Team Planning for Unmanned Vehicles in the Risk-Aware Mixed-Initiative Dynamic Replanning System

John Wilde, Dino DiBiaso, and Margaret Nervegna

Abstract—**The Navy envisions using heterogeneous teams of Unmanned Aerial Vehicles (UAVs) and Unmanned Undersea Vehicles (UUVs) for future missions. The operator must interact with both the team as an entity as well as with individual vehicles. To accomplish these missions, the team must autonomously plan activities to meet the team mission objectives while abiding by operating constraints.**

As part of the Office of Naval Research's (ONR) Intelligent Autonomy program, Draper is developing the Risk-aware Mixed-initiative Dynamic Replanning (RMDR) system. The RMDR operator specifies team-level tasking, approves generated plans, and may re-task the team during the mission. The operator tasking includes Search tasks with value and precedence constraints, UUV Sentry tasks, UAV Communication Relay tasks, as well as vehicle, task, and region constraints. The RMDR team planner allocates the tasking among the available vehicles to accomplish the team objectives, abiding by the constraints.

This paper focuses on the team-level planning capabilities and algorithms, specifically the Search task optimization and the Communication Relay task assignment. The Search task algorithm allocates Search tasks to vehicles using a mixed-integer optimization formulation that minimizes the total mission time and maximizes mission value. The Communication Relay task assignment algorithm calculates communication locations and time intervals for UAVs to maximize communication coverage for the UUV tasks.

I. Introduction

This paper describes the techniques used to perform initial mission planning and to dynamically replan a group of Unmanned Undersea Vehicles (UUVs) and Unmanned Aerial Vehicles (UAVs). The algorithms that are described were developed under the Risk-aware Mixed-initiative Dynamic Replanning (RMDR) program [1], part of the Office of Naval Research's (ONR) Intelligent Autonomy initiative [2]. A goal of the initiative is to enable forward-deployed naval platforms with limited manning to effectively plan and execute complex missions. The methods described in this paper give a single operator the ability to quickly generate coordinated plans across multiple vehicle types and to maintain that plan during a mission.

This work was supported in part by the Office of Naval Research under Contract N00014-03-C-0182.

J. Wilde is with the Charles Stark Draper Laboratory, Cambridge, MA 02139 USA; phone: 617-258-1669; email: jwilde@draper.com.

M. Nervegna is with the Charles Stark Draper Laboratory, Cambridge, MA 02139 USA; email: mnervegna@draper.com.

D. DiBiaso is with the Charles Stark Draper Laboratory, Cambridge, MA 02139 USA; email: ddibiaso@draper.com.

The RMDR system is designed for maritime reconnaissance scenarios, typically involving up to four UUVs and two UAVs. The RMDR operator specifies team-level tasking, approves generated plans, and may re-task the team during the mission. The operator tasking includes Search tasks with value and precedence constraints, UUV Sentry tasks, UAV Communication Relay tasks, as well as vehicle, task, and region constraints. Fig. 1 shows an example of a typical RMDR scenario with three UUVs performing Search tasks and one UAV performing a Communication Relay task.

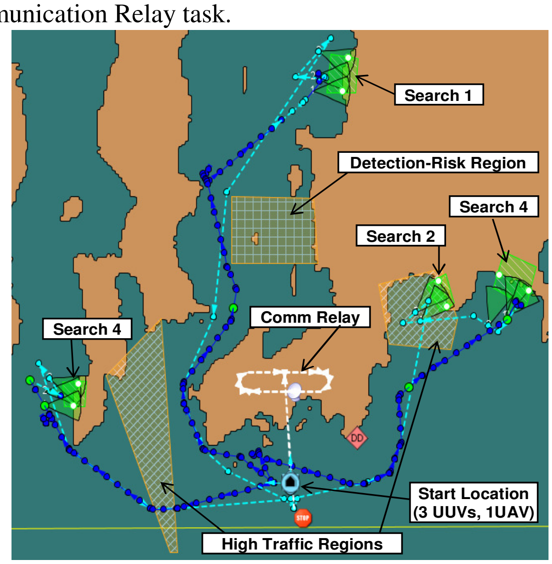

Figure 1. Example RMDR multi-vehicle scenario

In RMDR, planning is performed on a single host platform and onboard each individual vehicle. This paper focuses on the team planner on the host platform. The team planner allocates the tasking among the available vehicles to accomplish the team objectives, enables cooperation among the vehicles, and responds to opportunities or disturbances that arise during mission execution. The individual planner on each vehicle further refines its tasking based on its vehicle and sensor capabilities, as well as environmental and mission constraints. Team planning consists of three main steps:

1. <u>Preprocessing</u>: A single goal point is created to represent each Search task. If the task corresponds to a coastline region, the midpoint of the coastline is selected as the goal point; otherwise the centroid of the region is used. If a coastline region associated with a Search task is larger than a configurable amount (e.g., the coastline to be searched is greater

than 2.5 km), the region is broken into a series of smaller regions to allow the area to be partially searched or searched by a combination of different vehicles. A high-level waypoint path planner calculates a time estimate for each feasible route that a vehicle can take to accomplish each objective. The path planner accounts for avoidance regions and the operating area, and for the UUV, ensures that the vehicle can transit at a safe depth over the path.

2. Search task optimization: A mixed-integer optimization problem is formulated and solved to find the best allocation and sequencing of the available vehicles to perform the possible Search tasks, as well as the times that the tasks should be performed.

3. Communication coverage planning: Communication Relay tasks are allocated to UAVs to provide high bandwidth communication link coverage between the UUVs performing Search tasks and the host platform. The communication planning algorithm uses the output of the previous step (Search task locations, times, and vehicle assignments) to determine locations and times for UAVs to be stationed to perform Communication Relay tasks.

Section II provides an overview of the RMDR system and the system-level objectives and constraints input to the team planning module. Section III and Section IV present the two main team planner algorithms. Section V describes how the RMDR team planner reacts to deviations from the original plan, and Section VI summarizes the results.

II. Background

The RMDR system supports common tasking and control of heterogeneous naval unmanned systems based on high-level mission objectives and constraints. Using an operator interface, the operator defines different types of tasks: Search, Communication Relay, Sentry, and End Mission. The operator can construct regions that, together with the tasks, define the objectives and constraints of the mission. In addition, the operator specifies vehicle operating constraints, such as speed, travel depth/altitude, and launch and recovery times.

Fig. 2 shows the sequence of mission planning in the RMDR system. After the operator defines the system-level tasking, the operator interface sends the System Objective containing the team tasking objectives to the team planning module. When the team planner has completed allocating the tasks to vehicles, a Vehicle Objective containing the tasking instructions is sent to each vehicle. Each vehicle's planner generates a detailed route plan using a trajectory path planner, taking into account vehicle dynamics and constraints. Additionally, the vehicle's plan details activities associated with the imaging and communication tasks. After completing planning, each vehicle sends its detailed Vehicle Plan to the team planner on the host platform. If problems

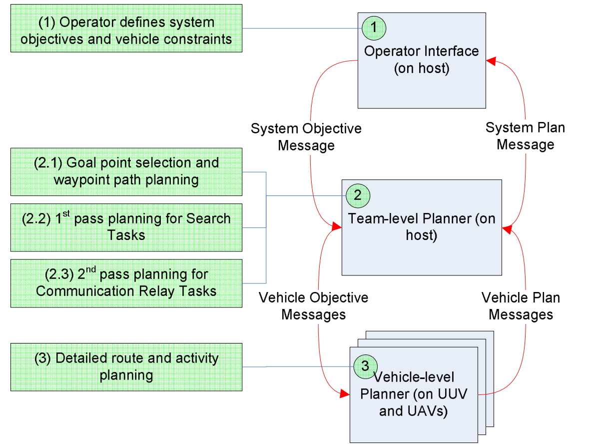

Figure 2. RMDR mission planning sequence

were encountered during planning, warning messages or failure notifications will also be reported to the host platform. The host platform waits for Vehicle Plans from each tasked vehicle and aggregates them into a System Plan which is then sent to the operator interface.

A. *Regions*

The operator can define different types of regions that are associated with the mission. Regions may optionally be constrained to apply to specific vehicles and/or to apply during certain periods of time. Fig. 3 shows a typical operational area (~ 25 km^2) and an example of each type of region. Regions (1), (2), and (3) show the three classes of Search regions that can be associated with a Search task – entirely on land, entirely in water, and overlapping a coastline, respectively. Region (4) is an area where vehicles are prohibited from communicating, and region (5) is an area where UUVs are not allowed to surface. Region (6) is an area where vehicles are not allowed to enter, and regions (7) and (8) are areas of elevated risk where vehicles incur a penalty but are not prohibited from entering. Finally, region (9) defines the operating area in which the vehicles must remain for the entire mission.

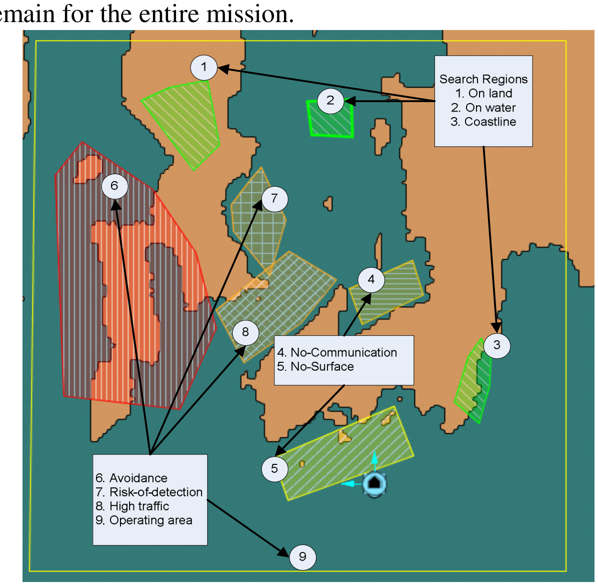

Figure 3. Region types in RMDR

B. Search Task

An RMDR mission typically involves one or more Search tasks. Two key characteristics of a Search task are (1) the geographic region to be searched and (2) the type of sensor to be used. The Search region can be located anywhere within the operational area, but its location (in water, on land, or overlapping a coastline) and sensor type (Electro-Optic (EO), or Electronic Support Measures (ESM)) constrains the type of vehicle that can be assigned to the corresponding Search task. UUVs can only be assigned to EO Search tasks with a Search region that overlaps a coastline and ESM Search tasks with a region completely in water, while UAVs can be assigned to any of these classes of Search tasks. An ESM sensor is used to detect radio and radar emitters, and an EO imaging sensor is used to take pictures from a UUV while on the surface or from a UAV while in flight.

In addition to specifying a Search region and sensor type, the operator can constrain a task to be performed by a subset of the available vehicles. The operator can also specify task start and end times, require one task to be performed prior to another task, and/or specify that a Search task is more important than another. The complete list of parameters for Search task planning is summarized in Table I.

TABLE I
SEARCH TASK PARAMETERS

Search Type 1) Electronic Support Measures (ESM): Performed by a UUV in a region completely in water, or by a UAV at any location. *2)* Electro-Optic (EO): Performed by a UUV along a coastline, or by a UAV at any location.
Region Defined by a two-dimensional convex polygon indicating the area to be searched. A goal point is created for each region, and this point is referred to as a *node* in the Search task formulation description in Section III.
Region Value Integer from 1 to 100 that specifies a measure of the importance of this task. In the absence of other constraints, higher-valued tasks will be searched before lower-valued tasks.
Vehicle(s) A subset of the vehicles in the scenario can be selected. The task will be allocated to only a member of this set
Precedence Precedence constraints ensure that one Search task is performed before another Search task, even if the tasks are allocated to different vehicles.
Time Windows The operator can constrain the start and end times at which a task can be searched (e.g. to ensure good lighting conditions).

C. Communication Relay Task

Communication between the UUVs and the host platform is over a satellite link or relayed through a UAV. The RMDR system assumes that a satellite link is available for high-priority and low-bandwidth communication, such as to report a vehicle plan or to receive new tasking. However, a faster link is needed to transmit larger messages, such as high resolution images captured during EO Search tasks.

The Communication Relay task involves placing a UAV in a racetrack orbit at a specified location for a period of time. While performing this task, the UAV is available to relay messages from UUVs to the host platform. A UAV may have Communication Relay tasks interleaved with other tasks, such as Search tasks. Two types of communication tasking can be specified by the operator:

1. <u>At Location</u>: One or more UAVs, a location and time interval are specified. RMDR does only minimal planning – the team planner tasks the UAV(s) to provide continuous communication coverage at the specified location throughout the time interval. At most one UAV is tasked at the location at any given time.
2. <u>Support Vehicles</u>: A set of UAVs (to be used) and set of UUVs (to support) are specified. RMDR planning involves finding the best locations, time windows, and UAV selection to provide communication coverage for the UUVs while performing their assigned Search tasks.

The “At Location” tasking allows the operator precise control of the UAVs while the “Support Vehicles” tasking delegates autonomous planning decisions to the team planner.

III. SEARCH TASK PLANNING

The team planner must evaluate all of the Search tasks, assign these tasks to individual vehicles, and determine the times at which the tasks should occur. It is also possible that system constraints make it impossible to accomplish all the Search tasks. For example, a task may be so far away that it cannot be reached by any vehicle without violating the recovery time constraint. Another example is if multiple tasks are given the same time window constraint and the available vehicles cannot service all the tasks within the allotted windows. If all tasks cannot be performed, then the planner selects the set of tasks that accrue the most region value.

The type of problem that has been described matches closely with the type of combinatorial optimization problem known as the Vehicle Routing Problem with Time Windows (VRPTW) which has been studied extensively in the literature ([3],[4],[5],[6],[7],[8]). Due to the difficulty of this class of problems, the most widely used solution techniques involve heuristic or meta-heuristic methods. While these methods do not perform a full search of the solution space and do not guarantee that the best solution will be found, a heuristic will generally produce good solutions with less computation time. However, due to the relatively small problem space of RMDR (having fewer than 8 regions on average), the VRPTW was formulated as a Mixed Integer Linear Program (MILP) and solved to optimality using commercial or open-source optimization software. To guarantee a solution in real time, if the optimal solution is not found within a specified time (e.g. 1 minute), the current best, yet feasible solution is used. This procedure has produced good results for the types of scenarios the RMDR system was designed to execute.

A straightforward optimization formulation was chosen because the problem instances were small enough to solve in a reasonable time, and the precedence and time window constraints could be easily introduced. Larger problems may require taking advantage of the problem structure in the formulation to more efficiently explore the search space ([10],[11],[12]) or implementing a heuristic or meta-heuristic algorithm ([9]) to find a good enough solution. Using the formulation described below, the Search tasks are translated into a matrix data structure and passed through the solver-independent Open Solver Interface (OSI). The open-source SYMPHONY solver and the commercial XPRESS-MP solvers have both been implemented and tested via the interface for the RMDR system.

A. *Mixed Integer Formulation and Implementation*

The terminology used in the problem formulation is defined in Table II. The problem is formulated by labeling each Search task (which corresponds to the goal point of a Search region, described in Table I) as a *node* from 2…*N*-1. The first node and the *Nth* node are not actual Search tasks, but rather the starting and ending locations for the vehicle. An *edge* is a path that has been calculated for a vehicle between any two nodes. An edge will have a travel time associated with it. An *edge set* for a vehicle is the complete set of possible paths that a vehicle is allowed to travel. A vehicle has a set of *valid nodes* that includes each node corresponding to a Search task that this vehicle is eligible to search. The *edge set* for each vehicle consists of: (a) edges from the vehicle's starting location to each of its valid nodes, (b) edges between all of this vehicle's valid nodes, (c) edges from each of this vehicle's valid nodes to this vehicle's end location, and (d) an edge from the start to the end node.

TABLE II
PROBLEM FORMULATION TERMINOLOGY

N	Number of nodes
V	Number of vehicles
E	$\{(i,j,v)$:an edge from node i to node j for vehicle v exists$\}$
S_i^v	Service time for vehicle v at node i
$value_i$	Reward value collected by visiting node i
$time_{ij}^v$	Travel time for vehicle v going from node i to node j
PB	$\{\ i \in 2...N-1$: Node i has precedence constraints$\}$
TW	$\{\ i \in 2...N-1$: Node i has time window constraints$\}$
B_i	Set of nodes that must be visited before node $i, i \in PB$
$WindowStart_i$	Earliest time that searching at node i may be started, $i \in TW$
$WindowEnd_i$	Latest time that searching at node i may be finished, $i \in TW$
$Start_v$	Earliest that vehicle v may leave start node
$Finish_v$	Latest that vehicle v may arrive at end node
$PrecedenceTime$	The minimum difference in arrival time between two nodes with precedence constraints

1) *Decision variables*

The decision variables in the problem are a mixture of integer-valued variables (x and u) and continuous variables (z and z_max), as shown in Table III.

TABLE III
PROBLEM DECISION VARIABLES

x_{ij}^v	$\begin{cases}1 \text{ if vehicle } v \text{ transitions from node } i \text{ to node } j \\ 0 \text{ otherwise}\end{cases}$
u_i	Order in which node i is visited, from among all nodes assigned to the vehicle visiting node i
z_i	$\begin{cases}\text{time a vehicle arrives at node } i, \forall i = 2..N-1 \\ 0 \text{ if no vehicles visit node } i, \forall i = 2..N-1\end{cases}$
z_1^v	Time a vehicle departs start node
z_N^v	Time a vehicle arrives at end node
z_max	Time last vehicle arrives at end node

To keep the problem as small as possible, a particular x_{ij}^v variable is created only if $(i, j, v) \in E$.

2) *Network flow constraints*

The first set of constraints defines the flow through the node network. Each vehicle must leave the start node (1) and enter the end node (2). Each vehicle must also preserve the flow into and out of all other nodes (3). All self-loops (i.e., x_{ii}^v) have been eliminated because this is never a valid edge for a vehicle. These constraints ensure that flow does not accumulate at any node except the end node.

$$\sum_{j:(1,j,v)\in E} x_{1j}^v = 1 \qquad \forall v = 1..V \tag{1}$$

$$\sum_{i:(i,N,v)\in E} x_{iN}^v = 1 \qquad \forall v = 1..V \tag{2}$$

$$\sum_{i:(i,j,v)\in E} x_{ij}^v = \sum_{i:(j,i,v)\in E} x_{ji}^v \qquad \forall v = 1..V,\ j = 2..N-1 \tag{3}$$

Constraints must also be added to insure that the optimal solution does not have any sub-tours. A sub-tour exists when a set of nodes are connected to one another in isolation, which does not violate the flow constraints but is an invalid solution to this problem. The u variables, which are integers and constrained to be less than the total number of nodes, can be interpreted as the order in which the nodes will be visited. The constraints of (4) prevent the formation of sub-tours because a sub-tour cycle cannot be labeled with a valid sequence of u_i values. The start and end nodes are excluded from this constraint because multiple vehicles are allowed to travel directly from the start to end nodes (if, for example, there are no feasible tasks for these vehicles to perform).

$$u_i - u_j + N\sum_{v=1}^{V} x_{ij}^v \le N-1 \qquad \begin{array}{l}\forall v = 1..V, \\ \forall (i,j,v: \\ except(i=1, j=N, v)) \in E\end{array} \tag{4}$$

$$u_i \le N \qquad \forall i = 1..N \tag{5}$$

3) Restrict visiting a node twice

A node, j, is visited by vehicle v if $\sum_{i:(i,j,v)\in E} x_{ij}^{v}$ is positive. Constraints (6) ensure that a search region is visited by at most one vehicle.

$$\sum_{v=1}^{V} \sum_{i:(i,j,v)\in E} x_{ij}^{v} \le 1 \qquad \forall j = 2..N-1 \qquad (6)$$

An alternative is to require that every node be visited exactly once by changing (6) to equality constraints. However, by allowing regions to go unsearched, plans can be generated in more highly constrained scenarios.

4) Arrival time constraints

The constraints in (7) relate the arrival time at node j to the planned arrival time at node i.

$$z_i + S_i^{v} + time_{ij}^{v} \le z_j + (1 - x_{ij}^{v})M \qquad \forall v = 1..V, (i,j,v)\in E \qquad (7)$$

For this constraint to be effective, the value of M should be larger than the sum of the maximum service time and travel time. Finally, the constraints in (8) ensure that if no vehicles visit a particular node, the arrival time at that node is zero.

$$z_j \le M \sum_{v=1}^{V} \sum_{i:(i,j,v)\in E} x_{ij}^{v} \le 0 \qquad \forall j = 1..N \qquad (8)$$

This assumes that M is larger than any possible arrival time (e.g. the maximum mission time). Note that these constraints would not be needed if every node were required to be visited exactly once.

5) Node Precedence Constraints

The node precedence constraints are implemented by placing bounds on the node arrival times (9). If nodes that have precedence constraints between them are searched by different vehicles, the *PrecedenceTime* parameter will prevent these searches from starting simultaneously.

$$z_i \le z_{j:\forall j\in B_i} + PrecedenceTime \qquad \forall i \in PB \qquad (9)$$

6) Time Windows and Vehicle Start/Recovery Constraints

Time window constraints for a particular node (10)(11) and vehicle start time (12) and end time (13) constraints are modeled by placing bounds on the values of z, the time a vehicle arrives at a node. If a vehicle were to arrive early to a node, a wait must be scheduled. To account for this, the arrival time variables must be interpreted as the time at which the search begins rather than the time at which a vehicle actually arrives at a node. This formulation implies that if a node has a time window constraint, that node *must* be visited by a vehicle.

$$z_i \ge WindowStart_i \qquad \forall i \in TW \qquad (10)$$

$$z_j + \sum_{i,v:(i,j,v)\in E} x_{ij}^{v} S_j^{v} \le WindowEnd_j \qquad \forall j \in TW \qquad (11)$$

$$z_1^{v} \ge Start_v \qquad \forall v = 1..V \qquad (12)$$

$$z_N^{v} \le Finish_v \qquad \forall v = 1..V \qquad (13)$$

7) Assignment constraints

If the tasking specifies that a particular vehicle must visit some node j, that is modeled by removing all x variables entering and exiting node j for all other vehicles.

8) Maximum vehicle finish time constraint

The decision variable *z_max* is constrained to be greater than the finish time of all the vehicles (14). The objective function can then minimize this variable in order to minimize the time that the last vehicle returns.

$$z_N^{v} \le z\_max \qquad \forall v = 1..V \qquad (14)$$

B. Two objective functions

The problem is solved twice using two objective functions: the first to find an initial solution using looser constraints and the second to find the final solution considering all constraints. The initial objective function finds the nodes that must be visited to maximize mission value and the worst case maximum mission time to do so. The final objective function improves the initial solution by maximizing collected mission value over time and minimizing mission time. After solving with the initial objective function, the “restrict visiting a node twice” constraints are changed to require the nodes chosen using the initial objective function be visited exactly once.

1) Initial objective function

The objective is to maximize mission value by greedily visiting as many higher value nodes as possible using the maximum possible time within the mission constraints. The purpose of this is to determine set of “must visit” nodes and worst case maximum mission time constraints in order for the final objective function to arrive at a reasonable solution. The *MaxMissionTime* parameter is the maximum time any vehicle may spend performing a mission. (15) and (16) are calculated to arrive at the normalization factors (17) and (18). These normalization factors are used by the objective function (19), so that both the “value” term and the “mission time” term will range between zero and one and can be fairly compared to each other.

$$SumReward = \sum_{i=2}^{N-1} value_i \qquad (15)$$

$$MaxFinishTime = \max_{\forall v=1..V} (Start_v) + MaxMissionTime \qquad (16)$$

$$nMV = \left(\frac{1}{SumReward}\right) \qquad (17)$$

$$nMS = \left(\frac{1}{MaxFinishTime}\right) \qquad (18)$$

$$\sum_{i:(i,j=2..N-1,v)\in E} \left(x_{ij}^{v} value_j\right)(nMV) + z\_max(nMS) \qquad (19)$$

2) Final objective function

The second objective function (25) rewards both visiting higher-valued nodes first and reducing the time the last vehicle finishes. The objective function is the sum of a mission value term and a mission time term. The mission value term multiplies the node value by the order in which it is visited ($u_i value_i$) – since this is a maximization problem, it is weighted with a negative scaling factor (23) to cause higher-valued nodes to be visited earlier (lower *u* values). Normalizing by *N*SumReward* is a simple way to ensure the term lies in the range of -1 to 0. The second term is simply the time at which the last vehicle finishes, normalized by the maximum finish time, as determined using the first objective function. The preference for each of these terms is adjusted using a weighting factor, *K*. This configurable factor is useful for tuning the algorithm to perform well on a variety of scenarios. The *MaxRecoveryTime* parameter is the value of *z_max* after solving using the initial objective function. Since both terms are scaled to have an absolute range of 0...1, the *K* factor can fairly weigh one term against the other.

$$SumReward = \sum_{i=2}^{N-1} value_i \quad (20)$$

$$MaxFinishTime = MaxRecoveryTime \quad (21)$$

$$K = \begin{cases} 0 & \text{Minimize time} \\ 1 & \text{Maximize value} \\ 0 < K < 1 & \text{Combine objectives} \end{cases} \quad (22)$$

$$nMV = \left(\frac{-1}{(SumReward)N}\right)K \quad (23)$$

$$nMS = \left(\frac{-1}{MaxFinishTime}\right)(1-K) \quad (24)$$

$$\sum_{i=2..N-1} (u_i value_i)(nMV) + z\_max(nMS) \quad (25)$$

C. Example of Search Task Planning

The UUV plan shown in Fig. 4 demonstrates the preference for searching higher-valued Search tasks first and the ability to skip Search tasks, if necessary, to meet a recovery time constraint. This simple scenario involves one UUV and five Search tasks (whose values are shown) and a restrictive recovery time constraint on the UUV, that makes it impossible to search all five tasks without violating the constraint. The plan depicts the UUV searching the tasks, higher-valued ones first. The lowest-valued task is skipped because the UUV must meet the recovery time deadline.

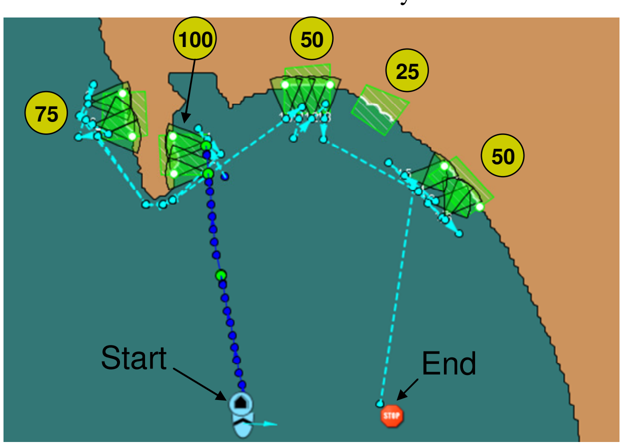

Figure 4. Example of Search task planning

IV. Communication Relay Task Planning

Communication Relay task planning involves making the best use of the available UAV resources so that they can perform both any allocated Search tasks and be stationed in other locations when needed to provide communication support for the UUV tasks. Fig. 5 shows the essential steps of the Communication Relay tasking algorithm.

Search task planning is performed prior to Communication Relay task planning, and the resulting set of UUV tasks are an input to the Communication Relay planning algorithm. The algorithm first enumerates all feasible tasking options at different times in the mission, as shown in Fig. 6. The UUV and UAV tasks are shown on a timeline with the times for Communication tasks set to the constraints that the operator has specified for that task. If no constraints are specified, the UAV vehicle's start/end times are used as default values. If a UAV has been assigned both Search and Communication tasks, the Search tasks will always take precedence, and the UAV will not be available during the time when it is performing a Search task. The timeline is broken into intervals (marked by vertical lines in Fig. 6) during which the set of tasks the system is performing does not change.

During the next step, each time interval is analyzed. For each interval, a list of candidate Communication tasks is generated for each available UAV. The candidate list consists of a set of Search tasks that occur during this interval, specified by a location and the associated UAV that will be performing the task. Also, at this step each candidate is validated to insure that the candidate is feasible. In the

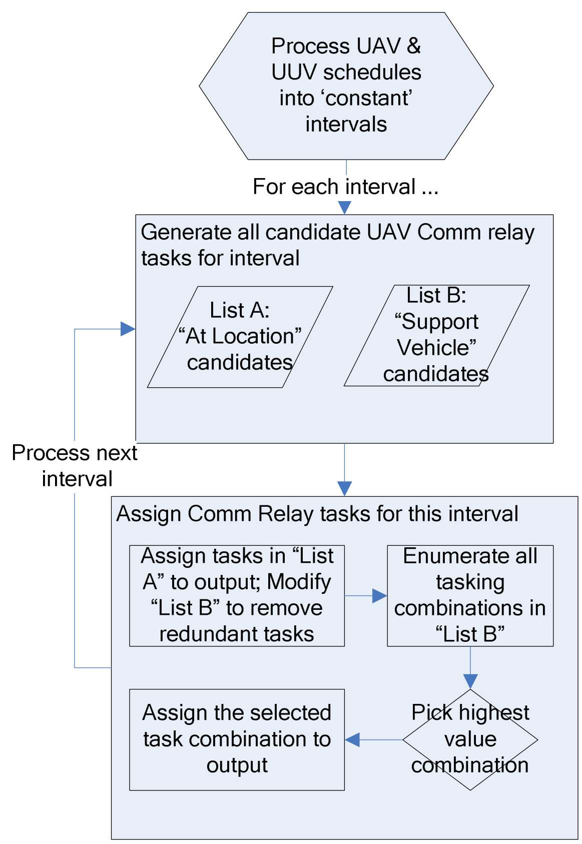

Figure 5. Communication tasking algorithm

case of an "At Location" task (the task for UAV 1 in Fig. 6), the Search tasks it supports must be within a configurable *UUVToUAVRange*, nominally 8000m, of the assigned location. For a "Support Vehicle" task (the tasks for UAV 2 in Fig. 6), the set of Search tasks being supported must (1) be performed by a UUV that this Communication task was assigned to support and (2) a feasible location must exist from which all the Search tasks in the set are within the *UUVToUAVRange* of the location. Furthermore, all candidate locations for all UAVs must be within a configurable *UAVToHostRange* (nominally 50,000 m).

Looking at the second interval in Fig. 6, the candidate set of tasks for both UAVs is {af, a, f}. This is the candidate set because these are the possible combinations of active Search tasks that can be supported during this time interval. UAV 1 is performing an "At Location" type task, so the distance between the assigned location and the centroid of the regions associated with each of the Search tasks {a,f} is calculated. If those distances are within the *UUVToUAVRange*, then the candidate set remains unchanged. If, for example, the UUV will be out of *UUVToUAVRange* while performing task {f}, then the candidate set for UAV 1 becomes just {a} because both {f} and {af} are now invalid candidates. Now, consider UAV 2 for this same interval. It has been assigned the "Support Vehicle" type of task, and is only responsible for UUVs 1 and 2. Since UUV 3 is performing task {f}, all candidates that contain this task are removed, resulting in {a} being the only valid candidate.

The final step is to make the Communication task assignment for the interval. "At Location" tasks are given priority over "Support Vehicle" tasks so these tasks are assigned first. To ensure the most efficient allocation of resources, the list of "Support Vehicle" candidates is analyzed, and any candidate that includes a task that is covered by any "At Location" task is removed. Next, all possible combinations of the remaining "Support Vehicle" candidates are enumerated. Each enumerated combination is

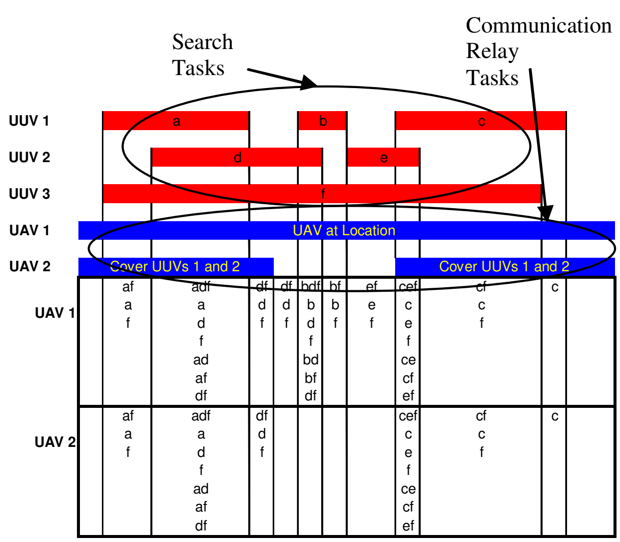

Figure 6. Communication tasking timeline and task enumeration

assigned a value equal to the sum of the values of the Search tasks for this combination. The highest value combination is selected, and those tasks are added to the output. In the case of a tie, the previous interval's assignments are examined, and preference is given to a task combination that will tend to have UAVs continue performing tasks that they have been performing during the previous interval. Communication task planning is complete after all intervals have been processed.

A. *Example of Communication Relay Task Planning*

An example involving two UUVs performing Search tasks and a UAV providing Communication Relay support is shown in Fig. 7. The UAV has been tasked with supporting both UUVs. The resulting plan shows three UAV orbits: at (1), only Search task A is active because the other UUV is still en route to Search task B; at location (2) both tasks A and B are active so a UAV orbit that is centrally located between both task locations is selected; at the last location, (3), only Search task B is active (task A has finished) so the UAV moves closer to provide dedicated coverage.

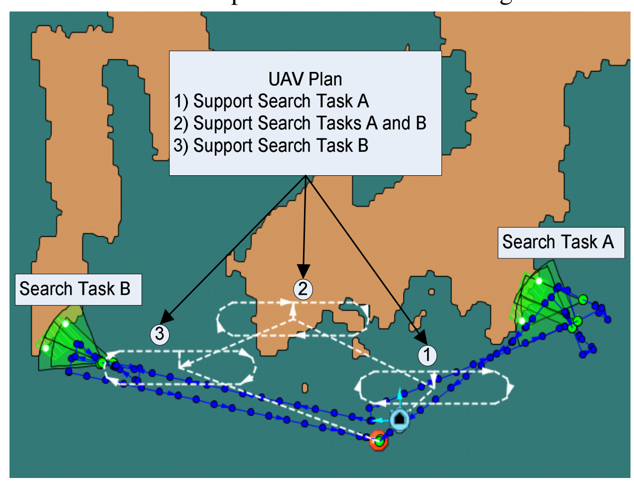

Figure 7. Communication Relay task Example

V. Replanning Events

As a mission proceeds, the actual execution of a plan may deviate from the expected plan. In these cases, such as a vehicle sensor failing or a vehicle finishing its assigned tasks earlier than planned, it is often desirable to autonomously replan the team to maintain plan synchronization across vehicles. The same algorithms that performed the initial planning are used during team replanning.

1) *Team Replanning After Sensor Failure*

While performing a Search task, it is possible that a vehicle's search sensor may fail. If this occurs, the vehicle reports this failure to the team planner, and the team planner determines if the failed vehicle has Search tasks that have yet to be completed. The team planner runs the Search task planning algorithm to attempt to allocate the remaining unsearched regions to other available vehicles, followed by the Communication Relay task planning algorithm. After determining a new allocation of tasks, the team planner may have to wait before the vehicles that need to be retasked are available to communicate (e.g. for a UUV, this means the UUV must be on the surface and not located in a no-communication region). As each vehicle establishes communication with the host platform, the team planner sends it a new Vehicle Objective and waits to receive its detailed Vehicle Plan. After all vehicles have responded, the new System Plan is sent to the operator interface and displayed for operator approval or disapproval.

2) Team Replanning When Vehicle Finishes Early

When a vehicle has completed its set of assigned tasks, it requests permission to proceed home from the team planner. This gives the team planner an opportunity to determine if an improved system plan can be created for the remaining portion of the mission by reassigning one or more of the remaining Search tasks (currently assigned to other vehicles) to vehicle that just finished. If an improved task allocation is found using the newly available vehicle, the team is retasked with these new objectives. Team retasking proceeds as described above.

3) Communication Task Synchronization

The Communication Relay tasks are initially constructed based on estimates of when UUVs will be at certain locations in the future. As a mission proceeds, the original estimates may become inaccurate due to environmental uncertainties such as unanticipated water currents, or operator intervention with a UUV during a Search activity such as commanding the UUV to "re-image" or "track" a suspected target of interest. A Communication Relay task is ineffective if it is not kept synchronized with currently executing the UUV plans. The Communication Relay task planning algorithm is triggered to replan using the most up-to-date UUV plans whenever the host platform receives a notification that a UUV is about to begin executing a new Search activity. The Communication Relay planning algorithm uses the new estimates of each vehicle's state from the most recent set of Vehicle Plans received by the host. If the resulting Communication Relay tasks are significantly different from the currently executing set of tasks, then the affected UAVs are dynamically retasked with new Vehicle Objectives.

VI. Summary and Conclusions

The algorithms for team-level allocation and sequencing of Search and Communication Relay tasks in the RMDR system have been presented. The two major team planning components are: (1) the allocation of UUVs and UAVs to perform Search tasks that involve numerous logistical constraints, in particular, time window constraints and (2) the placement and sequencing of UAVs to support the UUVs as they perform Search tasks. The Search task assignment problem was formulated using a MILP formulation of Vehicle Routing Problem with Time Windows and solving using a third-party optimization solver. Communication Relay task assignment is accomplished using the solution to the first problem as input to an enumeration algorithm that finds all feasible solutions and chooses the best one.

The RMDR multi-vehicle system has been demonstrated in high-fidelity simulation in five different operational areas, with scenarios consisting of combinations of two to six UUVs and UAVs, as well as a host platform. Additionally, the RMDR system has been demonstrated in-water with one live asset, the Autonomous Maritime Navigation (AMN) Unmanned Surface Vehicle (USV) acting as a surrogate UUV, along with simulated UUV and UAV vehicles, in three different operating areas – Fort Monroe, VA; Patuxent River, MD; and Wallops Island, VA. The system-level team planning approach presented here has shown to be capable of generating and maintaining team plans across numerous scenarios in many different littoral operational areas.

References

[1] M. Ricard, M. Nervegna, “Risk-aware Mixed-initiative Dynamic Replanning (RMDR) Program Update,” in Proceedings of AUVSI Unmanned Systems North America 2006, Orlando, FL, August 2006.
[2] M. Steinberg, "Intelligent Autonomy for Unmanned Naval Systems,” SPIE Unmanned Systems Technology Conference, Defense and Security Symposium, 2006.
[3] Cordeau, J.-F., Desaulniers, G., Desrosiers, J., Solomon, M.M., Soumis, F. "VRP with Time Windows". In P. Toth and D. Vigo (eds.): The Vehicle Routing Problem, SIAM Monographs on Discrete Mathematics and Applications, vol. 9, Philadelphia, PA, 157-193. 2002.
[4] M. M. Solomon. "Algorithms for the Vehicle Routing Problem with Time Windows". Transportation Science, 29(2), pp. 156-166. 1995.
[5] H.C. Lau, M. Sim and K.M. Teo. "Vehicle Routing Problem with Time Windows and a Limited Number of Vehicles", European Journal of Operational Research 148: 559-569, Elsevier, 2003.
[6] L.H. Lee, K.C. Tan, K. Ou and Y.H. Chew. "Vehicle Capacity Planning System: A Case Study on Vehicle Routing Problem With Time Windows", IEEE Transactions on Systems, Man and Cybernetics, Part A, 33(2), 169-178, 2003.
[7] A. Rhalibi and G. Kelleher. "An Approach to Dynamic Vehicle Routing, Rescheduling and Disruption Metrics", Systems, man and cybernetics. Proceedings (SMCC'03), Washington D.C., U.S.A., 3613-3618, 2003.
[8] Russell RA. Hybrid heuristics for the vehicle routing problem with time windows. Transportation Science 29: 156–66. 1995.
[9] K,C. Tan, L.H. Lee, Q.L. Zhu and K. Ou, "Heuristic methods for vehicle routing problem with time windows" Artificial Intelligence in Engineering, 15, 281-295, 2001.
[10] M. L. Fisher, "Optimal Solution of Vehicle Routing Problems Using Minimum K-trees", Operations Research 42, 626-642, 1994.
[11] T.K. Ralphs, L. Kopman, W.R. Pulleyblank, and L.E. Trotter Jr. "On the Capacitated Vehicle Routing Problem", Mathematical Programming Series B 94 (2003), 343.
[12] U. Blasum, and W. Hochstättler. "Application of the Branch and Cut Method to the Vehicle Routing Problem". Technical Report zaik2000-386, Centre of Applied Computer Science, University of Cologne, Germany. 2000.