Automated Scheduling and Mission Visualization for Mine Countermeasure Operations

Matthew J. Bays, Richard D. Tatum, Lee Cofer, and James R. Perkins

***Abstract*— We present a novel schedule optimization framework to plan tasks for multiple unmanned surface vessels (USVs) and unmanned underwater vehicles (UUVs) performing a mine countermeasure (MCM) mission. The framework employs mixed-integer linear programming (MILP) to determine efficient schedules. In the framework, the USVs are capable of transporting the UUVs, processing contact information, and neutralizing targets. The UUVs perform survey as well as reacquire and identify (RI) operations. Additionally, we present a novel MCM mission visualizer. The goal of the visualizer is to display in 3D an entire planned MCM scenario for use in scenario evaluation as well as provide a detailed multi-vehicle testing platform for MCM-related vehicle and autonomy research and development.**

I. INTRODUCTION

We are motivated by the trend of using autonomous and unmanned systems within increasingly complex and longer duration maritime operations. One such task is large-scale mine countermeasure (MCM) operations. Future MCM operations could require specialized vehicles to perform distinct phases of an operation: unmanned underwater vehicles (UUVs) with large-coverage, low resolution sonar to detect potential targets, UUVs with low-coverage, high-resolution sensors to identify the objects as mines or non-mines, as well as unmanned surface vessels (USVs) to transport the UUVs and neutralize targets. Coordinating the different sets of vehicles and determining an efficient schedule, as well as quickly and intuitively visualizing the sequence of events for an operation over multiple areas are becoming increasingly important as the vehicles are given ever more complicated and longer-duration tasks. We present a method to model a notional MCM operation as a mixed-integer linear programming (MILP) problem, allowing for its optimization using standard MILP solvers. Currently, large-scale MCM planning is performed by hand, and can therefore contain constraint violations or inefficiencies that can be prevented using formal mathematical optimization techniques such as MILP, thereby increasing the efficiency and optimality of the schedules. We then present a simulation of an optimized MCM operation within a three-dimensional visualization environment.

Previous work in MCM scheduling has typically focused on the task allocation problem [1], [2]. For example, several efforts have been undertaken to determine optimal partitioning of search areas for undersea search with various assumptions on sensor modeling and false alarm information in [3], [4]. Our work extends these results by focusing on the *when* and *how* to optimally have vehicles prosecute different mission areas, as opposed to simply *where* vehicles should be tasked. More recently, work by Molieneaux et al have developed algorithms to aid in schedule repair for previously-developed MCM operation schedules using a cognitive architecture framework [5]. With their framework, the authors attempt to repair user-designed MCM schedules that have become infeasible due to unforeseen circumstances such as asset breakdown or tasking delays. In our work, we attempt to solve the initial step by providing a formal, automated method to plan an overall MCM schedule from a mathematical model of generalized MCM operation constraints.

An equally important thrust for future MCM planning technology is the means by which operators and/or autonomy developers may visualize planned missions for use in plan refinement and evaluation. Most previous work in plan displays has been with developing basic but intuitive tactical decision aids (TDAs) to both plan the mission and display key information such as asset, contact, and environmental information [6]. For autonomy development and visualization, interfaces for the maritime domain have been rather limited, consisting of lightweight, two-dimensional graphical interfaces for visualizing either full paths or small-scale real-time simulations. An example of such lightweight interface is the pMarineViewer interface found in the Mission-Oriented Operating Suite - Interval Programming (MOOS-IvP) autonomy framework [7]. We extend these results with a feature-rich visualization environment that may be used for planning and simulating missions in a full, three-dimensional environment.

This paper is outlined as follows: in Section II, we discuss the general setup of the MCM scheduling problem as a MILP problem, and provide a brief overview of the survey and logistics constraints. In Section II-B, we outline the mathematical constraints that make up the reacquire and identify (RI) component of the notional MCM problem. We then discus the neutralization constraints in II-C. Next, we discuss the mission visualizer interface that takes as input the overall MCM vehicle tasking, and visualizes the full MCM mission. Finally, simulation results showing the performance of various implementations of the MCM scheduling algorithm are discussed, and screen-captures of the MCM simulation are presented in VI.

M. J. Bays, R. D. Tatum, Lee Cofer, and J. R. Perkins are with the Naval Surface Warfare Center, Panama City Division, Panama City, FL 32407 {matthew.bays, richard.d.tatum, lee.cofer, james.r.perkins}@navy.mil

II. Mine Countermeasure Logistics Scheduling using MILP

We will now outline an MCM scheduling problem by developing a formal mathematical framework with which to ultimately optimize a vehicle schedule. Figure 1 shows a notional illustration of the survey UUV transport problem. Let there exist a set autonomous vehicles $\mathcal{A} = \{1, \ldots, A\}$ that are tasked with surveying a number of survey areas $\mathcal{S} = \{1, \ldots, S\}$. The survey areas are connected by a directed graph $\langle \mathcal{D}, \mathcal{V} \rangle$. The nodes $\mathcal{D} = \{1, \ldots, D\}$ represent locations at which a vehicle $a \in \mathcal{A}$ may enter a survey area $s \in \mathcal{S}$ in order to perform operations. The vehicles may move from node i to node j along an edge of the graph $v = (i, j) \in \mathcal{V}$. The subset $\mathcal{D}_s \subset \mathcal{D}$ represents nodes from which vehicles may survey the survey area s.

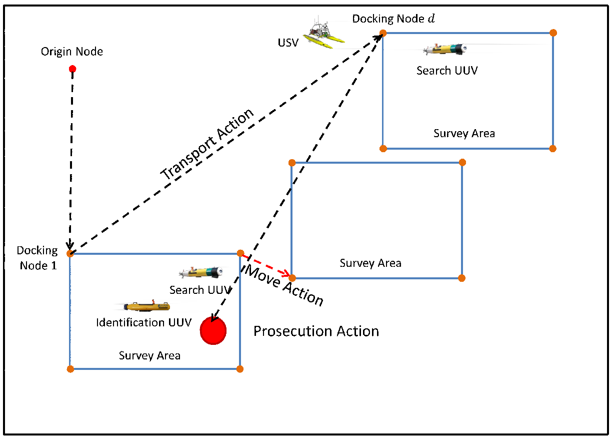


Fig. 1. Notional illustration of an MCM mission utilizing unmanned vehicles, including USVs to transport UUVs.

The vehicles are separated into two mutually exclusive and collectively exhaustive subsets $\mathcal{M}$ and $\mathcal{N}$ consisting of survey UUVs and USVs, respectively. A survey UUV $m \in \mathcal{M}$ is capable of directly surveying the survey areas, while a USV $n \in \mathcal{N}$ may collect a fixed number of the survey UUVs and transport them between nodes. The survey UUVs may move between nodes with a time cost c_{ij}^{move}, however are significantly slower than USVs with transportation cost of $c_{ij}^{trans} < c_{ij}^{move}$. The USVs also require a fixed amount of time to collect and deploy the survey UUVs. The collection and deployment costs are denoted c^{dock} and c^{deploy}, respectively. The vehicles initially start at an origin node o, with each USV n containing A_n survey UUVs.

We assume it is known due to enemy course of action (ECOA) intelligence, environmental information, and other sources that there is an expected number of mine-like contacts (MILCOs) $\hat{N}_s^{MILCO}$ before completion of survey actions within area s, as well as an expected number of mines $\hat{N}_s^{mines}$ before completion of RI actions in area s. Upon completion of the survey and RI actions, these quantities and their expected locations will be known, and denoted as N_s^{MILCO} and N_s^{mines}, respectively. Until the locations are known from the actual survey and RI actions, we assume the location of the MILCOs and mines follow density functions $p_{MILCO}(s)$ and $p_{mine}(s)$. Using the expected number of MILCOs and mines, we assume that expected time costs for performing RI and neutralize actions can be estimated for each service area s. We denote the expected RI and neutralization costs as $\hat{c}_{r,s}^{RI}$ and $\hat{c}_{n,s}^{neut}$ using vehicle r or n on service area s.

We outline the general logistical constraints for an operation where multiple UUVs may be transported throughout the field via an USV based largely on [8]. We then expand the logistical constraints with the specific constraints developed for general survey, reacquire and identify (RI), and prosecution tasking such as found in MCM operations.

The variable $\bar{\mathbf{F}}_{m,p}$ is assigned a value using the constraints

$$\forall m \in \mathcal{M}, \forall p \in \mathcal{P} \tag{1}$$

$$\begin{aligned} \bar{\mathbf{F}}_{m,p} = \sum_{p' \leq p} \Bigg[\bar{\mathbf{F}}_{m,p'}^{charge} - \sum_{s \in \mathcal{S}} f_{m,s}^{survey} \bar{\mathbf{I}}_{m,p',s}^{survey} \\ - \sum_{d \in \mathcal{D}} \left(f_{m,d}^{dock} \bar{\mathbf{I}}_{m,p',d}^{dock} + f_{m,d}^{deploy} \bar{\mathbf{I}}_{m,p',d}^{deploy} \right) \\ - \sum_{v \in \mathcal{V}} f_{m,p',v}^{move} \bar{\mathbf{I}}_{m,p',v}^{move} \Bigg] \end{aligned}$$

where $f_{m,s}^{survey}$, $f_{m,d}^{dock}$, $f_{m,d}^{deploy}$, and $f_{m,v}^{move}$ are fixed costs for executing their respective goals, and $\bar{\mathbf{F}}_{m,p}^{charge}$ is a floating variable indicating the amount a vehicle is charged during a phase.

A. Survey Constraints

The first constraint for a surveying action is that a survey area can only be surveyed if it is scheduled after a deployment action at a node $d \in \mathcal{D}_s$ or move action. Thus, we have the constraint

$$\forall m \in \mathcal{M}, \forall n \in \mathcal{N}, \forall p \in \mathcal{P}, \forall d \in \mathcal{D}, \forall s \in \mathcal{S} \tag{2}$$

$$\begin{aligned} \bar{\mathbf{I}}_{m,p,s}^{survey} = 1 \Rightarrow \sum_{n \in \mathcal{N}} \bar{\mathbf{I}}_{mn,p-1,d}^{deploy} \\ + \sum_{d' \in D} \bar{\mathbf{I}}_{m,p-1,(d',d)}^{move} = 1. \end{aligned}$$

where $(d', d) \in \mathcal{V}$ Next, we have the constraint that a survey action can only be performed if it is scheduled while the vehicle is deployed

$$\forall m \in \mathcal{M}, \forall p \in \mathcal{P} \qquad \sum_{s \in \mathcal{S}} \bar{\mathbf{I}}_{m,p,s}^{survey} = 1 \Rightarrow \bar{\mathbf{D}}_{m,p} = 0 \tag{3}$$

Additionally, every survey goal should be completed exactly once by the group of vehicles. Thus,

$$\forall s \in \mathcal{S} \qquad \sum_{m \in \mathcal{M}} \sum_{p \in \mathcal{P}} \bar{\mathbf{I}}_{m,p,s}^{survey} = 1. \tag{4}$$

B. Reacquire and Identify Scheduling Constraints

An example of such mixed-integer linear program constraints is shown in the following. Let $\{p, p', p''\}$ represent indices in the set of mission phases $\mathcal{P}$. Index s iterates over the set $\mathcal{S}$ of survey areas, and $d \in \mathcal{D}$ represents the set of dock/deploy points for the mission where UUVs may onboard or offboard the USV. Three criteria must be met prior to the RI action:

1) the area s must have been surveyed by a survey UUV,
2) the survey UUV m from survey UUV set $\mathcal{M}$ must have been docked with a USV,
3) sufficient time must have elapsed from when the survey UUV had docked with a USV to perform post-mission analysis (PMA).

The first two criteria may be captured by the constraint

$$\begin{aligned} &\forall r \in \mathcal{R}, \forall p \in \mathcal{P}, \forall s \in \mathcal{S} \\ &\bar{\mathbf{I}}^{RI}_{r,p,s} = 1 \Rightarrow \sum_{m \in \mathcal{M}} \left[\sum_{p' < p} \left[\sum_{n \in \mathcal{N}, d \in D} \bar{\mathbf{I}}^{dock}_{m,p',s} \geq 1 \right.\right. \\ &\left.\left. \wedge \sum_{p'' < p'} \bar{\mathbf{I}}^{survey}_{m,p'',s} \geq 1 \right]\right] \geq 1, \end{aligned} \tag{5}$$

where $\bar{\mathbf{I}}^{RI}_{r,p,s}$ is the action indicator variable for scheduling RI UUV r performing an RI action during phase p in area s. In order to capture the third criteria, we create the constraint

$$\begin{aligned} &\forall n \in \mathcal{N}, \forall p \in \mathcal{P}, \forall d_n \in \mathcal{D}_n \\ &\bar{\mathbf{I}}^{RI}_{n,p,d_n} = 1 \Rightarrow \bar{\mathbf{T}}^{start}_{n,p} \geq \bar{\mathbf{T}}^{PMA,end}_{s}, \end{aligned} \tag{6}$$

where $\bar{\mathbf{T}}^{PMA,end}_{m,s}$ is a new time indicator variable indicating the time PMA has ended for the RI component of an operation on survey area s, which is defined using the constraints

$$\begin{aligned} &\forall n \in \mathcal{N}, \forall p \in \mathcal{P}, \forall s \in \mathcal{S} \\ &\sum_{m \in \mathcal{M}} \left[\left[\sum_{p' \leq p} \bar{\mathbf{I}}^{survey}_{m,p',s} \geq 1 \right] \wedge \left[\bar{\mathbf{I}}^{dock}_{m,p,s} \geq 1 \right] \right] \geq 1 \Rightarrow \\ &\bar{\mathbf{T}}^{PMA,start}_{n,p,s} = \bar{\mathbf{T}}^{end}_{n,p} \\ &\forall n \in \mathcal{N}, \forall p \in \mathcal{P}, \forall s \in \mathcal{S} \\ &\sum_{M \in \mathcal{M}} \left[\left[\sum_{p' \leq p} \bar{\mathbf{I}}^{survey}_{m,p',s} \geq 1 \right] \wedge \left[\bar{\mathbf{I}}^{dock}_{m,p,s} \geq 1 \right] \right] = 0 \Rightarrow \\ &\bar{\mathbf{T}}^{PMA,start}_{s} = \bar{\mathbf{T}}^{big} \\ &\forall s \in \mathcal{S} \\ &T^{PMA,start}_{s} = \min_{n \in \mathcal{A}^{trans}, p \in \mathcal{P}} T^{PMA,start}_{n,p,s} \\ &\forall s \in \mathcal{S} \\ &\bar{\mathbf{T}}^{PMA,end}_{s} = \bar{\mathbf{T}}^{PMA,start}_{s} + \hat{c}^{PMA}_{s}, \end{aligned} \tag{7}$$

where $\bar{\mathbf{T}}^{big}$ is a large constant.

C. Neutralization Constraints

We now turn to determining a proper schedule for the neutralization phase of a mission. In the scheduling of neutralization activities of a given survey area s, three criteria must be met prior to the Neutralization action:

1) the reacquire and identification task must have been performed on area s by an RI UUV $r \in \mathcal{R}$,
2) the RI UUV r must have been docked with a USV,
3) sufficient time must have elapsed from when the survey UUV had docked with a USV to perform post-mission analysis.

The first two criteria may be captured by the constraint

$$\begin{aligned} &\forall n \in \mathcal{N}, \forall p \in \mathcal{P}, \forall d_n \in \mathcal{D}_n \\ &\bar{\mathbf{I}}^{neut}_{n,p,d_n} = 1 \Rightarrow \sum_{r \in \mathcal{R}} \left[\sum_{p' < p} \left[\sum_{n \in \mathcal{N}} \bar{\mathbf{I}}^{dock}_{rn,p',s} \geq 1 \right.\right. \\ &\left.\left. \wedge \left[\sum_{p'' < p'} \bar{\mathbf{I}}^{RI}_{r,p'',s} \geq 1 \right] \right]\right] \geq 1, \end{aligned} \tag{8}$$

where $\bar{\mathbf{I}}^{neut}_{n,p,d_n}$ is the action indicator variable for scheduling USV n to perform an neutralization action during phase p at the neutralization point $d_n \in \mathcal{D}_n$. In order to capture the third criteria, we create the constraint

$$\begin{aligned} &\forall n \in \mathcal{N}, \forall p \in \mathcal{P}, \forall d_n \in \mathcal{D}_n \\ &\bar{\mathbf{I}}^{neut}_{n,p,d_n} = 1 \Rightarrow \bar{\mathbf{T}}^{start}_{n,p} \geq \bar{\mathbf{T}}^{PMA',end}_{s}, \end{aligned} \tag{9}$$

where $\bar{\mathbf{T}}^{PMA',end}_{m,s}$ is a new time indicator variable indicating the time PMA has ended for the RI component of an operation on survey area s, which is defined using the constraints

$$\begin{aligned} &\forall n \in \mathcal{N}, \forall p \in \mathcal{P}, \forall s \in \mathcal{S} \\ &\sum_{r \in \mathcal{R}} \left[\left[\sum_{p' \leq p} \bar{\mathbf{I}}^{RI}_{r,p',s} \geq 1 \right] \wedge \left[\bar{\mathbf{I}}^{dock}_{r,p,s} \geq 1 \right] \right] \geq 1 \Rightarrow \\ &\bar{\mathbf{T}}^{PMA',start}_{n,p,s} = \bar{\mathbf{T}}^{end}_{n,p} \\ &\forall n \in \mathcal{N}, \forall p \in \mathcal{P}, \forall s \in \mathcal{S} \\ &\sum_{r \in \mathcal{R}} \left[\left[\sum_{p' \leq p} \bar{\mathbf{I}}^{RI}_{r,p',s} \geq 1 \right] \wedge \left[\bar{\mathbf{I}}^{dock}_{r,p,s} \geq 1 \right] \right] = 0 \Rightarrow \\ &\bar{\mathbf{T}}^{PMA',start}_{s} = \bar{\mathbf{T}}^{big} \\ &\forall s \in \mathcal{S} \\ &T^{PMA',start}_{s} = \min_{n \in \mathcal{A}^{trans}, p \in \mathcal{P}} T^{PMA',start}_{n,p,s} \\ &\forall s \in \mathcal{S} \\ &\bar{\mathbf{T}}^{PMA',end}_{s} = \bar{\mathbf{T}}^{PMA',start}_{s} + \hat{c}^{PMA'}_{s}. \end{aligned} \tag{10}$$

With the above constraints, we can incorporate the scheduling of neutralization sorties within our scheduling algorithm. However, in order to incorporate the physical constraints of allowing a USV to travel to a location, and requiring the transit for a neutralization action, we must include the following constraints. First, we require that in order for a

USV to neutralize a target, it must have previously traveled to that target, represented by the constraint

$$\forall n \in \mathcal{N}, \forall p \in \mathcal{P} \; s.t. \; p \neq 1, \forall d_n \in \mathcal{D} \tag{11}$$
$$\bar{\mathbf{I}}^{neut}_{n,p,d_n} = 1 \Rightarrow \sum_{d'_n \in \mathcal{D}_n \; s.t. \; <d'_n, d> \in \mathcal{V}_n} \bar{\mathbf{I}}^{move_n}_{n,p-1,d'} = 1.$$

Secondly, in order to allow transits between dock/deploy nodes $\mathcal{D}$ and neutralization nodes $\mathcal{D}_n$, we require the constraint

$$\forall n \in \mathcal{N}, \forall p \in \mathcal{P} \; s.t. \; p \neq 1, \forall d_n \in \mathcal{D}_n \tag{12}$$
$$\sum d'_n \in \mathcal{D}_n \; s.t. \; < d'_n, d_n > \in \mathcal{V}_n \bar{\mathbf{I}}^{move_n}_{n,p,d_n} = 1 \Rightarrow$$
$$\sum_{d'_n \in \mathcal{D}_n \; s.t. \; <d'_n, d_n> \in \mathcal{V}_n} \bar{\mathbf{I}}^{move_n}_{n,p-1,d'} = 1.$$

We now conclude the constraints that implement neutralization actions, and turn to the overall cost function used in the optimization process.

III. Cost Function

The cost function to be minimized in our initial formulation is to minimize the end time of the last phase over all vehicles. That is, our objective is

$$\text{minimize} \; \max_{a} \bar{\mathbf{T}}^{end}_{a,P}. \tag{13}$$

We now turn to a complexity reduction strategy we have developed in order to optimize the different operation phases for MCM scheduling.

IV. Schedule Update and Complexity Reduction using Task Fixing

Next, we will discuss the methodology by which we update estimated future tasks with the latest information, while simultaneously reducing the computational complexity of the overall optimization problem. We update the overall mission plan and reduce the computational complexity by means of *task fixing*. In task fixing, schedule optimization is run multiple times, once for each general phase of an MCM operation: survey, RI, and neutralization. In each phase, the assets required to perform that phase are optimized, as well as the assets for the following phase using estimates of the time required given prior information. For example, environmental and ECOA information may be used to develop an estimate for the number of MILCOs each survey area may have, which allows for an estimate of the time required to perform an RI operation on a given area, $\hat{c}^{RI}$. Once one component of the operation is completed as scheduled, the information is updated using obtained information such as actual number of MILCOS, and any previous and currently on-going components of that phase's schedule are treated as fixed constraints. Then, a second optimization is performed with the schedule horizon extended to the following phases. Figure 2 shows an illustration of task-fixing. In the figure, a survey operation is scheduled, as well as an RI operation based on *predicted* duration during Phase 1. In Phase 2, the survey UUV's tasks and required USV's tasks are treated as fixed, information for the RI operation is updated, and the schedule is re-optimized while including estimates for the neutralization phase. During each optimization, we extend the number of phases and thus the end phase from which we evaluate the objective function found in (13).

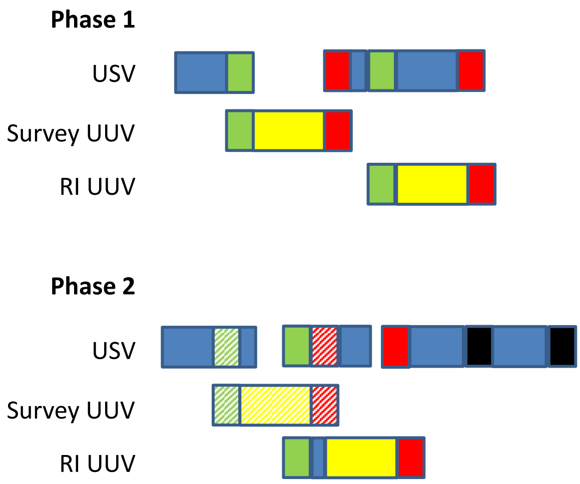


Fig. 2. Notional illustration of task-fixing. In Phase 1, all blocks represent optimized actions. In Phase 2, solid colors represent actions that may be optimized. Shaded colors represent tasks fixed from the previous phase that may not be optimized, but instead treated as constants within the optimization process.

V. A Novel Visualizer for Mine Warfare

We now present a recently-developed three-dimensional mine warfare (MIW) visualization system that may be used for visualizing simulated systems in different scenarios. The MIW Visualizer was created for system developers to use as a debugging and demonstration tool for real-time simulated and conceptual systems. Any system (land, air or sea) can be displayed concurrently and events can either be pre-planned and executed or input dynamically from an external source. We will provide a technical overview of the visualization framework and present a simulation utilizing the plan as designed by the automated MCM scheduling algorithms discussed and visualized with the MIW Visualizer.

A. Graphical User Interface

The first component of the MIW Vizualizer we will discuss is the graphical user interface (GUI) front-end. The GUI is used to configure an MCM scenario, consisting of what vehicles will be rendered, contact lists of various types such as known mines and mine-like contacts, as well as false contacts for research and development purposes. The GUI, based on Javascript, is shown in Figure 4. In the figure, a number of mines within a mine laydown, a large ship icon, and visual queues from mission plans are drawn for input into the visualizer.

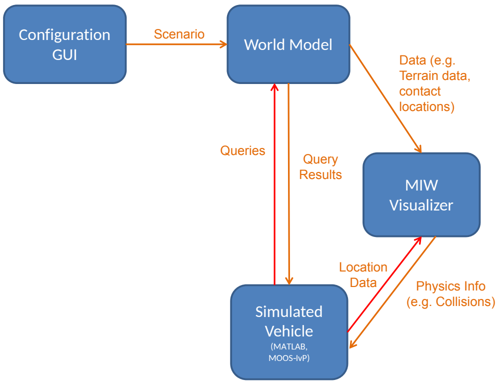


Fig. 3. Interface diagram for the MIW Visualizer.

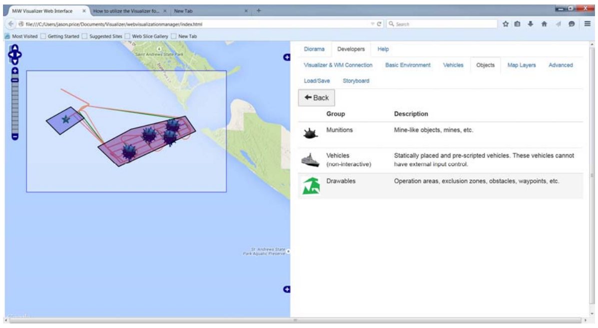


Fig. 4. MIW Visualizer front-end.

B. Communication via Google Protocol Buffers

We interface the GUI and vehicle position information using Google protocol buffers, also known as protobufs. Google protocol buffers are used to support serialization of data that can be transmitted through standard networking protocols [9]. The user generates protobuf files that contain all of the pertinent information that is to be shared between two communicating programs. However, the protobuf files must translated to source code, and as such, a language-specific source code generator is required. Standard protobuf generators include C++, Java, and Python.

For our purposes, a Java protobuf source code generator was used to provide the underlying data structures used while communicating between the matlab scripts and the visualizer. This choice was made as MATLAB provides extensive support for Java. The protobuf messages that must be sent to the visualizer include an environment message, a vehicle broadcast, and a vehicle position message. The the environment message establishes the region for which the mission will occur. Next, a vehicle broadcast message is sent to inform the visualizer as to which vehicles will participate in the scenario. The vehicle broadcast message allows the visualizer to render the correct model for each vehicle. Finally, time-stamped vehicle position messages containing position and orientation information for each vehicle in the scenario are sent to notify the visualizer to maintain the updated visual state of each vehicle.

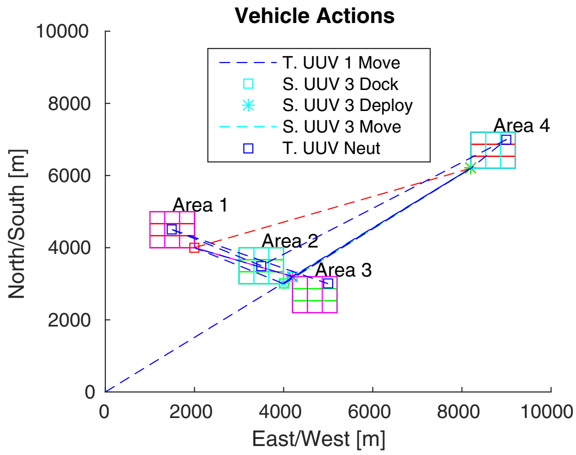


Fig. 5. Plot of vehicle actions within an MCM area.

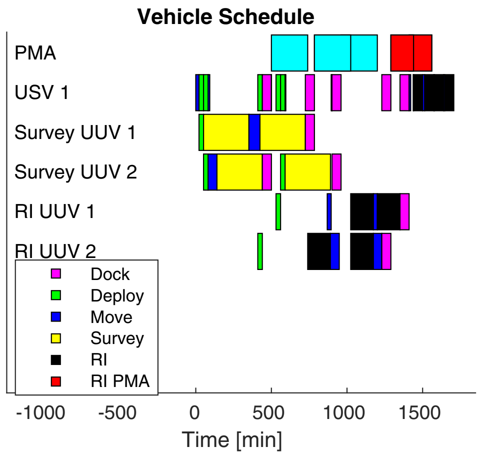
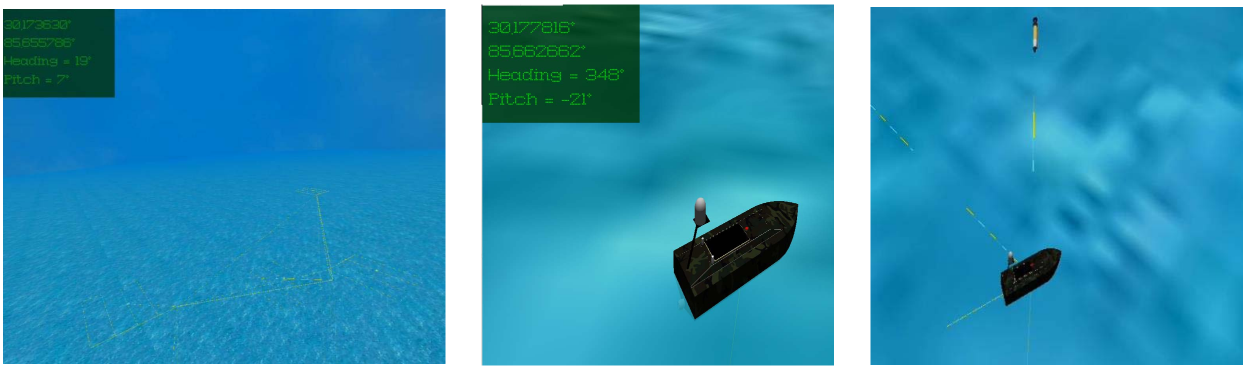


Fig. 6. Simulated MCM Schedule. Gantt chart represents the duration of each action performed by each vehicle at a point in time. Additionally, PMA processing times are shown at the top. Blue PMA times represents PMA for survey components of mission. Red PMA times represent RI components. Green and magenta represent vehicle deployment and docking, respectively. Blue represents vehicle transits. Yellow represents surveys. Black on RI vehicles represents RI actions. Black on the USV row represents neutralization actions.

VI. Simulation Results

We now present simulation results for the proposed scheduling framework and visualization environment. The schedule optimization was performed using the IBM's ILOG CPLEX optimization software [10]. In the following simulations, we assume that a USV has a speed of 8 m/s, while UUVs have a speed of 1.5 m/s. We assume each of the $S = 4$ survey areas require $c^{survey} = 300$ minutes. RI and Neutralize tasks require $c^{RI} = 150$ and $c^{Neut} = 60$ minutes, respectively. Furthermore, we assume that each docking action requires $c^{dock} = 60$ minutes, and $c^{deploy} = 30$ minutes. Pos-mission analysis time for the survey and neutralize phases required $\hat{c}_s^{PMA'} = 240$ and $\hat{c}_{RI}^{PMA'} = 15-$ minutes, respectively. All simulations were performed on a 2.8 GHz quad-core CPU equipped with 8 GB RAM.

Figures 5 and 6 illustrate a simulation of $N = 1$ USV, $M = 2$ survey UUVs, $R = 2$ RI UUVs, and $P = 26$ phases.



(a) Simulation showing planned paths. (b) USV. (c) USV deploying UUV.

Fig. 7. Images from visualization of MCM scheduling operation using the MIW Vizualizer. Left: Image of a UUV deploying from a USV. Middle: Image of a planned set of survey areas. Right: Close-up image of the USV.

Fig. 8. Image from the MIW Visualizer of several simulation-capable vehicle models.

Figure 5 represents the spatial map of the various vehicles moving throughout the field. Figure 6 shows a temporal schedule of the overall mission. The MILP optimization software was run with a cutoff time of 10 minutes for each optimization phase (Survey-RI, and RI-Neutralize). For the purposes of this simulation, we assumed that all targets were identified as mines, and therefore no additional Neutralize-only phase was necessary. As seen in the figure, the USV travels to the center-most survey area, survey area 2, to deploy the survey vehicles. The USV then travels to a further area to deploy the RI vehicles. The USV then commences to travel about the area docking with various vehicles as the vehicle travel under their own power to maximize the speed at which post-mission analysis commences, including docking with a vehicle to immediately deploy it again in order to survey the final area. It is noted that the UUVs largely use their own motion in order to actually travel to areas to prosecute. We suspect this is due to sacrificing speed on-station for maximizing PMA processing speed.

Figure 7 shows three images of an MCM scheduling operation depicted using the MIW visualizer. The left-most image shows an aerial view of the mission area, including planned survey paths and vehicle transits. The center figure shows a close-up of a USV to serve as the transport vehicle in the illustrative simulation. The right-most figure shows the USV deploying a UUV. Figure 8 shows other naval assets that can be rendered as well as given waypoint commands using interfaces to various autonomy architectures such as the MOOS-IvP autonomy framework.

VII. Conclusions

We have presented algorithms to use MILP optimization to autonomously develop schedules for multiple unmanned vehicles to perform a notional MCM operation, as well as a novel MIW visualizer under development to effectively simulate vehicles executing the schedule. Simulation results show that the automated scheduling algorithm generates efficient schedules, incorporating post-mission analysis constraints, vehicle path planning, and mission constraints into an overall MCM operation schedule.

In future efforts, scenario configurability of the visualizer will be expanded, allowing a user to use real bathymetry data, place contacts and obstacles in specific locations and display different types of sensor data. Simulated vehicles will be able to query and understand the environment, ingesting that information into its autonomy. Additionally, we plan to develop a decentralized optimization framework to increase the tractability and efficiency of the optimization process, as well as more formally incorporate schedule uncertainty in order to create robust schedules.

VIII. Acknowledgment

This work has been supported by the Office of Naval Research Code 32 and Independent Applied Research programs.

References

[1] S. Sariel, T. Balch, and N. Erdogan, "Naval mine countermeasure missions," *IEEE Robotics & Automation Magazine*, vol. 15, pp. 45–52, 2008.

[2] A. J. Shafer, M. R. Benjamin, J. J. Leonard, and J. Curcio, "Autonomous cooperation of heterogeneous platforms for sea-based search tasks," in *MTS/IEEE OCEANS*, 2008.

[3] R. Costa and T. A. Wettergren, “Managing the areas of responsibility for collaborative undersea searchers,” in *IEEE/MTS OCEANS*, 2010.
[4] J. C. Hyland and C. M. Smith, “Automated area segmentation for ocean bottom surveys,” in *Proc. SPIE 9454, Detection and Sensing of Mines, Explosive Objects, and Obscured Targets XX*, 2015.
[5] M. Molineaux, B. Auslander, P. G. Moore, and K. M. Gupta, “Minimally disruptive schedule repair for mcm missions,” in *Proc. SPIE 9454, Detection and Sensing of Mines, Explosive Objects, and Obscured Targets XX*, 2015.
[6] G. W. Pollitt, “Mine countermeasures (mcm) tactical decision aids (tdas), a historical review,” *Military Operations Research*, vol. 11, no. 3, pp. 7–17, 2006.
[7] M. R. Benjamin, J. J. Leonard, H. Schmidt, and P. M. Newman, “An overview of moos-ivp and a brief users guide to the ivp helm autonomy software,” Massachusetts Institute of Technology, Technical Report MIT-CSAIL-TR-2009-028, 2009.
[8] M. J. Bays and T. A. Wettergren, “A solution to the service agent transport problem,” in *IEEE International Conference on Intelligent Robots and Systems*, To appear.
[9] Google. (2015, July) Google protocol buffers. [Online]. Available: https://developers.google.com/protocol-buffers/
[10] *IBM ILOG CPLEX User Manual*, IBM Corporation, 2013.