Bounded Cost HTN Planning for Marine Autonomy

Tony X. Lin[1], Mengxue Hou[1], Catherine R. Edwards[2], Michael Cox[3], Fumin Zhang[1]

***Abstract*—Due to complex oceanic environments, underwater gliders typically must satisfy a variety of environmental conditions in order to complete high level objectives. Underwater navigation, for example, requires that a glider must periodically surface and re-localize in order to ensure adequate progress is being made. Such conditions may be directly encoded in Hierarchical Task Network (HTN) planners to ensure that glider actions are valid over the execution of a plan. However, HTN planners may not be able to find good solutions when actions have uncertain costs, such as when a glider is disturbed by a flow field. We propose a bounded cost HTN planner that leverages a modified potential search method in order to find good navigation plans that satisfy user-defined constraints. Simulation results are presented to validate the approach.**

I. Introduction

Underwater gliders are mobile robotic sensing systems [1] that are able to operate in a wide range of oceanic environments. Due to their high reliability and endurance [2], these platforms have become increasingly popular for persistent surveying missions in which a platform must collect data over weeks to months. Gliders have been widely used in applications such as oil spill surveying [3], acoustic field mapping [4], regional and global ocean observing efforts [5], and improving tropical storm intensity forecasting [6].

The potential for adaptive sampling makes gliders an attractive platform for fish tracking through acoustic methods, and gliders have also demonstrated success in detecting tagged fish by collecting acoustic telemetry data [7], [8]. Acoustic telemetry is a useful tool for understanding habitat use and the daily, seasonal, and long-distance migrations of marine wildlife, which are critical components to the conservation and management of marine fisheries, especially in marine protected areas (MPAs). Accurate knowledge of migration patterns may indicate new MPA locations and determine whether current restrictions on fisheries should be relaxed or tightened. However, fish tracking poses a difficult challenge as uncertainties in the marine environment may make tracking and navigation tasks difficult. While control architectures have been designed to handle a variety of ocean-related control situations [9], [10], each depends on a unique set of environmental assumptions and may cause issues if violated. During a mission, imperfect ballasting, bio-fouling, loss of a wing, or mechanical failures may lead to loss of navigational efficiency. Remora, for example, have been known to latch onto underwater gliders during missions and may immobilize the glider [11]. In these situations, normal fish-tracking controllers will undoubtedly fail as the actuation capabilities of the glider have been significantly changed.

In contrast, cognitive architectures serve as a general method to design intelligent and robust behavior in autonomous systems from a high-level perspective. These approaches seek to model autonomy as a knowledge-based architecture in which models of the system and environment guide decision-making [12]. Previous work has demonstrated the use of cognitive architectures to improve human-robot cooperation [13] and to control a hexapod mobile robot [14] by requesting information as new situations arise. In this paper, we discuss our ongoing efforts to improve autonomous fish tracking operations through the use of an intelligent physical system that leverages a goal-driven cognitive architecture. Through persistent interactions with a marine environment, the intelligent physical system can construct a library of control architectures and their associated conditions for use, allowing the system to robustly handle the dynamic ocean environment.

Two key aspects of using cognitive architectures to achieve underwater autonomy are 1) constructing the abstract and logical description of a set of tasks that include the effects of taking the tasks and the conditions necessary to activate the tasks, and 2) finding a valid sequence of tasks, commonly called a plan, that achieves a desired goal. Planners that solve for the constraints generated by the preconditions for these tasks in order to achieve a target goal are known as Hierarchical Task Network (HTN) planners [15]. In HTN planners, tasks are higher level actions that are composed from primitive actions and/or other tasks that alter Boolean-valued predicates that represent aspects of the agent's state. For example, a task `start_car` can be decomposed into tasks (or primitive actions) `move_into_car`, `insert_key`, and `turn_key` and has the precondition predicates of `near_car`, and `has_key`, and changes the predicates to ¬`near_car`, `_in_car`, ¬`has_key`, and `car_on`. By abstracting lower-level actions or tasks into higher level tasks, an HTN planner is able to construct plans that solve complex goals without having to perform very deep searches. As such, our first objective is in constructing the HTN for marine autonomy. We provide, in this work, the necessary predicates, actions, and tasks to solve navigation goals with localization constraints. Our second objective is to design an algorithm that solves a bounded cost search problem in a HTN when certain transitions have uncertain branch costs. To overcome this difficulty, we use the modified potential search (PTS) method introduced in [16] to solve for the optimal plan. The modified PTS method

[1]Tony X. Lin, Mengxue Hou, and Fumin Zhang are with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA {tlin339, mhou30, fumin}@gatech.edu

[2]Catherine R. Edwards is with the Skidaway Institute of Oceanography, University of Georgia, Savannah, GA 31411, USA catherine.edwards@skio.uga.edu

[3]Michael Cox is with the Wright State Research Institute, Beavercreek, OH 45431, USA, michael.cox@wright.edu

Global Oceans 2020: Singapore - U.S. Gulf Coast | 978-1-7281-5446-6/20/$31.00 ©2020 IEEE | DOI: 10.1109/IEEECONF38699.2020.9389201

introduces a key function, which is an implicit evaluation of the probability that a node is on a path satisfying the bounded cost constraint, and searches for the plan that optimizes the key function value. The major contributions of this paper are as follows:

- construction of a HTN for marine navigation problems with constraints on localization error
- using the modified PTS method to solve a planning problem with uncertainty in the cost of transitioning from one state to another in the task network. To the best of our knowledge, this contribution is the first time that uncertain transition cost is considered in hierarchical task network planning methods.

The paper is thus organized as follows. In section II, we review the HTN planning framework and potential search method for solving planning problems with uncertain transition costs. In section III, we formulate the HTN navigation problem with uncertain transition costs and in section IV, we solve the formulated problem using our combined HTN and potential search method. Finally, in section VI we demonstrate our approach with a simulated glider navigation problem and in section VII, we discuss our future efforts.

II. Preliminaries

In this section, we briefly review HTNs and the potential search method used to plan when branch costs may be uncertain.

A. HTN Planning

Planning in cognitive architectures aims to find a sequence of actions, through which the agent fulfills a set of tasks, while satisfying some constraints on those tasks [17]. Many of the existing methods, e.g., SHOP (Simple Hierarchical Ordered Planner) [18], SHOP2 [19], SIP2-2 (System for Interactive Planning and Execution) [20], etc, take advantage of the hierarchical nature of of task structure, and finds the plan by repeatedly decomposing tasks into smaller sub-tasks until a primitive task is reached. While these methods find feasible plans, they are unable to distinguish between plans of different quality.

Since the underwater vehicles have limited battery life, it is crucial that the planning algorithm generates plans that minimize certain costs associated with the engineering or flight characteristics (battery life, travel time). The work of [21] proposes a HTNPBP method that performs a best-first search in the plan space according to certain cost function. Other recent work in [22] demonstrates an SAT solver based approach to find optimal HTN planning solutions.

However, the high uncertainty in marine environment poses difficulty in designing the planning method in task networks. Ocean flow variability is often the dominant factor affecting motion of the underwater gliders. While operational ocean models, such as the Regional Ocean Modeling System [23] and the Hybrid Coordinate Ocean Model [24] can provide flow information over a large spatial domain and forecast over several days, the available flow field forecast may still contain high uncertainty and error. The uncertainty comes from multiple sources, including the incomplete physics or boundary conditions [25] and representation of some terms in the equations themselves [26]. Due to the uncertainty in flow forecast, the cost, e.g., travel time or energy consumption, of transitioning from one state to another in the task network is also uncertain. In this case, finding a path optimizing the total cost is an overly ambitious goal. Thus we formulate the planning problem as a bounded cost search problem, that is, finding the plan with the highest probability of having the total cost less than a pre-assigned upper bound. The bounded cost search problem has been solved in [27]–[29]. However, all of the above-mentioned works assume deterministic transition cost.

B. Potential Search

The bounded cost search problem is first proposed in [29]. The authors present two algorithms to solve the bounded cost search problem: the Potential Search (PTS) and the Anytime Potential Search (APTS). The PTS method defines the Potential Ordering Function, which is an implicit evaluation of the probability that a node is on a path satisfying the bounded cost constraint, and iteratively expands the nodes in the graph with the highest Potential Ordering Function value. The wavefront expansion terminates when the goal node has been expanded, and the path is found by backtracking of the wavefronts. The APTS method runs the PTS algorithm iteratively to improve on the incumbent solution, with the upper bound on total cost lowered in each iteration of the algorithm. Our previous work [16] proposed a modified PTS method that solves bounded cost search problem in a graph whose branch cost is uncertain. A novel key function is introduced to implicitly evaluate the probability of a path satisfying the bounded cost constraint, and the optimal path is generated by searching for the plan that optimizes the key function value.

III. Problem Formulation

We first define the dynamical model of the glider. Consider a glider with planar position $x \in \mathcal{D} \subset \mathbb{R}^2$, control heading angle $\psi(u) = \left[\cos(u), \sin(u)\right]^{\intercal}$, where u is the heading angle, and constant velocity v_c that has the following dynamics:

$$\dot{x} = F_R(x) + v_c\psi(u) \tag{1}$$

where $F_R(x)$ is the flow-field. We assume a model $\hat{F}_R(x)$ of the flow-field is known before hand and models $\hat{F}_R(x)$ as a piece-wise constant flow field discretized into rectangular cells $\mathcal{C}_i$, for $i = 1, 2, \ldots, N$ such that the flow speed in each cell $\mathcal{C}_i$ can be represented as a single constant vector $\hat{F}_i$. We assume also that our glider must periodically re-surface when navigating to a desired position in the environment in order to acquire a GPS fix on its position.

We are therefore interested in computing a sequence of inputs $u_1, u_2, \ldots, u_T$ and surfacing times so that our glider is able to navigate to a desired location, while at the same time with a large probability of having total cost less than an upper bound. In this work, the cost is defined as the travel time of the vehicle to reach the goal position from the starting position.

IV. Planning the Glider Trajectory

We now define the planning domain of the glider model, which will allow us to solve navigation planning problems where some actions yield uncertain costs. We construct predicates $at\_1(AUV)$, $at\_2(AUV)$, ..., $at\_N(AUV)$ which correspond to the AUV being located in one of the N discretized cells. In addition, we construct predicates $at\_surface(AUV)$ and $below\_surface(AUV)$ to indicate whether the glider is underwater and two predicates $BLoc\_HIGH(AUV)$ and $BLoc\_LOW(AUV)$ to indicate whether the glider's confidence in its current position while underwater is high, medium, or low. These predicates allow us to describe the glider's current abstract state which include the current cell location, whether underwater, and how confident the position estimate is. An example of the glider in various combinations of the described predicates is shown in Figure 1.

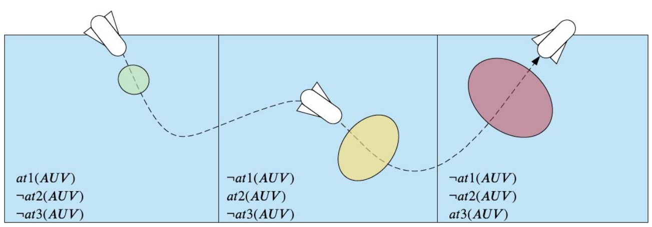

Fig. 1. Various examples of the abstracted state's active predicates along its yo-shaped trajectory up and down in the water column. Left: $at\_1(AUV)$, $at\_surface(AUV)$, $BLoc\_HIGH(AUV)$. Center: $at\_2(AUV)$, $below\_surface(AUV)$, $BLoc\_LOW(AUV)$. Right: $at\_3(AUV)$, $at\_surface(AUV)$, $BLoc\_LOW(AUV)$. The size and color of the shaded shape represents uncertainty.

Next, we define the primitive actions and their precondition constraints that allow our agent to interact with the environment. Let E define all pairs of neighboring cells. We construct primitive actions $move\_ij(AUV)$ for each $(i,j) \in E$ with the associated preconditions $at\_i(AUV)$, $below\_surface(AUV)$, and $BLoc\_HIGH(AUV)$ must all be true. These preconditions indicate our agent must be in the starting cell, underwater, and confident of its current position in order to move to the target cell. We assume that our agent maintains an internal metric valued state $BLoc \in \mathbb{R}^+$ which allows us to evaluate $BLoc\_LOW(AUV)$ and $BLoc\_HIGH(AUV)$ as a discretization of $BLoc$.

In order to improve our confidence in the position estimate, we provide the agent with an action $sense(AUV)$ with preconditions $at\_surface(AUV)$ that allows $BLoc\_HIGH(AUV)$ to become active and resets $BLoc$ to a minimum uncertainty value $\bar{\epsilon}$. In order to incorporate the need to periodically re-assess our position, we associate growth in $BLoc$ with taking any $move\_ij(AUV)$ action. For a plan $\pi = a_1, a_2, \ldots a_T$, $BLoc$ has growth dynamics

$$BLoc[k+1] = \begin{cases} \sqrt{BLoc[k]^2 + \epsilon^2}, & \text{if } a_k = move\_ij(AUV) \\ \bar{\epsilon}, & \text{if } a_k = sense(AUV) \\ BLoc[k], & \text{otherwise} \end{cases} \tag{2}$$

where ϵ is a fixed uncertainty growth that is defined beforehand. By incorporating (2), we ensure that any constructed plans must periodically surface and check that the plan is still sufficiently valid. We then construct a plan from some starting cell predicate to a goal cell predicate by using the potential search method detailed in Algorithm 1. This can be achieved by constructing a task $search(AUV, cell\_s, cell\_g)$ that produces a sequence of $move\_ij(AUV)$ actions which an HTN planner can then inject surfacing actions as needed to ensure the glider achieves a desired confidence level. An example of this is shown in Figure 2.

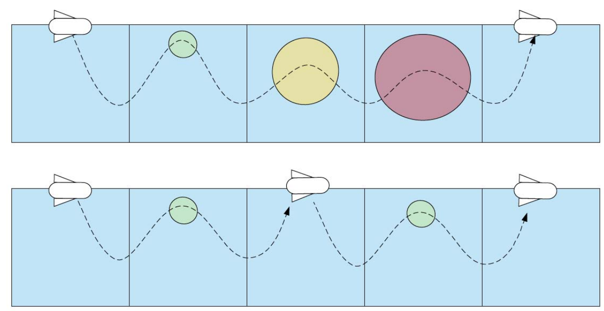

Fig. 2. Example of how an HTN planner can take a generated plan and inject necessary actions in order to ensure all move preconditions are satisfied. Top: An example potential search planned trajectory where the uncertainty grows too large. Bottom: The modified potential search planned trajectory using HTN to surface in order to ensure the glider's position uncertainty is limited. The size and shading of the circles represents uncertainty.

V. Solving the HTN Planning by the Modified PTS

First we derive cost of the primitive actions. Given the branch cost of the graph, the modified PTS method is applied to solve the HTN planning problem.

A. Cost of the primitive action

Let $i, j \in E$ denote two vertices of the same grid cell $\mathcal{C}_\alpha$. Let d denote the displacement of moving from vertex i to j. Since modeled flow field is assumed to be constant in each cell, the cost of performing the action $move\_ij(AUV)$ can be described as

$$t(d) = \frac{\|d\|}{\|v_c\psi + \hat{F}_\alpha\|}. \tag{3}$$

Since the vehicle moves at constant speed, the total vehicle speed $v_c\psi + \hat{F}_\alpha$ must be in the same direction as the segment of the path,

$$\frac{d}{\|d\|} = \frac{v_c\psi + \hat{F}_\alpha}{\|v_c\psi + \hat{F}_\alpha\|}. \tag{4}$$

Combining (3) and (4) we have

$$t(d) = \frac{1}{\|\hat{F}_\alpha\|^2 - v_c^2} \Big(d^T \hat{F}_\alpha - \sqrt{\big(d^T \hat{F}_\alpha\big)^2 + \|d\|^2 (v_c^2 - \|\hat{F}_\alpha\|^2)} \Big). \tag{5}$$

(5) describes the cost of taking the $move\_ij(AUV)$ action, given that the vehicle travels in the modeled flow field.

B. Planning using the modified PTS

A graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ can be constructed from the rectangular grid, where $\mathcal{V}$ is a set containing all the vertices of the grid cells, and $\mathcal{E}$ denotes the set containing all edges connecting the vertices. The branch cost between two nodes is the time of traveling between the corresponding vertices in the rectangular grid cell. Since the modeled flow field $\hat{F}_R$ might be different from the actual flow field, the travel time in (5) is an estimated branch cost, and is different from the actual branch cost.

The primitive action $move\_ij(AUV)$ enables the vehicle to move from one node to another in the graph. Thus the planning problem is a graph search problem, that is, finding a sequence of nodes in the graph that connects the starting node to the goal node, with high probability of having total cost less than an upper bound. All graph search algorithms search the nodes in the graph in certain sequence. The algorithms maintain an OPEN list and a CLOSED list. A graph node is labeled NEW if it has not been searched by the algorithm. The OPEN list contains all the nodes that are searched, but still have a NEW neighbor. The CLOSED list consists of all the nodes that have been accessed by the search algorithm. To determine which cells should be searched first, the algorithm computes the cost-of-arrival, which is the minimal cost of going from the starting node s to an arbitrary node n, and cost-to-go, which is the minimal cost of going from n to the goal point g.

Let $g^*(n)$ denote the actual cost-of-arrival, and let $h^*(n)$ denote the actual cost-to-go of a node n. Since the actual cost-to-go is unknown during the search, the following estimated cost-to-go is used to guide the search:

$$h(n) = \frac{|g - n|}{v_c + \max \|\hat{F}_R\|}. \tag{6}$$

The heuristic function defined in (6) is the travel time of the vehicle traveling in the most favorable flow condition. Hence, $h(n) \leq h^*(n)$, which guarantees admissibility of the algorithm. Since the branch cost is unknown in our problem, we introduce the estimated cost-of-arrival, denoted as $g(n)$. The estimated cost-of-arrival is computed by summation of the estimated branch cost along trajectory.

The potential function of a node n is defined as $PT(n) = Pr(h^*(n) + g^*(n) < C)$. The potential function characterizes the probability that a node is on a path that satisfies the bounded cost constraint. Nodes with high potential have higher probability to be part of the desired path. However, the exact potential of nodes cannot be computed or compared, since both $h^*(n)$ and $g^*(n)$ are unknown. Therefore, PTS algorithms usually design a key function to determine the nodes that need to be searched at each step of the graph search. Nodes in the OPEN list are sorted by the key function value. Various key functions have been proposed for different path planning problems with bounded cost [27]–[29]. Here we use the key function proposed in [16] as the evaluation function,

$$K(n) = \begin{cases} \dfrac{h(n)g(n)}{(C - h(n) - g(n))^2}, & \text{if } h(n) + g(n) < C \\ \infty & \text{, otherwise} \end{cases}. \tag{7}$$

Nodes with lower key function value have a higher probability of being on a path satisfying the bounded cost constraint. The intuition is that, if $h(n) + g(n) < C$, the key function value increases if either $h(n)$ or $g(n)$ is larger. In this case, the estimated cost $h(n) + g(n)$ increases, and will be closer to C, then it is less likely that the true cost satisfies the bounded cost constraint, and the node n is less likely to be on a feasible path. If $h(n) + g(n) \geq C$, then n cannot be on a path satisfying the bounded cost constraint, since $h^*(n) + g^*(n) \geq h(n) + g(n) \geq C$. In this case, the key function is set as positive infinity.

The PTS is then applied in HTN planning in Algorithm 1. The search algorithm consists of two processes: the expansion process and the backtracking process. During the iterative expansion process, the algorithm orders the nodes in the OPEN set according to the key function value, and inserts the node with the lowest key function value to the CLOSED set (lines 14, 15). Neighbors of this node and their key function values are updated if the neighboring nodes can be reached with a lower cost through the current node (lines 23, 24). The propagation continues until the OPEN list is depleted, or the goal node is in the OPEN set. Starting from the goal position, the backtracking process searches for the predecessor of the last node in the path set and add it to the path, until the starting node is included in the path (lines 32, 33).

VI. Simulations

In this section, we detail our simulation study and illustrate how our integrated HTN and Potential Search approach allows for a simulated glider to achieve bounded cost navigation while ensuring user-defined constraints are satisfied. We consider a domain of $N = 64$ cells where a glider starts at the surface in cell 1, i.e. $at\_1(AUV)$, $at\_surface(AUV)$, and $BLoc\_HIGH(AUV)$, and must surface in cell 64, i.e. achieve $at\_64(AUV)$ and $at\_surface(AUV)$. The cells are arranged in a grid the move actions are generated such that transitions may occur horizontally, vertically, and diagonally, but no transitions are available that move the glider out of the grid. We describe the belief dynamic parameters as $\epsilon = 1.5$ and $\bar{\epsilon} = 1$. We use the PyHop HTN planner to implement our modified HTN planning approach [30].

We provide two different $\hat{F}_R$ models to illustrate the effects of the flow conditions on the modified HTN plans. In Fig. 3, no flow is present in the light blue cells while a flow moving in the southwest direction is shown in the light teal cells. Notice that the planned trajectory from the potential search guides the glider (shown as the red dot) to move tangentially around the strong flow field and that due to the belief dynamic

Algorithm 1: The Modified PTS method

Data: Start and goal node n_s, n_g, travel cost upper bound C, graph $\mathcal{G}$.

Output: Optimal path γ.

Initialization.

$g(n) = \infty, h(n) = \infty, K(n) = \infty, \forall n \in \mathcal{G}$;

$n_s \rightarrow \{\text{CLOSED}\}$;

Set the heuristics and estimated cost-of-arrival of n_s;

(OPEN, CLOSED) = Expand(n_s, OPEN, CLOSED, $\mathcal{G}$);

while *OPEN is not empty* **do**

  if $g \in$ *CLOSED* **then**

    Backtracking(n_s, n_g)$\rightarrow \alpha$

  end

  $v = \arg\min_{n \in \text{OPEN}} K(n)$;

  (OPEN, CLOSED) = Expand(v, OPEN, CLOSED);

end

Function Expand(v, *OPEN, CLOSED,* $\mathcal{G}$) **:**

  $\{\text{OPEN}\} \setminus v \rightarrow \{\text{OPEN}\}$;

  $v \cup \{\text{CLOSED}\} \rightarrow \{\text{CLOSED}\}$;

  Find adjacent nodes $\{n_i\}_{i=1}^{m}$ to v in $\mathcal{G}$;

  for $i = 1$ *to* m **do**

    Compute estimated branch cost $w^*(v, n_i)$ using (5) ;

    if $w^*(v, n_i) + g(v) < g(n_i)$ **then**

      n_i.predecessor = v;

      $\{\text{OPEN}\} \cup n_i \rightarrow \{\text{OPEN}\}$;

      Compute $h(n_i)$ using (6);

      $g(n_i) = g(v) + w^*(v, n_i)$;

      Update $K(n_i)$ using (7);

    end

  end

  return *OPEN, CLOSED*

Function Backtrack(n_s, n_g) **:**

  $n_g \rightarrow \{\alpha\}$;

  while $n_s \notin \{\alpha\}$ **do**

    $v = \alpha(\text{end})$;

    v.predecessor $\cup\{\alpha\} \rightarrow \{\alpha\}$;

  end

  return $\{\alpha\}$

parameters, the glider surfaces every three move actions. The value of $BLoc$ is shown graphically by the green and red circles centered on the glider's position and the transparency of the red dot indicates whether the glider is underwater.

In our second example, we provide a more complicated flow (visualized in Fig. 4) where once again, light blue cells have no flow while light teal cells have a strong flow along the southwest direction. Again, our modified HTN planner produces a sequence of compliant actions that ensure localization uncertainty is properly addressed throughout the mission. In addition, the glider's trajectory has been altered to minimally enter the strong southwest flow.

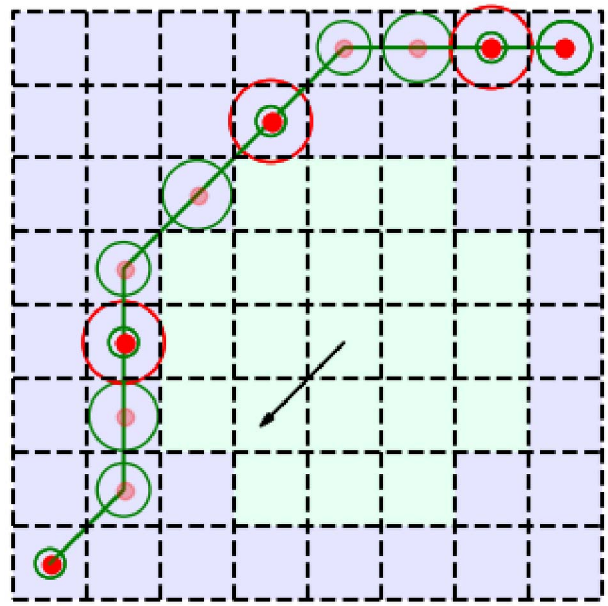

Fig. 3. First flow model test of the modified HTN planning method. The flow model in this case exhibits a strong southwest flow away from the target cell. The black arrow indicates the direction of the flow field.

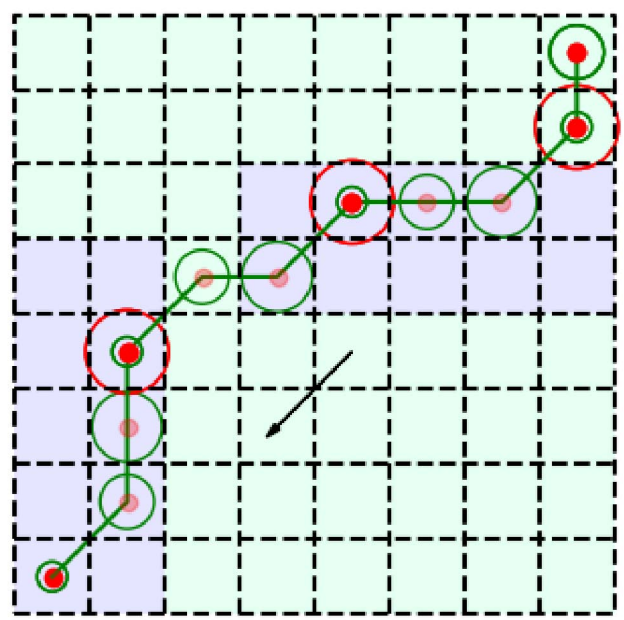

Fig. 4. Second flow model test of the modified HTN planning method. The flow model in this case exhibits a strong southwest flow away from the target cell and strongly favors paths that traverse the flow where the flow is only one cell wide. The black arrow indicates the direction of the flow field.

VII. Conclusion

In this paper, we have demonstrated a modified HTN planning method that leverages a potential search algorithm to generate navigation plans with high probabilities of achieving a bounded cost. In addition, any user-defined constraints provided to our search are obeyed during the execution of our potential search found plan. Our proposed approach was demonstrated in a marine autonomy application that can be incorporated with high-level cognitive architectures as a planning method to achieve navigation goals.

In future work, we intend to combine the modified HTN planner with the Metcognitive Integrated Dual-Cycle Architecture (MIDCA) cognitive architecture [12] in order to provide MIDCA with the ability to generate bounded cost plans that can achieve desired navigation goals. This approach will allow the MIDCA cognitive architecture to not only select more appropriate plans but also to check whether or not a valid plan exists.

Further, we plan to incorporate our approach with other constraints and objectives necessary in order to perform fish tracking. For example, tasks such as sensing acoustic signals emitted by tagged fish, monitoring for anomalous behavior caused by glider damage or remora attacks, and sensing local water properties are useful and necessary to detect the presence of a tagged fish and to improve our sensing and navigation capabilities.

ACKNOWLEDGEMENTS

T. Lin, M. Hou, and F. Zhang were supported by ONR grants N00014-19-1-2556 and N00014-19-1-2266; NSF grants OCE-1559475, CNS-1828678, and S&AS-1849228; NRL grants N00173-17-1-G001 and N00173-19-P-1412 ; and NOAA grant NA16NOS0120028. C. Edwards has been supported by NSF grants OCE-1559475, S&AS-1849137, as well as NOAA award NA16NOS0120028.

REFERENCES

[1] K. M. Lynch, I. B. Schwartz, P. Yang, and R. A. Freeman, "Decentralized environmental modeling by mobile sensor networks," *IEEE transactions on robotics*, vol. 24, no. 3, pp. 710–724, 2008.

[2] C. C. Eriksen, T. J. Osse, R. D. Light, T. Wen, T. W. Lehman, P. L. Sabin, J. W. Ballard, and A. M. Chiodi, "Seaglider: A long-range autonomous underwater vehicle for oceanographic research," *IEEE Journal of oceanic Engineering*, vol. 26, no. 4, pp. 424–436, 2001.

[3] O. Schofield, J. Kohut, S. Glenn, J. Morell, J. Capella, J. Corredor, J. Orcutt, M. Arrott, I. Krueger, M. Meisinger *et al.*, "A regional slocum glider network in the mid-atlantic bight leverages broad community engagement," *Marine Technology Society Journal*, vol. 44, no. 6, pp. 185–195, 2010.

[4] W. Wu, A. Song, P. Varnell, and F. Zhang, "Cooperatively mapping of the underwater acoustic channel by robot swarms," in *Proceedings of the International Conference on Underwater Networks & Systems*. ACM, 2014, p. 20.

[5] P. Testor, B. DeYoung, D. L. Rudnick, S. Glenn, D. Hayes, C. Lee, C. B. Pattiaratchi, K. L. Hill, E. Heslop, V. Turpin *et al.*, "Oceangliders: a component of the integrated goos," *Frontiers in Marine Science*, vol. 6, p. 422, 2019.

[6] G. R. Halliwell Jr, G. J. Goni, M. F. Mehari, V. H. Kourafalou, M. Baringer, and R. Atlas, "Osse assessment of underwater glider arrays to improve ocean model initialization for tropical cyclone prediction," *Journal of Atmospheric and Oceanic Technology*, vol. 37, no. 3, pp. 467–487, 2020.

[7] M. J. Oliver, M. W. Breece, D. A. Fox, D. E. Haulsee, J. T. Kohut, J. Manderson, and T. Savoy, "Shrinking the haystack: using an auv in an integrated ocean observatory to map atlantic sturgeon in the coastal ocean," *Fisheries*, vol. 38, no. 5, pp. 210–216, 2013.

[8] C. R. Edwards, S. Cho, F. Zhang, and S. Fangman, "Field and numerical studies to assess performance of acoustic telemetry collected by autonomous mobile platforms," *Ser. Nat. Marine Fisheries, ONMS-20-08*, 2020.

[9] D. Chang, F. Zhang, and C. R. Edwards, "Real-time guidance of underwater gliders assisted by predictive ocean models," *Journal of Atmospheric and Oceanic Technology*, vol. 32, no. 3, pp. 562–578, 2015.

[10] K. Shojaei and M. Dolatshahi, "Line-of-sight target tracking control of underactuated autonomous underwater vehicles," *Ocean Engineering*, vol. 133, pp. 244–252, 2017.

[11] C. Jones and D. Webb, "Slocum gliders–advancing oceanography," in *Proceedings of the 15th International Symposium on Unmanned Untethered Submersible Technology conference (UUST'07)*, vol. 15, 2007.

[12] M. T. Cox, Z. Alavi, D. Dannenhauer, V. Eyorokon, H. Munoz-Avila, and D. Perlis, "Midca: A metacognitive, integrated dual-cycle architecture for self-regulated autonomy," in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[13] J. G. Trafton, L. M. Hiatt, A. M. Harrison, F. P. Tamborello II, S. S. Khemlani, and A. C. Schultz, "ACT-R/E: An embodied cognitive architecture for human-robot interaction," *Journal of Human-Robot Interaction*, vol. 2, no. 1, pp. 30–55, 2013.

[14] S. D. Hanford, O. Janrathitikarn, and L. N. Long, "Control of mobile robots using the soar cognitive architecture," *Journal of Aerospace Computing, Information, and Communication*, vol. 6, no. 2, pp. 69–91, 2009.

[15] E. D. Sacerdoti, "A structure for plans and behavior," 1977.

[16] M. Hou, S. Cho, H. Zhou, C. R. Edwards, and F. Zhang, "Bounded cost path planning for underwater vehicles assisted by a partitioned flow field model," *Frontiers in Robotics and AI*, under review.

[17] M. Ghallab, D. Nau, and P. Traverso, *Automated Planning: theory and practice*. Elsevier, 2004.

[18] D. Nau, Y. Cao, A. Lotem, and H. Munoz-Avila, "SHOP: Simple hierarchical ordered planner," in *Proceedings of the 16th international joint conference on Artificial intelligence-Volume 2*, 1999, pp. 968–973.

[19] D. S. Nau, T.-C. Au, O. Ilghami, U. Kuter, J. W. Murdock, D. Wu, and F. Yaman, "SHOP2: An HTN planning system," *Journal of artificial intelligence research*, vol. 20, pp. 379–404, 2003.

[20] D. E. Wilkins, "Can AI planners solve practical problems?" *Computational intelligence*, vol. 6, no. 4, pp. 232–246, 1990.

[21] S. Sohrabi, J. A. Baier, and S. A. McIlraith, "HTN planning with preferences," in *Proceedings of the 21st International Jont Conference on Artifical Intelligence*, ser. IJCAI'09. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2009, p. 1790–1797.

[22] G. Behnke, D. Höller, and S. Biundo, "Finding optimal solutions in HTN planning-A SAT-based approach," in *IJCAI*, 2019, pp. 5500–5508.

[23] A. F. Shchepetkin and J. C. McWilliams, "The Regional Oceanic Modeling System (ROMS): a split-explicit, free-surface, topography-following-coordinate oceanic model," *Ocean Modelling*, vol. 9, no. 4, pp. 347 – 404, 2005.

[24] E. P. Chassignet, H. E. Hurlburt, O. M. Smedstad, G. R. Halliwell, P. J. Hogan, A. J. Wallcraft, R. Baraille, and R. Bleck, "The HYCOM (HYbrid Coordinate Ocean Model) data assimilative system," *Journal of Marine Systems*, vol. 65, no. 1, pp. 60 – 83, 2007.

[25] A. C. Haza, L. I. Piterbarg, P. Martin, T. M. Ozgokmen, and A. Griffa, "A Lagrangian subgridscale model for particle transport improvement and application in the Adriatic Sea using the Navy Coastal Ocean Model," *Ocean Modelling*, vol. 17, no. 1, pp. 68–91, 2007.

[26] P. F. Lermusiaux, "Uncertainty estimation and prediction for interdisciplinary ocean dynamics," *Journal of Computational Physics*, vol. 217, no. 1, pp. 176–199, 2006.

[27] J. T. Thayer, R. Stern, A. Felner, and W. Ruml, "Faster bounded-cost search using inadmissible estimates," in *Twenty-Second International Conference on Automated Planning and Scheduling*, 2012.

[28] R. T. Stern, R. Puzis, and A. Felner, "Potential search: A bounded-cost search algorithm," in *Twenty-First International Conference on Automated Planning and Scheduling*, 2011.

[29] R. Stern, A. Felner, J. van den Berg, R. Puzis, R. Shah, and K. Goldberg, "Potential-based bounded-cost search and anytime non-parametric A*," *Artificial Intelligence*, vol. 214, pp. 1–25, 2014.

[30] D. Nau, "Pyhop: A simple HTN planning system written in python," *Available: https*, 2013.