2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

Efficient Optimal Planning in non-FIFO Time-Dependent Flow Fields

James Ju Heon Lee[1], Chanyeol Yoo[1], Stuart Anstee[2] and Robert Fitch[1]

Abstract— **We propose an algorithm for solving the time-dependent shortest path problem in flow fields where the FIFO (first-in-first-out) assumption is violated. This problem variant is important for autonomous vehicles in the ocean, for example, that cannot arbitrarily hover in a fixed position and that are strongly influenced by time-varying ocean currents. Although polynomial-time solutions are available for discrete-time problems, the continuous-time non-FIFO case is NP-hard with no known relevant special cases. Our main result is to show that this problem can be solved in polynomial time if the edge travel time functions are piecewise-constant, agreeing with existing worst-case bounds for FIFO problems with restricted slopes. We present a minimum-time algorithm for graphs that allows for paths with finite-length cycles, and then embed this algorithm within an asymptotically optimal sampling-based framework to find time-optimal paths in flows. The algorithm relies on an efficient data structure to represent and manipulate piecewise-constant functions and is straightforward to implement. We illustrate the behaviour of the algorithm in an example based on a common ocean vortex model.**

I. INTRODUCTION

Many minimum-time planning problems in robotics inherently involve time-costs that are non-static. In terms of finding shortest paths on graphs, this means that edge traversal time is not a scalar value, but instead is a function that varies over time. The importance of developing shortest path algorithms for non-static travel times is well recognised, and somewhat surprisingly, has been studied for over 50 years [2]. In comparison to static shortest path problems, progress in developing a theoretical understanding of the *time-dependent shortest path* (TDSP) problem has proved far more elusive. Our goal is to explore relatively recent theoretical results in an effort to develop practical algorithms for robotics applications.

Our main motivation is planning for robots and vehicles that are influenced by fluid flows, such as those in the ocean and the atmosphere [37]. Planning in ocean currents is important for many applications such as oil and gas exploration [27], environmental monitoring [6, 26] and defence [14], with platforms such as underwater gliders, surface vessels, Wave Gliders, and profiling floats. Planning is critical when the maximum vehicle velocity is comparable to the prevailing current [20, 32–34]; the success of autonomous navigation is then directly tied to the ability to model [22, 31, 38] and exploit current predictions.

Known TDSP solution approaches remain difficult to apply due to the many subtle problem variants [3] whose complexity has only recently come to light. Perhaps as a consequence, many published algorithms have no stated performance bounds or, worse, make incorrect claims as noted in [9]. One important property of TDSP problems over graphs is the *FIFO* (first-in-first-out) property, which essentially states that delaying departure time can never result in earlier arrival. Therefore, in FIFO problems, remaining at any given node is never beneficial. Waiting is critical for optimality in the non-FIFO case, although if arbitrary waiting is permitted then a non-FIFO problem can be transformed into an equivalent FIFO version [3]. A second important property lies in characterising the edge travel time function as either discrete or continuous time. The discrete-time case (both FIFO and non-FIFO) is known to be polynomial in the number of edges and the length of the time horizon [3]. It is natural to exploit this discrete structure by searching the time-expanded graph using common algorithms such as A*. Execution time, however, can quickly become unwieldy for long time horizons, especially for slow-moving vehicles in the ocean. The continuous time case, again for both FIFO and non-FIFO problems, is non-polynomial in the general case, even for piecewise-linear cost functions [9].

Whereas it is not possible to avoid the worst-case bound in general, it is interesting to consider special cases. One restriction in FIFO problems is to limit the set of possible slopes for the pieces of the piecewise linear functions. This limitation is helpful because it allows for polynomial-time algorithms [9].

In this paper, we consider a previously unidentified special case for non-FIFO problems where the edge travel time function is piecewise constant and show that its worst-case bound is polynomial. We present an algorithm based on a data structure that supports efficient manipulations of the piecewise edge functions. We do not allow arbitrary waiting, which would not be possible in flow fields. Finally, our algorithm outputs solutions from every node and time to the destination, which means that the solution is in the form of a *policy* that can be used for replanning, for example. The piecewise-constant assumption is reasonable in practice because ocean current estimates are typically provided in this form [1], similar to upper atmosphere estimates [29].

A preliminary version of this work originally appeared in preprint form [18] and is presented formally here. Applications of the initial ideas appear in [19].

*This work is supported by an Australian Government Research Training Program (RTP) Scholarship, Australia's Defence Science and Technology Group and the University of Technology Sydney.

[1]Authors are with the University of Technology Sydney, Ultimo, NSW 2006, Australia {JuHeon.Lee,Chanyeol.Yoo,Robert.Fitch}@uts.edu.au
[2]Author is with the Defence Science and Technology Group, Department of Defence, Australia Stuart.Anstee@dst.defence.gov.au

II. RELATED WORK

Our work is related to the seminal paper [9] that showed that the computational complexity of FIFO problems with piecewise linear edge travel time functions is super-polynomial in the number of graph nodes. The paper further showed the existence of polynomial time special cases where the slopes of the travel time functions are restricted. We show that this bound also holds for finding minimum travel time paths in non-FIFO problems with piecewise constant edge functions, and present an algorithm with a straightforward implementation. The main distinction is that shortest paths in non-FIFO problems (that do not allow arbitrary waiting) may include cycles. Our analysis essentially bounds the length of cycles by relating the worst-case cycle time to properties of the edge functions.

A useful categorisation of problem variants along with complexity results was presented [3]. Interestingly, the tightest known worst-case bounds for general FIFO and non-FIFO problems are both polynomial (in the number of graph edges and the time horizon) in the discrete-time case, and are both super-polynomial in continuous time.

In discrete-time problems, optimal solutions may be found by searching the *time-expanded graph* [11, 12], where nodes are duplicated per unit time. In [5] first applied Dijkstra's algorithm in this case. More recent work is based on A* [7, 8, 13] with performance improvements using adaptive discretisation [17], precomputed heuristics [16], and bidirectional search [4]. Other approaches are based on time-aggregated graphs [10], where edge functions are represented as time series. Recent work [28] proposes a planning framework for time-varying currents where a locally-optimal solution is found at every timestep. Our approach is different in that we find a globally optimal solution in polynomial time.

A level set approach [24, 25] is presented for continuous-time problems. Such problem is often motivated by path planning for underwater or surface vehicles in the ocean. Recent work [23] formulates the problem as a time-varying Markov decision process. Since these methods assume general edge functions, known complexity results suggest that their worst-case running time is non-polynomial. Our approach finds time-optimal paths in flow fields by embedding our efficient TDSP solution within a sampling-based framework that is asymptotically optimal.

III. BACKGROUND

A. Time-dependent directed graph

We consider a directed graph $G = (S, E)$ that consists of a finite set of states S and edges $(s, s') \in E$ where $s, s' \in S$. The set of immediately reachable states from state s is denoted as $S_s \subseteq S$. A set of goal states is denoted as $S_g \subset S$ where $|S_g| \geq 1$. We restrict consideration to graphs in which goal states are reachable from the initial state.

We define an $(n+1)$-length path Γ within G as a sequence of states $\Gamma = s_0 s_1 \cdots s_n$, where $s_k \in S \setminus S_g$, $(s_k, s_{k+1}) \in E$ for all $k \in [0, n)$, and $s_n \in S_g$. We denote Γ_k as the prefix of Γ up to the k-th state in the path (i.e., $\Gamma_k = s_0 s_1 \cdots s_k$). Given an edge (s, s'), we define *edge time* $C_{ss'}(t)$ as the time to traverse from state s to s' after departing at time t. Without loss of generality, path traversal begins no earlier than $t = 0$, and edge time $C_{ss'}(t)$ is ∞ for all $t \leq 0$.

We define *arrival time* as the time to arrive at the final state in path Γ, departing from state s_0 at time t [3]. If the path only contains 2 states (i.e., $\Gamma = s_0 s_1$), then the arrival time is $a_\Gamma(t) = a_{s_0,s_1}(t) = C_{s_0 s_1}(t) + t$. For an n-length path, the arrival time is expressed recursively as $a_\Gamma(t) = a_{s_{n-2},s_{n-1}}(a_{\Gamma_{n-2}}(t))$. Similarly, we define *travel time* $T_\Gamma(t)$ as the time to complete the path Γ, departing from the initial state s_0 at time t. Formally,

$$T_\Gamma(t) = a_\Gamma(t) - t. \tag{1}$$

B. Definition of piecewise-constant function (PF)

A piecewise-constant function $f : \mathbb{R} \to \mathbb{R}$ is defined as a sequence of subdomain and constant pairs, where each subdomain is a time interval in which the value is constant. We define a piecewise-constant function $f(t)$ on subdomains p_i^f indexed backwards in time as follows

$$f(t) = \begin{cases} v_1^f, & \text{if } t > p_1^f \\ \vdots & \\ v_k^f, & \text{if } p_{k-1}^f \geq t > p_k^f \\ \vdots & \\ v_n^f, & \text{if } p_{n-1}^f \geq t > p_n^f \\ \infty, & \text{else} \end{cases} \equiv \begin{cases} v_1^f, & \text{if } t > p_1^f \\ \vdots & \\ v_k^f, & \text{ef } t > p_k^f \\ \vdots & \\ v_n^f, & \text{ef } t > p_n^f \\ \infty, & \text{else} \end{cases} \tag{2}$$

where $k \in \mathbb{N}$ and $p_{k+1}^f < p_k^f \ \forall k \in \mathbb{N}$. We use the short form 'ef' for 'else if'. Future PF definition will also use a short form omitting the '$else$' case, i.e., $t \in (-\infty, 0]$. Operations over PF are defined in [36]. Given two PF $f(t)$ and $g(t)$, we define an additional operation called *recursion* $f(t+g(t))$ using conditioning and merging operators such that:

$$f(t + g(t)) = (f(t + v_1^g) \ominus p_1^g) \oplus \cdots \oplus (f(t + v_n^g) \ominus p_n^g). \tag{3}$$

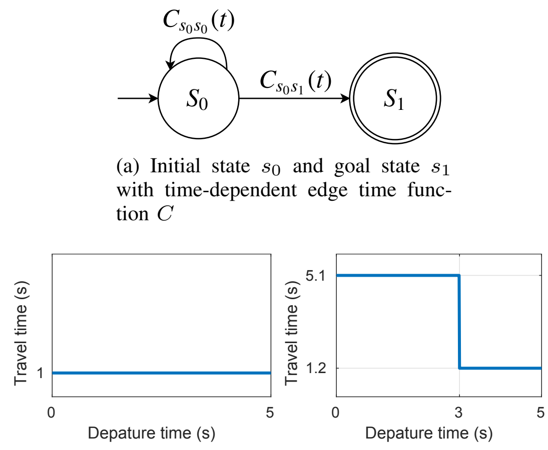
C. FIFO properties

The TDSP problem is often solved for minimal arrival time assuming *first-in-first-out* (FIFO) behaviour. A graph exhibits FIFO behaviour if it satisfies [3, 9] $t + C_{ss'}(t) \leq t' + C_{ss'}(t')$ for any edge $(s, s') \in E$ and $t \leq t'$. Intuitively, the arrival time $a_\Gamma(t)$ is non-decreasing with respect to departure time t.

Under the FIFO condition, optimal solutions exhibit the following properties [3]: 1) waiting at any state is not beneficial at any time, 2) optimal paths are acyclic (i.e., they do not revisit states), and 3) any subset of the optimal path is also a shortest path.

IV. PROBLEM STATEMENT

The objective is to minimise travel time, given initial and goal states. In many practical robotics applications, travel time is more important than arrival time since the constraints such as energy and cost are often tightly coupled to travel time. With the notation defined, the *time-dependent shortest path* (TDSP) problem for travel time is defined as follows.



(a) Initial state s_0 and goal state s_1 with time-dependent edge time function C

(b) Edge time $C_{s_0s_0}(t)$

(c) Edge time $C_{s_0s_1}(t)$

Fig. 1. Two-state graph example with time-dependent edge times

Problem 1 (Minimum travel time problem)**.** *Given a directed graph G with time-dependent edge time function C, find an optimal path Γ^* and initial departure time t_0^* that minimises the travel time T, such that*

$$(\Gamma^*, t_0^*) = \arg\min_{\Gamma,t} a_\Gamma(t) - t. \tag{4}$$

We are also interested in solving for the minimum travel time for a set departure time t_0 as follows.

Problem 2 (Minimum travel time problem given initial departure time)**.**

$$\Gamma^* = \arg\min_{\Gamma} a_\Gamma(t_0) - t_0. \tag{5}$$

We show later in this paper that the problems are equivalent under the proposed framework. Note that we do not consider waiting at an arbitrary node.

We are interested in a non-FIFO graph where the arrival time may not be non-decreasing with respect to departure time.

Example 1 (Non-FIFO graph)**.** *Consider a graph with time-dependent edge time function as shown in Fig. 1. The edge time $C_{s_0s_1}$ is 5.1 seconds for the first 3.5 seconds and then reduces to 1.2 seconds. The self-transition edge time $C_{s_0s_0}$ is 1 second for all departure time.*

Clearly, the graph does not exhibit FIFO behaviour. Transitioning immediately to goal state s_1 from s_0 takes longer than self-transitioning at state s_0 for a few times before arriving at goal state s_1. Note that self-transitioning is different to waiting; the former is defined as an edge transition allowed by a graph, whereas the latter is an arbitrary hold duration which is not defined in a graph.

V. PIECEWISE-CONSTANT REPRESENTATION

We present travel and edge time functions in the form of *piecewise-constant functions* (PF) [35, 36]. We illustrate the form and the relevant operations required to understand the PF-based value iteration where the solution is given as a policy rather than a path.

A. TDSP with piecewise-constant functions

We represent the travel time $T_\Gamma(t)$ and edge time $C_{ss'}$ using PF, as defined in (2). The travel time function in (1) can be re-written in an iterative form where $T_s^k(t)$ is the travel time for k edge transitions starting at state s. Suppose a (k+1)-length state transition from s_0 is $s_0s_1s_2...s_k$. Then,

$$\begin{aligned}
T_{s_0}^0(t) =& 0\\
T_{s_0}^1(t) =& C_{s_0s_1}(t)\\
T_{s_0}^2(t) =& C_{s_0s_1}(t) + C_{s_1s_2}(t + C_{s_0s_1}(t))\\
=& C_{s_0s_1}(t) + T_{s_1}^1(t + C_{s_0s_1}(t))\\
T_{s_0}^3(t) =& C_{s_0s_1}(t) + C_{s_1s_2}(t + C_{s_0s_1}(t))\\
&+ C_{s_2s_3}(C_{s_0s_1}(t) + C_{s_1s_2}(t + C_{s_0s_1}(t)))\\
=& C_{s_0s_1}(t) + T_{s_1}^2(t + C_{s_0s_1}(t))\\
&\vdots\\
T_{s_0}^k(t) =& C_{s_0s_1}(t) + T_{s_1}^{k-1}(t + C_{s_0s_1}(t)),
\end{aligned} \tag{6}$$

which can be formally written as a Bellman Equation:

$$T_s^{k+1}(t) = \min_{s' \in S_s} C_{ss'}(t) + T_{s'}^k(t + C_{ss'}(t)). \tag{7}$$

We denote by $T_s^*(t)$ the converged travel time, where $T_s^{k+1}(t) = T_s^k(t)$ for all $t \in \mathbb{R}$ and some finite $k \in \mathbb{Z}$.

We give the solution to the general TDSP problem as a *travel policy* $\pi_s(t)$ represented as a PF for each state, allowing for time-dependent transitions to other states. Formally,

$$\pi_s(t) = \begin{cases} \vdots & \vdots \\ s_s^i \in S_s & \text{if } t > p_s^i \\ \vdots & \vdots \end{cases}, \tag{8}$$

where s_s^i is the state visited during the i-th subdomain. Given a travel policy π_s, the travel time function is written as

$$T_s^{k+1}(t) = \begin{cases} \vdots & \vdots \\ C_{ss_s^i}(t) + T_{s_s^i}^k(t + C_{ss_s^i}(t)) & \text{if } t > p_s^i \\ \vdots & \vdots \end{cases}, \tag{9}$$

where p_s^i is the beginning of the i-th subdomain in policy π_s.

B. Optimal travel policy for non-FIFO TDSP problems

Given a directed graph G and time-dependent edge time function $C_{ss'}(t)$, the optimal travel policy π_s^* to reach the goal state $s_g \in S_g$ from state s can be expressed as

$$\pi_s^*(t) = \begin{cases} \vdots & \vdots \\ \arg\min_{s' \in S_s} C_{ss'}(t) + T_{s'}^*(t + C_{ss'}(t)) & \text{ef } t > p_s^{i*} \\ \vdots & \vdots \end{cases}, \tag{10}$$

where the travel time for the goal states $s \in S_g$ at any iteration k is 0 for all $t \in \mathbb{R}$. In principle, the optimal solution of (10) at state s could be computed iteratively, by finding the optimal next state $s' \in S_s$ for each time $t > 0$. We

Algorithm 1 Solving for optimal policy $\pi_s^*(t)\ \forall s \in S$

Inputs: Directed graph $G=(S,E)$, time-dependent edge time function C, goal state s_g and max number of edge transitions K
Outputs: Optimal travel policy π^{k*} and travel time T^{k*}
$T_s^0 \leftarrow \begin{cases} 0, & \text{if } t>0 \end{cases}, \forall s \in S$
$E \leftarrow (s_g, s_g)$
$C_{s_g s_g} \leftarrow \begin{cases} 0, & \text{if } t>0 \end{cases}$
for $k \leftarrow 1$ **to** K **do**
  for all $s \in S \setminus s_n$ **do**
    $T_{ss'}^{k+1} \leftarrow \begin{cases} \infty & \text{if } t>0 \end{cases}, \forall s' \in S$ ▷ (11)
    $P_s^{k+1*} \leftarrow \{0\}$ ▷ (12)
    for all $s' \in S_s$ **do**
      $T_{ss'}^{k+1} \leftarrow \begin{cases} C_{ss'}(t) + T_{s'}^k(t + C_{ss'}(t)) & \text{if } t>0 \end{cases}$ ▷ (11)
      $P_s^{k+1*} \leftarrow P_s^{k+1*} \cup P_{ss'}^{k+1*}$ ▷ (12)
    $T_s^{k+1} \leftarrow \begin{cases} \vdots & \vdots \\ \min\limits_{s' \in S_s} T_{ss'}^{k+1} & \text{ef } t > p_s^{i*}, \forall p_s^{i*} \in P_s^{k+1*} \\ \vdots & \vdots \end{cases}$
    $\pi_s^{k+1} \leftarrow \begin{cases} \vdots & \vdots \\ \arg\min\limits_{s' \in S_s} T_{ss'}^{k+1} & \text{ef } t > p_s^{i*}, \forall p_s^{i*} \in P_s^{k+1*} \\ \vdots & \vdots \end{cases}$
  if $T_s^{k+1}(t) \equiv T_s^k(t), \forall t \in P_s^{k+1*}, s \in S$ **then**
    break
return $\pi_s^k(t)$ and $T_s^k(t)\ \forall s \in S$

prove in Sec. VI that the number of subdomains converge in a finite time. Let $T_{ss'}^{k+1}$ be an *immediate travel time function* in which the transition from state s to s' occurs over all time t after k edge transitions. Formally,

$$T_{ss'}^{k+1} = \begin{cases} C_{ss'}(t) + T_{s'}^k(t + C_{ss'}(t)) & \text{if } t>0 \end{cases}. \tag{11}$$

Let P_s^k be the set of subdomains in travel time function T_s^k. Such a set for optimal travel time T_s^{k+1*} is

$$P_s^{k+1*} = \{0\} \cup \bigcup_{s' \in S_s} P_{ss'}^{k+1*}, \tag{12}$$

where subdomain set $P_{ss'}^{k+1*}$ is from immediate travel function $T_{ss'}^{k+1*}$ and $P_s^0 = \{0\}$. The optimisation problem in (10) is then solved over a finite set of subdomains P_s^{k+1*} where $p_s^{i*} \in P_s^{k+1*}$ in (10). The pseudocode is in Alg. 1.

VI. ANALYSIS

Without loss of generality, we assume $S_s = S$ for all states $s \in S \setminus S_g$ (i.e., all states are immediately reachable from any state). Given (11) and (12), the set of subdomains can be written as

$$P_s^{k+1} = \{0\} \cup \left\{ p \in \bigcup_{s' \in S_s} C_{ss'} \cup \left(P_{s'}^k - C_{ss'}\right) \mid p \geq 0 \right\}, \tag{13}$$

where $A - B = \{a - b \mid \forall a \in A \text{ and } b \in B\}$ for two sets A and B. We slightly abuse notation for $C_{ss'}$ to denote the set of subdomains and constants in the corresponding edge time function. Using a short form $\frac{A}{0} = \{a \in A \mid a \geq 0\}$, we have

$$\begin{aligned} P_s^{k+1} =& \{0\} \cup \bigcup_{s'} C_{ss'} \cup \bigcup_{s'} \frac{P_{s'}^k - C_{ss'}}{0} \\ =& \{0\} \cup \bigcup_{s'} C_{ss'} \cup \bigcup_{s'} \bigcup_{s''} \frac{C_{s's''} - C_{ss'}}{0} \\ & \cup \bigcup_{s'} \bigcup_{s''} \frac{\frac{P_{s'}^{k-1} - C_{s's''}}{0} - C_{ss'}}{0}. \end{aligned} \tag{14}$$

Since all subdomain sets for edge time function are non-negative by definition, the last term can be written in a form

$$\frac{\frac{A}{0} - \frac{B}{0}}{0} = \{a - b \mid a \in A, b \in B, a \geq 0, b \geq 0, a - b \geq 0\}. \tag{15}$$

Since B (i.e., $C_{ss'}$) is non-negative,

$$\frac{A - B}{0} = \{a - b \mid a \in A, b \in B, a - b \geq 0 \text{ and } b \geq 0\}. \tag{16}$$

Intuitively, if $a - b \geq 0$ and $b \geq 0$, then $a \geq 0$. Therefore

$$\frac{\frac{A}{0} - B}{0} \equiv \frac{A - B}{0}, \tag{17}$$

if B is a non-negative set. Then the last term in (14) becomes

$$\frac{\frac{P^{k-1} - C_{s's''}}{0} - C_{ss'}}{0} = \frac{P_{s'}^{k-1} - C_{ss'} - C_{s's''}}{0}. \tag{18}$$

Therefore, the set of subdomains can be recursively written as

$$\begin{aligned} P_s^{k+1} =& \{0\} \cup \bigcup_{s'} C_{ss'} \cup \bigcup_{s'} \bigcup_{s''} \frac{C_{s's''} - C_{ss'}}{0} \\ & \cup \cdots \cup \bigcup_{s'} \cdots \bigcup_{s^k} \frac{C_{s^{k-1}s^k} - \cdots - C_{ss'}}{0}. \end{aligned} \tag{19}$$

Lemma 1 (Finite edge transitions for convergence given infinite length path)**.** *Given an arbitrary and infinite length path* Γ*, the set of subdomains converges to a unique and finite set within a finite number of edge transitions.*

All subdomains in any edge time function are non-negative by definition. Therefore subdomains in (19) monotonically decrease as the number of edge transitions increase. Since all subdomains in travel time function are non-negative, there exists a finite maximum number of edge transitions $K_{\max}$ before convergence.

Lemma 2 (Convergence in finite edge transitions)**.** *Given an arbitrary and infinite length path* Γ*, the worst case number of edge transitions before convergence in subdomain set is*

$$K_{\max} = \text{ceil}\left(\frac{\max C}{\min C}\right), \tag{20}$$

where $C = \bigcup_{s \in S} \bigcup_{s' \in S} C_{ss'} \setminus \{0\}$.

The worst-case number of edge transition $K_{\max}$ is the last edge transition before the subdomain set in $K_{\max}$-th term becomes empty in (19) since subdomains are non-negative. The longest such term is $\max C - \sum_{k=1}^{K_{\max}} \min C$.

Remark 1 (Convergence and cyclic paths)**.** *The length of an optimal path that may include cycles is bounded by a finite number of edge transitions found in Lemma 2.*

Theorem 1 (Convergence in optimal algorithm)**.** *The optimal algorithm in Alg. 1 converges in a finite time $K_{\max}$ as shown in Lemma 2.*

By Lemma 1 and 2, there exists a unique and finite set of subdomains for travel time functions. Since the optimisation problem is to find optimal travel policy for each subdomain, the problem is then solved in a finite number of iterations.

Theorem 2 (Time complexity)**.** *The overall time complexity for Alg. 1 is $\mathcal{O}((|S| \cdot |C_m|)^{k+2})$, where $|S|$ is the number of states in graph G, $|C_m|$ is the maximum number of subdomains over a set of edge time functions and k is the number of edge transitions.*

By (9), the overall complexity is related to the number of subdomains, which is bounded by

$$\begin{aligned} |P_s^{k+1}| &= \left| \{0\} \cup \bigcup_{s'} C_{ss'} \cup \bigcup_{s'} \frac{P_{s'}^k - C_{ss'}}{0} \right| \\ &\leq 1 + (|S| \cdot |C_m|) + (|S| \cdot |C_m|) \cdot |P_{s'}^k| \\ &\leq 1 + 2(|S| \cdot |C_m|) + 2(|S| \cdot |C_m|)^2 \\ &+ \cdots + 2(|S| \cdot |C_m|)^{k+1} \\ &= \mathcal{O}((|S| \cdot |C_m|)^{k+2}), \end{aligned} \tag{21}$$

where C_m is the set of subdomains and constants with the maximum cardinality over all edges $e \in E$. In the worst case, we find the optimal policy for each subdomain using value iteration in Alg. 1. Since the time complexity for solving such value iteration is polynomial in number of states, the overall time complexity for the proposed algorithm is $\mathcal{O}((|S| \cdot |C_m|)^{k+2})$.

Remark 2 (Time-static reduction)**.** *From Theorem 2, the time complexity for time-static edge functions is reduced to $\mathcal{O}(|S|^2)$ (i.e., $|C_m| = 1$ and $K = 0$) which agrees with the complexity of static shortest path problems.*

Remark 3 (Time complexity in practice)**.** *The time complexity in Theorem 2 is based on three worst-case conditions: 1) all states are connected to all the others, 2) the maximum number of iterations depends on the smallest edge subdomain and 3) no overlapping subdomains.*

The worst-case conditions in Remark 3 occur rarely in practice, particularly Condition 3. When subdomains in a set overlap, they merge and form a much smaller set. Therefore the set does not grow indefinitely until convergence.

By Theorem 2, the complexity heavily depends on the number of edge transitions (i.e., iterations). Since the maximum number of edge transitions depends on the subdomains among all edge functions as shown in Lemma 2, the running time of the algorithm can be improved significantly by adaptively pruning early rapid changes to increase the denominator (i.e., $\min C$). Furthermore, we can also reduce the size of the edge functions by merging consecutive constants that are similar (i.e., reducing $|C_m|$).

VII. OPTIMAL PLANNING OVER TIME-DEPENDENT FLOW

We present a flow field scenario where we sample a graph over a continuous flow field and solve the path planning problem in an asymptotically optimal manner. The algorithm was run on a standard laptop with Intel i5-6300 2.5GHz CPU and 8GB RAM.

A. Time-dependent edge time functions from flow field

Suppose we have a vehicle R modelled as $\dot{\mathbf{x}} = \mathbf{v}_R(\theta) + F(\mathbf{x}, t)$, where $\mathbf{x} \in \mathbb{R}^2$ is the position of the vehicle, $F(\mathbf{x}, t) = [u_{\mathbf{x},t}, v_{\mathbf{x},t}]^T$ is the time-dependent flow vector and $\mathbf{v}_R \in \mathbb{R}^2$ is the vehicle velocity relative to the flow. The vehicle is controlled by varying the bearing angle θ whilst travelling at constant speed $V_{\max}$. The discrete time model for the vehicle is represented as

$$\mathbf{x}[k+1] = \mathbf{x}[k] + (\mathbf{v}_R(\theta) + F(\mathbf{x}[k], t)) \cdot \Delta t. \tag{22}$$

Given two graph states s and s' that are located at $\mathbf{x}_s$ and $\mathbf{x}_{s'}$, respectively, we enumerate a set of control samples $\Theta = \{\theta_0, \cdots\}$ (i.e., bearing angles) to compute the corresponding set of trajectories $\mathbf{X} = \{\mathbf{x}^0, \cdots\}$ for a predefined time horizon h. For a given control θ_i and departure time t, we start from $\mathbf{x}^i[k] = \mathbf{x}_s$ until $k = H$ using (22), where $H = \text{ceil}(h/\Delta t)$ is discrete time horizon. Once the enumeration is completed, we find the trajectory $\mathbf{x}^{i^*} \in \mathbf{X}$ that approaches closest to $\mathbf{x}_{s'}$, such that

$$i^* = \arg\min_i \min_{k \leq H} \|\mathbf{x}_{s'} - \mathbf{x}^i[k]\|. \tag{23}$$

The time taken for the trajectory to reach $\mathbf{x}_{s'}$ is denoted as the edge time at departure time t. By Lemma 1, there exists a finite number of subdomains assuming that the flow forecast is also given in a form of piecewise-constant functions. This is a practically valid assumption as discussed in Sec. VI.

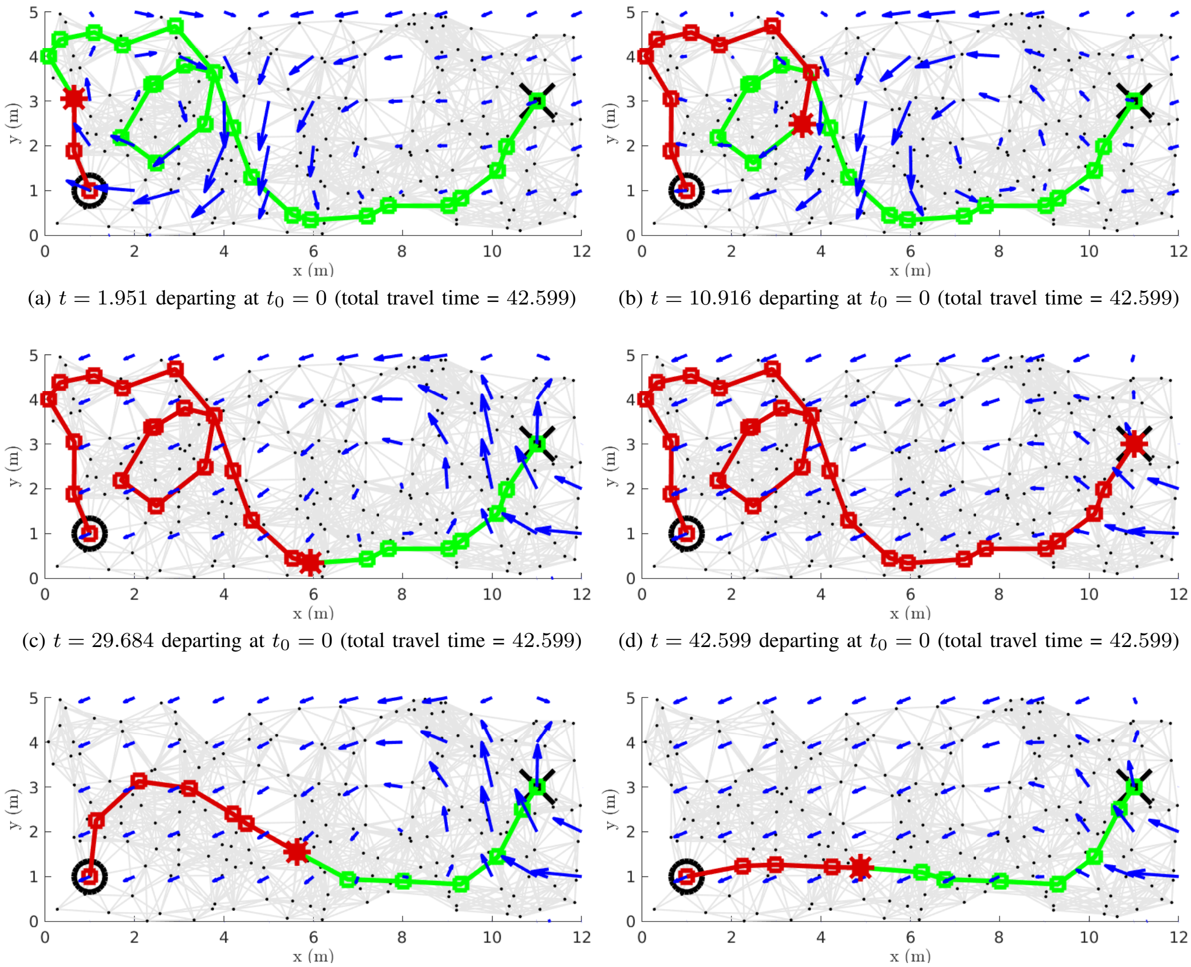
B. Asymptotically optimal planning for TDSP problems

We use the *probabilistic roadmap** (PRM*) [15] to construct a roadmap graph on which we implement our algorithm. As the roadmap contains cyclic edge connections, the full capability of our approach can be demonstrated.

C. Example

We generated a time-dependent flow field using the Taylor-Green gyre vortex model [30], which is commonly used to model ocean currents as shown in Fig. 2 (in blue). The strongest flow field velocity was set to 0.7 m/s As described in Sec. VII-B, we randomly sampled 200 states over the space and used a connection radius of $r = 1.735$ to connect them. Edge cost was evaluated using $V_{max} = 0.5$ m/s. Sampled states and edges are shown in black.

The vehicle starts from the bottom left corner (circle) to reach the top right corner (cross). Figures 2a-2d illustrate



(a) $t = 1.951$ departing at $t_0 = 0$ (total travel time = 42.599) (b) $t = 10.916$ departing at $t_0 = 0$ (total travel time = 42.599)

(c) $t = 29.684$ departing at $t_0 = 0$ (total travel time = 42.599) (d) $t = 42.599$ departing at $t_0 = 0$ (total travel time = 42.599)

(e) $t = 31.824$ departing at $t_0 = 12$ (total travel time = 34.837) (f) $t = 40.432$ departing at $t_0 = 25$ (total travel time = 36.263)

Fig. 2. Navigating at different initial departure time $t_0 = 0$, 12 and 25 through time-dependent flow field from start (bottom left) and to destination (top right). The red line is the path prefix and the green is the suffix, where t is the travel time for the prefix since the departure t_0. The red asterisk is the current location. Blue arrows represent the flow vector at a given position. The PRM*nodes and edges are shown in black.

its progress over the optimal path at $t_0 = 0$ and the flow evolution over time. The red line represents the trajectory followed up to time t and the green line is the remaining path. The total travel time is 42.599. The vehicle spends time on the left side until the flow in the middle weakens. Then it moves towards the goal state against a weak opposing flow. An interesting observation is that the vehicle visits the same state twice, which is concrete evidence of the non-FIFO property. The optimal travel policy was found after 44 iterations and the running time was 583 seconds. It is important to note that the policy can provide the optimal solution at any given initial departure time instantly.

In Fig. 2e and 2f, we demonstrate optimal paths for two different initial departure times, $t_0 = 12$ and 25, respectively. Since the flows are more relaxed than departing at $t_0 = 0$, the overall travel times are reduced although the arrival times are later. It is important to note that optimising travel time for ocean vehicles is more valuable than optimising arrival time when their endurance is dominated by travel time, as is often the case. We have shown that our proposed framework can find the optimal departure time that minimises the overall travel time in one such example.

VIII. Conclusion and Future Work

We have presented a new algorithm for finding the shortest paths in time-dependent graphs and have shown how it can be combined with sampling-based methods for planning in time-dependent flow fields. This result expands the set of known polynomial-time special cases for TDSP to include non-FIFO problems with piecewise-constant edge travel time functions. Previously, special cases with restricted slopes were restricted to FIFO problems. Our result also provides a new practical solution for planning in flow fields, with known performance bounds, that is applicable to autonomous vehicles in the ocean [21, 32]. An important avenue of future work is to develop efficient algorithms with performance guarantees for stochastic cases where flow field velocities and vehicle control are uncertain. It is also important to explore further special cases in non-FIFO TDSP problems that may be solved in polynomial time.

REFERENCES

[1] E. P. Chassignet, H. E. Hurlburt, E. J. Metzger, O. M. Smedstad, J. A. Cummings, G. R. Halliwell, R. Bleck, R. Baraille, A. J. Wallcraft, C. Lozano, *et al.*, "US GODAE: global ocean prediction with the HYbrid Coordinate Ocean Model (HYCOM)," *Oceanography*, vol. 22, no. 2, pp. 64–75, 2009.

[2] K. L. Cooke and E. Halsey, "The shortest route through a network with time-dependent internodal transit times," *J. Math. Anal. Appl.*, vol. 14, no. 3, pp. 493–498, 1966.

[3] B. C. Dean, "Shortest paths in FIFO time-dependent networks: Theory and algorithms," MIT, Tech. Rep., 2004.

[4] U. Demiryurek, F. Banaei-Kashani, C. Shahabi, and A. Ranganathan, "Online computation of fastest path in time-dependent spatial networks," in *Proc. of SSTD*, 2011, pp. 92–111.

[5] S. E. Dreyfus, "An Appraisal of Some Shortest-Path Algorithms," *Oper. Res.*, vol. 17, no. 3, pp. 395–412, 1969.

[6] G. D'urso, J. J. H. Lee, O. Pizarro, C. Yoo, and R. Fitch, "Hierarchical MCTS for Scalable Multi-Vessel Multi-Float Systems," in *Proc. of IEEE ICRA*, 2021.

[7] E. Fernández-Perdomo, J. Cabrera-Gámez, D. Hernández-Sosa, J. Isern-González, A. C. Domínguez-Brito, A. Redondo, J. Coca, A. G. Ramos, E. Fanjul, and M. García, "Path planning for gliders using Regional Ocean Models: Application of Pinzón path planner with the ESEOAT model and the RU27 trans-Atlantic flight data," in *Proc. of IEEE OCEANS*, 2010, pp. 1–10.

[8] E. Fernández-Perdomo, D. Hernández-Sosa, J. Isern-González, J. Cabrera-Gámez, A. C. Domínguez-Brito, and V. Prieto-Marañón, "Single and multiple glider path planning using an optimization-based approach," in *Proc. of IEEE OCEANS*, 2011, pp. 1–10.

[9] L. Foschini, J. Hershberger, and S. Suri, "On the complexity of time-dependent shortest paths," *Algorithmica*, vol. 68, no. 4, pp. 1075–1097, 2014.

[10] B. George and S. Shekhar, "Time-aggregated graphs for modeling spatio-temporal networks," *J. Semantics Data*, vol. 9, pp. 191–212, 2008.

[11] V. M. V. Gunturi, E. Nunes, K. Yang, and S. Shekhar, "A critical-time-point approach to all-start-time Lagrangian shortest paths: A summary of results," *Adv. in Spat. and Temporal Databases*, vol. 6849 LNCS, pp. 74–91, 2011.

[12] V. M. Gunturi, S. Shekhar, and K. Yang, "A Critical-Time-Point Approach to All-Departure-Time Lagrangian Shortest Paths," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 10, pp. 2591–2603, 2015.

[13] J. Isern-Gonzalez, D. Hernandez-Sosa, E. Fernandez-Perdomo, J. Cabrera-Gamez, A. C. Domınguez-Brito, and V. Prieto-Maranon, "Path planning for underwater gliders using iterative optimization," in *Proc. of IEEE ICRA*, 2011, pp. 1538–1543.

[14] H. Johannsson, M. Kaess, B. Englot, F. Hover, and J. Leonard, "Imaging sonar-aided navigation for autonomous underwater harbor surveillance," in *Proc. of IEEE/RSJ IROS*, 2010, pp. 4396–4403.

[15] S. Karaman and E. Frazzoli, "Sampling-based Algorithms for Optimal Motion Planning," *Int. J. Robot. Res.*, vol. 30, no. 7, p. 20, 2010.

[16] S. Kontogiannis and C. Zaroliagis, "Distance Oracles for Time-Dependent Networks," *Algorithmica*, vol. 74, no. 4, pp. 1404–1434, 2016.

[17] D. Kularatne, S. Bhattacharya, and M. A. Hsieh, "Optimal Path Planning in Time-Varying Flows Using Adaptive Discretization," *IEEE RA-L*, vol. 3, no. 1, pp. 458–465, 2018.

[18] J. J. H. Lee, C. Yoo, S. Anstee, and R. Fitch, "Efficient Optimal Planning in non-FIFO Time-Dependent Flow Fields," *arXiv e-prints*, p. arXiv:1909.02198, Sep 2019.

[19] J. J. H. Lee, C. Yoo, S. Anstee, and R. Fitch, "Hierarchical planning in time-dependent flow fields for marine robots," in *Proc. of IEEE ICRA*, 2020.

[20] J. J. H. Lee, C. Yoo, R. Hall, S. Anstee, and R. Fitch, "Energy-optimal kinodynamic planning for underwater gliders in flow fields," in *Proc. of ARAA ACRA*, 2017.

[21] K. M. B. Lee, J. J. H. Lee, C. Yoo, B. Hollings, and R. Fitch, "Active perception for plume source localisation with underwater gliders," in *Proc. of ARAA ACRA*, 2018.

[22] K. M. B. Lee, C. Yoo, B. Hollings, S. Anstee, S. Huang, and R. Fitch, "Online estimation of ocean current from sparse GPS data for underwater vehicles," *Proc. of IEEE ICRA*, 2019.

[23] L. Liu and G. S. Sukhatme, "A Solution to Time-Varying Markov Decision Processes," in *Proc. of IEEE ICRA*, 2018.

[24] T. Lolla, P. F. J. Lermusiaux, M. P. Ueckermann, and P. J. Haley, "Time-optimal path planning in dynamic flows using level set equations: theory and schemes," *Ocean Dynam.*, vol. 64, no. 10, pp. 1373–1397, 2014.

[25] T. Lolla, M. P. Ueckermann, K. Yigit, P. J. Haley, and P. F. J. Lermusiaux, "Path planning in time dependent flow fields using level set methods," in *Proc. of IEEE ICRA*, 2012, pp. 166–173.

[26] D. L. Rudnick, R. E. Davis, C. C. Eriksen, D. M. Fratantoni, and M. J. Perry, "Underwater gliders for ocean research," *Mar. Technol. Soc. J.*, vol. 38, no. 2, pp. 73–84, 2004.

[27] L. M. Russell-Cargill, B. S. Craddock, R. B. Dinsdale, J. G. Doran, B. N. Hunt, and B. Hollings, "Using Autonomous Underwater Gliders for Geochemical Exploration Surveys," *APPEA J.*, vol. 58, pp. 367–380, 2018.

[28] M. Shu, X. Zheng, F. Li, K. Wang, and Q. Li, "Numerical simulation of time-optimal path planning for autonomous underwater vehicles using a markov decision process method," *Appl. Sci.*, 2022.

[29] A. Stein, R. R. Draxler, G. D. Rolph, B. J. Stunder, M. Cohen, and F. Ngan, "NOAA's HYSPLIT atmospheric transport and dispersion modeling system," *B. Am. Meteorol. Soc.*, vol. 96, no. 12, pp. 2059–2077, 2015.

[30] G. I. Taylor and A. E. Green, "Mechanism of the production of small eddies from large ones," *Proc. R. Soc. Lond. A*, vol. 158, no. 895, pp. 499–521, 1937.

[31] K. Y. C. To, F. H. Kong, K. M. B. Lee, C. Yoo, S. Anstee, and R. Fitch, "Estimation of spatially-correlated ocean currents from ensemble forecasts and online measurements," *Proc. of IEEE ICRA*, 2021.

[32] K. Y. C. To, K. M. B. L. Lee, C. Yoo, S. Anstee, and R. Fitch, "Streamlines for motion planning in underwater currents," *Proc. of IEEE ICRA*, 2019.

[33] K. C. To, J. J. H. Lee, C. Yoo, S. Anstee, and R. Fitch, "Streamline-based control of underwater gliders in 3d environments," in *Proc. of IEEE CDC*, 2019, pp. 8303–8310.

[34] K. C. To, C. Yoo, S. Anstee, and R. Fitch, "Distance and steering heuristics for streamline-based flow field planning," in *Proc. of IEEE ICRA*, 2020, pp. 1867–1873.

[35] C. Yoo, "Provably-correct task planning for autonomous outdoor robots," Ph.D. dissertation, University of Sydney, 2014.

[36] C. Yoo, R. Fitch, and S. Sukkarieh, "Probabilistic Temporal Logic for Motion Planning with Resource Threshold Constraints," in *Proc. of RSS*, 2012.

[37] ——, "Online task planning and control for fuel-constrained aerial robots in wind fields," *Int. J. Robot. Res.*, vol. 35, no. 5, pp. 438–453, 2016.

[38] C. Yoo, J. J. H. Lee, S. Anstee, and R. Fitch, "Path planning in uncertain ocean currents using ensemble forecasts," in *Proc. of IEEE ICRA*, 2021, pp. 8323–8329.