Minimally Disruptive Schedule Repair for MCM Missions

Matthew Molineaux, Bryan Auslander
Philip G. Moore, and Kalyan Moy Gupta

Knexus Research Corp.
174 Waterfront Street, Suite 310, National Harbor, MD 20745
firstname.lastname@knexusresearch.com

Abstract

Mine countermeasures (MCM) missions entail planning and operations in very dynamic and uncertain operating environments, which pose considerable risk to personnel and equipment. Frequent schedule repairs are needed that consider the latest operating conditions to keep mission on target. Presently no decision support tools are available for the challenging task of MCM mission rescheduling. To address this capability gap, we have developed the CARPE system to assist operation planners. CARPE constantly monitors the operational environment for changes and recommends alternative repaired schedules in response. It includes a novel schedule repair algorithm called CLOSR that automatically repairs broken schedules while satisfying the requirement of minimal operational disruption. It uses a case-based approach to represent repair strategies and apply them to new situations. Evaluation of CLOSR on simulated MCM operations demonstrates the effectiveness of case-based strategy. Schedule repairs are generated rapidly, ensure the elimination of all mines, and achieve required levels of clearance.

1. MOTIVATION & BACKGROUND

The location, identification, and neutralization of enemy explosive ordnance are key to naval power projection and sea control, two core capabilities of U.S. maritime power, as characterized by A Cooperative Strategy for 21st Century Seapower (2010). Missions that accomplish these tasks are referred to as mine countermeasures (MCM) missions, and they are more important than ever due to the increasing prevalence of asymmetric warfare, and the relatively low expense and high impact of mine threats. A key efficiency roadblock to these missions is the complexity and uncertainty of their schedules, which are meticulously constructed and revised frequently once a mission has started. The complexity of these missions arise from a large number of potential combinations that must be considered to coordinate personnel, equipment, and autonomous vehicles; each resource has its own set of capabilities and operational constraints, as well as characteristic failure points. Achieving mission success requires combining sensor data from all resources to reduce high levels of uncertainty, and mistakes can result in loss of personnel and expensive highly specialized equipment. Frequent disruptions in MCM operations occur due to changes in sea state, visibility, weather, resource and communication bandwidth in the area of operations, and equipment failure, among other factors, all of which interfere with resource availability and/or readiness. Therefore, schedules for MCM operations require frequent changes and updates. Current practice calls for manually creating schedules, which is manpower intensive, error prone and prevents thorough exploration of efficient alternatives. Technological solutions for assisting with these problems are, as yet, nonexistent. In particular, there is no support for exploiting the most up to date situation information to reschedule tasks so as to maximize mission performance (i.e., increase clearance and reduce risk).

To meet these needs, we are developing a system for MCM operations decision making and planning support called Cognitive Architecture for RePlanning and Execution (CARPE). CARPE builds upon a foundation of cognitive architecture components and algorithms to perform the real-time monitoring, analysis, and rescheduling tasks that MCM planners perform on a frequent basis. CARPE is the first system to support interactive rescheduling of entire
MCM operations. A pivotal component of this system is the automated rescheduling component, which modifies existing schedules in response to problems as they arise. CARPE’s rescheduling component is a novel case-based rescheduling algorithm, Case-Based Local Schedule Repair (CLOSR), which applies expert knowledge to create, remove, and reassign tasks as necessary to adapt to real-world problems so as to cause minimal operational disruptions.

In this paper, we discuss the challenges of continuous situation monitoring, mission disruption root cause analysis, and interactive rescheduling in the context of MCM operations; the design of the CARPE system which assists with these tasks; and the design of CLOSR, CARPE’s automated rescheduling algorithm. We close with an empirical study that demonstrates that CLOSR proposes effective schedule modifications that achieve mission success despite problems severe enough to cause the original schedule to become invalid.

2. MINE COUNTERMEASURES MISSION SCHEDULING & OPERATIONS

MCM operations involve the location, identification, and neutralization of sea mines (Naval Expeditionary Warfare Vision, 2010). These operations employ surface vehicles, aircraft, divers, and unmanned vehicles, and can take weeks to plan and execute. During execution, they are disrupted early and often by events such as unforeseen weather conditions, technological failures, and incorrect enemy course of action estimations. Handling these disruptions requires constant monitoring and frequent modification of the schedule. While technology exists to automatically create an initial schedule, distribute tasks, and track task completion, the critical monitoring and rescheduling tasks have been, to date, poorly supported (Garcia and Wettergren, 2012). Figure 1 shows how this adversely affects the efficiency and effectiveness of MCM decision-making and operations, due to a gap in the well-known observe, orient, decide, act (OODA) decision cycle (Boyd, 1995).

![Figure 1. Lack of support for observe, orient, decide, act (OODA) loop in current MCM operations](image)

MCM operations involve a unique set of specialized tasks that must be scheduled to minimize the risk to ships from sea mines. What follows is a brief description of the tasks in an MCM operation and their characteristics. The schedule for an MCM operation tasks multiple vehicles to repeatedly hunt and/or sweep subsections of a specified threat area where mines are expected, slowly transiting back and forth in a lawnmower-like search pattern, until the risk of remaining mines is reduced to an acceptably low level. The paths followed by these search vehicles are referred to as tracks.
**Hunting** is a search and destroy activity that encompasses (1) use of specialized sensors to find underwater objects that are *mine-like*, (2) identification of mine-like objects as *mines* or *non-mines*, and (3) neutralization of all discovered mines. The sensor apparatus used for hunting is limited in two dimensions: a characteristic *search distance* at which objects can be successfully detected, and a *probability of detection*, which describes the equipment’s sensitivity within that range to the size and reflectivity of mine casings. Because mines may be missed by the sensors, it’s important to space tracks when hunting to maintain a sufficiently high probability that a mine at any given position will be detected; this is referred to as *clearance*. Mine hunting occurs at multiple *depths*, as mines may be laid on the bottom, or *moored* throughout the water column.

**Sweeping** is an activity that uses specialized apparatus to destroy all mines present in a given area. This activity is subdivided into different types of activity based on different apparatus that must be used to eliminate specific mines. *Mechanical sweeping* eliminates moored mines by cutting the chains that connect them to the ocean floor. *Magnetic and acoustic sweeping* employ signal generators which mimic the magnetic and acoustic signatures, respectively, of ships, to trigger mines that are activated by those signatures.

These tasks must be assigned to the various available units, which may include helicopters, surface ships, and autonomous vehicles, in such a way as to ensure that a very low probability remains of any mine existing of the types described in the enemy course of action.

**Example MCM Mission**

As an illustration of the size and the complexity of these problems, consider the following scenario. The operational objective is to clear a staging area 15 nautical miles by 16 nautical miles in size. Enemy course of action analysis indicates the possible presence of influence mines on the sea bottom and contact mines near the water surface. Available resources include four helicopters and two surface ships, all of which can be outfitted with equipment for hunting and/or sweeping various mine threats. As the near-surface mines are a danger to some of our sea surface-based search vehicles, any area they enter must first be swept of these mines. The initial schedule includes the following tasks:

- 41 sweep tasks apportioned among three helicopters for near-surface moored contact mines, each to be performed for 4 hours, during the first week.
- 43 tasks apportioned among the two surface ships, to hunt for and neutralize bottom influence mines, starting on day two and continuing for four weeks.
- 14 tasks apportioned among the three helicopters to search for bottom influence mines.
- 14 tasks apportioned among the three helicopters to analyze the search results.
- 14 tasks apportioned among the three helicopters to reacquire all contacts found during search, identify them, and neutralize them if they prove to be mines.

This schedule must be repeatedly adjusted over the course of the operation in response to unexpected events which invalidate it. For example, a breakdown of one of the helicopters (which is reportedly frequent) is an unexpected event that may necessitate reassignment of its current or next scheduled task, and/or a delay in the schedule. A storm may delay all tasks for hours or days. Discovery of an unexpected mine type may require the addition of many new hunt and/or sweep tasks to neutralize them. Constant monitoring is necessary to identify and respond to these problems quickly. The task of keeping the schedule up to date despite hundreds of interrelated tasks is complex, difficult, and laborious, particularly given the constant time pressure of typical operations. Modifications to schedules are kept to a minimum, in order to reduce expense and opportunities for error; we refer to this characteristic as *minimal disruption*. However, modified schedules must also fulfill operational requirements such as percent clearance, time limits, and risk to equipment.

These difficult tasks (i.e., monitoring, response, and rescheduling) can be greatly aided by new computational tools. Our new system, CARPE, aims to reduce the burden on MCM planning staff by providing decision aids that reduce these tasks’ complexity and assist in their resolution.
3. CARPE

CARPE is a decision support system for MCM personnel that supports continuous situation assessment, analysis, and rescheduling. CARPE is designed to interoperate with existing deployed systems, and principally the Mine Warfare and Environmental Decision Aids Library (MEDAL), a standard tactical decision aid (Pollitt, 2006), as a complete solution to help keep MCM operations on track and improve overall mission performance. CARPE’s user interface provides access to MCM mission data in a user-friendly Common Operational Picture (COP) for a human operator to view, as well as an alert generation and guided response capability through which CARPE’s intelligent services assist the user in responding to developing situations through assumptions and rescheduling.

3.1 System Architecture

Figure 2 shows the CARPE architecture designed to interoperate with MEDAL. It depicts the services provided by both CARPE and MEDAL as an integrated system for scheduling, monitoring, troubleshooting, and visualizing MCM missions. The CARPE architecture comprises three categories of components:

User Interface: These components allow users to interact with the CARPE system, receive important updates, and visualize the status of an ongoing mission. This includes displays of disruption alerts (e.g. worsening weather conditions or loss of communications with a task group), views for root cause analysis, management of assumptions made, and schedule editing and selection windows. MEDAL provides user interface components for entering mission data (e.g. the starting schedule and bathymetry data) and updating situation data.

Services: These reasoning components automate and assist MCM operations staff with monitoring operational progress and disruptions, analyzing problems, and rescheduling. CARPE services include (1) the Discrepancy Recognizer, which monitors incoming situation data in real time to detect discrepancies; (2) the Root Cause Analyzer, which forms hypotheses about the underlying reasons for discrepancies and identifies test questions and observations that an operator can use to gather evidence for confirming or refuting cause hypotheses; and (3) the Rescheduler, which is responsible for creating revised schedules that compensate for changes in the environment and/or new assumptions. MEDAL services publish situation data and mission data that CARPE accesses.

Memory: These data components comprise an internal store of information about the ongoing mission, as well as knowledge necessary to perform reasoning activities. CARPE’s memory stores models for detecting discrepancies, analyzing root causes and modifying schedules, whereas MEDAL stores data about the situation and MCM mission, in addition to knowledge artifacts representing expert knowledge about the MCM domain.
3.2 Decision Support Session: Automated Alerts & Guided Response

CARPE continually monitors the execution of a mission for discrepancies and, when they arise, alerts the user, helps the user to analyze the situation, and interactively modifies the schedule in response. This interaction, the primary use case of CARPE, proceeds as follows:

1. **Automated Monitoring & Alert Presentation:** CARPE continually monitors for problems that may impact an ongoing operation. When monitoring detects a discrepancy between mission expectations and incoming observations that may suggest such a problem, it alerts operators with a message. For example, CARPE might detect that the number of mine-like sensor contacts is lower than expected based on prior surveys of the area.

2. **Conversational Root Cause Analysis:** In this step a user interactively identifies a root cause for the alerted discrepancy by answering a series of questions posed by the root cause analyzer. The system displays a list of possible root causes and a set of tests or observations that could be made to reduce the set of root causes. Each root cause corresponds to a possible underlying problem or incorrect assumption that may have caused the discrepancy to occur; positively identifying this problem is necessary to fix it. For example, a low number of mine-like sensor contacts may be caused by a lowered probability of detection. The user responds to questions until all but a single cause are eliminated. The confirmed root cause also indicates a new assumption about the world that will be used henceforth in schedule repairs.

3. **Repaired Schedules Recommendation:** Based on the confirmed root cause, CARPE automatically generates alternative versions of repaired active schedules and recommends them to user. These schedules are ordered based on computed disruptiveness metrics, including the number and type of changes made. As this ordering is difficult to assess, human mission planners provide the final decision of which alternatives to proceed with.

4. **Interactive Schedule Editing:** The user can view the recommended schedules and manually edit them prior to enacting them (see Figure 3). They may choose to remove or add new tasks to the schedule, edit the attributes of an existing tasks, or reassign tasks to another unit. CARPE automatically checks the edited schedule for conflicts (e.g., a single unit is assigned multiple simultaneous tasks).

5. **Return to Execution:** Once a new schedule is confirmed, CARPE activates it and transmits new task orders to the task groups involved in the mission.

The following section describes the rescheduling algorithm in further detail.

![Figure 3. CARPE schedule display](image-url)
4. MINIMALLY DISRUPTIVE SCHEDULE REPAIR

CARPE automatically repairs an existing schedule as part of the “Repaired Schedule Recommendation” step described in section 3. This schedule repair procedure alters a schedule to work around one or more changes in the world that impacts its efficiency or effectiveness. Schedule repair takes as input an original schedule with known flaws, constraints that a resulting schedule must meet, and a representation of the goals to be achieved by executing the schedule; as output, it returns alternative repaired schedules that achieve the same goals and satisfy the constraints. Such an algorithm may also make use of encoded expert knowledge about the domain.

One commonly used repair technique is to compute an entire schedule that meets the original goals. While research in automated planning and scheduling has produced many feasible and efficient algorithms for accomplishing such a task, the results may be suboptimal for schedule repair, because they ignore the prior schedule. In keeping with the principle of minimum disruption, MCM planners don’t want to make unnecessary large and impractical changes to ongoing operations which may be difficult to verify and explain to their commanders. Creating a completely new schedule and disregarding existing operations can be highly disruptive to the ongoing operations; therefore, we use different schedule repair techniques that are minimally disruptive. This implies that the differences between the original and repaired schedules are small or have been reached with a minimum number of edits. In addition, we provide multiple schedule alternatives for a user to consider rather than a single schedule which may be suboptimal due to information unavailable to CARPE.

The algorithm that provides minimally disruptive schedule repair is called Case-Based Local Schedule Repair (CLOSR). CLOSR repairs an existing schedule by applying specialized knowledge about common repair types (see Figure 4). We describe the knowledge representation and the algorithm in detail in the following subsection. Next, we describe the performance requirements for CLOSR in the context of the CARPE system.

![Figure 4. CLOSR algorithm](image)

4.1 CLOSR Knowledge Representation

CLOSR uses two types of knowledge about schedule repair in the application domain (i.e., MCM operations planning): (i) repair goal formulation rules, and (ii) the rescheduling case base. Each goal formulation rule represents knowledge about what must be done to repair a schedule in a certain situation. The applicable mission and situation forms the antecedent of the rule, a logical statement about the state to which the rule applies. The repair goal forms the consequent of the rule. For example, the following goal formulation rule asserts that when a piece of equipment is nonfunctional, a “fix_and_continue” goal should be asserted:
A rescheduling case consists of two parts: problem and solution. The problem is specified as a *repair goal type* and a list of variable *parameters*. The case applies to all repair goals with the specified type; the parameters indicate how the repair can be specialized to a specific schedule and problem instance. The solution of a rescheduling case is specified as a list of *revision tactics*. The tactics contain references to the parameter variables, and are performed in order during schedule adaptation. Table 1 shows a rescheduling case that handles the “fix_and_continue” goal by adding a new fix task to the schedule, splitting its current task, and inserting the new fix task immediately, so that the modified schedule requires the fixed unit to continue its current task once repair is completed.

**Table 1: CLOSR rescheduling case example**

<table>
<thead>
<tr>
<th>Repair Goal</th>
<th>fix_and_continue</th>
<th>Parameters</th>
<th>?unit, ?time-to-fix</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Revision Tactics</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AddTask (type = Fix, performer = ?unit, duration = ?time-to-fix)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SplitInProgressTask (performer = ?unit)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>InsertTaskImmediately (type = Fix)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FixOverlaps ()</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**4.2 CLOSR Algorithm**

CLOSR performs the following steps to find correct repairs:

1. **Formulate Repair Goals**: To find appropriate repair goals, CARPE finds all consequences of the goal formulation rules given the current situation data and mission parameters.
2. **Retrieve Repair Case(s)**: To find cases that will resolve a particular repair goal, CLOSR searches through the list of known rescheduling cases to find all cases that specify a matching repair goal type. Each such case provides a candidate schedule repair procedure.
3. **Apply Repair Solutions**: To repair a schedule, the parameter values indicated by a specific repair goal are substituted for the variable parameters specified by an individual case problem throughout that case’s revision tactics. Then, each tactic is applied in order to the original schedule by executing a tactic procedure that maps to that tactic. Table 2 gives a partial list of the tactics that CLOSR recognizes and can execute.
4. **Resolve Conflicts**: Finally, the resulting schedule must be sanitized to resolve any problems introduced by the addition and removal of tasks. This includes resolving issues where a resource is double-booked as well as domain-specific precedences, such as those that require an area to be swept of certain mine types before a ship can move in. All conflicts are resolved by either moving the start time of an illegal task to the earliest time when it could legally occur, or exchanging the start times of two tasks performed by the same unit which must occur in the reverse order.

**Table 2. Example CLOSR Repair Tactics**

<table>
<thead>
<tr>
<th>Tactic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AddTask</td>
<td>Creates a task that did not exist in schedule previously</td>
</tr>
<tr>
<td>InsertTaskImmediately, InsertTaskAfterCurrentTask, InsertTaskAfter, ...</td>
<td>Inserts an added task at a relative position in the schedule</td>
</tr>
<tr>
<td>ReassignTasks</td>
<td>Reassigns all tasks from a removed unit to another unit of the same type</td>
</tr>
<tr>
<td>SplitInProgressTask</td>
<td>Splits a task into two separate pieces to allow later resumption of an activity</td>
</tr>
</tbody>
</table>
When multiple repair goals and/or repair cases occur, steps 3 and 4 are repeated for each, repairing the initial schedule in several different ways. This results in alternative schedules that a user can choose among to resolve an issue.

4.5 CLOSR Functional Performance Requirements in CARPE

To meet CARPE decision support objectives, CLOSR must meet the following functional requirements: (1) resolve flaws (i.e., invalidated elements) in existing schedules, (2) result in minimal operational disruption, (3) provide alternative repaired schedules, (4) allow end-users to author knowledge and (5) generate human readable and easy to understand descriptions of the schedule changes.

To meet the first requirement, it is necessary to provide accurate rescheduling cases to CARPE; experimental evidence (see Section 5) indicates that with sufficient knowledge, CLOSR can successfully repair invalidated schedules. The second requirement of minimal disruption is met by executing only the specific changes enumerated in a case’s repair tactics. Such knowledge-guided repairs only introduce change that are deemed strictly necessary by an expert. CLOSR meets the third requirement by investigating multiple repair solutions in parallel, resulting in alternative schedules to be presented to the planners. We are yet to develop approaches to meet the fourth requirement. However, we believe that the procedural knowledge encoded in case tactics for CLOSR is intuitive and easy to provide for subject matter experts. Finally, the tactics inherently provide a means of explaining exactly how a schedule was repaired. This is accomplished by generating a plain English description using parameterized templates associated with each tactic applied to repair a schedule. Unlike CLOSR, a plan generation algorithm that generates entire schedules cannot easily meet the essential CARPE function requirements 2, 4, and 5.

5. Evaluation

5.1 Study Objectives

We hypothesize that the scheduling capability of CLOSR is sufficient to repair an original schedule such that it achieves stated objectives despite the occurrence of disruptive events. To demonstrate this, we ran the CARPE system in an automated manner on a series of simulated MCM operations that are interrupted by events. We measured and compared the performances of two decision makers: (1) a decision maker that ignores all alerts and keeps the original schedule, and (2) a decision maker that chooses a random schedule recommended by CLOSR; comparison indicates the performance improvement that can be achieved by adopting the recommendations made by the CLOSR system without end user edits.

Typical performance objectives in an MCM mission require that all mines be neutralized, and that search attains a 95 percent clearance, meaning that there is a 95% chance that a mine at any given point in the search area would be observed if it existed. We hypothesize that the decision maker using CLOSR will achieve these performance objectives, and that the decision maker that does not reschedule will not. This will demonstrate both that rescheduling is necessary to achieve an acceptable level of performance under simulated conditions, and that CLOSR is sufficient to achieve that performance.

5.2 Experimental Framework

A simulator for MCM operations, Search and Coverage SIMulator (SCSim), was developed at Knexus Research Corporation to support rapid and repeated evaluation and testing of MCM decision support systems and component algorithms. SCSim simulates search missions involving multiple heterogeneous search units, including ships and helicopters, each with different available equipment configurations. Mines and mine-like objects are distributed randomly by SCSim in fields and lines according to pre-set distributions with variable density and object counts. This facilitates evaluation of algorithm performance under varying operating conditions. As a benchmark, automated testing of a two month operation takes less than one minute.
SCSim simulates the assignment of parameterized tasks to units according to a schedule, including transit, sweep, and hunt tasks. Task parameters include, for example, the equipment to use for sweeping, and sensor depth for hunting. To simulate a mission, SCSim automatically generates appropriate tracks for each task and simultaneously changes the position of each vehicle along its assigned tracks. Observations (e.g., contacts) are generated based on vehicle’s positions and the sensor equipment in use. Interactions of deployed sweeping equipment is also simulated, and changes the internally represented status of mines. In addition to the scheduled tasks, SCSim is responsible for simulating random events, the unexpected difficulties that invalidate an existing schedule. Examples of such events include equipment failure, bad weather, and operator errors.

An individual mission test using SCSim is controlled by a scenario description. Scenario descriptions include, at a minimum, the vehicles and equipment available for use, threat areas to be cleared of sea mines, and task areas where vehicles will operate. Other elements of the scenario specify random distributions for mine like objects, mine line placements, and events that may occur.

To mimic the real world as closely as possible, SCSim provides only partial observations for the purposes of rescheduling. For example, when a helicopter’s communications system fails, its position is no longer reported to the system. As a result, the helicopter appears not to move.

Experiments in CARPE are driven by a test harness that integrates with SCSim. The test harness takes scenario parameters as input, which specify the area of operations, available assets, and the ranges of random experimental variables, such as what mine types will be deployed and when events will trigger. The Test Generator uses these parameters to generate a set of random scenarios. The Test Runner enacts each scenario by initializing SCSim and an appropriate Decision Maker that acts as a user of the system. Each decision maker encodes different responses to situations, such as alerts, that arise during the mission simulation. To compare the performance of different decision makers, every decision maker is run through the same pool of randomly generated scenarios. As each simulated mission completes, metrics are collected and recorded. After all simulated missions are complete, the Performance Evaluator tabulates and summarizes these results in a human readable form.

Figure 5. CLOSR simulation driven evaluation

5.3. Experiment Setup
Our experiments used two decision makers and ten randomly generated test scenarios. The first decision maker, “CLOSR DM”, confirms the correct root cause and selects a new schedule at random from those generated by
CLOSR to activate. The second decision maker, our baseline, “Ignore DM”, ignores CARPE’s recommendations, never changing its schedule when prompted. Comparing performance of these two decision makers allows us to measure the efficacy and correctness of schedules generated by the CLOSR system.

The performance of each decision maker was evaluated in each of the ten randomly generated scenarios, generated as summarized in Table 3. Scenarios differ primarily in the random events that occur, and the positions of mines and mine-like objects. 30 events occurred in each scenario, and were generated randomly according to the distribution in Table 4. Each event was additionally parameterized with a trigger time (chosen randomly over the first 600 hours of the mission) and target unit (chosen randomly among the 6 tasked assets). The times were chosen in this fashion because events that occur when a unit has already performed all its tasks cause no problems, and therefore are uninteresting to our study. Non-mine mine-like bottom objects (NOMBOs) were generated uniformly throughout the threat area in each scenario. 3 bottom influence mine lines along with 1 mine line containing near-surface contact mines were placed randomly in each scenario, each with a length (i.e., mine count) between ten and thirty. In addition a fifth mine line of magnetic mines was generated on the surface 50 percent of the time with the same mine count distribution.

<table>
<thead>
<tr>
<th>Attributes</th>
<th>Value Ranges</th>
<th>Variation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Mission/Op Type</strong></td>
<td>Amphibious Assault</td>
<td>Fixed</td>
</tr>
<tr>
<td><strong>OP Area Size and Type</strong></td>
<td>SEA area (15 x 16nm)</td>
<td>Fixed</td>
</tr>
<tr>
<td><strong>Enemy Order of Battle</strong></td>
<td>3 bottom influence mine lines, 1 surface contact mine line, 0-1 surface magnetic mines mine lines in SEA</td>
<td>Random location and number</td>
</tr>
<tr>
<td><strong>Commanders order of Battle</strong></td>
<td>4 MH-53Es (Helicopters) 2 SMCM (Ships), 1 support ship (command ship and divers)</td>
<td>Fixed</td>
</tr>
<tr>
<td><strong>Bottom Types</strong></td>
<td>A1, B1, B2</td>
<td>Random contact rates</td>
</tr>
<tr>
<td><strong>Events</strong></td>
<td>Adjustments per equipment, sensor malfunction, communications loss, disabled vessel, weather</td>
<td>Random targets and timing</td>
</tr>
</tbody>
</table>

The fixed parameters used in all scenarios included the area searched, and 7 assets, consisting of 4 helicopters, 2 MCM ships, and 1 support ship that performs no tasks itself. Each ship has available equipment for hunting mines, and helicopters have equipment used for magnetic sweeping, contact sweeping, detection, and mine neutralization.

### Table 3. Test scenario parameters

<table>
<thead>
<tr>
<th>Event</th>
<th>Description</th>
<th>Chance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Communication Failure</td>
<td>Unit loses ability to communicate with operator</td>
<td>48.8%</td>
</tr>
<tr>
<td>Incapacitation</td>
<td>Unit breaks down and can not complete tasks</td>
<td>2.4%</td>
</tr>
<tr>
<td>Wrong Assigned Location</td>
<td>Unit executes task in an incorrect (adjacent) area</td>
<td>48.8%</td>
</tr>
</tbody>
</table>

5.4. Metrics

We evaluated CLOSR DM and Ignore DM using the following three metrics:

1. **Percent contacts detected**: This measures the percentage of mines detected by a unit
2. **Percent mines neutralized**: Percentage all mines mines neutralized by a unit.
3. **Operation duration**: Total simulation time required to complete the operation
The first two metrics are calculated based on the true number of mines and mine-like objects generated in the scenario. These summarize the plan’s effectiveness in terms of how well the MCM mission goal of searching for and eliminating mines was achieved. Each scenario generated includes a large number of non-mine mine-like objects uniformly spread throughout the threat area, so the percent contacts detected value is an approximation of the percent clearance, or probability that a mine would be detected at any given location. The third metric, operation duration, illustrates a plan’s efficiency by measuring the total simulation time required to complete all tasks.

5.5. Results

Experiments were run on a laptop with an i7 processor, which took about one hour to complete. The results of our experiments are summarized below. Figure 6 shows a scatter plot that displays the percent of existing contacts that were detected and duration of each mission operation. It’s clear that the duration of an operation performed by Ignore DM varies little, as the original schedule is never updated; in contrast, the time taken by CLOSR DM varies greatly. A schedule can be greatly lengthened when a vehicle breaks down, or a new mine type is discovered, requiring many additional hunts and/or sweeps. The increased time, however, enables CLOSR DM to consistently outperform Ignore DM by detecting between 95 and 100% of the mine like objects in every mission.

Table 5 shows the average and standard deviation for each metric and decision maker, along with a confidence value. These confidence values are obtained from a one-tailed t-test with paired samples, and indicate the (small) likelihood that Ignore DM might on average achieve higher values than CLOSR DM if many more experiments were undertaken. From the test we can see that CLOSR DM’s results are significantly higher than Ignore DM’s in all three metrics. In every scenario CLOSR DM consistently neutralized every mine, while the base agent only neutralized 76% on average. Similar results can be seen for contacts detected: CLOSR DM found almost 98% on average, while Ignore DM found only 85%. On average, CLOSR DM took 34% longer to complete a mission.

![Figure 6. Scatter Plot of Operation Duration to Percent Contacts Detected](image)

<table>
<thead>
<tr>
<th>Metric</th>
<th>CLOSR DM</th>
<th>Ignore DM</th>
<th>P-Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Percent Contacts Detected</td>
<td>97.6% (±1.6%)</td>
<td>84.8% (±8.4%)</td>
<td>0.0005</td>
</tr>
<tr>
<td>Percent Mines Neutralized</td>
<td>100% (±0%)</td>
<td>75.9% (±18.6%)</td>
<td>0.0013</td>
</tr>
<tr>
<td>Operation Duration</td>
<td>919.8 (±218.8)</td>
<td>685.5 (±5)</td>
<td>0.0040</td>
</tr>
</tbody>
</table>
In a successful mission, we would expect to find 95% of the mine-like contacts and 100% of the mines. Our results support our hypothesis, showing that use of CLOSR reschedules can be the difference between mission success and mission failure. While response time was not precisely measured, observations indicate sub-second turnaround times for schedule repair.

6. RELATED WORK & DISCUSSION

To give some perspective on the task of automated schedule repair, we review other work in the field of modifying existing plans and schedules. We limit our review only to closely related work on plan and schedule repair and exclude the much larger body of work on automated planning and scheduling based on an initial state and goal, because it is not directly applicable to our problem.

A number of studies have treated plan repair or schedule repair as a search problem, where a model of the problem is given, but no example solutions. The GPG (Blum and Furst, 1997) and POPR (van der Krogt and de Weerdt, 2005) replanners reduce the plan repair problem to a planning problem by searching the space of subplans found by iteratively removing actions from the original plan, then performing a standard model-based planning search forward from each resulting plan. Some scheduling techniques search through a constraint space. For example, Sakkout and Wallace (2000) modelled schedule repair as a linear programming problem, and searched through a space of problem constraints to find a set sufficiently constrained so as to exclude all non-solutions, then applied a standard solver. Wang (2005) applied both a genetic algorithm and a simulated annealing search to product development scheduling, modeled as a constraint satisfaction problem. In each case, no tasks were added during plan repair, but tasks could be reassigned and start times changed. In general, these search techniques are inefficient compared to CLOSR, which uses a priori knowledge to solve problems in linear time. However, CLOSR does not share their flexibility, in that they can solve any problem in their model’s search space, whereas CLOSR can solve only pre-specified problems.

Rather than performing search, O-Plan (Wang and Chien, 1997) simply recognizes known failures and inserts a pre-constructed repair subplan into the current plan to resolve a failure. This is similar to CLOSR, but not as flexible, as the subplans cannot be subsequently adapted.

Case-based reasoning (Aamodt and Plaza, 1994; Richter and Weber, 2013) is a family of intelligent algorithms based on the adaptation and application of known solutions to new problems. It has been applied to many different domains and problems besides repair of plans and schedules. Two related systems that use case-based techniques to modify an existing plan or schedule are the CHEF (Hammond, 1986) system, which specializes in construction of recipes, and the Darmok system (Sughandh et al, 2008), which acts as a computer opponent in a real-time strategy game. Darmok adapts plans to new situations by fulfilling subgoals of an existing plan on demand, based on known plans that solve those goals. CHEF is more similar, in that cases specify “repair strategies” which describe how to fix a problem in a recipe. Neither of these systems, however, addresses the challenges of use of constrained resources present in a typical scheduling problem.

Case-based solutions to the resource-constrained schedule repair problem include the CABINS system (Miyashita and Sycara, 1995), and Petrovic et al’s nurse rostering system CABAROST (2003). CABINS does not work with faulty schedules, but rather unoptimized schedules. As such, the cases respond to global schedule heuristics, such as the tardiness and work-in-process inventory of the schedule. CLOSER, in contrast, responds when a schedule has become untenable. CABAROST resolves faulty schedules in a manner similar to CLOSR, but cannot introduce or change start times of tasks to resolve problems, only reassign resources.

To our knowledge, schedule repair techniques have not previously been applied to MCM schedules. The tactical decision aid Commander’s Estimate of the Situation interactively assists in the creation of an original schedule, but does not aid in repair (Garcia and Wettergren, 2012). Track spacing (e.g., Williams, 2010) and path planning (e.g., Piatko et al, 2001) for mine countermeasures have also been studied; these techniques determine paths for individual
units performing MCM tasks, but provide no guidance regarding the order in which tasks should be performed. PATHA (Percival and Stoddard, 2010), goes much further, calculating optimal taskings for heterogeneous agents in a large area. However, PATHA does not help to repair the schedule in response to problems that occur during a mission.

The capability to repair MCM schedules automatically is therefore new. CLOSR begins to fulfill this need, as evidenced by the measured increase in performance from following its schedules. However, this work is still preliminary. Future evaluations should include the simulation of additional events, and integrate However, CLOSR currently has no ability to search for modified schedules that resolve issues not encoded in the case base, as do search-based replanners such as POPR and GPG. Therefore, in addition to adding cases that resolve additional issues, a future version of CLOSR should include a search capability, to be used when a solution is not available in the case base.

7. CONCLUSION

MCM operation scheduling is challenging due to the complexities resulting from a large number of tasks that must be allocated over numerous resources. This complexity and decision making difficulty are further compounded by environmental uncertainty that routinely invalidate schedules. In this paper, we presented a decision support system called CARPE that includes components that assist operation planners by constantly monitoring the environment for changes, assisting in analysis of discrepancies, and recommending alternative repaired schedules. We introduced the requirement of minimally disruptive repair as a key operational requirement for automatic schedule repair algorithms in MCM applications.

We presented CLOSR, a novel schedule repair algorithm included in CARPE, which automatically proposes alternative repaired schedules that satisfy the requirement of minimal operational disruption. We evaluated the performance of CLOSR in a simulated MCM operations scenario and demonstrated that its proposed repairs indeed improve operational effectiveness. Our results indicate the efficacy of a case-based strategy; schedule repair was rapid, and created new schedules on demand that ensured the elimination of all mines and increased clearance to a reasonable level. This presents a novel and measurable increase in automated MCM rescheduling capabilities.

Our future studies will examine the efficacy of our root cause analysis system, as well as measuring user response to root cause analyses, schedules created by CLOSR, and other user interface features. We will extend CLOSR with the capability to respond to additional events, acquire rescheduling experiences from expert users, and search for admissible reschedules as back-up when no cases are available.

8. ACKNOWLEDGEMENTS

We thank the Office of Naval Research, Code 321, for supporting this research. We also thank Joe Kann, Micah Heineck, and Kishan Gajera of Knexus Research Corporation for their contributions to the development and testing of the CARPE system and the CLOSR algorithm.

9. REFERENCES


