Scheduling Multiple Mobile Sinks in Underwater Sensor Networks

Fahad Ahmad Khan, Saad Ahmad Khan, Damla Turgut and Ladislau Bölöni

Department of Electrical Engineering and Computer Science
University of Central Florida, Orlando, FL
Email: fahad.khan@ucf.edu, {skhan, turgut, lboloni}@eecs.ucf.edu

***Abstract*—Underwater Sensor Networks (UWSNs) provide valuable data for research studies and underwater monitoring and protection. UWSNs need to overcome the handicap that high data rate wireless transmissions are not available underwater. Acoustic communications are used as a medium but they are only good for transmitting e.g. signalling information. Autonomous Underwater Vehicles (AUVs) can serve as mobile sinks that gather and deliver larger amounts of data from the underwater sensor network nodes. Value of Information (VoI) is a data tag that encodes the importance and time-based-relevance of a data chunk residing at a sensor node. VoI, therefore, can serve as a heuristic for path planning and prioritizing data retrieval from nodes. The novelty of this paper lies in providing algorithms which schedule multiple mobile sinks (AUVs) for data retrieval from nodes while maximizing the retrieved VoI. The class of algorithms discussed are based on greedy heuristics.**

I. Introduction

Time-critical data delivery is crucial for active ocean monitoring of oil plumes. For example, the famous event of *Deepwater Horizon* oil spill was tracked after the large oil slick was visible at a former rig site [9]. The oil plume may have different direction as compared to the oil slick on the surface. This is because the movement of plume can take any direction due to the differential between the surface and the water currents below it. Thus it is necessary to have an active ocean monitoring on the rig sites that can provide the precise sensing of the information of the pollutants at an early stage.

A practical approach for the ocean monitoring is the use of underwater acoustic sensor network - a network of underwater sensor nodes that either transmit data on multihop routes or use a data-mule (autonomous underwater vehicle) for the collection of sensed data via using acoustic communication [3], [6]. Further, it is impractical to increase the spatial resolution of the sensor network as is not cost effective to sufficiently cover large sites with hundreds of these sensors [13], [8].

The use of autonomous underwater vehicles increases the latency of the data delivery but helps significantly in the reduction of the energy consumption of nodes that would otherwise use the multihop forwarding approach for data-forwarding. To reduce the latency of the data-delivery optimization we must consider resources such as speed of the AUV, its payload capacity and the path traversed for the data collection. An optimal path would collect the maximum amount of information in limited time while using minimum amount of fuel and other resources.

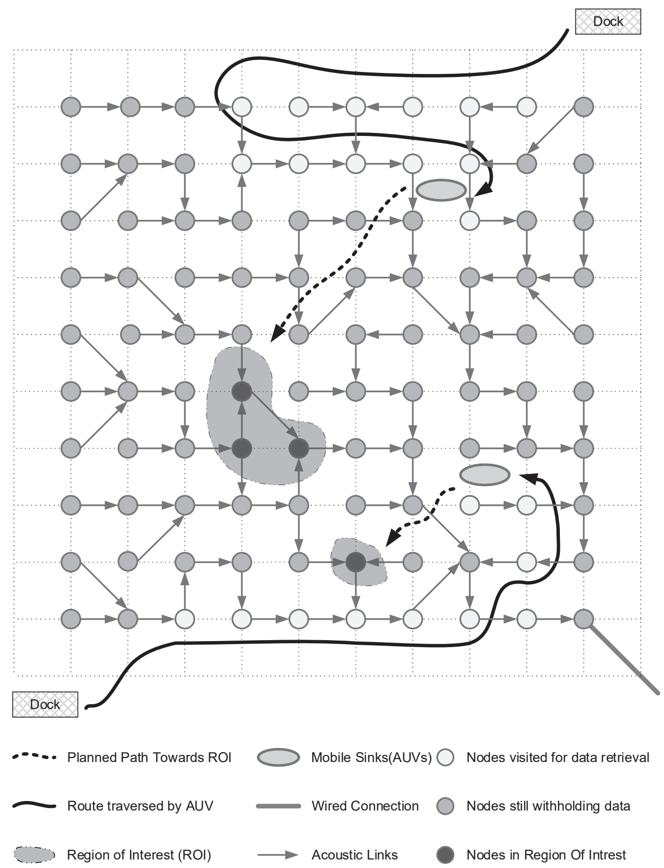


Fig. 1. Multiple AUVs collecting information from the acoustic sensor network

The use of multiple mini-sized AUVs has certain advantages as compared to the use of a single large AUV. Using multiple AUVs helps in the scalability of the area coverage and also provides the advantage for the fault-tolerance ability of the acoustic sensor network - an AUV can reschedule and complete the tasks for another AUV that has malfunctioned. Further, with the use of appropriate scheduling algorithms, one can use the heterogeneous abilities of different AUVs as an advantage. For example, a high speed AUV can be used to collect the information from the regions with time-sensitive data whereas the low speed AUV can be used for the collection of non-sensitive data from the rest of the sensor nodes. However, the data monitoring missions near the coastal or reef areas are difficult as there are strong temporal tidal currents and as we go further deep into the ocean environment, these currents have greater impact on the trajectories of the AUVs. In such cases the path of a straight line from the dock to the sensor node may not be the best path due to tidal current affecting the AUV motion resistance [16], [11], [12].

For our work, we consider multiple AUVs, as shown in Figure 1, that are equipped with acoustic communication devices and can traverse throughout the sensing area. The AUV can resurface at any point during collection of data and can communicate using wireless medium to transfer the accumulated data up till that instant to a base-station. Once it transmits all the data it returns to resume offloading data from the underwater sensor nodes. The path planning problem for multiple AUVs is to schedule paths for data retrieval that minimizes the cost of operation and increases the value of information. The traversal cost for the AUVs depends on the time required to move from one sensor node to the other sensor node as the value of information depends decreases with time.

Our path planning problem is closely related to the travelling salesman problem (TSP) with "prizes". This variant of TSP has to maximize the total collected prize associate with each visited point while we return to the starting point within a give time frame. This problem is known to be NP-hard and a number of related variants [5] have been proposed depending upon the constraints on the path. For our work, the *Value of Information (VoI)* is a monotonically decreasing function in time. Each data chunk has a VoI function associated with it, which describes the initial VoI of the data as well as the way in which it decreases in time. The key difference between our problem and regular "TSP with prizes" is:

- The prize value is uncertain due to unreliable channel of the acoustic model. The value of the information gathered from different sensors uses a probabilistic model - weak acoustic channel communication can provide a different value of information.
- Empirical evidence is required to model the posterior value of information at the sensor nodes which is dependent upon on the precise value of information of the neighbouring nodes.
- The information gathered from various locations is associated with the decay of the value of information.
- During the path traversal, a sensor node can be skipped with an additional cost penalty.

We are considering homogeneous AUVs - each AUV has the same construction, capabilities and resources such as fuel and sonar sensor. Thus, all of the AUVs have the ability to move to any sensor location to collect the data chunks of information. Scheduling and balancing multiple AUVs introduces challenges for the path scheduling with the goal of maximum information collection - not only should the planned paths maximize the amount of information but should be balanced for planning swift collection and delivery of the gathered data. The task of maximizing the value of information can be broken down as: determining the path with maximum information, scheduling the path traversal with goal of collaborative task assignment and reducing the data delivery delay latency.

Our contributions are:

- Formulation of greedy heuristics for the path planning problem using multiple AUVs
- Modelling of algorithms which can be used for both single AUV and multiple AUVs
- Modelling scenarios that are affected by the monotonically decreasing value of information and discussing the importance of greedy algorithms for those scenarios
- Demonstrating the effectiveness of different greedy algorithms by extensive simulation based evaluation of the proposed algorithms

The rest of the paper is organized as follows. In Section III, we introduce the concept of the value of information with the graphical explanation provided in Figure 2a. In Section III, we also discuss the problem of maximizing *value of information* using multiple AUVs. In Section IV, we discuss the propositions and the heuristics for modelling the *greedy path planners* for maximizing the *VoI* and finally we discuss our results in Section V.

II. Related Work

Turgut and Bölöni [14] introduced an information quality metric on the basis of which a methodology was devised to retrieve data in a particular priority from different types of sensor networks. The information quality metric was VoI which essentially makes the user aware of the degradation/decay of the quality of information of data (pertaining to a set of sensed stimuli) with respect to time (since the time the stimuli were first recorded to the present instant). The VoI has been initially applied to intruder tracking sensor networks [14], [15]. The transmission scheduling of sensor nodes via acoustic links to the sink as well as the path planning of an AUV to collect data from sensor nodes have been also explored [4], [1]. In either of the application domain, the goal was to maximize the quality of the gathered data using AUVs [7].

For cooperative path planning, the approaches used to solve the problem can be divided into two categories - the centralized approach and non-centralized approach [2]. For a centralized approach the search space for path planning for multiple AUVs is combined into one large composite configuration space and the solution is searched using the composite system. But the solution using centralized approach has a higher time complexity growing exponentially in the dimension of the composite space. For a non-centralized approach, the path planning of multiple AUVs is handled separately and then any possible conflicts are resolved in the cooperative planned paths. Thus, for real world scenarios we have to use heuristics for the exploration of the increasing search space of the composite system.

III. Value of Information (VoI)

We define the VoI as a function that assigns to a data packet a value that is monotonically decreasing in time. These functions can be designed in a variety of different ways so as to suit one's narrative of an application scenario. The functions have two types of parameters as shown in Figure 2a. The parameters are A_x and B_y. The parameter A_x assigns a value to the 'initial importance' of the information associated with a data packet while the parameter B_y controls the 'decrease of relevance' of information with time. Figure 2a shows three different types of monotonically decreasing VoI functions ($f_{VoI}(t)$) ; Step, Ramp and Exponential. The Step function uses parameters B_1 and B_2 to control the decrease in step size for the VoI while the Ramp function uses B_4 to control the slope B_3 for a continuously degrading VoI. In both of these functions the $f_{VoI}(t)$ value ends up at zero value ($f_{VoI}(t)$ can be designed to end up at any other constant value) i.e. after a particular length of time each data packet will have the same VoI value, thus any information of time precedence i.e. when the packet was acquired, will be lost. This way, the VoI value can also be used to detect which old packets can be discarded. But if the goal is to always retain the information retain the information precedence then Exponential functions serve the purpose better. In exponential functions A_x controls the initial value of the information while B_y control the decay in the VoI. The two exponential functions could be thought of being applied to an application scenario as follows. The bottom left exponential in Figure 2a can be understood to be assigned to a data packet that has a higher importance (e.g. reporting a hazard or abnormal activity) as it has a larger initial value. In contrast the bottom right exponential can be assigned to data pertaining to normal events as it has a lower initial value. Moreover, the exponential assigned to the more important events has a faster decay implying that their worth will become irrelevant soon if not retrieved quickly by the end user while a slower decay implies data still being relevant after longer periods of time. If it is required that the more important data always has a more VoI than the normal data then it could be designed by setting $A_2 > A_3$ and $B_5 > B_6$.

For a single AUV, planning the path is to find a priority path for visiting the nodes for data collection for maximizing the data collection. For each chunk of data, the sensing nodes attribute a value, decaying in time. The more urgent the data, the faster the decay of the value of information.

The case of cooperative planning for multiple AUVs is a difficult problem and the spatial and temporal domain overlap for the multiple AUVs. For the spatial problem, we require that all AUVs collect the data from distinct nodes - the nodes that are not visited by other AUVs. For the temporal overlap, we require the time to be scheduled to maximize the aggregate value of information collected by all AUVs. Thus, we not only want to schedule the path of the individual AUVs but we should also ensure the fair job division between the AUVs - each AUV should visit the same number of nodes or at least spend a similar amount of time traversing the sensor network region. Also, the AUVs should plan a path that would cover the maximum area of the sensor node deployment while decreasing the cost of traversal between the sensor nodes.

IV. Greedy Heuristics

In these section we propose a series of greedy heuristics for the path planning of AUVs. These heuristics are formulated around a series of propositions which we abbreviate as P_{NM}, P_{MT}, P_{NMV}, P_{LB}, P_{VPA} & P_{MP}. Propositions P_{LB}, P_{VPA} & P_{MP} specifically address the scheduling of multiple AUVs while propositions P_{NM}, P_{MT} & P_{NMV} are more foundational and generic. The intuition behind P_{NM}, P_{MT} & P_{NMV} are explained through scenarios shown in Figure 2.

A. The Propositions

*1) **Next Node Visit is based on Maximum VoI(P_{NM}):*** This proposition suggests that the AUV (mobile sink) should visit nodes in a prioritized sequence based upon the amount of VoI at offer from the sensor nodes. Consider the scenario in shown in Figure 2b. Let the value of information at sensor node *x* be $f_{VoIx}(t)$ and the value of information at sensor node *y* be $f_{VoIy}(t)$

$$f_{VoIx}(t) = A_x e^{-B_x t}$$
$$f_{VoIy}(t) = A_y e^{-B_y t}$$

Then,

$$VoI(Path_{mxy}) = f_{VoIx}(t_{mx}) + f_{VoIy}(t_{mx} + t_{xy})$$
$$VoI(Path_{myx}) = f_{VoIx}(t_{my}) + f_{VoIy}(t_{my} + t_{yx})$$

Scenario A Let the constraints be:

$$A_x > A_y \text{ and } B_x = B_y = B$$
$$t_{mx} = t_{my} = t_1 > 0,$$
$$t_{xy} = t_{yx} = t_2 > 0$$

Then we hypothesize,

$$VoI_{S1A}(Path_{mxy}) > VoI_{S1A}(Path_{myx}) \tag{1}$$

Substituting values,

$$A_x e^{-B(t_{mx})} + A_y e^{-B(t_{mx}+t_{xy})} \geq A_x e^{-B(t_{my}+t_{yx})} + A_y e^{-B(t_{my})}$$

Implies,

$$A_x e^{-B(t_1)} + A_y e^{-B(t_1+t_2)} \geq A_x e^{-B(t_1+t_2)} + A_y e^{-B(t_1)}$$

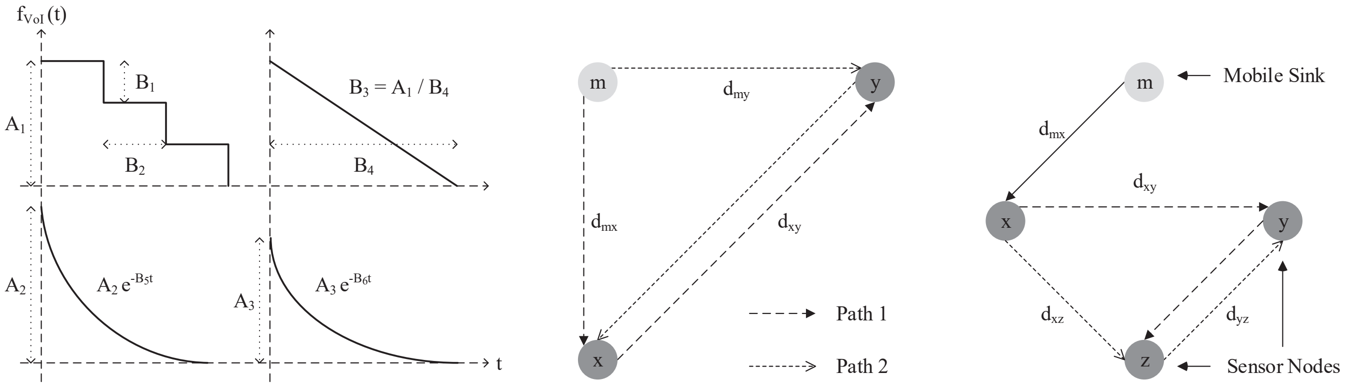
Simplifying,

$$e^{-B(t_1)} > e^{-B(t_1+t_2)}$$

We know that $t_1 > 0$ & $t_2 > 0$, therefore, the hypothesis holds true.

Scenario B Let the constraints be,

$$B_x > B_y \text{ and } A_x = A_y = A$$
$$t_{mx} = t_{my} = t_1 > 0,$$
$$t_{xy} = t_{yx} = t_2 > 0$$



(a) Different VoI functions. Note how A_x & B_y control the shapes of the various functions. Top-Left: Descending staircase function, Top-Right: Negative ramp function, Bottom: Two different decaying exponential functions where the function on the left side can be assigned to higher priority events because of its larger initial magnitude and quicker decay rate while the exponential on the right hand can be assigned to lower priority events as it has a comparatively lower initial magnitude and a slower decay rate.

(b) An example scenario of the greedy heuristic G1. Here $d_{xy} > d_{mx} = d_{my}$ while $f_{VoIx}(t) = A_x e^{-B_x t}$ & $f_{VoIy}(t) = A_y e^{-B_y t}$. Also $d_{xy} \propto t_{xy}$, $d_{mx} \propto t_{mx}$, $d_{my} \propto t_{my}$ where d is distance & t is time.

(c) An example scenario for the greedy heuristic G2. Here $d_{xy} > d_{mx} = d_{xz} = d_{yz}$ while $f_{VoIx}(t) = f_{VoIy}(t) = f_{VoIz}(t) = Ae^{-Bt}$. Also $d_{xy} \propto t_{xy}$, $d_{mx} \propto t_{mx}$, $d_{yz} \propto t_{yz}$, $d_{xz} \propto t_{xz}$ where d is distance & t is time.

Fig. 2. In Figure 2b & Figure 2c, *Path 1* and *Path 2* are the two routes that the mobile sink m will use to traverse to retrieve data from the sensor nodes x, y & z.

Then we hypothesize,

$$VoI_{S1B}(Path_{mxy}) > VoI_{S1B}(Path_{myx}) \tag{2}$$

Substituting values,

$$Ae^{-B_x(t_{mx})} + Ae^{-B_y(t_{mx}+t_{xy})} > Ae^{-B_x(t_{my}+t_{yx})} + Ae^{-B_y(t_{my})}$$

Implies,

$$e^{-B_x(t_1)} + e^{-B_y(t_1+t_2)} > e^{-B_x(t_1+t_2)} + e^{-B_y(t_1)}$$

Simplifying,

$$e^{-B_x(t_1)} - e^{-B_x(t_1+t_2)} > e^{-B_y(t_1)} - e^{-B_y(t_1+t_2)}$$

We know that $B_z > B_y$ and contributes to a faster decaying exponential which in this inequality leads to a larger Δ, therefore, the hypothesis holds true.

Scenario C Let the constraints be,

$$\begin{aligned} A_x > A_y \text{ and } B_x > B_y, \\ t_{mx} = t_{my} = t_1 > 0, \\ t_{xy} = t_{yx} = t_2 > 0 \end{aligned}$$

Then we hypothesize,

$$VoI_{S1C}(Path_{mxy}) > VoI_{S1C}(Path_{myx}) \tag{3}$$

As inequality (1) and inequality (2) hold true, therefore, inequality (3) holds true. Hence P_{NM} is a valid proposition under the aforementioned scenarios.

2) ***Minimize Tour Time for Path Traversal(P_{MT}):*** Here the proposal is to minimize tour times by the AUV as it is a metric that should improve VoI. Consider the scenario shown in Figure 2c.

Let,

$$\begin{aligned} f_{VoIx}(t) &= A_x e^{-B_x t} \\ f_{VoIy}(t) &= A_y e^{-B_y t} \\ f_{VoIz}(t) &= A_z e^{-B_z t} \end{aligned}$$

Then,

$$\begin{aligned} VoI(Path_{mxyz}) =& f_{VoIx}(t_{mx}) + f_{VoIy}(t_{mx}+t_{xy}) + \\ & f_{VoIz}(t_{mx}+t_{xy}+t_{yz}) \end{aligned}$$

$$\begin{aligned} VoI(Path_{mxzy}) =& f_{VoIx}(t_{mx}) + f_{VoIz}(t_{mx}+t_{xz}) + \\ & f_{VoIy}(t_{mx}+t_{xz}+t_{zy}) \end{aligned}$$

Scenario D Let the constraints be,

$$\begin{aligned} A_x = A_y = A_z = A, \\ B_x = B_y = B_z = B, \\ t_{mx} = t_{xz} = t_{zy} = t_{yz} = t_1 > 0, \\ t_{xy} = t_2 > t_1 > 0 \end{aligned}$$

Then we hypothesize,

$$VoI_{S2}(Path_{mxzy}) > VoI_{S2}(Path_{mxyz})$$

Substituting values,

$$Ae^{-B(t_1)} + Ae^{-B(2t_1)} + Ae^{-B(3t_1)} > Ae^{-B(t_1)} + Ae^{-B(t_1+t_2)} + Ae^{-B(2t_1+t_2)}$$

Simplifying,

$$e^{-B(2t_1)} > e^{-B(t_1+t_2)}$$

We know that $t_2 > t_1$, therefore, the hypothesis holds true. Hence P_{MT} is a valid proposition under the aforementioned scenarios.

*3) **Proposition P_{NM} with Intermediate Neighbour Visit (P_{NMV}):*** According to this proposition sensor nodes should be visited in-order of the VoI values they offer just as in P_{NM} but while moving from a source to a destination node the mobile sink should retrieve data from nodes that it encounters on its path. The definition of nodes encountered on the path can be determined for example by deciding the next node to visit based upon if it is that immediate neighbour of the current node who is closest to the destination. This will help in minimizing the tour time and information loss by avoiding delayed visits to nodes whose visitation at an earlier point in time would have had been less taxing in terms of information loss and fuel expenditure. Technically, P_{NMV} is a combination P_{NM} followed by P_{MT}.

*4) **Load Balancing in terms of Nodes Visited (P_{LB}):*** The proposition demands allocation of an equal number of nodes to each AUV for their tours. The intuition is that this will improve the chances of minimizing the overall time for retrieving data from all the nodes because the chances of some AUVs collecting data while others sitting idle will be reduced.

*5) **Balanced Distribution of nodes in terms of VoI(P_{VPA}):*** This proposition suggests that the nodes being assigned to AUVs should be in round-robin fashion such that at each iteration an unassigned node with the maximum VoI should be assigned to the next AUV. This implies that no single AUV will be accessing a large number of high priority nodes (nodes with higher VoI). This should ensure that nodes with higher priority will be visited earlier and this according to P_{NM} will improve the overall VoI gathered.

*6) **Partitioning Map on basis of Node Proximity(P_{MP}):*** The goal of this proposition is to reduce the average traveling time of each AUV which should in turn improve the overall VoI gathered according to P_{MT}. This proposition does this by partitioning the map into segments of nodes for the AUVs. Their is a one-to-one mapping of these segments to the AUVs. The partitions have nodes that are physically collocated or have physically proximity i.e. nodes in a partition are reachable to each other another by recursively traversing through their one-hop neighbours. This proposition should reduce the traveling time intuitively because each AUV will now has a lesser amount of area to cover (because of the newly partitioned regions that are smaller in size) and hence a shorter average distance covered will result in a shorter average time.

B. Scheduling Heuristics for Scheduling and Path Planning for Multiple AUVs

We define the path as the list of sensor nodes that the mobile sink will traverse for collecting the value of information. The goal for a scheduling algorithm is to find a planned path (tour) for an AUV to maximize the collected VoI. Therefore, in the case of multiple AUVs, the scheduling algorithm should find a unique schedule for each AUV for collecting the VoI. The unique schedule demands that the scheduling algorithm should be able to find a Hamiltonian subpath for each mobile AUV - the sensor node is only visited once by any of the assigned AUVs. Scheduling a planned path for multiple AUV's has not only the goal of maximizing the VoI using Hamiltonian paths but the AUVs should have a balanced workload. This means that they should balance between them the number of nodes they would visit. Hence, the output of the scheduling algorithm would be a unique set (with an equal number of sensor nodes) in the planned path for each AUV. The algorithms assume a Mesh deployment of nodes and can be conceptually extended to other deployments as well. The algorithms employ the aforementioned propositions in different combinations. Table I provides a listing of these combinations. The columns are the propositions while the rows are the various path planning algorithms.

TABLE I
LISTING OF WHICH ALGORITHMS EMPLOY WHICH PROPOSITIONS

	P_{NM}	P_{MT}	P_{NMV}	P_{LB}	P_{VPA}	P_{MP}
RPP				✓		
ZPP				✓		✓
LPP		✓		✓		✓
GPP	✓			✓		
GIPP			✓	✓ !		
GPP-B	✓			✓	✓	
GIPP-B			✓	✓ !	✓	
GPP-P	✓			✓		✓
GIPP-P			✓	✓ !		✓

In Table I, the exclamation mark '**!**' is a specific remark on the effect of P_{NMV} on P_{LB}. The remark is that *GIPP* uses the same algorithm as *GPP* but has an extra step of Tour Adjustment in the end to enforce proposition P_{NMV}. This Tour Adjustment might disturb the load balancing which earlier had resulted because of employing P_{LB}. Therefore, the P_{LB} for *GIPP* is in a weak form of implementation here. The same discussion holds for *GPP-B* and *GPP-P*. These algorithms are stated as Algorithm 4, 5 and 6 respectively and the last three lines for each state the optional Tour Adjustment pseudo-code segment.

We discuss the heuristics in the forthcoming subsections. In the pseudo-codes for all the heuristics we use the notations given in Table II.

1) Random Path Planner - RPP: For a baseline comparison we use a Random Path Planner. It creates a tour by randomly assigning nodes to the tours. However, it ensures that the tours are balanced. The algorithm for the *RPP* is given in Algorithm 1.

2) Lawn-Mower Path Planner - LPP: The Lawn-Mower Path Planner is based on the propositions P_{MT}, P_{LB} & P_{MP}. It is a deterministic planner. This planner has specifically been designed for Mesh deployment of nodes. It plans a traversal

```
procedure RPP(N, A)
    count ← ⌈N.size/A.size⌉
    for all a ∈ A do
        T_a ← φ
        for j ← 1, count do
            if N.isNotEmpty then
                n ← N.getRandom()
                T_a ←_add n
                N.remove(n)
```

Algorithm 1: Random Assignment

```
procedure LPP(N, A)
    Order(N, nun)
    count ← ⌈N.size/A.size⌉
    for all a ∈ A do
        T_a ← φ
        for j ← 1, count do
            if N.isNotEmpty then
                n ← N.getFirst()
                T_a ←_add n
                N.remove(n)
```

Algorithm 2: Lawn-Mower

```
procedure ZPP(N, A)
    Order(N, fun)
    count ← ⌈N.size/A.size⌉
    for all a ∈ A do
        T_a ← φ
        for j ← 1, count do
            if N.isNotEmpty then
                n ← N.getFirst()
                T_a ←_add n
                N.remove(n)
```

Algorithm 3: Zig-Zag

```
procedure GPP(N, A, adjust)
    SortDescending(N, VoI)
    count ← ⌈N.size/A.size⌉
    for all a ∈ A do
        T_a ← φ
        for j ← 1, count do
            if N.isNotEmpty then
                T_a ←_add N.getFirst()
                N.removeFirst()
    if adjust.isTrue then
        for all a ∈ A do
            T_a.adjustTour()
```

Algorithm 4: Naive Greedy

```
procedure GPP-B(N, A, adjust)
    SortDescending(N, VoI)
    for all a ∈ A do
        T_a ← φ
    for all n ∈ N do
        for all a ∈ A do
            if N.isNotEmpty then
                T_a ←_add N.getFirst()
                N.removeFirst()
    if adjust.isTrue then
        for all a ∈ A do
            T_a.adjustTour()
```

Algorithm 5: VoI Priority Greedy

```
procedure GPP-P(N, A, adjust)
    Order(N, run)
    count ← ⌈N.size/A.size⌉
    for all a ∈ A do
        T_a ← φ
        for j ← 1, count do
            if N.isNotEmpty then
                T_a ←_add N.getFirst()
                N.removeFirst()
        Sort(T_a, VoI)
    if adjust.isTrue then
        for all a ∈ A do
            T_a.adjustTour()
```

Algorithm 6: Partitioned Greedy

TABLE II
HEURISTIC NOTATIONS USED FOR THE PSEUDO-CODES

N	$\Rightarrow$	Collection of Nodes
	$\Rightarrow$	$\{n_1, n_2, ...n_k\}$
A	$\Rightarrow$	Collection of AUVs
	$\Rightarrow$	$\{a_1, a_2, ...a_k\}$
n	$\Rightarrow$	Member of Collection N
	$\Rightarrow$	$\{n \mid n \in N \wedge \{n\} \cap \{N - \{n\}\} = \phi\}$
a	$\Rightarrow$	Member of Collection A
	$\Rightarrow$	$\{a \mid a \in A \wedge \{a\} \cap \{A - \{a\}\} = \phi\}$
$count$	$\Rightarrow$	Number of Nodes in each Tour
	$\Rightarrow$	$\mid N \mid / \mid A \mid$
T_a	$\Rightarrow$	Tour for each AUV
	$\Rightarrow$	$\{a \mid a \in A\} \wedge T_a \subseteq N$
$adjust$	$\Rightarrow$	Boolean to determine Tour Adjustment
	$\Rightarrow$	If True then enforce P_{NMV}

of the mesh row-by-row from one end to another. Once it reaches the end of a row, it starts traversing the immediate next row from the node which is the immediate neighbour of the last node that the AUV just visited. This neighbouring node is the closest non-visited node. The Lawn-Mower Path Planner is essentially a form Shortest-Path algorithm for the Mesh deployment. i.e. it will traverse through all the nodes the quickest. For multiple Mobile Sinks the Mesh rows in the Map are divided among the AUVs. The algorithm also takes care of load balancing. The algorithm for the *LPP* is given in Algorithm 2. In $Order(N, nun)$ in Algorithm 2 'nun' implies that the next node to visited after the end of a row would be the nearest unvisited neighbour.

3) Zig-Zag Path Planner - ZPP: This planner, like *LPP*, is deterministic too but it will also act as a baseline for comparison with *LPP*. It also plans a traversal of the mesh row-by-row from one end to another. However, once it reaches the end of a row, unlike LPP, it starts traversing the immediate next row from the node that is farthest from the last node the AUV just visited. Algorithm 3 gives the pseudo-code. It is similar Algorithm 2 except that it uses a different order for the initial list of nodes. In $Order(N, fun)$ in Algorithm 3 'fun' implies that the next node to visited after the end of a row would be the farthest node in the neighbouring unvisited row.

4) Naive Greedy Path Planner - GPP & GIPP): This path planner will act as a baseline for comparisons with non-deterministic path planners based on greedy heuristics. It is based on the proposition P_{NM}. This planner sorts nodes in descending order of their VoI. Afterwards, it assigns them in batches of k nodes (where, $k = \mid N \mid / \mid A \mid$) to each AUV turn by turn. The first batch (and therefore corresponding tour) has nodes with the highest values of VoI while the last batch has nodes with the lowest values of VoI. The planner automatically takes care of load balancing. After assignment, the tours can be adjusted if the requirement is to visit intermediate nodes (named as *GIPP*) in accordance with proposition P_{NMV}. Algorithm 4 lists the scheduling procedure.

5) VoI Balanced Greedy Path Planner - GPP-B & GIPP-B): This path planner is based on proposition P_{NM}, P_{LB} & P_{BD} and also may include the heuristic P_{NMV} (in case of *GIPP-B*). For this scheduler see Algorithm 5

6) Partitioned Map Greedy Path Planner - (GPP-P & GIPP-P): This path planner first employs the P_{MP} proposition to partition the Map. Afterwards it employs P_{NM} to maximize VoI. P_{LB} and P_{BD} are irrelevant here as there is no cross sharing of nodes in the partitions. As in the earlier greedy algorithms *GPP-P* may use P_{NMV} and in such a scenario will be termed as *GIPP-P*. Algorithm 6 states the procedure for this scheduler. In $Order(N, run)$ in Algorithm 6 'run' implies that the partition should be done so that each node in the partition is reachable by a recursive traversal of immediate neighbour hopping.

V. Experimental results

We consider an underwater sensor network of 100 nodes, deployed in a uniform grid over a 10 x 10 km^2 area as shown in Figure 1. These sensor nodes have an acoustic communication medium that allows them to send the data to the AUV. We consider the AUV, a Katrina boat [10] that can speed up to 10 knots in rough water. The travel time of the AUV varies between two sensor nodes with the speed of wind gushes and roughness of water surface. For our experiments, we ignore these two environmental factors and consider a calm water profile.

We evaluate the effectiveness of the proposed greedy algorithms by varying the three important characteristics of the hotspot - its spatial information and its time-critical information and the number of hotspots in the region of interest. For the simulation set-up, during each iteration, we randomly assign to each sensor node the value of the information and the number of packets for that information. To infer the results for each scenario, we run 150 iterations for each algorithm and analyse the results for those scenarios. Each of the AUVs is located near the boundary of the mesh deployment and they are deployed equidistant from each other. We are considering four hotspot with variable locations and the decay time of information at the hotspots is different but the VoI has the same magnitude.

A. Studying the effect of the spatial characteristics of the hotspots

For this set of experiments, we consider four different deployments of the hotspots:

- *The Zero deployment* - this is the normal operating scenario with no active hotspot.
- *The Random deployment* - in this scenario, we randomly generate different locations of hotspots.
- *The Average deployment* - this scenario spatially divides the sensor network region into equal sized sub-regions and initializes the hotspots in those regions at equidistant.
- *The Skewed deployment* - in this scenario, we randomly select a corner of the deployed sensor network and initialize the hotspots for that corner

From the results in Figure 3, we can see that in the "zero hotspot" scenario, *LPP* performs the best because in this scenario the use of *LPP* minimizes the amount of traversal time. For the rest of the scenarios, where the hotspots might be located at a random position or might be skewed at the corner of the sensor network or are location at equidistant, the greedy algorithms specially *GIPP-P* performs the best. Comparing the results of Figure 3 and Figure 4, we see that the proposed algorithms balance the path traversal load between them. As the number of AUVs increase, they show similar a trend for maximizing the VoI irrespective of the increase in the number of AUVs.

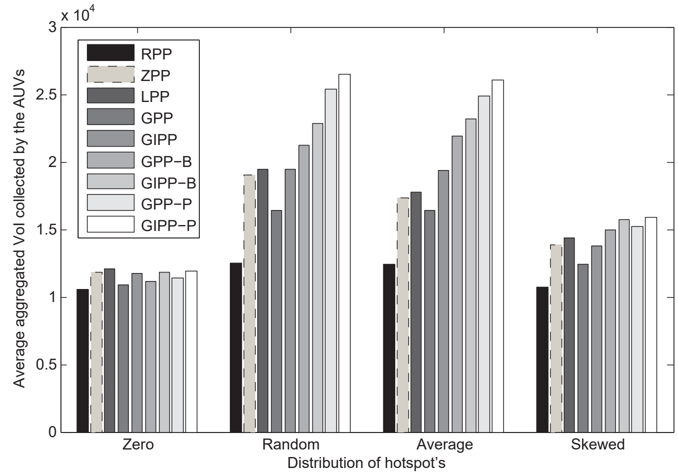


Fig. 3. The average aggregated VoI by a path planner with two AUVs in the deployed region

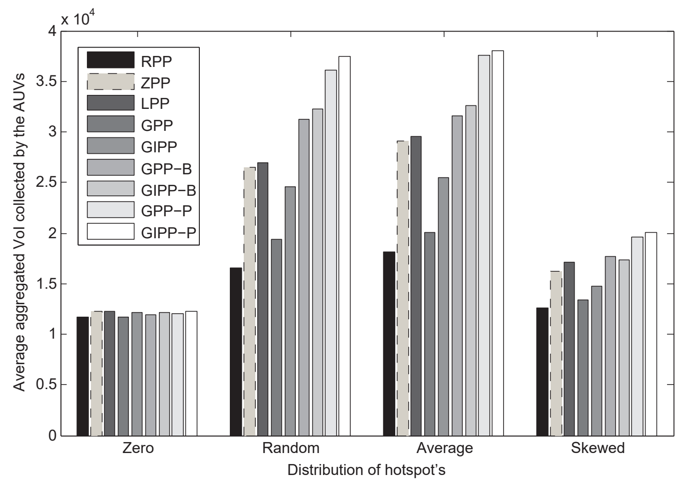


Fig. 4. The average aggregated VoI by a path planner with five AUVs in the deployed region

B. Studying the effect of the increase in the number of hotspots

To study the effect of the balanced resource consumption by the AUVs, we consider the average aggregated distance travelled by the AUVs using the proposed algorithms. For this experiment, we vary the number of AUVs from one to ten. The two variations of the hotspots considered for this experiment is that the sensor network would have a non-active hotspot or will locate an active hotspot.

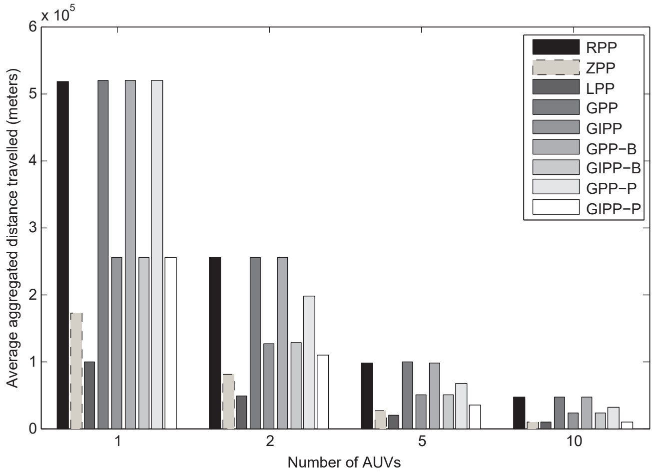


Fig. 5. The distance travelled during path traversal by the AUV using different the path planners

From results in Figure 5, we see that *LPP* and *ZPP* take the least amount of fuel to cover the entire sensor network. This is because of the inherent property of these two algorithms for moving towards the next nearest neighbour, irrespective of the hotspot location. These two algorithms consume the minimal amount of fuel and are the best algorithms for sensor network with no hotspots. But in the case of increasing number of hotspots, the greedy algorithm *GIPP-P* is fuel effective and also maximizes the time-critical VoI.

VI. Conclusion

Scheduling and balancing the use of multiple AUVs introduces challenges for the path planning algorithms as they should be able to balance the task for data collection and should also maximum the collected information. We used a heuristic called Value-of-Information (VoI) as the metric for the path planning. The VoI is a strictly monotonically decreasing function associated with each data bundle and serves as a marker for decay of the quality of information with respect to time of that data bundle. In this paper, we formulated greedy heuristics for the path planning problem with multiple AUVs. We proposed a number of greedy algorithms based on these heuristic which can be used for both single AUV and multiple AUVs. Through simulations, we demonstrated the effectiveness of different greedy heuristics and proposed algorithms by balancing and scheduling the collection of VoI using multiple AUVs.

References

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

[2] M. Bennewitz, W. Burgard, and S. Thrun, "Optimizing schedules for prioritized path planning of multi-robot systems," in *Proc. of IEEE Int'l Conf. on Robotics and Automation (ICRA)*, vol. 1, 2001, pp. 271–276.

[3] N. W. Bergmann, J. Juergens, L. Hou, Y. Wang, and J. Trevathan, "Wireless underwater power and data transfer," in *Proc. of IEEE SenseApp'13*, 2013, pp. 104–107.

[4] L. Bölöni, D. Turgut, S. Basagni, and C. Petrioli, "Scheduling data transmissions of underwater sensor nodes for maximizing value of information," in *Proc. of IEEE GLOBECOM'13*, 2013, pp. 460–465.

[5] G. A. Hollinger, S. Choudhary, P. Qarabaqi, C. Murphy, U. Mitra, G. 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.

[6] T. Hu and Y. Fei, "Dsh-mac: Medium access control based on decoupled and suppressed handshaking for long-delay underwater acoustic sensor networks," in *Proc. of IEEE LCN'13*, 2013, pp. 523–531.

[7] 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 P2MNet'14*, 2014, pp. 610–615.

[8] S. A. Khan, L. Bölöni, and D. Turgut, "Bridge protection algorithms–a technique for fault-tolerance in sensor networks," *Ad Hoc Networks*, vol. 24, pp. 186–199, 2015.

[9] M. K. McNutt, R. Camilli, T. J. Crone, G. D. Guthrie, P. A. Hsieh, T. B. Ryerson, O. Savas, and F. Shaffer, "Review of flow rate estimates of the deepwater horizon oil spill," *Proc. of the National Academy of Sciences*, vol. 109, no. 50, pp. 20 260–20 267, 2012.

[10] MITSG, "Katrina boat: A diesel electric vertical profiling autonomous surface craft." MIT Sea Grant AUV Lab, 2010.

[11] R. Rahmatizadeh, S. A. Khan, L. Bölöni, and D. Turgut, "Routing towards a mobile sink using virtual coordinates in a wireless sensor network," in *Proc. of IEEE ICC'14*, June 2014, pp. 12–17.

[12] ——, "Circular update directional virtual coordinate routing protocol in sensor networks," in *Proc. of IEEE GLOBECOM'15*, December 2015.

[13] F. Senel, K. Akkaya, and T. Yilmaz, "Autonomous deployment of sensors for maximized coverage and guaranteed connectivity in underwater acoustic sensor networks," in *Proc. of IEEE LCN'13*, 2013, pp. 211–218.

[14] D. Turgut and L. Bölöni, "A pragmatic value-of-information approach for intruder tracking sensor networks," in *Proc. of the IEEE ICC'12*, June 2012, pp. 4931–4936.

[15] ——, "IVE: Improving the value of information in energy-constrained intruder tracking sensor networks," in *Proc. of IEEE ICC'13*, 2013, pp. 6360–6364.

[16] J. Witt and M. Dunbabin, "Go with the flow: Optimal AUV path planning in coastal environments," in *Australian Conf. on Robotics and Automation*, no. 2, 2008.