**Naval Research Laboratory**
Washington, DC 20375-5320

**NRL/7540/MR—2022/1**

# Inference and Optimization in Navy Planning Systems

DAVID SIDOTI

*Decision Systems Section*
*Marine Meteorology Division*

May 17, 2022

# REPORT DOCUMENTATION PAGE

*Form Approved*
*OMB No. 0704-0188*



| **1. REPORT DATE** *(DD-MM-YYYY)* | **2. REPORT TYPE** | **3. DATES COVERED** *(From - To)* |
|---|---|---|
| 17-05-2022 | NRL Memorandum Report | 9/17/2018 – 9/17/2020 |

**4. TITLE AND SUBTITLE**

Inference and Optimization in Navy Planning Systems

**5a. CONTRACT NUMBER**

**5b. GRANT NUMBER**

**5c. PROGRAM ELEMENT NUMBER**
NISE

**6. AUTHOR(S)**

David Sidoti

**5d. PROJECT NUMBER**

**5e. TASK NUMBER**

**5f. WORK UNIT NUMBER**
N2R2

**7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**

Naval Research Laboratory
4555 Overlook Avenue, SW
Washington, DC 20375-5320

**8. PERFORMING ORGANIZATION REPORT NUMBER**

NRL/7540/MR--2022/1

**9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)**

Naval Research Laboratory
4555 Overlook Avenue, SW
Washington, DC 20375-5320

**10. SPONSOR / MONITOR'S ACRONYM(S)**
NRL-NISE

**11. SPONSOR / MONITOR'S REPORT NUMBER(S)**

**12. DISTRIBUTION / AVAILABILITY STATEMENT**

**DISTRIBUTION STATEMENT A:** Approved for public release; distribution is unlimited.

**13. SUPPLEMENTARY NOTES**
Karles Fellowship

**14. ABSTRACT**

In the proposed research, we develop software and concomitant algorithms for intelligent, proactive, and automated decision support in Navy planning systems. The algorithms act as the conduit for course of action recommendations for multi-domain warfare battlespace management. Additionally, we investigate Human-Artificial Intelligence symbiosis to enhance the algorithms with efficient hybrid human-machine effort allocation for maximized alignment of the decision with the Commander's intent, human preferences, and dynamic context.

**15. SUBJECT TERMS**

Nonlinear program
Operations research
Submarine waterspace management

| **16. SECURITY CLASSIFICATION OF:** | | | **17. LIMITATION OF ABSTRACT** | **18. NUMBER OF PAGES** | **19a. NAME OF RESPONSIBLE PERSON** |
|---|---|---|---|---|---|
| | | | | | David Sidoti |
| **a. REPORT** | **b. ABSTRACT** | **c. THIS PAGE** | U | 12 | **19b. TELEPHONE NUMBER** *(include area code)* |
| U | U | U | | | (831) 656-4622 |

This page intentionally left blank.

# Inference and Optimization in Navy Planning Systems

David Sidoti

## Abstract

In the proposed research, we develop software and concomitant algorithms for intelligent, proactive, and automated decision support in Navy planning systems. The algorithms act as the conduit for course of action recommendations for multi-domain warfare battlespace management. Additionally, we investigate Human-Artificial Intelligence symbiosis to enhance the algorithms with efficient hybrid human-machine effort allocation for maximized alignment of the decision with the Commander's intent, human preferences, and dynamic context.

## Introduction

Manpower is the literal driving force behind the Navy's day-to-day operations. Developing and implementing tactical decision aids that enhance manpower efficiency allows those in uniform to do their job faster and smarter. The primary gaps targeted for this research was that of collision avoidance in submarine waterspace operations.

## Waterspace Management

The conflict identification tool, herein referred to as CONFIDENT, aims to automatically perform prevention of mutual interference (PMI) in both space and time across any or all of the Submarine Operating Authorities' (SUBOPAUTHs) waterspaces. The tool was developed with the intent of aiding officers in understanding evolving waterspaces and to lighten the burden and work load of prevention of mutual interference operations.

### Background

The legacy Global Command and Control System (GCCS) - M will be obsolete in the near- to mid-term future. WaSP aims to seamlessly transition the watchfloor from this legacy system to a more modernized architecture and framework to provide SUBOPAUTH planning capability and to eventually automate the generation and ingestion of MPA-relevant messages. Naval Research Laboratory, Monterey (NRL-MRY) and the University of Connecticut (UCONN) have collaborated to create CONFIDENT and make it practical, fast, efficient, and compatible with the WaSP architecture and SUBOPAUTH CONOPs.

CONFIDENT employs hyperefficient datastructures for structured shape and track information, utilizing R*-Trees to organize all submarine assignments on a global scale and in a timely manner. The tool additionally employs nonlinear programming, exploiting moving haven geometry to quickly and accurately identify possible mutual interference for moving submarines in both space and time. The tool solves a generalized geometric problem, applicable across multiple domains. CONFIDENT comprises a two-phase approach to submarine de-confliction operations: 1) Broad Phase, and 2) Narrow Phase. In the broad phase, minimum bounding boxes (MBBs) are compared for all relevant assignments to

Manuscript approved May 6, 2022.

identify overlap or possible overlap between submarines/exercises. As a consequence of simplifying assignments to MBBs, there is a possibility of false positives that may arise, hence, the output of the broad phase if fed as input into the Narrow Phase, where these false alarms are eliminated and true positives are returned. The narrow phase comprises a more in-depth look at each assignment, and through a line-sweeping algorithm, is able to quickly and accurately identify if there truly exists interference between assignments (evensofar as determining whether there is a border conflict, time conflict, depth conflict, etc.). The narrow phase also contains a unique and novel algorithm to determine if one or more moving objects collide with other static or mobile objects via nonlinear programming.

In the context of submarine operations, an "assignment" refers to an *n*-point polygon that may move in space and/or time, wherein a submarine is allocated to and must reside for a specified duration. In alignment with the Navy doctrine and standards, a submarine may never traverse outside of its assigned boundaries. We term the world's oceans as "waterspace" and the submarine allocations therein as "assignments" or "waterspace assignments."

## Stationary Assignments

The first type of assignments we choose to describe are those that do *not* move in time. Stationary waterspace assignments are assumed to be valid[1] polygons (convex or nonconvex) that are temporal only in that they have an associated start time and end time. For all assignments, indexed by, index by *i,* we assume the start time $t_s^i$, and end time $t_e^i$ such that $t_s^i \leq t_e^i \forall_i$. A convex polygon is a simple polygon (not self-intersecting) in which no line segment that connects two points that are part of the boundary ever goes outside of the polygon. In a **convex** polygon, all interior angles (i.e., for each boundary point) must be less than or equal to 180◦. Conversely, **nonconvex** polygons that are used for waterspace assignments do not meet either or both of these conditions – examples include a square with a hole in it, where a hole is a topological structure that notionally prevents a polygon from being shrunk to a single point, or an acute angle (i.e., less than 90◦) followed by a reflex angle (i.e., greater than 180◦ but less than 360◦) as is the case in a 5-point star, which has this alternating combination at each boundary point. Stationary assignments in waterspace planning are referred to as either area assignments or event assignments, depending on the conventions of how the polygons are defined and where. Since the primary difference between area and event assignments are primarily based on conventions/a reference grid, in this document, we cater to the general case of an area assignment, where no underlying grid is necessarily assigned.

## Mobile Assignments

Mobile assignments are those waterspace boundaries that move in space and time. Such assignments are typically rectangles or circles with an associated velocity of intended movement for each boundary point, for a given assumed navigation type (i.e., Great Circle[2]- or rhumb line[3]-based movement). Mobile assignments in waterspace planning are referred to as track assignments.

---

[1] By valid, we mean a shape that has no self-intersections (i.e., "simple"), with a clearly defined boundary (no unconnected segments)

[2] A Great Circle path is one that is the shortest distance between two points lying on the surface of Earth. It is also known as the geodesic distance.

[3] A rhumb line is a path along the surface of Earth with constant bearing as measured with respect to true North.

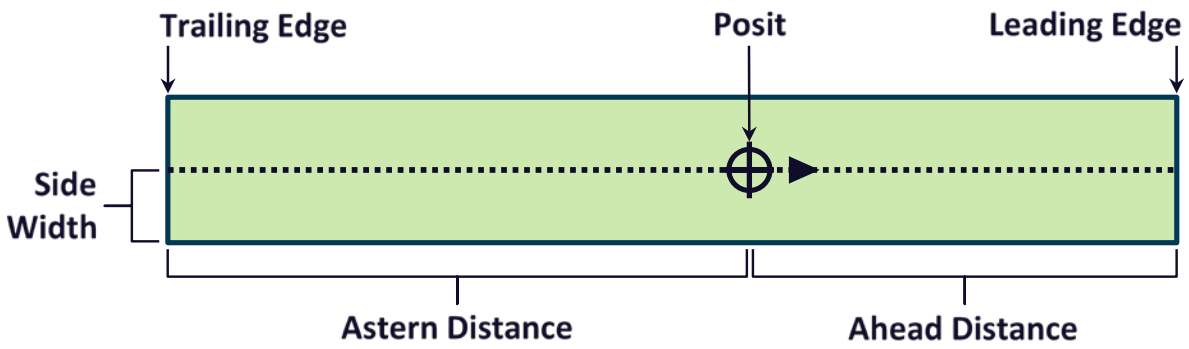


***Figure 1: Moving haven components with posit arrow indicating velocity direction. The moving haven comprises an astern, ahead, and side width, which, when put together, result in a rectangular buffer around a posit (reference point) assuming no intermediate PIMs exist between the leading and trailing edges.***

A track assignment in submarine waterspace planning consists of two or more coordinates (referred to as positions of intended movement, or PIMs) specified by latitude and longitude pairs, each with a corresponding date time group (i.e., year, month, day, hour, minute, and second specification), a corresponding speed, a corresponding shape (and properties), and a corresponding upper/lower depth boundary. Additionally, there is a start and end time ($t_s^i$ and $t_e^i$, respectively) of the overall track assignment that provides information pertaining to when the moving polygon appears and disappears. Assuming assignment *i* has a set of PIMs, indexed by *r*,the time the moving polygon should be centered at the *r*-th PIM is $t_r^i$ , for *r = 1, ..., R*. In modern waterspace planning, the first and last PIMs do not have a time that correspond to the overall start and end time of the track assignment, i.e., $t_s^i \neq t_1^i$, $t_e^i \neq t_R^i$. While this may not be intuitive, it is mathematically derivable and relates directly to the track assignment's shape properties.

## Moving Haven Properties

For each leg of a track assignment, a corresponding shape must be specified. In waterspace planning, a track assignment's shape is more often referred to as a "moving haven." A moving haven has the following properties, visually illustrated in Figure 1:

- Posit – the centerpoint of the overall geometry and the exact point along the track assignment's associated centerline, where a submarine would be assumed to exist if traversing at a constant speed along that leg and constrained to the centerline only. It is the reference center point for the subsequent properties.
- Ahead Width – the total length of the rectangle[4], in nautical miles, in front of the posit.
- Side Width – the total length of the rectangle, in nautical miles, to either side of the posit.
- Astern Width – the total length of the rectangle, in nautical miles, behind the posit.

In addition to these properties, two terms often used when describing a moving haven are the leading edge and the trailing edge. The leading edge corresponds to the front side of the rectangle as it

[4] For all purposes of this document, we will explain the methodology for interference identification with the moving haven's boundaries specified as a rectangle, i.e., four points, however, it can be easily generalized to *n* > 4 points if a circle were to be approximated.

traverses along the track, while the trailing edge corresponds to the back (astern) side of the rectangle as it traverses along the track.

The last characteristic of a track assignment is its activation and deactivation, i.e., when and how the moving haven appears and disappears, respectively. Upon activation, the recently-agreed-upon standard among the submarine operating authorities (SUBOPAUTHs) is that of the moving haven appearing full-sized and prior to the first PIM, with the leading edge centered at the first PIM. It is assumed to have the same bearing/orientation as the first track leg. Conversely, when deactivating, the moving haven traverses through the last PIM, maintaining its same dimensions and the same bearing associated with the ultimate leg, and proceeds until the trailing edge is centered at the last PIM.

Thus, the start and end times of a track assignment may be calculated based on the first PIM's start time $t_s^i$. Denote the ahead and astern widths of the moving haven as $a$ and $b$, respectively. Denoting the velocity of intended movement along the first and last legs as $v_1^i$ and $v_{R-1}^i$, respectively, the activation and deactivation times for a track assignment *i* is

$$t_s^i = t_1^i - \frac{a}{v_1^i} \tag{1}$$

$$t_e^i = t_R^i + \frac{b}{v_{R-1}^i} \tag{2}$$

## Track assignment decomposition given moving haven properties

Any given track assignment typically comprises two core components. When decomposed correctly, a track assignment can be treated as a collection of both stationary and mobile assignments. The first component consists of the track legs (i.e., the straight lines connecting one PIM to the next). At each PIM, however, a moving haven must "turn" and align itself with the next track leg. With regards to the second component, the turning methodology accepted for track assignments in waterspace planning requires the use of stationary circular sectors[5],herein referred to simply as sectors, at each PIM. In a similar manner to other stationary assignments, the sectors have an activation and deactivation time. Upon a leading (trailing) edge reaching an intermediary (i.e., not the start or end) PIM, the sector activates (deactivates) according to the ahead and astern widths; however, since we account for the fact that one track leg may specify different ahead, side, and astern widths from the next, we denote the widths for track leg $r$ as $a_r$, $b_r$, and $c_r$, respectively. For a given intermediary PIM $r > 1$, the turning methodology gives rise to two stationary sector assignments – one accounting for the shape of the sector corresponding to track leg $r$'s side width $c_r$, and one considering track leg $r + 1$'s side width $c_{r+1}$. The second sector activates immediately as the first sector deactivates. Hence, treating the first sector in assignment i as a subassignment *j* and the second activated sector as subassignment $j + 1$, the corresponding activation and deactivation times of the two sectors, are calculated in (3) and (4) for the first activated sector, respectively, and (5) and (6) for the second activated sector, respectively.

[5] A circular sector is a portion of a disk enclosed by two radii and an arc, i.e., a "wedge."

$$t^j_{s,r} = t^i_r - \frac{a_r}{v^i_{r-1}} \tag{3}$$

$$t^j_{e,r} = t^i_r \tag{4}$$

$$t^{j+1}_{s,r} = t^i_r \tag{5}$$

$$t^{j+1}_{e,r} = t^i_r + \frac{b_{r+1}}{v^i_{\cdot}} \tag{6}$$

For assignment *i*, the first and second sectors are spatially constructed given a PIM $r$'s location (longitude and latitude) which acts as the center point, the posit's location (longitude and latitude), and the courses (degrees or radians) associated with traversal along legs $r$ and $r + 1$. Whichever track leg the posit is on determines the radii used to construct the sector. If the posit is on track leg r, the radii used are of length $c_r$, and similarly so if the posit is on track leg $r + 1$. Denoting the course of the track leg $r$ as $\theta_r$, and the course of the (subsequent) track leg $r + 1$ as $\theta_{r+1}$, the first sector is constructed from $\theta_r \rightarrow \theta_{r+1}$ using radii of $c_r$ until time $t^i_r$ , wherein it deactivates and the second sector activates with radii of $c_{r+1}$, and with the same angle range as previously ($\theta_r \rightarrow \theta_{r+1}$), thereafter, until deactivation. See Figure 2 for an illustration of this concept.

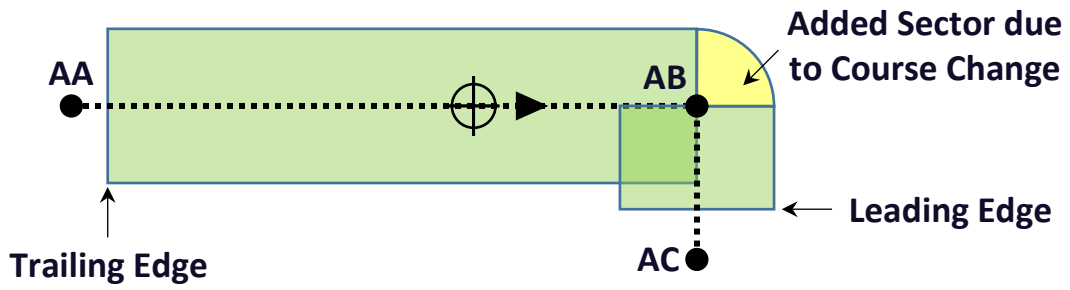


***Figure 2: Moving haven components when an intermediate PIM lies between the leading and trailing edges such that a change in overall course is required upon arrival at the PIM.***

## Prevention of Mutual Interference

Safe waterspace planning requires that no submarine ever come into contact with another submarine. Hence, a submarine must stay within its assigned waterspace between the specified start and end times, while also needing to account for future assignments that it may receive, e.g., move from one polygon to the next, where there is an overlap in start/end times of the two assigned waterspaces so that it can safely traverse from one area to the next. The prevention of collisions between a submarine and any object, whether a known hazard or another submarine, is known as the prevention of mutual interference, or PMI. In the literature, collision avoidance is another name for PMI, hence we use the two terms interchangeably. PMI algorithms were created, developed, and implemented under NISE funding. These algorithms, referred to as CONFIDENT for **Conf**lict **Ident**ication, identify interference (overlap or conflict) for each of the three different fundamental assignment pair possibilities, namely,

1. Area (Stationary) versus area (stationary) assignments
2. Area (Stationary) versus track (mobile) assignments
3. Track (Mobile) versus track (mobile) assignments

## Solution Approach

Our overall method for finding the set of intersecting objects among all these pair possibilities is to use successively more complicated methods to narrow down the set of potentially conflicting objects. The earlier stages (e.g., broad phase purging of undeniably non-intersecting objects) will be faster, but not as selective in removing pairs of objects from the set of potential conflicts in that there may be false alarms. Later stages will be slower, but will have a smaller input set to work with due to the screening work from the earlier stages. The first stage will be to use axis-aligned bounding boxes in an R*-Tree. The second stage is to do polygon intersection with a sweep-line algorithm. The third-stage is to determine the time of intersection by the projection of the intersection vertices, and the fourth-stage is to use nonlinear programming to determine the time of intersection for cases where both objects are time-varying.

The first step is to insert the Axis-Aligned Bounding Boxes (AABB) of all objects into an R*-Tree data structure to query for potential conflicts. The AABB is simply the minimum and maximum longitude and latitude of the polygon. For moving polygons, a bounding box that spans the entire object is used. That is, the left, right, bottom and top of the AABB are defined as

$$L^i = \min_{t,k} p_x^i(k,t)$$

$$R^i = \max_{t,k} p_x^i(k,t)$$

$$B^i = \min_{t,k} p_y^i(k,t)$$

$$T^i = \max_{t,k} p_y^i(k,t)$$

R*-Trees group the AABBs of objects into larger AABBs in a balanced tree structure, which can then be quickly queried to determine if a point is within one of the AABBs. Once all the objects are added to the R-Tree, each object is queried to determine which other objects the AABB conflicts with. The output of this step is a set of pairs of potential conflicts. Because the polygons are represented by AABBs, there is a possibility of false detection of conflicts, but there will not be any missed detections. Because each query for an object returns all the other potential conflicts with that object, this method is very fast at uncovering potential conflicts, and can be used to remove unquestionably non-conflicting objects. This is useful because the dataset is likely to be such that there are many non-conflicting objects and few objects with conflicts.

### Area of Intersection

Upon this filtration of the larger dataset only to the relevant subset, the algorithm commences examining (in a successively complicated manner) pairwise assignment overlap to eliminate false alarms. If both areas are static assignments, this is simply checking for the intersection of the two polygons both in time and space.

If one or both of the objects are dynamic polygons, this intersection operation is not adequate to determine definitively that there is a conflict, but it can provide useful information and will not have any missed detection of a conflict. For these objects, the vertices of the swept area of the track is calculated and used in the intersection algorithm. The swept area is defined as the convex hull of the polygon at the initial and final times. The convex hull of a set of points is defined as the smallest convex set

containing all points. The convex hull operation is trivial for some shapes, such as rectangles. Using this swept area gives us the spatial information of where the conflict could potentially be, but not the temporal information of whether the objects are in the same place at the same time.

To calculate the intersection of two polygons, we used the Python library Shapely, which is based on the C++ library GEOS and the Java Topology Suite (JTS). This algorithm will provide the vertices of the intersection of two polygons. If this intersection is empty, there is no conflict. If it is not empty, we not only learn that there is definitely a conflict, but we also get the additional spatial information of where the overlapping area is. If both polygons are static, this is the final step for determining if a conflict exists. If one or both of the conflicts are dynamic polygons, further computational steps are needed to ensure that a conflict does in fact exist, and the information from the Shapely step is used in the next step.

### Projected Time of Intersection

If *one* of the polygons corresponds with a *mobile* assignment, then a simple projection of vertices is and back-calculation, provided the moving haven dimensions, is all that is needed to determine conflict (if any). Using (7),

$$\rho^{ij}(k) = \frac{(\underline{c}^i(t_f^i) - \underline{c}^i(t_0^i))^T(\underline{a}^{ij}(k) - \underline{c}^i(t_0^i))}{\left\|\underline{c}^i(t_f^i) - \underline{c}^i(t_0^i)\right\|_2^2} \tag{7}$$

If $\rho^{ij}(k)$ is 0, it means that the vertex was projected to the reference point at the start time, if $\rho^{ij}(k)$ is 1, it means the vertex was projected to the reference point at the end time. See Figure 3 for an illustration. Operators are interested in the shape of the moving haven immediately prior to observed overlap with a static object, as well as the shape of the moving haven immediately after an observed overlap with a static object. In order to determine the time of conflict, we use the distance from the reference point to the vertex that is furthest forward and backward in the direction of travel. We aim to then maximize or minimize (7) corresponding to the two operational desires listed previously.

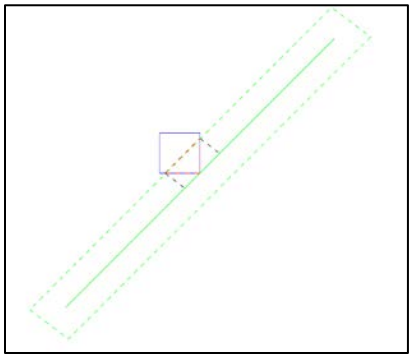

***Figure 3: Projection of the overlapping area's vertices onto the centerline of a track assignment.***

When two mobile assignments comprise the pair, we can invoke (7) twice: once where object *i* is treated as the mobile assignment and object *j* is treated as the static assignment and then once again, vice versa. Doing so allows a conservative guess at narrowing the time bounds for the next (successively complicated) step.

## PMI for Two Mobile Assignments via Nonlinear Programming

If both assignments are mobile, then one last step is required to determine mutual interference. Under this funding, we have developed a novel nonlinear program to find the minimum and maximum time that a convex combination of the vertices of each assignment (polygon) can be found to be the same point. This problem has a linear cost function, but nonlinear constraints. This is due to the location of the vertices varies nonlinearly with time, and also there exists a cross-term between the convex coefficients and the time-varying vertices.

$$\min_{\underline{\alpha},\underline{\beta},t} \quad t \tag{8}$$

$$s.t. \quad \sum_{k=1}^{N_i} \alpha_k \underline{p}^i(k,t) = \sum_{k=1}^{N_j} \beta_k \underline{p}^j(k,t) \tag{8a}$$

$$\sum_{k=1}^{N_i} \alpha_k = 1$$

$$\sum_{k=1}^{N_j} \beta_k = 1$$

$$\alpha_k, \beta_k \geq 0 \; \forall k$$

$$t_{min} \leq t \leq t_{max}$$

The $t_{min}$ and $t_{max}$ limits used in the constraints in (8) are found via the previous section's solution, as detailed wherein each object can be treated temporarily as a static assignment iteratively. This yields conservative bounds that can then be used in (8). It is important to note that if the vertices are moving at a constant velocity, this problem becomes a linear programming problem, by replacing (8a) with

$$\sum_{k=1}^{N_i} \alpha_k \underline{p}^i(k,t_0^i) + \alpha_k v^i t = \sum_{k=1}^{N_j} \beta_k \underline{p}^j\left(k,t_0^j\right) + \beta_k v^j t \tag{9}$$

## Results

A simple demonstration of the results is shown in Figure 4. Here a mobile assignment (zigzag) interferes with a 3-area static assignment consisting of three rectangles. Using the successively complicated proposed framework, we are able to identify the start and end times of mutual interference in three separate instances, each corresponding to one rectangle among the 3-area static assignment.

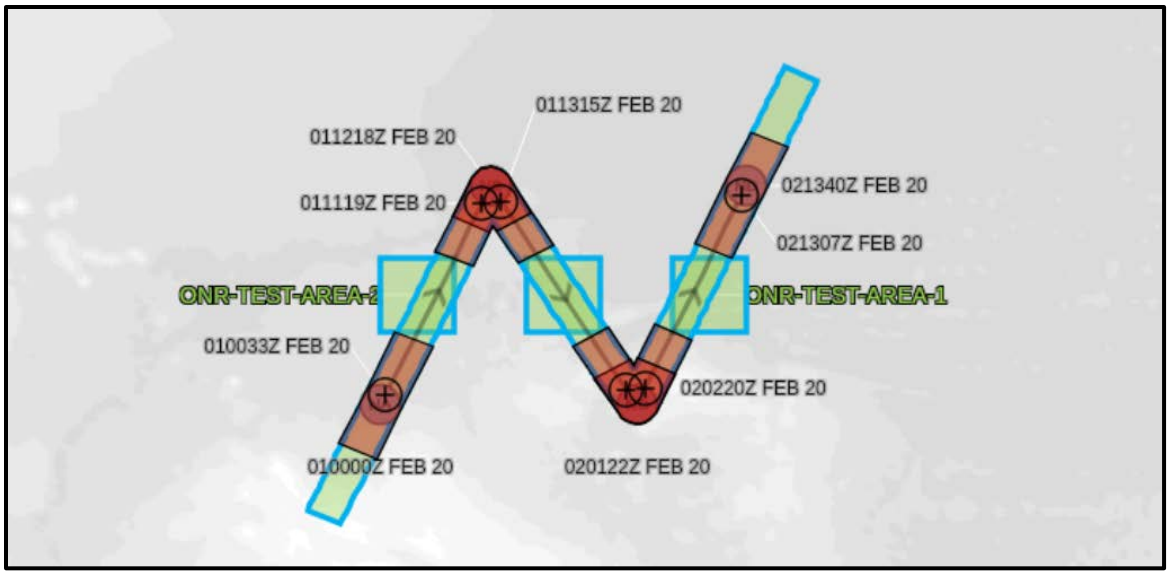


*Figure 4: Example of interference results when examining a pair consisting of one mobile assignment (zigzag) and one static assignment (a multiarea assignment consisting of three separate rectangles).*

## Conclusions

Submarine waterspace is a complicated, fast-evolving problem space with a large number of discontinuities interwoven throughout. We have developed an efficient framework to filter these large complicated waterspace sets down to relevants pairs of assignments to for rapid identification of whether there is a possibility of collision (mutual interference). Towards the prevention of mutual interference (PMI), we have constructed a successively complicated multi-phase approach wherein Axis Aligned Bounding Boxes are utilized in conjunction with R*-trees to generally get pertinent waterspace assignments, relative to a queried (proposed) waterspace assignment. This phase often begets false detections (of mutual interference) and hence requires further inspection with more complex mechanisms for overlap nature/shape determination. Depending on the nature of the pair of assignments, we invoke a line sweeping algorithm to identify spatial overlap, logic operators to verify a temporal overlap, and if a mobile assignment is involved, we make use of these checks to inform more mathematically intense computations. We subsequent (for those pairs of assignments that comprise one or both mobile assignments), utilize simple projection of overlap vertices onto the centerline of the mobile assignment. If both assignments are mobile, we then proceed to the final step where we have developed a novel nonlinear program for collision avoidance, assuming a decomposed waterspace. The findings here are general and more complex derivations are left for future work wherein accuracy is guaranteed on a representative World Geodetic System 1984 (WGS84) ellipsoid to within micrometers. The opinions expressed in this work are solely of the author's.