Modeling and Optimization of Docking Stations and AUVs for Ice Floe Measurement

Fang Lu
The University of Texas at Austin
Austin, Texas, USA
lufang05@mail.utexas.edu
Zhichao Shu, John Hasenbein, David Morton,
The University of Texas at Austin
Austin, Texas, USA
shuzhichao@gmail.com, jhas@mail.utexas.edu, morton@mail.utexas.edu
Dominic Berta, Khalid Soofi
ConocoPhilips Company
Houston, Texas, USA
khalid.a.soofi@conocophillips.com, Dominic.P.Berta@conocophillips.com

ABSTRACT

We consider an Arctic drilling platform, which must withstand seasonal ice floes and be protected against extreme ice features. Adequate ice floe thickness estimates are not discernible from satellite images, so autonomous underwater vehicles (AUVs) equipped with sonar scan the thickness of the floes. We approximate this system by an M/G/k+G queue and develop a high fidelity simulation model. We develop and solve a stochastic facility location model, which addresses the design and operational decisions related to the system. Finally, we create a visualization tool to graphically represent the docking station configurations, ice floe arrivals, and the AUV scanning process. These simulations were part of a desktop study and have not been validated through field trials, which we hope to accomplish in the next phase of this study.

KEY WORDS: AUVs; M/G/k+G queue; stochastic facility location; simulation; visualization.

INTRODUCTION

Some oil and gas companies are drilling and developing fields in the Arctic Ocean, which has an environment with sea ice called ice floes. These companies must protect their platforms or the structures where the operations and drilling take place. These structures are designed to resist a collision with an ice floe whose thickness is below a threshold. However, for a threatening floe whose thickness exceeds this threshold, it would be necessary to evacuate the personnel and flush all the pipelines, which takes about 72 hours. This 72-hour circle around the platform is called the alert zone.

By using satellite imaging, we can track the ice floes on a near-continuous basis, predict their movement, and measure their areas. However, the thickness of an ice floe cannot be measured satisfactorily at this time through satellite imaging. We can measure the topography of the surface of an ice floe, but we cannot measure the under-water topography of a floe. One possibility is to use autonomous underwater vehicles (AUVs) to carry out this task. To do so, requires the use of docking stations to launch and charge these AUVs, as well as upload the data they collect. Because the ocean water becomes deeper outside the alert zone, it is not economical to place the docking stations there, and so we need to locate the docking stations inside the alert zone.

When an ice floe is ready to be served by an AUV, because the floe exceeds the alert zone, it is not economical to place the docking stations there, and so we need to locate the docking stations inside the alert zone. Ideally, we can measure the thickness of every arriving ice floe and upload it to the docking station before the floe enters the alert zone. The process of an AUV thus serving an ice floe includes the following five steps: (1) the AUV is deployed to meet the ice floe; (2) the AUV uses sonar to measure the thickness profile of the ice floe; (3) the AUV returns to its original docking station; (4) the AUV uploads the data it has measured; and, (5) the AUV’s battery is recharged.

Queueing Approximation and Simulation

Queueing Approximation

Queueing theory is the mathematical study of queues, which arise in diverse areas, including telecommunications, computer science, operations management, manufacturing, and health care; see, for example, Gross and Harris (1998). Our queueing system includes several related processes, among which there are three basic ones: (1) the arrival process, which describes the characteristics of the incoming customers; (2) the service process, which consists of the scheduling policy along with the stochastic process governing the service times; and, (3) the abandonment process, which represents the impatience of
the customers while they are waiting in the queue. We use the standard notation $A/B/k+C$ to describe this queueing system, where $A$ characterizes the arrival process, $B$ characterizes the service process, $k$ is the total number of servers, and $C$ characterizes the abandonment process. When we replace $A$, $B$, or $C$ with an $M$, it means the inter-arrival, service, or abandonment times have an exponential (Markov or memoryless) distribution. If instead we use a $G$, this indicates that these times have a general distribution.

Given the locations of the docking stations, the arrival trajectories of the ice floes, and the number of AUVs at each docking station, we can approximate the ice floe measurement process by an $M/G/k+G$ queueing system, where the ice floes are modeled as customers arriving according to a Poisson process and the AUVs are servers with a general service time distribution, which depends on the mechanics of scanning the ice floes, uploading data, and recharging batteries. Adverse ice floe events are equivalent to abandonments in the queueing system because the customer leaves the system without having been served. Readers are referred to Mandelbaum and Zeltyn (2005) for a summary of queueing systems with abandonments and Whitt (1993) for analytical approximations of $G/G/k$ queues.

We say that the $M/G/k+G$ queueing system approximates the system in which AUVs serve ice floes for multiple reasons. The probability distributions we employ for the arrival, service, and abandonment processes must be estimated. We assume independence of inter-arrival times, and that service times do not depend on the state of the system. That said, one key aspect of the ice floe system that an $M/G/k+G$ queueing approximation neglects, is that the service time for an ice floe depends on how long the floe has waited to be served. We can assess the extent to which, and under what conditions, this compromises the quality of an $M/G/k+G$ approximation by employing a more detailed simulation model. Further advantages of the simulation model we describe next are that it also allows us to explore a richer class of rules for dispatching AUVs to serve floes, and it allows us to analyze the sensitivity of results we infer to underlying system parameters.

### Simulation Model

The queueing model we sketch in the previous sections provides a compact mathematical representation of the dynamics of ice floe movement and the AUV servicing process, which we can use to gain insight or we can embed into larger optimization models. However, a more detailed simulation model, which we describe in this section, allows us to create a higher fidelity representation of the true process.

We model the ice floe arrival process using a spatial Poisson process with arrival rate $\lambda$. We assume, for now, a deterministic and fixed velocity vector for a given ice floe (although each ice floe may have a different vector). Since we only need to investigate the arrival of ice floes that will enter the alert zone, we characterize an arrival trajectory of an ice floe using three parameters, $\theta$, $\alpha$, and $\xi$, as follows:

- $\theta$ generates the pair $(R \sin \theta, R \cos \theta)$ on the circumference of the alert zone circle, where $R$ is the radius of the alert zone. So the parameter $\theta$ determines where the ice floe would intersect the perimeter of the alert zone.
- $\alpha$ determines the direction of the ice floe’s velocity, $\alpha \in [0, \pi]$. The angle $\alpha$ is relative to the tangent of the arrival point of the ice floe on the circumference of the alert zone, as depicted in Figure 1.
- $\xi$ represents the speed of the ice floe.

So, $\theta$ and $\alpha$ uniquely determine the trajectory of an arriving ice floe and $\xi$ is the speed of the ice floe on this trajectory. Note that we can restrict attention to $\alpha$ values in $(0, \pi)$ because we only consider floes that will enter the alert zone. The distribution of $\theta$ and $\alpha$ depends on the characteristics of the ice floes’ movement.

We have used the commercial software package Arena (cf. Kelton et al. 2009) to implement our simulation model and the flow chart of the simulation model is depicted in Figure 2.

![Figure 1: Simulation of an arriving ice floe’s trajectory](image)

**Figure 1: Simulation of an arriving ice floe’s trajectory**

**Figure 2: Flow chart for the simulation model**

### OPTIMIZATION FOR THE SIMPLIFIED DOCKING STATION LOCATION PROBLEM

In this section, we consider a simplified optimization model for the problem of locating docking stations. This approach gives some insight into the placement of candidate docking station locations in a more detailed location model, which is developed in the following section. In this model, we assume that stations are placed uniformly on a circle inside the alert zone. Such a configuration is optimal if the arrival process distribution is radially homogeneous. We wish to analyze the optimal radius of this docking-station circle. Again, denote the radius...
for the alert zone by $R$ and assume we have $N$ stations to place uniformly on the docking-station circle with radius $r$. Then, we seek the value of $\eta$ (ratio between $r$ and $R$), which minimizes the AUV’s expected one-way travel time.

Our simplified model employs the following assumptions: (1) there are ample AUVs at each station, so there is no queueing; (2) an AUV can instantaneously scan an ice floe; (3) every ice floe is assigned to the station with the shortest travel time to that floe; (4) the speed is the same for all ice floes; (5) ice floes arrive according to the spatial Poisson process described in the previous section; and, (6) an AUV is dispatched to an ice floe at the latest feasible time, i.e., so that data regarding the floe’s thickness is uploaded just as the floe reaches the alert zone. Figure 3 shows the relationship between the expected value of the distance of a single AUV trip and $\eta$, when $N=2,3,4$, respectively. For these computations, the radius of the alert zone is assumed to be 36 nmi (nautical mile) by considering that the speed of the ice floes could be as large as 0.5 knots. The ratio of an AUV’s speed to that of the ice floe is set to 16.

So for the simplified station location problem, if the total number of stations is given, we can find the optimal radius for the docking-station circle, which can give us some insight regarding the station locations for the stochastic optimization model in the next section.

Figure 3: Relationship between the AUV’s expected one-way travel time and the ratio of radii

**STOCHASTIC FACILITY LOCATION PROBLEM**

Generally speaking, a facility location problem consists of a set of potential facility sites and a set of spatially located demands that are to be served by the facilities. The goal is to open a set of facilities, accounting for both an operations cost and a cost of system design. For example, subject to a budget constraint limiting facility installation cost, we may seek to locate facilities to minimize a weighted sum of distances from each demand point to its nearest facility. The model is stochastic when, for example, the set of customers requiring service is known only through a probability distribution when we must locate the facilities. Further model embellishments can include a time window during which each customer can be served, and objective functions such as minimizing the expected number of unserved customers. See Snyder (2006) for a review of facility location problems under uncertainty.

In our application of a stochastic facility location model to locating AUV docking stations, we assume we have $n$ predetermined potential locations to place the docking stations. We further assume that we can place at most $m$ stations; i.e., each candidate location has an equal installation cost. Also, at most $n_i$ AUVs can be placed at location $i$ due to a docking station’s limited number of docks. Figure 4 shows an example of the potential locations for stations, which are inside the alert zone. The arrival circle is a user-defined circular region outside of the alert zone. An AUV can only be dispatched to an ice floe when the floe enters the arrival circle. A large arrival circle increases the travel time for the AUVs and a small arrival circle narrows the time window of service for each ice floe. Either extreme may lead to an unsatisfactory number of abandonments. Thus the arrival circle should be chosen judiciously in order to minimize the number of abandonments.

The allocation of the stations and AUVs take into account the uncertainty of the floe arrival dynamics, which we use to generate input scenarios. Our stochastic facility location model is a two-stage stochastic integer program (see, e.g., Schultz et al. 1996) because of the timing of the design decisions, the realizations of randomness, and the operations decisions. Specifically, the docking stations must be located prior to observing the arrival process of the ice floes. However, we assume the decisions governing the dispatching of AUVs to serve ice floes can be made after observing the set of ice floes to be served under a specific arrival-process scenario. The objective function that we minimize is the probability of an adverse ice floe event or equivalently,
the expected number of abandonments.

Figure 4: An example of candidate locations for docking stations

Our optimization model has two kinds of decision variables: (1) the locations of the docking stations and the number of AUVs at each station, and (2) the routing and dispatching policy of the AUVs. These are the output of the optimization model. In our stochastic integer program, we divide time into discrete periods. Models with smaller time units are larger in size but may have higher fidelity. Also, we restrict our attention to ice floes that will eventually enter the alert zone, since we need not scan an ice floe that will not threaten the platform because we assume the velocity vector is fixed for a given ice floe. When an ice floe enters the system, we calculate a time window for each ice floe-docking station pair, and any station can deploy an AUV to scan an ice floe, provided it can do so within the specified time window, and the service process will not exceed the AUV’s battery life. An AUV can first be dispatched to scan an ice floe when the floe enters the arrival circle, and an AUV will not be dispatched if it cannot complete the service process before the ice floe enters the alert zone. An ice floe can be served by at most one AUV, and each AUV needs to return to its original docking station to upload the thickness information and recharge the battery after each scanning sortie.

We need the following primitives to calculate the inputs of the facility location model. For the AUVs, we need to know their speed, battery life, and battery recharge time. For each ice floe, we need to know when and where it arrives, its velocity and the scan time. For the basic problem parameters, we further specify $n$ potential locations to place the docking stations; $m$ the maximum number of stations allowed; and, $n_o$ the maximum number of AUVs, which can be placed at each location $i$. We can generate the required input scenarios either according to the historical data or through simulation, as we describe in the previous section.

In Appendix A, we provide further modeling details and a full mathematical formulation of our stochastic facility location model.

HEURISTIC METHODS

The deterministic facility location problem is NP-hard (Garey et al. 1979), which means it is hard to solve to optimality as the problem size grows large. In addition to uncertainty regarding the floes requiring service, the stochastic facility location problem we present in the previous section involved scheduling dynamics associated with dispatching AUVs, presenting further computational challenges. Next, we provide an example that shows the scale of the problem size.

Suppose we have the following input: (1) the time horizon is one week and the basic time unit is one hour, so $T = 168$; (2) we have 12 potential locations to put the stations; (3) there are 10 scenarios in total and approximately 200 ice floes per scenario; (4) the time window for each ice floe is about 20 hours; and, (5) every ice floe has a list of four stations that are eligible to serve it. This yields a stochastic integer program with 84,040 integer decision variables and 1,366,173 constraints, which will likely challenge current integer programming solvers. So, we seek heuristic algorithms that can solve such a large-scale problem efficiently.

The simplified docking station location problem can give us some insights about the optimal radius of the docking station circle, which is closely related to the first-stage decision in our problem. Given the locations of the docking stations, we then determine the second-stage decision variables, which involve dispatching the AUVs, by heuristic methods.

Minimize the Total Number of Tardy Jobs

In the scheduling literature, tardy jobs refer to those jobs that cannot be finished before their due dates. The total number of tardy jobs is an important performance measure, which has received significant attention in scheduling theory. If we have a set of jobs, which may have different due dates, that are all available at the beginning of the time horizon and there is only one single server to process them, there exists an algorithm that can easily determine the optimal sequence of jobs that minimizes the total number of tardy jobs (Pinedo, 2008, Chapter 3). However, if we assume the jobs become available at different times and there are multiple servers, the problem becomes hard to solve, and we rely on heuristic methods to find a good solution when the problem size is large.

Given the docking station locations, our problem of dispatching AUVs to minimize the number of unserved ice floes has much in common with these scheduling problems. The ice floes correspond to jobs with different arrival times $a_i$, where the arrival time is defined as the time that the ice floe enters the arrival circle. The due date $d_i$ for each ice floe is the latest feasible time that an AUV can serve the floe, which is the arrival time plus the abandonment time. Note that the service time, $p_i$, depends on the waiting time in our problem, which makes the problem “nonstandard” and harder to solve.

Heuristics and Scheduling Policies

We describe three heuristics to determine the dispatch policies for the stochastic facility location problem. We assume the docking station locations, and the number of AUVs at each station, are fixed when we apply these algorithms. We present our computational experience with these methods in the next section.

Algorithm 1

When an ice floe arrives, it is assigned to the station that would require the least effort to serve that floe (the shortest travel time), assuming an AUV is available. If there is no available AUV at the station, then we deploy the AUV to serve the floe; if not, the floe joins the queue for the station. Suppose there are $j$ jobs in the queue, then we order the ice floes in a way such that $d_1 \leq d_2 \leq \ldots \leq d_j$.

Step 0: Define $J = \emptyset$, $J' = \{1, \ldots, f\}$, $J'' = \emptyset$, $k = 1$;

Define $a_i$, $i \in G = \{1, \ldots, g\}$, as the next available time of AUV $i$;

Step 1: Add job $k$ to $J$, delete job $k$ from $J'$, go to step 2;

Step 2: Let $u$ denote the AUV such that $a_u = \min_{i \in G} \{a_i\}$ and schedule job $k$ at that time $s_k = a_u$. If $s_k < d_k$, set $a_u = a_u + p_i$ and go to step 3; Otherwise, delete job $k$ from $J$, add $k$ to $J'$ and reschedule the jobs in set $J$.
Step 3: If \( k = f \), STOP. Otherwise, \( k = k + 1 \), go to step 1.

Algorithm 2

The only difference between Algorithm 1 and Algorithm 2 is in the following step:

Step 2: Let \( u \) denote the AUV such that \( na_u = \min_{i \in G} \{na_i\} \) and schedule job \( k \) at time \( s_k = na_u \). If \( s_k \leq d_k \), set \( na_u = na_u + p_k \) and go to step 3; Otherwise, let \( z \) denote the job such that \( p_z = \max_{i \in J_c} \{p_i\} \), delete job \( z \) from \( J \), add \( z \) to \( J' \) and reschedule the jobs in set \( J' \).

Algorithm 3

When an ice floe arrives, suppose there are \( q \) stations that are eligible to serve it and the corresponding processing times are \( p_j^i \), \( i = 1, \ldots, q \). Order the stations such that \( p_j^i \leq p_j^{i+1} \leq \cdots \leq p_j^q \) and define \( v = 1 \). Suppose we only consider the first \( r \) (\( r < q \)) stations as candidate stations to serve the ice floe.

Step 0: If there is an available AUV at station \( v \), then deploy it to serve the floe; if not, the floe joins the queue for station \( v \). Suppose there are \( f \) jobs in the queue for station \( v \), then order the ice floes such that \( d_1 \leq d_2 \leq \cdots \leq d_f \). Order the stations \( J_v = \{1, \ldots, f\}, J'_v = \emptyset, k = 1 \), as the next available time of AUV \( i \);

Step 1: Define \( J_r = \emptyset, J_v = \{1, \ldots, f\}, J'_v = \emptyset \), \( k = 1 \); Define \( na_n, i \in G = \{1, \ldots, q\} \), as the next available time of AUV \( i \);

Step 2: Add job \( k \) to \( J_v \), delete job \( k \) from \( J_v \), go to step 3;

Step 3: Let \( u \) denote the AUV such that \( na_u = \min_{i \in G} \{na_i\} \) and schedule job \( k \) at time \( s_k = na_u \). If \( s_k \leq d_k \), set \( na_u = na_u + p_k \) and go to step 4; Otherwise, let \( z \) denote the job such that \( p_z = \max_{i \in J_c} \{p_i\} \), delete job \( z \) from \( J_v \), add \( z \) to \( J_v \), and reschedule the jobs in set \( J_v' \);

Step 4: If \( k = f \) and \( v = r \), STOP.

If \( k = f \) and \( v < r \), assign the jobs \( J_v \) to station \( v + 1 \) as lower priority floes, set \( v = v + 1 \) and go to step 0; Otherwise, \( k = k + 1 \), go to step 2.

Unlike Algorithm 1, Algorithm 2 abandons those ice floes with the longest service time in the queue, so Algorithm 2 may be able to serve more ice floes than Algorithm 1. Both Algorithm 1 and 2 assign the ice floes to a single docking station when they arrive, while Algorithm 3 allows collaboration between docking stations. When \( r = 1 \), Algorithm 3 reduces to Algorithm 2. If there are lower priority floes in the queue, the station will not serve lower priority floes until all the other floes receive service without being forced to abandon.

A VISUALIZATION TOOL

To validate the results of our queuing simulation and optimization models, we developed a visualization tool, which has realistic depictions of ice floe trajectories and AUV dispatching. This tool permits different types of simulation investigation, using randomly generated data for ice floe arrivals or historical data, from satellite images or other sources. In the visualization tool, we also implemented the three heuristics from the previous section for dispatching AUVs to serve arriving ice floes.

We need the following information as input for our visualization tool: (1) the locations of the docking stations; (2) the number of AUVs at each docking station; (3) the AUV speed; (4) the velocities, locations and shapes of the ice floes; (5) the time required to scan each ice floe; and, (6) the dispatch policy for the AUVs.

We can obtain historical locations and shapes of ice floes through satellite images, and we assume that ice floes have constant velocity during the time interval associated with two adjacent images.

Otherwise we can randomly generate ice floe arrivals according to a stochastic process, as is currently done in the simulation model, as described above.

In detail, the dispatch policy of the AUVs describes at which time point and from which docking station we should deploy an AUV to serve which ice floe. This can be obtained from the solution to the stochastic optimization model or the heuristic algorithms embedded in the visualization tool.

We have developed a basic version of the visualization tool using the geospatial data abstraction library GDAL/OGR on the C++ platform. Figure 5 shows an output image from the tool. The larger circle is the arrival circle, which has a radius of 60 nmi. We can only start to serve an ice floe after it reaches the arrival circle. The smaller circle is the alert zone with a radius of 35 nmi. In this example the platform is about 12 nmi from the coast and we have 12 candidate locations for docking stations, which are distributed on three concentric circles with radii of 20, 25, and 30 nmi. Using data from satellite images, we depict 23 ice floes moving towards the platform. The speed of the ice floes is up to 0.5 knots and they have small size of about 1 square kilometer.

With respect to the required data listed above, we can extract input (4) through satellite images, and other data such as (3) and (5) are estimates of AUV capabilities. We obtain input (1), (2), and (6) from the solution to the stochastic facility location model, which otherwise has the same input as the visualization tool. In the scenario depicted in Figure 5, we show optimal locations of the three docking stations (the orange squares) derived from the optimization model.

The output file format of the visualization tool is GeoTiff, which can embed georeferencing information within a Tiff file. We can also combine these GeoTiff images to generate video files.

![Image 5](531x244 to 531x414)

**Figure 5:** An image from the visualization tool

COMPUTATIONAL RESULTS

Assume we have six potential sites (Figure 6) to locate the docking stations on a circle with radius 30 nmi. Due to the space limitation, we can locate at most six AUVs at each station. Given the number of stations and AUVs, we can solve the stochastic facility location model to determine the locations of the stations and number of AUVs per station to minimize the total number of abandonments. However, when the problem size is too large, it is hard to solve the problem to optimality, so we resort to heuristic methods.
Historical Data

We use data sampled from April 19 to May 3, 2010. A total of 102 ice floes arrived to the system during this period. The time unit is one hour, so the length of the time horizon is 360 hours. The AUV speed is 8 knots and the speed of the ice floes varies from 0.01 to 0.5 knots, with the majority between 0.1 and 0.5 knots. The time to scan an ice floe, which is a complicated function of the ice floe’s size, its underwater topography, and the density of the ocean water, is between 1 and 8 hours. The AUV battery is recharged for 4 hours after every scanning sortie.

Tables 1 to 3 show the results of the optimization model as we change the number of stations and the total number of AUVs. We can see that the total number of abandonments drops quickly when we increase the number of stations further decreases the number of abandonments. For a fixed number of docking stations, the station index vector is relatively stable when we increase the total number of AUVs.

### Table 1: Optimization results with 1 docking station

<table>
<thead>
<tr>
<th>Total AUV #</th>
<th>Station Index</th>
<th>AUV # per Station</th>
<th>Total # of Abandonments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5</td>
<td>1</td>
<td>74</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>2</td>
<td>53</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>3</td>
<td>36</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>4</td>
<td>22</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>5</td>
<td>10</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>6</td>
<td>0</td>
</tr>
</tbody>
</table>

### Table 2: Optimization results with 2 docking stations

<table>
<thead>
<tr>
<th>Total AUV #</th>
<th>Station Index</th>
<th>AUV # per Station</th>
<th>Total # of Abandonments</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>(1,5)</td>
<td>(1,1)</td>
<td>50</td>
</tr>
<tr>
<td>3</td>
<td>(1,5)</td>
<td>(1,2)</td>
<td>29</td>
</tr>
<tr>
<td>4</td>
<td>(1,5)</td>
<td>(2,2)</td>
<td>13</td>
</tr>
<tr>
<td>5</td>
<td>(1,4)</td>
<td>(2,3)</td>
<td>0</td>
</tr>
</tbody>
</table>

### Table 3: Optimization results with 3 docking stations

<table>
<thead>
<tr>
<th>Total AUV #</th>
<th>Station Index</th>
<th>AUV # per Station</th>
<th>Total # of Abandonments</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>(1,3,5)</td>
<td>(1,1,1)</td>
<td>28</td>
</tr>
<tr>
<td>4</td>
<td>(1,3,5)</td>
<td>(1,1,2)</td>
<td>11</td>
</tr>
<tr>
<td>5</td>
<td>(1,4,5)</td>
<td>(2,1,2)</td>
<td>0</td>
</tr>
</tbody>
</table>

When the locations of the docking stations and the number of AUVs per station are fixed, we can apply the heuristic methods to determine the scheduling rules for the AUVs. Figure 7 shows the optimality gap between the solutions from the heuristic methods and the optimal solution. On average, Algorithm 3 has better performance than Algorithm 2, while Algorithm 2 is better than Algorithm 1. One thing to notice on Algorithm 3 is that increasing the value of r, which is the number of stations that collaborate, does not necessarily improve the performance of the heuristic algorithm.

![Figure 6: Potential locations to put docking stations](image)

![Figure 7: Total number of abandonments using historical data](image)

It takes several hours to solve the optimization model with the modeling language GAMS using CPLEX as the integer programming solver, but the heuristic algorithm only takes a few seconds to solve the problem.

Synthetic Data

Since we have limited historical data, we also generate synthetic data to form larger scale problems in order to obtain more robust solutions and test our algorithms. We use a spatial Poisson process with arrival rate λ to generate the arrival process, as we describe in the previous sections. According to the historical data, we use the estimate λ = 102/360 ≈ 0.284 floes per hour. We assume θ = U(-0.5π, 0.5π) and θ ~ U(0, π). Here, we use U(a,b) to denote a continuous uniform random variable on the interval (a,b). The ice floe scan time is modeled as a discrete uniform random variable on the integers between 1 and 8 hours and the ice floe speed is modeled as having a continuous uniform distribution between 0.1 to 0.5 knots. As above, the battery recharge time is four hours.

We generated 10 scenarios with the time horizon for each scenario being 100 days. The basic time unit is 1 hour, so the total length of the time horizon for each scenario is 2400 hours. We generated 6713 total arrivals in 10 scenarios. Table 9 contains the solutions from the heuristic methods when we have a single station at location 4 in Figure 6. Table 10 shows the performance of the heuristic algorithms when we have 2 docking stations placed at locations 1 and 5. Table 11 shows the heuristic algorithm solutions when we have 3 docking stations placed at locations 1, 3 and 5.

The performance of the heuristics is consistent with our observations in the previous section with historical data. We use the results from Algorithm 1 as a base case. The relative improvement using Algorithm 2 and 3 are given in parenthesis.
Table 9: Total number of abandonments with 1 station

<table>
<thead>
<tr>
<th>AUV # per Station</th>
<th>Algorithm 1</th>
<th>Algorithm 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>3713</td>
<td>3440</td>
</tr>
<tr>
<td></td>
<td>(7.35%)</td>
<td>(11.99%)</td>
</tr>
<tr>
<td>3</td>
<td>2326</td>
<td>2047</td>
</tr>
<tr>
<td></td>
<td>(13.74%)</td>
<td>(13.74%)</td>
</tr>
<tr>
<td>4</td>
<td>1092</td>
<td>942</td>
</tr>
<tr>
<td></td>
<td>(9.72%)</td>
<td>(9.72%)</td>
</tr>
<tr>
<td>5</td>
<td>288</td>
<td>260</td>
</tr>
<tr>
<td></td>
<td>(4.17%)</td>
<td>(4.17%)</td>
</tr>
<tr>
<td>6</td>
<td>48</td>
<td>46</td>
</tr>
<tr>
<td></td>
<td>(4.17%)</td>
<td>(4.17%)</td>
</tr>
</tbody>
</table>

Table 10: Total number of abandonments with 2 stations

<table>
<thead>
<tr>
<th>AUV # per Station</th>
<th>Algorithm 1</th>
<th>Algorithm 2</th>
<th>Algorithm 3 (r=2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1,1)</td>
<td>3447</td>
<td>3229</td>
<td>3202</td>
</tr>
<tr>
<td></td>
<td>(6.32%)</td>
<td>(6.32%)</td>
<td>(7.11%)</td>
</tr>
<tr>
<td>(1,2)</td>
<td>1947</td>
<td>1762</td>
<td>1668</td>
</tr>
<tr>
<td></td>
<td>(9.5%)</td>
<td>(9.5%)</td>
<td>(14.33%)</td>
</tr>
<tr>
<td>(2,1)</td>
<td>2486</td>
<td>2342</td>
<td>1749</td>
</tr>
<tr>
<td></td>
<td>(5.79%)</td>
<td>(5.79%)</td>
<td>(29.65%)</td>
</tr>
<tr>
<td>(2,2)</td>
<td>986</td>
<td>875</td>
<td>579</td>
</tr>
<tr>
<td></td>
<td>(11.26%)</td>
<td>(11.26%)</td>
<td>(41.28%)</td>
</tr>
</tbody>
</table>

Table 11: Total number of abandonments with 3 stations

<table>
<thead>
<tr>
<th>AUV # per Station</th>
<th>Algorithm 1</th>
<th>Algorithm 2</th>
<th>Algorithm 3 (r=2)</th>
<th>Algorithm 3 (r=3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1,1,1)</td>
<td>1852</td>
<td>1709</td>
<td>1363</td>
<td>1226</td>
</tr>
<tr>
<td></td>
<td>(7.72%)</td>
<td>(7.72%)</td>
<td>(26.4%)</td>
<td>(33.8%)</td>
</tr>
<tr>
<td>(2,1,1)</td>
<td>1787</td>
<td>1650</td>
<td>1112</td>
<td>316</td>
</tr>
<tr>
<td></td>
<td>(7.67%)</td>
<td>(7.67%)</td>
<td>(37.77%)</td>
<td>(82.32%)</td>
</tr>
<tr>
<td>(1,2,1)</td>
<td>1013</td>
<td>938</td>
<td>339</td>
<td>256</td>
</tr>
<tr>
<td></td>
<td>(7.4%)</td>
<td>(6.54%)</td>
<td>(66.54%)</td>
<td>(74.73%)</td>
</tr>
<tr>
<td>(1,1,2)</td>
<td>991</td>
<td>916</td>
<td>355</td>
<td>258</td>
</tr>
<tr>
<td></td>
<td>(7.57%)</td>
<td>(64.18%)</td>
<td>(64.18%)</td>
<td>(73.97%)</td>
</tr>
</tbody>
</table>

FUTURE RESEARCH

An important area of future research is to enhance the dynamic model of ice floe behavior. In particular, ice floe velocities are not truly fixed over time. A revised model would assume that the trajectory of each ice floe changes over time in a stochastic manner. However, we might assume that the system manager has a forecast, based on wind and sea conditions, of the floe’s projected future movement. Of course, this enhancement in the ice-floe movement model would necessitate a corresponding improvement in the simulation model. Also, the efficiency and performance of the heuristic algorithms can be further improved and we can relax the assumptions in the simplified docking station location problem to obtain a better approximation. When more historical ice floe data are available, we can use it to fit the parameters for the arrival simulation process to better reflect realistic ice floe dynamics.

REFERENCES


APPENDIX A: Optimization Model Formulation

We define the following sets: (1) $\omega \in \Omega$: set of scenarios; (2) $i \in I$: set of potential locations for docking stations; (3) $j \in J^\omega$: set of ice floes under scenario $\omega$; (4) $i \in I_j^\omega$: set of locations that can serve floe $j$ under scenario $\omega$; (5) $j \in J^\omega$: set of floes that can be served by location $i$ under scenario $\omega$; (6) $t = 0,1,...,T$: time (say, hours).

We calculate the following parameters from the given data on the arriving ice floes. Parameter $d_{i,j}^\omega$ is the round trip travel time from location $i$ to floe $j$ at time $t$ under scenario $\omega$; $s_j^\omega$ is the time needed to scan the thickness of floe $j$ under scenario $\omega$; $R_{ij}^\omega$ is the battery recharge time of an AUV dispatched from station $i$ to serve floe $j$ at time $t$ under scenario $\omega$; $[L_{i,j}^\omega,U_{i,j}^\omega]$ is the time window for location $i$ to dispatch an AUV to scan floe $j$ under scenario $\omega$; and, $p^\omega$ is the probability of scenario $\omega$. We have $m$ stations and $N$ AUVs in total. We can place at most $n_i$ AUVs at each station.

We use binary decision variable, $y_i$, to indicate whether a station is placed at location $i$; $x_{i,0}$ to denote the total number of AUVs placed at location $i$ at time $t$; and, $x_{i,t}^\omega$ to capture the inventory of available AUVs at location $i$ at time $t$, which are the AUVs that are not dispatched or charging. We use binary decision variable $r_{j}^\omega$ to indicate whether floe $j$ is served in scenario $\omega$; $z_{i,j}^\omega = 1$ if an AUV at location $i$ is dispatched to scan floe $j$ at time $t$ and $z_{i,j}^\omega = 0$ otherwise.

In (1), we minimize the expected number of adverse ice floe events. Constraint (2) indicates whether an ice floe has been served by the end of the time horizon. Constraint (3) indicates we can only dispatch an AUV from location $i$ if we have located a docking station at that site. Constraint (4) limits the number of docking stations. Constraint (5) restricts the number of AUVs at each location and constraint (6) limits the total number of AUVs. Constraint (7) keeps track of the inventory of AUVs, using the indicator function on its right-hand side to specify whether an AUV deployed at an early time period is now available.

\[
\min \sum_{\omega \in \Omega} \sum_{j \in J^\omega} (1 - r_{j}^\omega) \quad (1)
\]
\[
\text{s.t.} \quad \sum_{i \in I_j^\omega} z_{i,j}^\omega = r_{j}^\omega \quad \forall j \in J^\omega; \omega \in \Omega \quad (2)
\]
\[
z_{i,j}^\omega \leq y_i \quad \forall j \in J^\omega; i \in I_j^\omega; \omega \in \Omega; t = 1,2,...,T \quad (3)
\]
\[
\sum_{j \in J^\omega} y_i \leq m \quad \forall i \in I \quad (4)
\]
\[
x_{i,0} \leq n_i y_i \quad \forall i \in I \quad (5)
\]
\[
\sum_{i \in I} x_{i,0} \leq N \quad (6)
\]
\[
x_{i,t}^\omega + \sum_{j \in J^\omega} z_{i,j}^\omega = x_{i,t-1}^\omega + \sum_{j \in J^\omega} \sum_{\tau=1}^{T} z_{i,j}^\omega (r_{j}^\omega + d_{i,j}^\omega + s_j^\omega + R_{ij}^\omega = t) \quad \forall i \in I_j^\omega; t = 1,2,...,T; \omega \in \Omega \quad (7)
\]
\[
y_i \in \{0,1\} \quad \forall i \in I 
\]
\[
x_{i,0} \in \mathbb{Z}^+ \quad \forall i \in I 
\]
\[
x_{i,t}^\omega \in \mathbb{Z}^+ \quad \forall i \in I_j^\omega; t = 1,2,...,T; \omega \in \Omega 
\]
\[
z_{i,j}^\omega \in \{0,1\} \quad \forall j \in J^\omega; i \in I_j^\omega; \omega \in \Omega; t = 1,2,...,T 
\]
\[
r_{j}^\omega \in \{0,1\} \quad \forall j \in J^\omega; \omega \in \Omega 
\]