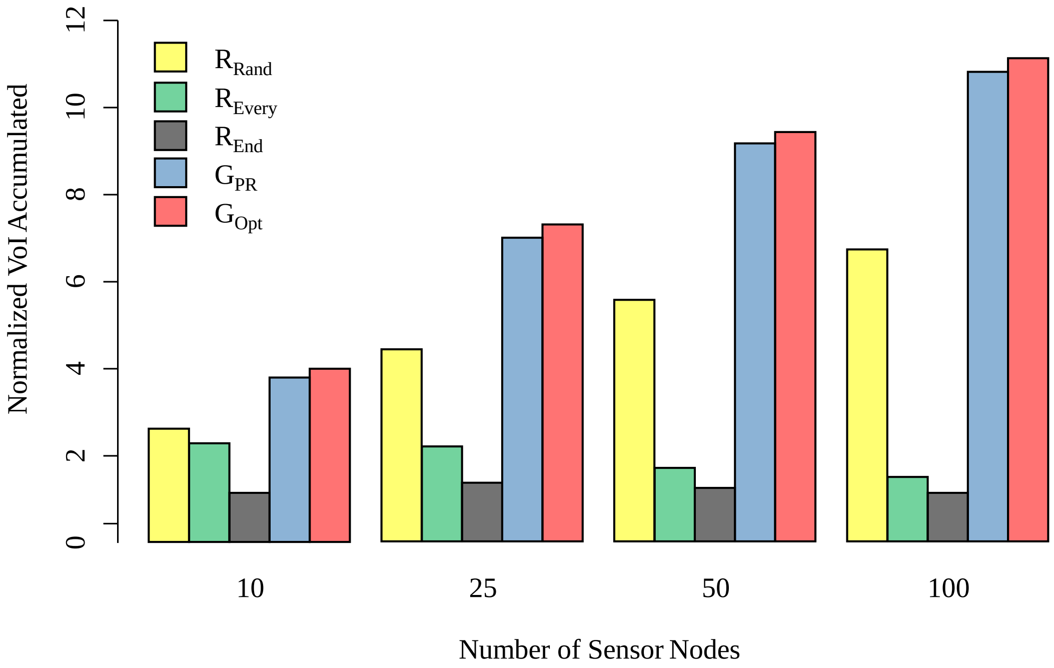


Figure 6.3: The VoI accumulated (Υ_S) by different schedulers as a function of the number of sensor nodes. The results are averaged over 50 simulation runs and have been normalized w.r.t. R_{End}.

The performance of a schedule is determined by the amount of VoI it accumulates, i.e. Υ_S. The results are shown in Figure 6.3. We use tour lengths of 10, 25, 50 and 100 sensor nodes for our experiments. In light of the results in Section 6.3.1, the ratio of S_D to S_I is set 1.0 for this experiment. Υ_S is an absolute measure and does not have units in our definition. Any amount of difference is a good result as it implies that more information has been gained in time for actuation purposes. The amount of difference can be magnified or diminished by controlling A_{jk} & B_{jk} settings but the results still signify the same information content. We have normalized all results w.r.t. R_{End} for interpretation purposes. All results are averaged over 50 simulation runs for each experiment.

From the results in Figure 6.3, we can infer that G_{PR} & G_{Opt} perform better than the baseline schedulers. The effectiveness of Heuristic H_{PR} is validated by the better results of G_{PR} over the baseline schedulers. Moreover, G_{Opt} generates more optimal schedules than G_{PR}, albeit its higher running cost. Choosing between G_{PR} & G_{Opt} is run time versus VoI optimization trade-off.

6.4 Remarks

In this chapter, we have developed a path planning approach for the case when an AUV needs to resurface to transmit collected data to a remote sink node. We propose a two-tiered approach to the path planning problem. As the first tier, we discover a path that optimizes VoI using algorithms in Chapter 5. This is an implicit step in the course of this chapter as it has already been discussed in Chapter 5. The second tier is to find the optimal resurfacing schedule for the AUV. So in this chapter, we address the problem that given a path P_S, what is the path with the resurfacing schedule P_{S+R} that can maximize the VoI collected.

We discover that the ratio R_{DI} has an important role to play in the resurfacing schedule. There are two extreme cases: if R_{DI} is too large then it makes sense to resurface only at the end of the tour; otherwise, if it is too small then the AUV should resurface after every node visit. From the analysis, we discover a range for R_{DI} in which it is reasonable to use an algorithm for determining a resurfacing schedule.

We develop two algorithms for this purpose. G_{Opt} is a combinatorial-optimization genetic algorithm for optimizing the resurfacing schedule in terms of VoI. G_{PR} uses a periodic resurfacing heuristic H_{PR} to find the VoI optimized route. While G_{Opt} is better at accumulating VoI, G_{PR} gives quite a comparable performance in this regard. The advantage of G_{PR} is its simplicity which requires far less computation time than G_{Opt}.

CHAPTER 7: MULTIPLE AUV PATH PLANNING

7.1 VoI Maximization and Path Planning Problem

In this chapter, we develop path planning algorithms to address the following VoI accumulation and maximization definitions, as given in Equation 3.26 & 3.27

$$\Upsilon_{Acc} = \sum_{i=1}^{a}\sum_{j=1}^{n}\sum_{k=1}^{d} A_{jk} e^{-B_{jk}(\tau_{f_{ij}} - \tau_{o_{jk}})} . v_{ij}^{visit}$$

$$\Upsilon_{Maximize}^{Multiple} \rightarrow \max \sum_{i=1}^{a}\sum_{j=1}^{n}\sum_{k=1}^{d} A_{jk} e^{-B_{jk}(\tau_{f_{ij}} - \tau_{o_{jk}})} . v_{ij}^{visit}$$

The path planning problem we solve in this chapter is described by Definition 3.42

$$\left[\Upsilon_{Acc}^{Alg^{PP}} \leftarrow P_{(S,A)}\right] \leftarrow Alg^{PP}\left[A, S, D, \Upsilon_{Acc}(t)\right]$$

where,

$\Upsilon_{Acc}^{Alg^{PP}}$ is the VoI accumulated from the sensor nodes S by employing the traversal sequence P_S,

$P_{(S,A)}$ is the set of all node visitation sequences for various AUVs determined by Alg^{PP},

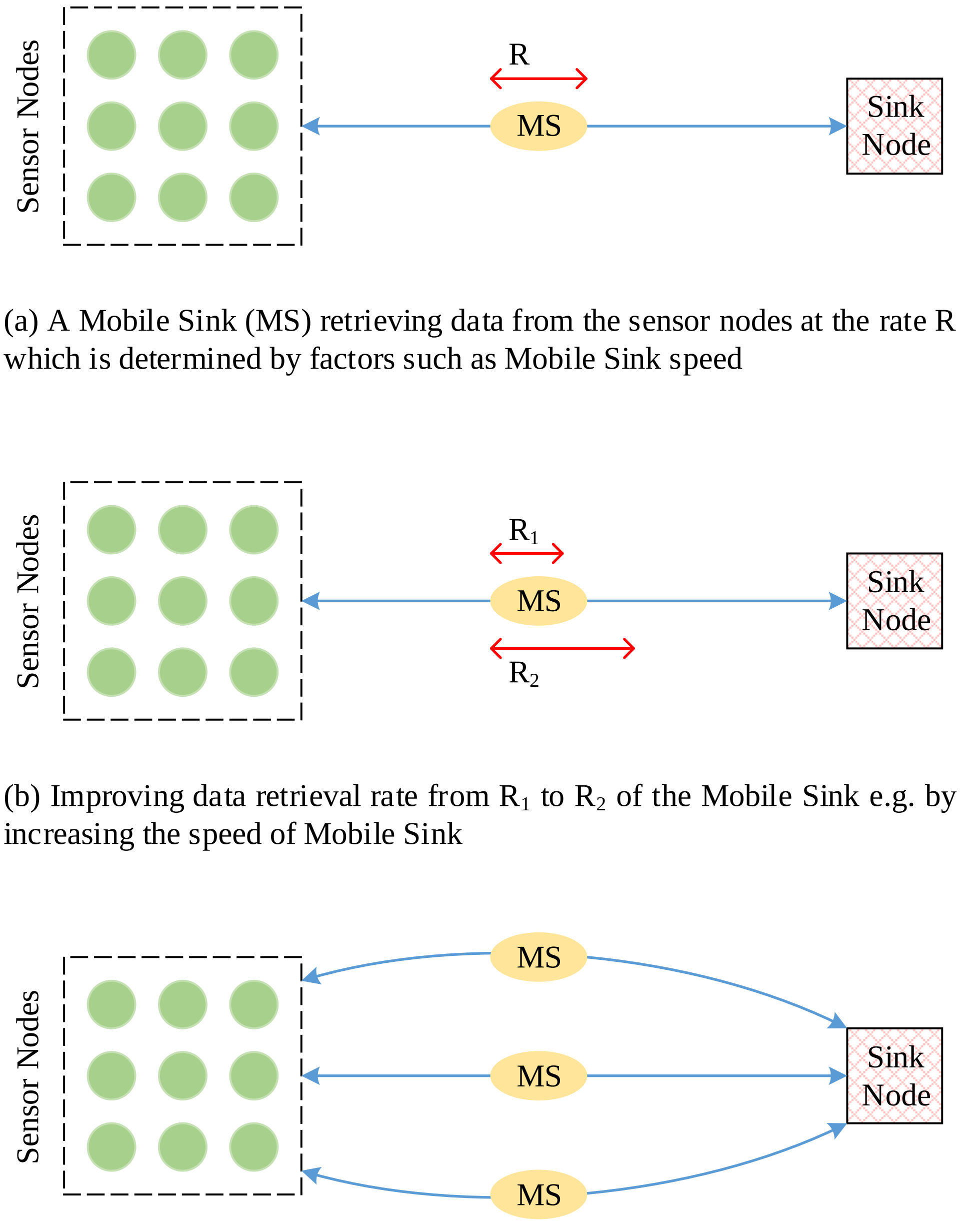
Alg^{PP} is a path planning algorithm that generates path P_S or $P_{(S,A)}$ such that $\Upsilon_{Acc}^{Alg^{PP}}$ is accumulated,

A is the set of all AUVs,

S is the set of all sensor nodes,

D is the set of all data reports,

$\Upsilon_{Acc}(t)$ is the function total VoI accumulated.



(a) A Mobile Sink (MS) retrieving data from the sensor nodes at the rate R which is determined by factors such as Mobile Sink speed

(b) Improving data retrieval rate from R_1 to R_2 of the Mobile Sink e.g. by increasing the speed of Mobile Sink

(c) Improving data retrieval rate by using multiple Mobile Sinks i.e. dividing data retrieval workload to reduce overall data retrieval time

Figure 7.1: Physical strategies to improve VoI accumulated by improving speed of a mobile sink (MS) or by using multiple mobile sinks.

7.2 Multiple AUVs - A Physical Strategy to Improve VoI

This chapter addresses the question of how to develop path planning algorithms for multiple AUVs. In this dissertation, for a single AUV, the planning path problem is to find a node visitation sequence for data collection such that it maximizes VoI accumulated. The multiple AUV path problem requires an additional step that is to determine how to distribute the nodes among the AUVs.

So the question is whether there are any physical tangible design choices that can improve VoI accumulated in a mobile sink based sensor network system. Examples of such design choices are shown in Figure 7.1. One possibility is to improve the rate of data collection from the sensor network. This minimizes time and should, therefore, improve VoI accumulated. This can be accomplished by a mobile sink that has a faster speed as shown in Figure 7.1(b). Another design choice can be using multiple AUVs as shown in Figure 7.1(c). Multiple AUVs can help to reduce the overall traversal time for covering all the nodes in the sensor network.

Let us assume a sensor network s_n with n nodes and an AUV a_1 that travels distance d to traverse all of these nodes. The map traversal time for this case is T_{Case-1}. We can divide these nodes into non-overlapping subsets s_{n_1} and s_{n_2} of sizes n_1 and n_2 respectively, such that, $n > n_1$, $n > n_2$ and $n = n_1 + n_2$. These subsets of nodes are assigned to a_1 and another new AUV a_2, i.e. $a_1 \leftarrow s_{n_1}$ and $a_2 \leftarrow s_{n_2}$. Let us assume that the schedule of visitation for subsets s_{n_1} and s_{n_2} is in the same sequence as it is in the visitation schedule for s_n. Let a_1 travel distance d_1 to cover n_1 nodes in time t_1, and a_2 travel distance d_2 to cover n_2 nodes in time t_2. Then, as the visitation sequences are in the same order, we can infer $d = d_1 + d_2$ and also $d > d_1$ & $d > d_2$. Now, if the tours have started such that their operational time overlaps each other, then the new map traversal time T_{Case-2} should guarantee the condition $T_{Case-2} < T_{Case-1} = t_1 + t_2$. Moreover, if the AUVs start their tours at the same time, then we can determine traversal time as either $T_{Case-3} = t_1$ or as $T_{Case-3} = t_2$, depending upon which is longer among t_1 and t_2.

This implies that $T_{Case-3} \leq T_{Case-2} < T_{Case-1}$. Thus, it can be easily inferred that if the nodes are split into subsets and assigned to AUVs which start their tour with a time overlap, then there should be a reduction in the overall map traversal time T.

We argued in Chapter 4 that time minimization heuristics can help in improving VoI accumulated. Therefore, we can safely say that using multiple AUVs can improve VoI accumulated by reducing the map traversal time.

7.3 Various Heuristic Combinations for Path Planning Algorithms

In Chapter 5, a tour is defined as a sequence of sensor nodes that the AUV traverses for data collection. The goal of a path planning algorithm is to find a tour for AUV that maximizes the VoI accumulated from the underwater sensor networks. Therefore, in the case of multiple AUVs, the path planning algorithms should find a schedule for each AUV, such that all the schedules collectively contribute towards maximizing VoI. The tours planned in this dissertation are all classified as Hamiltonian, i.e. any sensor node is only visited once during the traversal of a map. Hence, each sensor node will be uniquely assigned to an AUV. The algorithms assume a mesh deployment of nodes and can be easily extended to other node arrangements once the underlying concepts have been grasped. The concepts we are referring to are the heuristics developed in Chapter 4. The algorithms in this chapter employ the aforementioned heuristics in various combinations. Table 7.1 provides a listing of these combinations for each multiple AUV path planning algorithm. The columns in the table present the names of the heuristics while the rows correspond to the names of the path planning algorithms. Details of these various combinations are discussed in following subsections for each path planning algorithm.

Table 7.1: A listing of heuristics employed by various algorithms

		Heuristics					
		H_{MaxVoI}	H_{ShPath}	$H_{IntVisit}$	$H_{NodeBal}$	H_{VoIBal}	$H_{MapPart}$
Path Planning Algorithms	*RPP*				✓		
	ZPP				✓		✓
	LPP		✓		✓		✓
	GPP_N	✓			✓		
	$GIPP_N$	✓		✓	✓ !		
	GPP_{VB}	✓			✓	✓	
	$GIPP_{VB}$	✓		✓	✓ !	✓	
	GPP_{MP}	✓			✓		✓
	$GIPP_{MP}$	✓		✓	✓ !		✓

In Table 7.1, an exclamation mark '**!**' is used to identify combinations of the heuristic $H_{NodeBal}$ and path planning algorithms that are *IPPs*. *IPPs* implement $H_{IntVisit}$ using the *TourIntermediate()* procedure. This *TourIntermediate()* procedure, detailed in Algorithm 9, disturbs the exact node balancing which was earlier enforced due to $H_{NodeBal}$. The node balancing disturbance is because of the addition of smaller sub-tours. These smaller sub-tours, when incorporated in the main AUV tours, result in varying the length of the main tours. The algorithms that use $H_{IntVisit}$ are $GIPP_N$, $GIPP_{VB}$ (Algorithm 11) and $GIPP_{MP}$ (Algorithm 13). For all the path planners, supporting procedures are listed in Algorithm 8 and Algorithm 9. The procedures that help in determining VoI are given Algorithm 8, while Algorithm 9 includes procedures that assist in path planning by discovering sub-tours or by planning detailed geo-physical tours (although geo-physical tours are not in the scope of this dissertation, there is extensive literature on this topic).

7.4 Multiple AUV Path Planning Algorithms

The algorithms described in this section are:

- *RPP* - Random Path Planner

  $RPP \leftarrow H_{NodeBal}$

- *ZPP* - Zig-Zag Path Planner

  $ZPP \leftarrow H_{NodeBal} + H_{MapPart}$

- *LPP* - Lawn-Mower Path Planner

  $LPP \leftarrow H_{NodeBal} + H_{MapPart} + H_{ShPath}$

- GPP_N - Greedy Path Planner with Node Balancing

  $GPP_N \leftarrow H_{NodeBal} + H_{MaxVoI}$

- $GIPP_N$ - Greedy Path Planner with Node Balancing and Intermediate Node Visitation

  $GIPP_N \leftarrow H_{NodeBal} + H_{MaxVoI} + H_{IntVisit}$

- GPP_{VB} - Greedy Path Planner with VoI Balancing

  $GPP_{VB} \leftarrow H_{NodeBal} + H_{MaxVoI} + H_{VoIBal}$

- $GIPP_{VB}$ - Greedy Path Planner with VoI Balancing and Intermediate Node Visitation

  $GIPP_{VB} \leftarrow H_{NodeBal} + H_{MaxVoI} + H_{VoIBal} + H_{IntVisit}$

- GPP_{MP} - Map Partitioned Greedy Path Planner

  $GPP_{MP} \leftarrow H_{NodeBal} + H_{MaxVoI} + H_{MapPart}$

- $GIPP_{MP}$ - Map Partitioned Greedy Path Planner with Intermediate Node Visitation

  $GIPP_{MP} \leftarrow H_{NodeBal} + H_{MaxVoI} + H_{MapPart} + H_{IntVisit}$

7.4.1 *Random Path Planner - RPP*

This is the baseline path planner against which the performance of other path planning algorithms will be judged. In *RPP* nodes that have not yet been visited are chosen randomly from the map and then assigned to the AUVs in a round-robin fashion that helps node balancing. Therefore, the only heuristic *RPP* implements is $H_{NodeBal}$. This heuristic has been kept in the random path planner so that comparison with the other path-planners becomes more meaningful.

7.4.2 *Zig-Zag Path Planner - ZPP*

This is also a baseline path planner which implements two heuristics, namely $H_{MapPart}$ and $H_{NodeBal}$. Compared to *RPP* it shows the incremental effect of using map partitioning. However, the route it plans for the AUVs is predetermined like *LPP* path planner but it is non-optimal in terms of distance. To be precise, the routes *ZPP* generates are approximately double in length to the *LPP* algorithm.

It plans a row-by-row traversal of the mesh just as the *LPP* algorithm. However, once it reaches the end of a row, it starts traversing the immediate next row from the node that is farthest from the last node of the current row that the AUV just visited. In this manner, the AUVs traverse the nodes in the map in a row-by-row zig-zag manner, and so the name *ZPP*.

7.4.3 *Lawn-Mower Path Planner - LPP*

The Lawn-Mower path planner for multiple AUVs is a simple variant of the Lawn-Mower Algorithm in Chapter 5. This path planner is detailed as Algorithm 6. As the first step, this algorithm partitions the map into equal sized regions using procedure *PartitionMap*(). This enforces the heuristic $H_{MapPart}$ and $H_{NodeBal}$ in the path planner. Then, using a direction-priority-list, the algorithm plans a Lawn-Mower path for each AUV, thus encoding the heuristic H_{ShPath}. The final result of this path planner is each AUV having the shortest path tour in a mesh setting of nodes.

Algorithm 6 Shortest Path Lawn-Mower Path Planner – *LPP*

```
procedure LPP($\{s_1, s_2, ..., s_n\}, \{a_1, a_2, ..., a_a\}$)
    $S \leftarrow \{s_1, s_2, ..., s_n\}$                              ▷ Set of sensor nodes
    $A \leftarrow \{a_1, a_2, ..., a_a\}$                              ▷ Set of AUVs
    $P_D \leftarrow \{(East, West, South, North)\}$                   ▷ Direction priority list
    for $m \leftarrow 1$ to $a$ do
        $S_m \leftarrow \emptyset$                ▷ Subset of sensor nodes to be visited by $m_{th}$ AUV
        $V_m \leftarrow \emptyset$                ▷ Visitation sequence for $m_{th}$ AUV
    end for
    $S_T \leftarrow \{S_1, S_2, ..., S_a\}$                            ▷ Set of subsets $S_m$
    PARTITIONMAP($S_T, S$)        ▷ $H_{MapPart}$ & $H_{NodeBal}$ - Partition map while balancing nodes
    for $m \leftarrow 1$ to $a$ do
        $i \leftarrow$ GETTOURSTARTNODE($S_m, a_m$)
        while $S_m \neq \emptyset$ do
            $N \leftarrow$ NEIGHBORHOOD($i, S_m$)
            $j \leftarrow s_x$ from $N$ in the direction given by $P_D$
            $V_m \leftarrow V_m + j$
            $S_m \leftarrow S_m - j$
            $i \leftarrow j$
        end while
    end for
    $V_T \leftarrow \{(a_1, V_1), (a_2, V_2), ..., (a_a, V_a)\}$   ▷ Set of key-value pairs of AUVs and visitation sequences
    return $V_T$
end procedure
```

7.4.4 Greedy Path Planner with Node Balancing - GPP_N

This path planner is a simple greedy algorithm for scheduling visits by multiple AUVs. It is based on heuristics H_{MaxVoI} and $H_{NodeBal}$. Note that this algorithm does not employ H_{VoIBal}.

Algorithm 7 lists the details of GPP_N. The set of nodes are first divided into smaller non-overlapping subsets of size approximately n/a, where n is the number of sensor nodes and a is the number of AUVs. Because the subsets are chosen with no VoI-specific metric, therefore, the subsets may have a disproportionate number of hot-spots in them. After division into subsets, each subset is assigned to an AUV. The algorithm then uses H_{MaxVoI} to determine the traversal sequence for each AUV from among the subset of nodes that it has been assigned.

GPP_N serves as our basic greedy algorithm for scheduling multiple AUVs. We use it as a baseline for comparison with GPP_{VB} which employs the H_{VoIBal} heuristic for VoI balancing so as to avoid a disproportionate assignment of hot-spots. Results for GPP_N are also compared against GPP_{MP} which uses $H_{MapPart}$ to reduce average travelling times while still maintaining the node balancing heuristic $H_{NodeBal}$.

7.4.5 Greedy Path Planner with Node Balancing and Intermediate Node Visitation - $GIPP_N$

This algorithm is exactly similar to GPP_N in construction, except that, it uses $H_{IntVisit}$ to plan subtours between a source-destination pair once a destination node has been selected using H_{MaxVoI}. This sub-tour discovery is done using the procedure $TourIntermediate()$. Therefore, $H_{IntVisit}$ is enforced by the procedure $TourIntermediate()$. As stated earlier node balancing can be disturbed by $TourIntermediate()$ i.e. some AUVs might end up visiting slightly more number of nodes than the others. Hence, IPP algorithms can be termed as best effort algorithms for $H_{NodeBal}$ and not necessarily optimal.

Algorithm 7 Naive Greedy Path Planner – GPP_N

```
procedure $GPP_N(\{s_1, s_2, ..., s_n\}, \{a_1, a_2, ..., a_a\})$
    $S \leftarrow \{s_1, s_2, ..., s_n\}$                                    ▷ Set of sensor nodes
    $A \leftarrow \{a_1, a_2, ..., a_a\}$                                    ▷ Set of AUVs
    for $m \leftarrow 1$ to $a$ do
        $V_m \leftarrow \emptyset$
        if $m \leq (n \mod a)$ then               ▷ $H_{NodeBal}$ - Divide nodes as equally as possible
            $q \leftarrow \left\lceil \frac{n}{a} \right\rceil$
        else
            $q \leftarrow \left\lfloor \frac{n}{a} \right\rfloor$
        end if
        $i \leftarrow$ GETTOURSTARTNODE$(S, a_m)$                ▷ Last node visited by the $m_{th}$ AUV
        for $p \leftarrow 1$ to $q$ & $S \neq \emptyset$ do
            $j \leftarrow$ GETNODETHATHASMAXVOI$(S, i)$ ▷ $H_{MaxVoI}$ - Visit node with highest VoI on priority
            $V_m \leftarrow V_m + j$
            $S \leftarrow S - j$
            $i \leftarrow j$
        end for
    end for
    $V_T \leftarrow \{(a_1, V_1), (a_2, V_2), ..., (a_a, V_a)\}$     ▷ Set of key-value pairs of AUVs and visitation sequences
    return $V_T$
end procedure
```

Algorithm 8 Procedures for Handling Value of Information

```
procedure GetNodeThatHasMaxVoI($S_{remaining}, s_{current}$)
    $S_r \leftarrow S_{remaining}$
    $s_c \leftarrow s_{current}$
    if Valuation is based on Data Reports then
        $\forall s_x \in S_r$ determine $\Upsilon_{s_x}$ using DetermineNodeVoI($s_c, s_x, S_r, data$)
    else if Valuation is based on Node then
        $\forall s_x \in S_r$ determine $\Upsilon_{s_x}$ using DetermineNodeVoI($s_c, s_x, S_r, node$)
    end if
    $k \leftarrow s_x \in S_r$ such that $\Upsilon_{s_x}$ is max $\sum Ae^{-B(t-\tau_o)}$
    return $k$
end procedure
procedure DetermineNodeVoI($S_{remaining}, s_{current}, s_{target}, ValuationBasis$)
    $S_r \leftarrow S_{remaining}$
    $s_c \leftarrow s_{current}$
    $s_t \leftarrow s_{target}$
    $\tau_f \leftarrow$ GetExactTourTime($S_r, s_c, s_t$)
    if $ValuationBasis$ is $data$ then
        $D \leftarrow \{d_{s_t 1}, d_{s_t 2}, ..., d_{s_t d}\}$                ▷ Data reports at node $s_{target}$
        $\Upsilon \leftarrow 0$
        while $D \neq \emptyset$ do
            $d_y \leftarrow$ GetNextDataReport($D$)
            $\alpha \leftarrow$ Get$A(d_y)$
            $\beta \leftarrow$ Get$B(d_y)$
            $\tau_o \leftarrow$ Get$\tau_o(d_y)$
            $\Upsilon += \alpha e^{-\beta(\tau_f - \tau_o)}$
            $D \leftarrow D - d_y$
        end while
    else if $ValuationBasis$ is $node$ then
        $\alpha \leftarrow$ Get$A(s_t)$
        $\beta \leftarrow$ Get$B(s_t)$
        $\tau_o \leftarrow$ Get$\tau_o(s_t)$
        $\Upsilon \leftarrow \alpha e^{-\beta(\tau_f - \tau_o)}$
    end if
end procedure
```

Algorithm 9 Intermediate Node Traversal and Exact Travel Time Procedures

```
procedure TourIntermediate($source, destination, SensorNodes$)
    $p \leftarrow source$
    $q \leftarrow destination$
    $S_N \leftarrow SensorNodes$
    $T_{Int} \leftarrow \emptyset$                              ▷ Tour for intermediate node visitation sequence
    while $p \neq q$ do
        $N \leftarrow$ GetNeighborhood($p, S_N$)
        $i \leftarrow s_x \in N$ such that InterNodeDistance($s_x, q$) is minimized
        $T_{Int} \leftarrow T_{Int} + i$
        $p \leftarrow i$
    end while
    $T_{Int} \leftarrow T_{Int} + q$
    return $T_{int}$
end procedure
procedure PlanInDepthTour($source, destination, SensorNodes, PathPlannerType$)
    $Tour \leftarrow \emptyset$
    $Tour \leftarrow Tour + source$
    if $PathPlannerType$ is $intermediate\ node\ visitation$ then
        $Tour \leftarrow Tour+$ TourIntermediate($source, destination, SensorNodes$)
    else
        $Tour \leftarrow Tour + destination$
    end if
    $time \leftarrow$ RunGeoPhysicalSimulation($Tour, PhysicalDeploymentMap$)
    return $Tour, time$
end procedure
procedure GetExactTourTime($S_{remaining}, s_{current}, s_{target}$)
    $S_r \leftarrow S_{remaining}$
    $s_c \leftarrow s_{current}$
    $s_t \leftarrow s_{target}$
    $Tour, time \leftarrow$ PlanInDepthTour($S_r, s_c, s_t$)
    return $time$
end procedure
```

7.4.6 Greedy Path Planner with VoI Balancing - GPP_{VB}

This path planner is based on proposition H_{MaxVoI}, $H_{NodeBal}$ & H_{VoIBal}. The details of this algorithm can be seen in Algorithm 10.

The nodes are assigned in a round-robin fashion to the AUVs based on the VoI they have to offer. In descending priority they are assigned to an AUV. This priority-based round-robin arrangement enforces H_{VoIBal} and $H_{NodeBal}$. After each assignment, a tour is planned towards the node, thereby, encoding H_{MaxVoI} into the algorithm.

Algorithm 10 VoI Balanced Greedy Path Planner – GPP_{VB}

```
procedure $GPP_{VB}(\{s_1, s_2, ..., s_n\}, \{a_1, a_2, ..., a_a\})$
    $S \leftarrow \{s_1, s_2, ..., s_n\}$                                    ▷ Set of sensor nodes
    $A \leftarrow \{a_1, a_2, ..., a_a\}$                                          ▷ Set of AUVs
    for $m \leftarrow 1$ to $a$ do
        $V_m \leftarrow \emptyset$                                   ▷ Visitation sequence for $m_{th}$ AUV
        $i_m \leftarrow$ GETTOURSTARTNODE($S, a_m$)             ▷ Last node visited by the $m_{th}$ AUV
    end for
    $l \leftarrow 0$
    while $S \neq \emptyset$ do
        $m \leftarrow (l \mod a) + 1$                  ▷ $H_{VoIBal}$ & $H_{NodeBal}$ - Switch AUV in each iteration
        $j \leftarrow$ GETNODETHATHASMAXVOI($S, i_m$)    ▷ $H_{MaxVoI}$ - Visit node with highest VoI on priority
        $V_m \leftarrow V_m + j$
        $S \leftarrow S - j$
        $i_m \leftarrow j$
        $l \leftarrow l + 1$
    end while
    $V_T \leftarrow \{(a_1, V_1), (a_2, V_2), ..., (a_a, V_a)\}$    ▷ Set of key-value pairs of AUVs and visitation sequences
    return $V_T$
end procedure
```

7.4.7 Greedy Path Planner with VoI Balancing and Intermediate Node Visitation - $GIPP_{VB}$

The construction of this algorithm is identical to GPP_{VB} in Algorithm 10, except that, it uses $H_{IntVisit}$ to plan sub-tours between a source-destination pair once a destination node has been selected using H_{MaxVoI}. This sub-tour discovery is done using the procedure $TourIntermediate()$. The details of this path planner are given in Algorithm 11.

Algorithm 11 VoI Balanced Greedy Path Planner with $H_{IntVisit} - GIPP_{VB}$

```
procedure $GIPP_{VB}(\{s_1, s_2, ..., s_n\}, \{a_1, a_2, ..., a_a\})$
    $S \leftarrow \{s_1, s_2, ..., s_n\}$                              ▷ Set of sensor nodes
    $A \leftarrow \{a_1, a_2, ..., a_a\}$                                    ▷ Set of AUVs
    for $m \leftarrow 1$ to $a$ do
        $V_m \leftarrow \emptyset$                              ▷ Visitation sequence for $m_{th}$ AUV
        $i_m \leftarrow$ GETTOURSTARTNODE$(S, a_m)$           ▷ Last node visited by the $m_{th}$ AUV
    end for
    $l \leftarrow 0$
    while $S \neq \emptyset$ do
        $m \leftarrow m$ is index of $a_m$ such that $V_m$ has fewest nodes        ▷ $H_{VoIBal}$ & $H_{NodeBal}$
        $j \leftarrow$ GETNODETHATHASMAXVOI$(S, i_m)$   ▷ $H_{MaxVoI}$ - Visit node with highest VoI on priority
        $T \leftarrow \emptyset$
        $T \leftarrow$ TOURINTERMEDIATE$(i_m, j, S_m)$      ▷ $H_{IntVisit}$ - Intermediate node visitation sequence
        $V_m \leftarrow V_m + T$
        $S_m \leftarrow S_m - T$
        $i_m \leftarrow j$
    end while
    $V_T \leftarrow \{(a_1, V_1), (a_2, V_2), ..., (a_a, V_a)\}$   ▷ Set of key-value pairs of AUVs and visitation sequences
    return $V_T$
end procedure
```

7.4.8 Map Partitioned Greedy Path Planner - GPP_{MP}

The path planner GPP_{MP} first employs the $H_{MapPart}$ heuristic to partition the Map. The partitioning is done in a way to maintain heuristic $H_{NodeBal}$. Afterward, it employs H_{MaxVoI} to visit nodes with maximum VoI first. Algorithm 12 states the procedure for this path planner. Note how the algorithm uses heuristics $H_{MapPart}$, $H_{NodeBal}$ and H_{MaxVoI} in sequence for the path planning process. This algorithm does not implement the heuristic H_{VoIBal} as it is irrelevant here. This is because there in cross-sharing of nodes across partitions of the map for traversal purposes, and hence, no VoI balancing procedure can be implemented. If there are multiple high priority nodes such as hot-pots in a region, then they can only be visited by the AUV assigned to that partition.

Algorithm 12 Map Partitioned Greedy Path Planner – GPP_{MP}

```
procedure $GPP_{MP}(\{s_1, s_2, ..., s_n\}, \{a_1, a_2, ..., a_a\})$
    $S \leftarrow \{s_1, s_2, ..., s_n\}$    ▷ Set of sensor nodes
    $A \leftarrow \{a_1, a_2, ..., a_a\}$    ▷ Set of AUVs
    for $m \leftarrow 1$ to $a$ do
        $S_m \leftarrow \emptyset$    ▷ Subset of sensor nodes to be visited by $m_{th}$ AUV
        $V_m \leftarrow \emptyset$    ▷ Visitation sequence for $m_{th}$ AUV
    end for
    $S_T \leftarrow \{S_1, S_2, ..., S_a\}$    ▷ Set of subsets $S_m$
    PARTITIONMAP($S_T, S$)    ▷ $H_{MapPart}$ & $H_{NodeBal}$ - Partition map while balancing nodes
    for $m \leftarrow 1$ to $a$ do
        $i \leftarrow$ GETTOURSTARTNODE($S_m, a_m$)    ▷ Last node visited by the $m_{th}$ AUV
        while $S_m \neq \emptyset$ do
            $j \leftarrow$ GETNODETHATHASMAXVOI($S_m, i$)    ▷ $H_{MaxVoI}$ - Visit node with highest VoI
            $V_m \leftarrow V_m + j$
            $S_m \leftarrow S_m - j$
            $i \leftarrow j$
        end while
    end for
    $V_T \leftarrow \{(a_1, V_1), (a_2, V_2), ..., (a_a, V_a)\}$    ▷ Set of key-value pairs of AUVs and visitation sequences
    return $V_T$
end procedure
```

7.4.9 Map Partitioned Greedy Path Planner with Intermediate Node Visitation - $GIPP_{MP}$

The construction of $GIPP_{MP}$ is similar to GPP_{MP} as in Algorithm 12, but in additon it uses $H_{IntVisit}$ for intermediate node visitation. $GIPP_{MP}$ is given as Algorithm 13. Note how the algorithm uses heuristics $H_{MapPart}$, $H_{NodeBal}$, H_{MaxVoI} and $H_{IntVisit}$ for its planning process.

Algorithm 13 Map Partitioned Greedy Path Planner with $H_{IntVisit} - GIPP_{MP}$

```
procedure $GIPP_{MP}(\{s_1, s_2, ..., s_n\}, \{a_1, a_2, ..., a_a\})$
    $S \leftarrow \{s_1, s_2, ..., s_n\}$                                   ▷ Set of sensor nodes
    $A \leftarrow \{a_1, a_2, ..., a_a\}$                                   ▷ Set of AUVs
    for $m \leftarrow 1$ to $a$ do
        $S_m \leftarrow \emptyset$                     ▷ Subset of sensor nodes to be visited by $m_{th}$ AUV
        $V_m \leftarrow \emptyset$                     ▷ Visitation sequence for $m_{th}$ AUV
    end for
    $S_T \leftarrow \{S_1, S_2, ..., S_a\}$                                 ▷ Set of subsets $S_m$
    PARTITIONMAP($S_T, S$)        ▷ $H_{MapPart}$ & $H_{NodeBal}$ - Partition map while balancing nodes
    for $m \leftarrow 1$ to $a$ do
        $i \leftarrow$ GETTOURSTARTNODE($S_m, a_m$)          ▷ Last node visited by the $m_{th}$ AUV
        while $S_m \neq \emptyset$ do
            $j \leftarrow$ GETNODETHATHASMAXVOI($S_m, i$)     ▷ $H_{MaxVoI}$ - Visit node with highest VoI
            $T \leftarrow \emptyset$
            $T \leftarrow$ TOURINTERMEDIATE($i, j, S_m$)   ▷ $H_{IntVisit}$ - Intermediate node visitation sequence
            $V_m \leftarrow V_m + T$
            $S_m \leftarrow S_m - T$
            $i \leftarrow j$
        end while
    end for
    $V_T \leftarrow \{(a_1, V_1), (a_2, V_2), ..., (a_a, V_a)\}$   ▷ Set of key-value pairs of AUVs and visitation sequences
    return $V_T$
end procedure
```

7.5 Simulation Setup & Results

We consider an underwater sensor network of 100 nodes deployed in a uniform grid over a 10x10 km^2 area as shown in Figure 3.3. This renders an inter-node distance of 1 *km* each for the vertical and horizontal neighbors, while the distance between neighboring nodes in the diagonals is 1.41 *km*. We consider an AUV that operates at a speed of 10 m/s. Such a setting can, for example, be deployed in the Strait of Gibraltar. This is shown in Figure 7.2 where the deployment is to scale and the nodes are approximately a kilometer apart each.

We evaluate the comparative performance of the path planning algorithms through three different experiments. First, we assess the VoI collected in the case of two AUVs and five AUVs. We also experiment with various distributions of hot-spots to see their effect on VoI collected by the path planners. Then we asses the time required to reach the first hot-spot so as to compare the emergency response of various path planning algorithms. Lastly, we analyze the average distance traveled by the AUVs given one, two, five and ten AUVs.

For the simulation setup, we iterate over multiple scenarios, each of which corresponds to a particular UWSN VoI setting. The results are averaged over 150 different arrangements of the UWSN setting for VoI. Over each arrangement, all the path planners are executed to gauge their performance. In all of the scenarios, we consider four hot-spots which are situated at variable locations across the simulation iterations. All the valuation coefficients A are set to unity i.e. $A = 1$ for all data segments residing on nodes either inside or outside the hot-spot regions. We have a binary model for the decay coefficient B i.e. it has a different value for indicating whether a data segment is reporting a hot-spot or a normal event. To each sensor node, we assign a random number of data segments. The starting point of the AUVs is located at the boundary of the mesh deployment. Multiple AUVs are deployed in an equidistant fashion from each other.

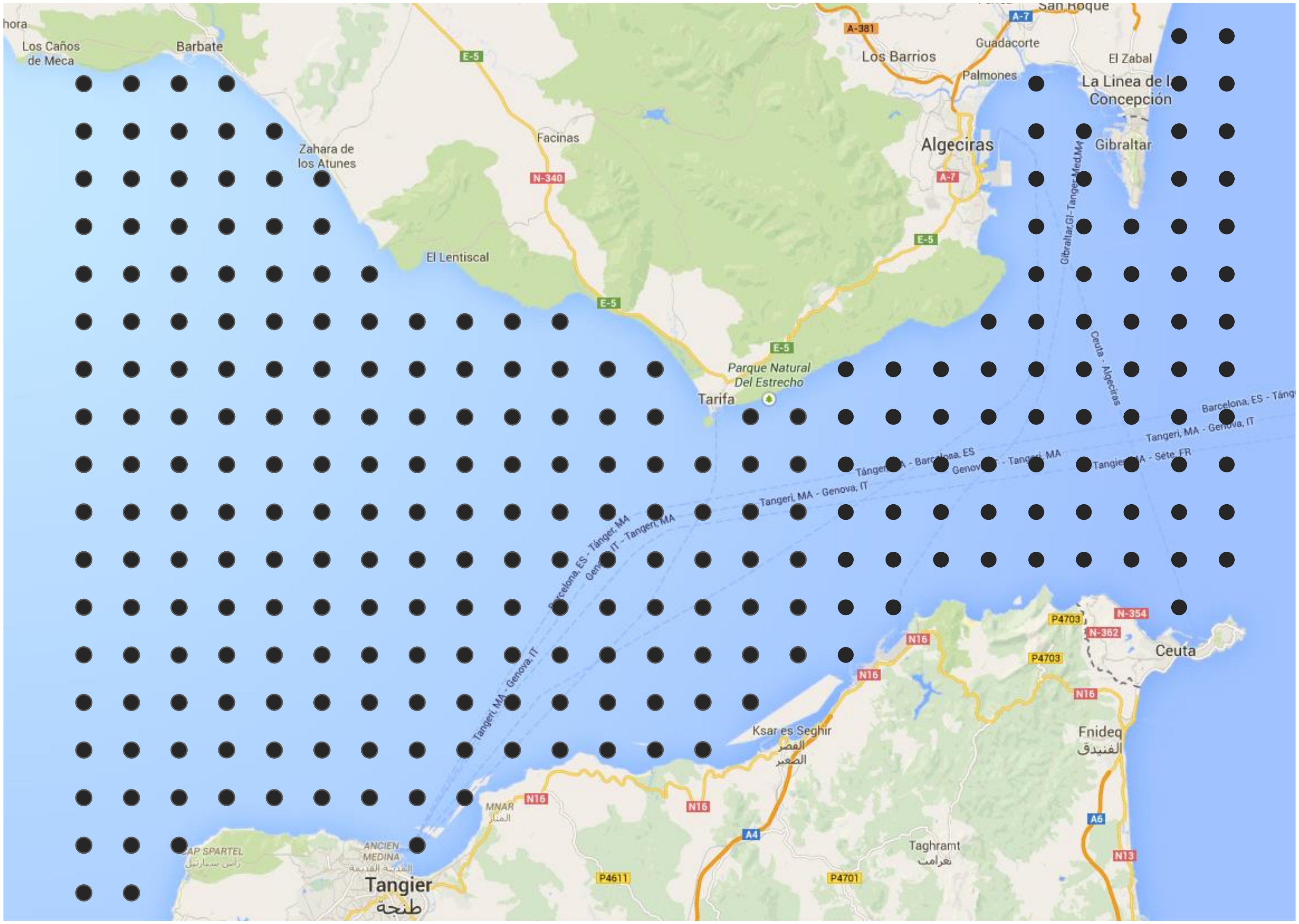


Figure 7.2: An example of UWSN deployed in the Strait of Gibraltar. The inter-node distance is approximately equal to 1 *km* and is drawn to-scale in the figure.

7.5.1 VoI Accumulated in various Spatial Distributions of Hot-Spots

By spatial distribution we imply how the hot-spots are distributed across the UWSN map; are they in close proximity i.e. collocated on the map, or are they evenly distributed across the map, or are they just randomly occurring in nature. These spatial distributions will help us in specifically assessing the performance of path planners in terms of map partitioning i.e. the heuristic $H_{MapPart}$. In this experiment, we consider four different deployments of the hot-spots:

- *No Hot-Spots*: This will be used as a baseline in which all events reported will be normal. During the course of the simulation, no active hot-spot will be reported.
- *Random*: Here we will generate hot-spots in randomly selected locations of the map.
- *Even*: This scenario is to simulate a balanced distribution of hot-spots across the map. The map is divided into as many similarly sized regions as are the number of hot-spots. A hot-spot is initialized in every region of the map.
- *Skewed*: In this scenario, we will randomly select a portion of the map and then initialize hot-spots in only this selected portion of the map.

We analyze results for the case where two AUVs (Figure 7.5.1) and five AUVs (Figure 7.5.1) are deployed for data collection. From the results in Figure 7.5.1, we can see that in the *No Hot-Spots* scenario, *LPP* performs the best in terms of VoI accumulated, and this is because it minimizes the AUV traversal time. This result is consistent with the results for the single AUV *LPP* path planner in Chapter 5. For the remaining scenarios, i.e. *Random*, *Even* and *Skewed* distribution of hot-spots, the greedy algorithms with map partitioning and load balancing perform better in terms of VoI accumulated.

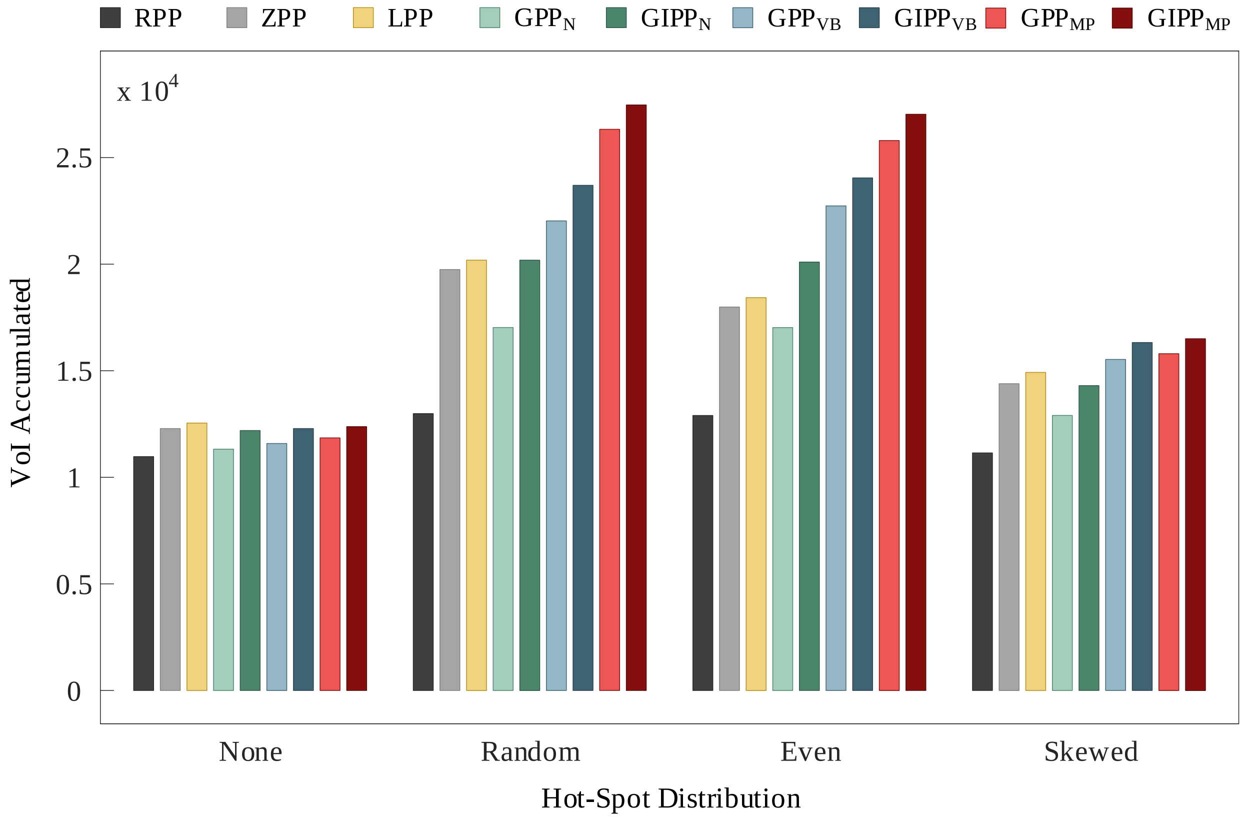


Figure 7.3: VoI accumulated by various path planners with two AUVs.

With algorithms GPP_N and $GIPP_N$ we show that it was not enough to just distribute the node equally among the AUVs. Therefore, these algorithms are designed with only the $H_{NodeBal}$ multiple AUV heuristic. Other than this they also have the heuristics H_{MaxVoI} and H_{ShPath} encoded in them with $H_{IntVisit}$ present additionally in $GIPP_N$. From the results, we can see that GPP_N and $GIPP_N$ do worse than the other greedy variants that are GPP_{VB}, $GIPP_{VB}$, GPP_{MP} and $GIPP_{MP}$. It is interesting to note that it is also outperformed by the other map partitioning algorithms LPP and ZPP. This speaks of the effectiveness of using $H_{MapPart}$ in the path planning algorithms.

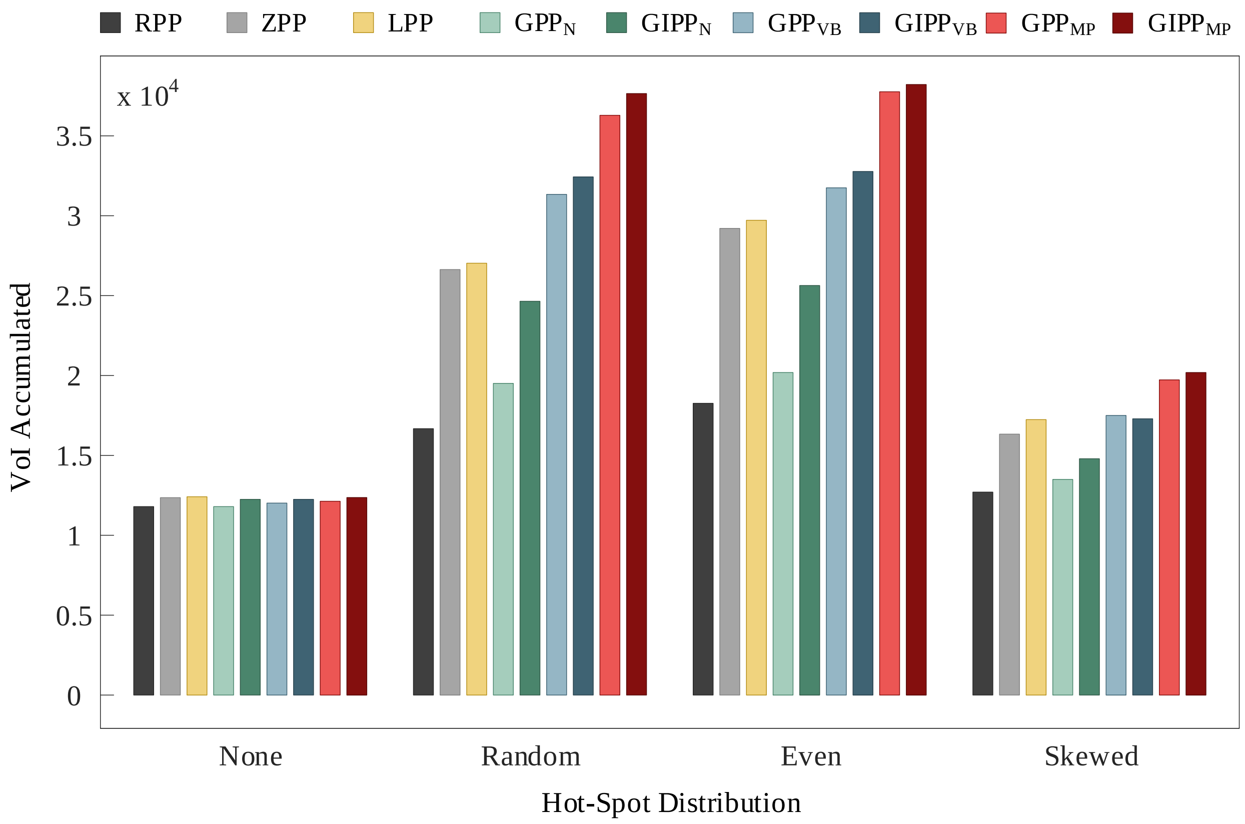


Figure 7.4: VoI accumulated by various path planners with five AUVs.

The greedy algorithms GPP_{VB} and $GIPP_{VB}$ are designed with keeping VoI balancing in the view using the heuristic H_{VoIBal}. The utility of this heuristic is immediately apparent in the results where they perform better than their GPP_N and $GIPP_N$ counterparts.

The best performing algorithms are GPP_{MP} and $GIPP_{MP}$. The collect the highest amount of VoI. In all the distribution of hot-spots, including the case of *No Hot-Spots*, the best-performing algorithms are always those which deploy the $H_{MapPart}$ heuristic. Note that in case of *Skewed* the hot-spots have been deployed in a manner that they'll be assigned only to a single AUV if the algorithm uses $H_{MapPart}$, therefore, implying that there will be no VoI balancing. Even in this case where VoI balancing is at its worst, the map partitioning algorithms work very well, thereby, highlighting the highly positive impact of time minimization heuristics.

The effect of time minimization can also be seen by comparing between the *IPP* variants of the greedy algorithms. The *IPP* variants employ $H_{IntVisit}$ which reduces travelling time. All *IPPs* pereform better than their non-*IPP* counterparts i.e. $GIPP_N$, $GIPP_{VB}$ and $GIPP_{MP}$ perform better than GPP_N, GPP_{VB} and GPP_{MP} respectively. Again, this result is consistent with findings in Chapter 5 where *GIPP* collected more VoI than *GPP* for a single AUV.

The results for five AUVs in Figure 7.5.1 corroborate the results for two AUVs shown in Figure 7.5.1. Therefore, the results can be scaled to an increasing number of AUVs.

Other than these comparisons, what is most important to note is that five AUVs collect more VoI than two AUVs. This reiterates the argument, which is the basis of this chapter, that using multiple AUVs improves the amount VoI collected.

As a closing remark to this analysis we can make two general inferences on performance of heuristics in terms of VoI collection; one is that for multiple AUVs $H_{MapPart} > H_{VoIBal} > H_{NodeBal}$; while in terms of a singular AUV entity among multiple AUVs $H_{IntVisit} + H_{MaxVoI} > H_{MaxVoI}$.

7.5.2 AUV Tour Time with increasing number of AUVs

In this experiment, we study that how an increasing number of AUVs contribute to lesser map traversal time and, therefore, a higher VoI collected. Moreover, the conjecture while proposing the $H_{MapPart}$ heuristic was that it contributes to reducing AUV travel time. In this experiment, we will gauge whether this conjecture is correct.

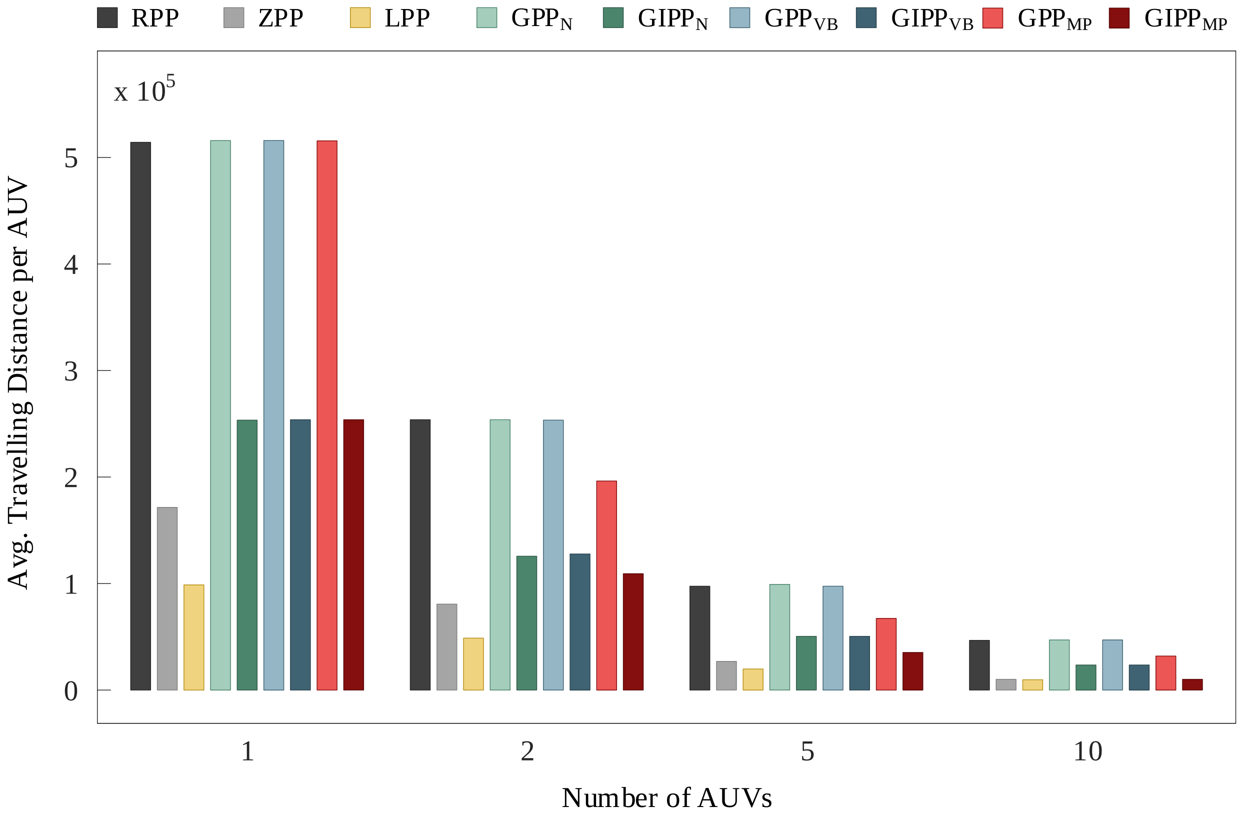


Figure 7.5: Average distance travelled by each AUV with increasing number of AUVs.

In this study we have recorded distance traveled, but as the underlying operation of our simulations strictly based upon the equation $S = V \times T$ i.e. distance traveled S is directly proportional to velocity V and traveling time T, therefore, we can use the results of distance traveled to make factually correct inferences of tour time.

We record the average distance traveled by the AUVs for various path planning algorithms. We vary the number of AUVs from one to ten. The hot-spots are located randomly on the map, however, their existence should not affect this study at all i.e. even in the case that there are no hot-spots, the results should be similar to the graphs generated for this study.

From results in Figure 7.5, we see that the shortest path algorithm *LPP* performs the best in terms of distance traveled. Also the *IPPs* have a lesser traveling distance as compared to their non-*IPP* counterparts. This explains why algorithms that employ $H_{IntVisit}$ perform better than those that do not employ this heuristic. All of these results are consistent with the single AUV findings in Chapter 5.

We can also observe that greedy map partitioning algorithms travel a smaller distance to traverse the map as compared to greedy algorithms that do not employ $H_{MapPart}$. This establishes the conjecture that map partitioning should result in lesser traveling time.

With an increase in the number of AUVs, we observe a reduction in the average distance traveled by the AUVs. This should be intuitive as the responsibility of visiting and collecting data from all the nodes is now being shared by a greater the number of AUVs. Therefore, the more the AUVs the lesser is the tour time and hence greater the VoI collected.

Map partitioning algorithms, and for that matter also *IPPs* can also be advocated on the basis that they result in fuel savings because of shorter tour times or distances.

Overall, $GIPP_{MP}$ seems to be a good candidate for being the best algorithm as it combines the good in both experiments i.e. highest VoI collection with shorter tour times and good fuel savings.

7.5.3 Emergency Response - Time to hit First Hot-Spot

In Chapter 5 we performed an emergency response analysis on the path planning algorithms. We will do a similar analysis in this section by analyzing the time required to hit the first hot-spot by various path planning algorithms.

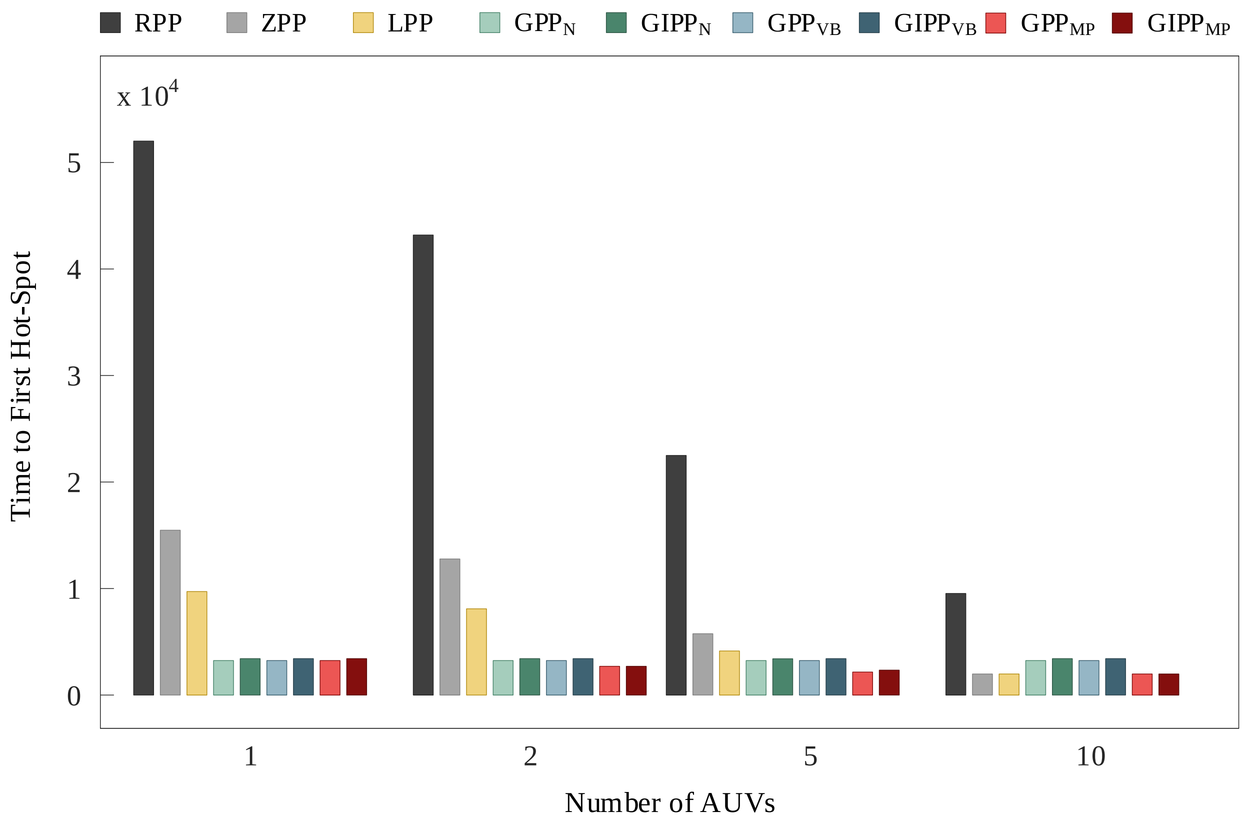


Figure 7.6: Time taken to hit the first hot-spot given the hot-spots are randomly distributed.

For this experiment, we use a *Random* spatial distribution for deploying hot-pots in the map. The first observation from the results in Figure 7.6 is that the greedy algorithms are best in terms of hitting the target at the earliest. Moreover, in this experiment, the non-*IPP* versions of the algorithms come out on top of the *IPP* variants. Clearly, if reaching a hot-spot is a priority then $H_{IntVisit}$ should be skipped for that segment of the tour.

Map partitioned algorithms perform the best in this experiment. The reason is that the algorithms have been not designed in a way that the AUV closest to the hot-spot visits it first, but the map-partitioned algorithms have this feature automatically encoded in them. Because each AUV is assigned to a specific region, therefore, if there is a hot-spot in its region then the AUV would be automatically the closest one to visit it. This feature i.e. assigning a hot-spot to the closest AUV can be included in the algorithms with ease by not assigning nodes in a blind round-robin fashion. If this is made part of the algorithms then the other *GPP* and *GIPP* variants will also perform as good as the GPP_{MP} and $GIPP_{MP}$ path planners in terms of emergency response

The result for 10 AUVs is an anomaly in terms of the performance of *LPP*. Here *LPP* is as good as GPP_{MP} and $GIPP_{MP}$. This result is an artifact of the way the map has been partitioned for this case as here the partitions are singular rows with 10 nodes each. The AUV only has to follow a straight line to get to the hot-spot and, hence, the identical performance in terms of getting to the first hot-spot.

7.6 Remarks

In this chapter we conclude, that uses multiple mobile sinks for data collection improves VoI accumulated. Scheduling node visitation sequences for multiple AUVs introduces challenges that are different from the single AUV path planning problem. We have developed a number of algorithms based on a combination of heuristics. Through detailed simulations, we demonstrated the efficacy of different heuristics and algorithms. We conclude that node balance map partitioning is a very effective heuristic for VoI accumulation. Coupling this with intermediate node visitation gives us the highest VoI accumulation results.

CHAPTER 8: CONCLUSION

In this dissertation, we have used value of information (VoI) in the form of Infotentials for path planning of autonomous underwater vehicles (AUVs) in underwater sensor networks (UWSNs). These types of networks have wireless communications issues and transferring large amounts of data can be problematic due to low-bandwidth acoustic channels. Therefore, one of the strategies proposed by the researchers is the use of AUVs which act as data mules for retrieving data from the subsurface sensor nodes.

We propose the use of time-decaying VoI as a means to develop a mechanism in which nodes in high-priority regions can be distinguished from nodes in low-priority regions. We develop a UWSN model for VoI using Infotentials and based on this, we further develop the VoI maximization and AUV path planning algorithms for various scenarios.

We propose a greedy approach in which the nodes advertise the VoI they offer at various times and the AUV traverses the nodes in descending order of the VoI gains from the node visits. We explore this approach in greater detail by discovering factors that affect VoI in path planning and then proposing several path planning algorithms accordingly. We also propose three heuristics, namely, VoI maximization, tour time minimization by shortest-path, and tour time minimization by intermediate node visitation. Based on these heuristics, we develop various algorithms and then discuss various factors and scenarios under which these algorithms perform comparative to each other.

One of these factors is the valuation ratio which tells whether a VoI maximization strategy or a time minimization strategy would perform better for a certain value of this ratio. The other factor is the emergency response, which gives insights for the manner in which one wants to design a node visitation strategy in case of emergency situations.

We also explore the multiple AUV path planning with the goal of maximizing VoI retrieved from the underwater sensor networks. We propose an extended range of heuristics for aiding path planning algorithms. We argue that an equal distribution of nodes among various AUVs, i.e. load-balancing among AUVs, improves the VoI accumulation. We further show that distributing the nodes in a way that VoI is distributed more fairly, i.e. balancing-VoI, increases the amount of VoI gathered. Lastly, we discover that if the map is partitioned in a way that the AUVs have to travel a smaller distance on average as compared to roaming the whole map, then VoI accumulation is improved by virtue of saving time which is in turn a factor in minimizing losses in Infotentials, i.e. time decaying VoI functions.

In the aforementioned problems the AUV acts as the sink node and, therefore, VoI decay stops once the AUV has retrieved data from the sensor nodes. In such a case the AUV is acting as the node where sensor fusion takes place and, hence, can be deemed as a sink node. The other scenario can be where the AUV is not well-equipped or informed-enough to perform the required sensor fusion activity or trigger any necessary actuation decisions. In such a case the AUV will need to resurface to transmit data to a remote sink node, and hence, the VoI will stop decaying once it is received by the remote sink node. This dramatically changes the path planning strategies. We highlight various challenges in this regard. We study the impact of inter-node distance and resurfacing distance on VoI accumulation. We then find a range where optimizing a resurfacing schedule for path planning purposes should yield better results. We also find the range where resurfacing at every node or not resurfacing until the very end of the tour makes more sense. Given a planned path, using path planning algorithms in Chapter 5 & 7, we augment it with optimal resurfacing locations. We use genetic algorithms for discovering these optimal resurfacing schedules. We also propose the use of a periodic resurfacing heuristic to develop an algorithm that is almost as efficient as the optimal VoI genetic algorithm.

LIST OF REFERENCES

[1] H. Shen and G. Bai, "Routing in wireless multimedia sensor networks: A survey and challenges ahead," *Journal of Network and Computer Applications*, vol. 71, pp. 30–49, 2016.

[2] C. Lv, Q. Wang, W. Yan, and Y. Shen, "Energy-balanced compressive data gathering in wireless sensor networks," *Journal of Network and Computer Applications*, vol. 61, pp. 102–114, 2016.

[3] M. Ayaz, I. Baig, A. Abdullah, and I. Faye, "A survey on routing techniques in underwater wireless sensor networks," *Journal of Network and Computer Applications*, vol. 34, no. 6, pp. 1908–1927, 2011.

[4] R. R. Esch, F. Protti, and V. C. Barbosa, "Adaptive event sensing in networks of autonomous mobile agents," *Journal of Network and Computer Applications*, 2016.

[5] C. Bisdikian, L. M. Kaplan, M. B. Srivastava, D. J. Thornley, D. Verma, and R. I. Young, "Building principles for a quality of information specification for sensor information," in *Information Fusion, 2009. FUSION'09. 12th International Conference on*, pp. 1370–1377, IEEE, 2009.

[6] C. Bisdikian, J. Branch, K. K. Leung, and R. I. Young, "A letter soup for the quality of information in sensor networks," in *Pervasive Computing and Communications, 2009. PerCom 2009. IEEE International Conference on*, pp. 1–6, IEEE, 2009.

[7] C. Bisdikian, L. M. Kaplan, and M. B. Srivastava, "On the quality and value of information in sensor networks," *ACM Transactions on Sensor Networks (TOSN)*, vol. 9, no. 4, p. 48, 2013.

[8] D. Chen and P. K. Varshney, "Qos support in wireless sensor networks: A survey.," in *International conference on wireless networks*, vol. 233, pp. 1–7, 2004.

[9] V. Sachidananda, A. Khelil, and N. Suri, “Quality of information in wireless sensor networks: A survey,” in *International Conference on Information Quality*, 2010.

[10] G. T. Singh and F. M. Al-Turjman, “A data delivery framework for cognitive information-centric sensor networks in smart outdoor monitoring,” *Computer Communications*, vol. 74, pp. 38–51, 2016.

[11] C. Aurrecoechea, A. T. Campbell, and L. Hauw, “A survey of qos architectures,” *Multimedia systems*, vol. 6, no. 3, pp. 138–151, 1998.

[12] A. Boukerche, M. Ahmad, B. Turgut, and D. Turgut, “A taxonomy of ad hoc routing protocols,” in *Algorithms and Protocols for Wireless, Mobile Ad hoc Networks* (A. Boukerche, ed.), ch. 5, pp. 129–164, Wiley, 2008.

[13] A. Boukerche, M. Ahmad, B. Turgut, and D. Turgut, “A taxonomy of routing protocols in sensor networks,” in *Algorithms and Protocols for Wireless Sensor Networks* (A. Boukerche, ed.), ch. 6, pp. 129–160, Wiley, 2008.

[14] S. Ehsan and B. Hamdaoui, “A survey on energy-efficient routing techniques with qos assurances for wireless multimedia sensor networks,” *IEEE Communications Surveys & Tutorials*, vol. 14, no. 2, pp. 265–278, 2012.

[15] M. Z. Hasan, H. Al-Rizzo, and F. Al-Turjman, “A survey on multipath routing protocols for qos assurances in real-time wireless multimedia sensor networks,” *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1424–1456, 2017.

[16] L. Bölöni and D. Turgut, “Should I send now or send later? a decision-theoretic approach to transmission scheduling in sensor networks with mobile sinks,” *Special Issue of Wiley’s Wireless Communications and Mobile Computing Journal (WCMC) on Mobility Management and Wireless Access*, vol. 8, pp. 385–403, March 2008.

[17] D. Turgut and L. Bölöni, “Heuristic approaches for transmission scheduling in sensor networks with multiple mobile sinks,” *The Computer Journal*, vol. 54, pp. 332–344, March 2011.

[18] D. Turgut and L. Bölöni, “Three heuristics for transmission scheduling in sensor networks with multiple mobile sinks,” in *Proceedings of International Workshop on Agent Technology for Sensor Networks (ATSN-08), in conjunction with the Seventh Joint Conference on Autonomous and Multi-Agent Systems (AAMAS 2008)*, pp. 1–8, May 2008.

[19] L. Hanzo and R. Tafazolli, “A survey of qos routing solutions for mobile ad hoc networks,” *IEEE Communications Surveys & Tutorials*, vol. 9, no. 2, pp. 50–70, 2007.

[20] L. Chen and W. B. Heinzelman, “A survey of routing protocols that support qos in mobile ad hoc networks,” *IEEE Network*, vol. 21, no. 6, 2007.

[21] S. H. H. Nazhad, M. Shojafar, S. Shamshirband, and M. Conti, “An efficient routing protocol for the qos support of large-scale manets,” *International Journal of Communication Systems*, vol. 31, no. 1, p. e3384, 2018.

[22] A. Boukerche, B. Turgut, N. Aydin, M. Ahmad, L. Bölöni, and D. Turgut, “Routing protocols in ad hoc networks: a survey,” *Computer Networks (Elsevier)*, vol. 55, pp. 3032–3080, September 2011.

[23] G. Wang, D. Turgut, L. Bölöni, Y. Ji, and D. Marinescu, “Improving routing performance through m-limited forwarding in power-constrained wireless networks,” *Journal of Parallel and Distributed Computing (JPDC)*, vol. 68, pp. 501–514, April 2008.

[24] R. Rahmatizadeh, S. Khan, A. Jayasumana, D. Turgut, and L. Bölöni, “Routing towards a mobile sink using virtual coordinates in a wireless sensor network,” in *Proceedings of IEEE ICC’14*, pp. 12–17, June 2014.

[25] R. Rahmatizadeh, S. Khan, A. Jayasumana, D. Turgut, and L. Bölöni, "Circular update directional virtual coordinate routing protocol in sensor networks," in *Proceedings of IEEE GLOBECOM'15*, pp. 1–6, December 2015.

[26] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of things (iot): A vision, architectural elements, and future directions," *Future generation computer systems*, vol. 29, no. 7, pp. 1645–1660, 2013.

[27] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*, pp. 13–16, ACM, 2012.

[28] D. Rosário, Z. Zhao, A. Santos, T. Braun, and E. Cerqueira, "A beaconless opportunistic routing based on a cross-layer approach for efficient video dissemination in mobile multimedia iot applications," *Computer communications*, vol. 45, pp. 21–31, 2014.

[29] D. Airehrour, J. Gutierrez, and S. K. Ray, "Secure routing for internet of things: A survey," *Journal of Network and Computer Applications*, vol. 66, pp. 198–213, 2016.

[30] P. G. V. Naranjo, M. Shojafar, H. Mostafaei, Z. Pooranian, and E. Baccarelli, "P-sep: A prolong stable election routing algorithm for energy-limited heterogeneous fog-supported wireless sensor networks," *The Journal of Supercomputing*, vol. 73, no. 2, pp. 733–755, 2017.

[31] M. Chu, H. Haussecker, and F. Zhao, "Scalable information-driven sensor querying and routing for ad hoc heterogeneous sensor networks," *International Journal of High Performance Computing Applications*, vol. 16, no. 3, pp. 293–313, 2002.

[32] F. Zhao, J. Shin, and J. Reich, "Information-driven dynamic sensor collaboration," *IEEE Signal processing magazine*, vol. 19, no. 2, pp. 61–72, 2002.

[33] H.-X. Tan, M.-C. Chan, W. Xiao, P.-Y. Kong, and C.-K. Tham, "Information quality aware routing in event-driven sensor networks," in *Proc. of IEEE INFOCOM*, pp. 1–9, IEEE, 2010.

[34] D. Turgut and L. Bölöni, "A pragmatic value-of-information approach for intruder tracking sensor networks," in *Communications (ICC), 2012 IEEE International Conference on*, pp. 4931–4936, IEEE, 2012.

[35] D. Turgut and L. Bölöni, "IVE: improving the value of information in energy-constrained intruder tracking sensor networks," in *Communications (ICC), 2013 IEEE International Conference on*, pp. 6360–6364, IEEE, 2013.

[36] V. Pryyma, D. Turgut, and L. Bölöni, "Active time scheduling for rechargeable sensor networks," *Computer Networks*, vol. 54, no. 4, pp. 631–640, 2010.

[37] V. Pryyma, L. Bölöni, and D. Turgut, "Uniform sensing protocol for autonomous rechargeable sensor networks," in *ACM/IEEE International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM'08)*, pp. 92–99, October 2008.

[38] N. Nasser, L. Karim, and T. Taleb, "Dynamic multilevel priority packet scheduling scheme for wireless sensor network," *IEEE transactions on wireless communications*, vol. 12, no. 4, pp. 1448–1459, 2013.

[39] L. Huang and M. J. Neely, "Utility optimal scheduling in energy-harvesting networks," *IEEE/ACM Transactions on Networking (TON)*, vol. 21, no. 4, pp. 1117–1130, 2013.

[40] X. Yu, X. Xiaosong, and W. Wenyong, "Priority-based low-power task scheduling for wireless sensor network," in *Autonomous Decentralized Systems, 2009. ISADS'09. International Symposium on*, pp. 1–5, IEEE, 2009.

[41] H. Hao, K. Wang, H. Ji, X. Li, and H. Zhang, "Utility-based scheduling algorithm for wireless multi-media sensor networks," in *Personal, Indoor, and Mobile Radio Communications (PIMRC), 2015 IEEE 26th Annual International Symposium on*, pp. 1052–1056, IEEE, 2015.

[42] M. A. Yigitel, O. D. Incel, and C. Ersoy, "Qos-aware mac protocols for wireless sensor networks: A survey," *Computer Networks*, vol. 55, no. 8, pp. 1982–2004, 2011.

[43] H. Kim and S.-G. Min, "Priority-based qos mac protocol for wireless sensor networks," in *Parallel & Distributed Processing, 2009. IPDPS 2009. IEEE International Symposium on*, pp. 1–8, IEEE, 2009.

[44] Y. Gu, F. Ren, Y. Ji, and J. Li, "The evolution of sink mobility management in wireless sensor networks: A survey," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 507–524, 2016.

[45] R. Sugihara and R. K. Gupta, "Speed control and scheduling of data mules in sensor networks," *ACM Transactions on Sensor Networks (TOSN)*, vol. 7, no. 1, p. 4, 2010.

[46] R. Sugihara and R. K. Gupta, "Path planning of data mules in sensor networks," *ACM Transactions on Sensor Networks (TOSN)*, vol. 8, no. 1, p. 1, 2011.

[47] J. Heidemann, M. Stojanovic, and M. Zorzi, "Underwater sensor networks: applications, advances and challenges," *Phil. Trans. R. Soc. A*, vol. 370, no. 1958, pp. 158–175, 2012.

[48] I. Vasilescu, K. Kotay, D. Rus, M. Dunbabin, and P. Corke, "Data collection, storage, and retrieval with an underwater sensor network," in *Proceedings of the 3rd international conference on Embedded networked sensor systems*, pp. 154–165, ACM, 2005.

[49] P. A. Forero, S. K. Lapic, C. Wakayama, and M. Zorzi, "Rollout algorithms for data storage- and energy-aware data retrieval using autonomous underwater vehicles," in *Proceedings of the International Conference on Underwater Networks & Systems*, p. 22, ACM, 2014.

[50] G. A. Hollinger, S. Choudhary, P. Qarabaqi, C. Murphy, U. Mitra, G. S. Sukhatme, M. Stojanovic, H. Singh, and F. Hover, "Underwater data collection using robotic sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 30, no. 5, pp. 899–911, 2012.

[51] E. Magistretti, J. Kong, U. Lee, M. Gerla, P. Bellavista, and A. Corradi, "A mobile delay-tolerant approach to long-term energy-efficient underwater sensor networking," in *2007 IEEE Wireless Communications and Networking Conference*, pp. 2866–2871, IEEE, 2007.

[52] L. Bölöni, D. Turgut, S. Basagni, and C. Petrioli, "Scheduling data transmissions of underwater sensor nodes for maximizing value of information," in *Global Communications Conference (GLOBECOM)*, pp. 438–443, 2013.

[53] P. Gjanci, C. Petrioli, S. Basagni, C. A. Phillips, L. Bölöni, and D. Turgut, "Path finding for maximum value of information in multi-modal underwater wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 17, no. 2, pp. 404–418, 2018.

[54] S. Basagni, L. Bölöni, P. Gjanci, C. Petrioli, C. Phillips, and D. Turgut, "Maximizing the value of sensed information in underwater wireless sensor networks via an autonomous underwater vehicle," in *IEEE INFOCOM'14*, pp. 988–996, April-May 2014.

[55] F. Khan, S. Butt, S. Khan, D. Turgut, and L. Bölöni, "Value of information based data retrieval in UWSNs," *Sensors*, vol. 18, October 2018.

[56] F. A. Khan, S. A. Khan, D. Turgut, and L. Bölöni, "Greedy path planning for maximizing value of information in underwater sensor networks," in *Proc. of IEEE 39th Conf. on Local Computer Networks Workshops (LCN Workshops)*, pp. 610–615, 2014.

[57] F. A. Khan, S. A. Khan, D. Turgut, and L. Bölöni, "Scheduling multiple mobile sinks in underwater sensor networks," in *Proc. of IEEE 40th Conf. on Local Computer Networks (LCN)*, pp. 149–156, 2015.

[58] F. A. Khan, S. A. Khan, D. Turgut, and L. Bölöni, “Optimizing resurfacing schedules to maximize value of information in uwsns,” in *Global Communications Conference (GLOBECOM)*, pp. 1–5, 2016.

[59] S. M. LaValle, *Planning algorithms*. Cambridge university press, 2006.

[60] Y. K. Hwang and N. Ahuja, “Gross motion planning—a survey,” *ACM Computing Surveys (CSUR)*, vol. 24, no. 3, pp. 219–291, 1992.

[61] C. Vasudevan and K. Ganesan, “Case-based path planning for autonomous underwater vehicles,” *Autonomous Robots*, vol. 3, no. 2-3, pp. 79–89, 1996.

[62] K. P. Carroll, S. R. McClaran, E. L. Nelson, D. M. Barnett, D. K. Friesen, and G. William, “AUV path planning: an A* approach to path planning with consideration of variable vehicle speeds and multiple, overlapping, time-dependent exclusion zones,” in *Proc. of IEEE Symposium on Autonomous Underwater Vehicle Technology*, pp. 79–84, 1992.

[63] C. Petres, Y. Pailhas, P. Patron, Y. Petillot, J. Evans, and D. Lane, “Path planning for autonomous underwater vehicles,” *IEEE Transactions on Robotics*, vol. 23, no. 2, pp. 331–341, 2007.

[64] B. Garau, A. Alvarez, and G. Oliver, “Path planning of autonomous underwater vehicles in current fields with complex spatial variability: an A* approach,” in *Proc. of IEEE Int’l Conf. on Robotics and Automation (ICRA)*, pp. 194–198, 2005.

[65] D. Fu-guang, J. Peng, B. Xin-qian, and W. Hong-Jian, “AUV local path planning based on virtual potential field,” in *Proc. of IEEE Int’l Conf. of Mechatronics and Automation*, vol. 4, pp. 1711–1716, 2005.

[66] K. Sugihara and J. Yuh, “GA-based motion planning for underwater robotic vehicles,” in *Int’l Symposium on Unmanned Untethered Submersible Technology*, pp. 406–415, 1997.

[67] R. Fox, A. Garcia, and M. Nelson, "A three-dimensional path planning algorithm for autonomous vehicles," in *Proc. of Int'l Symposium on Unmaned Untethered Submersible Technology*, pp. 546–553, University of New Hampshire - Marine Systems, 1999.

[68] A. Alvarez, A. Caiti, and R. Onken, "Evolutionary path planning for autonomous underwater vehicles in a variable ocean," *IEEE Journal of Oceanic Engineering*, vol. 29, no. 2, pp. 418–429, 2004.

[69] D. Rathbun, S. Kragelund, A. Pongpunwattana, and B. Capozzi, "An evolution based path planning algorithm for autonomous motion of a UAV through uncertain environments," in *21st Proc. of IEEE Conf. of Digital Avionics Systems*, vol. 2, pp. 8D2–1–8D2–12, 2002.

[70] R. A. Howard, "Information value theory," *Systems Science and Cybernetics, IEEE Transactions on*, vol. 2, no. 1, pp. 22–26, 1966.