Optimal Path Finding in Direction, Location and Time Dependent Environments

Irina S. Dolinskaya
Department of Industrial Engineering and Management Sciences,
Northwestern University, Evanston, IL 60208

Abstract

This paper examines optimal path finding problems where cost function and constraints are direction, location and time dependent. Recent advancements in sensor and data-processing technology facilitate the collection of detailed real-time information about the environment surrounding a ground vehicle, an airplane or a naval vessel. We present a navigation model that makes use of such information. We relax a number of assumptions from existing literature on path-finding problems and create an accurate, yet tractable, model suitable for implementation for a large class of problems. We present a dynamic programming model which integrates our earlier results for direction-dependent, time and space homogeneous environment and, consequently, improves its accuracy, efficiency and run-time. The proposed path finding model also addresses limited information about the surrounding environment, control-feasibility of the considered paths, such as sharpest feasible turns a vehicle can make, and computational demands of a time-dependent environment. To demonstrate the applicability and performance of our path-finding algorithm, computational experiments for a short-range ship routing in dynamic wave-field problem are presented.

Keywords: path-planning, dynamic programming, vessel routing, dynamic environment

1 Introduction

A variety of cutting-edge sensor and data-processing technology is currently being developed that is capable of collecting real-time information about the environment surrounding a ground vehicle, an airplane or a naval vessel. An explicit incorporation of this information into a path-planning process has potential to significantly enhance its efficiency and benefits. In this paper, we present an optimal path finding algorithm that makes use of such information. For example, an innovative coherent (Doppler) X-band radar mounted on a ship can measure the ocean wave-field surrounding a moving or stationary vessel in real-time [23, 43]. Then, wave evolution and propagation models can use this information to forecast how the wave-field changes over time [32, 33]. The presented path finding model makes use of the gathered information to improve accuracy of the decision making process. An optimum vessel performance in an evolving wave-field problem [13] motivated this research, and the short-range ship routing is used to illustrate practical application of the path-finding model. Nevertheless, the analysis and results presented here are general enough to be applicable to a large group of problems and could be easily translated to other applications.

We study optimal path finding problems with direction, location and time dependent environments, where the main complexity of the problems arises from the dependence of cost function and constraints on the location of the vehicle, the direction it is heading, and time. The direction-dependence of the surrounding medium implies that the travel time function is asymmetric, i.e., traveling from point a to point b does not necessarily incur the same cost as the reversed path from b to a . Consequently, our cost function is not a metric and, in general, violates the triangle inequality, which prohibits the use of more traditional and established approaches to solving optimal-path finding problems.

To accommodate the limited visibility horizon of the physical sensors, our model presumes to have complete information about the environment (i.e., cost function and constraints) within the sensor-visible region and only limited information beyond that. To further facilitate implementation of our path finding model to real-life problems, we integrate additional constraints that are critical to a wide range of applications. In the navigation of aerial, ground and naval vehicles, the dynamics of an agent constrains the curvature of a feasible path by a minimum turning radius function. This prohibits the use of a traditional dynamic programming path finding model that optimizes over a set of piecewise-linear paths. Subsequently, instead of the conventional approach of addressing the optimal-path finding and path-following aspects of the problem separately, we integrate the system's operability and dynamic constraints into an optimization model resulting in a control-feasible solution.

The benefits of innovative sensor technology and real-time data collection are subject to timely utilization of the available information in a decision making process. Therefore, computational demand and run-time of the optimal path finding algorithm is of particular significance. Traditional modeling of path finding in a dynamic network (i.e., time-dependent cost functions) requires addition of the time variable to its state space.

This increase in the dimension of a model state space increments the computational time of an algorithm by orders of magnitude. In the presented model, we undertake this, and other, computationally demanding aspects of the problem and introduce modeling and implementation methods that significantly improve the run-time.

1.1 Related Work

We give a broad overview of the published work related to the optimal path finding problems in a direction, location and time dependent medium. A more detailed analysis of the literature is presented in later sections of the paper, as they pertain to the specific focus areas of the discussion.

An optimal path finding problem in a location, and possibly time, dependent environment is widely studied in the literature. For example, the Zermelo Navigation Problem, introduced by Zermelo [48], is to steer a vessel along a minimum-time path through a region of the position-dependent strong current vectors, while the ship's velocity relative to water is assumed to be constant. This is one of the classical problems in the fields of calculus of variation and optimal control theory [5], and a number of extensions and variations have been studied since the introduction of the problem. The majority of published work assumes that a closed form speed function is available to the user, facilitating an application of various optimal control and calculus tools to the analytical function [19, 20, 28, 37, 39]. Similarly, Kimball and Story [25] draw analogy between sailboats and light rays to extend application of optical principles to optimal sailing trajectories.

In [38], Perakis and Papadakis express vessel speed function in terms of location- and, in some case, time-dependent variables that characterize the surrounding wave-field. The authors then analyze the speed function to find the fastest paths for deterministic and stochastic settings. It is important to note that in both settings of [38], the derived function provides the expected value of the speed for a given wave distribution, which assumes that the vessel travels at a specified direction for prolonged time to encounter enough waves to accurately represent the distribution. The new technology such as X-band radar that motivated our work provides access to the detailed information about each individual wave surrounding the vessel, and we integrate this information into our path planning model, avoiding the stationary assumptions.

When analytical cost and constraint functions are not available, discrete dynamic programming (DP) provides a relatively efficient optimal path finding model. DP has been widely studied and applied to an extensive list of problems for over half a century, since the publication of a ground breaking book by Bellman

[3]. The application of dynamic programming is particulary beneficial to the complex problems requiring sequential decision making. DP simplifies the problem at hand by creating a set of sub-problems, whose solution results in an optimal solution to the original problem. Dynamic programming modeling of the path finding problems involves the discretization of the path domain into a set of 'waypoints' (or nodes), consequently reducing the problem to optimization in a directed network [10]. For example, in [2, 40, 42] researchers apply DP algorithm to an optimal yacht sailing problem using speed prediction model from [41] that evaluates the vessel speed for a specified range of wind speeds and yacht heading angles. We expand the dynamic programming approach to deliver a control-feasible path in a dynamic environment while decreasing the algorithm's run-time and computational demand.

1.2 Overview of the Results

In this paper, we analyze the optimal path finding problems with a direction, location and time dependent environment. We relax the assumptions made in our earlier work [14, 15] of time and space homogeneity and deliver an efficient path finding algorithm that incorporates the detailed real-time information about the surrounding environment.

Information collected by sensors is usually evaluated by discrete computer simulation and forecasting models, and a cost function (e.g., vessel's speed) is not available in closed form nor can it be evaluated analytically. To address this, we discretize the path space into a set of *waypoints* (intermediate nodes) and construct a dynamic programming (DP) model that finds an optimal ordered set of waypoints to traverse from a given starting point to a predetermined target point. A traditional dynamic programming formulation of an optimal path finding problem implements the straight line segments between the pairs of consecutive waypoints. We present an alternative model that delivers a control-feasible path and does not violate the system dynamics restrictions. In other words, we take into account the sharpest possible turns a vessel or other mobile agent can make. To do so, the heading angle of a vehicle is added to the state space of our DP model. A minimum turning radius function constrains the curvature of feasible paths to be considered between a pair of DP states. We assume that a local region enclosing the set of neighboring waypoints is stationary and implement our previous results for a fastest path with bounded curvature [14] to find the cost of traveling from one DP state to the next (see Figure 3).

To incorporate the physical limitations of the data-collecting sensors and radars, we introduce a notion of a *visibility horizon* that restricts the distance of how far, from the vehicle's current location, the sensors can accurately monitor the surrounding environment. Due to the limited knowledge about the sensor-invisible region, the environment beyond the visibility horizon is assumed to be random with a stationary distribution. With these assumptions, our previous results for time and space homogeneous environment [14, 15] can be implemented to find an optimal path to continue the travel from the sensor visibility horizon to the target point located further away (see Figure 2).

Finally, we address the computational demand associated with the time-dependent nature of the medium. A traditional approach to path finding in a dynamic network (i.e., with time-dependent arc costs) is to include the time variable into the DP state. We present a dynamic programming model that integrates the vehicle's speed into its decision space, while simultaneously eliminating the time variable from the model's state space. We demonstrate that the resulting formulation of the problem decreases computational time by orders of magnitude. Algorithm 1 presented in this paper integrates all of the presented work into a single optimal path finding model for a direction, location and time dependent environment.

The rest of the paper is organized as follows. We conclude the introduction with Section 1.3 by presenting the notation to be used throughout the paper and a formal statement of the problem. Section 2 discusses the three main aspects addressed in our path finding model: limited visibility horizon of the on-board sensors (Section 2.1), integration of the vehicle's dynamic constraints, such as turning radius, (Section 2.2), and efficient integration of the environment time-dependency (Section 2.3). Section 3 integrates all the discussed aspects into a single path finding model and presents Algorithm 1. Section 4 discusses specific techniques that further improve the efficiency of our Algorithm 1 at its implementation and programming stage. Here, we also present the results of implementing the path finding model to the short-range ship routing in evolving wave-field problem. Finally, Section 5 concludes this paper.

1.3 Notation and Problem Statement

We are interested in finding a fastest path from a given starting point $s \in \mathbb{R}^2$ to a given target point $t \in \mathbb{R}^2$, where t_0 denotes the start time. Without loss of generality, we assume $t_0 = 0$. Since the curvature of any feasible path is constrained by a minimum turning radius function, the initial and target heading angles at points s and t can affect the set of feasible paths considered in the problem. Therefore, we integrate the starting heading, denoted by $\theta_s \in S^1$, and the final heading, $\theta_t \in S^1$, as inputs to our problem. For many applications, the target heading angle is not explicitly specified, in which case we find a path minimizing the travel time over all possible values of θ_t.

All paths from s to t lie in a direction, location and time dependent environment in $\mathbb{R}^2$. We let $R_H \in \mathbb{R}^+$ denote a *visibility horizon* that restricts how far the on-board sensors can collect information about the surrounding environment relative to the vehicle's current location (see Figure 1). Thus, for any point a inside the visible region (i.e., $\|s - a\| \leq R_H$) and $t_a \geq t_0$, we are given the functions $V(a, \theta_a, t_a)$, denoting a maximum attainable speed, and $R(a, \theta_a, t_a)$, indicating the minimum turning radius, for a vehicle located at point a with heading direction θ_a at time t_a. We assume that in addition to knowing information about the environment at time t_0, the forecasting tools use real-time data to evaluate the functions $V(a, \theta_a, t_a)$ and $R(a, \theta_a, t_a)$ for $t_a \geq t_0$. We do not explicitly integrate an upper bound on t_a for which the information is available, implying that a vehicle leaves the visible region prior to reaching the forecasting time bound.

The definition of R_H implies that we do not have the explicit $V(a, \theta_a, t_a)$ and $R(a, \theta_a, t_a)$ functions outside the visibility horizon. We defer our discussion on how to collect, characterize and integrate the necessary information for the region beyond R_H to Section 2.1.

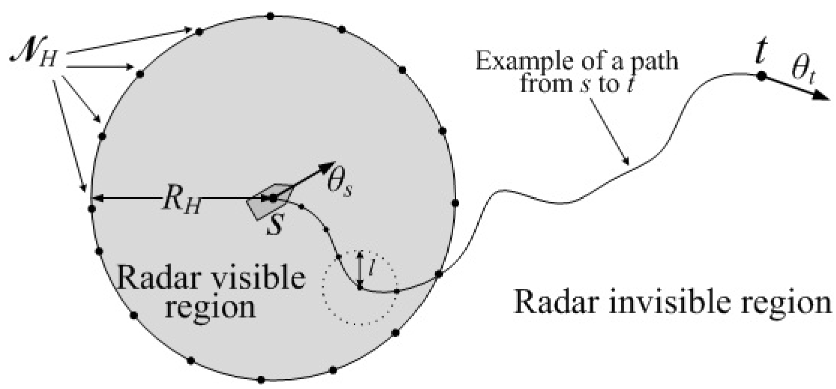

Figure 1: Illustration of notation.

Problem statement: Find a fastest path starting at time $t_0 = 0$ from the initial state $(s, \theta_s) \in \mathbb{R}^2 \times S^1$ to the target state $(t, \theta_t) \in \mathbb{R}^2 \times S^1$, where for all $a \in \mathbb{R}^2 : \|s - a\| \leq R_H$ and all $t_a \geq t_0$ the curvature of a feasible path is constrained by a minimum turning radius function $R(a, \theta_a, t_a)$, and the maximum attainable speed is described by function $V(a, \theta_a, t_a)$.

For completeness, we also introduce the notation to be used later in the discussion of our dynamic programming model. To construct a discrete DP path finding model we discretize the $\mathbb{R}^2$ path domain into a set of waypoints. We let l denote a dynamic programming discretization parameter representing the distance between any pair of consecutive waypoints connected by an arc. We assume that a 'local region' surrounding a waypoint with its radius equal to l can be approximated by a time and space homogeneous environment.

In addition, we let $\mathcal{N}_H$ denote a discrete set of waypoints (or nodes) on the visibility horizon through which all the paths from s to t considered by our model have to pass (the assumption $\|t - s\| > R_H$ is implied). That is, $\forall a \in \mathcal{N}_H, \|a - s\| = R_H$. Usually, set $\mathcal{N}_H$ consists of equally spaced points on the circle centered at s with the radius equal to R_H. A more detailed discussion of the discretization parameters is presented later in the paper when they are introduced in the path finding model.

2 Essential Components of the Path Planning Model

We now address three essential aspects of the path finding problem that are integrated into our model to facilitate real-life implementation. In Section 2.1, we discuss how limitation of the on-board sensors' visibility is incorporated into the decision making process. Integration of the vehicle's dynamic constraints, such as sharpest feasible turns, is presented in Section 2.2. Finally, Section 2.3 describes an efficient integration of the surrounding environment time-dependency into a dynamic programming model.

2.1 Limited Visibility Horizon

Despite the continuous improvement of sensor and data-collection technology, all physical systems have limitations, and it would be unrealistic for an optimization model to assume unlimited availability of information. To incorporate this restriction, we introduce the concept of *visibility horizon* and let R_H denote the radius of the sensor-visible region surrounding an aerial, ground or sea-surface vehicle. Thus, the on-board sensor system is assumed to collect and forecast all the necessary information about the environment (i.e., speed function $V(a, \theta_a, t_a)$ and turning radius function $R(a, \theta_a, t_a)$) inside the visibility horizon. However, the model does not have access to the detailed real-time information about the medium lying further than R_H distance away from the vehicle's current location.

Depending on a specific application, there are various ways to estimate and characterize the invisible environment. In some cases, information collected by the sensors inside the visible region can be extrapolated beyond the horizon R_H. Alternatively, prior experience and historical data for a given or similar region can be used to evaluate a stationary distribution of the environment that a vehicle expects to encounter (such as in [38]). In other cases, supplementary forecast and sensor technology can be employed to give a global estimation of the medium. For example, aircrafts and vessels use meteorological and hydrological forecasts provided by the national and international agencies to obtain information about the surrounding environment (e.g., National Oceanic and Atmospheric Administration [31]).

The forecast methods described above provide information about the surrounding environment on a global scale, by characterizing the expected distribution of waves or wind. As a result, limited information beyond R_H often leads to an assumption of a stationary environment for that region, implying time and space homogeneity of the cost function and constraints. This is especially the case for the short-range trips such as in the instance of our motivating application. Consequently, we assume that the environment outside the visibility horizon is a stationary distributed stochastic system, such that for each instance of the problem, a single fixed parameter can characterize the random distribution. For example, in the case of naval navigation, a parameter called 'sea state' describes the wave spectrum as a stationary random process over a short-term "time period in the range from $1/2$ hour to ... 10 hours" [18].

Time and space homogeneity of the region beyond the sensor visibility horizon eliminates the need for a dynamic programming path finding model beyond R_H. Instead, the closed form solutions presented in our earlier work [14, 15] can quickly deliver an optimal path in the stationary environment. In turn, we develop a dynamic programming model that evaluates the fastest paths from s to all the points on the border of the visible region (i.e., all the points in set $\mathcal{N}_H$), and use results from [14] and [15] to find the best path to continue the travel in the homogeneous environment from R_H border to the target point t. Note that when the distance between points in $\mathcal{N}_H$ and the destination point is long relative to the minimum turning radius of a vehicle, the curvature constraint can be neglected in the homogeneous environment, facilitating the implementation of the results from [15] in the invisible region (see Figure 2). Furthermore, the extreme weather conditions and other obstacles beyond R_H can be integrated into the optimization model by implementing the obstacle-avoiding fastest-path finding algorithm also discussed in [15].

Real-time data collection and timely implementation of the fastest-path finding algorithm allows us to continuously reevaluate an optimal path to reflect the most current information about the surrounding environment. As vehicle moves along the path progressing closer towards the destination point, the sensor-visible region also moves. Our goal is to deliver an optimal path finding algorithm with the computational time small enough to allow the user to reevaluate the path before traveling beyond the visibility horizon. When that is achieved, the vehicle never traverses the sensor-invisible region, and additional real-time information

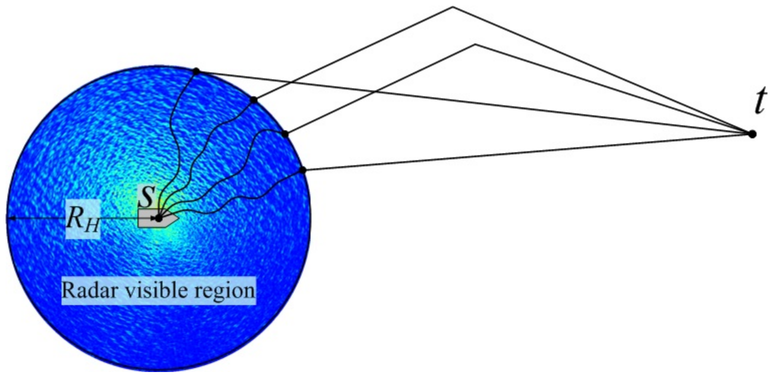

Figure 2: DP model evaluates the fastest paths to the points on the visibility horizon, then algorithm from [15] is used to find the best paths to continue.

is collected prior to reaching the visibility horizon.

The continuous reevaluation of an optimal path is similar in concept to a common dynamic programming practice called 'rolling horizon' [1, 26, 36]. However, it is important to differentiate that in a traditional DP rolling horizon approach the user has to make a decision on how far out into the future the model should look before choosing an action for the following time period. In the case of limited visibility horizon, we do not have a choice of how much information to integrate into the optimization model, and our goal is to find the best path based on all the available information about the environment.

To illustrate the scale of a visibility horizon observed in practice, we provide an example from the short-range vessel routing problem [13]. In this particular application, we are interested in missions with the target point located approximately 30 to 90 minutes of travel time away from the start. It is clear that more sophisticated radar equipment with greater visibility radius would always deliver a better path. However, the current limitations of the radar system restrict the visibility radius to at most 10 minutes worth of travel time [23, 43]. While this implies that we do not have the complete information about the environment from the path start to its finish, we have a sufficiently large R_H for our model to reevaluate an optimal path before vessel travels outside the radar visible region.

2.2 System Dynamics Restrictions

We now discuss the details of a dynamic programming model to be implemented inside the visible region, which finds the fastest paths from point s to all the points in $\mathcal{N}_H$. First, we direct our attention to the integration of the system dynamic constraints, such as sharpest feasible turns.

Dynamic programming is a predominant approach to the fastest-path finding problems in a time and/or location dependent environment. A traditional DP path finding model discretizes the domain of a path into a set of *waypoints* (or nodes) and a straight line path is implemented between every pair of the neighboring waypoints. As a result, the classical dynamic programming model finds an optimal path that is piecewise-linear. However, in many applications (e.g., vessels, airplanes and cars) a vehicle cannot instantaneously change its heading, making such paths infeasible. We introduce a minimum turning radius function $R(a, \theta_a, t_a)$ that restricts the curvature of feasible paths and has to be integrated into the path finding model.

The control-infeasibility of an optimal path found using the standard DP approach comes from a traditional practice of considering the optimal-path finding and path-following to be the separate stages of a problem. In such practice, an optimization model is first used to find a fastest path while neglecting the system dynamics. Then, a control model is implemented to assist the vehicle following the optimal path as closely as possible. We know that an optimal path found by the traditional DP model is control-infeasible for our problem, and the inherited error cannot be avoided during the implementation of such a path. To address this predicament, we integrate the system's operability and dynamic constraints into the optimization model, which in turn, delivers a control-feasible solution.

The problem of finding a fastest path with bounded curvature has been predominantly studied in the field of robotics for time and space homogeneous environment. Dubins [17] was first to introduce the problem and characterize an optimal path. However, his research and most of the work that followed (e.g., [4, 6, 8, 44, 45, 46, 47]) focused on the problems with direction-independent speed and minimum turning radius. This assumption is too strong for applications where the vehicle's motion is affected by the direction of waves, winds or slope of the terrain, like the class of problems discussed in this paper.

In the current literature on unmanned aerial vehicle (UAV) routing [29, 35], the researchers introduce a uniform wind vector field, where the aircraft velocity and wind velocity are assumed to have constant magnitude. The actual velocity of the vehicle is equal to the sum of the two vectors. While the resulting speed and minimum turning radius functions are direction-dependent, they have the very distinct structures that are too restrictive for many applications. McNeely *et al.* [30] consider the fastest-path finding problem for UAVs in the presence of a time-dependent wind vector field. However, the resulting speed of the vehicle also has a specific structure.

In our earlier work [14], we found the fastest path with bounded curvature in a time and space homogeneous environment for the very general direction-dependent speed and minimum turning radius functions. We have shown that such a path has a specific structure consisting of only sharpest turn arc and straight line segments (for more details see [14]). Here, we integrate these results into the dynamic programming model to find an optimal path in the time and space dependent environment that satisfies the vehicle dynamics.

We augment heading angle of a vehicle onto the state space of our dynamic programming model, resulting in the new state of the system that represents the vehicle's location in $\mathbb{R}^2$ and the heading angle at which it arrived to that waypoint. Then, instead of implementing a straight line path between a pair of waypoints, as it is done in a traditional DP model, we find a fastest path between a pair of the new DP states that does not violate the vehicle dynamics. Thus, the decision space of our model contains the next waypoint of the path and the arrival direction for that point (see Figure 3).

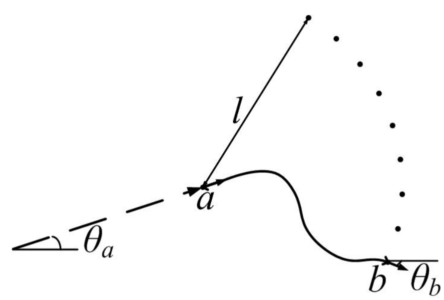

Figure 3: We integrate vehicle's heading angle into the state space of the model and find a fastest path satisfying the curvature constraints between a pair of the new DP states (e.g., (a, θ_a) and (b, θ_b)). The discretization parameter $l > 0$ denotes the distance between a pair of consecutive waypoints.

The discretization parameter $l > 0$ denotes the distance between a pair of consecutive waypoints in our DP model. For a given location and time instance, (a, t_a), we assume that the surrounding local region with radius l is small enough to be accurately approximated by a time and space homogeneous environment. Then, we use information about the environment in the local region to evaluate its stationary approximation. Speed function $V_{a,t_a}(\theta)$ and minimum turning radius function $R_{a,t_a}(\theta)$ are obtained from the functions $V(x, \theta, t_a)$ and $R(x, \theta, t_a)$ $\forall x \in \mathbb{R}^2 : \|x - a\| \leq l$, respectively, to correspond to the local time and space homogeneous region. Consequently, the problem of finding a minimum travel time path from one state of the model to the next is reduced to a problem of fastest path with bounded curvature in the direction-dependent media. We implement the results from [14] and compute the arc cost for every pair of "neighboring" DP states. The forthcoming Section 3 illustrates mathematical integration of the presented approach into the dynamic programming model.

2.3 Computational Demand of a Time-Dependent Environment

We continue our discussion of the dynamic programming path-finding model to be implemented inside the sensor-visible region by addressing the computational demand associated with the time-dependency of the medium. In a time-dependent environment it is not necessarily optimal to arrive at an intermediate waypoint of a path as soon as possible. For example, a vehicle arriving at some point along a path just a few minutes later might observe more favorable weather conditions, resulting in an overall better cost. To account for this fact, the time variable is traditionally added to the DP state to keep track of all possible times at which a vehicle might arrive at, and consequently leave, a particular waypoint. This additional variable significantly increases the number of DP states to be considered by an algorithm. We present an alternative formulation of the dynamic programming functional equation that allows us to eliminate the time variable from the DP state space.

The majority of published work addressing fastest-path finding problems in a time-dependent network considers two distinct cases. In the first case, no waiting is permitted in the network nodes (i.e., path waypoints), and the time one leaves a given node has to equal its arrival time. In the second set of problems, waiting is allowed in the network nodes, and the vehicle's departure time from a node can be greater than its arrival time.

In general, the problems with prohibited waiting are more difficult to solve than when unlimited waiting is permitted at the nodes. In fact, Orda and Rom [34] show that some instances of such problems are NP hard. Kaufman and Smith [24] introduce a *consistency* condition (or *first in first out* property) which guaranties that it is always advantageous to arrive at the intermediate nodes of a path as early as possible and continue the travel without a delay. They show that under consistency, the time-dependent fastest path can be calculated with exactly the same computational complexity as the static path. However, the consistency condition might be restrictive and does not apply to a number of applications, such as aerial and naval vehicle navigation.

Cooke and Halsey [9] propose an algorithm for the problems with prohibited waiting where cost functions have discrete range and domain. Their algorithm first finds an upper bound on the total travel time and then uses this bound to set the boundary condition for the DP backward recursive equation. Alternatively, Chabini [7] assumes that the travel time function is time-dependent only for a finite time interval, and the problem becomes static beyond some time horizon. This assumption allows Chabini to use the static shortest path to set the boundary condition for his dynamic programming formulation. Both papers propose the backward DP formulations of the problem with a time variable being part of the DP state. Since their algorithms evaluate the fastest paths for all possible times of arrival at the destination node, they are efficient for the problems with the objective to find paths from multiple departure nodes in the network or with multiple departure times. However, these methods are not as efficient in finding a fastest path from one origin to one destination point with a single fixed departure time.

When unlimited waiting in the nodes is allowed, the dynamic fastest path problem can be solved using Dijkstra's algorithm [11] just as efficiently as in the case of a static network. Since in such problems a vehicle is permitted to wait at any given node until the optimal time to continue its travel, it is favorable to arrive at each node as fast as possible. Then, the optimal waiting time at each node is part of the decision a fastest path algorithm must make. It might appear that adding waiting time to the decision space of the model increases the complexity of the optimal path finding procedure, the reality turns out to be otherwise.

Dreyfus [16] was the first one to address the optimal path finding problems in a dynamic network where unlimited waiting is permitted. His approach is based on redefining a traversing time function for each arc. When a delayed departure from node i decreases the time of arrival, the traverse time function represents the elapsed time between the time of arrival to node i and the corresponding earliest possible time of arrival to the following node j. With this alternative definition of arc cost, the consistency condition defined in [24] holds true. Therefore, we can formulate the dynamic programming functional equation without the time variable being present in the DP state. In this manner, the problem is reduced to the optimal path finding problem in a static network. Correspondingly, the optimal solution to a fastest-path finding problem with unrestricted wait consists of the ordered set of nodes and the optimal delay (or waiting time) at each of those nodes.

Waiting at the nodes (i.e., waypoints) is not permitted for the problems presented in this paper. However, by allowing an agent to vary its speed, it is possible to slow down enough for the vehicle to arrive at node i at the optimal time to continue traversing the following arc (i, j). Thus, instead of arriving at node i as soon as possible and waiting until the optimal time to depart it, our vehicle intentionally slows down and arrives at node i at the precise time of the optimal departure.

We implement Dreyfus' approach to find the optimal times to depart each waypoint, while using the maximum attainable speed of the vehicle ($V(.)$) to evaluate the arc cost. Then, we assume that it is always feasible to travel along an arc with any speed less than or equal to the maximum attainable speed. We also assume that there are no constraints on how quickly we can change the vehicle's speed. If this assumption is not valid, applying a slower speed for some arc might not allow the vehicle to speed up enough to reach the desired speed for the following arc of an optimal path. In our future work, we plan to relax this assumption and integrate bounded acceleration and deceleration into the optimal path finding model.

The following section 3 includes a more detailed and rigorous implementation of the presented approach.

3 Dynamic Programming Model and Optimal Path Planning Algorithm

In the preceding section 2 we discuss how to address various aspects of an optimal path finding problem: limited visibility horizon, mobile system dynamics restrictions (such as, sharpest feasible turns), and computational demand of the time-dependent environment. In this section, we integrate all the components discussed above into a single dynamic programming path finding model and provide an algorithm to facilitate its implementation.

Let $\tau(a, \theta_a, b, \theta_b, t_a)$ denote the travel time along a fastest path with bounded curvature (i.e., satisfying sharpest feasible turns constraints) from point a to point b starting with a heading angle θ_a at time t_a and arriving at b with a heading angle θ_b. We use the input parameters a and t_a to approximate the local region (within a distance l) as a stationary time and space homogeneous environment. As discussed in Section 2.2, we implement our earlier results for finding a fastest path with bounded curvature in a direction-dependent medium (see [14]) to evaluate an optimal path from the initial state (a, θ_a) at time t_a to the target state (b, θ_b), assuming that a vehicle moves with its maximum attainable speed. The value of $\tau(a, \theta_a, b, \theta_b, t_a)$ is equal to the travel time associated with the found path.

Next, we define $T(a, \theta_a, b, \theta_b, t_a)$ to be the smallest increment of time from the moment of arrival at point a at time t_a until arriving at the state (b, θ_b) (this definition is adapted from [16]). In other words, we allow a vehicle to leave point a at any time after arriving there at time t_a, with an objective to arrive at the state (b, θ_b) at the earliest possible time. Alternatively, we can define $T(.)$ as follows,

$$T(a, \theta_a, b, \theta_b, t_a) := \min_{\Delta_t \geq 0} \{\Delta_t + \tau(a, \theta_a, b, \theta_b, t_a + \Delta_t)\}. \quad (1)$$

It is important to note that in the minimization of (1), we have a natural upper bound on the value of Δ_t, which is equal to $\tau(a, \theta_a, b, \theta_b, t_a)$. Since $\tau(a, \theta_a, b, \theta_b, t_a)$ is the minimum travel time corresponding to a vehicle leaving point a without a delay, it is never advantageous to delay the departure longer than that time value. As a result, for each state of the system (e.g., (a, θ_a)) we only consider the departure times belonging to the interval $[t_a, t_a + \max_{(b,\theta_b)}\{\tau(a, \theta_a, b, \theta_b, t_a)\}]$. Alternatively, a traditional DP formulation of an optimal path finding problem in a dynamic network considers the departure times belonging to an interval $[t_a, T^*]$, where T^* is an upper bound on the agent's arrival time at the target point of the network (see [7] and [9] for examples). Since $T^* >> t_a + \max_{(b,\theta_b)}\{\tau(a, \theta_a, b, \theta_b, t_a)\}$ for most states, our model significantly decreases the computational time of the path finding algorithm. In the following section discussing the efficient implementation of our algorithm, we present an even tighter upper bound on Δ_t, which we use in practice.

We now present our dynamic programming functional equation. For each state $(a, \theta_a) : a \in \mathbb{R}^2, \|a - s\| \leq R_H$ and $\theta_a \in S^1$, define the optimal value function $g(a, \theta_a)$, such that $g(a, \theta_a)$ is the minimum travel time over all the paths from the initial position (s, θ_s) to point a, that start at time $t_0 = 0$ and arrive at a with the heading angle θ_a. Then, we have the following dynamic programming forward functional equation,

$$g(b, \theta_b) = \begin{cases} \min_{\{a,\theta_a : \|b-a\|=l\}}\{g(a, \theta_a) + T(a, \theta_a, b, \theta_b, g(a, \theta_a))\} \\ 0 \ \text{ for } \ (b, \theta_b) = (s, \theta_s) \end{cases} \tag{2}$$

Recursive application of the functional equation (2) delivers the fastest paths from (s, θ_s) to all the points in $\mathcal{N}_H$. To find these optimal paths, one can apply Dijkstra's algorithm [11], an A* algorithm [22] or any other efficient algorithms for a static network. However, we find the implementation of Dijkstra's method to be more advantageous since it automatically delivers the optimal paths to all the nodes in the network. Consequently, we obtain the fastest paths to all waypoints in $\mathcal{N}_H$ with one run of the dynamic programming algorithm.

Every optimal solution found using (2) includes an ordered set of states and an optimal wait time for each node along the path. However, our problem statement assumes that no stopping is permitted anywhere along the path. Thus, we use the found optimal wait times to calculate the speed for each arc, in order to ensure that a node arrival time is equal to the optimal node departure time.

To calculate the optimal speed, consider an arc belonging to an optimal path that connects the states (a, θ_a) and (b, θ_b). We are also given an optimal departure time for node a, denoted by t'_a, and an optimal wait time at node b, denoted by w_b. Then, when a vehicle leaves state (a, θ_a) at time t'_a and travels with the maximum attainable speed along a fastest path with bounded curvature (see [14]), it arrives at state (b, θ_b) at time $t'_a + \tau(a, \theta_a, b, \theta_b, t'_a)$. However, the optimal time to depart state (b, θ_b) is equal to $t'_a + \tau(a, \theta_a, b, \theta_b, t'_a) + w_b$. Consequently, we need to adjust the vehicle's travel time from (a, θ_a) to (b, θ_b) by a factor $\rho_{(a,\theta_a),(b,\theta_b)} \geq 1$, where

$$\rho_{(a,\theta_a),(b,\theta_b)} = \frac{\tau(a, \theta_a, b, \theta_b, t'_a) + w_b}{\tau(a, \theta_a, b, \theta_b, t'_a)}. \tag{3}$$

A vehicle then adjusts the speed (by varying engine revolutions per minute, or other controllers) to a $1/\rho_{(a,\theta_a),(b,\theta_b)}$ fraction of the maximum attainable speed in order to arrive at state (b, θ_b) precisely at the time equal to $t'_a + \tau(a, \theta_a, b, \theta_b, t'_a) + w_b$.

Note that the minimum turning radius function for the time and space homogeneous environment is only dependent on the heading angle and does not explicitly depend on the vehicle's speed. Therefore, slowing down along an arc does not affect the curvature constraint and the feasibility of the originally constructed path. For some applications, this assumption might not be realistic. We plan to relax this assumption in our future work. However, we would like to note that in most practical scenarios a slower speed of a vehicle results in smaller minimum turning radius, and does not restrict the feasibility of a path found with the maximum attainable speed. In the case when a slow speed results in a larger minimum turning radius (e.g., vessel navigation) the continuity of a feasible speed and system controllability (established in [14]) allows us to find a feasible path from (a, θ_a) to (b, θ_b) with the desired travel time for most practical applications.

The following Algorithm 1 summarizes our discussion and provides a concise step-by-step procedure for finding an optimal path from (s, θ_s) to (t, θ_t).

4 Application and Numerical Results

Algorithm 1 integrates the analysis presented in the preceding section of the paper, as well as the results from the author's earlier work [14, 15], into a single fastest-path finding model. In this section, we discuss the practical aspects of the algorithm implementation and demonstrate its application to a short-range ship routing in dynamic wave-field problem [13]. While we lead our discussion in the content of a vessel routing problem, the implementation techniques used to further improve the efficiency of the algorithm can be applied to any optimal path finding problem.

Algorithm 1 Fastest Path from (s, θ_s) to (t, θ_t).

Step 1. Apply algorithm from [14] to compute the values of $\tau(a, \theta_a, b, \theta_b, t_a)$ for all inputs where $\|a - s\| \leq R_H$, $\|b - s\| \leq R_H$ and $\|a - b\| = l$.

Step 2. Compute the smallest elapse of time function $T(a, \theta_a, b, \theta_b, t_a)$ using equation (1).

Step 3. Apply Dijkstra's algorithm to the DP recursive equation (2) to find the fastest paths from (s, θ_s) to all the points in $\mathcal{N}_H$.

Step 4. Depending on the specific problem, apply algorithms presented in [14] or [15] to find the fastest paths from all points in $\mathcal{N}_H$ to the target state (t, θ_t).

Step 5. Find the point in $\mathcal{N}_H$ that has the smallest sum of the corresponding travel times found in Step 3 and Step 4. A fastest path passing through such point is the optimal path.

Step 6. For the optimal path found in Step 5, adjust the speed for each arc by a factor $1/\rho$ defined in (3).

4.1 Efficient Implementation

The benefits of real-time data collection are limited to our ability to utilize it in timely manner, and the relatively small computational time of the optimal path finding algorithm is essential for real-time implementation. Here, we present some implementation-specific methods that ensure a time-efficient execution of Algorithm 1. We have aimed to improve the run-time of our code to the greatest of our ability; however, it is important to note that an expert computer scientist should be able to improve the efficiency of the code even further. The main purpose of the program developed as part of this work is to test the algorithm and the found optimal paths, which we successfully achieve.

One of the main techniques to speed up the run-time of the algorithm is to pre-process as much of computation as possible. Thus, we perform a portion of the calculations off-line, before the ship starts its travel. In fact, a set of calculations need to be evaluated only once for a specific vessel, and the results are then stored in the form of a look-up table as part of the navigation software.

Observe that the values of the travel time function $\tau(a, \theta_a, b, \theta_b, t_a)$ in Step 1 of the algorithm can be pre-computed for all possible stationary distributions of a local environment. Thus, we use the input parameters a and t_a to evaluate and then characterize the local medium (e.g., sea state characterizing local waves distribution). Because the distance between a pair of consecutive waypoints is fixed to the value of l, for a given sea state, only values of θ_{ab}, θ_a and θ_b are needed to look up the value of the $\tau()$ function in a pre-computed table.

Similarly to Step 1 of the algorithm, we can pre-compute Step 4 either for all possible global sea state values or perform the calculations shortly before the start of a particular trip when the distribution of the global environment becomes available. Unlike evaluating the values of the $\tau()$ function in Step 1, the distance between points in $\mathcal{N}_H$ and the target point t is not fixed, however the characterization of paths found in Step 4 stays the same regardless of that distance. This is especially true when we choose to implement one of the algorithms presented in [15], neglecting the minimum turning radius.

In addition, we partially pre-compute Step 2 of the algorithm, by evaluating when for some instances of the function $\tau(a, \theta_a, b, \theta_b, t_a)$ the delay is not beneficial. For example, for some sea states we observe the minimum value of the function, implying that any other departure time (possibly corresponding to a different distribution of the local environment) would never result in a smaller travel time making delay suboptimal. We also evaluate an upper bound on the possible decrease in travel time by comparing a given value of the $\tau()$ function for the best case scenario, which provides a tighter bound on the interval of Δ_t values considered in the minimization of equation (1).

Another effective technique for decreasing the run-time of our algorithm is parallel computing, where a number of calculations are carried out simultaneously. While our test runs are performed on a single processor, there is a potential run-time improvement when multiple processors are used for the implementation.

4.2 Computational Results

We test the performance of our path-finding algorithm on a short-range ship routing problem with the objective to find a fastest path between given starting and target positions. We use existing computer models to simulated the time-varying wave-field and the S-175 containership behavior among the waves. First, we generate a random wave-field region by specifying the wave distribution parameters corresponding to a desired sea state. The resulting wave surface instance serves as the initial setting for our test case. Then, the wave propagation model [32, 33] forecasts the evolution of the created wave-field over a time interval, i.e., it incorporates the natural propagation and interaction of the waves to predict how the initial wave surface changes. Next, the evolving wave-field data is used to produce a ‘local sea state’ map for the location and time dependent environment. That is, for each waypoint we approximate the waves within l-distance radius by a stationary wave-distribution. As a result, we obtain a lookup table that delivers the local sea state parameter for each waypoint and time instance considered by the dynamic programming algorithm (i.e., for each DP state).

Added drag model evaluates the forces acting on the ship by the surrounding waves to derive the maximum attainable speed ($V(a, \theta_a, t_a)$) table. This table provides the maximum vessel speed that can be achieved for each local sea state (wave distribution) and heading angle without violating the operability constraints (e.g., maximum root-mean-squared roll and probability of wet deck). (See [12] for the detailed discussion and computation of the speed values and operability constraints.) Our colleague, Li [27], uses Fossen's vessel model ([21]) to provide the minimum turning radius ($R(a, \theta_a, t_a)$) table specified for each value of local sea state and ship heading angle. The speed and turning radius input tables are used to compute the $\tau()$ values for each pair of consecutive waypoints. That is, we create a lookup table that, for the given sea state and ship heading angle, returns the next set of DP states (i.e., each waypoint location and heading angle) and the cost of traversing to each of those states. Note that for a specific ship type these calculations are done only once, and they do not contribute to the run-time of our path finding algorithm.

4.2.1 Algorithm Run-Time

We implement our algorithm using MATLAB to ease the communication and compatibility with the ship motion prediction and control models. MATLAB is considered to be more user friendly than other programming languages, and it is a preferred language in many engineering fields. However, it is important to note that C++ is well known to have a significantly faster run-time, and experienced computer scientists report improvement by factors ranging between 30 and 50. For example, in order to speed up our code, we integrate a sorting function (the major component of Dijkstra's algorithm) written using C++ language into our MATLAB code. This modification decreased the sorting time of our program by a factor greater than ten. We anticipate that all parts of the navigation system will be translated into C++ for implementation in real-life, and we expect the run-time of our algorithm to decrease significantly.

All the test runs are performed on a PC machine with the Microsoft Windows Vista operating system and a single 2.9 GHz processor. For each run we record the number of DP states that have to be *explored* by the algorithm in order to find an optimal path. A dynamic programming state is considered to be *explored* when the algorithm finds the minimum cost associated with reaching that state and updates the optimal value function $g(a, \theta_a)$ for the states that immediately follow it (i.e., its direct successors). Each of our test runs explored between 57,000 and 132,000 DP states with the corresponding run-time ranging linearly between

105 and 413 seconds. As mentioned earlier, the run-time is expected to decrease significantly once the algorithm is reprogrammed using C++ language. After the acceleration of the sorting function using C++, the current program spends approximately 65% of the time performing other computations in MATLAB. Based on our experience with sorting, the conservative estimate is that we can decrease the time required to perform those computations by 10 times, resulting in the overall decrease in run-time by a factor of 2.5. In addition, faster processors are currently available on the market with speeds of up to 3.7 GHz, which would correspond to a potential decrease in the computational time by at least 20%.

4.2.2 Optimal Path

One of the driving forces behind the development of innovative sensor and data-processing technology is to provide users with more detailed and accurate information in order to help them make better decisions. Furthermore, the ability to collect, process and use information about the environment surrounding a vehicle in a computer-friendly form facilitates the use of autonomous navigation systems, which frees-up human time and allows them to shift their attention to more critical tasks. These goals are consistent with the short-range ship routing problem that motivated our research. Thus, one of the main objectives of our numerical test runs is to evaluate and quantify the benefits of the information collected by the on-board radar. To achieve this, we compare the travel times for paths found by the presented algorithm (incorporating the wave-field data provided by the radar) with those for paths that are chosen when the information about the environment (waves) surrounding the ship is unavailable.

To evaluate the improvement in vessel travel time when Algorithm 1 is implemented, we test our model in the simulated wave-field corresponding to sea state number 6.5. The average wave height of the highest one-third of the waves (called *significant wave height*) for such sea state is equal to 7 meters, and the peak wave period is equal to 15 seconds. The global sea state characterizing the waves beyond the radar visibility horizon is also set to 6.5 to ensure consistency when comparing the travel times for various paths. The test S175 containership is 175 meters long with the minimum turning radius values ranging between 290 and 305 meters. The conservative estimate of the visibility horizon for the radar currently in development is approximately 1.5 to 2 miles, equivalent to approximately 2,500 to 3,000 meters. Thus, the wave-field region generated for our test runs corresponds to the current radar capabilities (i.e., $R_H = 2500$ meters).

The predominant applications of interest that motivated our research involve short range trips, and we consider a set of target points approximately 18,000 meters away from the ship current location. The target points are spread out such that the directions from s to t span 360 degrees with a 20-degree increment, in order to capture the direction-dependent nature of our problem and its effect on the found optimal paths. The distance between a pair of consecutive waypoints is set to 250 meters ($l = 250$), and we consider all consecutive states that can be reached from the predecessor without leaving the local sea state region (i.e., the circle with radius equal to l). In Step 4 of Algorithm 1, we choose to apply an algorithm from [15] to compute the path travel time through the radar invisible region beyond R_H.

For each considered target point we run the MATLAB code to find an optimal path. The resulting travel time is then compared to that of the following three benchmark paths that one might follow when the detailed information about surrounding waves is not available (see Figure 4).

Benchmark path 1 (p_1): a straight line path from starting point s to the target point t;

Benchmark path 2 (p_2): a one-waypoint (or two line segments) path from s to t found using results from [15] where vessel makes a left turn at the waypoint; and

Benchmark path 3 (p_3): a one-waypoint path from s to t found using results from [15] where vessel makes a right turn at the waypoint.

Results of [15] show that the one-waypoint paths p_2 and p_3 are optimal for the time and space homogeneous environment. Therefore, by comparing the optimal path to these benchmark paths, we observe the travel time improvement one achieves by integrating the collected wave-field information into the path planning process.

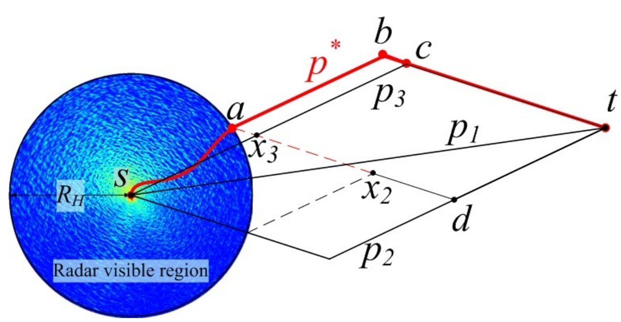

Figure 4: Example of an optimal path p^*, benchmark paths p_1, p_2 and p_3, and the merge points x_2 and x_3.

To make the comparison of travel times as accurate as possible, we set the initial heading of the vessel to be used in Algorithm 1 equal to the direction from the starting to the target point, denoted by θ_{st}. Nevertheless, since our dynamic programming model integrates the minimum turning radius constraint while neglecting it in the travel time calculations for paths p_2 and p_3, the actual travel time for the benchmark paths has to be greater than our estimates, resulting in even more significant improvement in the minimum travel time than we report.

The same algorithm (Algorithm 1 from [15]) is used to find the part of an optimal path beyond the radar visibility horizon, as well as to find the benchmark paths p_2 and p_3. Consequently, a large part of these three paths is identical and has the same travel time, especially due to the time and space homogeneous environment assumed beyond R_H (see Figure 4). This is particularly true when the radar visibility horizon is small relative to the distance between the starting and target points. In such a case, the directions from all the points on the boundary of R_H to the target point t are very close to each other, implying that the radar-invisible region part of the optimal path and of the alternative paths p_2 and p_3 consist of the linear segments with the same heading angles (see Figure 7). To accurately quantify the decrease in travel time for an optimal path over the benchmark paths, we compare only the portions of the paths that are distinct among the paths. We define the *merge points* x_2 and x_3 (corresponding to paths p_2 and p_3, respectively) that lie in the time and space homogeneous region outside R_H. These points are defined such that paths p_2 and p_3, differ from an optimal path between the start point s and the corresponding merge points, and the benchmark paths are the same as the optimal path for the remaining part of travel. For example, note on Figure 4 that traveling along the paths ab, x_3c and dt correspond to the same travel times when the wave-field is time and space homogeneous. Thus, in comparing the minimum travel time to the paths p_2 and p_3, we only use travel times of each path between the start point s and the corresponding merge points. Recall that as the vessel moves along a path, the new wave-field information is collected and an optimal path is reevaluated. Therefore, by the time a vessel reaches a merge point, the paths are reevaluated and we expect to see a comparable benefit of following a new optimal path for the next part of the path. The continuous reevaluation of an optimal path justifies the comparison of travel times only for the first part of paths leading up to the merge point.

In [15] we have established that in some scenarios one-waypoint paths are not optimal for time and space homogeneous environment, and paths p_2 and p_3 instead correspond to a single line segment. An approach similar to merge points is used to compare the optimal travel time in such cases. When the heading angle of a benchmark straight line path is similar to that of the optimal path beyond R_H, we consider an alternative target point (equivalent to a merge point) that is closer to the boundary of the radar visible region and compare the paths from s to this new point. See Figure 5 and Figure 6 at the end of this section for an illustration.

We now report the numerical results. Columns titled "R_{min}" in Table 1 summarize the results of our test runs. The angle between starting and target points is denoted by θ_{st} and the minimum travel time is represented by $t(p^*)$. We let $t(p_i)$ for $i = 1, 2, 3$ denote the trave time along a path p_i from the starting point s to the corresponding merge point as discussed above. For consistency, it is implied that $t(p^*)$ is the travel time along an optimal path between the same pair of points (start and merge), when comparison is conducted. The values in the third column ($t(p^*)/t(p_1)$) that are labeled by '*' correspond to the cases when the total travel time from s to t are compared, as considering a closer target point would alter the structure of an optimal path. In such cases, a large part of both paths passes through the assumed time and space homogeneous environment, and the improvement values presented in the table are significantly lower than expected to be observed during the real-life implementation. The dashes ($-$) in Table 1 correspond to the cases when all three benchmark paths are identical and we only report the travel time improvement corresponding to a straight line path (p_1).

Based on our analysis, we observe up to 9.7$\%$ improvement and on average between 4$\%$ and 6$\%$ improvement, in comparison to implementing the paths optimal for time and space homogeneous environment while neglecting the wave-field data collected by the radar. The improvement reported here is expected to be higher in the real-life applications, since the current models simulating the wave-field and vessel dynamics (e.g., speed) are very restrictive and do not fully capture the non-homogeneity of the system. In the following subsection, we discuss in detail the limitations of the data available to us for the numerical test runs and their adverse effect on the reported travel time improvement.

Also, note that the travel time decrease presented in Table 1 is a conservative estimate, since our analysis neglects the bounded curvature for paths p_1, p_2 and p_3. In addition, the initial heading of the vessel is set to the direction θ_{st}, and the found optimal paths involve less vessel maneuvers than we would see for other starting heading angles. However, it is important to remember that one of the advantages of our dynamic programming model is the integration of heading angle and turning radius constraints of the vessel into the path finding model. *Therefore, the presented path-finding model not only improves the travel time, but it*

		R_{min}			$0.5R_{min}$		
Run No.	θ_{st} (deg)	$t(p^*)/t(p_1)$	$t(p^*)/t(p_2)$	$t(p^*)/t(p_3)$	$t(p^*)/t(p_1)$	$t(p^*)/t(p_2)$	$t(p^*)/t(p_3)$
1	0	0.9994	-	-	0.9808	-	-
2	20	0.9418	-	-	0.9551	-	-
3	40	0.9125	0.9476	0.9473	0.8811	0.8576	0.8686
4	60	0.9846*	0.9334	0.9230	0.8592	0.8348	0.8468
5	80	0.9456	-	-	0.9544	-	-
6	100	0.9835*	0.9377	0.9684	0.9411	0.8942	0.8854
7	120	0.9796*	0.9516	0.9605	0.9041	0.9036	0.9014
8	140	0.9650	-	-	0.8930	-	-
9	160	0.9033	-	-	0.8189	-	-
10	180	0.9890	0.9601	0.9506	0.8946	0.8762	0.9304
11	200	0.9307	-	-	0.8820	-	-
12	220	1.0000	-	-	0.8480	-	-
13	240	0.9634*	0.9488	0.9497	0.8118	0.7934	0.8218
14	260	0.9822*	0.9479	0.9593	0.9026	0.8849	0.9210
15	280	0.9823	-	-	0.9600	-	-
16	300	0.9882*	0.9350	0.9495	0.8458	0.8112	0.8112
17	320	0.9838*	0.9458	0.9526	0.7800	0.8053	0.8047
18	340	0.9433	-	-	0.9266	-	-

Table 1: Comparison of minimum travel time to the trave times for paths p_1, p_2 and p_3.

also delivers a control-feasible path for any initial and target positions of the vessel.

The following Figure 5, Figure 6, Figure 7 and Figure 8 illustrate examples of the found optimal paths and the alternative benchmark paths.

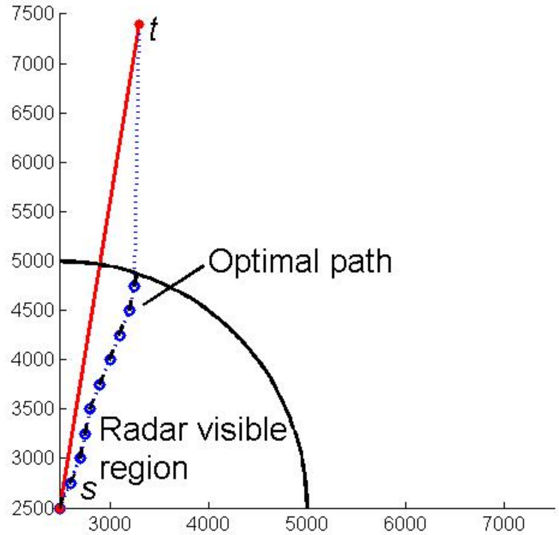

Figure 5: Test run number 5, where $\theta_{st} = 80$ degrees.

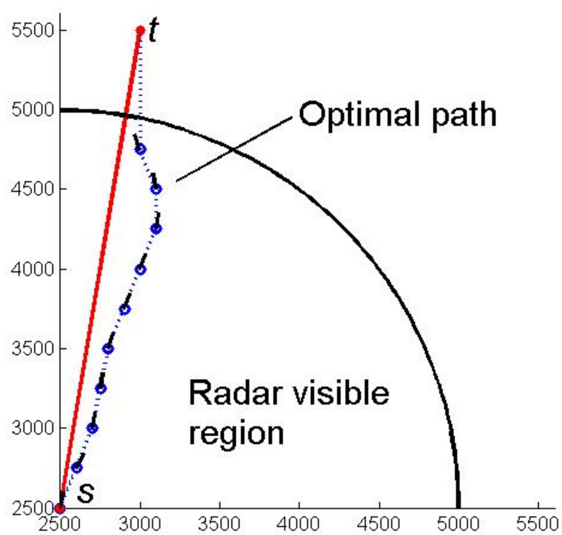

Figure 6: Test run number 5 with the target point relocated closer to the radar visible region.

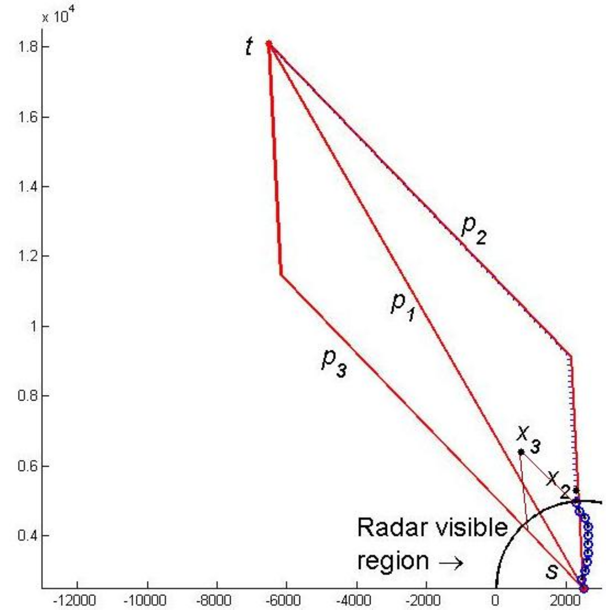

Figure 7: Test run number 7, where $\theta_{st} = 120$ degrees. The merge points for paths p_2 and p_3 are denoted by x_2 and x_3, respectively.

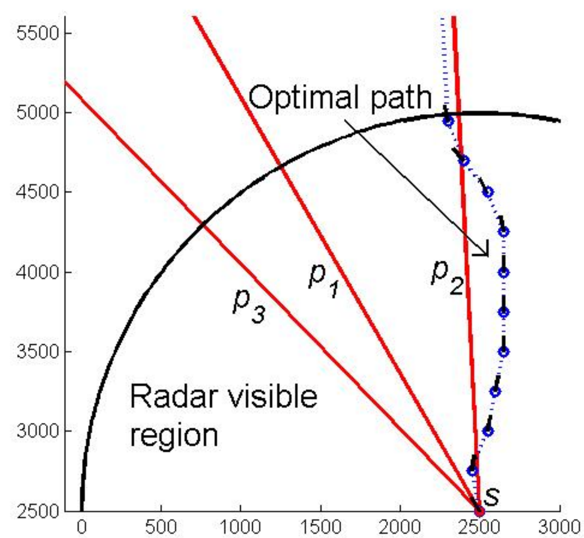

Figure 8: A detailed figure for test run number 7, illustrating the paths within the radar visible region.

4.2.3 Limitations of the Data Available for Numerical Analysis

The results of test runs summarized in Table 1 (columns $3-5$) illustrate an improvement in the vessel travel time for 16 out of 18 cases, corresponding to a decrease ranging between 1.8% and 9.7%. However, due to the limitations of the data available to us for the numerical analysis, the reported benefits of our path finding model integrating the real-time radar information are conservative, and we anticipate observing a greater decrease in vessel travel time in practice. In this section, we discuss the limitations of the data available for numerical results and their effects on our analysis.

The analysis is conducted for a large 175-meter long container ship that has relatively slow speed (ranging between 12 knots and 22.15 knots, or equivalently 6.2 - 11.4 meters per second) and a large turning radius (ranging between 290 and 305 meters). Since an average wave length for the considered sea state (No. 6.5) is equal to 90 meters, the vessel is not maneuverable enough to navigate around the large waves. Consequently, the presented path finding algorithm is not used to its fullest potential. While all the gathered information about the surrounding wave-field is integrated into the path planning process, a significant amount of that information has no effect on the navigation for such a large vessel. For comparison, we scale the minimum turning radius of a vessel by a factor of 0.5 and recompute the optimal paths for our test runs (in addition, we set $R_H = 5000$ meters and $l = 150$). The results for the new runs can be seen in Table 1 (columns $6-8$, titled "$0.5R_{min}$"). The minimum travel time for the new optimal path corresponds to the improvement for all 18 test runs ranging between 2% and 22% with an average improvement of 12.5%. In addition to improved maneuverability, smaller vessels are also more suspectable to the effects of individual waves, further increasing potential benefits of navigation around each wave.

In addition to limited maneuverability of the vessel, the model currently used to compute the added drag and corresponding vessel speed reduction is not as detailed and accurate as the model we anticipate to integrate in the future. The maximum attainable speed function is currently averaged over a distribution of waves that a vessel is expected to observe for a specified sea state and heading angle. Therefore, it does not explicitly incorporate the effect an individual wave has on the vessel. We expect that a more intricate vessel motion model will result in more complex optimal paths and greater time savings, especially so for the lower sea states (calmer waves).

To compare the minimum travel time to the benchmark paths, we use a simulation program to generate a realization of a time-evolving wave-field. The input to the program is a set of parameters characterizing the distributions of the waves, which results in the simulation generating a stationary wave-field. Subsequently, the large vessel turning radius and limitations of the vessel speed prediction model reduce our problem to optimal path finding in an ergodic system. This overall 'averaging' of the waves encountered by a vessel limits an improvement in travel time observed by our model. The main advantage of real-time sensor data collection is that the environment does not have to be assumed or approximated by a stationary distribution. Therefore, in real-life applications, we expect to see greater benefits of navigating around rough parts of the sea, and our path finding model would result in a greater decrease in travel time.

The estimates of travel time improvement presented in Table 1 are also understated due to the fact that the minimum turning radius restriction is integrated into the optimal path, while the constraint is relaxed for the benchmark paths p_1, p_2 and p_3. Integration of the turning constraint for the alternative paths would increase their travel times and the corresponding improvement of the optimal path, especially when the initial heading of a vessel (θ_s) is not aligned with the position of the target point (t).

5 Conclusion

In this paper, we relax the time and space homogeneous assumption of our previous work and deliver an efficient model for finding an optimal path in a direction, location and time dependent environment. We address the problems where real-time detailed information about the environment is dynamically revealed to the vehicle and develop a navigation model that is capable to integrate this information. Our dynamic programming model integrates a number of aspects that are missing from the traditional path finding DP models. First, we address the limitations of the sensor and data collecting technology, which are associated with the reduced information available beyond a specified visibility horizon, by applying our earlier results for a time and space homogeneous environment. Second, to incorporate the dynamic constraints of the system, such as sharpest feasible turns of a vehicle, we employ our algorithm for optimal paths with bounded curvature and evaluate an optimal arc to traverse between each pair of DP states. Finally, by allowing the vehicle to adjust its speed along a path, we are able to eliminate the time variable from the dynamic programming state space and significantly improve the run-time of our algorithm. We demonstrate the application and numerical results for our motivating problem, a short-range ship routing in dynamic wave-field.

The presented model makes a number of assumptions that we plan to relax in future work. For example, constraints on maximum acceleration and deceleration have to be introduced into the model to bound the feasible change in vehicle speed along a path. In addition, current model does not consider the case when the sharpest feasible turn of a vehicle not only depends on the heading angle, location and time, but also on the vehicle speed, thus expanding the set of feasible paths. Our current algorithm also assumes perfect sensor information and forecasting models, which result in a deterministic problem setting. In practice, there are a number of factors that introduce noise and error into the information available to navigation system about the surrounding environment. We plan to analyze and characterize the types of uncertainty observed in such problems and study their stochastic versions. One of the motivations for work presented in this paper is the development of navigation systems for unmanned and autonomous vehicles, and further study of broader objective functions is needed to improve practicality of such systems. We are currently adopting our algorithm to non-additive objective functions, as well as problems without explicitly defined target states (e.g., surveillance and exploration) to achieve this goal.

References

[1] Jeffrey M. Alden and Robert L. Smith. Rolling horizon procedures in nonhomogeneous Markov decision processes. *Oper. Res.*, 40(suppl. 2):S183–S194, 1992.

[2] Toby Allsopp, Andrew Mason, and Andy Philpott. Optimal sailing routes with uncertain weather. In *Proceedings of The 35th Annual Conference of the Operational Research Society of New Zealand*, pages 65–74, December 2000.

[3] Richard Bellman. *Dynamic programming*. Princeton University Press, Princeton, N. J., 1957.

[4] Jean-Daniel Boissonnat, André Cérézo, and Juliette Leblond. A note on shortest paths in the plane subject to a constraint on the derivative of the curvature. Research Report RR-2160, INRIA, 1994.

[5] Arthur E. Bryson, Jr. and Yu Chi Ho. *Applied optimal control*. Hemisphere Publishing Corp. Washington, D. C., 1975. Optimization, estimation, and control, Revised printing.

[6] Xuân-Nam Bui, Philippe Souères, Jean-Daniel Boissonnat, and Jean-Paul Laumond. Shortest path synthesis for Dubins non-holonomic robot. In *Proceedings of the 11th IEEE Internatational Conference on Robotics Automation*, pages 2–7, 1994.

[7] Ismail Chabini. Discrete dynamic shortest path problems in transportation applications: Complexity and algorithms with optimal run time. *Transportation Research Record*, 1645:170–175, 1998.

[8] Hamidreza Chitsaz and Steven M. LaValle. Time-optimal paths for a dubins airplane. In *Proceedings of the 46th IEEE Conference on Decision and Control*, pages 2379–2384, New Orleans, LA, December 2007.

[9] Kenneth L. Cooke and Eric Halsey. The shortest route through a network with time-dependent internodal transit times. *J. Math. Anal. Appl.*, 14:493–498, 1966.

[10] Eric V. Denardo. *Dynamic programming*. Dover Publications Inc., Mineola, NY, 2003. Models and applications, Corrected reprint of the 1982 original.

[11] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1):269–271, December 1959.

[12] I. S. Dolinskaya, M. Kotinis, M. G. Parsons, and R. L. Smith. Optimal short-range routing of vessels in a seaway. *Journal of Ship Research*, 53(3):121–129, September 2009.

[13] Irina S. Dolinskaya. *Optimal Path Finding in Direction, Location and Time Dependent Environments*. PhD thesis, University of Michigan, 2009.

[14] Irina S. Dolinskaya and Alvaro Maggiar. Time-optimal trajectories with bounded curvature in anisotropic medium. *Under Review, available at http://users.iems.northwestern.edu/ dolira/publications.html*.

[15] Irina S. Dolinskaya and Robert L. Smith. Path planning in an anisotropic medium. *Under Review, available at http://users.iems.northwestern.edu/ dolira/publications.html*.

[16] Stuart E. Dreyfus. An appraisal of some shortest-path algorithms. *Operations Research*, 17(3):395–412, 1969.

[17] L. E. Dubins. On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. *Amer. J. Math.*, 79:497–516, 1957.

[18] O. M. Faltinsen. *Sea loads on ships and offshore structures / O.M. Faltinsen*. Cambridge University Press, Cambridge ; New York :, 1990.

[19] Frank D. Faulkner. A general numerical method for determining optimum ship routes. *Navigation*, 10(2):143–148, 1963.

[20] Frank D. Faulkner. Numerical methods for determining optimum ship routes. *Navigation: Journal of The Institute of Navigation*, 10(4):351–367, Winter 1963.

[21] Thor I. Fossen. *Guidance and Control of Ocean Vehicles*. John Wiley & Sons Ltd., 1994.

[22] P. E. Hart, N. J. Nilsson, and B. Raphael. Correction to "a formal basis for the heuristic determination of minimum cost paths". *SIGART Newsletter*, 37:28–29, 1972.

[23] Joel T. Johnson, Robert J. Burkholder, Jakov V. Toporkov, David R. Lyzenga, and William J. Plant. A numerical study of the retrieval of sea surface height profiles from low grazing angle radar data. *IEEE Transactions On Geoscience and Remote Sensing*, 47(6):1641–1650, June 2009.

[24] David E. Kaufman and Robert L. Smith. Fastest paths in time-dependent networks for intelligent vehicle-highway systems application. *IVHS Journal*, 1(1):1–11, 1993.

[25] J. C. Kimball and H. Story. Fermat's principle, Huygens' principle, Hamilton's optics and sailing strategy. *European Journal of Physics*, 19:15–24, January 1998.

[26] Chung-Yee Lee and Eric V. Denardo. Rolling planning horizons: Error bounds for the dynamic lot size model. *Mathematics of Operations Research*, 11(3):423–432, 1986.

[27] Zhen Li. *Path Following with Roll Constraints for Marine Surface Vessels in Wave Fields*. PhD thesis, University of Michigan, 2009.

[28] W. Marks, T. R. Goodman, Jr. W. J. Pierson, L. J. Tick, and L. A. Vassilopoulos. An automated system for optimum ship routing. *Transactions - The Society of Naval Architects and Marine Engineers*, 76:22–55, 1968.

[29] Timothy G. McGee, Stephen Spry, and J. Karl Hedrick. Optimal path planning in a constant wind with a bounded turning rate. In *Proceedings of the AIAA Conference on Guidance, Navigation and Control*, Ketstone, Colorado, August 2006.

[30] Rachelle L. McNeely, Ram V. Iver, and Phillip R. Chandler. Tour planning for an unmanned air vehicle under wind conditions. *Journal of Guidance, Control, and Dynamics*, 30(5):1299–1306, September-October 2007.

[31] NOAA. National oceanic and atmospheric administration. website: www.noaa.gov, 2011.

[32] Okey G. Nwogu. Interaction of finite-amplitude waves with vertically-sheared current fields. *Journal of Fluid Mechanics*, 627:179–213, 2009.

[33] Okey G. Nwogu and David R. Lyzenga. Surface wavefield estimation from coherent marine radars. *IEEE GEOSCIENCE AND REMOTE SENSING LETTERS*, 7(4):631 – 635, October 2010.

[34] Ariel Orda and Raphael Rom. Shortest-path and minimum-delay algorithms in networks with time-dependent edge-length. *J. Assoc. Comput. Mach.*, 37(3):607–625, 1990.

[35] John Osborne and Rolf Rysdyk. Waypoint guidance for small UAVs in wind. In *Proceedings of the American Institute of Aeronautics and Astronautics Infotech@Aerospace Conference*, Arlington, VA, 2005.

[36] I. M. Ovacikt and R. Uzsoy. Rolling horizon algorithms for a single-machine dynamic scheduling problem with sequence-dependent setup times. *International Journal of Production Research*, 32(6):1243–1263, 1994.

[37] Nikiforos A. Papadakis and Anastassios N. Perakis. Deterministic minimal time vessel routing. *Oper. Res.*, 38(3):426–438, 1990.

[38] Anastassios N. Perakis and Nikiforos Papadakis. New models for minimal time ship weather routing. *Society of Naval Architecture and Marine Engineering Transactions*, 96:247–269, 1988.

[39] Anastassios N. Perakis and Nikiforos A. Papadakis. Minimal time vessel routing in a time-dependent environment. *Transportation Sci.*, 23(4):266–276, 1989.

[40] A. B. Philpott. Stochastic optimization and yacht racing. In *Applications of stochastic programming*, volume 5 of *MPS/SIAM Ser. Optim.*, pages 315–336. SIAM, Philadelphia, PA, 2005.

[41] A. B. Philpott, R. M. Sullivan, and P. S. Jackson. Yacht velocity prediction using mathematical programming. *European Journal of Operational Research*, 67(1):13–24, May 1993.

[42] Andy B. Philpott and Andrew Mason. Optimising yacht routes under uncertainty. In *The 15th Cheasapeake Sailing Yacht Symposium*, 2001.

[43] William J. Plant, William C. Keller, and Kenneth Hayes. Simultaneous measurement of ocean winds and waves with an airborne coherent real aperture radar. *Journal of Atmospheric and Oceanic Technology - Special Section*, 22:832 – 846, July 2005.

[44] J. A. Reeds and L. A. Shepp. Optimal paths for a car that goes both forwards and backwards. *Pacific J. Math.*, 145(2):367–393, 1990.

[45] P. Souères and J. D. Boissonnat. Optimal trajectories for nonholonomic mobile robots. In Jean-Paul Laumond, editor, *Robot Motion Planning and Control*, pages 93–170. Springer, 1998.

[46] Philippe Souères and Jean-Paul Laumond. Shortest paths synthesis for a car-like robot. *IEEE Trans. Automat. Control*, 41(5):672–688, 1996.

[47] Hctor J. Sussmann and Guoqing Tang. Shortest path for the Reeds-Shepp car: A worked out example of the use of geometric techniques in nonlinear optimal control. Technical Report SYCON-91-10, Rutgers Center for Systems and Control, September 1991.

[48] E. Zermelo. Über das navigationsproblem bei ruhender oder veränderlicher windverteilung. *Zeitschrift für Angewandte Mathematik und Mechanik*, 11(2):114–124, 1931.