Route Optimization Under Uncertainty for Unmanned Underwater Vehicles

by

Jacob Roy Cates

B.S, Mathematics
United States Naval Academy (2009)

Submitted to the Sloan School of Management
in partial fulfillment of the requirements for the degree of

MASTERS OF SCIENCE IN OPERATIONS RESEARCH

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2011

Author ....
Operations Research Center
May 20, 2011

Certified by ....
Dr. Michael J. Ricard
Distinguished Member of the Technical Staff
Autonomous Mission Control
Charles Stark Draper Laboratory
Technical Supervisor

Certified by ....
Professor Patrick Jaillet
Dugald C. Jackson Professor
Department of Electrical Engineering and Computer Science
Co-Director, Operations Research Center
Thesis Supervisor

Accepted by ....
Professor Dimitris J. Bertsimas
Boeing Professor of Operations Research
Co-Director, Operations Research Center

THIS PAGE INTENTIONALLY LEFT BLANK

Route Optimization Under Uncertainty for Unmanned Underwater Vehicles

by

Jacob Roy Cates

B.S, Mathematics
United States Naval Academy (2009)

Submitted to the Sloan School of Management
in partial fulfillment of the requirements for the degree of

MASTERS OF SCIENCE IN OPERATIONS RESEARCH

ABSTRACT

As our technology continues to increase, our military wishes to reduce the risk service members endure by using unmanned vehicles. These unmanned vehicles will, over time, become more independent and trustworthy in our operational military. The goal of this thesis is to improve the intelligence an unmanned vehicle has in its decision making to keep up with ever increasing capabilities.

In this thesis, we assume an Unmanned Underwater Vehicle (UUV) is given tasks and must decide which ones to perform (or not perform) in which order. If there is enough time and energy to perform all of the tasks, then the UUV only needs to solve a traveling salesman problem to find the best order. We focus on a tightly constrained situation, where the UUV must choose which tasks to perform to collect the highest reward.

In prize collecting traveling salesman problems, authors are often dismissive about stochastic problem parameters, and are satisfied with using expected value of random variables. In this thesis a more rigorous probabilistic model is formulated which establishes a guaranteed confidence level for the probability the UUV breaks mission critical time and energy constraints. The formulation developed takes the stochasticity of the problem parameters into account and produces solutions which are robust.

The thesis first presents a linear programming problem which calculates the transition probabilities for a specific route. This linear programming problem is then used to create a constraint which forces the UUV to choose a route that maintains an appropriate confidence level for satisfying the time and energy constraints. Once the exact model is created, heuristics are discussed and analyzed. The heuristics are designed to provide "good" solutions for larger sized problems and maintain a relatively low run time.

Technical Supervisor: Dr. Michael J. Ricard
Title: Distinguished Member of the Technical Staff
Autonomous Mission Control
Charles Stark Draper Laboratory

Thesis Adviser: Professor Patrick Jaillet
Title: Dugald C. Jackson Professor
Department of Electrical Engineering and Computer Science
Co-Director, Operations Research Center

Acknowledgments

First to my readers, thank you for your time and interest. For any comments, questions, or corrections I can be contacted with the email address jcates@alum.mit.edu

I am thankful for the opportunity I have had here at MIT. I thank the U.S. Navy and the U.S. Naval Academy for choosing and trusting me to represent them at MIT for two years. Thank you Draper Laboratories and the Naval Undersea Warfare Center for funding my graduate education, I hope that my research is beneficial to both of you.

I am grateful for the superb mathematical background I received from the USNA Math Department over 4 years. I was able to hit the ground running at MIT and never had to deal with filling holes in my undergraduate education.

To my adviser at Draper Labs, Dr. Ricard, thank you for all of the guidance and support you have provided, for trusting me to do work on my own time even though it tended to be at odd hours in the night, and for giving me the freedom to find an area of research which interested me.

To my MIT adviser, Prof. Jaillet, thank you for the feedback and insight you provided even with my strange affinity for variables and complicated notation. The day you suggested Time Expanded Networks for my research was the day I became confident I was on the path to writing a useful thesis with original ideas.

Thank you to all my friends who supported me throughout my time at MIT. The weekend visits to peel me away from work were much appreciated. A special thanks to T, whose IT support made my life much easier.

Finally, thank you to my wonderful family. I would not be here today without the endless love and support from my Mom and Dad. Thank you to my brothers Danny and Cory for being my role models and for giving me my competitive spirit, which helped me win consecutive intramural football championships at MIT.

Jacob Roy Cates, Ensign, US Navy, May 20, 2011

Assignment

Draper Laboratory Report Number T-1690

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Navy, Department of Defense, of the United States of America.

This thesis was prepared at The Charles Stark Draper Laboratory, Inc. under Sponsored Research Projects 24586-001 and 25607-001 (Advanced Autonomy).

The author hereby grants to the Massachusetts Institute of Technology and Charles Stark Draper Laboratory Inc. permission to reproduce and to distribute publicly paper and electronic copies of this thesis document in whole or in part.

Jacob Roy Cates, Ensign, US Navy, May 20, 2011

THIS PAGE INTENTIONALLY LEFT BLANK

Contents

List of Figures

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 1

Introduction and Background

The application of unmanned vehicles continues to expand and grow as new capabilities are demonstrated. The vision for these unmanned vehicles includes roles such as surveillance, and reconnaissance, as well as mine countermeasures. Specific missions, such as minefield detection and clearance as well as improvised explosive devise disposal are roles that are perfectly suited for unmanned vehicles, as it reduces risk to personnel. In some instances, such as mine hunting, unmanned vehicles are capable of performing the role faster and with greater accuracy than humans.

The Navy Unmanned Underwater Vehicle (UUV) Master Plan [**17**] identifies several areas where research and development continues to be required, and specifically discusses the development of autonomy as a required research area. To ensure that the U.S. Navy maintains sea superiority, the development and employment of the technologies surrounding the unmanned vehicles must continue at a pace to meet the expected roles.

1.1. Problem Motivation

For aerial, and terrestrial unmanned vehicles, communication is capable to allow for human guidance. For UUVs communication is only available when the vehicle or its antenna is above the surface of the water. This requires UUVs to have a higher level of autonomy as compared to its counterparts. Along with a higher mandatory level of autonomy comes a greater level of required trust. A UUV must be trusted to make the right decisions, and follow specific guidelines with limited guidance. The most important axiom a UUV must follow is it must not run out of the time and energy it has available. When a UUV is deployed for a mission, it is given strict guidelines as to when and where to rendezvous for retrieval. If a UUV does not reach the appropriate rendezvous point, manned ships are left guessing as to whether the UUV is late, or if it will return at all. This could have catastrophic results such as adding significant risk to manned units. For this reason, a UUV must be able to navigate through the mission in a way that it can be sure it reaches the rendezvous point without running out of time or energy.

A UUV is given a number of tasks to complete and must choose which tasks to perform in which order to be as efficient as possible with its time and energy. This thesis focuses on finding routes which the UUV can follow that are robust. There are many interpretations of robustness but in this thesis we focus on two main definitions. First, that with slight changes in problem parameters such as ocean current or area bathymetry a UUV's route will not become infeasible, meaning these slight changes are not likely to cause the UUV to break its time and energy constraints. Second, to conserve time and energy a UUV cannot re-solve for new optimal routes often, so we want a route which remains near optimal for slight changes in problem parameters. In this thesis we will formulate exact methods and heuristic for the UUV to find a route which is robust.

1.2. Network Optimization Background

A graph $G = (N, A)$ is a directed network defined by a node set N and an arc set A. We can assign flow variables x_{ij} which represent the flow (of the UUV, energy, time, etc) across an arc $(i, j) \in A$. We associate with each node $i \in N$ a number $d(i)$ representing its supply/demand. To constrain the flow variables we will establish constraints relating the flow in and out of a node i to the number $d(i)$. Standard notation uses flow out minus flow in for the constraint which is written as,

$$\sum_{(i,j)\in A} x_{ij} - \sum_{(j,i)\in A} x_{ji} \leq, =, \geq d(i) \;\forall\; i \in N.$$

In matrix notation we write these $|N|$ constraints with,

$$\mathbf{Bx} \leq, =, \geq \mathbf{d}.$$

For these flow constraints $\mathbf{B}$ is called the node-arc incident matrix where each column $\mathbf{B}^{(i,j)}$ in the matrix corresponds to the arc $(i, j) \in A$. The column $\mathbf{B}^{(i,j)}$ has a $+1$ in the ith row, a -1 in the jth row, and the rest of its entries zero. We will refer to a path problem as a situation where one unit of flow must go from a node s to a node t. In a path problem $d(s) = 1$, $d(t) = -1$ and the rest of the entries in $\mathbf{d}$ are zero. For a general flow problem we will call $\mathbf{d}$ the demand vector, where node i having a supply of α means $d(i) = \alpha$, and node i having a demand of α means $d(i) = -\alpha$.

A time expanded version of a graph G is made up of copies of G at each time step, to account properly for the evolution of the underlying system over time. An arc $(i, j) \in A$ moves down in the time expanded graph to represent moving forward in time. The amount of time necessary to transit on an arc (i, j) is represented by how far down the arc goes in the time expanded graph. In time expanded graphs there are no arcs which go up in the graph because this represents going backward in time.

For a more in depth look at graphs and time dependent graphs see [**1**].

1.3. Outline

In Chapter 2, we present the basic problem the UUV faces and present a simple formulation as the basis for the thesis. Once this formulation is established, we point out the flaws that the solutions are not robust, and we cannot nearly encompass the circumstances of a real world problem. We use the simple formulation as a stepping stone to build formulations for more realistically complex problems.

Chapter 3 uses a deterministic time expanded graph of the simple formulation to handle time dependent no go zones which affect transit times for the UUV. We then point out how once again, this formulation does not create a robust route for the UUV.

Chapter 4 takes the time expanded graph for the deterministic model and alters it to take the stochasticity of the transit times into consideration. This formulation is the main contribution of this thesis. The stochastic time expanded formulation is designed to give the UUV a confidence level for whether or not the time constraint will be broken. Since the formulation takes into account the stochasticity of problem parameters it creates a route which is robust against small changes in these parameters. We then present another more complex formulation which gives the UUV even more control over the route it chooses called the route alteration formulation. Then we look at different objective functions for these formulations and readdress the energy constraint for the UUV.

Chapter 5 contains analysis and heuristics of the proposed formulations. Analysis is only done on the stochastic time expanded formulation and the route alteration formulation because these are considered the main contributions of this thesis. We start this analysis by looking at the run time of the exact formulations. Then we propose a nearest neighbor heuristic and analyze whether this type of heuristic would perform similarly to optimal solutions or not. We also propose a rounding heuristic and analyze its performance as compared to the exact formulations.

Chapter 6 Discusses the usefulness of this thesis, and whether the proposed formulations and heuristics should be considered for application on board UUVs for real world problems. Then we proposes different areas of further work available for this problem. First, we suggest further analysis which should be done for the stochastic time expanded formulation and the route alteration formulation. Then we discuss how more problem characteristics could be added to create a formulation which encompasses more real world issues.

The Appendix, explains a number of issues which we did not take the time to explain in the body of the thesis. First we discuss the issue of overtaking in the deterministic time expanded graph and the stochastic time expanded graph and how it can be assumed away. Second, we discuss a case when there is a simple transformation between the deterministic time expanded graph and the stochastic time expanded graph. Finally we included a lemma which gives us a way to combine two different optimization problems as one equal optimization problem.

1.4. Common Terminology and Notation

- UUV - Unmanned Underwater Vehicle
- Route - A path or plan for the UUV to follow in connection with the decision variable $\mathbf{y}$ in all formulations.
- $\mathbf{v}$ - Bold lower case letters used for vectors.
- $\mathbf{A}$ - Bold upper case letters used for matrices.
- $\mathbf{B}$ - Used as a node arc incident matrix.
- $\mathbf{I}$ - Identity matrix.
- $\mathbf{B}'$ - The transpose of $\mathbf{B}$.
- S - Used to denote the start node.
- D - Used to denote the destination node.
- T - Total amount of time available for a mission in Chapter 2 and the total amount of time steps available for a mission in Chapters 3 and 4.
- E - Total amount of energy available for UUV mission.
- G - The Decision Graph defined in 2.1 in connection with Formulation 2.2.1.
- Simple Formulation - Formulation 2.2.1
- $(i.k, j.l)$ - An arc in the decision graph G.
- $y_{i.k}^{j.l}$ - The flow variable for arc $(i.k, j.l)$ in the decision graph G.
- $\mathbf{y}$ - Flow variable vector on decision graph G.
- G_d - Deterministic Time Expanded Graph defined in 3.2 in connection with Formulation 3.3.1.
- Deterministic Time Expanded Formulation - Formulation 3.3.1.
- G_s - Stochastic Time Expanded Graph defined in 4.1 in connection with Formulation 4.3.7.
- Stochastic time Expanded Formulation - Formulation 4.3.7.
- G_a - Route Alteration Graph defined in 4.14 in connection with Formulation 4.4.1.
- Route Alteration Formulation - Formulation 4.4.1.
- $(i.k.t, j.l.s)$ - An arc in the graphs G_d, G_s, and G_a.
- $x_{i.k.t}^{j.l.s}$ - The flow variable for arc $(i.k.t, j.l.s)$ in the graphs G_d, G_s, G_a.
- $\mathbf{x}$ - Flow variable on deterministic and stochastic time expanded graphs.
- $\mathbf{e}$ - Vector of energy consumption parameters.
- $\mathbf{z}$ - Decision variable for route alteration in Formulation 4.4.1.
- $z_{i.k.t}$ - Route alteration decision variable for node $i.k.t$ in Formulation 4.4.1.
- $V_{i.k.t}^{j.l.s}$ - Represents the event the UUV transits on the arc $(i.k.t, j.l.s)$.
- $U_{i.k.t}$ - Represents the event that the UUV visits node $i.k.t$.
- $W_{i.k}^{j.l}$ - Represents the event the UUV transits on the arc $(i.k, j.l)$ in the decision graph G.
- $Q_{i.k}^{j.l}$ - Represents the event the UUV transits on the arc $(i.k, j.l)$ in the decision graph G and reaches the destination node D by time step T.
- $R_{i.k}$ - Represents the event the UUV transits reaches node $i.k$ in the decision graph G and reaches the destination node D by time step T.
- r_i - Reward collected for performing task i.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 2

The Simple Version of the UUV Problem

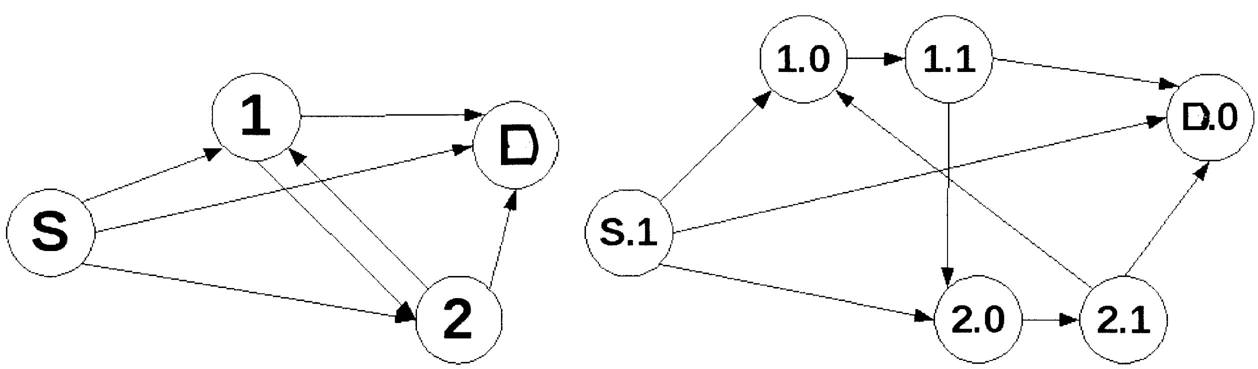
2.1. An Example Simple Problem

Let us first consider a specific situation to gain intuition on how we might formulate the simple version of our problem. We will assume the UUV is launched at a start location, S, and there are two different tasks, 1 and 2, the UUV can choose to perform and then must go to the destination location, D. Consider the network shown in Figure 1(a) which depicts this situation. We will assume that the UUV will only travel to a node if it performs the task at that location, and the UUV cannot perform a task more than once. This means in the graph in Figure 1(a) the arcs going into nodes 1 and 2 represent not only physical transit but also performing a task. Now let us create a graph where an arc either represents physical transit, or it represents performing a task, but not both. We will split up nodes 1 and 2 into two different states, one being before the task is performed and one being after the task is performed. The resulting graph is shown in Figure 1(b). We will call the graph in Figure 1(b) the decision graph because our goal is to decide exactly how the UUV should move in this graph. Specifically the decision graph as shown in Figure 1(b) has the following meaning,

- Node $S.1$ is the start node for the UUV.
- Node 1.0 (2.0) represents the UUV being at the location for task 1 (2) but not yet performing task 1 (2).
- Node 1.1 (2.1) represents the UUV being at the location for task 1 (2) after performing task 1 (2).
- Node $D.0$ is the destination node for the UUV.
- Arcs of type $(i.1, j.0)$ represent the physical transit between node i and node j.
- Arcs of type $(i.0, i.1)$ represent performing the task at node i.

By splitting up nodes 1 and 2 we can clearly see that the decision graph is bipartite. If the UUV is at a node of type $i.0$ it must go to node $i.1$ next, and if the UUV is at node $i.1$ it most go to a node of type $j.0$ next.

The UUV uses a certain amount of time and energy to transit on an arc. Consider the problem parameters for time and energy in Example 1.



(a) UUV Initial Example Network (b) UUV Decision Graph

FIGURE 2.1. Example Problem Graphs

Example Problem 1

Arc	$(S.1, 1.0)$	$(S.1, 2.0)$	$(S.1, D.0)$	$(1.0, 1.1)$	$(1.1, 2.0)$	$(1.1, D.0)$	$(2.0, 2.1)$	$(2.1, 1.0)$	$(2.1, D.0)$
time	1	2	2	1	1	1	1	1	1
energy	1	2	1	1	2	1	2	2	1

- The mission is 5 time units long, i.e. the UUV must be at node $D.0$ by time 5.
- The UUV has 6 units of energy available for consumption.
- The reward for completing task 1 is one.
- The reward for completing task 2 is two.

We will formulate this route planning problem as a flow problem on the decision graph shown in Figure 1(b). Let,

$$y^{j.l}_{i.k} = \text{ the number of times the UUV transits on arc } (i.k, j.l)$$

The UUV is only allowed to complete a task once, which means it cannot travel on an arc of type $(i.0, i.1)$ more than once. This means that the UUV cannot reach node $i.1$ more than once. If the UUV were at a node $i.1$ then it must choose a unique node $j.0$ to transit to next. Therefore, the UUV cannot transit on arcs of type $(i.1, j.0)$ more than once either. To meet these constraints we will make $\mathbf{y}$ binary, meaning $y^{j.l}_{i.k}$ is either 0 or 1 for all $(i.k, j.l) \in A$.

The decision variable $\mathbf{y}$ represents a route for the UUV to transit on in the decision graph G. We will now establish different constraints the UUV must satisfy to have a feasible route.

The UUV must leave node $S.1$, so the flow out of node $S.1$ must be equal to one, which is written as,

$$y^{1.0}_{S.1} + y^{2.0}_{S.1} + y^{D.0}_{S.1} = 1.$$

The UUV must end at node $D.0$, so flow into node $D.0$ must be equal to one, which is written as,

$$-y^{D.0}_{S.1} - y^{D.0}_{1.1} - y^{D.0}_{2.2} = -1.$$

If the UUV goes to node $i.k$ for $i \in \{1, 2\}$, $k \in \{0, 1\}$, then the UUV must also leave that node, so the flow out must equal the flow in, which for each node is written as,

$$\begin{aligned}
y^{1.1}_{1.0} - y^{1.0}_{S.1} - y^{1.0}_{2.1} &= 0 \\
y^{2.0}_{1.1} + y^{D.0}_{1.1} - y^{1.1}_{1.0} &= 0 \\
y^{2.1}_{2.0} - y^{2.0}_{S.1} - 2^{2.0}_{1.1} &= 0 \\
y^{1.0}_{2.1} + y^{D.0}_{2.1} - y^{2.1}_{2.0} &= 0
\end{aligned}$$

The constraints thus far are the flow constraints for the UUV for a path problem with source $S.1$ and sink $D.0$.

The UUV has only a finite amount of time available so we must place a time constraint on the decision vector $\mathbf{y}$. For each arc $(i.k, j.l)$ if $y^{j.l}_{i.k} = 1$ then the UUV is going to transit on the arc and we know how much time will be used. So overall we know $\mathbf{y}$ must satisfy,

$$y^{1.0}_{S.1} + 2y^{2.0}_{S.1} + 2y^{D.0}_{S.1} + y^{1.1}_{1.0} + y^{2.0}_{1.1} + y^{D.0}_{1.1} + y^{2.1}_{2.0} + y^{1.0}_{2.1} + y^{D.0}_{2.1} \leq 5.$$

Similarly the UUV has a finite amount of energy available so we have the following energy constraint,

$$y^{1.0}_{S.1} + 2y^{2.0}_{S.1} + y^{D.0}_{S.1} + y^{1.1}_{1.0} + 2y^{2.0}_{1.1} + y^{D.0}_{1.1} + 2y^{2.1}_{2.0} + 2y^{1.0}_{2.1} + y^{D.0}_{2.1} \leq 6.$$

Another circumstance which we must protect ourselves from is the formation of independent cycles in the decision variable $\mathbf{y}$. In this example if the total time and energy available were high enough the following solution may be feasible,

$$y^{D.0}_{S.1} = 1 \qquad y^{1.1}_{1.0} = 1 \qquad y^{2.0}_{1.1} = 1 \qquad y^{2.1}_{2.0} = 1 \qquad y^{1.0}_{2.1} = 1 \qquad y^{j.l}_{i.k} = 0 \text{ for all other arcs}$$

This solution has the UUV going from the start node to the destination node but also has it transiting on the cycle $1.0 \to 1.1 \to 2.0 \to 2.1 \to 1.0$. Obviously this situation is actually infeasible, so we must create constraints to eliminate such cycles. We can do this using constraints used in formulations of the traveling salesman problem which are written as,

$$\forall\ U \subseteq \{S.1, 1.0, 1.1, 2.0, 2.1, D.0\} \qquad \sum_{i.k, j.l \in U} y_{i.k}^{j.l} \leq |U| - 1$$

If the UUV goes to node 1.0(2.0) it must go to node 1.1(2.1) next. This means we can ignore the arcs which represent performing a task and eliminate cycles in arcs which represent physical transit. Furthermore, we know there are no arcs going into node $S.1$ and no arcs going out of node $D.0$ so there cannot be a cycle which contains either of these nodes. Hence, cycles can alternatively be eliminated with the constraints,

$$\forall\ U \subseteq \{1, 2\} \qquad \sum_{i,j \in U} y_{i.1}^{j.0} \leq |U| - 1$$

which for this small example is given by the single constraint $y_{1.1}^{2.0} + y_{2.1}^{1.0} \leq 1$.

This constraint says that for any subset of nodes, the number of arcs in between these nodes the UUV travels on must be less than or equal to the magnitude of the subset minus one. Thus, these constraints eliminate the possibility of any cycle in the decision vector $\mathbf{y}$.

We know we want to maximize the reward gained by completing tasks, so in this example the objective function is written as,

$$\max y_{1.0}^{1.1} + 2y_{2.0}^{2.1}.$$

Combining these constraints and the objective function, we have the following formulation,

$$\begin{array}{lr}
\max & y_{1.0}^{1.1} + 2y_{2.0}^{2.1} \\
\text{s.t.} & y_{S.1}^{1.0} + y_{S.1}^{2.0} + y_{S.1}^{D.0} = 1 \\
& -y_{S.1}^{D.0} - y_{1.1}^{D.0} - y_{2.2}^{D.0} = -1 \\
& y_{1.0}^{1.1} - y_{S.1}^{1.0} - y_{2.1}^{1.0} = 0 \\
& y_{1.1}^{2.0} + y_{1.1}^{D.0} - y_{1.0}^{1.1} = 0 \\
& y_{2.0}^{2.1} - y_{S.1}^{2.0} - 2_{1.1}^{2.0} = 0 \\
& y_{2.1}^{1.0} + y_{2.1}^{D.0} - y_{2.0}^{2.1} = 0 \\
& y_{S.1}^{1.0} + 2y_{S.1}^{2.0} + 2y_{S.1}^{D.0} + y_{1.0}^{1.1} + y_{1.1}^{2.0} + y_{1.1}^{D.0} + y_{2.0}^{2.1} + y_{2.1}^{1.0} + y_{2.1}^{D.0} \leq 5 \\
& y_{S.1}^{1.0} + 2y_{S.1}^{2.0} + y_{S.1}^{D.0} + y_{1.0}^{1.1} + 2y_{1.1}^{2.0} + y_{1.1}^{D.0} + 2y_{2.0}^{2.1} + 2y_{2.1}^{1.0} + y_{2.1}^{D.0} \leq 6 \\
& y_{i.1}^{2.0} + y_{2.1}^{1.0} \leq 1 \\
& \mathbf{y} \text{ binary}
\end{array}$$

In this simple example it is easy to see the optimal solution. The UUV cannot do task 1 and then task 2 because it would use too much energy, and the UUV cannot do task 2 and then task 1 because it would use too much time. So the UUV chooses to do only task 2 because it has a higher reward value than task 1. Solving this problem using the integer programming formulation we have constructed gives us the same result.

2.2. The General Simple Problem

Assume we are given a start location, S, a destination location D and the location of tasks $\{1, \ldots, n\}$ which the UUV can choose to complete. With this information we can build the initial physical graph that corresponds to Figure 1(a) in Example 1. We assume that the UUV only transits to a node if it performs the task. This means that arcs represent not only transit in between nodes, but the completion of tasks. As in the example, we create a new graph which makes it so that an arc either represents transiting between nodes, or performing a task, but not both.

Definition 2.1 (Decision Graph). *We will refer to the* ***decision graph*** *as a graph with the following structure.*

- $N = \{S.1\} \bigcup \{i.0 | i \text{ is a task}\} \bigcup \{i.1 | i \text{ is a task}\} \bigcup \{D.0\}$ *is the node set for the decision graph.*
- $A = \{(S.1, i.0) | i.0 \in N\} \bigcup \{(i.0, i.1) | i \text{ is a task}\} \bigcup \{(i.1, j.0) | i \text{ and } j \text{ are tasks}\} \bigcup \{(i.1, D.0) | i.1 \in N \setminus \{S.1\}\}$ *is the arc set for the decision graph.*
- $G = (N, A)$ *is the* ***decision graph*** *with node set N and arc set A.*
- $\boldsymbol{B}$ *will represent the node-arc incident matrix for the decision graph G.*
- $\boldsymbol{d}$ *will represent the demand vector for a path problem with source $S.1$ and sink $D.0$ on graph G.*

For Example 1 the decision graph was shown in Figure 1(b). The decision graph represents the following in our problem,

- Node $S.1$ is the start node for the UUV.
- Node $i.0$ represents the UUV being at the location of task i before performing task i.
- Node $i.1$ represents the UUV being at the location of task i after performing task i.
- Node $D.0$ is the destination node which the UUV must reach before the end time.
- Arcs of type $(i.1, j.0) \in A$ represent transiting from task i to task j.
- Arcs of type $(i.0, i.1) \in A$ represent performing the task at node i.

We call this the decision graph because our goal is to figure out the best way for the UUV to transit in this graph to receive the most reward with the time and energy the UUV has available. As in the example problem, the decision graph, G, is bipartite. Henceforth, we will refer to transit on arcs in the decision graph knowing that some arcs represent performing tasks while others represent physical transit.

Now we will create notation for our problem and describe the parameters of our problem.

- E is the total amount of energy we have available to consume.
- T is the amount of time we have until we must reach the destination node.
- r_i is the reward gained for completing task i.
- $t_{i.k}^{j.l}$ is the time it takes to go on arc $(i.k, j.l) \in A$, and $\mathbf{t}$ is the vector of transit times.
- $e_{i.k}^{j.l}$ is the energy used to go on arc $(i.k, j.l) \in A$, and $\mathbf{e}$ is the vector of energy consumption.

Now we create decision variables on each arc in the decision graph,

$$y_{i.k}^{j.l} = \text{ the number of times the UUV transits on arc } (i.k, j.l)$$

We will also refer to $\mathbf{y}$ as the vector of the decision variables $y_{i.k}^{j.l}$. As in the example we will make $\mathbf{y}$ binary to limit the number of times the UUV can complete a task to one.

Now we will construct constraints that the decision variables must satisfy. As in the example problem we know that we must have a path from $S.1$ to $D.0$ so the first set of constraints are the flow constraints given by,

$$\mathbf{By} = \mathbf{d}$$

We know that we have a limited amount of time and energy available so we must satisfy the constraints,

$$\mathbf{t}'\mathbf{y} \leq T$$
$$\mathbf{e}'\mathbf{y} \leq E$$

The last set of constraints needed are those which eliminate cycles. If we allowed cycles such as $(i.0, i.1, j.0, j.1, i.0)$ to be feasible we might get a reward for performing tasks i and j when in reality they are not on the path from $S.1$ to $D.0$. We eliminate cycles in the same way it is done in traveling salesman problem formulations with the constraints,

$$\sum_{(i.k, j.l) \in A | i.k, j.l \in U} y_{i.k}^{j.l} \leq |U| - 1 \qquad \text{where } U \subseteq N$$

By construction, if the UUV goes to node $i.0$ it must go to node $i.1$ next. Furthermore, by construction there are no arcs entering node $S.1$ and no arcs leaving node $D.0$. Similar to the example we can use these facts to consolidate the cycle eliminating constraints to,

$$\sum_{(i.1,j.0)\in A|i,j\in U} y_{i.1}^{j.0} \leq |U| - 1 \qquad \text{where } U \subseteq \{1,\ldots,n\}$$

We know we want to maximize the reward received for performing tasks so we get Formulation 2.2.1:

$$\begin{array}{llr}
\text{(2.2.1a)} & \max & \displaystyle\sum_{i=1}^{n}(r_i * y_{i.0}^{i.1}) \\
\text{(2.2.1b)} & \text{s.t.} & \mathbf{By} = \mathbf{d} \\
\text{(2.2.1c)} & & \mathbf{t}'\mathbf{y} \leq T \\
\text{(2.2.1d)} & & \mathbf{e}'\mathbf{y} \leq E \\
\text{(2.2.1e)} & & \displaystyle\sum_{(i.1,j.0)\in A|i,j\in U} y_{i.1}^{j.0} \leq |U| - 1 \ \forall\ U \subseteq \{1,\ldots,n\} \\
\text{(2.2.1f)} & & \mathbf{y} \text{ binary}
\end{array}$$

This formulation is simple and effective but does have considerable flaws in practice. If we were to encounter this situation in reality the time and energy consumption values would be estimations, which means the solution provided may not be robust. If we were to alter our example problem so that we now had 6 time units available for use, then the optimal solution would be to perform task 2 and then task 1. In this case our time consumption constraint would be tight, so if any of the time consumption estimations were too low the UUV may reach the destination node later than time 6. This situation could have catastrophic results so we need to develop solutions which are robust.

Another deficiency with this formulation is its inability to deal with real world situations. In real world situations, changing currents or time dependent no go zones may alter time and energy consumption values throughout a mission, so we will develop a formulation which has the capability of modeling the time and energy consumption values as functions of time.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 3

Handling Time Windows

In this chapter, we will alter the basic formulation presented in Chapter 2 to address time windows for transit in between nodes and for completing tasks. The inclusion of time windows is necessary for realistic problems. There are areas which may only be entered at certain times of the day. For example, there could be a shipping lane the UUV cannot enter at certain times throughout a day due to heavy commercial traffic. In some cases the UUV can take a longer route around restricted areas, and in other cases there may be no feasible route around the restricted areas.

3.1. Introduction

The way to efficiently model time windows is to create a time expanded graph which models the transit times explicitly. Time expanded graphs use discrete time units, so we must assume we have appropriately discretized time for the situation. Once we have discretized time, we will create the deterministic time expanded graph by creating a copy of the nodes in the decision graph for each time step and inserting arcs which explicitly model the transit times.

In the graph shown in Figure 3.1 we see how we will make copies of each node (except the start node) for each time step. In Figure 3.1 we also see one example arc pointing from node $i.k.t$ o node $j.l.s$. This means that if the UUV leaves node $i.k$ at time t headed for node $j.l$ it will arrive there at time s. All of the arcs we put in the deterministic time expanded graph will accurately represent the transit time needed to travel in between nodes for specific departure times. This allows us to model changes in transit times that are dependent on the departure time.

The overall goal is still to come up with a route, $\mathbf{y}$, in the decision graph, G, as we did in the simple model in Chapter 2. To do this we will establish a connection between the flow variables on the decision graph and the flow variables on the deterministic time expanded graph, so that the UUV's route must take the changing transit times into account.

3.2. Building the Graph

Now we will formally define how to build the time expanded graph. We first assume that we have discretized time and we have T time steps available for the mission. In Chapter 2 T is the total amount of time available but now (and in future chapters) we will use T as the number of time steps available with δt as the step size.

When defining the arc set for the deterministic time expanded graph we only want to include "valid" arcs. What we mean by this is that they accurately represent the transit time needed given the time dependent no go zones.

Definition 3.1 (valid). *An arc $(i.k.t, j.l.s)$ is* ***valid*** *if the following is true,*

- *It takes $s - t$ time units to travel to $j.l$ if the UUV leaves $i.k$ at time t.*
- $(i.k, j.l) \in A$.

Definition 3.2 (Deterministic Time Expanded Graph). *We will refer to the* ***deterministic time expanded graph*** *G_d as a graph with the following structure.*

- $N_d = \{i.k.t | i.k \in N \setminus \{S.1\}$ *and* $0 \leq t \leq T\} \bigcup \{S.1.0, D.1.T\}$ *is the node set for the deterministic time expanded graph.*

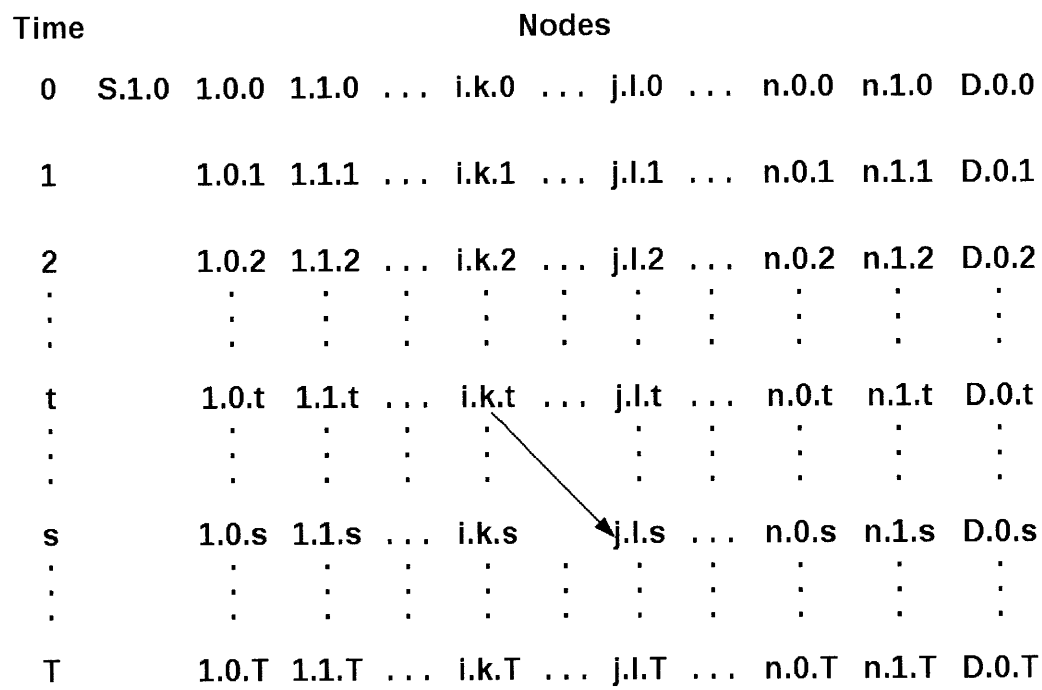


FIGURE 3.1. Introductory Deterministic Time Expanding Graph

- $A_d = \{(i.k.t, j.l.s) | i.k.t, j.l.s \in N_d$ *and the arc is valid*$\} \bigcup \{(D.0.t, D.1.T) | 0 \leq t \leq T\}$ *is the arc set for the time expanded graph.*
- $G_d = (N_d, A_d)$ *is the* ***deterministic time expanded graph*** *with node set* N_d *and arc set* A_d.
- $\boldsymbol{B}_d$ *will represent the node-arc incident matrix for the deterministic time expanded graph* G_d.
- $\boldsymbol{d}_d$ *will represent the demand vector for a path problem with source* $S.1.0$ *and sink* $D.1.T$ *on graph* G_d.

The deterministic time expanded graph represents the following in our problem.

- Node $i.k.t$ represents being at node $i.k \in N$ at time step t.
- Node $D.1.T$ we use as the super sink node which represents the end of the mission.
- An arc $(i.k.t, j.l.s) \in A_d$ means the UUV can leave node $i.k$ at time t and arrive at node $j.l$ at time s.

In words we create the deterministic time expanded graph in the following way,

- We know we start the mission at time 0, so we only include a copy of the start node $S.1$ for time 0.
- We include copies of all nodes in N other than $S.1$ for every time step 0 to T.
- We include destination node $D.1.T$ to use as a super sink node for the UUV's route.
- We include arc $(i.k.t, j.l.s)$ if it takes $s - t$ time units to travel on arc $(i.k, j.l) \in A$ if the UUV leaves at time t.
- We include arcs of type $(D.0.t, D.1.T)$ to use $D.1.T$ as a super sink node.

Now let us consider an example problem to help visualize the deterministic time expanded graph. Consider example problem 2.

Example 2 is a two task problem so it has the same decision graph as shown in Figure 1(b). Now we create the deterministic time expanded graph for the problem which is shown in Figure 3.2. We are claiming that the deterministic time expanded graph satisfies the time constraints of the problem explicitly. Let us look at the transit times from 2 to 1. Looking at the subgraph in Figure 3.3 we see the following properties,

Example Problem 2

- S is the start node.
- D is the destination node.
- The UUV must choose whether or not to do tasks 1 and 2.
- We will again use the decision graph $G = (N, A)$ as shown in Figure 1(b).
- The reward for completing task 1 is one.
- The reward for completing task 2 is two.
- The total amount of time available is 6 time units.
- The total amount of energy available is 7 energy units.

We will assume some of the transit times are constant. The constant transit times are shown in the table below with the row representing the departure node and the column representing the destination node.

Node	1	2	D
S	1	2	2
1	N/A	See Below	1
2	See Below	N/A	1

The transit time necessary to travel between node 1 and node 2 depend on the departure time, and has the following values,

Departure time	0	1	2	3	4	5	6
From Node 1 to Node 2	1	1	3	3	3	3	3
From Node 2 to Node 1	1	3	2	1	1	1	1

- It takes one time unit to complete the task at node 1.
- It takes one time unit to complete the task at node 2.
- It takes as many energy units to transit on an edge as it does time units for all arcs except $(1.1.5, D.0.6)$ for which it takes two energy units.

- $(2.1.0, 1.0.1) \in A_d$ because it takes one time unit to transit from 2 to 1 if the UUV leaves at time 0.
- $(2.1.1, 1.0.4) \in A_d$ because it takes three time units to transit from 2 to 1 if the UUV leaves at time 1.
- $(2.1.2, 1.0.4) \in A_d$ because it takes two time units to transit from 2 to 1 if the UUV leaves at time 2.
- $(2.1.3, 1.0.4) \in A_d$ because it takes one time unit to transit from 2 to 1 if the UUV leaves at time 3.
- $(2.1.4, 1.0.5) \in A_d$ because it takes one time unit to transit from 2 to 1 if the UUV leaves at time 4.
- $(2.1.5, 1.0.6) \in A_d$ because it takes one time unit to transit from 2 to 1 if the UUV leaves at time 5.
- We do not include arcs such as $(2.1.6, 1.0.7)$ because the end time is 6.

All of the transit times for our example problem are explicitly part of the deterministic time expanded graph by limiting the arc set of the graph to coincide with the transit time parameters.

Overtaking in a time expanded graph is when there are arcs $(i.k.t, j.l.s)$, and $(i.k.t^+, j.l.s^-) \in A_d$ where $t^+ > t$ and $s^- < s$ which is explained further in A.1. This means that if the UUV leaves node $i.k$ at time t it will arrive at node $j.l$ later than it would if it left at time t^+, which is after time t. In A.1 we show there is a simple transformation which allows us to assume there is no overtaking arcs in the graph G_d.

In the time expanded graph G_d, moving forward in time is represented by moving down in the graph. Since the UUV cannot travel back in time, there are no arcs in G_d which go up in the graph. This means a cycle can only occur in G_d if there are arcs which move strictly horizontally, which corresponds to staying at a constant time t. If there were a cycle in graph G_d it would have to have the form $(i_1.k_1.t \to i_2.k_2.t \to \ldots \to i_M.k_M.t \to i_1.k_1.t)$, for a fixed time t which means that all of the transitions $i_m.k_m \to i_{m+1}.k_{m+1}$ take zero time. This would mean that it takes zero time to complete all of the tasks $i_1, \ldots, i_M$ and it takes

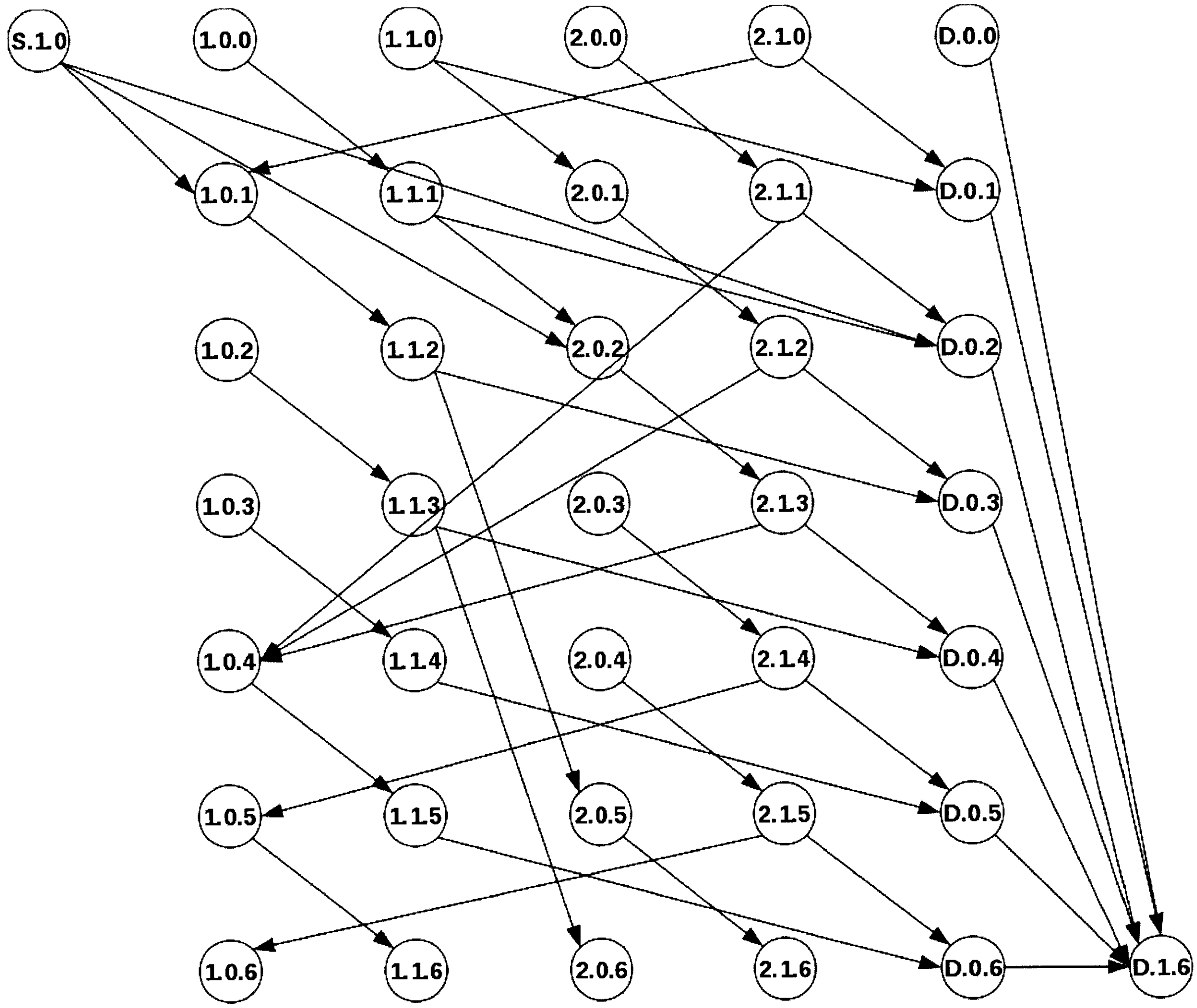


FIGURE 3.2. Deterministic Time Expanding Graph

zero time to transit in between those tasks. Cycles like this do not appear in realistic situations, so we will assume there are no cycles in the graph G_d. More formally, we will assume it takes at least one time unit to complete a task to remove all possibility of cycles in graph G_d.

Using a time expanded graph allows us to not only make time windows inherit in the graph but it also makes it much easier to incorporate small changes in transit times due to things like current and weather. By using time expanded graphs to model transit times, the complexity of the transit times does not effect the constraints or variables needed to solve the problem, just the construction of the time expanded graph.

3.3. The Formulation

We will use the deterministic time expanded graph to formulate our problem and maximize the reward the UUV collects.

For the energy constraint we will assume we have the following problem parameters,

- $e_{i.k.t}^{j.l.s}$ is the amount of energy consumed if the UUV transits on arc $(i.k.t, j.l.s) \in A_d$ and we will refer to $\mathbf{e}$ as the vector of these energy consumption values..
- E is the total amount of energy available for consumption.

The decision variables for this formulation will be flow variables on the graph G_d. Let,

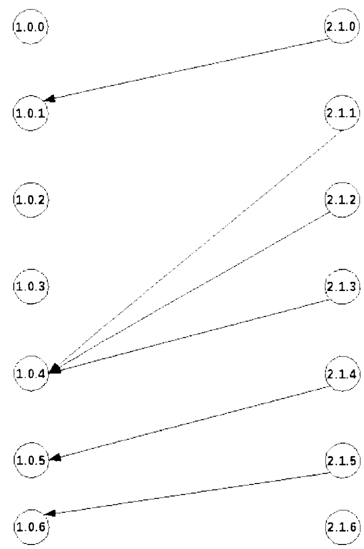


FIGURE 3.3. Deterministic Time Expanding Subgraph

$$x_{i.k.t}^{j.l.s} = \text{ the number of times the UUV transits on arc } (i.k.t, j.l.s)$$

We will also use variables on the decision graph G as we did in Chapter 2. Let,

$$y_{i.k}^{j.l} = \sum_{t,s|(i.k.t,j.l.s)\in A_d} x_{i.k.t}^{j.l.s}.$$

We will refer to $\mathbf{x}$ and $\mathbf{y}$ as the vector of these variables.

As in Chapter 2 the variable $y_{i.k}^{j.l}$ represents the number of times the UUV transits from $i.k$ to $j.l$. As in Formulation 2.2.1 we will force $\mathbf{y}$ to be binary to only allow the UUV to perform a task once. Forcing $\mathbf{y}$ to be binary means that $\mathbf{x}$ must also be binary by the construction of $\mathbf{y}$.

We know the UUV must start at node S and end at node D so we again have the flow constraints as in Chapter 2,

$$\mathbf{B}\mathbf{y} = \mathbf{d}.$$

Similarly the UUV must go from node $S.1.0$ to node $D.1.T$ in the deterministic time graph, which means the UUV must satisfy,

$$\mathbf{B}_d\mathbf{x} = \mathbf{d}_d.$$

We know we only have a limited amount of energy and we know how much energy is consumed if the UUV transits on an arc in A_d, so the UUV's route must satisfy the constraint,

$$\mathbf{e}'\mathbf{x} \leq E.$$

Now consider Formulation 3.3.1:

$$\text{(3.3.1a)} \qquad \max \quad \sum_{i=1}^{n} (r_i * y_{i.0}^{i.1})$$

$$\text{(3.3.1b)} \qquad \text{s.t.} \quad \mathbf{By} = \mathbf{d}$$

$$\text{(3.3.1c)} \qquad \mathbf{B}_d \mathbf{x} = \mathbf{d}_d$$

$$\text{(3.3.1d)} \qquad \sum_{s,t|(i.k.t,j.l.s)\in A_d} x_{i.k.t}^{j.l.s} = y_{i.k}^{j.l} \quad \forall (i.k, j.l) \in A$$

$$\text{(3.3.1e)} \qquad \mathbf{e}'\mathbf{x} \le E$$

$$\text{(3.3.1f)} \qquad \mathbf{y} \text{ binary, } \mathbf{x} \text{ binary}$$

Constraint 3.3.1c forces the UUV to go from the start node to the destination node. Constraints 3.3.1b and 3.3.1d limit the number of times a task is performed to one. Constraint 3.3.1e forces the UUV to satisfy the energy constraint. The objective function, 3.3.1a, maximizes the reward the UUV collects for performing tasks. The only constraint Formulation 3.3.1 is missing as compared to Formulation 2.2.1 are the constraints to disallow cycles. We can, however, prove that no cycles can exist in a feasible solution since we have assumed the graph G_d has no cycles. We prove this fact in three steps with Lemma 3.3, Corollary 3.4, and Corollary 3.5. Lemma 3.3 uses the constraints of Formulation 3.3.1 to show a useful property of a feasible $\mathbf{x}$. Then Corollary 3.4 shows that a cycle in $\mathbf{y}$ means there is a cycle in $\mathbf{x}$ by using Lemma 3.3. Finally Corollary 3.5 shows that by assuming there are no cycles in the graph G_d there cannot be a cycle in $\mathbf{y}$ or $\mathbf{x}$.

Lemma 3.3. *For feasible solutions $\boldsymbol{y}$ and $\boldsymbol{x}$ for problem 3.3.1, if $1 = x_{i.k.t_1}^{j.l.t_2} = x_{j.l.s_1}^{i'.k'.s_2}$ then $t_2 = s_1$.*

PROOF. Assume $\mathbf{y}$ and $\mathbf{x}$ are feasible solutions to Formulation 3.3.1 and $1 = x_{i.k.t_1}^{j.l.t_2} = x_{j.l.s_1}^{i'.k'.s_2}$. For a contradiction assume $t_2 \ne s_1$. By constraint 3.3.1d we know $1 = y_{i.k}^{j.l} = y_{j.l}^{i'.k'}$. We know that forcing $\mathbf{y}$ to be binary limits the number of times the UUV visits the node $j.l$ to no more than once. Since $y_{i.k}^{j.l} = 1$ this means $y_{j'.l'}^{j.l} = 0$ for all $j'.l' \ne i.k$.

From constraint 3.3.1d this means $x_{j'.l'.t'}^{j.l.s'} = 0$ if $j'.l' \ne i.k$. By assuming $t_2 \ne s_1$ it means the variables $x_{i.k.t_1}^{j.l.t_2}$ and $x_{i.k.t'}^{j.l.s_1}$ are unique for all $0 \le t' \le T$. So we can conclude,

$$\begin{aligned}
1 = y_{i.k}^{j.l} &= \sum_{t',s'|(i.k.t',j.l.s')\in A_d} x_{i.k.t'}^{j.l.s'} && \text{from 3.3.1d} \\
&\ge x_{i.k.t_1}^{j.l.t_2} + \sum_{t'|(i.k.t',j.l.s_1)\in A_d} x_{i.k.t'}^{j.l.s_1} && \text{because } t_2 \ne s_1 \text{ we can separate the variables} \\
&= x_{i.k.t_1}^{j.l.t_2} + \sum_{j'.l'.t'|(j'.l'.t',j.l.s_1)\in A_d} x_{j'.l'.t'}^{j.l.s_1} && \text{because } x_{j'.l'.t'}^{j.l.s_1} = 0 \text{ if } j'.l' \ne i.k \\
&= x_{i.k.t_1}^{j.l.t_2} + \sum_{j'.l'.t'|(j.l.s_1,j'.l'.t')\in A_d} x_{j.l.s_1}^{j'.l'.t'} && \text{from 3.3.1c} \\
&\ge x_{i.k.t_1}^{j.l.t_2} + x_{j.l.s_1}^{i'.k'.s_2} \\
&= 2
\end{aligned}$$

□

Corollary 3.4. *For feasible solutions $\boldsymbol{y}$ and $\boldsymbol{x}$ for problem 3.3.1, if there is a cycle in the flow variable $\boldsymbol{y}$ then there is a cycle in the flow variable $\boldsymbol{x}$.*

PROOF. Assume $\mathbf{y}$ and $\mathbf{x}$ are feasible solutions for problem 3.3.1 and there is a cycle in the flow variable $\mathbf{y}$. Assume the cycle in $\mathbf{y}$ is described by,

$$i_1.k_1 \to i_2.k_2 \to \ldots \to i_{m-1}.k_{m-1} \to i_m.k_m = i_1.k_1$$

Then by constraint 3.3.1d there exists times $t_1, s_1, t_2, s_2, \ldots, t_{m-1}, s_{m-1}$ such that,

$$1 = x_{i_1.k_1.t_1}^{i_2.k_2.s_1} = x_{i_2.k_2.t_2}^{i_3.k_3.s_2} = \ldots = x_{i_{m-1}.k_{m-1}.t_{m-1}}^{i_m.k_m.s_{m-1}}$$

Lemma 3.3 shows that $s_1 = t_2$, $s_2 = t_3$, ..., $s_{m-2} = t_{m-1}$, and $s_{m-1} = t_1$. This implies that we have the following cycle in the x variables,

$$i_1.j_1.t_1 \to i_2.k_2.t_2 \to \ldots \to i_{m-1}.k_{m-1}.t_{m-1} \to i_1.k_1.t_1.$$

□

Corollary 3.5 uses Corollary 3.4 and the assumption we have made that G_d contains no cycles to show that a feasible $\mathbf{y}$ to problem 3.3.1 cannot contain a cycle.

Corollary 3.5. *If there are no cycles in the graph G_d there cannot be a cycle in a feasible solution to problem 3.3.1.*

PROOF. Assume no cycles exist in graph G_d. Therefore there cannot be a cycle in a feasible $\mathbf{x}$. From Corollary 3.4 we know if there is a cycle in $\mathbf{y}$ there is a cycle in $\mathbf{x}$. Taking the equivalent contra-positive we know from Lemma 3.4 no cycle in $\mathbf{x}$ implies no cycle in $\mathbf{y}$. Since we know there is no cycle in $\mathbf{x}$, there is no cycle in $\mathbf{y}$. □

The deterministic problem is a special case of the stochastic problem, which will be formulated in Chapter 4. One of the results found when looking at the stochastic case is that if we relax Formulation 3.3.1 so that $0 \leq \mathbf{x} \leq 1$ instead of forcing $\mathbf{x}$ to be binary we get the same solution. This result is shown in Corollary 4.13. It is unclear as to how to constrain $\mathbf{x}$ to find solutions faster, but it should be noted that we do have the choice of making $\mathbf{x}$ binary, or $0 \leq \mathbf{x} \leq 1$.

In Chapter 2 we constructed Formulation 2.2.1 which had two main inadequacies, its inability to handle time dependent no go zones, and its lack of robustness. With Formulation 3.3.1 we are able to handle time dependent no go zones, but we may still get solutions which are not robust. The time consumption values which we use to create the deterministic time expanding graph, and the energy consumption values will be estimations in real problems. This means that if our estimated values are too low we could end up with solutions which would break time or energy constraints. Next we will develop a formulation which tries to take this into consideration so that we can generate solutions which have a high probability of not breaking the time and energy constraints.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 4

Finding A Robust Solution

Initially we will ignore energy consumption and focus on the time constraints. We will create a formulation designed to handle stochastic transit times and then we will reintroduce the energy constraint in Section 4.6 once we have developed this formulation.

4.1. Introduction

Similar to the deterministic case in Chapter 3 we will use a time expanded graph which models the transit times explicitly. The only change is now the transit times are random variables. In the deterministic time expanded model, if the UUV leaves node $i.k$ at time t it will get to node $j.l$ at some time s. In the stochastic model we will have arcs $(i.k.t, j.l.s_1)$, $(i.k.t, j.l.s_2)$, $(i.k.t, j.l.s_3)$ and so on to represent the UUV reaching node $j.l$ at time s_1, s_2, or s_3.

We make copies of each node in the decision graph G for each time step which is depicted in Figure 4.1. Unlike the deterministic case we will create copies of the start node $S.1$ to represent the UUV starting its mission at times other than time 0. We allow for this because realistically, the UUV could be launched late and thus begin its mission late. In Figure 4.1 we see examples of arcs pointing from node $i.k.t$ to nodes $j.l.s_1$, $j.l.s_2$ and $j.l.s_3$. These arcs represent the fact that if the UUV leaves node $i.k$ at time t headed for node $j.l$ there is a positive probability it will arrive there at time s_1, s_2, or s_3. All of the arcs in the stochastic time expanded graph will accurately represent the distribution of the transit time between nodes for specific departure times. This allows us to model changes in transit times as well as their stochasticity.

The overall goal is still to come up with a route $\mathbf{y}$ in the decision graph G as we have done in Chapters 2 and 3. To do this we will establish a way to calculate the transition probabilities for a given route with a linear programming problem. Then we will tie these calculated probabilities to the decision vector $\mathbf{y}$ and place a constraint on the probability the UUV reaches the destination node by the end time. This formulation will provide solutions which are robust in the sense that we will be guaranteed the UUV will reach the destination node with the time allowed with a specific probability.

4.2. Building the Graph

We want to develop a formulation which handles stochastic parameters. We again assume we have discretized time and T is the total amount of time steps available for the mission. We will assume we have a parameter $0 \leq \beta \leq 1$ where β is the confidence level we must have of reaching the destination node by time step T. So we want $\Pr\{\text{The UUV reaches the destination node by time step } T\} \geq \beta$.

Definition 4.1 (Stochastic Time Expanded Node Set). *The* ***stochastic time expanded node set*** *N_s is a node set with the following properties,*

- $N_s = \{i.k.t | i.k \in N \text{ and } 0 \leq t \leq T\} \bigcup \{S.0.0, D.1.T\}$.

In words we create the stochastic time expanded node set in the following way,

- Include copies of each node in the decision graph G for every time step 0 to T.
- Add a super source start node $S.0.0$ which represents the general start of the mission.
- Add a super sink destination node $D.1.T$ which represents the general end of the mission.

The stochastic time expanded node set is similar to the deterministic time expanded node set with the main difference being the addition of copies of the start node $S.1$ for all time steps. We do this because we are no longer going to assume the mission starts at the exact time planned, but could start at later times as well. This means that the *UUV* being at node $S.1.t$ represents the mission starting at time t.

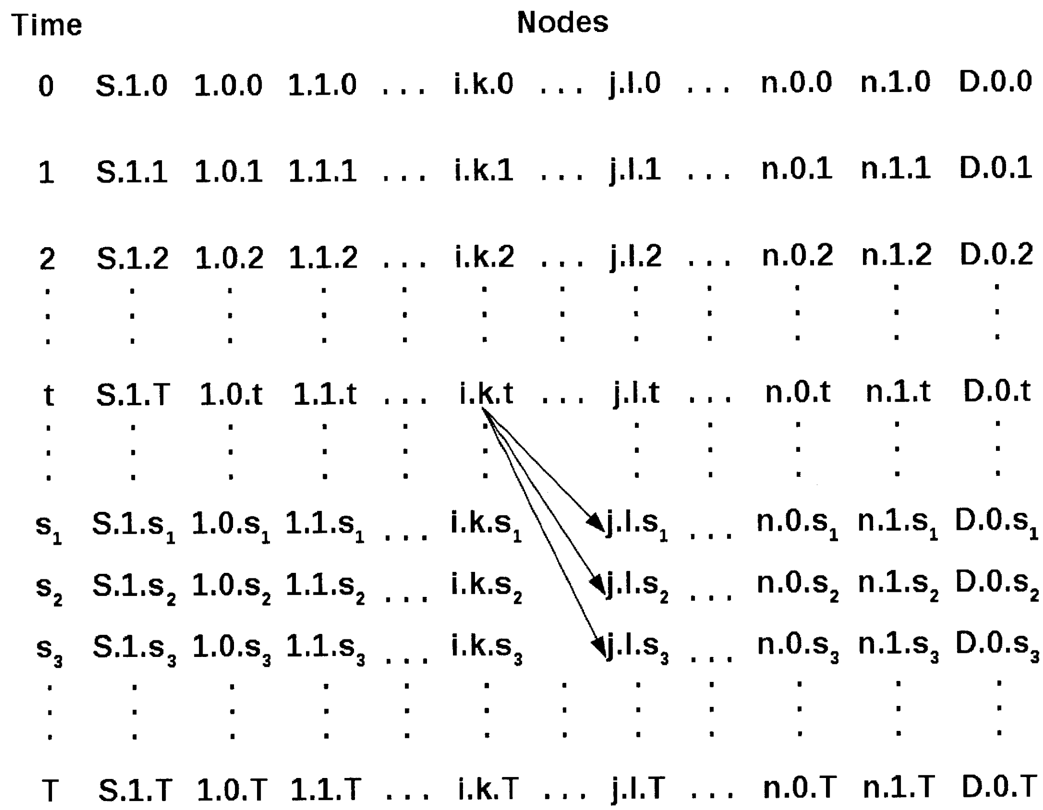


FIGURE 4.1. Introductory Stochastic Time Expanding Graph

Ultimately we want to calculate and bound the probability the UUV reaches the destination node by time step T. To do this we will calculate the probability the UUV transits on an arc in the stochastic time expanded graph. Then the probability the UUV reaches a node is the sum of the probability the UUV transits on an arc entering the node. We will then use the calculated probabilities to establish a confidence level for the probability the UUV reaches the destination node D by time T. To calculate the transition probabilities we will use the following events,

Notation 4.2. *$U_{i.k.t}$ is the event that at time t the UUV is at node $i.k$, for all $i.k \in N$.*

Notation 4.3. *$V_{i.k.t}^{j.l.s}$ is the event the UUV leaves node $i.k$ at time t heading towards $j.l$ and arrives at $j.l$ at time s, for all $(i.k, j.l) \in A$.*

Notation 4.4. *$W_{i.k}^{j.l}$ is the event the UUV leaves $i.k$ to go to node $j.l$ for $(i.k, j.l) \in A$.*

Our goal is to calculate the probabilities $\Pr\{V_{i.k.t}^{j.l.s}\}$.
By definition we have, $V_{i.k.t}^{j.l.s} \to U_{i.k.t}$ and $V_{i.k.t}^{j.l.s} \to W_{i.k}^{j.l}$. It obviously implies that,

$$\Pr\left\{V_{i.k.t}^{j.l.s} | \overline{U}_{i.k.t}\right\} = 0$$

$$\Pr\left\{V_{i.k.t}^{j.l.s} | \overline{W}_{i.k}^{j.l}\right\} = 0$$

$$\Pr\left\{V_{i.k.t}^{j.l.s} | U_{i.k.t} \bigcap \overline{W}_{i.k}^{j.l}\right\} = 0$$

So we can conclude the following,

$$
\begin{aligned}
\Pr\left\{V_{i.k.t}^{j.l.s}\right\} &= \Pr\left\{V_{i.k.t}^{j.l.s}|U_{i.k.t}\right\}\Pr\left\{U_{i.k.t}\right\} + \Pr\left\{V_{i.k.t}^{j.l.s}|\overline{U}_{i.k.t}\right\}\Pr\left\{\overline{U}_{i.k.t}\right\} \\
&= \Pr\left\{V_{i.k.t}^{j.l.s}|U_{i.k.t}\right\}\Pr\left\{U_{i.k.t}\right\} \\
&= \begin{pmatrix} \Pr\left\{V_{i.k.t}^{j.l.s}|U_{i.k.t}\bigcap W_{i.k}^{j.l}\right\}\Pr\left\{W_{i.k}^{j.l}|U_{i.k.t}\right\} \\ +\Pr\left\{V_{i.k.t}^{j.l.s}|U_{i.k.t}\bigcap \overline{W}_{i.k}^{j.l}\right\}\Pr\left\{\overline{W}_{i.k}^{j.l}|U_{i.k.t}\right\} \end{pmatrix}\Pr\left\{U_{i.k.t}\right\} \\
&= \Pr\left\{V_{i.k.t}^{j.l.s}|U_{i.k.t}\bigcap W_{i.k}^{j.l}\right\}\Pr\left\{W_{i.k}^{j.l}|U_{i.k.t}\right\}\Pr\left\{U_{i.k.t}\right\}
\end{aligned} \tag{4.2.1}
$$

The events $U_{i.k.t}$, $V_{i.k.t}^{j.l.s}$, $W_{i.k}^{j.l}$ are only well defined for $i.k$, $j.l \in N$ and $(i.k, j.l) \in A$. We can, however, remain consistent with our definitions and define these events for other situations.

There may be instances in which the UUV starts the mission later than the intended start time (time 0). We will represent a late start by using the node $S.0.0$. Starting at time t is like transiting between the general mission start node $S.0.0$ and the node $S.1.t$. So we will say $\Pr\{V_{S.0.0}^{S.1.t}\} = \Pr\{\text{The mission starts at time } t\}$. Let us assume we are certainly going to do the mission, so staying in line with our event definitions we can say $1 = \Pr\{\text{We do the mission}\} = \Pr\{U_{S.0.0}\}$. Similarly, since we also know we are going to do the mission we know that we must start the mission at some time so we can say $\Pr\{W_{S.0}^{S.1}|U_{S.0.0}\} = 1$. Now using the result shown in 4.2.1 using conditional probabilities we can say,

$$
\begin{aligned}
\Pr\{\text{The mission starts at time } t\} &= \Pr\{V_{S.0.0}^{S.1.t}\} \\
&= \Pr\left\{V_{S.0.0}^{S.1.t}|U_{S.0.0}\bigcap W_{S.0}^{S.1}\right\}\Pr\left\{W_{S.0}^{S.1}|U_{S.0.0}\right\}\Pr\left\{U_{S.0.0}\right\} \\
&= \Pr\left\{V_{S.0.0}^{S.1.t}|U_{S.0.0}\bigcap W_{S.0}^{S.1}\right\}.
\end{aligned} \tag{4.2.2}
$$

If the UUV is at node $D.0.t$ it represents being at the destination node D at time t but the mission is not yet over, which is why we use the node $D.1.T$. So we can say $V_{D.0.t}^{D.1.T}$ is the event the UUV reaches the destination node at time t and waits for the mission to end at time T. We can safely assume the UUV only goes to the destination node to wait for the end of the mission so we will say $\Pr\{V_{D.0.t}^{D.1.T}|W_{D.0}^{D.1}\bigcap U_{D.0.t}\} = 1 \ \forall\ 0 \le t \le T$.

We will assume the values $\Pr\{V_{i.k.t}^{j.l.s}|U_{i.k.t}\bigcap W_{i.k}^{j.l}\}$ are given as problem parameters so let us adopt the following notation,

Notation 4.5.

$$p_{i.k.t}^{j.l.s} = \Pr\left\{V_{i.k.t}^{j.l.s}|U_{i.k.t}\bigcap W_{i.k}^{j.l}\right\}$$

We will refer to $\mathbf{p}$ as the vector of these values. From 4.2.1 this means,

$$\Pr\{V_{i.k.t}^{j.l.s}\} = p_{i.k.t}^{j.l.s}\Pr\left\{W_{i.k}^{j.l}|U_{i.k.t}\right\}\Pr\left\{U_{i.k.t}\right\} \tag{4.2.3}$$

In words $\mathbf{p}$ has the following meaning,

$$
p_{i.k.t}^{j.l.s} = \begin{cases}
\Pr\left\{\begin{array}{l}\text{The UUV leaves node } i.k \text{ at} \\ \text{time } t \text{ heading towards } j.l \\ \text{and arrives at } j.l \text{ at time } s\end{array}\middle|\begin{array}{l}\text{At time } t \text{ the UUV is at node } i.k \\ \text{and it chooses to go to node } j.l\end{array}\right\} & \forall (i.k, j.l) \in A \\
\Pr\{\text{The UUV starts its mission at time } s\} \text{ for } j.l = S.1 \text{ and } i.k.t = S.0.0 & \\
1 = \Pr\left\{\begin{array}{l}\text{The UUV waits at the destination} \\ \text{node for the mission to end}\end{array}\middle|\begin{array}{l}\text{The UUV is at the} \\ \text{destination node at time } t\end{array}\right\} & \begin{array}{l}\text{for } j.l.s = D.1.T, \\ i.k = D.0, \text{ and } 0 \le t \le T\end{array}
\end{cases}
$$

It may initially seem unrealistic to assume that $\mathbf{p}$ is a problem parameter. Consider driving from Baltimore to Washington DC. It is reasonable to assume we know a probability distribution for the time it will take to make the drive. The time it takes to make the drive depends highly on when we leave, i.e. if we left at 4 AM we might get to Washington DC faster than if we left at 4 PM. This leads us to believe it is reasonable to assume there is a probability distribution for the time it will take to make the drive from Baltimore to Washington DC given that we leave at a certain time. This is the exact situation we have, only with discrete time. So continuing with this example we are saying,

$$\Pr\left\{\begin{array}{l|l} \text{We leave Baltimore at time } t \text{ and} & \text{At time } t \text{ we are in Baltimore and} \\ \text{get to Washington DC at time } s & \text{leave to go to Washington DC} \end{array}\right\}$$

$$\sim \Pr\left\{\begin{array}{l|l} \text{The UUV leaves node } i.k \text{ at time } t \text{ heading} & \text{At time } t \text{ the UUV is at node } i.k \\ \text{towards } j.l \text{ and arrives at } j.l \text{ at time } s & \text{and it chooses to go to node } j.l \end{array}\right\}$$

Now we will use the assumed values of $\mathbf{p}$ to create the stochastic time expanded graph.

Definition 4.6 (Stochastic Time Expanded Graph). *The* ***stochastic time expanded graph*** G_s *is a graph with the following properties,*

- $A_s = \{(i.k.t, j.l.s) | p_{i.k.t}^{j.l.s} > 0 \text{ and } 0 \leq t, s \leq T\}$ *is the arc set for the graph.*
- $G_s = (N_s, A_s)$ *is the stochastic time expanded graph with node set* N_s *and arc set* A_s*.*
- $\boldsymbol{B}_s$ *will represent the node-arc incident matrix for the stochastic time expanded graph* G_s*.*
- $\boldsymbol{d}_s(\alpha)$ *will represent the demand vector for a flow problem with* $S.0.0$ *having a supply of* 1*,* $D.1.T$ *having a demand of* α*, and all other nodes* $i.k.t \in N_s$ *having balanced flow.*

Since we already had the node set N_s we only needed to create the arc set A_s to have the stochastic time expanded graph. We add the arcs to the graph for which $p_{i.k.t}^{j.l.s} > 0$, which are the only arcs the UUV could feasibly ever travel on. Let us build an example graph to help us understand exactly how this would work.

Example Problem 3

- The node the UUV starts at is node S.
- The node the UUV must end at is node D.
- There is one node, node 1, which has a task and we need to decided to do the task or not. The reward for doing the task at 1 is one.
- We want to be at the destination node D by time 4.

The transit times are random with the following characteristics,

Departure Node	Arrival Node	Transit Time with Probabilities		
Mission Start ($S.0$)	$S.1$	0 with prob 1/3	1 with prob 1/3	2 with prob 1/3
$S.1$	1.0	1 with prob 1/3	2 with prob 1/3	3 with prob 1/3
$S.1$	$D.0$	1 with prob 1/2	2 with prob 1/2	
1.0	1.1	1 with prob 1/2	2 with prob 1/2	
1.1	$D.0$	1 with prob 1/2	2 with prob 1/2	

In Example 3 the transit times do not change depending on the departure time (like they do in the general case) but are stochastic to show us how the graph is built with stochastic transit times.

First we must be certain that Example 3 provides us with the parameter $\mathbf{p}$. Let us consider the transit from node $S.1$ to node $D.0$. We see in the table in Example 3 that it takes 1, or 2 time units to travel from S to D each with probability 1/2. This means that the probability it takes one time unit to travel from node $S.1$ to node $D.0$ is always 1/2. Or more specifically, if the UUV leaves node $S.1$ at time t headed towards node $D.0$, the probability it arrives at $D.0$ at time $t+1$ is 1/2. This previous statement is exactly the type of statement needed to produce the $\mathbf{p}$ values, so $\mathbf{p}$ is in fact a parameter for this problem. When we build the stochastic time expanded graph we get what is shown in Figure 4.2.

Reiterating what the nodes and arcs represent in Figure 4.2,

- Nodes of type $i.0.t$ represent the UUV being at node i at time step t before performing the task at the location.
- Nodes of type $i.1.t$ represent the UUV being at node i at time step t after performing the task at the location.

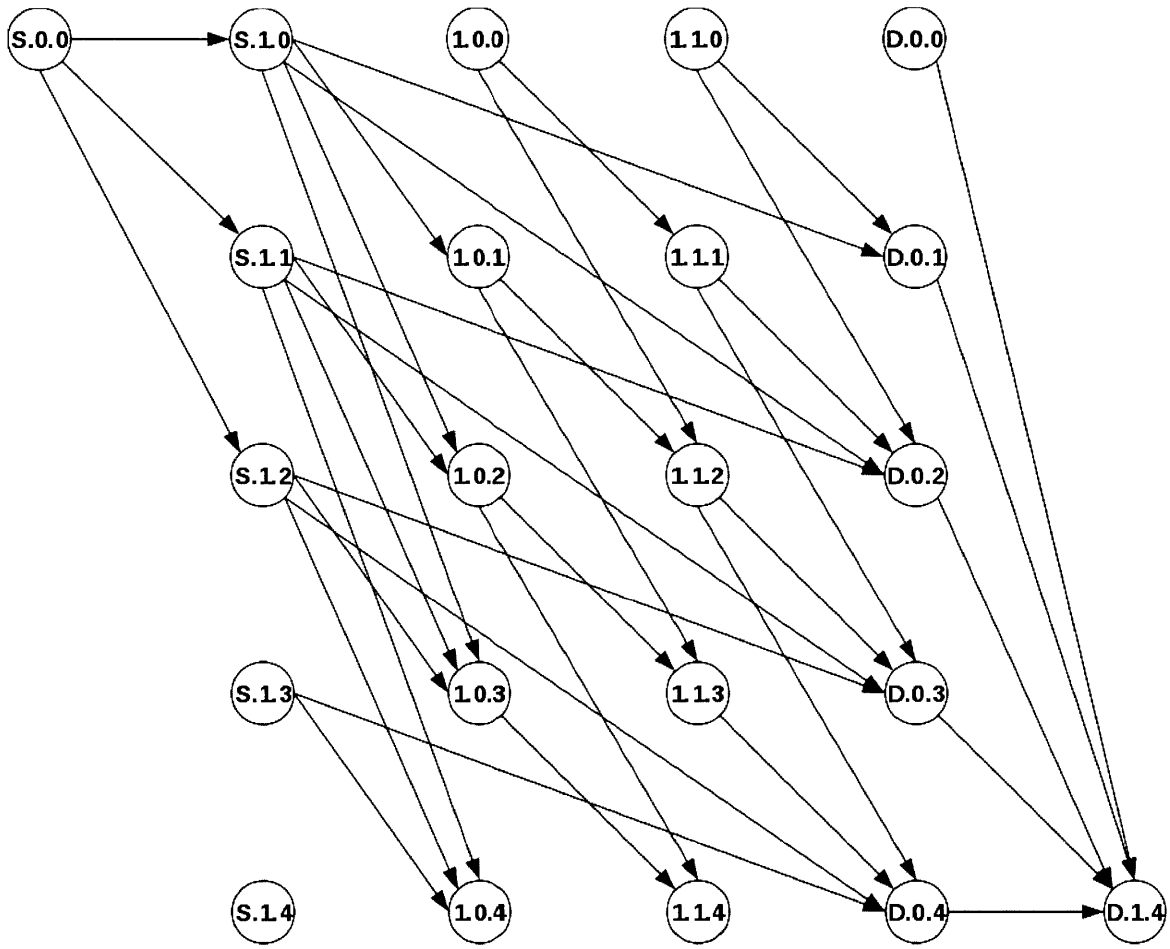


FIGURE 4.2. Stochastic Time Expanding Graph

- Node $S.0.0$ is a super source start node which represents the start of the mission.
- Node $D.1.4$ is a super sink destination node which represents the end of the mission.
- Arcs of type $(i.1.t, j.0.s) \in A_s$ represent transiting from node i to node j in $s - t$ time units.
- Arcs of type $(i.0.t, i.1.s) \in A_s$ represent performing the task at node i in $s - t$ time units.

Let us look at the transit from $S.1$ to 1.0 to help us understand why the example stochastic time expanded graph looks like it does. Looking at the subgraph in Figure 4.3 we see the following properties,

- Arcs $(S.1.0, 1.0.1)$, $(S.1.0, 1.0.2)$, and $(S.1.0, 1.0.3)$ are in the graph because if the UUV leaves node $S.1$ at time 0 to go to node 1.0 the UUV has a positive probability of getting to node 1.0 at time 1, 2, or 3.
- Arcs $(S.1.1, 1.0.2)$, $(S.1.1, 1.0.3)$, and $(S.1.1, 1.0.4)$ are in the graph because if the UUV leaves node $S.1$ at time 1 to go to node 1.0 the UUV has a positive probability of getting to node 1.0 at time 2, 3, or 4.
- Arcs $(S.1.2, 1.0.3)$, and $(S.1.2, 1.0.4)$ are in the graph because if the UUV leaves node $S.1$ at time 2 to go to node 1.0 the UUV has a positive probability of getting to node 1.0 at time 3, 4. There is also a positive probability of the UUV getting to node 1.0 at time 5 but this is not included in the graph because it is after the end time 4, which corresponds to a failure.
- Arc $(S.1.3, 1.0.4)$ is in the graph because if the UUV leaves node $S.1$ to go to node 1.0 at time 3 the UUV has a positive probability of getting to node 1.0 at time 4. There is also a positive probability

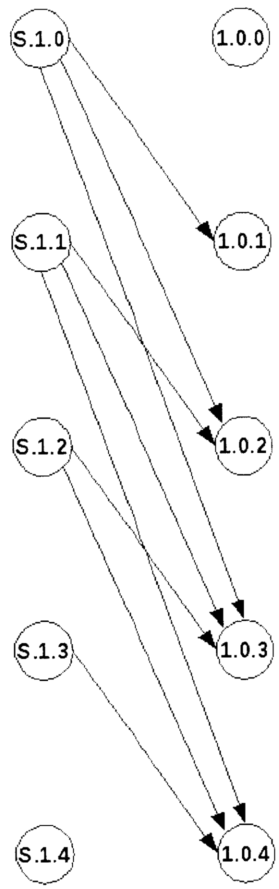


FIGURE 4.3. Stochastic Time Expanded Subgraph

of the UUV getting to node 1.0 at time 5 or 6 but this is not included in the graph because it is after the end time 4, which corresponds to a failure.

In the deterministic time expanded graph we were able to assume there were no overtaking arcs. Similarly, in the stochastic time expanded graph there is a simple transformation described in A.1 which allows us to assume the probability the UUV reaches node $j.l$ by time m leaving node $i.k$ at time t is greater than or equal to the probability the UUV reaches node $j.l$ by time m leaving node $i.k$ at a time after t. More formally, we will assume G_s has the following property,

$$\sum_{s=0}^{m} p_{i.k.t}^{j.l.s} \geq \sum_{s=0}^{m} p_{i.k.t^+}^{j.l.s} \ \forall\ t \leq t^+ \ \forall 0 \leq m \leq T.$$

Similar to the deterministic graph, if there were a cycle in G_s it would have to have the form $(i_1.k_1.t \rightarrow i_2.k_2.t \rightarrow \ldots \rightarrow i_M.k_M.t \rightarrow i_1.k_1.t)$ for a specific time t. If a cycle like this were actually in graph G_s it would mean that for all of the transitions $i_m.k_m \rightarrow i_{m+1}.k_{m+1}$ there is a positive probability they take zero time. Since we have assumed that the UUV only goes to a node if it completes the task at that node, this means that there is a positive probability it takes zero time to complete all of the tasks $i_1, \ldots, i_M$ and it takes zero time to transit in between those tasks. Similar to the deterministic time expanded graph, cycles like this do not appear in realistic problems the UUV will face so we will assume there are no cycles in the graph G_s. More formally we will assume it takes at least one time unit to complete a task to remove all possibility of cycles in graph G_d.

We have established the stochastic time expanded graph so we now will establish decision variables to solve the problem at hand.

4.3. Constructing the Formulation

We will again use decision variables on the decision graph G. The decision variables for this formulation will be,

$y_{i.k}^{j.l} =$ the number of times the UUV chooses to go from $i.k$ to $j.l$ where $(i.k, j.l) \in A$

This definition of $\mathbf{y}$ is worded differently than in previous chapters to highlight the fact that it represents how the UUV will choose to transit through the decision graph G. As in Chapters 2 and 3 we make $\mathbf{y}$ binary to limit the number of times the UUV performs a task to one. The UUV must choose to go on a path from the start node to the destination node so as in Chapters 2 and 3 the $\mathbf{y}$ decision variable must satisfy,

$$\mathbf{By} = \mathbf{d}.$$

The goal is to calculate and bound the probabilities $\Pr\{V_{i.k.t}^{j.l.s}\} = p_{i.k.t}^{j.l.s} \Pr\{W_{i.k}^{j.l}|U_{i.k.t}\} \Pr\{U_{i.k.t}\}$ for a route $\mathbf{y}$. To do this we will use a flow variable $\mathbf{x}$ on graph G_s and put $p_{i.k.t}^{j.l.s} \Pr\{W_{i.k}^{j.l}|U_{i.k.t}\} \Pr\{U_{i.k.t}\}$ in terms of $\mathbf{y}$ and $\mathbf{x}$. Since the decision variable $y_{i.k}^{j.l}$ and the event $W_{i.k}^{j.l}$ both deal with transit in the decision graph G we would like to be able to make the statement $p_{i.k.t}^{j.l.s} * \Pr\{W_{i.k}^{j.l}|U_{i.k.t}\} * \Pr\{U_{i.k.t}\} = p_{i.k.t}^{j.l.s} * y_{i.k}^{j.l} * \Pr\{U_{i.k.t}\}$. We prove this statement formally in Lemma 4.7 by using the definitions of the events $U_{i.k.t}$, $V_{i.k.t}^{j.l.s}$, and $W_{i.k}^{j.l}$ to observe that $y_{i.k}^{j.l} = 0 \rightarrow \Pr\{V_{i.k.t}^{j.l.s}\} = 0$ and $y_{i.k}^{j.l} = 1 \rightarrow \Pr\{W_{i.k}^{j.l}\} = 1$. We cannot say that $\Pr\{W_{i.k}^{j.l}|U_{i.k.t}\} = y_{i.k}^{j.l}$ because if $\Pr\{U_{i.k.t}\} = 0$ then $y_{i.k}^{j.l} = 0$ but it is unclear as to what $\Pr\{W_{i.k}^{j.l}|U_{i.k.t}\}$ would be. In words this means that when there is a positive probability the UUV goes to node $i.k$ then the probability we go to node $j.l$ next is equal to $y_{i.k}^{j.l}$, but if there is zero probability we go to node $i.k$ then $y_{i.k}^{j.l} = 0$ but it is unclear as to what the probability of going to node $j.l$ would be given that we were at node $i.k$ which we have zero probability of reaching.

Lemma 4.7.

$$p_{i.k.t}^{j.l.s} * \Pr\{W_{i.k}^{j.l}|U_{i.k.t}\} * \Pr\{U_{i.k.t}\} = p_{i.k.t}^{j.l.s} * y_{i.k}^{j.l} * \Pr\{U_{i.k.t}\} \ \forall (i.k, j.l) \in A \ 0 \le s, t \le T$$

PROOF. Let $(i.k, j.l) \in A$ and $t, s \ge 0$. We know $\mathbf{y}$ is binary so let us consider the following cases,

Case 1: Assume $y_{i.k}^{j.l} = 0$. This means that the UUV never chooses to go to node $j.l$ from node $i.k$. Therefore the UUV can never leave node $i.k$ heading for $j.l$ so by definition $\Pr\{V_{i.k.t}^{j.l.s}\} = 0$ which means,

$$p_{i.k.t}^{j.l.s} * y_{i.k}^{j.l} * \Pr\{U_{i.k.t}\} = 0 = \Pr\{V_{i.k.t}^{j.l.s}\} = p_{i.k.t}^{j.l.s} * \Pr\{W_{i.k}^{j.l}|U_{i.k.t}\} * \Pr\{U_{i.k.t}\}.$$

Case 2: Assume $y_{i.k}^{j.l} = 1$. This means that the UUV chooses to go to node $j.l$ from node $i.k$. Therefore if we are given the fact that the UUV is at node $i.k$ it will certainly go to node $j.l$ next. By definition this implies that $\Pr\{W_{i.k}^{j.l}|U_{i.k.t}\} = 1$. So clearly it follows that,

$$p_{i.k.t}^{j.l.s} * \Pr\{W_{i.k}^{j.l}|U_{i.k.t}\} * \Pr\{U_{i.k.t}\} = p_{i.k.t}^{j.l.s} * y_{i.k}^{j.l} * \Pr\{U_{i.k.t}\}.$$

So we have proved the result is true for an arbitrary $(i.k, j.l) \in A$ and $0 \le t, s \le T$ in both cases. □

We previously discussed that $\Pr\{U_{S.0.0}\} = 1$ because event $U_{S.0.0}$ represents starting the mission and we have assumed we are certainly going to start the mission. For all other nodes $i.k.t \in N_s$ the probability the UUV goes to node $i.k.t$ must be equal to the probability the UUV travels on an arc which goes into node $i.k.t$. So we can say,

$$\Pr\{U_{i.k.t}\} = \sum_{j.l.s|(j.l.s,i.k.t)\in A_s} \Pr\{V_{j.l.s}^{i.k.t}\} \ \forall \ i.k.t \in N_s \setminus \{S.0.0\} \tag{4.3.1}$$

This means that $\forall (i.k, j.l) \in A, \ 0 \le s, t \le T$,

$$\Pr\left\{V_{i.k.t}^{j.l.s}\right\} = p_{i.k.t}^{j.l.s} * \Pr\{U_{i.k.t}\} * y_{i.k}^{j.l} = p_{i.k.t}^{j.l.s} * \left(\sum_{i'.k'.t'|(i'.k'.t',i.k.t)\in A_s} \Pr\{V_{i'.k'.t'}^{i.k.t}\} \right) * y_{i.k}^{j.l}. \tag{4.3.2}$$

Since $\mathbf{y}$ is binary we know from 4.3.2 the following inequalities must hold $\forall (i.k, j.l) \in A, \ 0 \le s, t \le T$,

$$\Pr\{V_{i.k.t}^{j.l.s}\} \leq y_{i.k}^{j.l} \tag{4.3.3}$$

$$\Pr\{V_{i.k.t}^{j.l.s}\} \leq p_{i.k.t}^{j.l.s} * \left(\sum_{i'.k'.t' | (i'.k'.t', i.k.t) \in A_s} \Pr\{V_{i'.k'.t'}^{i.k.t}\} \right) \tag{4.3.4}$$

Lemma 4.8 shows that Equation 4.3.3 is equivalent to a much stronger and useful constraint. The result in the lemma follows by observing which of the $V_{i.k.t}^{j.l.s}$ events are mutually exclusive. Specifically we observe that since the UUV can only perform a task one time and we only go to a node if we complete the task at that node, the event $V_{i.k.t_1}^{j_1.l_1.s_1}$ is mutually exclusive from event $V_{i.k.t_2}^{j_2.l_2.s_2}$ if either $t_1 \neq t_2$ or $j_1.l_1.s_1 \neq j_2.l_2.s_2$.

Lemma 4.8.

$$\Pr\{V_{i.k.t}^{j.l.s}\} \leq y_{i.k}^{j.l} \; \forall (i.k, j.l) \in A \; 0 \leq s,t \leq T \Leftrightarrow \sum_{t,s|(i.k.t,j.l.s) \in A_s} \Pr\{V_{i.k.t}^{j.l.s}\} \leq y_{i.k}^{j.l} \; \forall \; (i.k, j.l) \in A$$

PROOF. $(\Rightarrow)$ Assume,

$$\Pr\{V_{i.k.t}^{j.l.s}\} \leq y_{i.k}^{j.l} \; \forall (i.k, j.l) \in A, \; 0 \leq s,t \leq T. \tag{4.3.5}$$

Let $(i.k, j.l) \in A$. We know $\mathbf{y}$ is binary so consider each case.

Case 1: Assume $y_{i.k}^{j.l} = 0$. Then from 4.3.5,

$$\begin{aligned} \Pr\{V_{i.k.t}^{j.l.s}\} \leq 0 \; \forall \; 0 \leq s,t \leq T \to & \sum_{t,s|(i.k.t,j.l.s) \in A_s} \Pr\{V_{i.k.t}^{j.l.s}\} \leq 0 \\ \to & \sum_{t,s|(i.k.t,j.l.s) \in A_s} \Pr\{V_{i.k.t}^{j.l.s}\} \leq y_{i.k}^{j.l} \end{aligned}$$

Case 2: Assume $y_{i.k}^{j.l} = 1$. Since we we are assuming we can only do the task at a node once it means the UUV can only transit out of node $i.k$ one time. This means that the event $V_{i.k.t_1}^{j_1.l_1.s_1}$ is mutually exclusive from the event $V_{i.k.t_2}^{j_2.l_2.s_2}$ if either $t_1 \neq t_2$ or $j_1.l_1.s_1 \neq j_2.l_2.s_2$. This means,

$$\begin{aligned} \sum_{t.j'.l'.s|(i.k.t,j'.l'.s) \in A_s} \Pr\{V_{i.k.t}^{j'.l'.s}\} \leq 1 \to & \sum_{t,s|(i.k.t,j.l.s) \in A_s} \Pr\{V_{i.k.t}^{j.l.s}\} \leq 1 \\ \to & \sum_{t,s|(i.k.t,j.l.s) \in A_s} \Pr\{V_{i.k.t}^{j.l.s}\} \leq y_{i.k}^{j.l} \end{aligned}$$

So in each case we have that

$$\sum_{t,s|(i.k.t,j.l.s) \in A_s} \Pr\{V_{i.k.t}^{j.l.s}\} \leq y_{i.k}^{j.l}.$$

$(\Leftarrow)$ Assume

$$\sum_{t,s|(i.k.t,j.l.s) \in A_s} \Pr\{V_{i.k.t}^{j.l.s}\} \leq y_{i.k}^{j.l} \; \forall \; (i.k, j.l) \in A.$$

Let $(i.k, j.l) \in A$ and $0 \leq s,t \leq T$. Then,

$$\begin{aligned} \Pr\{V_{i.k.t}^{j.l.s}\} &\leq \sum_{t',s'|(i.k.t',j.l.s') \in A_s} \Pr\{V_{i.k.t'}^{j.l.s'}\} \\ &\leq y_{i.k}^{j.l} \end{aligned}$$

□

Now assume the decision variable $\mathbf{y}$ satisfies $\mathbf{By} = \mathbf{d}$ and consider the linear programming problem 4.3.6 where $\mathbf{x}$ is a flow variable on graph G_s meant to calculate the values $\Pr\{V_{i.k.t}^{j.l.s}\}$,

$$
\begin{array}{lll}
(4.3.6a) & \max & \alpha \\
(4.3.6b) & \text{s.t.} & \mathbf{B}_s\mathbf{x} \leq \mathbf{d}_s(\alpha) \\
(4.3.6c) & & \displaystyle\sum_{s.t|(i.k.t,j.l.s)\in A_s} x_{i.k.t}^{j.l.s} \leq y_{i.k}^{j.l} \quad \forall (i.k, j.l) \in A \\
(4.3.6d) & & p_{i.k.t}^{j.l.s} * \left(\displaystyle\sum_{i'.k'.t'|(i'.k'.t',i.k.t)\in A_s} x_{i'.k'.t'}^{i.k.t} \right) \geq x_{i.k.t}^{j.l.s} \quad \forall (i.k.t, j.l.s) \in A_s \\
(4.3.6e) & & x_{S.0.0}^{S.1.t} \leq p_{S.0.0}^{S.1.t} \ \forall \ t \text{ such that } (S.0.0, S.1.t) \in A_s \\
(4.3.6f) & & 0 \leq \mathbf{x} \leq 1 \quad \alpha \geq 0
\end{array}
$$

Inequality 4.3.3, Lemma 4.8, and Inequality 4.2.2 show that for $\mathbf{x}$ to calculate the probabilities $\Pr\{V_{i.k.t}^{j.l.s}\}$ it must satisfy 4.3.6c, 4.3.6d, and 4.3.6e respectively. Constraint 4.3.6b bounds $\mathbf{x}$ to ensure its values are valid probabilities. The value α is the sum of the flow variable $\mathbf{x}$ into node $D.1.T$ which is equal to the probability the UUV reaches the destination node by time T if $\mathbf{x}$ accurately calculates the values $\Pr\{V_{i.k.t}^{j.l.s}\}$.

We want to show that an optimal solution $\mathbf{x}$ to problem 4.3.6 calculates (for all intents and purposes) the values $\Pr\{V_{i.k.t}^{j.l.s}\}$. We will do this in two steps with Lemmas 4.9 and 4.10. Lemma 4.9 first establishes the bound $x_{i.k.t}^{j.l.s} \leq \Pr\{V_{i.k.t}^{j.l.s}\}$ and then Lemma 4.10 shows us that $x_{i.k.t}^{j.l.s} = \Pr\{V_{i.k.t}^{j.l.s}\}$ is an optimal solution to problem 4.3.6.

Lemma 4.9 shows that if $\mathbf{y}$ satisfies $\mathbf{By} = \mathbf{d}$ then a feasible solution $\mathbf{x}$ to problem 4.3.6 satisfies $x_{i.k.t}^{j.l.s} \leq \Pr\{V_{i.k.t}^{j.l.s}\}$. To prove this we first observe that the only $\mathbf{x}$ values which can be nonzero are the arcs corresponding to the route in $\mathbf{y}$. Then the lemma shows that $x_{j.l.s}^{S.1.t} \leq \Pr\{V_{j.l.s}^{S.1.t}\}$ and by induction continues to show $x_{i.k.t}^{j.l.s} \leq \Pr\{V_{i.k.t}^{j.l.s}\}$ $\forall (i.k.t, j.l.s) \in A_s$.

Lemma 4.9. *For a decision vector $\boldsymbol{y}$ that satisfies $\boldsymbol{By} = \mathbf{d}$ and a feasible solution $\boldsymbol{x}$, α to problem 4.3.6, $x_{i.k.t}^{j.l.s} \leq \Pr\{V_{i.k.t}^{j.l.s}\}$ $\forall$ $(i.k.t, j.l.s) \in A_s$.*

PROOF. Assume Formulation 4.3.6 is feasible and $\mathbf{x}$, α is a feasible solution for a given route $\mathbf{y}$ that satisfies $\mathbf{By} = \mathbf{d}$. The we know $\mathbf{y}$ contains a path P of the form,

$$P = \{S.1 = i_0.k_0 \to i_1.k_1 \to i_2.k_2 \to i_M.k_M = D.0\}.$$

Constraint 4.3.6c tells us $x_{i.k.t}^{j.l.s} = 0 = \Pr\{V_{i.k.t}^{j.l.s}\}$ if $i.k \in P$ $j.l \notin P$, or $i.k \notin P$ $j.l \in P$. Since there are no cycles in graph G_s constraint 4.3.6b tells us $x_{i.k.t}^{j.l.s} = 0 = \Pr\{V_{i.k.t}^{j.l.s}\}$ $\forall$ $i.k \notin P$ $j.l \notin P$.

So now we only need to consider $x_{i.k.t}^{j.l.s}$ such that $i.k \to j.l$ is in P. Constraint 4.3.6e says $x_{S.0.0}^{S.1.t} \leq p_{S.0.0}^{S.1.t} = \Pr\{V_{S.0.0}^{S.1.t}\}$ $\forall (S.0.0, S.1.t) \in A_s$. Therefore,

$$\sum_{j.l.s|(j.l.s,S.1.t)\in A_s} x_{j.l.s}^{S.1.t} \leq \sum_{j.l.s|(j.l.s,S.1.t)\in A_s} \Pr\{V_{j.l.s}^{S.1.t}\} = \Pr\{U_{S.1.t}\} \ \forall 0 \leq t \leq T.$$

For the induction step assume, $0 \leq m < M$, $i_m.k_m.t \in N_s$ and,

$$\sum_{j.l.s|(j.l.s,i_m.k_m.t)\in A_s} x_{j.l.s}^{i_m.k_m.t} \leq \Pr\{U_{i_m.k_m.t}\}$$

Then from constraint 4.3.6d

$$x_{i_m.k_m.t}^{i_{m+1}.k_{m+1}.s} \leq p_{i_m.k_m.t}^{i_{m+1}.k_{m+1}.s} * \left(\sum_{j.l.s|(j.l.s,i_m.k_m.t)\in A_s} x_{j.l.s}^{i_m.k_m.t} \right) \leq p_{i_m.k_m.t}^{i_{m+1}.k_{m+1}.s} * \Pr\{U_{i_m.k_m.t}\}$$

This combined with the fact that $x_{i_m.k_m.t}^{i_{m+1}.k_{m+1}.s} \leq y_{i_m.k_m}^{i_{m+1}.k_{m+1}} = 1$ tells us,

$$x_{i_m.k_m.t}^{i_{m+1}.k_{m+1}.s} \leq p_{i_m.k_m.t}^{i_{m+1}.k_{m+1}.s} * \Pr\{U_{i_m.k_m.t}\} * y_{i_m.k_m}^{i_{m+1}.k_{m+1}} = \Pr\{V_{i_m.k_m.t}^{i_{m+1}.k_{m+1}.s}\}.$$

Finally resulting in the inequality,

$$\sum_{j.l.t|(j.l.t,i_{m+1}.k_{m+1}.s)\in A_s} x_{j.l.t}^{i_{m+1}.k_{m+1}.s} \le \sum_{j.l.s|(j.l.t,i_{m+1}.k_{m+1}.s)\in A_s} \Pr\{V_{j.l.t}^{i_{m+1}.k_{m+1}.s}\} = \Pr\{U_{i_{m+1}.k_{m+1}.s}\}.$$

□

Lemma 4.10 shows that if $\mathbf{y}$ satisfies $\mathbf{By} = \mathbf{d}$ then $x_{i.k.t}^{j.l.s} = \Pr\{V_{i.k.t}^{j.l.s}\}$ is a feasible, optimal solution to problem 4.3.6. To prove feasibility we use the geometry of graph G_s, and the mutual exclusiveness of events $V_{i.k.t}^{j.l.s}$. To prove optimality we use the result from Lemma 4.9.

Lemma 4.10. *For a given problem we have the route* $\boldsymbol{y}$ *such that* $\boldsymbol{By} = \mathbf{d}$ *then* $\exists\ \boldsymbol{x}$ *and* α *such that* $\boldsymbol{x}$ *and* α *are the optimal solution to linear programming problem 4.3.6 and* $x_{i.k.t}^{j.l.s} = \Pr\{V_{i.k.t}^{j.l.s}\}$ $\forall (i.k.t, j.l.s) \in A_s$.

Proof. Let

$$x_{i.k.t}^{j.l.s} = \Pr\{V_{i.k.t}^{j.l.s}\} \ \forall\ (i.k.t, j.l.s) \in A_s$$
$$\text{and } \alpha = \sum_{i.k.t|(i.k.t,D.1.T)\in A_s} x_{i.k.t}^{D.1.T}$$

We first will show that $\mathbf{x}$, α is a feasible solution to problem 4.3.6, and then show that $\mathbf{x}$, α is optimal.

Feasibility: By setting $x_{i.k.t}^{j.l.s} = \Pr\{V_{i.k.t}^{j.l.s}\}$ we know from 4.2.2, 4.3.3, and 4.3.4 that $\mathbf{x}$ satisfies constraints 4.3.6c, 4.3.6d, and 4.3.6e so we only need to check the flow constraints 4.3.6b. Therefore we only need to show that the flow constraints are satisfied for each node $i.k.t \in N_s$.

Case 1: Let $i.k.t = S.0.0$. For node $S.0.0$ constraint 4.3.6b says the flow out of node $S.0.0$ must be less than or equal to 1. Let us consider the value of the flow out of node $S.0.0$,

$$\begin{aligned}\sum_{i.k.t|(S.0.0,i.k.t)\in A_s} x_{S.0.0}^{i.k.t} &= \sum_{t|(S.0.0,S.1.t)\in A_s} x_{S.0.0}^{S.1.t} \text{ because only arcs of type } (S.0.0, S.1.t) \in A_s\\ &= \sum_{t|(S.0.0,S.1.t)\in A_s} \Pr\{V_{S.0.0}^{S.1.t}\}\\ &= \sum_{t=0}^{T} \Pr\{V_{S.0.0}^{S.1.t}\}\\ &= \Pr\{\text{ the mission starts at or before time } T\}\\ &\le 1\end{aligned}$$

Case 2: Let $i.k.t = D.1.T$. For node $D.1.T$ constraint 4.3.6b says the flow in must be greater than or equal to $-alpha$. We set α to be equal to the flow into node $D.1.T$. Therefore the constraint just says $\alpha \le \alpha$, which is clearly true.

Case 3: Let $i.k.t \in N_s$ such that $i.k \in N$. For node $i.k.t$ constraint 4.3.6b says the flow out minus the flow in must be less than or equal to zero. So we have,

$$\begin{aligned}&\sum_{j.l.s|(i.k.t,j.l.s)\in A_s} x_{i.k.t}^{j.l.s} - \sum_{j.l.s|(j.l.s.i.k.t)\in A_s} x_{j.l.s}^{i.k.t}\\ =& \sum_{j.l.s|(i.k.t,j.l.s)\in A_s} \Pr\{V_{i.k.t}^{j.l.s}\} - \sum_{j.l.s|(j.l.s,i.k.t)\in A_s} \Pr\{V_{j.l.s}^{i.k.t}\}\\ =& \sum_{j.l.s|(i.k.t,j.l.s)\in A_s} \left(p_{i.k.t}^{j.l.s} * y_{i.k}^{j.l} * \Pr\{U_{i.k.t}\}\right) - \Pr\{U_{i.k.t}\} \quad \text{from 4.3.1}\\ =& \left(\sum_{j.l.s|(i.k.t,j.l.s)\in A_s} \left(p_{i.k.t}^{j.l.s} * y_{i.k}^{j.l}\right) - 1\right) * \Pr\{U_{i.k.t}\} \quad \text{since } \mathbf{y} \text{ is binary}\\ \le& \left(\sum_{j.l.s|(i.k.t,j.l.s)\in A_s} \left(p_{i.k.t}^{j.l.s}\right) - 1\right) * \Pr\{U_{i.k.t}\}\end{aligned}$$

Since we know event $V_{i.k.t}^{j.l.s}$ is mutually exclusive from event $V_{i.k.t}^{j'.l'.s'}$ if $j.l.s \neq j'.l'.s'$ then we know

$$\sum_{j.l.s|(i.k.t,j.l.s)\in A_s} \left(p_{i.k.t}^{j.l.s}\right) \leq 1.$$

Therefore,

$$\begin{aligned}&\sum_{j.l.s|(i.k.t,j.l.s)\in A_s} x_{i.k.t}^{j.l.s} - \sum_{j.l.s|(j.l.s,i.k.t)\in A_s} x_{j.l.s}^{i.k.t}\\ \leq&\left(\sum_{j.l.s|(i.k.t,j.l.s)\in A_s} \left(p_{i.k.t}^{j.l.s}\right) - 1\right) * \Pr\{U_{i.k.t}\}\\ \leq&(1-1) * \Pr\{U_{i.k.t}\} = 0\end{aligned}$$

So in all cases the flow constraints are met. Therefore, $\mathbf{x}$ and α are feasible for problem 4.3.6. Optimality: Now assume $\mathbf{v}$, γ is a feasible solution to problem 4.3.6. From lemma 4.9 we conclude the following,

$$v_{i.k.t}^{j.l.s} \leq \Pr\{V_{i.k.t}^{j.l.s}\} = x_{i.k.t}^{j.l.s}\ \forall (i.k.t, j.l.s) \in A_s.$$

This result combined with the flow constraint on node $D.1.T$ tell us,

$$\begin{aligned}\gamma \leq& \sum_{i.k.t|(i.k.t,D.1.T)\in A_s} v_{i.k.t}^{D.1.T}\\ \leq& \sum_{i.k.t|(i.k.t,D.1.T)\in A_s} x_{i.k.t}^{D.1.T}\\ =&\ \alpha\end{aligned}$$

Therefore the objective function value for the solution $\mathbf{x}$, α is greater than or equal to the objective function value for all other feasible solutions.

Therefore $\mathbf{x}$, α is a feasible solution to problem 4.3.6 in which all other feasible solutions have a objective function value less than or equal to α. Hence, $\mathbf{x}$, α is an optimal solution to problem 4.3.6. □

Lemma 4.10 is the key step to being able to bound the probability the UUV reaches the destination node by time T. We have proven that a solution $\mathbf{x}$ to 4.3.6 is bounded above by the values $\Pr\{V_{i.k.t}^{j.l.s}\}$ and that $x_{i.k.t}^{j.l.s} = \Pr\{V_{i.k.t}^{j.l.s}\}$is an optimal solution. This means that if we bound $\mathbf{x}$ below we are in fact bounding $\Pr\{V_{i.k.t}^{j.l.s}\}$ below, and if we maximize over $\mathbf{x}$ then we are maximizing over the values $\Pr\{V_{i.k.t}^{j.l.s}\}$, which is exactly what we will do in Formulation 4.3.7.

Now we will take a moment to show that in Chapter 3 for Formulation 3.3.1 using the deterministic time expanded graph we can either have $\mathbf{x}$ binary or $0 \leq \mathbf{x} \leq 1$. We establish this property in three steps with corollaries 4.11, 4.12, and 4.13. Corollary 4.11 establishes the decreasing nature of a feasible $\mathbf{x}$ to problem 4.3.6 on a given path P. Corollary 4.12 shows that if we have a feasible $\mathbf{x}$, α to problem 4.3.6 such that $\alpha = 1$ then $x_{i.k.t}^{j.l.s} = \Pr\{V_{i.k.t}^{j.l.s}\}$. Corollary 4.13 then concludes to show that we can relax $\mathbf{x}$ in Formulation 3.3.1 to $0 \leq \mathbf{x} \leq 1$ and $\mathbf{x}$ remains binary.

Corollary 4.11 shows us the decreasing nature of the $\mathbf{x}$ values on a path. To prove the corollary we use the fact that the only $\mathbf{x}$ which can be nonzero are the arcs corresponding to the path in $\mathbf{y}$ (which we also used in lemma 4.9), and we use constraint 4.3.6b.

Corollary 4.11. *For a decision vector $\boldsymbol{y}$ which contains a path P of the form,*

$$P = \{S.1 = i_1.k_1 \to i_2.k_2 \to \ldots \to i_M.k_M = D.0\}$$

if $\boldsymbol{x}$, α is a feasible solution to problem 4.3.6 then,

$$\sum_{j.l.s,t|(i_{m_1}.k_{m_1}.t,j.l.s)\in A_s} x_{i_{m_1}.k_{m_1}.t}^{j.l.s} \geq \sum_{j.l.s,t|(i_{m_2}.k_{m_2}.t,j.l.s)\in A_s} x_{i_{m_2}.k_{m_2}.t}^{j.l.s}\ \forall m_1 \leq m_2.$$

PROOF. Let $1 \leq m \leq M$. In lemma 4.9 we established that if $i.k \to j.l$ is not in the path P then $x_{i.k.t}^{j.l.s} = 0 \;\forall\; s, t$. Therefore proving the corollary is equivalent to proving,

$$\sum_{j.l.s,t|(i_m.k_m.t,j.l.s)\in A_s} x_{i_m.k_m.t}^{j.l.s} \geq \sum_{j.l.s,t|(i_{m+1}.k_{m+1}.t,j.l.s)\in A_s} x_{i_{m+1}.k_{m+1}.t}^{j.l.s}$$

The corollary result is now straightforward,

$$\begin{aligned}
\sum_{j.l.s,t|(i_m.k_m.t,j.l.s)\in A_s} x_{i_m.k_m.t}^{j.l.s} &= \sum_{s,t|(i_m.k_m.t,i_{m+1}.j_{m+1}.s)\in A_s} x_{i_m.k_m.t}^{i_{m+1}.k_{m+1}.s} \text{ since } x_{i_m.k_m.t_m}^{j.l.s} = 0 \text{ if } j.l \neq i_{m+1}.k_{m+1} \\
&= \sum_{j.l.s,t|(j.l.s,i_{m+1}.k_{m+1}.t)\in A_s} x_{j.l.s}^{i_{m+1}.k_{m+1}.t} \text{ since } x_{j.l.s}^{i_{m+1}.k_{m+1}.t_{m+1}} = 0 \text{ if } j.l \neq i_m.k_m \\
&\geq \sum_{j.l.s,t|(i_{m+1}.k_{m+1}.t,j.l.s)\in A_s} x_{i_{m+1}.k_{m+1}.t}^{j.l.s} \text{ from inequality 4.3.6}b
\end{aligned}$$

The first two equalities follow from the fact that the values of $\mathbf{x}$ not on the path depicted by $\mathbf{y}$ are zero and the greater than or equal to inequality follows from inequality 4.3.6b. □

Corollary 4.12 shows that if there is a feasible solution $\mathbf{x}$ to Problem 4.3.6 such that the flow into node $D.1.T$ is one then $x_{i.k.t}^{j.l.s} = \Pr\{V_{i.k.t}^{j.l.s}\} \;\forall (i.k.t, j.l.s) \in A_s$. To prove this we first use lemma 4.8 where we observed the mutual exclusiveness of the events $V_{i.k.t}^{j.l.s}$. Then we use lemma 4.9 and we have the result.

Corollary 4.12. *For a decision vector* $\boldsymbol{y}$ *which satisfies* $\boldsymbol{By} = \boldsymbol{d}$ *if* $\exists \boldsymbol{x}$, α *such that they are feasible to problem 4.3.6 and* $\alpha = 1$ *then* $x_{i.k.t}^{j.l.s} = \Pr\{V_{i.k.t}^{j.l.s}\} \;\forall\; (i.k.t, j.l.s) \in A_s$.

PROOF. From lemma 4.9 we know we only need to show $x_{i.k.t}^{j.l.s} \geq \Pr\{V_{i.k.t}^{j.l.s}\} \;\forall\; (i.k.t, j.l.s) \in A_s$. We know that a path P must be contained in $\mathbf{y}$ such that,

$$P = \{S.1 = i_1.k_1 \to i_2.k_2 \to \ldots \to i_M.k_M = D.0\}.$$

Let $1 \leq m < M$.

$$\begin{aligned}
1 &= y_{i_m.k_m}^{i_{m+1}.k_{m+1}} \\
&\geq \sum_{t,s|(i_m.k_m.t,i_{m+1}.k_{m+1}.s)\in A_s} \Pr\{V_{i_m.k_m.t}^{i_{m+1}.k_{m+1}.s}\} \text{ from Lemma 4.8} \\
&\geq \sum_{t,s|(i_m.k_m.t,i_{m+1}.k_{m+1}.s)\in A_s} x_{i_m.k_m.t}^{i_{m+1}.k_{m+1}.s} \text{ from Lemma 4.9} \\
&\geq \sum_{t|(D.0.t,D.1.T)\in A_s} x_{D.0.t}^{D.1.T} \text{ from Corollary 4.11} \\
&\geq \alpha \quad \text{from Constraint 4.3.6}b \\
&= 1
\end{aligned}$$

The first greater than or equal to inequality follows from Lemma 4.8 where we observed that the events $V_{i.k.t_1}^{j_1.l_1.s_1}$ and $V_{i.k.t_2}^{j_2.l_2.s_2}$ are mutually exclusive if either $t_1 \neq t_2$ or $j_1.l_1.t_1 \neq j_2.l_2.t_2$. The second greater than or equal to inequality follows from Lemma 4.9. The third inequality follows from Corollary 4.11.

So now the result follows straightforward,

$$\begin{aligned}
\Pr\{V_{i_m.k_m.t_1}^{i_{m+1}.k_{m+1}.t_2}\} &= 1 - \sum_{s_1\neq t_1,s_2\neq t_2|(i_m.k_m.s_1,i_{m+1}.k_{m+1}.s_2)\in A_s} \Pr\{V_{i_m.k_m.s_1}^{i_{m+1}.k_{m+1}.s_2}\} \text{ from above equality} \\
&\leq 1 - \sum_{s_1\neq t_1,s_2\neq t_2|(i_m.k_m.s_1,i_{m+1}.k_{m+1}.s_2)\in A_s} x_{i_m.k_m.s_1}^{i_{m+1}.k_{m+1}.s_2} \text{ from Lemma 4.9} \\
&= x_{i_m.k_m.t_1}^{i_{m+1}.k_{m+1}.t_2} \text{ from above equality}
\end{aligned}$$

□

Corollary 4.13 shows us that in the deterministic Formulation 3.3.1 we can equivalently replace $\mathbf{x}$ binary with $0 \leq \mathbf{x} \leq 1$. To prove this we first observe that in the deterministic case G_s is essentially the same graph as G_d. Then we take a feasible $0 \leq \mathbf{x} \leq 1$ to Problem 3.3.1, show it is feasible in 4.3.6 and use Corollary 4.12 to show that $x_{i.k.t}^{j.l.s} = \Pr\{V_{i.k.t}^{j.l.s}\}$. Then we make the observation that $\Pr\{V_{i.k.t}^{j.l.s}\}$ is binary in a deterministic case, which gives us the result that $\mathbf{x}$ is binary.

Corollary 4.13. *Let $\boldsymbol{y}$ and $\boldsymbol{x}$ be feasible solutions to a relaxed version of 3.3.1 where we replace the constraint $\boldsymbol{x}$ binary with $0 \leq \boldsymbol{x} \leq 1$. Then $\boldsymbol{x}$ is binary.*

Proof. If we build the graph G_s for a deterministic problem we know from A.2 there is a simple transformation to the graph G_d which allows us to consider the flow $\mathbf{x}$ as a flow for problem 4.3.7. By satisfying constraints 3.3.1c and 3.3.1d we know that $\mathbf{x}$ must also satisfy inequalities 4.3.6b, and 4.3.6c. Since we are dealing with a deterministic problem $p_{i.k.t}^{j.l.s} = 1 \; \forall \; (i.k.t, j.l.s) \in A_s$, so $\mathbf{x}$ satisfies 4.3.6d since it satisfies 3.3.1c. Therefore $\mathbf{x}$ is feasible in problem 4.3.6. Equality 3.3.1c also tells us,

$$\sum_{0 \leq t \leq T} x_{D.0.t}^{D.1.T} = 1.$$

This means that $\mathbf{x}$, $\alpha = 1$ is a feasible solution to problem 4.3.6. From Corollary 4.12 this means $x_{i.k.t}^{j.l.s} = \Pr\{V_{i.k.t}^{j.l.s}\} \; \forall (i.k.t, j.l.s)$. Since we are dealing with a deterministic problem $\Pr\{V_{i.k.t}^{j.l.s}\}$ is binary, which means $\mathbf{x}$ is binary. □

Returning to the stochastic time expanded formulation, we will use the result of Lemma 4.10 to constrain the probability of reaching the destination node D by the end time step T. We are going to think of $U_{D.1.T}$ as the event the UUV reaches the destination node by time step T, so we want to include the constraint $\Pr\{U_{D.1.T}\} \geq \beta$. Substituting in for $U_{D.1.T}$ we want the constraint,

$$\beta \leq \sum_{i.k.t | (i.k.t, D.1.T) \in A_s} \Pr\{V_{i.k.t}^{D.1.T}\}.$$

Since $\mathbf{x}$ is bounded above by the values $\Pr\{V_{i.k.t}^{j.l.s}\}$ and $x_{i.k.t}^{j.l.s} = \Pr\{V_{i.k.t}^{j.l.s}\}$ is a feasible solution to 4.3.6, then A.3 shows that the constraint $\Pr\{U_{D.1.T}\} \geq \beta$ is equal to the following constraints,

$$\begin{aligned}
\alpha &\geq \beta \\
\mathbf{B}_s\mathbf{x} &\leq \mathbf{d}_s(\alpha) \\
\sum_{s,t | (i.k.t, j.l.s) \in A_s} x_{i.k.t}^{j.l.s} &\leq y_{i.k}^{j.l} \quad \forall (i.k, j.l) \in A \\
p_{i.k.t}^{j.l.s} * \left(\sum_{i'.k'.t' | (i'.k'.t', i.k.t) \in A_s} x_{i'.k'.t'}^{i.k.t} \right) &\geq x_{i.k.t}^{j.l.s} \quad \forall (i.k.t, j.l.s) \in A_s \\
x_{S.0.0}^{S.1.t} &\leq p_{S.0.0}^{S.1.t} \;\; \forall t \text{ such that } (S.0.0, S.1.t) \in A_s \\
0 \leq \mathbf{x} \leq 1 &\quad \alpha \geq 0
\end{aligned}$$

Or we can plug β in for α and use the equivalent constraints,

$$\mathbf{B}_s\mathbf{x} \leq \mathbf{d}_s(\beta)$$

$$\sum_{s,t|(i.k.t,j.l.s)\in A_s} x_{i.k.t}^{j.l.s} \leq y_{i.k}^{j.l} \quad \forall (i.k,j.l) \in A$$

$$p_{i.k.t}^{j.l.s} * \left(\sum_{i'.k'.t'|(i'.k'.t',i.k.t)\in A_s} x_{i'.k'.t'}^{i.k.t} \right) \geq x_{i.k.t}^{j.l.s} \quad \forall (i.k.t,j.l.s) \in A_s$$

$$x_{S.0.0}^{S.1.t} \leq p_{S.0.0}^{S.1.t} \ \forall t \text{ such that } (S.0.0, S.1.t) \in A_s$$

$$0 \leq x \leq 1$$

In this stochastic setting it is unclear as to what we want the objective function to be, and in the same respect how much we want to penalize the UUV for having a positive probability of not reaching the destination node in time. Right now we will maximize the expected value for the reward gained for a route. Let us assume that r_i is the reward for completing the task at i, so we want to maximize the following,

$$\max \sum_i r_i * \Pr\{\text{Completing task i before time } T\} = \sum_i r_i * \left(\sum_{t,s|(i.0.t,i.1.s)\in A_s} \Pr\{V_{i.0.t}^{i.1.s}\} \right).$$

Since $\mathbf{x}$ is bounded above by the values $\Pr\{V_{i.k.t}^{j.l.s}\}$ and $x_{i.k.t}^{j.l.s} = \Pr\{V_{i.k.t}^{j.l.s}\}$ is a feasible solution to 4.3.6 we can equivalently maximize the value,

$$\sum_i r_i * \left(\sum_{t,s|(i.0.t,i.1.s)\in A_s} x_{i.0.t}^{i.1.s} \right)$$

By constraining $\mathbf{x}$ from below and maximizing over $\mathbf{x}$ Lemmas 4.9 and 4.10 show us that we are in fact bounding $\Pr\{V_{i.k.t}^{j.l.s}\}$ from below and maximizing over these probabilities. Due to this strong connecting we will use the values $x_{i.k.t}^{j.l.s}$ and $\Pr\{V_{i.k.t}^{j.l.s}\}$ interchangeably.

Now consider the following stochastic time expanded formulation,

$$\text{(4.3.7a)} \quad \max \quad \sum_{i=1}^{n} \left(r_i * \left(\sum_{t,s|(i.0.t,i.1.s)\in A_s} x_{i.0.t}^{i.1.s} \right) \right)$$

$$\text{(4.3.7b)} \quad \text{s.t.} \quad \mathbf{B}\mathbf{y} = \mathbf{d}$$

$$\text{(4.3.7c)} \quad \mathbf{B}_s\mathbf{x} \leq \mathbf{d}_s(\beta)$$

$$\text{(4.3.7d)} \quad \sum_{s,t|(i.k.t,j.l.s)\in A_s} x_{i.k.t}^{j.l.s} \leq y_{i.k}^{j.l} \quad \forall (i.k,j.l) \in A$$

$$\text{(4.3.7e)} \quad p_{i.k.t}^{j.l.s} * \left(\sum_{i'.k'.t'|(i'.k'.t',i.k.t)\in A_s} x_{i'.k'.t'}^{i.k.t} \right) \geq x_{i.k.t}^{j.l.s} \quad \forall (i.k.t,j.l.s) \in A_s$$

$$\text{(4.3.7f)} \quad x_{S.0.0}^{S.1.t} \leq p_{S.0.0}^{S.1.t} \quad \forall t \text{ such that } (S.0.0, S.1.t) \in A_s$$

$$\text{(4.3.7g)} \quad \mathbf{y} \text{ binary }, 0 \leq \mathbf{x} \leq 1$$

This formulation maximizes the expected reward gained while ensuring that the UUV reaches the destination node by time T with probability β. Again, we need to address the situation of cycles in $\mathbf{y}$. Let us assume we have a cycle $i.0 \to i.1 \to j.0 \to j.1 \to i.0$ in the decision vector $\mathbf{y}$. If this situation occurs then we know that the probability of actually reaching these nodes is zero, so the values $\Pr\{V_{i.k.t}^{j.l.s}\}$ are zero for these nodes, which means the corresponding variables $x_{i.k.t}^{j.l.s}$ for these nodes are zero. Since we are maximizing over values of $\mathbf{x}$ this cycle would not add anything to the objective function, so we do not need to worry about cycles in our solutions.

The value of β greatly affects the solution to Formulation 4.3.7. In Example 3 it is easy to calculate the probability of reaching the destination node. If the UUV tries to performs the task at node 1 then the UUV has a probability of 5/36 of reaching the destination node by time 4, and if the UUV does not perform the task at node 1 then the UUV has a probability of 1 of reaching the destination node by time 4. Formulation 4.3.7 for Example 3 gives the solutions,

β	Solution
$0 \leq \beta \leq 5/36$	$S.1 \to 1.0 \to 1.1 \to D.0$
$5/36 < \beta \leq 1$	$S.1 \to D.0$

These solutions tell us that if we want the UUV to have a probability of greater than 5/36 of reaching the destination node by time 4 we cannot do the task at node 1, but otherwise we can. This is consistent with the problem parameters in Example 3.

Formulation 4.3.7 allows us to handle time windows and stochastic parameters, but the solutions may be somewhat naive. When the transit times are realized we may want to complete more tasks if they are lower than expected, or less tasks if higher than expected. One way to model this would be to attach a time to the nodes in N which tells the UUV to continue with a route if the UUV arrives before a certain time, and alter the route if the UUV arrives after a certain time. This is what we will add to our model in the next section.

4.4. How to Allow Route Alteration

We want to modify Formulation 4.3.7 so that the UUV can now alter the route based on the realization of the transit times. We are going to allow the UUV to alter the route in two ways:

(1) When the UUV is at a node of type $i.0$ it can choose to skip performing the task and go to the next node.
(2) When the UUV is at a node of type $i.1$ it can choose to go to the destination node $D.0$ instead of continuing on with the route.

To allow the UUV to skip performing a task we will add arcs of type $(i.0.t, i.1.t)$ to the graph G_s. For simplicity we have assumed each task takes at least one time unit to complete. This means that the arcs of type $(i.0.t, i.1.t)$ are not in graph G_s, so there will be no duplicate arcs added. Adding these arcs does mean that the resulting graph is more likely to have cycles, but if we do not place any positive reward on these arcs there will be no positive cost cycles in the new graph. This means our solutions will not be affected by the possibility of having cycles [**1**]. These arcs are the only arcs we will add to create the route alteration graph G_a.

Definition 4.14 (Route Alteration Graph). *The **route alteration graph** G_a is a graph with the following form,*

- $A_a = A_s \bigcup \{(i.0.t, i.1.t) | (i.0, i.1) \in G$ *and* $0 \leq t \leq T\}$.
- $G_a = (N_s, A_a)$ *is the route alteration graph with node set* N_s *and arc set* A_a.
- $\boldsymbol{B}_a$ *will represent the node arc incident matrix for the route alteration graph* G_a.
- $\boldsymbol{d}_a(\alpha)$ *will represent the demand vector for a flow problem with* $S.0.0$ *having a supply of* 1, $D.1.T$ *having a demand of* α *and all other nodes* $i.k.t \in N_s$ *having balance flow in graph* G_a.

Due to the close connection between A_s and A_a there are many constraints which can be written using either set. Often the distinction is unimportant, but we must remember that A_a has arcs of type $(i.0.t, i.1.t)$ and A_s does not.

The route alteration graph has arcs of type $(i.0.t, i.1.t)$ which the UUV could travel on to skip performing task i. Looking again at Example 3 the route alteration graph for the problem looks as in Figure 4.4. Let us look at the arcs out of nodes of type $(1.0.t)$ shown in the subgraph in Figure 4.5 to understand the arcs which have been added to the route alteration graph.

- Arcs of type $(1.0.t, 1.1.t + 1)$ are in the graph because if the UUV does not alter the route at node $1.0.t$ it has a positive probability of going to node $1.1.t + 1$.
- Arcs of type $(1.0.t, 1.1.t + 2)$ are in the graph because if the UUV does not alter the route at node $1.0.t$ it has a positive probability of going to node $1.1.t + 2$.

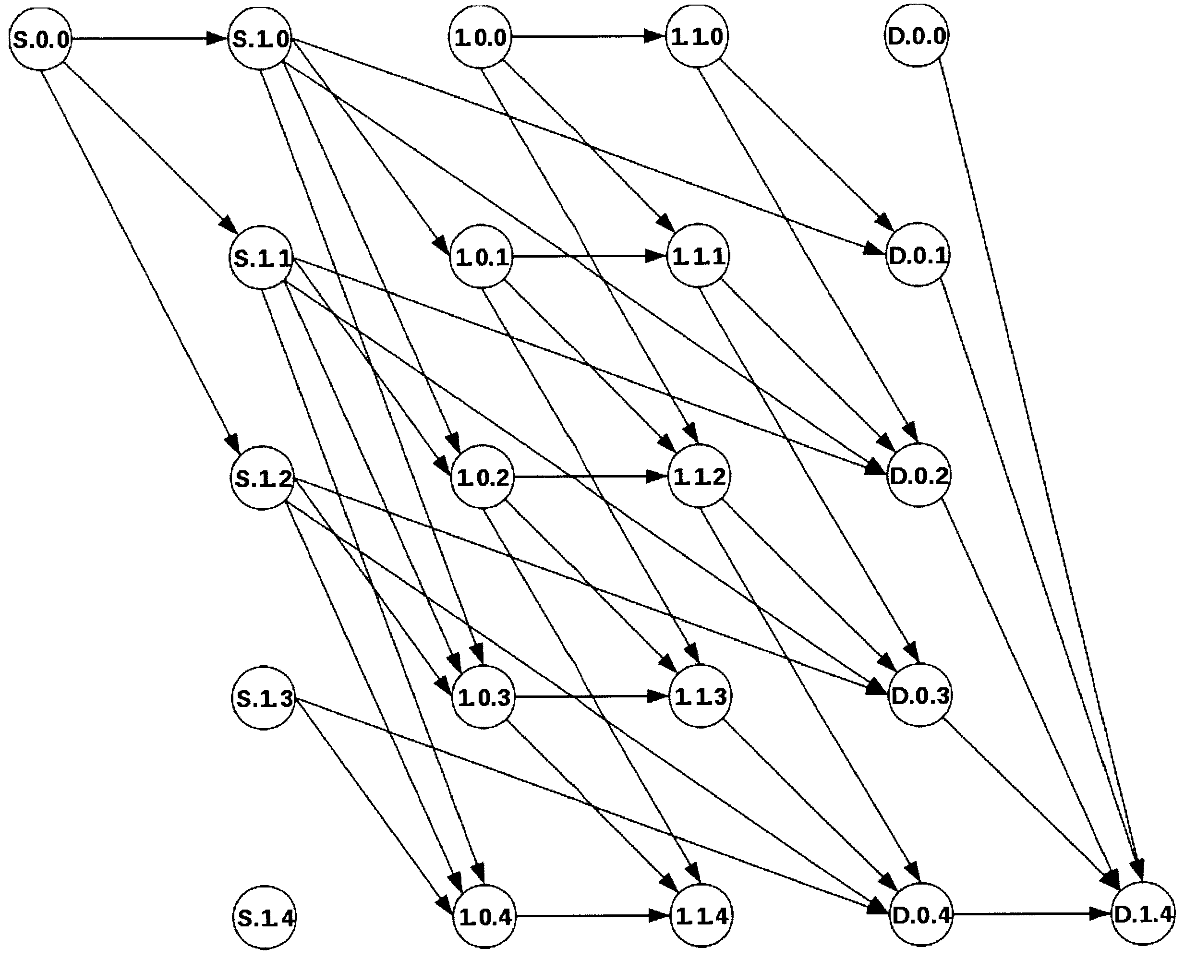


FIGURE 4.4. Route Alteration Example Graph

- Arcs of type $(1.0.t, 1.1.t)$ are in the graph because if the UUV alters the route and chooses not to perform the task at 1 it goes to node $1.1.t$ with no reward gained.

We will use the variables $\mathbf{y}$ for the UUV's route and $\mathbf{x}$ to calculate transition probabilities from Formulation 4.3.7 and add variables and constraints to allow for route alteration.

Let,

$$z_{i.0.t} = \begin{cases} 1 \text{ if the UUV skips performing the task at node } i.0.t \\ 0 \text{ otherwise} \end{cases}$$

$$z_{i.1.t} = \begin{cases} 1 \text{ if the UUV goes to the destination node from node } i.1.t \\ 0 \text{ otherwise} \end{cases}$$

We know the route must be a path from the start node to the destination node so as in Chapter 2 the UUV still must satisfy,

$$\mathbf{B}\mathbf{y} = \mathbf{d}.$$

To calculate the probabilities $\Pr\{V_{i.k.t}^{j.l.s}\}$, $\mathbf{x}$ must still satisfy the constraints,

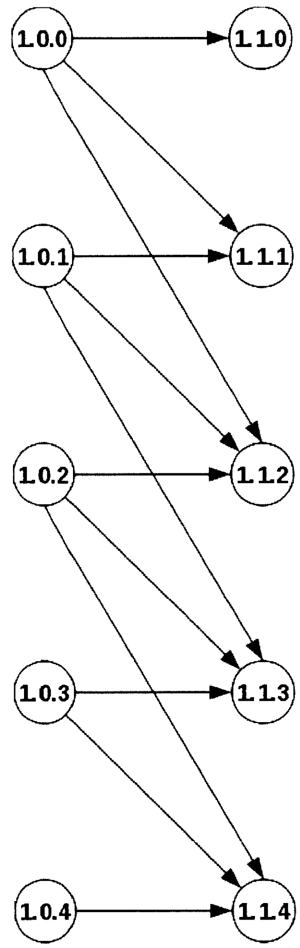


FIGURE 4.5. Route Alteration Example Subgraph

$$\mathbf{B}_a \mathbf{x} \leq \mathbf{d}_a(\beta)$$

$$p_{i.k.t}^{j.l.s} * \left(\sum_{i'.k'.t' | (i'.k'.t', i.k.t) \in A_s} x_{i'.k'.t'}^{i.k.t} \right) \geq x_{i.k.t}^{j.l.s} \quad \forall (i.k.t, j.l.s) \in A_s$$

$$x_{S.0.0}^{S.1.t} \leq p_{S.0.0}^{S.1.t} \quad \forall t \text{ such that } (S.0.0, S.1.t) \in A_s$$

$$\mathbf{y} \text{ binary }, 0 \leq \mathbf{x} \leq 1$$

For arcs $(i.k, j.l)$ such that $j.l \neq D.0$ we know we can only transit on them if $y_{i.k}^{j.l} = 1$ so we get the constraint,

$$\sum_{s,t | (i.k.t.j.l.s) \in A_s} x_{i.k.t}^{j.l.s} \leq y_{i.k}^{j.l} \; \forall (i.k, j.l) \in A | j.l \neq D.0.$$

If the UUV is at node $i.0.t$ the UUV can go to node $i.1.t$ if and only if $z_{i.0.t} = 1$ so we know the following inequalities must be satisfied,

$$x_{i.0.t}^{i.1.t} \leq z_{i.0.t}$$

$$\sum_{s>t|(i.0.t,i.1.s)\in A_a} x_{i.0.t}^{i.1.s} \leq 1 - z_{i.0.t}.$$

At a node of type $i.1.t$ the UUV can only go to the destination node if and only if $z_{i.1.t} = 1$ so we know the following inequalities must be satisfied,

$$\sum_{s|(i.1.t,D.0.s)\in A_a} x_{i.1.t}^{D.0.s} \leq z_{i.1.t}$$

$$\sum_{s,j\neq D|(i.1.t,j.0.s)\in A_a} x_{i.1.t}^{j.0.s} \leq 1 - z_{i.1.t}.$$

For simplicity we want to the UUV to establish a specific time t for each node in the decision graph to alter the route, and not alter the route before this time. This means that the UUV cannot alter the route at time t if it does not alter the route at time $t+1$. So we want $\mathbf{z}$ to satisfy,

$$z_{i.k.t} \leq z_{i.k.t+1} \;\forall i.k.t, i.k.t+1 \in N_s.$$

These constraints result in Formulation 4.4.1.

$$
\begin{array}{lll}
(4.4.1a)\ \max \displaystyle\sum_{i=1}^{n}\left(r_i * \left(\sum_{t,s>t|(i.0.t,i.1.s)\in A_s} x_{i.0.t}^{i.1.s}\right)\right) & & \\
(4.4.1b)\text{s.t.} & \mathbf{B}\mathbf{y} = \mathbf{d} & \\
(4.4.1c) & \mathbf{B}_a\mathbf{x} \leq \mathbf{d}_a(\beta) & \\
(4.4.1d) & \displaystyle\sum_{s,t|(i.k.t,j.l.s)\in A_s} x_{i.k.t}^{j.l.s} \leq y_{i.k}^{j.l} & \forall (i.k.t,j.l.s)\in A_s|(i.k,j.l)\in A, j.l\neq D.0 \\
(4.4.1e) & p_{i.k.t}^{j.l.s} * \left(\displaystyle\sum_{i'.k'.t'|(i'.k'.t',i.k.t)\in A_s} x_{i'.k'.t'}^{i.k.t}\right) \geq x_{i.k.t}^{j.l.s} & \forall (i.k.t,j.l.s)\in A_s \\
(4.4.1f) & x_{S.0.0}^{S.1.t} \leq p_{S.0.0}^{S.1.t} & \forall t \text{ such that } (S.0.0,S.1.t)\in A_s \\
(4.4.1g) & x_{i.0.t}^{i.1.t} \leq z_{i.0.t} & \forall i.0.t \in N_s|i.0\neq D.0 \\
(4.4.1h) & \displaystyle\sum_{s>t|(i.0.t,i.1.s)\in A_a} x_{i.0.t}^{i.1.s} \leq 1 - z_{i.0.t} & \forall i.0.t\in N_s \\
(4.4.1i) & \displaystyle\sum_{s|(i.1.t,D.0.s)\in A_a} x_{i.1.t}^{D.0.s} \leq z_{i.1.t} & \forall i.1.t\in N_s \\
(4.4.1j) & \displaystyle\sum_{s,j\neq D|(i.1.t,j.0.s)\in A_a} x_{i.1.t}^{j.0.s} \leq 1 - z_{i.1.t} & \forall i.1.t\in N_s \\
(4.4.1k) & z_{i.k.t} \leq z_{i.k.t+1} & \forall i.k.t, i.k.t+1\in N_s \\
(4.4.1l) & \mathbf{y} \text{ binary}, 0\leq \mathbf{x}\leq 1, \mathbf{z} \text{ binary} &
\end{array}
$$

Formulation 4.4.1 allows the UUV to use the fact that the transit times are stochastic and alter the plan accordingly. The route alteration formulation still does have some drawbacks. It does not allow the UUV to ever skip going to a task location all together to do a later task in the route, or equivalently add tasks which have not been completed yet and are not on the current route. These situations greatly increase the complexity of the problem, so we will not create a formulation which allows for these types of changes. In some instances post processing of a solution to Formulation 4.4.1 would tell the UUV to skip a task in favor of a task further along in the route. For example, assume the solution to Formulation 4.4.1 has the UUV performing the tasks $i_1 \rightarrow i_2 \rightarrow i_3$. Assume that if the UUV is at node $i_1.1.t$ it will go to either node $i_2.0.s_1$

or $i_2.0.s_2$ depending on the realization of the transit time random variable. If the route alteration strategy has $1 = z_{i_2.0.s_1} = z_{i_2.0.s_1}$, then the UUV has zero probability of completing the task at node i_2 if it is at node $i_1.1.t$. With post processing we can identify this situation and conclude that if the UUV arrives at node $i_1.1.t$ it should go straight to node i_3 next.

We can also use solutions to Formulation 4.4.1 (or Formulation 4.3.7) to best decide when the UUV should resolve the problem with the remaining time, energy, and tasks available. One advantage of using Formulation 4.4.1 (4.3.7) is we calculate the probability of reaching certain states. This means when the transit times are realized we know the state the UUV is in and we can tell if the UUV route has been performed relatively fast (or slow) thus far. We can use this knowledge to help the UUV decide when to resolve the problem for the situation going forward.

4.5. The Objective Function

The question of what the reward function should be changes with each mission and the specific consequences a UUV has for its decisions. An advantage of our current formulation is that we calculate the transition probabilities for a given route with $\mathbf{x}$, so we can alter the reward on these arcs.

When altering the objective function we must be careful. We previously observed that in our formulation we do not need to worry about cycles in $\mathbf{y}$ because it is not in our objective function. This means if an objective function was created using the decision vector $\mathbf{y}$ we must calculate if cycles will inappropriately affect the objective function. If this were the case we must reintroduce the cycle eliminating constraints as in the simple formulation in Chapter 2. Also, we have assumed we can use $\mathbf{x}$ interchangeably with $\Pr\{V_{i.k.t}^{j.l.s}\}$ because thus far we have bounded $\mathbf{x}$ below and maximized over $\mathbf{x}$. This means if we use an objective function which maximizes $-\mathbf{x}$ terms we can no longer assume $x_{i.k.t}^{j.l.s} = \Pr\{V_{i.k.t}^{j.l.s}\}$.

We will proceed to discuss different objective functions or alterations to our formulations which may be useful in different situations, but this compilation is by no means complete.

We will discuss ways to alter the objective function for Formulation 4.4.1. Due to the strong similarity between Formulation 4.4.1 and Formulation 4.3.7 all of the following objective functions can be used with Formulation 4.3.7 by making the necessary minor adjustments.

In many situations it may be useful to add the term,

$$\frac{1}{M}\left(\sum_{t=0}^{T} x_{D.0.t}^{D.1.T}\right)$$

to the objective function. This term is using M assuming it is a sufficiently large number. This expression maximizes the flow into node $D.1.T$ but gives it a very small weight as to not effect the optimal solution for which route the UUV should perform. Adding this term to the objective function would ensure the route alteration strategy maximizes the probability of reaching the destination node without affecting the actual reward for the route.

Up to this point we have been using the expected value for the reward gained for performing tasks as the objective function. This objective function provides solutions which have some potentially unwanted behaviors. In Example 3 if we do task 1 then we have a positive probability of reaching node 1.1.4 which represents the UUV being at node 1 at time 4 after performing the task. This is somewhat problematic, however, because the UUV cannot reach the destination node by time 4 but has collected the reward for completing the task at node 1.

A simple solution to this situation can be reached by slightly changing our objective function. Let us say we only want the UUV to collect a reward if a node satisfies,

$$\sum_{s|(i.1.t,D.0.s)\in A_a} p_{i.1.t}^{D.0.s} \geq \gamma,$$

for some $0 \leq \gamma \leq 1$. This statement says that we only want the UUV to collect a reward if it has a probability greater than or equal to γ of reaching the destination node before the end time if the UUV were to go to the destination node next. So let us compile the set of all the nodes which satisfy this constraint,

$$I = \left\{ i.1.t| \sum_{s|(i.1.t,D.0.s)\in A_a} p_{i.1.t}^{D.0.s} \geq \gamma \right\}.$$

All we need to do to change the formulation is set objective function coefficients to zero for arcs entering nodes which are not in I. So we can use the objective function,

$$\max \sum_{i=1}^{n} r_i * \left(\sum_{t,s|(i.0.t,i.1.s)\in A_a, i.1.s\in I} x_{i.0.t}^{i.1.s} \right).$$

We will refer to this objective function as maximizing the expected reward gained over a characterizing set I. We could easily change the character set I to represent a different feature in the situation at hand.

In some situations we may not have a specific value for β in mind but we want the UUV to choose a route which is conservative. To do this we can maximize $r_i * \Pr\{$The UUV does task i and reaches the destination node by time $T\}$. To understand how to do this consider the following events,

Notation 4.15. *$Q_{i.k}^{j.l}$ is the event the UUV travels on arc $(i.k, j.l) \in A$ and reaches the destination node by time T.*

Notation 4.16. *$R_{i.k}$ is the event the UUV visits node $i.k$ and reaches the destination node by time T.*

We know the UUV starts at the node $S.1$ so the probability event $D_{S.1}$ occurs is the probability the UUV reaches the destination node by time T. So we know,

$$\Pr\{R_{S.1}\} = \sum_{t=0}^{T} \Pr\{V_{D.0.t}^{D.1.T}\} \approx \sum_{t=0}^{T} x_{D.0.t}^{D.1.T} \tag{4.5.1}$$

Similarly the probability event $R_{D.0}$ occurs is the probability the UUV reaches reaches the destination node by time T. Therefore,

$$\Pr\{R_{D.0}\} = \sum_{t=0}^{T} \Pr\{V_{D.0.t}^{D.1.T}\} \approx \sum_{t=0}^{T} x_{D.0.t}^{D.1.T} \tag{4.5.2}$$

For nodes other than $S.1$, and $D.0$ the probability event $R_{i.k}$ occurs, is equal to the sum of the probabilities events of type $Q_{j.l}^{i.k}$ occur, and also equal to the sum of the probability events of type $Q_{i.k}^{j.l}$ occur, which can be written as,

$$\sum_{j.l|(j.l,i.k)\in A} \Pr\{Q_{j.l}^{i.k}\} = \Pr\{R_{i.k}\} = \sum_{j.l|(i.k,j.l)\in A} \Pr\{Q_{i.k}^{j.l}\} \tag{4.5.3}$$

For arcs of type $(j.1, D.0)$ (arcs that go to the destination node) the probability we transit on the arcs and reach the destination node by time T is equal to the probability we transit on the arcs, so we have the equality,

$$\Pr\{Q_{j.l}^{D.0}\} = \sum_{t,s|(j.l.s,D.0.t)\in A_s} \Pr\{V_{j.l.s}^{D.0.t}\} \approx \sum_{t,s|(j.l.s,D.0.t)\in A_s} x_{j.l.s}^{D.0.t} \ \forall (j.l, D.0) \in A. \tag{4.5.4}$$

We know that the probability event $C_{i.k}^{j.l}$ occurs is less than or equal to the probability the UUV travels on arc $(i.k, j.l) \in A$ so we know,

$$\Pr\{C_{i.k}^{j.l}\} \leq \sum_{t,s|(i.k.t,j.l.s)\in A_a} \Pr\{V_{i.k.t}^{j.l.s}\} \approx \sum_{t,s|(i.k.t,j.l.s)\in A_a} x_{i.k.t}^{j.l.s}. \tag{4.5.5}$$

As we used the flow variable $\mathbf{x}$ to calculate the values $\Pr\{V_{i.k.t}^{j.l.s}\}$, we will use a flow variable $\mathbf{w}$ to on graph G to calculate the values $\Pr\{Q_{i.k}^{j.l}\}$. The constraints below force $w_{i.k}^{j.l} = \Pr\{Q_{i.k}^{j.l}\}$.

$$\text{(4.5.6a)} \qquad \text{s.t.} \quad \mathbf{Bw} = \left(\sum_{t=0}^{T} x_{D.0.t}^{D.1.T}\right) * \mathbf{d}$$

$$\text{(4.5.6b)} \qquad w_{i.k}^{D.0} = \sum_{t,s|(i.k.t,D.0.s)\in A_a} x_{i.k.t}^{D.0.s} \ \forall (i.k, D.0) \in A$$

$$\text{(4.5.6c)} \qquad w_{i.k}^{j.l} \leq \sum_{t,s|(i.k.t,j.l.s)\in A_a} x_{i.k.t}^{D.0.s} \ \ \forall (i.k, j.l) \in A$$

$$\text{(4.5.6d)} \qquad 0 \leq \mathbf{w} \leq 1$$

Equality 4.5.6a forces $\mathbf{w}$ to satisfy 4.5.1, 4.5.2, and 4.5.3. Equality 4.5.6b forces $\mathbf{w}$ to satisfy 4.5.4. Inequality 4.5.6c forces $\mathbf{w}$ to satisfy 4.5.5.

If the flow variable $\mathbf{w}$ satisfies all Constraints 4.5.6 then $w_{i.k}^{j.l} = \Pr\{Q_{i.k}^{j.l}\}$. To get the desired objective function for Formulation 4.4.1 we add the flow variable $\mathbf{w}$, include all Constraints 4.5.6 and maximize the term,

$$\max \sum_{i=1}^{n} r_i * w_{i.0}^{i.1}.$$

4.6. Reintroducing the Energy Constraint

Up to this point in this chapter we have not considered the energy constraint the UUV faces. We know that time and energy consumption are correlated so by using a time expanded network for time we also gain information about the energy consumption. Let us again consider the situation of driving from Baltimore to Washington DC. Similar to the time it is reasonable to assume we know a distribution for how much fuel we will consume to make the drive. Now assume we know the exact time we are going to leave Baltimore and the exact time we are going to arrive in Washington DC. Knowing this information would tells us a lot about the fuel consumption so it is reasonable to assume that if we know the exact departure and arrival times then we know the probability distribution of the fuel consumption and that the variance of the distribution is relatively small. This is the type of thinking we will apply to our problem.

Let,

$$e_{i.k.t}^{j.l.s} = \left(\begin{array}{c|l} \text{The expected amount} & \text{The UUV leaves node } i.k \text{ at time } t \\ \text{of energy consumed} & \text{and arrives at node } j.l \text{ at time } s \end{array} \right).$$

We will refer to $\mathbf{e}$ as the vector of the values $e_{i.k.t}^{j.l.s}$. Again, we are assuming that $\mathbf{e}$ is the expected energy consumption, and the variance of the actual energy consumed is relatively small.

We will add the energy constraint to both Formulation 4.3.7 and Formulation 4.4.1, but we will handle them separately to avoid confusion.

4.6.1. Energy Linear Programming Problem for Formulation 4.3.7. First consider a feasible solution to Formulation 4.3.7 with the route $\mathbf{y}$ being the result. Consider the graph which has only the arcs the UUV has a positive probability of traveling on. More specifically,

Definition 4.17 (Energy Graph). *The **energy graph** is a graph with the following form,*

- $A_e = \{(i.k.t, j.l.s) | (i.k.t, j.l.s) \in A_s$ *and* $\Pr\{V_{i.k.t}^{j.l.s}\} > 0$ *is the arc set for the energy graph.*
- $G_e = (N_s, A_e)$ *is the energy graph with node set* N_s *and arc set* A_e.
- $\boldsymbol{B}_e$ *will represent the node-arc incident matrix for the energy graph* G_e.
- $\boldsymbol{d}_e$ *will represent the demand vector for a path problem with source* $S.0.0$ *and sink* $D.1.T$ *for the energy graph* G_e.

One way to constrain the energy consumed is to constrain the maximum distance problem on G_e where we start at node $S.0.0$ and end at node $D.1.T$ and the distance associated with each arc is the energy weight on that arc. Ordinarily the maximum distance problem is an NP hard problem. We have assumed G_s has no cycles, which means that G_e has no cycles either, since it is a subgraph of G_s. This means we can solve the longest path problem on G_e with the shortest path problem below,

$$\min -\mathbf{e}'\mathbf{v} \tag{4.6.1a}$$
$$\text{s.t.} \quad \mathbf{B}_e\mathbf{v} = \mathbf{d}_e \tag{4.6.1b}$$
$$\mathbf{v} \geq 0 \tag{4.6.1c}$$

Let us assume that there is a positive probability the UUV reaches the destination node by time step T, which implies that there does exist a path from node $S.0.0$ to node $D.1.T$ in graph G_e. This assumption means the problem 4.6.1 has a feasible solution. We know a finite optimal solution exists because there are no cycles in G_e [**1**].

The goal is to constrain the value of 4.6.1 to be less than the total amount of energy the UUV has available. We used the values $\Pr\{V_{i.k.t}^{j.l.s}\} > 0$ to build graph G_e so we would like to reformulate 4.6.1 in more direct terms of $\Pr\{V_{i.k.t}^{j.l.s}\}$. Let us consider the inequality,

$$v_{i.k.t}^{j.l.s} \leq M * \Pr\{V_{i.k.t}^{j.l.s}\}. \tag{4.6.2}$$

where M is sufficiently large enough so that $M * \Pr\{V_{i.k.t}^{j.l.s}\} > 1 \ \forall \ \Pr\{V_{i.k.t}^{j.l.s}\} > 0$. We know an M exists because there are finitely many positive numbers $\Pr\{V_{i.k.t}^{j.l.s}\}$.

This means that inequality 4.6.2 says,

- $v_{i.k.t}^{j.l.s} = 0$ if $\Pr\{V_{i.k.t}^{j.l.s}\} = 0$.
- $v_{i.k.t}^{j.l.s} \leq M * \Pr\{V_{i.k.t}^{j.l.s}\} > 1$ if $\Pr\{V_{i.k.t}^{j.l.s}\} > 0$.

Since we are dealing with a shortest path problem we know that $v_{i.k.t}^{j.l.s} \leq 1$. So if $\Pr\{V_{i.k.t}^{j.l.s}\} > 0$ inequality 4.6.2 is met automatically.

This means that 4.6.1 is equivalent to 4.6.3.

$$\min -\mathbf{e}'\mathbf{v} \tag{4.6.3a}$$
$$\text{s.t.} \quad \mathbf{B}_s\mathbf{v} = \mathbf{d}_e \tag{4.6.3b}$$
$$v_{i.k.t}^{j.l.s} \leq M * \Pr\{V_{i.k.t}^{j.l.s}\} \ \forall (i.k.t, j.l.s) \in A_s \tag{4.6.3c}$$
$$\mathbf{v} \geq 0 \tag{4.6.3d}$$

Problem 4.6.3 can be thought of as a shortest path problem with flow capacities. Since we know 4.6.1 has a finite optimal solution then the equivalent 4.6.3 has a finite optimal solution.

Instead of using the values $\Pr\{V_{i.k.t}^{j.l.s}\}$ to constrain the shortest path variable $\mathbf{v}$ in 4.6.3 we can alternatively use $\mathbf{y}$. Consider 4.6.4 which we will prove is equivalent to 4.6.3.

$$\min -\mathbf{e}'\mathbf{v} \tag{4.6.4a}$$
$$\text{s.t.} \quad \mathbf{B}_s\mathbf{v} = \mathbf{d}_e \tag{4.6.4b}$$
$$v_{i.k.t}^{j.l.s} \leq 2 * y_{i.k}^{j.l} \ \forall (i.k.t, j.l.s) \in A_s \text{ such that } (i.k, j.l) \in A \tag{4.6.4c}$$
$$\mathbf{v} \geq 0 \tag{4.6.4d}$$

To gain intuition on the differences between problems, Table 1 depicts the connection between $\mathbf{y}$, $\Pr\{V_{i.k.t}^{j.l.s}\}$, and $\mathbf{v}$ in 4.6.3 and 4.6.4. In words table 1 has the following meaning,

- When $y_{i.k} = 0$ then $\Pr\{V_{i.k.t}^{j.l.s}\} = 0$ and $v_{i.k.t}^{j.l.s} = 0$ in problems 4.6.3 and 4.6.4.
- When $y_{i.k} = 1$ then either $\Pr\{V_{i.k.t}^{j.l.s}\} = 0$ or $\Pr\{V_{i.k.t}^{j.l.s}\} > 0$.
- When $y_{i.k} = 1$ and $\Pr\{V_{i.k.t}^{j.l.s}\} = 0$ then $v_{i.k.t} = 0$ in problem 4.6.3 and $0 \leq v_{i.k.t} \leq 1$ in problem 4.6.4.
- When $y_{i.k} = 1$ and $\Pr\{V_{i.k.t}^{j.l.s}\} > 0$ then $0 \leq v_{i.k.t} \leq 1$ in problems 4.6.3 and 4.6.4.

This means that the only time 4.6.3 could give a different solution than 4.6.4 is if $v_{i.k.t}^{j.l.s} > 0$ when $y_{i.k}^{j.l} = 1$ and $\Pr\{V_{i.k.t}^{j.l.s}\} = 0$. The following lemma proves this event can never occur. The lemma uses the fact that a

TABLE 1. Energy Formulation Logic Table

$y_{i.k}^{j.l}$	$=0$	$=1$	
$\Pr\{V_{i.k.t}^{j.l.s}\}$	$=0$	$=0$	>0
$v_{i.k.t}^{j.l.s}$ in problem 4.6.3	$=0$	$=0$	≥ 0
$v_{i.k.t}^{j.l.s}$ in problem 4.6.4	$=0$	≥ 0	≥ 0

feasible $\mathbf{v}$ to 4.6.4 must correspond to a path from node $S.0.0$ to node $D.1.T$ and the only arcs $(i.k.t, j.l.s)$ in the graph G_s are arcs where $p_{i.k.t}^{j.l.s} > 0$.

Lemma 4.18. *For a feasible route* $\mathbf{y}$ *to 4.3.7 that has a positive probability of reaching the destination node* $D.1.T$, *if* $\mathbf{v}$ *is a feasible solution to 4.6.4 then* $\Pr\{V_{i.k.t}^{j.l.s}\} = 0 \rightarrow v_{i.k.t}^{j.l.s} = 0$.

PROOF. Table 1 tells us we only need to consider the case when $(i.k.t, j.l.s) \in A_s$ such that $y_{i.k}^{j.l} = 1$ and $\Pr\{V_{i.k.t}^{j.l.s}\} = 0$. For a contradiction assume we have a feasible $\mathbf{v}$ to problem 4.6.4 such that $v_{i.k.t}^{j.l.s} > 0$. The solution $\mathbf{v}$ must correspond to a path P from node $S.0.0$ to node $D.1.T$. Let,

$$P = \{S.0.0 = i_0.k_0.t_0 \rightarrow i_1.k_1.t_1 \rightarrow \ldots \rightarrow i_M.k_M.t_M = D.1.T\}$$

For $\mathbf{v}$ to be feasible to problem 4.6.4 then $y_{i_m.k_m}^{i_{m+1}.k_{m+1}} = 1$ for all $0 \leq m < M$. Since P is the path described by $\mathbf{v}$ there must be some $0 < h < M$ such that $i_h.k_h.t_h = i.k.t$. The only possible $j.l$ such that $y_{i.k}^{j.l} = 1$ is $j.l = i_{m+1}.k_{m+1}$. By construction of graph G_s we know that since $(S.0.0, i_1.k_1.t_1) \in G_s$ then $0 < p_{S.0.0}^{i_1.k_1.t_1} = \Pr\{V_{i.k.t}^{j.l.s}\}$.

Now for the induction step assume $\Pr\{V_{i_{m-1}.k_{m-1}.t_{m-1}}^{i_m.k_m.t_m}\} > 0$ for some $1 \leq m < M$. Therefore we know $\Pr\{U_{i_m.k_m.t_m}\} > 0$. Since $(i_m.k_m.t_m, i_{m+1}.k_{m+1}.t_{m+1}) \in G_s$ we know $p_{i_m.k_m.t_m}^{i_{m+1}.k_{m+1}.t_{m+1}} > 0$. Therefore $\Pr\{V_{i_m.k_m.t_m}^{i_{m+1}.k_{m+1}.t_{m+1}}\} = p_{i_m.k_m.t_m}^{i_{m+1}.k_{m+1}.t_{m+1}} * y_{i_m.k_m}^{i_{m+1}.k_{m+1}} * \Pr\{U_{i_m.k_m.t_m}\} > 0$.

So we have established that $\Pr\{V_{i_m.k_m.t_m}^{i_{m+1}.k_{m+1}.t_{m+1}}\} > 0$ for all $0 \leq m < M$. Therefore $\Pr\{V_{i_h.k_h.t_h}^{i_{h+1}.k_{h+1}.t_{h+1}}\} = \Pr\{V_{i.k.t}^{j.l.s}\} > 0$ which contradicts the assumption that $\Pr\{V_{i.k.t}^{j.l.s}\} = 0$. □

Lemma 4.18 shows us that we can replace Table 1 with Table 2, which means that 4.6.3 and 4.6.4 are equivalent.

TABLE 2. Energy Formulation Logic Table

$y_{i.k}^{j.l}$	$=0$	$=1$	
$\Pr\{V_{i.k.t}^{j.l.s}\}$	$=0$	$=0$	>0
$v_{i.k.t}^{j.l.s}$ in problem 4.6.3	$=0$	$=0$	≥ 0
$v_{i.k.t}^{j.l.s}$ in problem 4.6.4	$=0$	$=0$	≥ 0

Ultimately we wish to place a constraint on the energy consumption on route the UUV chooses. Problem 4.6.4 finds the worst case energy consumption for a route, so what we would like to do is place a constraint which says that the optimal solution to problem 4.6.4 is less than or equal to the total amount of energy the UUV has available. To do this we need to take the dual of problem 4.6.4 which is shown in problem 4.6.5:

$$\text{(4.6.5a)} \qquad \max \mathbf{d}_e^{'}\mathbf{u} + \sum_{(i.k.t,j.l.s)\in A_s|(i.k,j.l)\in A} 2 * y_{i.k}^{j.l} * w_{i.k.t}^{j.l.s}$$

$$\text{(4.6.5b)} \qquad \text{s.t.} \qquad \begin{bmatrix} \mathbf{B}_e^{'} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{w} \end{bmatrix} \leq -\mathbf{e}$$

$$\text{(4.6.5c)} \qquad w_{S.0.0}^{j.l.s} = 0 \quad \forall\ (S.0.0, j.l.s) \in A_s$$

$$\text{(4.6.5d)} \qquad w_{j.l.s}^{D.1.T} = 0 \quad \forall\ (j.l.s, D.1.T) \in \mathbb{A}_s$$

$$\text{(4.6.5e)} \qquad \mathbf{u} \text{ free } \mathbf{w} \leq 0$$

If we were to strictly take the dual of 4.6.4 the variables of type $w_{S.0.0}^{j.l.s}$ and $w_{j.l.s}^{D.1.T}$ would not be in the problem but for simplicity in notation we include these variables in 4.6.5 and set them to zero.

We concluded previously that 4.6.4 must have a finite optimal solution so its dual, problem 4.6.5, must also have a finite optimal solution by strong duality. Complimentary slackness in linear programming problems says if $\mathbf{w}$ is optimal it must satisfy,

$$\text{(4.6.6)} \qquad w_{i.k.t}^{j.l.s}\left(v_{i.k.t}^{j.l.s} - 2 * y_{i.k}^{j.l}\right) = 0.$$

For background information on strong duality and complimentary slackness see [**9**].

We know that since $\mathbf{v}$ is the decision vector for a shortest path problem that $v_{i.k.t}^{j.l.s} \leq 1$ so by 4.6.6 for $\mathbf{w}$ to be optimal it must satisfy,

$$\text{(4.6.7)} \qquad y_{i.k}^{j.l} = 1 \rightarrow w_{i.k.t}^{j.l.s} = 0.$$

The properties of an optimal $\mathbf{w}$ written in 4.6.7 can also be written with the following constraint,

$$\text{(4.6.8)} \qquad w_{i.k.t}^{j.l.s} \geq M * (y_{i.k}^{j.l} - 1).$$

where M is a sufficiently large number. We will later discuss what a sufficiently large number is for this constraint.

For an optimal $\mathbf{w}$, we can eliminate $\mathbf{w}$ from the objective function because Equation 4.6.8 tells us,

$$\sum_{(i.k.t,j.l.s)\in A_s|(i.k,j.l)\in A} 2 * y_{i.k}^{j.l} * w_{i.k.t}^{j.l.s} = 0.$$

This means that 4.6.5 is equivalent to 4.6.10:

$$\text{(4.6.9a)} \qquad \max \quad \mathbf{d}_e^{'}\mathbf{u}$$

$$\text{(4.6.9b)} \qquad \text{s.t.} \quad \begin{bmatrix} \mathbf{B}_e^{'} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{w} \end{bmatrix} \leq -\mathbf{e}$$

$$\text{(4.6.9c)} \qquad w_{S.0.0}^{j.l.s} = 0 \qquad \forall\ (S.0.0, j.l.s) \in A_s$$

$$\text{(4.6.9d)} \qquad w_{j.l.s}^{D.1.T} = 0 \qquad \forall\ (j.l.s, D.1.T) \in \mathbb{A}_s$$

$$\text{(4.6.9e)} \qquad w_{i.k.t}^{j.l.s} \geq M * (y_{i.k}^{j.l} - 1)\ \forall (i.k.t, j.l.s) \in A_s|(i.k, j.l) \in A$$

$$\text{(4.6.9f)} \qquad \mathbf{u} \text{ free } \mathbf{w} \leq 0$$

By multiplying through by negative one, and filling in the actual values of $\mathbf{d}_e$ we get the equivalent Problem 4.6.11:

$$
\begin{aligned}
&(4.6.10a) && \min u_{D.1.T} - u_{S.0.0} \\
&(4.6.10b) && \text{s.t.} \begin{bmatrix} \mathbf{B}_e^{'} & \mathbf{I} \end{bmatrix} \begin{bmatrix} -\mathbf{u} \\ \mathbf{w} \end{bmatrix} \geq \mathbf{e} \\
&(4.6.10c) && w_{S.0.0}^{j.l.s} = 0 \qquad \forall\ (S.0.0, j.l.s) \in A_s \\
&(4.6.10d) && w_{j.l.s}^{D.1.T} = 0 \qquad \forall\ (j.l.s, D.1.T) \in \mathbb{A}_s \\
&(4.6.10e) && w_{i.k.t}^{j.l.s} \leq M * (1 - y_{i.k}^{j.l}) \ \forall (i.k.t, j.l.s) \in A_s | (i.k, j.l) \in A \\
&(4.6.10f) && \mathbf{u} \text{ free } \mathbf{w} \geq 0
\end{aligned}
$$

4.6.2. Energy Linear Programming Problem for Formulation 4.4.1. The analysis and logic necessary to establish the energy linear programming problem for Formulation 4.4.1 is very similar to what we did to establish the energy constraint in Formulation 4.3.7 with some slight modifications.

Let us first consider a feasible solution to Formulation 4.4.1 with the route $\mathbf{y}$ and alteration strategy $\mathbf{z}$ being the result. For the route alteration formulation we redefine the energy graph using G_a.

Definition 4.19 (Energy Graph). *The* ***energy graph*** *is a graph with the following form,*

- $A_e = \{(i.k.t, j.l.s) | (i.k.t, j.l.s) \in A_a$ *and* $\Pr\{V_{i.k.t}^{j.l.s}\} > 0\}$ *is the arc set for the energy graph.*
- $G_e = (N_s, A_e)$ *is the energy graph with node set* N_s *and arc set* A_e.
- $\boldsymbol{B}_e$ *will represent the node-arc incident matrix for the energy graph* G_e.
- $\boldsymbol{d}_e$ *will represent the demand vector for a path problem with source* $S.0.0$ *and sink* $D.1.T$ *for the energy graph* G_e.

We again want to look at the maximum distance problem on G_e where we start at $S.0.0$ and end at node $D.1.T$ and the distance associated with each arc is the energy weight on that arc. Unlike G_s there may be cycles in G_a but we will set $c_{i.0.t}^{i.1.t} = 0$ so that G_e has no positive cost cycles so we can again solve the longest path problem as the shortest path problem shown in 4.6.1 [**1**]. We will again assume that there is a positive probability the UUV reaches the destination node by time step T which implies 4.6.1 has a finite optimal solution.

Converting shortest path problem 4.6.1 on graph G_e to a constrained shortest path problem on graph G_a we get 4.6.11:

$$
\begin{aligned}
&(4.6.11a) && \min -\mathbf{e}^{'}\mathbf{v} \\
&(4.6.11b) && \text{s.t.} \quad \mathbf{B}_a \mathbf{v} = \mathbf{d}_e \\
&(4.6.11c) && v_{i.k.t}^{j.l.s} \leq M * \Pr\{V_{i.k.t}^{j.l.s}\} \ \forall (i.k.t, j.l.s) \in A_s \\
&(4.6.11d) && \mathbf{v} \geq 0
\end{aligned}
$$

Instead of using the values $\Pr\{V_{i.k.t}^{j.l.s}\}$ to constrain the shortest path variable $\mathbf{v}$ in problem 4.6.3 we can alternatively use $\mathbf{y}$ and $\mathbf{z}$. Consider 4.6.12 which we will prove is equivalent to 4.6.11.

$$
\begin{aligned}
&(4.6.12a) && \min -\mathbf{e}^{'}\mathbf{v} \\
&(4.6.12b) && \text{s.t.} \quad \mathbf{B}_s \mathbf{v} = \mathbf{d}_e \\
&(4.6.12c) && v_{i.k.t}^{j.l.s} \leq 2 * y_{i.k}^{j.l} * (1 - z_{i.k.t}) \\
& && \qquad \forall (i.k.t, j.l.s) \in A_s \quad \text{such that } i.k, j.l \notin \{S.0, D.0, D.1\} \\
&(4.6.12d) && v_{i.k.t}^{D.0.s} \leq 2 * z_{i.k.t} \qquad \forall (i.k.t, D.0.s) \in A_s \\
&(4.6.12e) && v_{i.0.t}^{i.1.t} \leq 2 * z_{i.k.t} \qquad \forall (i.0.t, i.1.t) \in A_a \\
&(4.6.12f) && \mathbf{v} \geq 0
\end{aligned}
$$

TABLE 3. Energy Formulation Logic Table

	$(i.k.t, j.l.s) \in A_s$ such that $i.k, j.l \notin \{S.0, D.0, D.1\}$				$(i.k.t, j.l.s) \in A_a$ such that $k = 0$ and $s = t$ or $j.l = D.0$		
$z_{i.k.t}$	$= 0$			$= 1$	$= 0$	$= 1$	
$y_{i.k}^{j.l}$	$= 0$	$= 1$		Free	Free	Free	
$\Pr\{V_{i.k.t}^{j.l.s}\}$	$= 0$	$= 0$	> 0	$= 0$	$= 0$	$= 0$	> 0
$v_{i.k.t}^{j.l.s}$ in problem 4.6.11	$= 0$	$= 0$	≥ 0	$= 0$	$= 0$	$= 0$	≥ 0
$v_{i.k.t}^{j.l.s}$ in problem 4.6.12	$= 0$	≥ 0	≥ 0	$= 0$	$= 0$	≥ 0	≥ 0

Table 3 depicts the connection between $\mathbf{y}$, $\mathbf{z}$, $\Pr\{V_{i.k.t}^{j.l.s}\}$, and $\mathbf{v}$ in problems 4.6.11 and 4.6.12. In words table 3 has the following meaning,

Case 1 $(i.k.t, j.l.s) \in A_s$ such that $i.k, j.l \notin \{S.0, D.0, D.1\}$.
  Case 1.1 $z_{i.k.t} = 0$.
    Case 1.1.1 $y_{i.k}^{j.l} = 0$. Then $v_{i.k.t}^{j.l.s} = 0$ in both problems 4.6.11 and 4.6.12.
    Case 1.1.2 $y_{i.k}^{j.l} = 1$.
      Case 1.1.2.1 $\Pr\{V_{i.k.t}^{j.l.s}\} = 0$. Then $v_{i.k.t}^{j.l.s} = 0$ in problem 4.6.11 and $0 \leq v_{i.k.t}^{j.l.s} \leq 1$ in problem 4.6.12.
      Case 1.1.2.2 $\Pr\{V_{i.k.t}^{j.l.s}\} > 0$. Then $v_{i.k.t}^{j.l.s} \geq 0$ in both problems 4.6.11 and 4.6.12.
  Case 1.2 $z_{i.k.t} = 1$. Then $v_{i.k.t}^{j.l.s} = 0$ in both problems 4.6.11 and 4.6.12.
Case 2 $(i.k.t, j.l.s) \in A_a$ such that $k = 0$ and $s = t$ or $j.l = D.0$.
  Case 2.1 $z_{i.k.t} = 0$. Then $v_{i.k.t}^{j.l.s} = 0$ in both problems 4.6.11 and 4.6.12.
  Case 2.2 $z_{i.k.t} = 1$.
    Case 2.2.1 $\Pr\{V_{i.k.t}^{j.l.s}\} = 0$. Then $v_{i.k.t}^{j.l.s} = 0$ in problem 4.6.11 and $v_{i.k.t}^{j.l.s} \geq 0$ in problem 4.6.12.
    Case 2.2.2 $\Pr\{V_{i.k.t}^{j.l.s}\} > 0$. Then $v_{i.k.t}^{j.l.s} \geq 0$ in both problems 4.6.11 and 4.6.12.

This means that the only time 4.6.11 could give a different solution than 4.6.12 is if $v_{i.k.t}^{j.l.s} > 0$ when $z_{i.k.t} = 0$, $y_{i.k}^{j.l} = 1$ and $\Pr\{V_{i.k.t}^{j.l.s}\} = 0$ or $v_{i.k.t}^{j.l.s} > 0$ when $z_{i.k.t} = 1$ and $\Pr\{V_{i.k.t}^{j.l.s}\} = 0$. The following lemma proves these events can never occur. Similar to Lemma 4.18, Lemma 4.20 uses the fact that a feasible $\mathbf{v}$ to 4.6.12 must correspond to a path from node $S.0.0$ to node $D.1.T$ and the only arcs $(i.k.t, j.l.s)$, such that $v_{i.k.t}^{j.l.s} > 0$ are arcs where either $p_{i.k.t,j.l.s} > 0$ or $z_{i.k.t} = 1$.

Lemma 4.20. *For a feasible route $\boldsymbol{y}$ and feasible alteration strategy $\boldsymbol{z}$ to Formulation 4.4.1 that has a positive probability of reaching the destination node $D.1.T$, if $\boldsymbol{v}$ is a feasible solution to 4.6.12 then* $\Pr\{V_{i.k.t}^{j.l.s}\} = 0 \rightarrow v_{i.k.t}^{j.l.s} = 0$.

PROOF. Assume $\mathbf{v}$ is a feasible solution to 4.6.12. The solution $\mathbf{v}$ must correspond to a path P from node $S.0.0$ to node $D.1.T$. Let,

$$P = \{S.0.0 = i_0.k_0.t_0 \rightarrow i_1.k_1.t_1 \rightarrow \ldots \rightarrow i_M.k_M.t_M = D.1.T\}$$

We will prove the contra-positive that $v_{i.k.t}^{j.l.s} > 0 \rightarrow \Pr\{V_{i.k.t}^{j.l.s}\} > 0$. We know that only values of $\mathbf{v}$ which are non zero are those on the path P. So we only need to show that $\Pr\{V_{i_m.k_m.t_m}^{i_{m+1}.k_{m+1}.t_{m+1}}\} > 0$ for all $0 \leq m < M$.

By construction of graph G_a we know that since $(S.0.0, i_1.k_1.t_1) \in G_a$ then $0 < p_{S.0.0}^{i_1.k_1.t_1} = \Pr\{V_{i_0.k_0.t_0}^{i_1.k_1.t_1}\}$.

Now for the induction step assume $\Pr\{V_{i_{m-1}.k_{m-1}.t_{m-1}}^{i_m.k_m.t_m}\} > 0$ for some $1 \leq m < M$. Either $z_{i_m.k_m.t_m} = 0$ or $z_{i_m.k_m.t_m} = 1$.

Case 1 Assume $z_{i_m.k_m.t_m} = 0$. Since $v_{i_m.k_m.t_m}^{i_{m+1}.k_{m+1}.t_{m+1}} = 1$ we know that $y_{i_m.k_m}^{i_{m+1}.k_{m+1}} = 1$, and $(i_m.k_m.t_m, i_{m+1}.k_{m+1}.t_{m+1}) \in A_s$ which means $p_{i_m.k_m.t_m}^{i_{m+1}.k_{m+1}.t_{m+1}} > 0$. We have assumed $\Pr\{V_{i_{m-1}.k_{m-1}.t_{m-1}}^{i_m.k_m.t_m}\} > 0$ which means that $\Pr\{U_{i_m.k_m.t_m}\} > 0$. Therefore $\Pr\{V_{i_m.k_m.t_m}^{i_{m+1}.k_{m+1}.t_{m+1}}\} = p_{i_m.k_m.t_m}^{i_{m+1}.k_{m+1}.t_{m+1}} * y_{i_m.k_m}^{i_{m+1}.k_{m+1}} * \Pr\{U_{i_m.k_m.t_m}\} > 0$.

Case 2 Assume $z_{i_m.k_m.t_m} = 1$. Either $k_m = 0$ or $k_m = 1$.

Case 2.1 Assume $k_m = 0$. Since $v_{i_m.k_m.t_m}^{i_{m+1}.k_{m+1}.t_{m+1}} = 1$ by construction of graph G_a we know $i_{m+1}.k_{m+1}.t_{m+1} = i_m.1.t_m$. Then by definition of the route alteration variable $\mathbf{z}$ when $k_m = 0$, $\Pr\{V_{i_m.k_m.t_m}^{i_{m+1}.k_{m+1}.t_{m+1}}\} = \Pr\{U_{i_m.k_m.t_m}\} \geq \Pr\{V_{i_{m-1}.k_{m-1}.t_{m-1}}^{i_m.k_m.t_m}\} > 0$.

Case 2.2 Assume $k_m = 1$. Since $v_{i_m.k_m.t_m}^{i_{m+1}.k_{m+1}.t_{m+1}} = 1$ by construction of graph G_a we know $i_{m+1}.k_{m+1}.t_{m+1} = D.0.t_{m+1}$ and $p_{i_m.1.t_m}^{D.0.t_{m+1}} > 0$. Then by definition of the route alteration variable $\mathbf{z}$ when $k_m = 1$ and $z_{i_m.k_m.t_m} = 1$, $\Pr\{V_{i_m.k_m.t_m}^{i_{m+1}.k_{m+1}.t_{m+1}}\} = \Pr\{V_{i_m.k_m.t_m}^{D.0.t_{m+1}}\} = p_{i_m.1.t_m}^{D.0.t_{m+1}} * \Pr\{U_{i_m.1.t_m}\} \geq p_{i_m.1.t_m}^{D.0.t_{m+1}} * \Pr\{V_{i_{m-1}.k_{m-1}.t_{m-1}}^{i_m.k_m.t_m}\} > 0$.

So in all cases we have shown that $\Pr\{V_{i_m.k_m.t_m}^{i_{m+1}.k_{m+1}.t_{m+1}}\} > 0$.

By induction we have established that $\Pr\{V_{i_m.k_m.t_m}^{i_{m+1}.k_{m+1}.t_{m+1}}\} > 0$ for all $0 \leq m < M$. So we have proven the result by contra-positive. $\square$

Lemma 4.20 shows us that we can replace Table 3 with Table 4, which means that 4.6.11 and 4.6.12 are equivalent.

TABLE 4. Energy Formulation Logic Table

	$(i.k.t, j.l.s) \in A_s$ such that $i.k, j.l \notin \{S.0, D.0, D.1\}$				$(i.k.t, j.l.s) \in A_a$ such that $k = 0$ and $s = t$ or $j.l = D.0$		
$z_{i.k.t}$	$= 0$			$= 1$	$= 0$	$= 1$	
$y_{i.k}^{j.l}$	$= 0$	$= 1$		Free	Free	Free	
$\Pr\{V_{i.k.t}^{j.l.s}\}$	$= 0$	$= 0$	> 0	$= 0$	$= 0$	$= 0$	> 0
$v_{i.k.t}^{j.l.s}$ in problem 4.6.11	$= 0$	$= 0$	≥ 0	$= 0$	$= 0$	$= 0$	≥ 0
$v_{i.k.t}^{j.l.s}$ in problem 4.6.12	$= 0$	$= 0$	≥ 0	$= 0$	$= 0$	$= 0$	≥ 0

Like we did for the energy constraint for Formulation 4.3.7 we want to place a constraint which says that the optimal solution to 4.6.12 is less than or equal to the total amount of energy the UUV has available. To do this we need to take the dual of 4.6.12 which is shown in problem 4.6.13:

$$\max \mathbf{d}_e^{'}\mathbf{u} + \sum_{(i.k.t,j.l.s)\in A_s | i.k,j.l \notin \{S.0,D.0,D.1\}} 2 * y_{i.k}^{j.l} * (1 - z_{i.k.t}) * w_{i.k.t}^{j.l.s}$$

$$+ \sum_{(i.k.t,D.0.s)\in A_s} 2 * z_{i.k.t} * w_{i.k.t}^{D.0.s} + \sum_{(i.0.t,i.1.t)\in A_a} 2 * z_{i.0.t} * w_{i.0.t}^{i.1.t} \tag{4.6.13a}$$

(4.6.13b)

$$\text{s.t.} \quad \begin{bmatrix}\mathbf{B}_e^{'} & \mathbf{I}\end{bmatrix}\begin{bmatrix}\mathbf{u}\\ \mathbf{w}\end{bmatrix} \leq -\mathbf{e}$$

$$w_{S.0.0}^{j.l.s} = 0 \quad \forall\ (S.0.0, j.l.s) \in A_s \tag{4.6.13c}$$

$$w_{j.l.s}^{D.1.T} = 0 \quad \forall\ (j.l.s, D.1.T) \in A_s \tag{4.6.13d}$$

$$\mathbf{u} \text{ free } \mathbf{w} \leq 0 \tag{4.6.13e}$$

If we were to strictly take the dual of 4.6.12 the variables of type $w_{S.0.0}^{j.l.s}$ and $w_{j.l.s}^{D.1.T}$ would not be in the problem but for simplicity in notation we include those variables in 4.6.5 and set them to zero.

Similar to how we established the energy constraint for Formulation 4.3.7 we can use complimentary slackness of $\mathbf{w}$ to conclude that 4.6.13 is equivalent to problem 4.6.14:

$$
\begin{array}{lll}
(4.6.14a)\ \max & \mathbf{d}_e^{'}\mathbf{u} & \\
(4.6.14b)\text{s.t.} & \begin{bmatrix}\mathbf{B}_e^{'} & \mathbf{I}\end{bmatrix}\begin{bmatrix}\mathbf{u}\\ \mathbf{w}\end{bmatrix} \leq -\mathbf{e} & \\
(4.6.14c) & w_{S.0.0}^{j.l.s} = 0 & \forall\ (S.0.0, j.l.s) \in A_s \\
(4.6.14d) & w_{j.l.s}^{D.1.T} = 0 & \forall\ (j.l.s, D.1.T) \in A_s \\
(4.6.14e) & w_{i.k.t}^{j.l.s} \geq M * (y_{i.k}^{j.l} - 1 - z_{i.k.t}) & \forall\ (i.k.t, j.l.s) \text{ such that } i.k, j.l \notin \{S.0, D.0, D.1\} \\
(4.6.14f) & w_{i.0.t}^{i.1.t} \geq M * (z_{i.0.t} - 1) & \forall (i.0.t, i.1.t) \in A_a \\
(4.6.14g) & w_{i.k.t}^{D.0.s} \geq M * (z_{i.k.t} - 1) & \forall (i.k.t, D.0.s) \in A_a \\
(4.6.14h) & \mathbf{u} \text{ free } \mathbf{w} \leq 0 &
\end{array}
$$

By multiplying through by negative one, and filling in the actual values of $\mathbf{d}_e$ we get the equivalent problem 4.6.15:

$$
\begin{array}{lll}
(4.6.15a)\ \min & u_{D.1.T} - u_{S.0.0} & \\
(4.6.15b)\text{s.t.} & \begin{bmatrix}\mathbf{B}_e^{'} & \mathbf{I}\end{bmatrix}\begin{bmatrix}-\mathbf{u}\\ \mathbf{w}\end{bmatrix} \geq \mathbf{e} & \\
(4.6.15c) & w_{S.0.0}^{j.l.s} = 0 & \forall\ (S.0.0, j.l.s) \in A_s \\
(4.6.15d) & w_{j.l.s}^{D.1.T} = 0 & \forall\ (j.l.s, D.1.T) \in \mathbb{A}_s \\
(4.6.15e) & w_{i.k.t}^{j.l.s} \leq M * (1 - y_{i.k}^{j.l} + z_{i.k.t}) & \forall\ (i.k.t, j.l.s) \text{ such that } i.k, j.l \notin \{S.0, D.0, D.1\} \\
(4.6.15f) & w_{i.0.t}^{i.1.t} \leq M * (1 - z_{i.0.t}) & \forall (i.0.t, i.1.t) \in A_a \\
(4.6.15g) & w_{i.k.t}^{D.0.s} \leq M * (1 - z_{i.k.t}) & \forall (i.k.t, D.0.s) \in A_a \\
(4.6.15h) & \mathbf{u} \text{ free } \mathbf{w} \geq 0 &
\end{array}
$$

4.6.3. Concluding Energy Constraint Remarks. We have established the energy consumption linear programming problems 4.6.10 and 4.6.15 for Formulations 4.3.7 and 4.4.1 respectively. To add the energy constraint to the either formulation we add the constraint,

$$u_{D.1.T} - u_{S.0.0} \leq E$$

and the constraints of either 4.6.10 or 4.6.15 depending on which formulation we are using. We know that we can add the energy constraint in this way from Lemma A.3.

When using the energy consumption linear programming problems created, the variables $\mathbf{u}$ are the node potentials of the shortest path problem. Similar to other shortest path problems we can arbitrarily choose one node potential to set to zero due to the linear dependance of the node-arc incident matrix [**1**]. For convenience it is easiest to set $u_{S.0.0} = 0$ and the energy constraint just becomes $u_{D.1.T} \leq E$. By doing this it means that the large number M only has to be bigger then the total amount of energy available E. We will use $M = 2 * E$ to make it clear which values of $\mathbf{w}$ are used and which are not.

By setting $u_{S.0.0} = 0$ it makes it so that if there is a positive probability the UUV reaches node $i.k.t$ then $u_{i.k.t}$ is the maximum amount of energy the UUV will consume if it were to reach the node. This means we could add constraints such as $u_{i.k.t} \leq \rho$ if we wanted to limit the maximum amount of energy the UUV can spend to reach a given node.

At the beginning of this section we assumed $e_{i.k.t}^{j.l.s}$ was the expected energy consumption value and we reasoned that the energy consumption random variable would have a small variance. If we were not in a situation where the variance was small we could use $e_{i.k.t}^{j.l.s} = \mu + \gamma\sigma$ where μ is the expected value, σ^2 is the variance, and γ corresponds to the level of confidence we want that the UUV does not break the energy constraint. The larger γ is the more conservative we are about the energy constraint.

This method for constraining the energy limits the maximum energy consumption on a path from node $S.0.0$ to node $D.1.T$ which the UUV has a positive probability of transiting on. We have constructed the energy constraint in a general way which does not use the fact that time and energy consumption values are positively correlated. One of the main concerns for this method of constraining energy consumption is whether or not it is too conservative. To argue that this constraint is not too conservative we will use the fact that energy and time are positively correlated. We are only considering paths from $S.0.0$ to $D.1.T$ which the UUV has a positive probability of transiting on, which means we do not consider the energy consumption on paths where the UUV does not reach the destination node by the end time T. This means that we are not considering cases where a relatively large amount of time is consumed which results in the UUV not reaching the destination node by time T. Since time and energy are positively correlated this means we are also not considering cases where a relatively large amount of energy is consumed. This means we are taking the maximum energy consumption of cases where the time and energy values are not abnormally large. Therefore this method of constraining the energy consumption is not thought to be too conservative.

This method used for adding an energy constraint is rather general. If we wanted to add another resource constraint we could use this same method. We have argued that a positive correlation between time and energy makes it so that this method for the energy constraint is not too conservative. This implies that for a new resource, no correlation with time might make this method too conservative.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 5

Heuristics and Analysis

In this chapter we introduce potential heuristics for the different formulations we have developed. The main focus of this paper is the stochastic time expanded formulation and the route alteration formulation, so we will focus our run time analysis and heuristic performance analysis on these formulations. For the simple formulation we will describe the heuristic proposed in [**16**] for the UUV route planning problem. For the deterministic time expanded formulation we will discuss how to perform the nearest neighbor heuristic in conjunction with time windows, and we will introduce the rounding heuristic. Continuing with the stochastic time expanded model, we will analyze the run time of the exact formulation, look at how much nearest neighbors are used in the exact solution to analyze if the nearest neighbor algorithm would provide solutions similar to the exact solution, analyze the run time of the rounding heuristic, and analyze the performance of the rounding heuristic compared to the exact solution and the solution to the relaxed linear programming problem. For the route alteration formulation we look at the run time of the exact formulation and adapt the rounding heuristic for use with the route alteration formulation.

5.1. The Simple Formulation

The first heuristic we will discuss is a nearest neighbor algorithm. We want to find a "good" path,

$$P = \{S = i_0, i_1, \ldots, i_M = D\}.$$

We have a number of options of how to build a nearest neighbor path. Assume $f(i, j)$ is some measure of distance from i to j. We will later discuss some possibilities for $f(i, j)$. We can build the path from the start node to the destination node, from the destination node to the start node, or some combination of the two which meets in the middle. The algorithm below uses a combination of building the path start to destination, and destination to start.

In this nearest neighbor algorithm we have a set U which represents the current nearest neighbor partial path from the start node, and the set W which represents the current nearest neighbor partial path from the destination node. We choose to append the node which is closest to either U or W. The hope for this algorithm is that the two partial paths will meet in the middle.

Simple Formulation Nearest Neighbor Heuristic
begin
Let $i_1 = S$, $j_1 = D$, and $U = \{i_1\}$ and $W = \{j_1\}$ be ordered sets. Let W' represent the set W in reverse order.
Let $\mathbf{y}(U, W, k)$ correspond to the decision vector $\mathbf{y}$ that represents the path $P = U \bigcup \{k\} \bigcup W'$.
Let $|U|$ and $|W|$ denote the number of nodes in the sets U and W respectively.
while $V = \{k \in N \setminus (U \bigcup W) : \mathbf{t}'\mathbf{y}(U, W, k) \leq T \text{ and } \mathbf{e}'\mathbf{y}(U, W, k) \leq E\} \neq \emptyset$.
Find k and l such that
$f(i_{|U|}, k) = \min_a \{f(i_{|U|}, a) : a \in V\}$ and $f(l, j_{|W|}) = \min_b \{f(b, j_{|W|}) : b \in V\}$;
if $f(i_{|U|}, k) < f(l, j_{|W|})$ **then** append $i_{|U|+1} = k$ to U
else append $j_{|W|+1} = l$ to W.
end
The created path is $P = U \bigcup W'$.
end

We have a number of options as to what to choose $f(i, j)$ to be. Consider the functions below,

$$f(i,j) = \left(\frac{\alpha * (t_{i.1}^{j.0} + t_{j.0}^{j.1})/T + (1-\alpha) * (e_{i.1}^{j.0} + e_{j.0}^{j.1})/E}{r_j} \right)^{\gamma}$$

$$f(i,j) = \left(\frac{\max\{t_{i.1}^{j.0} + t_{j.0}^{j.1})/T, (e_{i.1}^{j.0} + e_{j.0}^{j.1})/E\}}{r_j} \right)^{\gamma}$$

where $0 \leq \alpha \leq 1$ and $\beta > 0$. The value α specifies which resource, time or energy, is most important. A value $\alpha > .5$ says time is the more important factor when choosing a nearest neighbor, and a value of $\alpha < .5$ says energy is the more important factor. The value of γ changes the weight between time and energy resources and the reward for performing a task. We know the numerator of $f(i, j)$ is less than one. So if we have a reward system where $r_i > 1$ then a value of $\gamma > 1$ would favor tasks with higher rewards and $\gamma < 1$ would favor tasks with a low consumption of resources. In the first version of $f(i, j)$ we take a weighted sum of the fraction of time consumption and the fraction of energy consumption and divide this value by the reward for task j. In the second version of $f(i, j)$ we take the maximum of the time consumption fraction and of the energy consumption fraction and divide by the reward for task j. These functions make logical sense because they find a nearest neighbor which uses a small amount of resources and has a higher value for the reward. The art of this this heuristic is in training the values α and γ to provide the best results for the situation at hand.

The second heuristic we will discuss is a cheapest insertion algorithm. We will start with the path from the start node to the destination node and gradually insert nodes to build a path. Assume $g(i, k, j)$ is some measure of the distance from i to k to j. We will later discuss some possibilities for the function g.

Simple Formulation Cheapest Insertion Algorithm
begin
  Let $P = \{S, D\}$.
  Let $P(i, k, j)$ be the path where k is inserted in between i and j, and i and j are consecutive in P.
  Let $\mathbf{y}(P)$ correspond to the decision vector $\mathbf{y}$ that represents the path P.
  Let improve = **true**.
  while improve =**true**
    Find i, j consecutive in P and $k \in N \setminus P$ such that $g(i, k, j)$ is minimal, $\mathbf{t}'\mathbf{y}(P(i, k, j)) \leq T$, and $\mathbf{e}'\mathbf{y}(P(i, k, j)) \leq E$.
    if no such i, j, k exist then improve = **false**
    else insert k in between i and j in P.
  end
end

Consider the some possibilities of g which are similar in nature to our previous function f.

$$g(i,k,j) = \left(\frac{\alpha * (t_{i.1}^{k.0} + t_{k.0}^{k.1} + t_{k.1}^{j.0} - t_{i.1}^{j.0})/T + (1-\alpha) * (e_{i.1}^{k.0} + e_{k.0}^{k.1} + e_{k.1}^{j.0} - e_{i.1}^{j.0})/E}{r_k} \right)^{\gamma}$$

$$g(i,k,j) = \left(\frac{\max\{(t_{i.1}^{k.0} + t_{k.0}^{k.1} + t_{k.1}^{j.0} - t_{i.1}^{j.0})/T, (e_{i.1}^{k.0} + e_{k.0}^{k.1} + e_{k.1}^{j.0} - e_{i.1}^{j.0})/E\}}{r_k} \right)^{\gamma}$$

where $0 \leq \alpha \leq 1$ and $\beta > 0$. The values of α and γ have the same meaning as they did with the nearest neighbor heuristic. Again, we have one function which uses a weighted average of the consumption fractions and another which takes a the max of the consumption fractions. These functions make logical sense because they punish a task for consuming resources but also favor tasks with a higher reward.

The goal of these heuristics is to create "good" solutions fast, so we can avoid the relatively large amount of time it takes to solve a mixed integer programming problem. One of the reasons these heuristics are considered fast is because from step to step there is little change in the values $\mathbf{t}'\mathbf{y}$ and $\mathbf{e}'\mathbf{y}$ where $\mathbf{y}$ represents the current path for the heuristic. In fact the change is small enough so we can recompute these values in $O(1)$ time. If we were to apply these heuristics to a formulation with time windows we have to recompute these values at every step, which is $O(n)$ time. The next section for the deterministic time expanded formulation discusses a start to destination nearest neighbor path, and introduces a rounding heuristic which uses the relaxed linear programming problem.

5.2. The Deterministic Time Expanded Formulation

For the simple formulation it takes $O(1)$ time to compute the change in the amount of time (and energy) a route y uses from step to step in the heuristics we explained. When we introduce time windows if we were to use the same type of heuristics it would take $O(n)$ time to compute the time and energy a route y will consume at each iteration of the heuristic. We can however use a start node to destination node nearest neighbor heuristic as opposed to the nearest neighbor heuristic which uses the nearest neighbor from the destination node and the start node.

Deterministic Time Expanded Nearest Neighbor Heuristic
begin
Let $i_1 = S$ and $U = \{i_1\}$ be an ordered set.
Let $\mathbf{x}(U, k)$ correspond to the solution $\mathbf{x}$ that represents the path $P = U \bigcup \{k, D\}$.
while $V = \{k \in N \setminus (U \bigcup \{D\}) : \mathbf{e}'\mathbf{x} \leq E$ and $\mathbf{A}_d\mathbf{x} = \mathbf{d}_d\} \neq \emptyset$
Find k such that
$f(U, k) = \min\{f(U, l) | l \in V\}$.
Append $i_{|U|+1} = k$ to U.
end
The created path is $P = U \bigcup \{D\}$.
end

Again we assume function $f(U, j)$ is some measure of distance from the last node in the set U to j. Consider the nearest neighbor heuristic below. This start node to destination node nearest neighbor heuristic is built so that the computation to see if $\mathbf{e}'\mathbf{x} \leq E$ and $\mathbf{A}_d\mathbf{x} = \mathbf{d}_d$ is $O(1)$ time. We choose to append the node to U which is closest to the last node added to the set U. The function to use for $f(U, j)$ in the deterministic time expanded formulation is similar to the simple formulation only now it uses the time expanded graph which has time windows to determine the distance. Let $i.1.t$ be the last node the UUV will reach if it follows the partial path U, $j.0.s_1 \in N_d$ be the node which the UUV will reach if it follows the partial path U and then goes to node j, and $(j.0.s_1, j.1.s_2) \in A_d$. Then some options for $f(U, j)$ are below,

$$f(U, j) = \left(\frac{\alpha * (s_2 - t)/T + (1 - \alpha) * (e_{i.1.t}^{j.0.s_1} + e_{j.0.s_1}^{j.1.s_2})/E}{r_j} \right)^\gamma$$

$$f(U, j) = \left(\frac{\max\{(s_2 - t)/T, (e_{i.1.t}^{j.0.s_1} + e_{j.0.s_1}^{j.1.s_2})/E\}}{r_j} \right)^\gamma$$

where $0 \leq \alpha \leq 1$ and $\beta > 0$. These are the same basic measures of distance we proposed for the simple formulation but now we take into account the changing time and energy consumption due to the time windows.

The second heuristic we will discuss we will call the rounding heuristic which takes the solution to the relaxed linear programming problem and finds solutions to the integer programming problem. When we relax problem 3.3.1 so that $0 \leq \mathbf{y} \leq 1$ instead of $\mathbf{y}$ binary (and $0 \leq \mathbf{x} \leq 1$), we can think of $\mathbf{y}$ as a strategy for the UUV to move through graph G. This means that $\mathbf{y}$ has the following meaning,

$$y_{i.k}^{j.l} = \text{ the probability the UUV goes from node } i.k \text{ to node } j.l.$$

This definition of $\mathbf{y}$ would also change the definition of $\mathbf{x}$ to,

$$x_{i.k.t}^{j.l.s} = \text{ the probability the UUV goes from node } i.k.t \text{ to node } j.l.s.$$

So the solution $\mathbf{y}, \mathbf{x}$ to the relaxed version of problem 3.3.1 would suggest that once the UUV reaches a node $i.k.t$ it should go to node $j.l.s$ with probability,

$$\frac{x_{i.k.t}^{j.l.s}}{\sum_{j'.l'.s'|(j'.l'.s',i.k.t)\in A_d} x_{j'.l'.s'}^{i.k.t}} \tag{5.2.1}$$

At first glance this appears to be a reasonable solution to the problem at hand, but this formulation does have unresolved issues.

The first issue comes with the energy constraint. If we were to just use the constraint, $\mathbf{e}'\mathbf{y} \leq E$, and we randomly chose what node to visit next according to a solution $\mathbf{y}, \mathbf{x}$ the UUV may go on a path which actually breaks the energy constraint. To handle the energy constraint we can use the method described in 4.6 but now we must use an M large enough that $My_{i.k}^{j.l} > E$ if $y_{i.k}^{j.l} > 0$. So we can handle the relaxed formulation of 3.3.1 and its energy constraint.

The next issue we cannot resolve deals with cycles in $\mathbf{y}$. If the UUV were to randomly choose a path according to a solution to the relaxed problem $\mathbf{x}$ as described in 5.2.1 it is possible the UUV does a task more than once. To fix this problem we first change the probability of going to a node to the following,

$$\Pr\left\{\begin{matrix}\text{The UUV goes from} \\ \text{node } i.k.t \text{ to node } j.l.s\end{matrix}\right\} = \begin{cases} \dfrac{x_{i.k.t}^{j.l.s}}{\sum_{j'.l'.s'|(i.k.t,j'.l'.s')\in A_d \text{ and } j'.l' \text{ not yet visited}} x_{i.k.t}^{j'.l'.s'}} & \begin{matrix}\text{if } j.l \text{ has not} \\ \text{yet been visted}\end{matrix} \\ 0 \quad \text{otherwise} & \end{cases} \tag{5.2.2}$$

The problem with defining the transition probabilities for the UUV as shown in 5.2.2 is that there could be no $j.l$ which have not yet been visited such that $x_{i.k.t}^{j.l.s} > 0$. This would be a serious problem because then the solution $\mathbf{x}$ to the relaxed problem 3.3.1 would provide no guidance on where the UUV should go next. This problem is caused by the fact that there could be a positive flow on a cycle in $\mathbf{y}$. Since $\mathbf{y}$ is no longer binary we cannot eliminate these cycles as we did in Chapter 2. This inability to eliminate cycles makes it so that we cannot use a solution $\mathbf{y}$ and $\mathbf{x}$ to problem relaxed 3.3.1 as an actual strategy for the UUV, but we can use the solution to find "good" solutions with the rounding heuristic.

Deterministic Time Expanded ϵ-Rounding Heuristic
begin
Relax problem 3.3.1 so that $0 \leq \mathbf{y} \leq 1$ and $0 \leq \mathbf{x} \leq 1$ and solve the relaxed problem.
Let $U = \{P | P$ is a path from $S.1$ to $D.0$ in $\mathbf{y}$ with flow greater than ϵ, and is a feasible route for the UUV$\}$.
Find $P_{\max} \in U$ such that the reward for P_{max} is greater than or equal to all other paths in U.
$P_{\max}$ is the path returned for the heuristic.
end

The ϵ-Rounding Heuristic looks at the solution $\mathbf{y}$ for the relaxed problem 3.3.1 and finds all paths with flow greater than ϵ and chooses the best feasible path among them. When checking if a path P is feasible we must ensure that the corresponding $\mathbf{y}_P$ and $\mathbf{x}_P$ satisfy the constraints in problem 3.3.1 which takes $O(n)$ time. Clearly the smaller ϵ is higher the number is of paths that must be checked. The hope for this heuristic is that the number of paths necessary to check is relatively small, and the solutions provided are "good". The art of this heuristic will be in choosing the right ϵ. Due to the similarity between the deterministic time expanded formulation and the stochastic time expanded formulation we differ our analysis of the rounding heuristic to the stochastic case, leaving the belief that the heuristic will perform similarly in each case as further work.

Another way to interpret the rounding heuristic is that we take the solution to the relaxed problem 3.3.1 and we eliminate all the arcs in $(i.k, j.l) \in A$ such that $y_{i.k}^{j.l} \leq \epsilon$ and the corresponding arcs $(i.k.t, j.l.s) \in A_d$ and we solve problem 3.3.1 with this smaller graph. The thought is that doing this eliminates enough arcs in A and A_d that we can solve the new problem much faster and the solutions are near optimal in the original problem. One variant of this heuristic would be to eliminate all arcs $(i.k, j.l) \in A$ such that $y_{i.k}^{j.l} \leq \epsilon$ and $j.l \neq D.0$ and eliminate all corresponding arcs in A_d. We leave the analysis of this variant of the rounding heuristic as further work.

5.3. The Stochastic Time Expanded Formulation

5.3.1. Random Generation of Example Problems. The majority of our analysis will be on the stochastic time expanded formulations and proposed heuristics for the problem. First we will describe the test set of problems we used to randomly simulate a problem for the UUV. The first item to address is the map design. The map used for the UUV problems is 10 nautical miles by 10 nautical miles with three time depended no go zones for the UUV. The first time dependent no go zone is supposed to simulate a shipping lane splitting the map in half from the bottom left corner to the top right corner and the UUV cannot pass through this zone from 8 AM to 8 PM. The second time dependent no go zone is supposed to simulate a fishing area as a polygon near the top left corner which the UUV cannot enter between 6 AM and 10 AM. The third time dependent no go zone is supposed to represent a anchorage area as a square near the bottom right corner which the UUV cannot enter between 10 PM and 2 AM. We also add current to the situation which changes over time to simulate the tide. The tide is assumed to have a period of 11 hours 25 minutes. In these problems we use seconds as the standard unit of time, so we simulate the tide with the functions below being used for the current in the north and east directions where a negative value implies a current in the opposite direction,

$$\sin(2\pi t/(11 \times 3600 + 25 \times 60)) \text{ is the current in the East direction for time } t$$
$$.5 * \sin(2\pi t/(11 \times 3600 + 25 \times 60)) \text{ is the current in the North direction for time } t$$

The UUV is given a relative speed of 2 knots but its actual speed over ground at a specific time t is dependent on the current.

Figure 5.1 depicts this setting for the UUV. For an example problem with n tasks, the start node, all n tasks, and the destination node are randomly placed on the map according to a uniform distribution. So far we have not introduced any random variables into the problem design to make the transit times stochastic. To introduce stochasticity into the situation we will use the problem we have designed thus far to calculate the expected value for the UUV's transit time. We will assume the transit time is normally distributed with the mean calculated from the map characteristics we have described and the standard deviation being the mean divided by 10. We have the standard deviation increasing linearly with the mean to create relatively large standard deviations so that the randomness of the transit times is an important factor in the problem.

We have established how we come up with the transit times in our example problems now we must establish the time to complete a task. To do this we will again use the map shown in Figure 5.1. For a specific task i to establish the expected time to complete a task choose two random points (A and B) on the map in Figure 5.1 with a uniform distribution over the map (independent of the actual location of task i) and the transit from A to B will represent the completion of the task at i.

For the stochastic time expanded formulation to build the graph G_s we need to discretize time so we also need to discretize the normal distribution of the transit times to come up with the problem parameters $p_{i.k.t}^{j.l.s}$. Let δt represent the time step we have chosen to use for the stochastic time expanded problem. Let us assume that the UUV is going to depart node $i.k$ at time step t to transit to node $j.l$ and let μ, σ be the mean and standard deviation of the transit time. Let,

$$t_{\min} = \begin{cases} 1 & \text{if } k = 0. \\ 0 & \text{otherwise.} \end{cases}$$
$$s_{\min} = \max\left\{ t_{\min}, \left\lfloor t + \frac{\mu - 2 * \sigma}{\delta t} \right\rfloor \right\}$$
$$s_{\max} = \left\lceil t + \frac{\mu + 2 * \sigma}{\delta t} \right\rceil$$

Let T represent the random variable for the transit time. We assign the values $p_{i.k.t}^{j.l.s}$ in the following way,

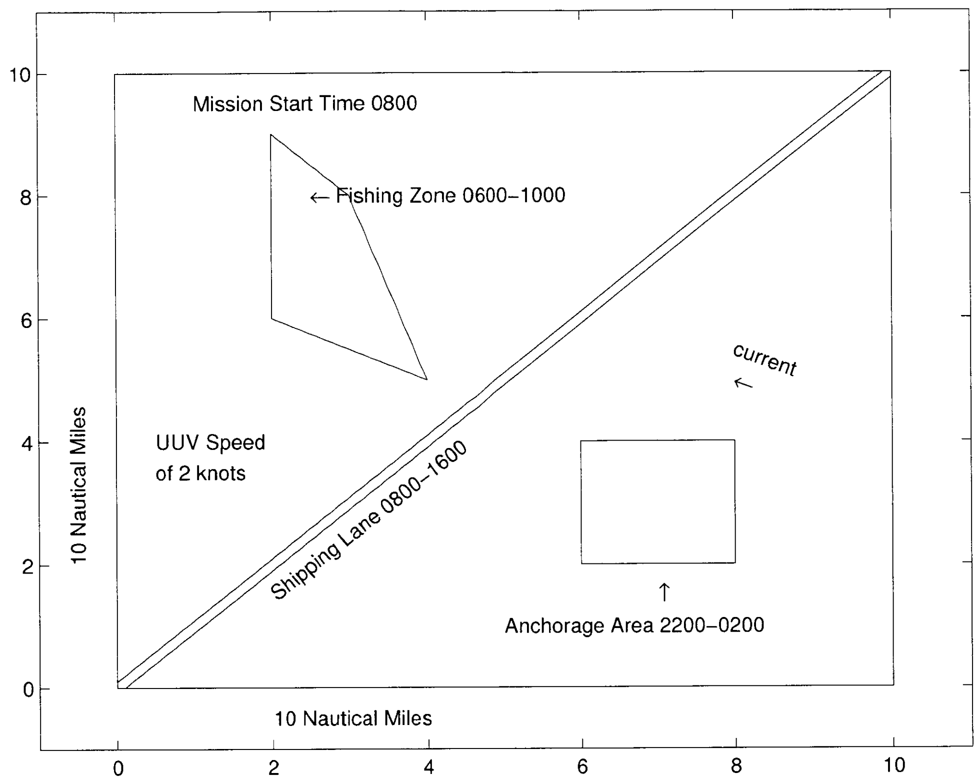


FIGURE 5.1. Map Used To Randomly Generate Example Problems

$$p_{i.k.t}^{j.l.s_{\min}} = \Pr\left\{T \le \mu - 2\sigma + \frac{\delta t}{2}\right\}$$
$$p_{i.k.t}^{j.l.s} = \Pr\left\{\delta t(s-t) - \frac{\delta t}{2} \le T \le \delta t(s-t) + \frac{\delta t}{2}\right\} \text{ for } s_{\min} < s < s_{\max}$$
$$p_{i.k.t}^{j.l.s_{\max}} = 1 - \Pr\left\{T \le \mu + 2\sigma - \frac{\delta t}{2}\right\}$$
$$p_{i.k.t}^{j.l.s} = 0 \text{ for all other } s$$

As an example assume $t = 0$, $\mu = 170/60hr$, and $\sigma = 17/60hr$. Figure 5.2 shows the continuous normal distribution for the transit time as well as the discretized distribution for $\delta t = .5hr$.

We have established a method for randomly creating a problem for the UUV, and for assigning the $\mathbf{p}$ problem parameters to build the graph G_s, but we still need a value for T. We are interested in values of T which force the time constraint to be tight so the UUV must carefully choose which tasks to complete in which order and will usually not be able to complete all tasks. To come up with a value for T we will consider the length of an optimal tour for a traveling salesman problem(TSP). Lets ignore the time dependent no go zones, and the current, and consider solving a TSP on a 10 nm by 10 nm area. Let $L(n, A)$ represent the optimal value of a TSP with n points randomly distributed over an area A with a uniform distribution. Using a Euclidean measure for distance we know from [**14**] that,

$$\frac{L(n,A)}{\sqrt{n}} \quad \overrightarrow{n \to \infty} \quad K\sqrt{A} \text{ with probability 1}$$

where K is a constant that has been approximated to .765 [**14**]. For $A = 100$ in our 10 nm by 10 nm square we will use this fact to approximate $L(n, A)$ with the value $.765 \times 10 \times \sqrt{n}$ (nm). The UUV is traveling at a

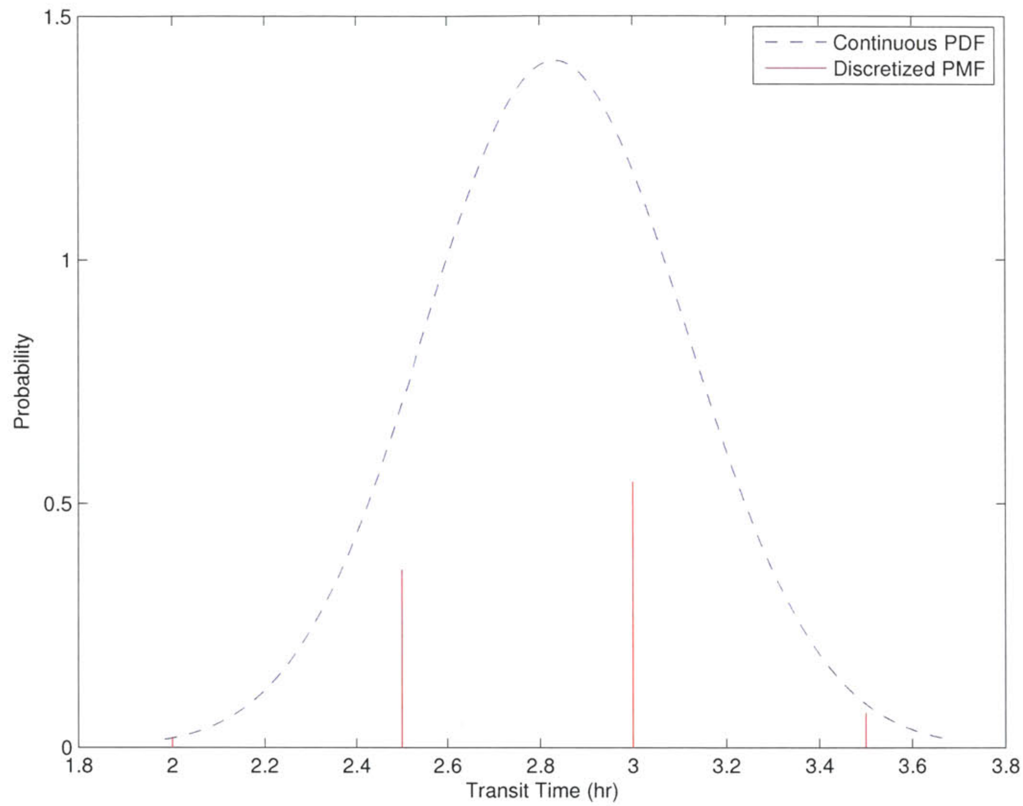


FIGURE 5.2. Example Discretization of Transit Time

speed of 2 knots so we can estimate $L(n, A)$ with the value $.765 \times 5 \times 3600 \times \sqrt(n)$ (sec). Switching from the TSP to our UUV problem the value $L(n, A)$ could be used as a value to estimate the time used to transit from task to task if all tasks were completed in a $n-1$ task problem. This value does not take into account the amount of time which will be needed to complete tasks. Since we are interested in problems where the UUV cannot likely complete all tasks we will ignore an estimation of the time it takes to complete a task and use $5 \times 3600 \times \sqrt{n}$ (sec) as the total amount of time available. Which means that,

$$T = \frac{5 \times 3600 \times \sqrt{n}}{\delta t}$$

where δt is the chosen step size.

For the reward for each task we randomly choose a real valued number between 0 and 10. The last parameter needed for randomly generating example problems is we set $\beta = .9$. We leave the analysis on the effects of changing the β value as further work.

5.3.2. Run Time Analysis for Optimal Solutions. We have set the stage for how we randomly create instances of problems to solve. The only parameter left undefined is δt which we will usually set to be 1800 (sec) but will do some analysis on how the value δt effects the run time for finding solutions.

The first analysis done is on the run time to find solutions to randomly generated example problems. We solved example problems with five different combinations of the total time and the time step as described below for n task problems,

- Total time $= 5 \times 3600 \times \sqrt{n}$ increases with the square root of n, and $\delta t = 1800$ remains constant.
- Total time $= 5 \times 3600 \times \sqrt{5}$ remains constant, and $\delta t = 1800$ remains constant.

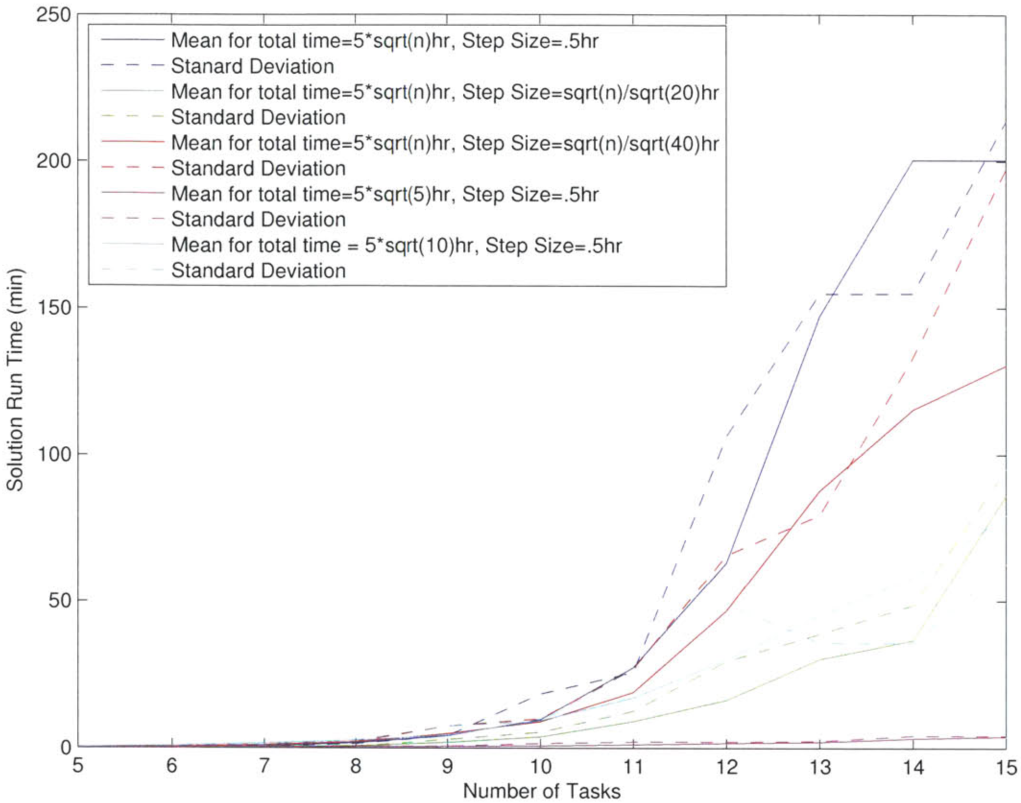


FIGURE 5.3. Stochastic Time Expanded Formulation Run Time

- Total time $= 5 \times 3600 \times \sqrt{n}$ increases with the square root of n, and $\delta t = \sqrt{n} \times 1800/\sqrt{10}$ increases with the square root of n to keep the ratio total time$/\delta t = 10 \times \sqrt{10}$ constant.
- Total time $= 5 \times 3600 \times \sqrt{10}$ remains constant, and $\delta t = 1800$ remains constant.
- Total time $= 5 \times 3600 \times \sqrt{n}$ increases with the square root of n, and $\delta t = \sqrt{n} \times 1800/\sqrt{5}$ increases with the square root of n to keep the ratio total time$/\delta t = 10 \times \sqrt{5}$ constant.

For each one of these combinations of the total time and the step size δt, one hundred randomly generated example problems were solved for different sizes of problems. The mean and standard deviation of the time it takes to solve the exact formulation is shown in Figure 5.3.

We see in Figure 5.3 that the slowest solution time is when the total time increases as δt remains constant. When the ratio between the total time and δt is held constant the resolution of the graph G_s decreases but we see faster solving times. When the total time and δt are held constant we see much faster solution times. This run time analysis helps us draw the following conclusions,

- As expected the more tasks n the UUV has the option of completing, the slower the solution time.
- The ratio of total time$/\delta t$ effects the solution time, seemingly more than the number of tasks n.
- The total time the UUV has available for a mission seems to have the greatest effect on the solution time.

The reason the ratio total time$/\delta t$ has such an impact on the solution time is due to the fact that as this ratio increases linearly the size of the vector $\mathbf{x}$ grows quadratically. This means that if we want to add complexity to the problem to encompass another problem parameter we should not consider adding another dimension to the graph G_s because this would again increase the number of $\mathbf{x}$ variables at least quadratically and ultimately cause the solution speed to become catastrophically slow. For example, when adding the energy constraint we could have discretized energy and had nodes of type $i.k.t.e$ where e is the

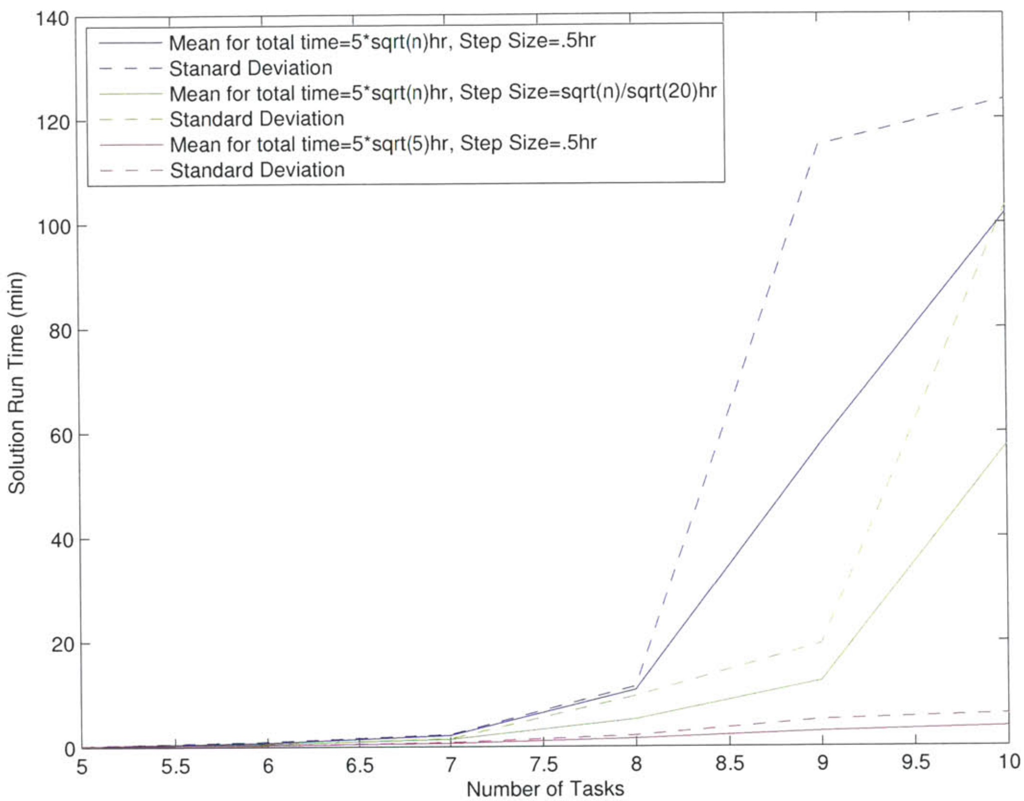


FIGURE 5.4. Stochastic Time Expanded Formulation With Energy Constraint Run Time

energy level but we were able to add the energy constraint without adding this extra dimension. The solution speed analysis thus far has been without considering the energy constraint. Now we will add the energy constraint to the formulation and again test the solution speed. To add the energy constraint we must first describe how we generate the values E and $e_{i.k.t}^{j.l.s}$ for a randomly generated example problem. We will use a random variable $H_{i.k}^{j.l}$ which is uniformly distributed between 1/2 and 3/2 for each arc in A and independent of each other and all other problem parameters. Then we assign the energy parameters in the following way,

- $E = T$.
- $e_{i.k.t}^{j.l.s} = (s-t)H_{i.k}^{j.l}$ for all $(i.k.t, j.l.s) \in A_s$.

Repeating the solution time analysis we get Figure 5.4 which shows us that adding the energy constraint to the formulation does slow down the run time. From this analysis we draw the conclusion that if we were to add another characteristic for the UUV without adding another dimension to the graph G_s the solution speed would be effected similarly to how it was for the energy constraint. We discuss adding another problem parameter in the further research section but leave the analysis on the effect adding the parameter has on the solution speed as further work.

5.3.3. Nearest Neighbor Heuristic. For a nearest neighbor heuristic for the stochastic problem we can use the same heuristic as in the deterministic problem but we must change the distance function $f(U, j)$ so that it handles stochastic parameters. The most obvious way to change the $f(U, j)$ function is to use expected values. If we know the UUV is going to go from node $i.k$ to node $j.l$ then we can estimate the expected time to transit on the arc $(i.k, j.l)$ with the function below,

$$E\begin{bmatrix}\text{The expected time to}\\ \text{transit from } i.k \text{ to } j.l\\ \end{bmatrix} = \begin{bmatrix} \displaystyle\sum_{(i.k.t,j.l.s)\in A_s}\left(x^{j.l.s}_{i.k.t} * p^{j.l.s}_{i.k.t} * \left(\frac{\sum_{(j'.l'.s',i.k.t)\in A_s} x^{i.k.t}_{j'.l'.s'}}{\sum_{(j'.l'.s',i.k.t')\in A_s} x^{i.k.t'}_{j'.l'.s'}}\right) * (s-t)\right)\\ \displaystyle +T * \left(\sum_{(j'.l'.s',i.k.t)\in A_s} x^{i.k.t}_{j'.l'.s'} - \sum_{(i.k.t,j.l.s)\in A_s} x^{j.l.s}_{i.k.t}\right)\end{bmatrix}$$

The first term is the probability the UUV goes on the arc $(i.k.t, j.ls)$ given that the UUV is at node $i.k$ before time step T, times the time the UUV uses on the arc, which is $(s-t)$. The second term is a penalty for the expected transit time if the UUV arrives at node $j.l$ after time step T. We have chosen T to be the penalty coefficient on the probability the UUV does not reach node $j.l$ by time step T but it may be useful to use another term if T is too harsh or lenient. This is the best way to calculate the expected transit time from $(i.k, j.l)$ if we restrict ourselves to only using the stochastic graph G_s, but in most situations it will possible to use the original continuous distribution for the amount of time it will take to make the transit from $i.k$ to $j.l$ for a given expected departure time. For example in the method we are using the randomly generate example problems we specifically calculate the expected transit time and use this value to create the graph G_s. To calculate the expected energy consumption we can use the same function as above but just replace $(s-t)$ with $e^{j.l.s}_{i.k.t}$ and T with E. Once we have established a way to calculate the expected time for a transit and expected energy consumption for a transit we can use the same function for $f(U,j)$ as we had with the determinist problem but replace $(s_2 - t)$ with the expected transit time and $(e^{j.0.s_1}_{i.1.t} + e^{j.1.s_2}_{j.0.s_1})$ with the expected energy consumption.

For our analysis on the using a nearest neighbor heuristic we will not include the energy constraint and focus on the nearest neighbors according to transit time. For all of the randomly generated example problems we will take the total time to be $5 \times 3600 \times \sqrt{n}$ and $\delta t = 1800$. We leave the actual implementation of a nearest neighbor heuristic and its performance as future work but we will look at whether a nearest neighbor heuristic would work probabilistically. We will solve randomly generated example problems and look at the percent the UUV goes from the start node to the task which is its nearest neighbor for different distance metrics. For a problem with n tasks we will say that a specific task has an a priori probability of $1/n$ of being chosen to be performed first. We will compare this probability to the actual fraction of time the nearest neighbor is chosen first. We will look at four different metrics for the distance $(d(i))$ to a node i from the start node S,

$$d(i) = \frac{(\text{The expected time to transit to } i \text{ and complete the task at } i)^2}{r_i}$$
$$d(i) = \frac{\text{The expected time to transit to } i \text{ and complete the task at } i}{r_i}$$
$$d(i) = \text{The expected time to transit to } i \text{ and complete the task at } i$$
$$d(i) = \frac{1}{r_i}$$

We solved one thousand problems for each example problem of each size $n = 3, 4, \ldots, 14$. Figure 5.5 plots the fraction of times the nearest neighbor to the start node is completed first in the optimal solution vs. the a priori probability $1/n$ of choosing a node to be performed first for each of the distance metrics $d(i)$. We see similar results for $Time^2/reward$ and $Time/reward$ while both of these give higher rates than the other two distance metrics. Figure 5.5 shows us that the rate the nearest neighbor is chosen to be performed first is statistically significant, and it is extremely unlikely that in the one thousand random example problems for each n the fraction the nearest neighbor is chosen first is actually due solely to chance. This suggests that a nearest neighbor heuristic may perform similarly to solving the problem exactly.

We can repeat the analysis done thus far for the second nearest neighbor as well. Figure 5.6 plots the fraction of time the second nearest neighbor to the start node is completed first in the optimal solution vs. the a priori probability $1/n$ of choosing a node to be performed first for each of the distance metrics $d(i)$. Again $Time^2/reward$ and $Time/reward$ have the highest fraction and perform similarly. This suggests that

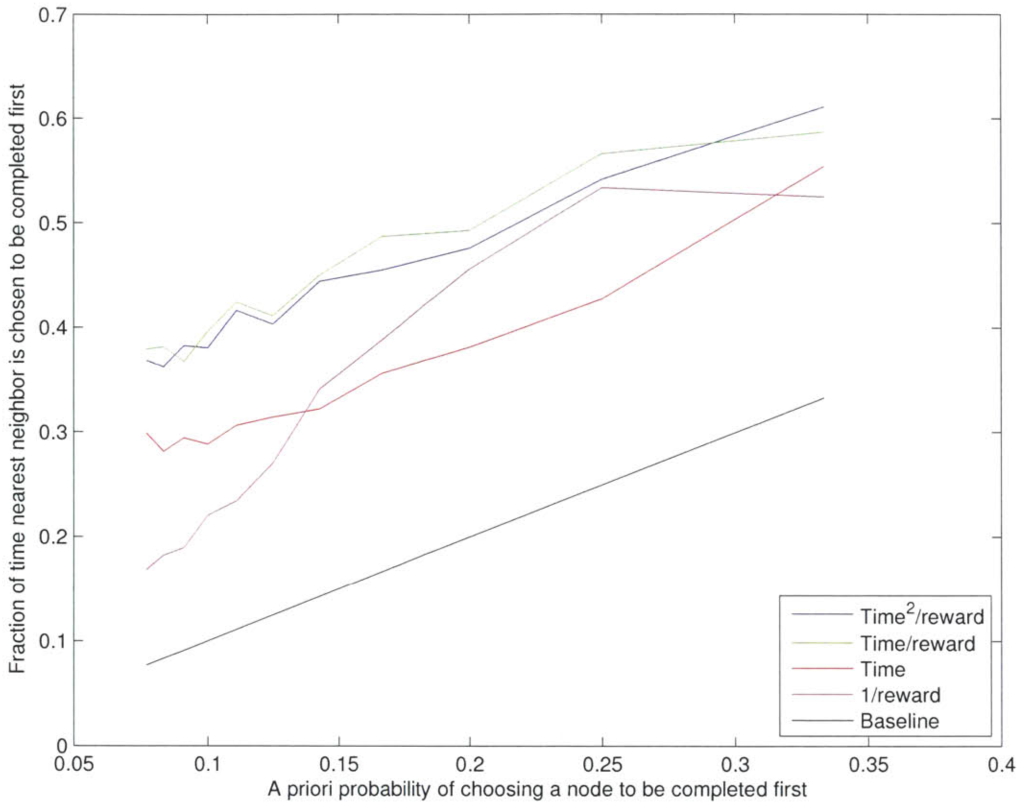


FIGURE 5.5. Statistical Significance that Nearest Neighbor is the Next Task Performed

we could use an alternative heuristic which uses random variables to create a route. We would say that the probability the nearest neighbor is chosen to be performed next equals the fraction of time the nearest neighbor is chosen first in these randomly generate problems, the probability the second nearest neighbor is chosen to be performed next equals the fraction of time the second nearest neighbor is chosen first in these randomly generated problems, and we continue in this fashion to define a probability of being chosen for all of the remaining tasks. The hope of this type of heuristic is that the majority of the time the nearest neighbor is chosen to be performed first in optimal solutions but it allows routes to be generated which do not always choose the nearest neighbor and that the more this randomization is repeated to come up with routes the greater the likelihood is that the optimal route is found.

For a problem with n tasks with an optimal solution which completes m of those tasks the a priori probability a specific task is performed on the optimal solution is thought to be m/n. Figure 5.7 plots the fraction of time the nearest neighbor to the start node is performed (but not necessarily completed first) vs. the a priori probability of m/n a task is performed in the optimal route for the distance metric $Time/reward$. This suggests that even though the nearest neighbor is not always performed first it does have a statistically significant probability of being completed on the optimal path. This suggests that a nearest neighbor heuristic coupled with a route improvement heuristic such as r-Opt [**15**] may perform well since the nearest neighbor is performed so often on the optimal path.

For all of this analysis we only considered the nearest neighbor to the start node due to the belief that once a task i is chosen to be completed we can just skip to node i and consider it as the new start node with the remaining uncompleted tasks available for completion. There are some flaws in this thinking because the UUV is likely to work towards the destination node, but we leave the analysis of this assumption as future work.

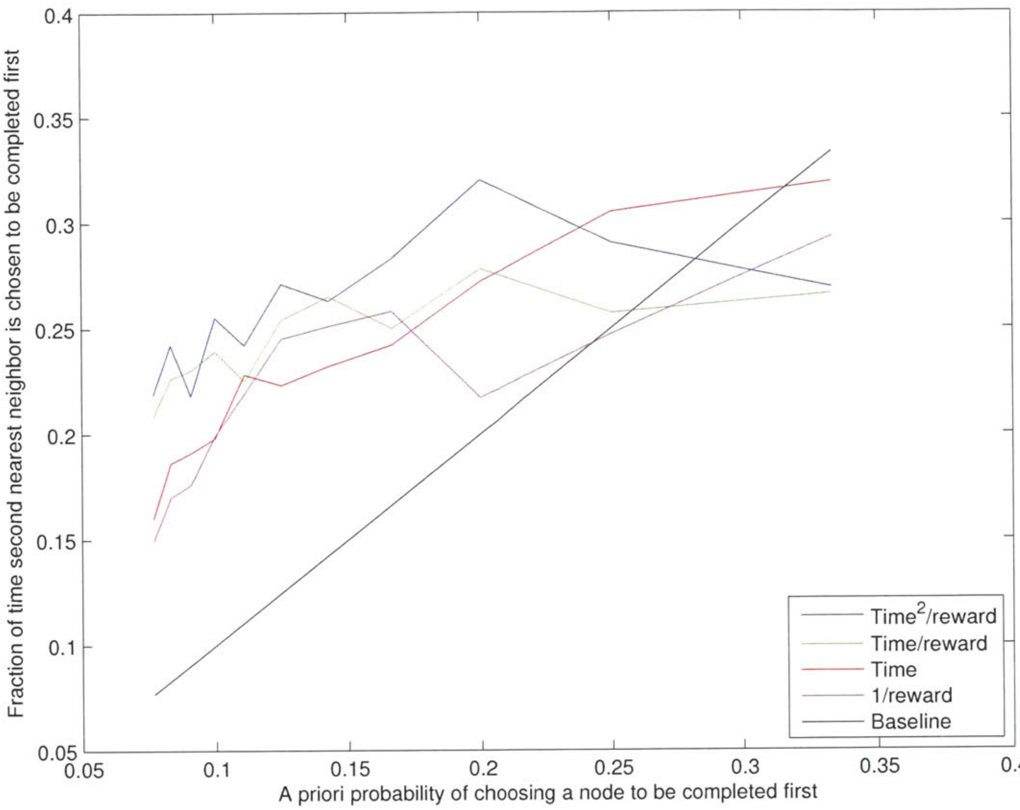


FIGURE 5.6. Statistical Significance that Second Nearest Neighbor is the Next Task Performed

The analysis we have done for nearest neighbor heuristics looks at whether or not they might be useful, but we leave the analysis of their actual performance as future work.

5.3.4. Rounding Heuristic. The rounding heuristic can be used for the stochastic problem just as it can be used in the deterministic problem. However, for the stochastic problem there are constraints which we can add to the relaxed formulation of 4.3.7 to get "better" solutions. The constraints below would be redundant or unneeded constraints in Formulation 4.3.7, but when they are added to the relaxed formulation they have an important impact on the quality of the solution for the relaxed problem.

$$x_{i.k.t}^{j.l.s} \leq p_{i.k.t}^{j.l.s} y_{i.k}^{j.l} \quad \forall (i.k.t, j.l.s) \in A_s | (i.k, j.l) \in A \tag{5.3.1}$$

$$\sum_{t,s|(i.k.t,j.l.s)\in A_s} x_{i.k.t}^{j.l.s} \geq \beta y_{i.k}^{j.l} \quad \forall (i.k, j.l) \in A \tag{5.3.2}$$

$$x_{i.k.t}^{j.l.s_1} p_{i.k.t}^{j.l.s_2} = x_{i.k.t}^{j.l.s_2} p_{i.k.t}^{j.l.s_1} \quad \forall (i.k.t, j.l.s_1), (i.k.t, j.l.s_2) \in A_s \tag{5.3.3}$$

The goal of constraints 5.3.1, 5.3.2, and 5.3.3 is to push the possible values of $\mathbf{y}$ for a relaxed problem closer to being binary. Constraint 5.3.1 is added to drive down the value of an $\mathbf{x}$ arc when its corresponding $\mathbf{y}$ arc has a value less than one. For larger values of $\mathbf{y}$ on an arc constraint 5.3.2 is designed to push the value of the sum of the flow across the corresponding $\mathbf{x}$ arcs up. Constraint 5.3.3 is a constraint which may be beneficial in the mixed integer formulation as well as the relaxed formulation of 4.3.7 and says that the ratio between the $\mathbf{p}$ values for arcs leaving a node in G_s must be equal to the ratio between the corresponding $\mathbf{x}$ values if the $\mathbf{x}$ values are non-zero, and otherwise they must both be zero. Constraint 5.3.3 is important because it makes it so that the relaxed problem solution cannot just send flow along arcs which represent a transit time random variable realizing a small value, but it must send flow along all the arcs.

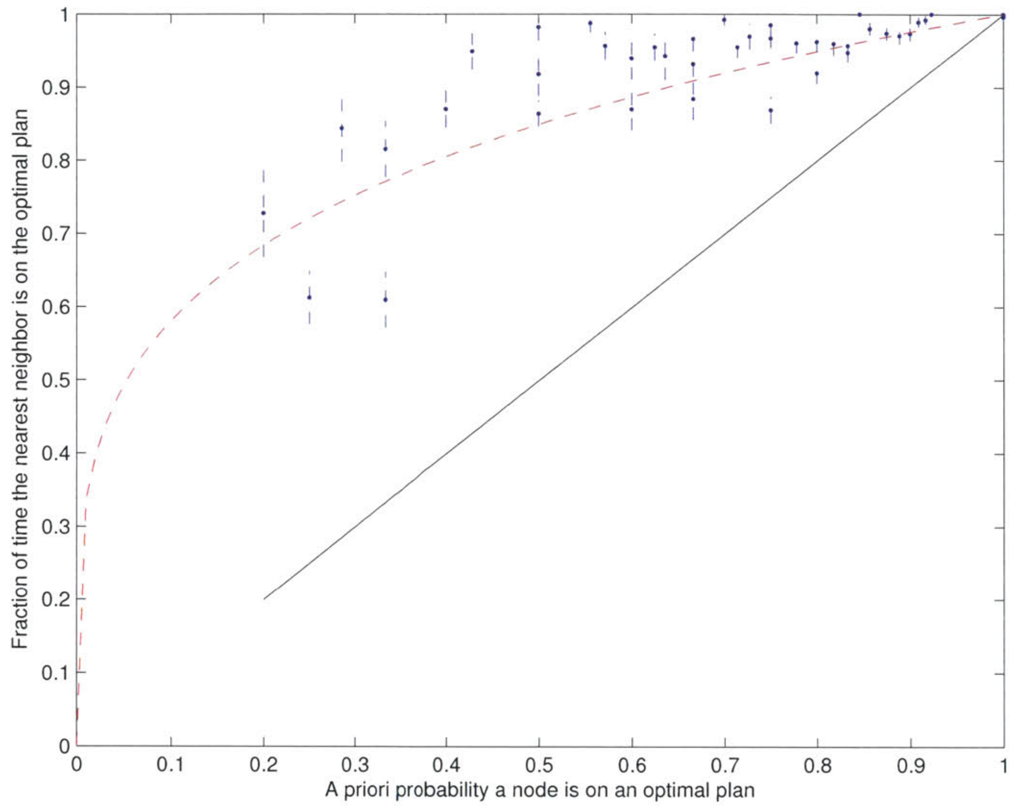


FIGURE 5.7. Statistical Significance that Nearest Neighbor is Performed on an Optimal Path

We will analyze the performance and efficiency of the rounding heuristic. For all of the randomly generated example problems we will take the total time to be $5 \times 3600 \times \sqrt{n}$ and $\delta t = 1800$.

Obviously when using the rounding heuristic the number of solutions checked and the performance of the rounding heuristic depend on the value of ϵ. Figure 5.8 is a plot which shows the average and standard deviation of the number of solutions checked for different values of ϵ. If we were to conduct a brute force search over all permutations for the optimal solution the number of solutions necessary to check for an n task problem would be,

$$\sum_{m=0}^{n} \frac{n!}{m!}$$

This means that for problems of size 10 and for $\epsilon = .000001$ we check on average 100 solutions as opposed to $9,864,101$ in the brute force search and then only 20 solutions are checked for when $\epsilon = .01$.

There is no arguing that the rounding heuristic looks at only a miniscule fraction of permutations compared to a brute force search, but does it find solutions which are relatively "good"? Figure 5.9 looks at the quality performance of the rounding heuristic for different values of ϵ. We compare the best solution found in the rounding heuristic for different values of ϵ to the true optimal value for one hundred randomly generate problems of each size n. We see from Figure 5.9 that the epsilon rounding heuristic performs quite well with mean optimality ratios staying above .9 for all tested values of ϵ. The lower group of lines shown in Figure 5.9 is the ratio between the objective function value found using the rounding heuristic and the objective function value found in the relaxed problem 4.3.7. The gap in between the upper group of lines and the lower group of lines in due to the duality gap [**9**]. As seen in Figure 5.9 we continued using the rounding heuristic for problems too large to be solved exactly with Formulation 4.3.7 but for these larger values of n the ratio between the rounding heuristic and relaxed problem decreases slightly but stays above .8 which suggests that the rounding heuristic slowly decreases its performance. This is because as n increases more

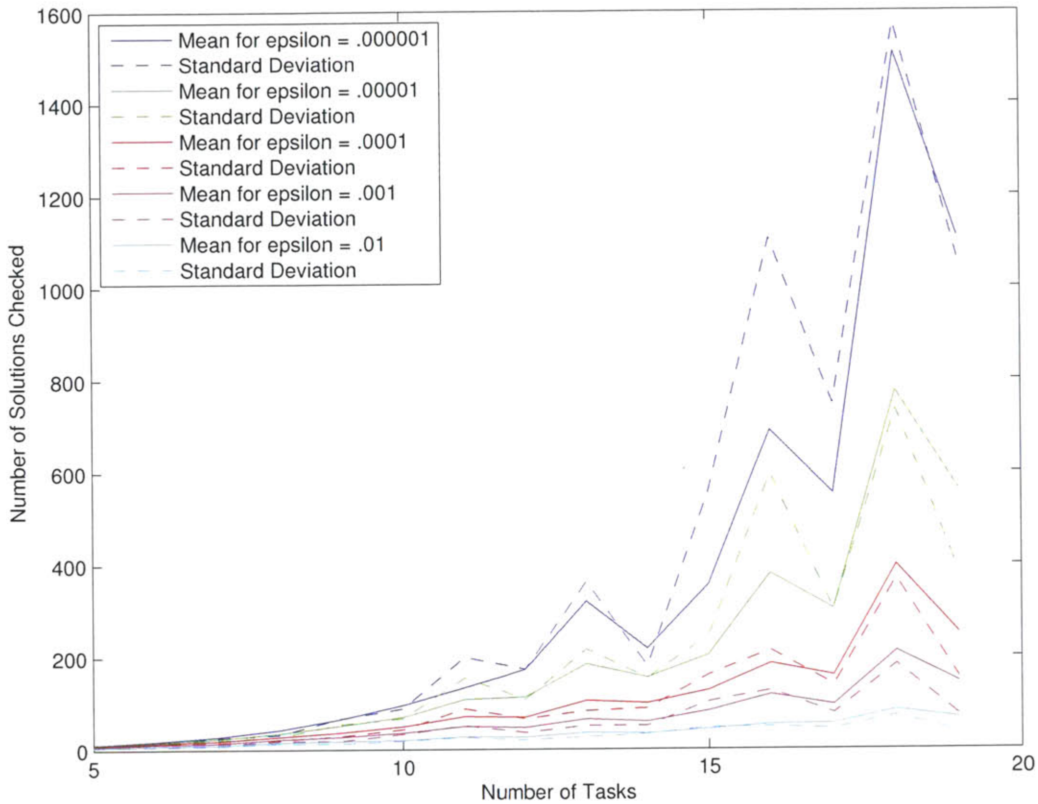


FIGURE 5.8. Number of Solutions Checked in Rounding Heuristic

arcs begin to have a positive flow across them, which means that to maintain a high level of performance we would need to decrease the value of ϵ with n.

We have established that the rounding heuristic greatly reduces the number of solutions checked and the performance of the rounding heuristic is near optimal for a well chosen value of ϵ, but we will now analyze how much faster the rounding heuristic is compared to the exact formulation. Figure 5.10 shows the mean and standard deviation of the time necessary to solve the relaxed version of problem 4.3.7 for one hundred example problems of each size n solved. As shown in Figure 5.3 the average time necessary to solve a problem of size 10 is one fifth the same amount of time necessary to solve the exact formulation. It takes $O(n)$ time to test each possible solution given from the rounding heuristic, but the analysis on the run time necessary to choose the best solution is left for future work. One of the benefits of the rounding heuristic is that once the relaxed problem is solved we can look at the solution to decide upon the best ϵ to use. Therefore we can always pick an ϵ which will provide a solution relatively fast, which is another reason why we are rather unconcerned with the run time necessary to choose the best solution from those which the rounding heuristic gives to check.

5.4. The Route Alteration Formulation

To analyze the route alteration formulation we will use the same method for randomly generating example problems as described for for the stochastic time expanded formulation.

5.4.1. Run Time Analysis. We will repeat the same analysis done for the run time of the stochastic time expanded formulation for the route alteration formulation. Figure 5.11 shows the mean solution time and standard deviation to find optimal solutions to the route alteration formulation. As expected Figure 5.11 shows us that the solution speed for the route alteration formulation is a lot slower than for the stochastic

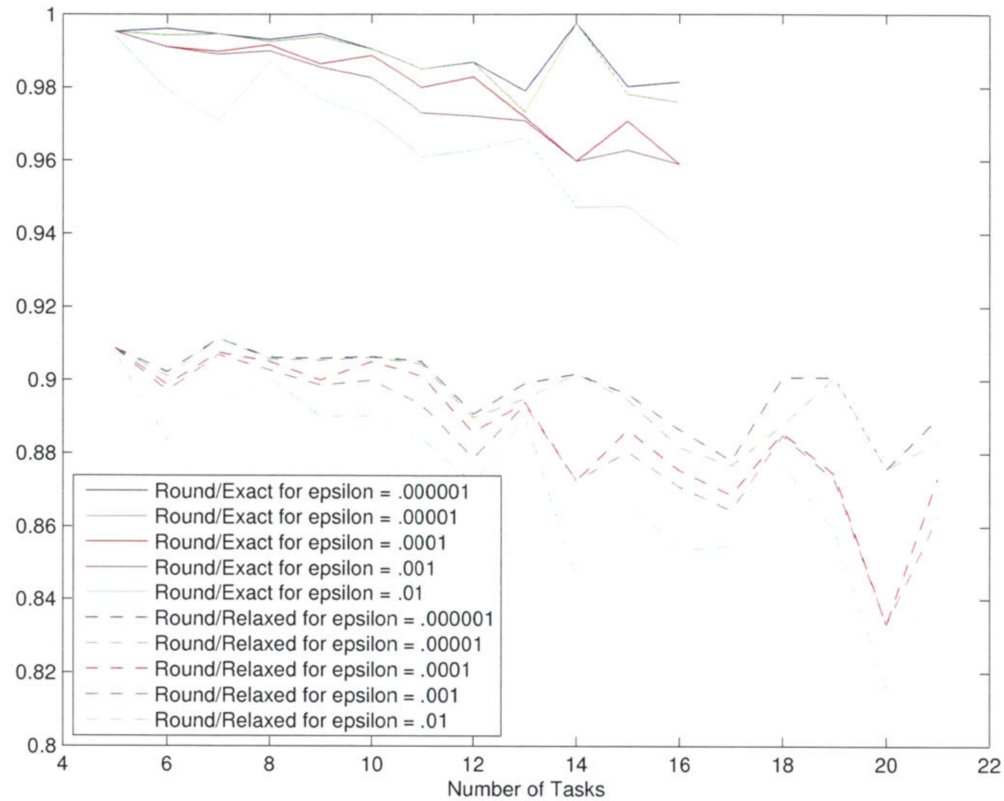


FIGURE 5.9. Performance of the Rounding Heuristic

time expanded formulation. To see if the added time of solving the alteration formulation is worthwhile we compare the optimal solution for the stochastic time expanded formulation to the alteration formulation. Figure 5.12 shows us that optimal solutions for the stochastic time expanded formulation stays within approximately 10 percent but its performance varies widely. The belief is that as n increases the benefit of using the route alteration formulation is diminished, but no strong conclusion can be gained from the data.

5.4.2. Heuristics. Heuristics for the route alteration formulation will involve two steps. First finding a "good" route to the stochastic time expanded formulation and using it to find a "good" route and alteration strategy for the route alteration formulation.

Nearest Neighbor/Rounding Heuristic for Plan Alteration Formulation
begin

Let $\widehat{\mathbf{y}}$ and $\widehat{\mathbf{x}}$ be the result from solving the stochastic time expanded formulation either exactly or with a heuristic.

Let m be the first task completed in the route $\widehat{\mathbf{y}}$ which satisfies, $\sum_{(m.1.t,j.0.s)\in A_s} \widehat{x}^{j.0.s}_{m.1.t} < 1$.

If no such m exists let m be the last task completed on route $\widehat{\mathbf{y}}$.

Let $\mathbf{y} = \widehat{\mathbf{y}}(m)$ represent the constraints,

$$y^{j.l}_{i.k} = \widehat{y}^{j.l}_{i.k} \text{ if } \widehat{y}^{j.l}_{i.k} = 1 \text{ and tasks } i,j \text{ are complete before task } m \text{ on route } \widehat{\mathbf{y}}$$

Let $\mathbf{y}(m)$ be the solution to the route alteration formulation with the added constraints $\mathbf{y} = \widehat{\mathbf{y}}(m)$.

Return the route $\mathbf{y}(m)$.

end

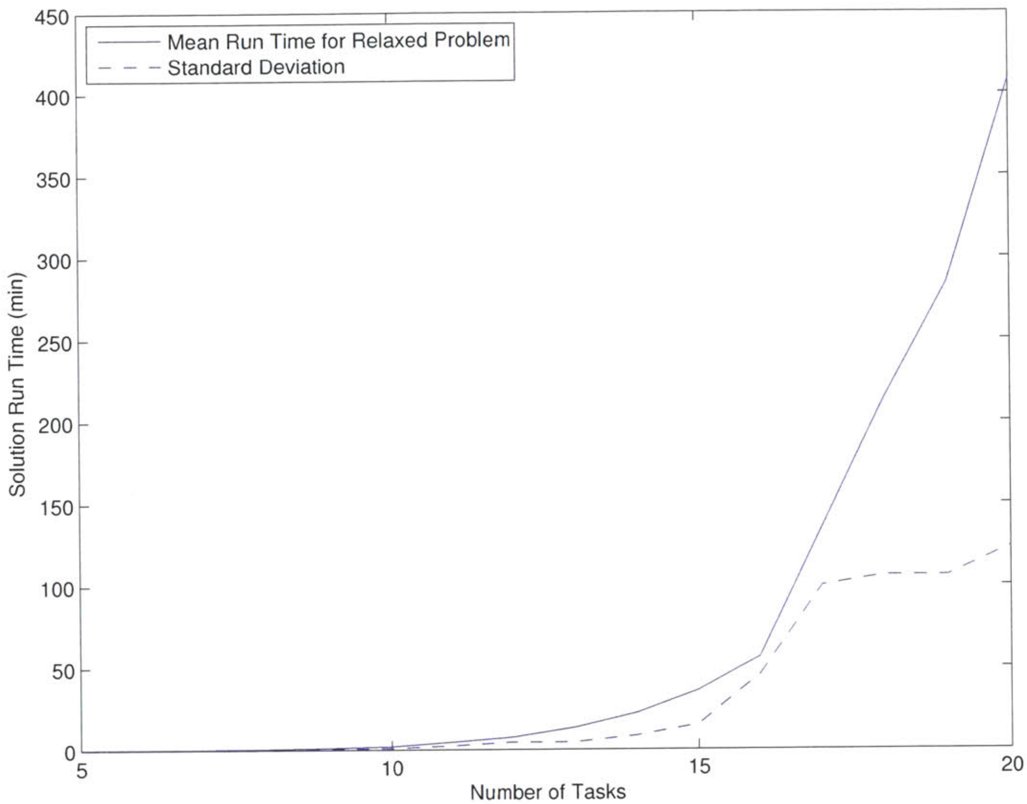


FIGURE 5.10. Run Time of the Relaxed Problem Necessary for the Rounding Heuristic

This heuristic depends on the fast calculation of the solution $\mathbf{y}(m)$. Figure 5.13 shows the average run time and standard deviation for the time needed to find the solution $\mathbf{y}(m)$. We can see that as compared the run time needed to find optimal routes the run time to compute $\mathbf{y}(m)$ is very fast.

To analyze the performance of the route alteration heuristic we will use the route given by the rounding heuristic with $\epsilon = .000001$ for the stochastic time expanded formulation as the base solution $\widehat{\mathbf{y}}$. Figure 5.14 shows us how close to optimal the route alteration heuristic performs and suggests that this heuristic performs within 10% of the optimal solution but there is a negative trend as n increases. A simple reason for this decline is caused by the slight decreasing performance of the ϵ heuristic for the stochastic formulation. Another reason believed to cause this decreasing performance is that by using a solution to the stochastic formulation as an initial solution to the route alteration formulation, the heuristic is not able to take advantage of being able to alter the route as much as in the exact formulation.

5.5. Conclusions

Route planning for a UUV is a hard problem to solve, as shown in the analysis on the run time to find solutions. It is thought, however, that as the total amount of time for a mission increases, the number of tasks will decrease and the resolution needed to solve the problem will become more coarse. This is because as operation areas increase in size and the total amount of time the UUV has available increases, users will define tasks in more general terms by grouping relatively short tasks which are related and are relatively close to one another. For these reasons we conclude that for large problems the rounding heuristic provides a "good" solution quickly, while for most real world problems Formulation 4.3.7 will be able to be solved exactly.

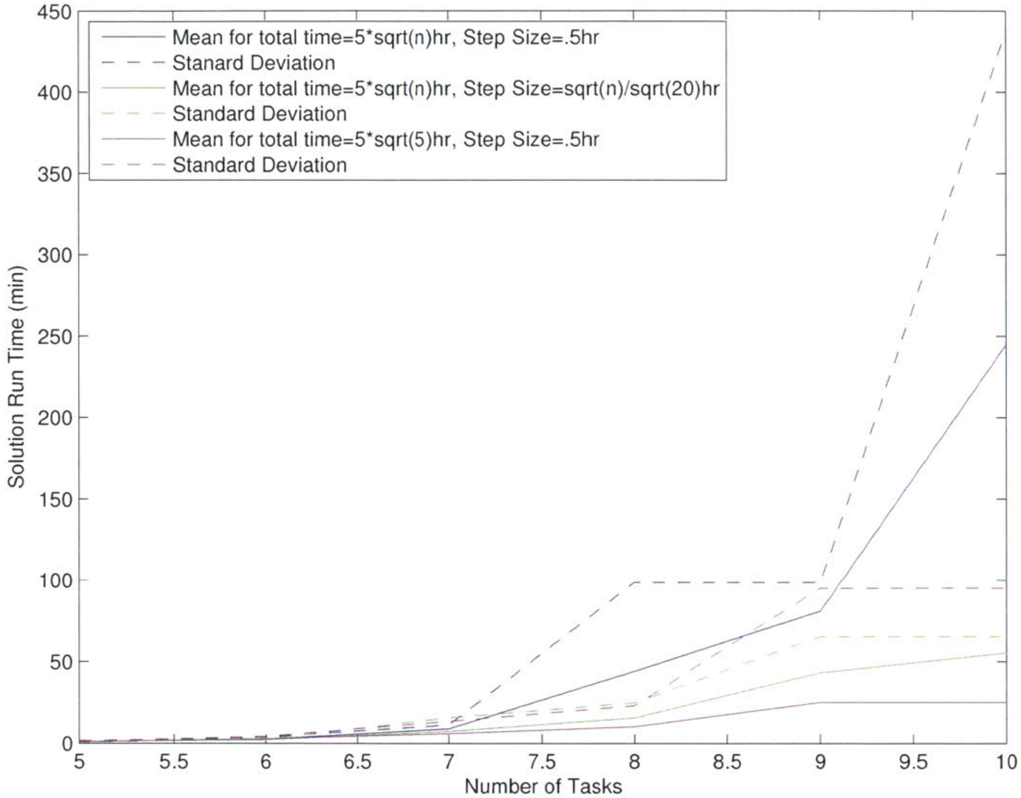


FIGURE 5.11. Route Alteration Formulation Run Time

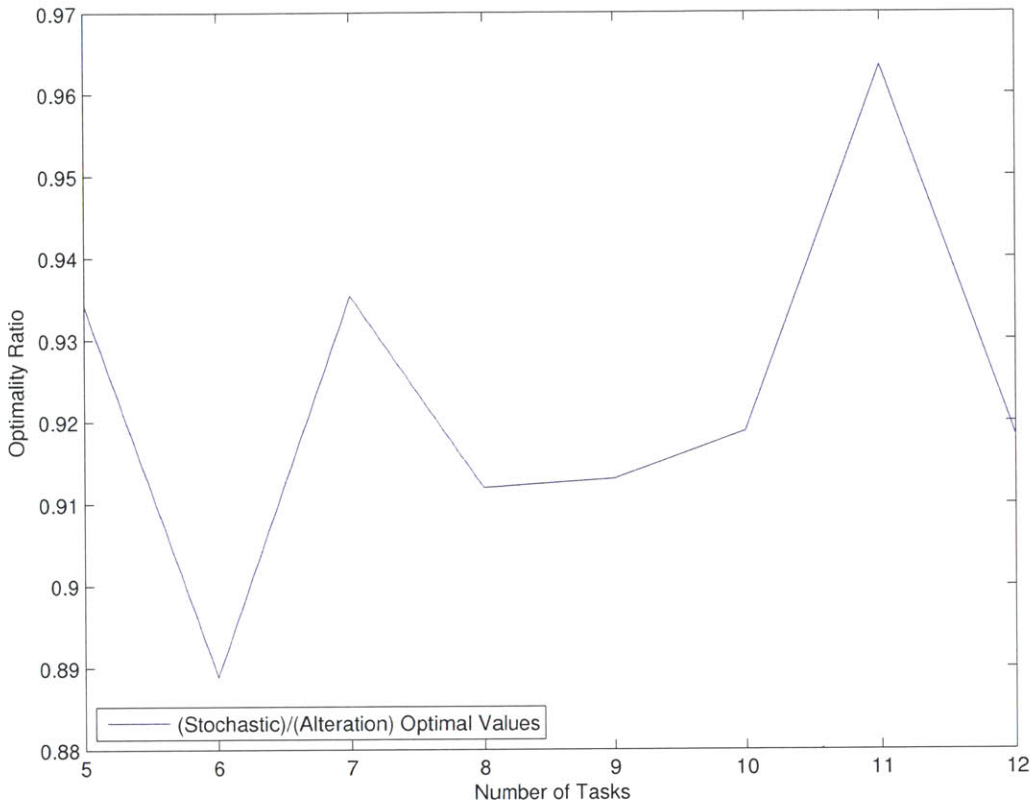


FIGURE 5.12. Stochastic Time Expanded Formulation Performance Compared to Plan Alteration Formulation

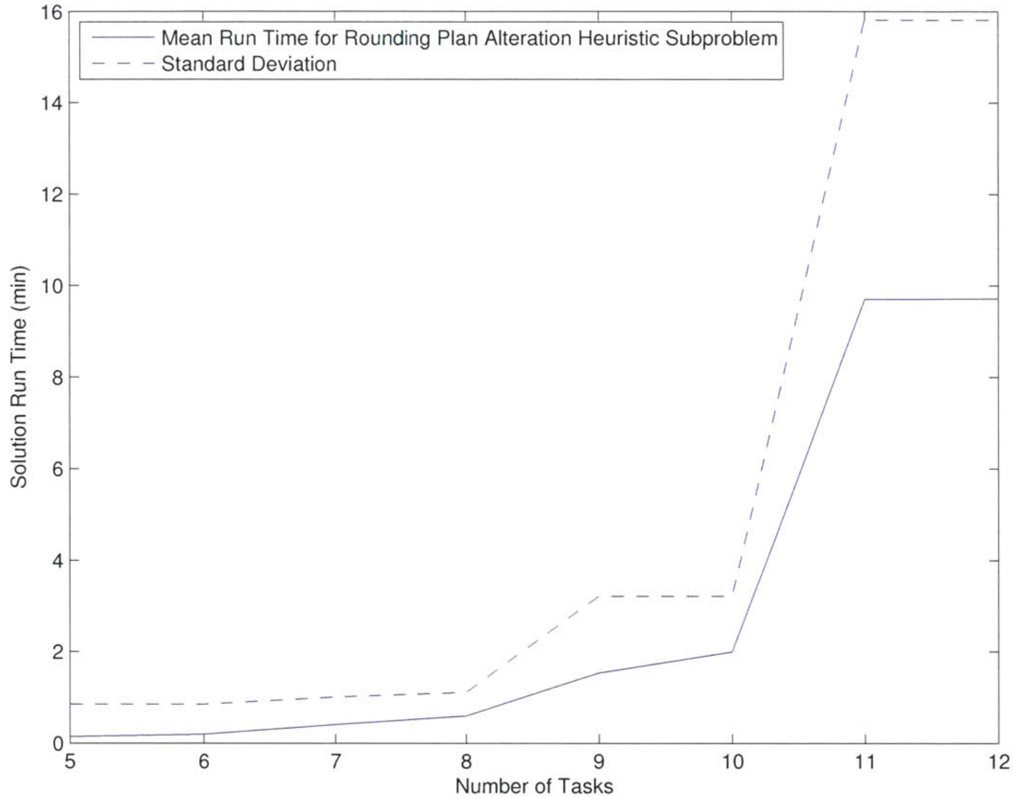


FIGURE 5.13. Rounding Heuristic for Plan Alteration Formulation Sub-problem Run Time

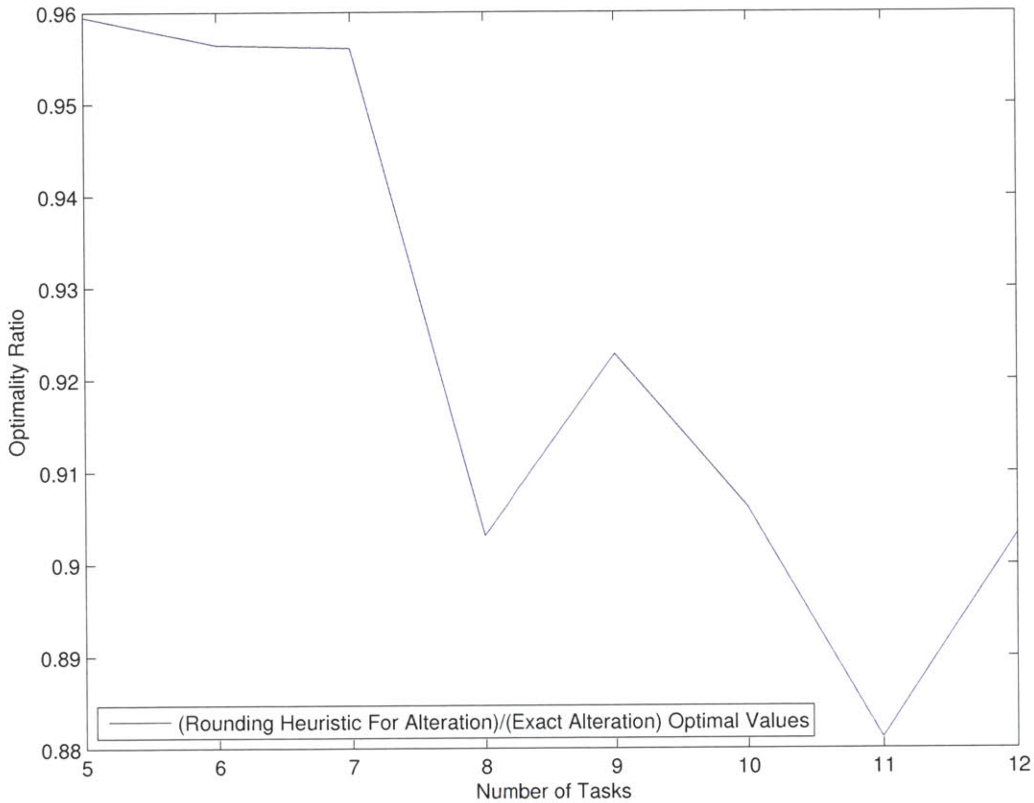


FIGURE 5.14. Rounding Heuristic for Plan Alteration Formulation Performance

CHAPTER 6

Conclusion

The objective of this research was to develop a usable formulation of the route planning problem for a UUV. This problem was approached through a series of formulations which increased in their ability to capture realistic problems a UUV will face. The thesis aimed to find robust solutions in a stochastic setting.

We started in Chapter 2 with a naive formulation which could not capture real world situations, and did not provide robust solutions, but provided guidance for subsequent formulations. The decision variables, constraints, and concept of the simple formulation 2.2.1 were the backbone of this thesis.

Chapter 3 continued, to introduced time dependent no go zones into the problem, and consequently introduced time expanding graphs to handle the added complexity. Formulation 3.3.1 handled a more realistic problem, but still provided solutions which were not robust. The deterministic time expanded graph led directly to the stochastic time expanded graph by taking an arc in the former and splitting it to model the stochasticity of the transit times and develop the latter.

With the stochastic time expanded graph built, the goal was to constrain from below the probability the UUV reached the destination node within the total time allotted. To accomplish this we derived constraints, and ultimately built the linear programming problem 4.3.6 designed to calculate the probability the UUV transits on an arc in the stochastic time expanded graph with a flow variable on the graph. Lemmas 4.9, and 4.10 proved the usefulness of 4.3.6 and allowed us to constrain from below and maximize over the flow variables on the stochastic time expanded graph to equivalently constrain from below and maximize over the probabilities of transiting on an arcs in the stochastic time expanded graph. This result led to formulation 4.3.7.

Next we took Formulation 4.3.7 and we continued to add even more realism in the decision making the UUV has with Formulation 4.4.1. Then we discussed potential objective functions for Formulations 4.3.7 and 4.4.1, and continued to reintroduce an energy constraint into these formulations. The method used for the energy constraint was general and provided a guideline for adding further resource constraints to a UUV route planning problem.

The conclusion of the main body of this thesis does analysis and provides heuristics for the UUV route planning problem. The analysis focused on Formulations 4.3.7 and 4.4.1 because these are considered the main contributions of this thesis. We looked at the run time of Formulation 4.3.7 with and without an energy constraint considered, and for differing values of the step size used to discretize the problem. We also looked at the trade off between Formulations 4.3.7 and 4.4.1 in performance and and computational speed. Then we considered the nearest neighbor heuristic, did analysis for whether or not this heuristic would provide solutions similar to optimal solutions, and for what type of distance metric should be used with the heuristic. Finally a rounding heuristic was introduced and analyzed in run time, and performance as compared to an optimal solution.

This thesis successfully develops a formulation of the route planning problem for a UUV which captures many real world characteristics. The route planning problem is shown to be computationally difficult with the evidence being the time needed to find exact solutions. The computational analysis this thesis provides a guideline for setting problem problem parameters so that a solution can be found. For larger problems the ϵ rounding heuristic developed is shown to be a good alternative to finding exact solutions and allows the user to balance computation time and performance with the choice of the value of ϵ. Overall, the results for the formulations developed provide promising results that suggest handling the route planning problem as we have in this thesis is relatively tractable when considering the complexity of the problem at hand.

6.1. Further Topics

6.1.1. Further Analysis. This paper has focused on the formulation of the UUV task planning problem but there are many areas of analysis left to be done. One of the important problem parameters which has not been analyzed is the value of β. In all of the analysis done in Chapter 5 we set $\beta = .9$. Sensitivity analysis on the value of β is necessary to help users decide on the right value of β for a specific real world problem. Due to the constraints used in the relaxation of Formulation 4.3.7 it is believed that smaller values of β would increase the duality gap between the exact formulation and the relaxed problem, which could effect the performance of the rounding heuristic.

For all of the randomly generated example problems we used the same 10 nm by 10 nm box with the same three time dependent no go zones and the same current. It is important to know how different problem settings would effect the use of the exact formulations and of the heuristics. More importantly, are there problem settings which lead to a drastic increase in the time needed to solve the exact formulation, or which lead to heuristics performing poorly? For example, one of the benefits of the rounding heuristics is the drastic reduction in the number of solutions which are checked which lends the question of if there are problem settings which cause the number of solutions which are checked to be relatively large.

The formulations proposed in this thesis depend on the discretization of time. An interesting area of further work would be to see how much this discretization effects the correctness of the formulations. When the continuous real valued random variables are realized, does the UUV really have a probability of β of reaching the destination node by the end time, and if not what is the true confidence level? Obviously the finer the resolution the more accurate the formulation will be, but we have seen that a finer resolution increases the time needed to find an optimal solution greatly. Finding the appropriate balance is necessary to implement the proposed formulations in real world problems.

6.1.2. When to Resolve the Problem. As a UUV is following a proposed route it will gain knowledge of its surrounds about the bathymetry and about the true strength of the current. One of the benefits of the formulations we have proposed is that the $\mathbf{x}$ values calculate the probability the UUV reaches nodes at certain times. This means that if the UUV finds itself in a low probability state, it may be beneficial to resolve the problem with the first hand knowledge gained about the situation. An important next step to this thesis would be to get an idea on when the UUV should resolve the problem, knowing that it wants to limit the total number of recalculations to not waste time and energy. Another issue is that we can more accurately predict problem settings such as current in the short term, so a formulation which has a high resolution at the beginning and a decreasing resolution as time continues may be beneficial as far as the time needed to find solutions, and then the UUV can choose to resolve the problem once a resolution threshold has been surpassed.

6.1.3. Generating More Robust Solutions. In the stochastic time expanded model we use $\mathbf{p}$ as a problem parameter. To calculate the values of $\mathbf{p}$ we use the knowledge we have about the bathymetry, the no go zones, and the current. In real world problems it is likely that we do not have perfect knowledge of these problem settings which means that the values $\mathbf{p}$ become random variables. It may, however, be much more practical to put each value in the vector $\mathbf{p}$ in a window so that $\mathbf{p} \in [\widehat{\mathbf{p}} - \widetilde{\mathbf{p}}, \widehat{\mathbf{p}} + \widetilde{\mathbf{p}}]$. This lends itself to applying the Bertsimas-Sims method to the problem, and testing how this formulation performs as compared to the formulations in this thesis.

6.1.4. A New Type of Task. In this thesis we only considered tasks where the UUV must go to a location and perform some type of act which we have an idea on the time, and energy it will take to complete. In real world situations UUV's must manage between performing tasks and surveillance. In surveillance the UUV must choose how long to stay on station and when to depart the location. We can incorporate surveillance into the route alteration formulation in the following way.

Let $i \in I$ be the nodes which represent the performance of a task. Let $i \in \overline{I}$ be the nodes which represent surveillance. Since the UUV can choose how long to stay on station for surveillance we can alter the parameter $\mathbf{p}$ by saying $p_{i.0.t}^{j.0.s} = 1 \ \forall s \geq t$ and $i \in \overline{I}$. Due to the randomness of transit times we do not know exactly when the UUV will arrive at a surveillance location but we can control when the UUV leaves. Let us say that we want to pick one specific time t the UUV must leave a surveillance location. That would mean we want to pick one t value such that $z_{i.1.t} = 0$. So we can incorporate the addition of these surveillance nodes by replacing the constraint,

$$z_{i.1.t} \leq z_{i.1.t+1} \text{ with } \sum_{t=0}^{T}(1 - z_{i.1.t}) = 1 \ \forall i \in \overline{I}.$$

This constraint would make the UUV choose one time t for the UUV to leave a surveillance node if it wants to continue on with the route, and if the UUV arrives as node $i.k.s$ where $s \neq t$ then the UUV must go to the destination node.

There would also be a change in the reward function. Let r_i be the reward gained for completing task i if $i \in I$ and let r_i be the rate the UUV gains reward for surveillance at i if $i \in \overline{I}$. Then we would change the objective function to be,

$$\sum_{i \in I} r_i \left(\sum_{(i.0.t,i.1.s) \in A_s} x_{i.0.t}^{i.1.s} \right) + \sum_{i \in \overline{I}} r_i \left(\sum_{(i.0.t,i.1.s) \in A_s} (s-t) x_{i.0.t}^{j.1.s} \right)$$

This is one possible way to formulate the addition of surveillance nodes in our problem. Further work needs to be done on the performance of this formulation.

6.1.5. Incorporating GPS Fixes. One nuance specific to a UUV problem is that when submerged the UUV cannot fix its location exactly and tries to estimate its position with its relative speed in the water and its best guess on the strength of the current. It happens quite frequently that a UUV believes it is performing a task such as sea floor mapping on the area intended but in reality it is performing the task on another area all together. This means that the quality of the completion of each task depends on the time since it last got an exact fix of its location. To get an exact fix on its location the UUV must surface, which takes time, energy, and increases the chances of detection. With this in mind it would be useful to have a formulation which allows the UUV to specifically plan when it will take GPS fixes. To incorporate fixes into a route we can add duplicate arcs to the decision graph G which represent the UUV getting a GPS fix. So we would have arcs $(i.1, j.0)$ which represent not taking a fix and arcs $(i.1, j.0f)$ which represent taking a fix. So we would have variables $y_{i.0}^{i.1}$ and $y_{i.0}^{i.1f}$ in to the formulation. Then in the time expanded graph we would have the corresponding variables $x_{i.1.t}^{j.0f.t}$. The addition of these variables does not change the general formulation of the stochastic time expanded graph. If we want the UUV to be able to sometimes choose to take a fix and sometimes not, depending on the time the UUV arrives at a node then we would need to incorporate new binary variables similar to the alteration variables $\mathbf{z}$. This addition in problem complexity can be added easily to the route alteration formulation. The real difficulty of incorporating GPS fixes is how to change the reward function for tasks to depend on when the UUV made its last fix. This reduces to being able to keep track of the time since the last fix for the UUV. As we saw in our analysis the time needed to solve exact formulations is very dependent on the number of $\mathbf{x}$ variables, which means we want to stay away from adding another dimension to the time expanded graph.

One way we can keep track of the last time since a GPS fix is by adding flow constraints to the decision graph G which represent the expected time since the last fix. Let $w_{i.k}^{j.l}$ represent the expected time since the last fix when the UUV arrives at node $j.l$. Consider the constraints below designed to set the values for $\mathbf{w}$.

$$\sum_{(i.k,j.l) \in A} w_{i.k}^{j.l} - \sum_{(j.l,i.k) \in A} w_{j.l}^{i.k} \geq \sum_{(i.k.t,j.l.s) \in A_s} (s-t) x_{i.k.t}^{j.l.s} - T y_{i.k}^{j.lf} \ \forall i.k \in N$$

This constraint says that if the UUV goes on arc $(i.k, j.lf)$ then the we can set the time flow out of node $i.k$ to zero. If the UUV goes on arc $(i.k, j.l)$ then the time flow out of node arc $i.k$ must be at least the amount of time flow into node $i.k$ plus the expected amount of time it will take to transit on arc $(i.k, j.l)$. Since $\mathbf{w}$ will have a negative impact on the objective function this constraint would properly calculate the values $w_{i.k}^{j.l}$. So then we could change the reward function to be,

$$\sum_{i=1}^{n} \left(r_i \left(\sum_{(i.0.t,i.1.s) \in A_s} x_{i.0.t}^{i.1.s} \right) - \alpha_i w_{i.0}^{i.1} \right).$$

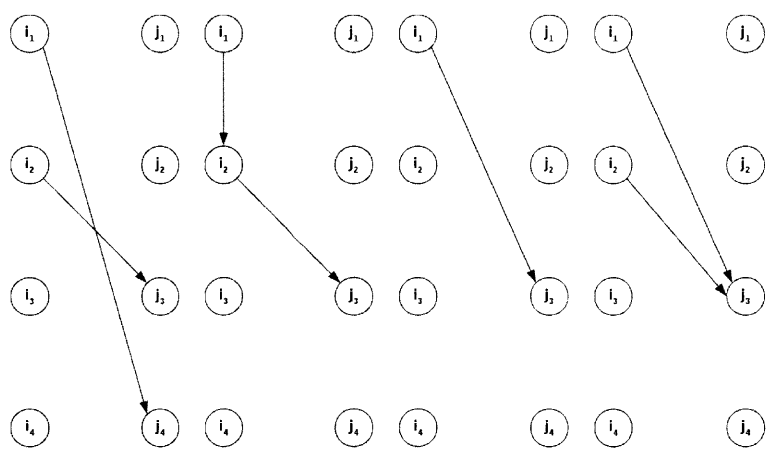
Where α_i is a specific amount that defines how much the reward decreases (linearly) as the time since the last fix increases. This linearly penalizes the time since the last fix for each task. This proposal for how to handle GPS fixes works for the stochastic time expanded formulation. An area for more research would be how to incorporate tracking the time since the last fix in a route alteration formulation. It is believed that time flow variables placed on the time expanded graph G_a would work for a route alteration formulation.

APPENDIX A

A.1. Overtaking

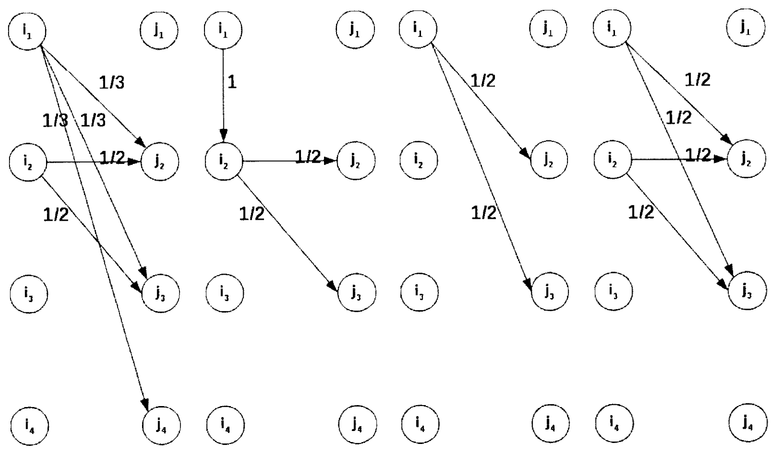
First we will discuss overtaking in a deterministic time expanded graph. To understand overtaking consider a time expanded network shown in Figure 1(a). The arc (i_2, j_3) overtakes arc (i_1, j_4) because its departure time is bigger but its arrival time is smaller. For the UUV problem overtaking arcs are caused by time dependent no go zones being active and then inactive. The UUV can always wait and depart at a later time, so it could follow the arcs shown in Figure 1(b). Instead of having arcs which represent the UUV waiting for a no go zone to become inactive, we can make the wait time inherent. So in Figure 1(c) we replace the arcs (i_1, i_2), and (i_2, j_3) with the single arc (i_1, j_3). This means that for the UUV problem we can transform the initial time expanded graph shown in Figure 1(a) with the graph shown in Figure 1(d). Making this transformation allows us to assume there are no overtaking arcs in the deterministic time expanded graph G_d for the UUV problem.

To consider overtaking in a stochastic time expanded graph consider Figure 2(a) where the number on the arcs is the probability we travel on that arc when departing from the corresponding node. As shown in Figure 2(a) if we leave node i_1 then the probability we reach node j by time 3 is 2/3 but if we leave from node i_2 the probability we reach node j by time 3 is 1. This is an example of overtaking in a probabilistic setting. For a UUV problem since the UUV can choose to wait then the best way to travel from node i_1 to node j to reduce the expected amount of time in transit is shown in Figure 2(b). To eliminate the arc (i_1, i_2) we can equivalently use the graph shown in Figure 2(c) to represent the best way to travel from node i_1 to node j. This means that for the UUV problem we can transform the initial stochastic time expanded graph shown in Figure 2(a) to the graph shown in Figure 2(d). Making this transformation allows us to assume that if a UUV leaves a node at time t the probability it reaches node j by time s is greater than or equal to the probability the UUV reaches node j by time s if it leaves at a time $t^+ \geq t$.



(a) Initial Graph (b) Allowing Waiting at a Node (c) Replacing Waiting Arcs (d) Transformed Graph

FIGURE A.1. Overtaking Figures For Deterministic Time Expanded Graph



(a) Initial Graph (b) Allowing Waiting at a Node (c) Replacing Waiting Arcs (d) Transformed Graph

FIGURE A.2. Overtaking Figures For Stochastic Time Expanded Graph

A.2. Deterministic and Stochastic Time Expanded Graph Transformation

In Corollary 4.13 we say that there is a simple transformation between a stochastic time expanded graph G_s built for a deterministic case, and the deterministic time expanded graph built for that same case. Since we are dealing with a deterministic case we know exactly what time the mission starts so we can assume the start time is time 0. This means that in the graph G_s there is no arcs entering to nodes $S.1.t$ for $t > 0$. Also, the only arc leaving node $S.0.0$ is arc $(S.0.0, S.1.0)$. This means we can delete all nodes $S.1.t$ for $t > 0$ and node $S.0.0$ from the graph since they are essentially useless. By deleting these nodes from G_s the resulting graph is exactly graph G_d. This is the simple transformation between G_s and G_d mentioned in Corollary 4.13 which makes the graphs G_s and G_d identical. With this transformation we can interchangeably use the flow variable $\mathbf{x}$ for problem 3.3.1 with the $\mathbf{x}$ flow variable in problem 4.3.7.

A.3. Useful Lemma

The following lemma is used when constructing Formulation 4.3.7, and when reinserting the energy constraint in Section 4.6.

Lemma A.1. *Let,*

$$
\begin{aligned}
f =& \max \boldsymbol{c}' \boldsymbol{x} \\
& s.t. \\
& \boldsymbol{x} \in X \\
& g(\boldsymbol{x}) \leq b \\
g(\boldsymbol{x}) =& \min \boldsymbol{d}' \boldsymbol{y} \\
& s.t. \\
& \boldsymbol{y} \in Y(\boldsymbol{x}) \\
h =& \max \boldsymbol{c}' \boldsymbol{x} \\
& s.t. \\
& \boldsymbol{x} \in X \\
& \boldsymbol{d}' \boldsymbol{y} \leq b \\
& \boldsymbol{y} \in Y(\boldsymbol{x})
\end{aligned}
$$

Where X is the feasible region for the decision vector $\boldsymbol{x}$ and $Y(\boldsymbol{x})$ is the feasible region for the decision vector $\boldsymbol{y}$ dependent on $\boldsymbol{x}$. Then $f = h$.

PROOF. ($\leq$) Let $\mathbf{x}$ be a feasible solution to f. Let $\mathbf{y}$ be the optimal solution to $g(\mathbf{x})$. Then $\mathbf{x}$, $\mathbf{y}$ are feasible solutions to h.

($\geq$) Let $\mathbf{x}$ and $\mathbf{y}$ be a feasible solution to h. Let $\mathbf{y}^*$ be the optimal solution to $g(\mathbf{x})$. Then $g(\mathbf{x}) = \mathbf{d}^{'}\mathbf{y}^* \leq \mathbf{d}^{'}\mathbf{y} \leq b$. So $\mathbf{x}$ is a feasible solution to f.

$\square$

THIS PAGE INTENTIONALLY LEFT BLANK

Bibliography

[1] Ahuja, Ravindra; Magnanti, Thomas; Orlin, James. *Network Flows.* Prentice Hall, INC., Upper Saddle River, NJ. 1993.

[2] Alvarenga Guilherme Bastos; Mateus Geraldo Robson; de Torni, G. *Finding near optimal solutions for vehicle routing problems with time windows using hybrid genetic algorithm.* citeseer.ist.psu.edu/646969.html.

[3] Alvarenga, Guilherme Bastos; Mateus, Geraldo Robson. *A two-phase genetic and set partitioning approach for the vehicle routing problem with time windows.* Hybrid Intelligent Systems, 2004.

[4] Bard, Johathan F.; Kontoravdis, George; Yu, Gang. *A branch-and-cut procedure for the vehicle routing problem with time windows.* Transportation Science. 36(2):250–269, May 2002.

[5] Bent, Russell; Hentenryck, Pascal Van. *Dynamic vehicle routing with stochastic requests.* 2003. citeseer.ist.psu.edu/bent03dynamic.html.

[6] Bent, Russell; Hentenryck, Pascal Van. *A two-stage hybrid local search for the vehicle routing problem with time windows.* Transportation Science, 38(4):515-530, 2004. citeseer.ist.psu.edu/brent01twostage.html.

[7] Bertsekas, Dimitri. *Dynamic Programming and Optimal Control* (Vol 1). Athena Scientific, Belmont, MA. 2005.

[8] Bertsimas, Dimitris; Sim, Melvin. *The Price of Robustness.* Operations Research. *52*, 35-53. 2004.

[9] Bertsimas, Dimitris; Tsitsiklis, John. *Introduction to Linear Optimization*, Athena Scientific, Belmont, MA. 1997.

[10] Braysy, Olli; Gendreau, Michel. *Vehicle Routing Problem with Time Windows, part i: Route construction and local search algorithms.*Transportation Science, 39(1):104-118, February,2005.

[11] Cates, Jacob. *3-Paving Circulants and the Kadison-Singer Extension Problem.* Senior Thesis. United States Naval Academy. 2009.

[12] Dejax, Pierre; Feillet, Dominique; Gendreau, Michel. *Traveling Salesman Problems with Profits.* Transportation Science. *39*, 188-205. 2005.

[13] Laporte, Gilbert; Martello, Silvano. *The Selective Travelling Salesman Problem.* Discrete Applied Mathematics. *26*, 193-207. 1990.

[14] Larson, Richard; Odoni, Amedeo. *Urban Operations Research.* Dynamic Ideas, Belmont, MA, 2007.

[15] Lawler, Eugene; Lenstra, Jan; Rinnooy Kan, Alexander; Shmoys, David. *The Traveling Salesman Problem, A Guided Tour of Combinatorial Optimization.* John Wiley & Sons Ltd., New York, NY. 1985.

[16] Tsiligirides, Theodore. *Heuristic Methods Applied to Orienteering.* Operational Research Society. *43*, 797-809. 1984.

[17] *The Navy Unmanned Undersea Vehicle (UUV) Master Plan*, April, 2004.