Mission-level Planning with Constraints on Risk

by

Cheng Fang

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of

Master of Science in Aeronautics and Astronautics

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2014

© Massachusetts Institute of Technology 2014. All rights reserved.

Signature redacted

Author ..................

Department of Aeronautics and Astronautics

May 22, 2014

Signature redacted

Certified by .................................

Brian C. Williams

Professor of Aeronautics and Astronautics

Thesis Supervisor

Signature redacted

Accepted by .................................

Paulo C. Lozano

Associate Professor of Aeronautics and Astronautics

Chair, Graduate Program Committee
Mission-level Planning with Constraints on Risk

by

Cheng Fang

Submitted to the Department of Aeronautics and Astronautics
on May 22, 2014, in partial fulfillment of the
requirements for the degree of
Master of Science in Aeronautics and Astronautics

Abstract

The problem of routing vehicles for data collection is common in the scientific world. In underwater surveys with autonomous vehicles, for example those conducted by the Woods Hole Oceanographic Institute, the autonomous underwater vehicle (AUV) must visit a list of sites to take samples. The order in which the locations are visited should be such that samples are taken during specified time windows, and the total distance travelled is relatively short. An automated planner could act as a force multiplier for all such scientific missions.

While current planners in the operations research community can perform vehicle routing with deterministic traversal and deterministic time windows, they are unable to account for stochasticity inherent in real world applications. For example, the time of traversal for vehicles between locations may be uncertain, due to disturbances such as varying currents. Arrival deadlines may also be stochastic, due to the probabilistic times of occurrence for interesting phenomena at each location. Lastly, there is a non-zero probability of failure during traversal, when the vehicle strikes the sea floor due to sudden changes in topography. The ability to plan under uncertainty is vital when we require probabilistic guarantees on success.

In this thesis, I consider a novel stochastic single-vehicle routing problem, and propose a chance-constrained solution method. As with previous work, I assume that there exists some roadmap representation of the world with associated costs of traversal, and the vehicle must visit a set of goals subject to a list of temporal constraints. The novelty of my problem is due to the addition of probabilistic times for edge traversal, probabilistic temporal constraints, and stochastic breakdowns on edge traversal. I present a solution method which returns a list of waypoints and a schedule of events which minimises the cost of traversal, subject to an upper bound on the probability of breakdowns and failure to meet temporal constraints.

Noting dependence of travel time on path taken, I decompose the problem into two stages. The first stage finds, in best first order, solutions which allow the vehicle to visit the goal locations subject to the upper bound on the probability of breakdowns. Using each solution, the planner can extract the corresponding probabilistic durations for travel times and rewrite these as probabilistic temporal constraints. The second
stage solves the resulting chance-constrained probabilistic simple temporal problem (CC-pSTP), in which events must be scheduled such that the probability of failing any of the temporal constraints is below the specified bound. I show that the joint solutions to the first and second stages correspond to valid waypoints and schedules for the original problem, and work through an example problem for an AUV survey mission.

Thesis Supervisor: Brian C. Williams
Title: Professor of Aeronautics and Astronautics
Acknowledgments

I am grateful for the heroic efforts of my advisor Professor Brian C. Williams, and Dr. Howard Shrobe. This quite literally would not have been possible without concessions and help from both, and I understand that professionalism in planning and scheduling is a necessary part of my development going forward. I also thank Dr. Scott Sanner for the initial chance he handed me and everything else since then.

I would also like to thank all the wonderful people floating around MIT during my time here thus far. To all the members of MERS, especially Pedro, Eric and Peng, thanks for making the lab an amazing place to work and banter. I thank Patrick, Wei, and Luis for welcoming me to the Danger Zone, and I thank Hang and Anthony for being house mates who never settle. I would also like to thank my friends from outside MIT, especially the bunch of crazies at Porter Square, and the artists and dancer and fellow USyd engineer in Boston, and the passive-aggressive players of Bang!. I would like to acknowledge the boys back home for all the support and encouragement, and the enjoyment of bad movies and the terrifying KFC pie. Life would have been a lot more dreary without all of you.

I thank Lele for being lovely.

Last but not least, I would like to thank my parents, Xuan Fang and Qin Guo, for the sacrifices they have made for me, the love and support they have given me, and for being inspirational figures. Both are highly competent, hard-working, honourable and professional people, and they have done all they can in the interests of my education and my welfare, even to the detriment of their own careers. I am beyond grateful, and I hope I have made and will continue to make you both proud.

This thesis was supported by the National Science Foundation under Grant No. IIS-1017992. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the view of the sponsoring agencies.
Contents

1 Introduction 9
  1.1 Problem characteristics 10
  1.2 Related approaches 11
  1.3 Solution approach 13

2 Failure and risk 15
  2.1 Failure 15
  2.2 Risk associated with choices 16
  2.3 Contributions 18

3 Chance-constrained path-planning with temporal constraints: the approach 19
  3.1 Problem statement 20
  3.2 Analysis of chance-constrained MPC 24
  3.3 Chance-constrained travelling salesman with temporal constraints 28
  3.4 Two-stage solution 32
  3.5 Conclusion 36

4 Chance-constrained roadmap planning 37
  4.1 Problem overview 37
    4.1.1 Overview of the approach 39
  4.2 Roadmap for chance-constrained path planning 41
  4.3 Mapping to resource-constrained traveling salesman 42
Chapter 1

Introduction

Using automation is fraught with risks. Google Now can under estimate the time that it takes to get to the airport. Autonomous underwater vehicles can become stuck on the sea floor while performing risky manoeuvres. With each introduction of new automation we receive great benefit, but we also assume greater risk. However, the burden of managing this risk today falls on the shoulders of the user. When failures occur, the users are accountable for failing to build in margins for error in specifying desired outcomes.

As autonomous systems become more ubiquitous, this paradigm becomes increasingly outdated. Studies overwhelmingly point to the inability of humans to assess risk, from finance [40] to vehicle operation [14]. When autonomous systems are increasingly used by individuals as a black box (see applications in robotic nursing homes [37] and medical robotics [2] among others), we can not reasonably expect humans to properly adjust expectations when dictating goals and conditions.

My thesis is that the autonomous system must be able to assess risk and propose options that operate within acceptable levels of risk, given goals set by humans. This thesis focuses on risk-sensitive decision making in the context of scheduling and path planning decisions, such as those that would enable Google Now to propose safe routes and schedules to make an important flight, or enable Woods Hole Oceanographers to better plan and schedule science missions to meet required science goals.
1.1 Problem characteristics

Coupled path-planning and time management lies at the heart of important applications. A route and schedule advisor like Google Now will have to work over possible routes, with varying degrees of traffic, to ensure that the user arrives at the airport before the boarding time of the flight. The app will have to provide plans and schedules accounting for the uncertain traffic conditions and flight arrivals. Without the ability to provide guarantees on meeting the schedule deadlines in the presence of uncertainty, such advisory systems will be pointless because the user will not have confidence in the advice.

An alternative application is the planning for scientific observation missions, for example in the multi-million dollar oceanography cruises conducted by Woods Hole Oceanographic Institute (WHOI). Given rotating shifts for crew members on a scientific cruise, Woods Hole Oceanographers currently plan traversals at sea between sampling sites such that data is gathered and analysed when the appropriate experts are awake. The chief scientist responsible for the plan of the day will need to further adjust for the uncertain time required for sample collection, and plan trajectories in the face of changing sea conditions. Failure to devise robust plans for the observation mission would result in failure to complete scientific objectives, wasteful in light of the large financial and time investment.

Optimisation for coupled path-planning and scheduling problems is thus a matter of urgency, as solutions will increase the efficiency of a wide range of operations, from daily routines to knowledge accumulation. The defining characteristics which must be addressed are as follows:

- Uncertainty in traversal and durational outcomes. In real world applications, we cannot take outcomes for granted. Traversal along a highway could take between 30 minutes to two hours, depending on congestion. The underwater vehicle could become stuck on the bottom of the seafloor during traversal, or the actual sampling could take longer than expected due to obscured features on the sea floor. The problem must capture such uncertainty and the solution
must leave margins of error.

- A close coupling between path planning and scheduling. The timing of events in such problems is dependent on the way the system moves. The timing constraint of arriving at the airport before boarding can only be checked given the path to the airport. The time of sample collection can only be determined once a proposed path is known. All of this points to the close connection between path-planning and the occurrence times of important events in the mission.

- A set of more than one location. For missions we are interested in, there are typically multiple destinations. A WHOI mission may need to perform microbial sampling at one site before moving to another to perform methane flux estimation. A user commuting to the airport may prefer to pick up a cup of coffee and rendezvous with colleagues before heading to the airport. Thus, it is crucial for the planner to allow multiple goal locations.

In the following section, I discuss how prior work has attempted to capture and work with the above characteristics, and how my approach extends the state of the art.

1.2 Related approaches

Traditionally, the community has studied the uncertainty in how the state of the system changes, and considered the risk of arriving at an undesirable state. Newer literature has sought to address the equally important problem of punctuality, so that agents can be guaranteed to be on time, despite uncertainty in the duration of actions. In this thesis, I consider a third case which requires both punctuality and safety of the agent.

Whereas prior work has adopted a paradigm balancing risk and reward, this should not be the only paradigm adopted in planning under uncertainty. Some applications, for example aviations, energy supply and science missions, may require probabilistic guarantees on successfully meeting requirements rather than a trade off between the
risk of failure and the cost of execution. This is because the cost of failure is too great and there is no way to balance cost and risk. Consider for example a science mission with a custom submersible vehicle, as often performed by the Woods Hole Oceanographic Institute. It is difficult to assign costs to failed data gathering during science mission. It is even more difficult to price a vehicle failure: costs for salvage and repair must be considered, in addition to the opportunity cost in terms of science data which may have otherwise been gathered on subsequent missions.

Thus in the case of potentially catastrophic failure, in which the price of failure is too great to balance against the cost of actions, an alternative approach must be adopted. In this thesis, I adopt the chance-constrained paradigm for planning under uncertainty. The approach results in executable plans which will be successful with a provable probabilistic guarantee. In this way, instead of balancing the risk of catastrophic failure, we choose to act in such a way that the probability of such outcomes occurring are limited and conform to a user specified bound.

In relation to the coupling of path planning and scheduling and the capability for handling multiple goals, there is prior work in the literature, for example the family of vehicle routing with time windows problems [15]. These have relevance to our problems, especially autonomous underwater vehicle (AUV) missions such as those outlined in [45], plans must be made a priori. For time-varying environments, the vehicle must visit the sites of interest during specific time windows in order to gather data. However, such formulations only capture the idea of time windows, or deterministic bounds on the timing of an event.

The prior arts with time windows do not account for stochasticity inherent in real world applications. For example, the time of traversal for vehicles between locations may be stochastic, the duration of interesting physical phenomena at each location may be stochastic, and there is a non-zero probability of component failure during the traversal of the map. These can not be naturally fitted into the vehicle routing with time windows framework. While there exist prior literature on vehicle routing with stochastic travel times [25], to the best of our knowledge no literature exists which addresses the problem in the generality desired.
1.3 Solution approach

Noting the difference between our requirements and the capabilities of prior arts, I proceed in this section to outline my approach to the problem.

I decompose the problem into two stages to address the coupling between path-planning and scheduling. In the first, I perform chance-constrained path-planning to generate candidate paths which visit a set of goal locations. In the second, I construct a chance-constrained scheduler, which attempts to assign times to events such that timing constraints can be met with probabilistic guarantees.

In this thesis I propose, implement and test the chance-constrained path-planning with probabilistic durations problem. I show that a resource-constrained travelling salesman approach is appropriate for generating the candidate paths, as the underlying roadmap representation allows us to work with longer horizons. I further formulate a chance-constrained scheduling problem based on the strong controllability algorithm for simple temporal problems with uncertainty, a classic formalism for robust scheduling under uncertainty. The chance-constrained scheduling problem is formed by the user imposed timing requirements, as well as those implied by the candidate paths, and solved using the chance-constrained scheduler developed.

In the subsequent chapters, I first outline a probability theoretic notion of failure and risk. Given the framework, I formally state the overall problem to be solved. Noting the dependence of the temporal problem on the path planning aspects of the problem, I decompose the overall problem and attack them separately as: a) Efficient chance-constrained path planning; b) Chance-constrained mission scheduling. I perform numerical experiments on the methods derived, and analyse the methods for behaviour with respect to chance-constraints and scalability.
Chapter 2

Failure and risk

In this chapter, we motivate and provide a general definition of failure. From this definition, we can describe the probability of failure given a set of choices. Under some technical assumptions about execution strategy used for making mission choices, we can further describe the risk of execution strategies. These abstract definitions are important in exposing the technical assumptions required, and hence provide the foundations of our rigorous chance-constrained approach to robust planning.

2.1 Failure

Suppose there exists a set of constraints on system states which must be satisfied. Failure occurs when the state of the system does not meet the constraints. This is the intuition behind our definition of failure.

Let \( X \in \mathbb{X} \) be the state of the system. In addition, let \( C = \{c_1 \}, \ldots, c_1 \) be a set of constraints, where each constraint \( c \in C \) is a tuple \( \langle f_c, A_c \rangle \), where:

- \( f_c : X \rightarrow \mathbb{R}^{m_c} \); and
- \( A_c \subset \mathbb{R}^{m_c} \)

for some \( m_c \in \mathbb{N} \).

Intuitively, we have a function \( f_c \) mapping from the state space to some feasible region \( A_c \). This is stated formally as:
Definition 1. For state $x \in X$, $x$ satisfies constraint $c \in C$ if $f_c(x) \in A_c$. A state $x$ is a success if it satisfies every $c \in C$. Otherwise, we denote $x$ a failure.

The definition reflects the intuition that, for a state to be a success, it must satisfy every constraint. Conversely, if any of the constraints are violated, we consider the state a failure. However, this definition does not exclude disjunctive constraints. In particular, we can define $A_c$ to be unions of disjunctive sets in state space. Note that state $x$ and the state space $X$ is defined generally. We illustrate the power of the definition in the example below.

Example 1. 2-D path finding with obstacle avoidance Consider a 2-D path planning problem for a vehicle with planning horizon $n$, we may consider the entire sequence of poses to be one state. That is, $x = \{x^{(0)}, y^{(0)}, \ldots, x^{(n)}, y^{(n)}\}$, $X = \mathbb{R}^2n$. Let the constraints be that the vehicle must end within some area $A_{\text{goal}}$, and must avoid a set of obstacles $A_O = \bigcup_i A_{O_i}, A_{\text{goal}}, A_{O_i} \subset \mathbb{R}^2$. Then, the constraints will be:

1. End in goal: $c_1 = (f_{c_1}, A_{c_1}), f_{c_1}(x) = \{x^{(n)}, y^{(n)}\}, A_{c_1} = A_{\text{goal}}$

2. Avoid obstacles: $c_2 = (f_{c_2}, A_{c_2}), f_{c_2}(x) = \{x^{(0)}, y^{(0)}, \ldots, x^{(n)}, y^{(n)}\}$, and $A_{c_2} = ((A_{O})^c)^n$ where $(A_{O})^c$ denotes the complement of $A_O$

We thus have a general definition of failure. We proceed to demonstrate one definition of the risk or probability of failure associated with choices.

2.2 Risk associated with choices

Consider stochasticity over the course of a mission. For example, in a vehicle routing problem, we may have uncertain durations of travel from one location to another, or unexpected component failures during the execution of activities. We may describe this randomness with continuous or discrete random variables, provided that we have some probabilistic model for these outcomes.

We assume that the state of the system can be uniquely determined by assignments to a set of controllable variables and outcomes of the random variables. Suppose we
have assigned the values of the controllable variables. Then for some set of outcomes of random variables, we have resulting states which are failures.

It is useful here to recall the standard definition of random variables as measurable functions from the sample space to the reals. That is, for some probability space \((\Omega, \mathcal{F}, P)\), each random variable is a function \(V : \Omega \to \mathbb{R}\), such that if \(V \in B(\mathbb{R})\), \(R^{-1}(B) \in \mathcal{F}\), \(B(\mathbb{R})\) the Borel \(\sigma\)-algebra generated by \(\mathbb{R}\). While measurability is vital for technical details such as well-defined probabilities and integrability, this definition is also helpful in relating failure to the underlying sample space \(\Omega\). We can thus provide a principled definition of the probability of failure as a measure of some subset in the sample space for the random variables, under some mild assumptions.

Let \(X \in X\) denote the state of the system, and suppose that there exists a tuple \((V, U)\), where:

- \(V\) is a vector of controllable variables, the value of which may be assigned from \(V\); and
- \(U\) is a vector of random variables \(U : \Omega \to U\), probability space \((\Omega, \mathcal{F}, P)\).

Assume that there exists a measurable function \(f : V \times U \to X\), such that \(X = f(V, U)\). Further, for each \(c \in C\), \(A_c \in B(\mathbb{R}^{m_c})\) and \(f_c\) measurable. We may then state Definition 2.

**Definition 2.** Suppose we have some assignment \(v\) to \(V\). Then \(r(v)\) the risk of \(v\) is defined as

\[
r(v) = P(A_v)
\]

where \(A_v = \{\omega \in \Omega | f(v, U(\omega))\ is\ a\ failure\}\)

The definition above explicitly defines the probability of failure when executing some set of choices \(v\). Given a set of choices, the probability of failure is just the probability of samples of the uncontrollable variables which result in failed states.

We show below that this formulation is principled.

**Theorem 1.** For any \(v \in V\), \(r(v)\), is well-defined.
Proof. Note that, for \( x \) successful, \( x \in \bigcap_{c \in C} B_c \), where \( B_c = \{ x' \mid f_c(x') \in A_c \} \) for each \( c \in C \). Thus for \( f(v,u) \) failure, \( f(v,u) \in (\bigcap_{c \in C} B_c)^c \).

By measurability of \( f_c \) and because \( A_c = B(\mathbb{R}^{m_c}) \) for all \( c \in C \), \( B_c \) measurable for all \( c \in C \). Thus, \( (\bigcap_{c \in C} B_c)^c \) measurable. Since \( f \) is a measurable function and \( U \) random variables, \( f(v,\cdot) \) is a measurable function from \( U \) for any \( v \).

Since \( A_v \) is the pre-image of \( (\bigcap_{c \in C} B_c)^c \) with respect to \( f(v,\cdot) \), \( A_v \) is measurable for any \( v \). The probability of \( A_v \) is then well defined. \( \square \)

2.3 Contributions

In this thesis, I propose to plan robustly by providing solutions with upper bounds on risks. Thus, it is important to define exactly what we mean by risks, and ensure that they are well-behaved. The definition must be general enough to cover a range of applications, while also outlining which restrictions must be met in order for the ideas to be well-defined in a rigorous sense.

Towards this end, I have made the following contributions in this chapter:

- I have characterised failure in a set-theory manner;
- I have characterised risk, or the probability of failure, in a measure-theoretic manner;
- I have shown that, under a set of specific conditions on what constitutes failure, risk is well-defined in the measure theoretic sense.

The last result is particularly important, as it informs us how we should formulate our problems. By stating our problems and making sure our constraints satisfy measurability, we may refer back to the results in this chapter to obtain the appropriate risk functions.
Chapter 3

Chance-constrained path-planning with temporal constraints: the approach

When we plan for an excursion, for example a trip to the airport with stopovers to pick up friends, or a multi-week oceanography mission coordinating travel times between sites of interest organised to match the 8 hour shifts of the crew, we are performing path-planning with temporal constraints under uncertainty.

When we are planning for a carpool to the airport, we consider the order with which friends are picked up, check the estimated driving times with apps like Google Maps, and tally the driving durations to see whether we will make it in time for boarding. If our initial plan does not work out, we rearrange the pick up order and departure times until we are assured of being punctual to the flight. We must do all this, while making estimates of how much the traffic conditions vary.

When mission plans are drafted aboard a WHOI oceanography cruise, the chief scientist consults the watch schedule for the working and sleeping hours of experimental scientists and AUV operators. He must then order the sequence of traversals and visits to sample sites so that the AUV operators are awake to conduct sampling operations, and scientists are available to analysis the time sensitive samples. After trying out several alternatives, the chief scientist must negotiate with the captain of
the ship and the chief engineer, who may request further planning based on timing constraints not known to the chief scientist. Through out all of this, the scientist must make estimates on how the traversal times, data gathering durations, and the required time for experiments vary.

In real life, the problem of path-planning in the presence of temporal constraints is thus widespread and important. In addition, as the examples show, the human planner must juggle a multitude of constraints in a potentially iterative process in order to arrive at solutions. Working over a large number of decision factors and constraints, it is difficult to find plans which would be optimal in any sense: usually we just accept a plan which meets the constraints. Given the multitude of constraints, we often also fail to give enough thought to the robustness of the solution, resulting in failing to meet timing constraints, or taking risky actions which result in failure.

A more desirable paradigm for planning would thus be the creation of autonomous decision support advisors, for example the new Google Now effort. The autonomous systems would be able to keep track of how each decision affects the overall plan, balancing risks and costs of traversal against the need to satisfy temporal constraint. While probability and risk assessment is counter-intuitive to humans, the autonomous system can make quantitative judgements on the probability of failure, providing guarantees on meeting goals and specifications.

This chapter outlines my approach to chance-constrained path-planning with temporal constraints. I give a full statement of the problem in the next section. Subsequently, I look at recent work using model predictive control as a basis for a solution to the problem, and analyse its short comings with respect to longer time scales. Instead, I propose a generate and test framework, in which candidate paths are generated by the path planner and tested for consistency using a probabilistic scheduler.

3.1 Problem statement

In this section, I extract the relevant characteristics of the chance-constrained path-planning problem with temporal constraints, and incorporate these into my definition.
The problem statement must encapsulate the following:

- A set of goal locations which must be visited - The typical mission would require traversal to a number of different locations. The commuter trying to catch a flight must pick up his friends, and the Woods Hole Oceanographers must perform heterogeneous sampling operations at a number of different sites.

- A cost of traversal - Not all paths are equal. While a detour for the commuter away from the city may lead to much higher fuel costs, and a planner advising this route may be resented. Similarly, the chief scientist on a WHOI cruise would likely prefer to get the cruise done as soon as possible, in order to fit in secondary goals in the remaining time.

- A notion of failure during traversal - Due to changing traffic conditions, a commuter may be stuck in a gridlock, never to move again in the time frame of the commute. The WHOI missions may include vehicle sampling trajectories which are unsafe, where the AUV collects samples over highly varying seafloor topologies and is more likely to become stuck on the bottom.

- A set of temporal constraints - Timing constraints are important, as these are the constraints we check feasibility against, and these are the deadlines we try to meet. A path minimising the fuel consumption is pointless if the commuter misses his or her plane, and a cruise plan which gathers all the samples will be meaningless if the scientists are not on watch to process the samples as they are collected.

- A notion of the uncertainty in the durations of activities - The duration of the activities depend on many factors. In the commuter case, a traverse activity has time varying with the traffic conditions. In the WHOI scenario, the time it takes for sampling to be conducted is difficult to predict beforehand, and depends on the familiarity of the operators with the vehicle, as well as the underwater conditions. The problem statement must allow some specification of the uncertainty in timing.
• A set of user specified confidence levels. We typically adjust our schedule and our decisions for an operation based on the guarantees of success which we require. For example, the commuter might start the commute much earlier in order to have some margin of error for the estimated traversal times. The chief scientist typically proposes ambitious plans which are adjusted by the ship's captain and the chief engineer, in order to have a better chance of success. The problem statement will also include an user specified risk bound, which upper bounds the probability of failure, whether in traversal, or with respect to temporal constraints.

The rigorous statement of the problem is given as follows:

**Problem 1. (Chance-constrained path planning problem with temporal constraints)**

Let $X_S$ be the state space. Let the goal set $G$ be a finite set of measurable subsets of the state space, such that $G = \{G_1, \ldots, G_{|G|}\}$, $G_1, \ldots, G_{|G|} \subseteq X_S$. Further, let the failure regions be $\mathcal{O}$ a finite set of measurable subsets of the state space, such that $\mathcal{O} = \{O_1, \ldots, O_{|\mathcal{O}|}\}$, $O_1, \ldots, O_{|\mathcal{O}|} \subseteq X_S$. We choose our path from a family of uncertain paths $\mathcal{C}$, a set of stochastic processes such that for every $C \in \mathcal{C}$ and $t \in [0, \infty)$, $C(t)$ is a random variable on $X_S$. We further have a cost function $f : \mathcal{C} \rightarrow \mathbb{R}^+$. In addition, let us assume a set of times $X$, describing events within the mission. A subset of time $X_b$ can be assigned by the planning capability. The complement of $X_b$ in $X$, $X_e$ is comprised of uncontrollable elements, such that each $e_i \in X_e$ is the sum of element in $X_b$ a one random variable over $\mathbb{R}$ representing an uncertain duration. Assume further that there are constraints $R_e$ on the difference between elements in $X$, requiring for each $r_e \in R_e$ that $x_{c_i} - x_{c,j} \in A_e$, for $A_e$ a measurable subset of $\mathbb{R}$ and $x_{c_i}, x_{c,j} \in X$.

Suppose we are given an upper bound on the probability of failing the traversal, $\Delta_E$, and an upper bound on the probability of failing to meet the temporal constraints, $\Delta_T$. Then we are required to find $X_b^*$ assignments to $X_b$ and $C^*$ an uncertain path such that:
\[ C^* = \arg \min_{C \in \mathcal{O}} f(C) \]

subject to

- \( P \left( \{ \bigcup_{i \in \mathcal{G}} \{ C(t) \notin G_i, \forall t \in [0, \infty) \} \} \cup \{ \exists t \in [0, \infty) s.t. \bigcup_{i \in \mathcal{O}} \{ C(t) \in O_i \} \} \right) < \Delta_E \)

- \( P(\bigcup_{e \in \mathcal{E}} \{ x_{e,i} - x_{c,j} \notin A_c \}) < \Delta_T \)

where \( P \) is the probability measure associated with the random variables.

In this formulation, the family of all possible trajectories through state space \( X_S \) are described by \( C \). Intuitively, we cannot uniquely determine where the system will be at each instant in time. The position of the commuter at each time instant during the commute cannot be exactly predicted. Similarly, we cannot know ahead of time whether an AUV will be stuck on the seabed. Thus, for each nominal trajectory we choose, we represent the actual outcomes as a stochastic process.

The choice of trajectories through state space can be considered one in which we pick a stochastic process which is likely to achieve the specifications on the state. These specifications are described by the goal regions in \( \mathcal{G} \) and the failure regions in \( \mathcal{O} \). It is required that the trajectory must reach all sets in \( \mathcal{G} \) while avoiding the sets in \( \mathcal{O} \), such that all goals are reached while the failure states are avoided. For the AUV example, the set of goals may be regions in the 2D space on the seafloor, while the failure region may be any region in 3D space on or below the seafloor. Through such a specification, the AUV is required to visit every goal region while staying above the seafloor.

The preference for some trajectories over others are captured by the objective function \( f \). This may represent considerations such as fuel consumption, expected speed of completion, among other factors.

Additionally, we have requirements on the timing of events. Note that we have events which are completely controllable by the system, and events which depend on outcomes in the environment. While we may schedule an AUV to start moving
towards a new goal, we can not control when a vehicle can arrive at the destination. We also impose restrictions on the timing constraints. For example, if we would like to specify that $G_1$ is visited after $G_2$, we may insert events for arrival at goals $T_1$ and $T_2$, and the temporal constraint $r$ which requires $T_1 - T_2 \in [0, \infty)$.

Associated with the specifications in state and time are the chance-constraints. In the first case, we must restrict the probability of the trajectory failing to visit all goal regions, or travelling into a failure region, to be less than $\Delta_E$. In the second case, we must limit the probability of the timings being outside the allowed bounds to be less than $\Delta_T$. Noting that the constraints work over measurable subsets of the state space and the times, we know that the probability of failure is well defined.

Note that there are two separate aspects of the problem here. There is a clear subset of the descriptions dealing with the state of the system, namely $X_S$, $G$, $C$ and $C$. There is another subset dealing with the temporal aspect of the system, namely $X$ and $R_c$. This motivates a combined approach to the overall problem building on two dedicated methods for the spatial and temporal subproblems. In the following section, I analyse a recent model predictive control approach to chance-constrained path planning, corresponding to the spatial subproblem. I show its deficiencies and motivating a roadmap approach for path planning over longer horizons.

### 3.2 Analysis of chance-constrained MPC

In this section I revisit the chance-constrained framework for performing path planning with uncertainty as proposed in [32]. I present an analysis of the exponential blow up of the decision variables with increasing time steps to motivate a reformulation of the problem.

Let us begin by restating the original problem. Intuitively, the problem attempts to find *control inputs* which allow an agent with probabilistic state transition to go from an uncertain initial state to a set of desired goal states, such that the probability of the agent being in a failure state at any time step is bounded. We revisit the finite horizon formulation of the problem below.
Consider a simple planning scenario, in which an underwater robot is to move
from some initial position to some goal position, with execution time known a priori,
such that there is less than $\Delta$ chance of crashing. Given $\mathcal{T}$ the set of time steps in
the execution, $\mathcal{K}$ the set of obstacles, and $\mathcal{E}_k$ the set of constraints (faces/edges) for
each obstacle $k \in \mathcal{K}$, we may formulate the problem as:

$$\min_{\mathbf{u}_0:|\mathcal{T}|-1 \in \mathcal{U}^{||\mathcal{T}||}} J(\mathbf{u}_0:|\mathcal{T}|-1, \bar{x}_0:|\mathcal{T}|)$$

s.t. $$\bar{x}_{t+1} = A_t \bar{x}_t + B_t \bar{u}_t \ \forall t \in \mathcal{T}$$

$$\Pr \left( \bigwedge_{t \in \mathcal{T}} \bigvee_{k \in \mathcal{K}} h_{t,k,j} \bar{x}_t - g_{t,k,j} \geq 0 \right) \geq 1 - \Delta$$ (3.1)

$$x_0 \sim \mathcal{N}(\bar{x}_0, \Sigma_{x_0}), \mathbf{u}_t \sim \mathcal{N}(\bar{u}, \Sigma_{\mathbf{u}_t})$$ (3.2)

where:

- $\mathbf{x}_t$ is the state vector at time $t$, $t \in \{0, ..., |\mathcal{T}|\}$;

- $\bar{x}_t := E[\mathbf{x}_t]$ is the nominal state at time $t$, $t \in \{0, ..., |\mathcal{T}|\}$;

- $\mathbf{u}_t$ is the control input at time $t$, $t \in \{0, ..., |\mathcal{T}| - 1\}$; and

- $\bar{u}_t := E[\mathbf{u}_t]$ is the nominal control input at time $t$, $t \in \{0, ..., |\mathcal{T}| - 1\}$.

In this problem, we assume that the family of stochastic processes governing state
transition, $C$, is limited to those with Brownian motion and a normal distribution for
the noise, captured in the state vectors $\mathbf{x}_t$ for each time step. Further, the chance-
constraint on state is captured by (3.1). This is a mapping of the spatial subproblem,
from that defined in Problem 1.

Following the formulation in [33], we can write the constraints for the obstacles
with binary variables in the space. $\mathcal{Z} = \{0, 1\}^{\sum_{k \in \mathcal{K}} |\mathcal{E}_k|}$. The problem then becomes:
**Problem 2** (pSulu problem).

\[
\begin{align*}
\min_{u_0|\mathcal{T}| - 1 \in \mathcal{U}|\mathcal{T}|, z \in \mathcal{Z}} & \quad J(u_0|\mathcal{T}| - 1, \bar{x}_0|\mathcal{T}|) \\
\text{s.t.} & \quad \bar{x}_{t+1} = A_t \bar{x}_t + B_t \hat{u}_t + w_t \quad \forall t \in \mathcal{T} \\
& \quad \Pr \left( \bigwedge_{t \in \mathcal{T}} \bigwedge_{k \in \mathcal{K}} \bigwedge_{j \in \mathcal{E}_k} h_{t,k,j} \bar{x}_t - g_{t,k,j} + M z_{t,k,j} \geq 0 \right) \geq 1 - \Delta \\
& \quad \sum_{j \in \mathcal{E}_k} z_{t,k,j} = |\mathcal{E}_k| - 1 \quad \forall k \in \mathcal{K}, t \in \mathcal{T} \\
& \quad x_0 \sim \mathcal{N}(\bar{x}_0, \Sigma_{x_0}), u_t \sim \mathcal{N}(0, \Sigma_{u_t}) \quad (3.3)
\end{align*}
\]

where \( M \) is a large number such that \( h_{t,k,j} \bar{x}_t - g_{t,k,j} + M z_{t,k,j} \geq 0 \) always evaluates to true for \( z_{t,k,j} = 1 \). Essentially the binary variables are used as guards to indicate which constraints are active for each obstacle at each time step.s

We begin by noting polygonal approximations for obstacles are possible [4]. General obstacles may be thus described by the intersection of multiple linear constraints denoting the boundaries of the obstacle. By defining failure as the vehicle state \( x_t \) at time \( t \) being in the closure of an obstacle region, we thus define failure as the vehicle state \( x_t \) satisfying a conjunction of linear constraints representing one obstacle. More generally, with multiple obstacles, failure occurs when the vehicle state is in the closure of a set of the state space, where the set is defined as an union of the obstacles. That is, the mission has failed if the vehicle has entered at least one obstacle region.

We note that the vehicle state at time \( t \) is defined as an affine function of the random variables \( x_0 \) and \( u_t, i \in 0, t - 1 \). Thus, the vehicle state is a measurable random variable. In addition, because the failure states are defined as the union of a finite number of obstacles, each defined as the intersection of a finite number of linear constraints. The obstacle regions are thus also measurable. The probability of failure is thus well-defined. Hence the probability of success in (3.1) is well-defined. The discussion of risk thus proceeds on rigorous foundations.

While there are several extensions (iterative multistage solutions [32], multiagent problems [34]), the core formulation remains that in Problem 2. The formulation explicitly plans for each time step. That is, for each time step in the mission, there is a
set of decision variables which must be assigned by the planner. This has implications on the scalability of the approach to long horizons. I demonstrate this by considering the number of constraints needed to describe the polygonal obstacle boundaries.

For obstacle \( k \), there are \(|\mathcal{E}_k|\) ways of choosing which constraint are to be satisfied. Thus, for the set of obstacles at some time step, there are \( \prod_{k \in \mathcal{K}} |\mathcal{E}_k| \) ways of choosing the set of constraints. For the course of the mission, we have \( (\prod_{k \in \mathcal{K}} |\mathcal{E}_k|)^T \) ways of choosing constraints. Thus, the time complexity of the mixed-integer problem will scale exponentially with the number of time steps.

The characteristics of chance-constrained MPC thus provide motivation for a different approach to the mission level chance-constrained path planner, which will be able to address the difficulties encountered. Instead, we propose a roadmap based method for solving the spatial subproblem. By adopting this representation, we are able to address mission level path planning with guarantees on the entirety of the trajectory. I begin by addressing the key trade off between model predictive control and roadmap planning, and why I have chosen a roadmap representation.

The key insight behind the approach is that, supposing each edge denotes a single manoeuvre from one area of the map to another, then the risk incurred by traversing along the edge can be determined a priori. Similarly, the cost of taking the edge can be determined a priori. Further, these costs and risks are calculated independent of the length of the traversals. We are thus able to define roadmaps at the start of the planning process, representing not only the locations in the map and the paths between them, but also the risks and costs incurred along the path.

This is in contrast to model predictive control (MPC) paradigms, which assume that the cost and risk incurred at each time step is not known beforehand, and must be evaluated during planning. The assumption is reasonable in MPC frameworks, because the exact controls are not known at the start of the planning procedure. Controls are optimised with respect to the cost and risk functions, and must be computed for each time step. Thus, MPC methods will yield controls which make the best use of risk at each time step of the planning process. However, as costs and risks are calculated for each time step, the computations becomes intensive, and are
intractable for longer horizons.

The key difference is thus the use of manoeuvres in the roadmap approach. We are trading off the flexibility of MPC in optimising the paths for every time step for scalability independent of mission duration. Given that the chance-constrained path planner is derived in service of a mission level planner, the roadmap approach is preferable over arbitrarily long time scales and adopted in my approach.

In the following section, I address the conversion of Problem 1, the chance-constrained path planning problem with temporal constraints to a roadmap based problem.

3.3 Chance-constrained travelling salesman with temporal constraints

For autonomous surveys, we wish to impose an upper bound on the probability of failure. In this case, let the upper bound on probability of failure due to edge traversal be $\Delta_E$ and the probability of failure to meet the set of all temporal constraints $R_c$ be $\Delta_T$. We wish to find a sequence of vertices and full assignments to controllable events, such that the total cost incurred during graph traversal is minimised, subject to the probability bounds are satisfied.

Note that we can represent the set of goals in the state space as vertices in a roadmap. Let $V$ denote a set of vertices representing sets in the state space, and let $V$ be the subset corresponding to the goals $G$. Suppose further that we have ways of traversing between regions represented by the vertices. That is, let $v_1$ represent region $A_1$ and $v_2$ represent $A_2$, for $A_1, A_2$ measurable subsets of $X_S$, and let the manoeuvre be $C$. We represent each such manoeuvre with an edge $e_C$, and assign to it the cost of the manoeuvre $f(C)$. Further, assume we have a set of failure regions $O$. We calculate the risk of failure,

\[
p = P\left(\bigcup_{G_i \in \{A_1, A_2\}} \{C(t) \notin G_i, \forall t \in [0, \infty)\} \cup \left\{\exists t \in [0, \infty) \text{s.t. } \bigcup_{O_i \in O} \{C(t) \in O_i\}\right\}\right)
\]
This is the risk of failure when traversing using manoeuvre $C$. We may represent this by associating a Bernoulli random variable with edge $e_C$, which takes value 0 with probability $p$ and value 1 otherwise, representing failure when evaluated to be 0.

Our problem is then in full a chance-constrained travelling salesman problem with probabilistic failure on edge traversal, probabilistic deadlines, and probabilistic travel duration, with additional temporal constraints. Denote it a chance-constrained travelling salesman with temporal constraints in short. The problem is described by the tuple $\langle V, \hat{V}, E, C, B, X_b, R_d, R_c, \Delta_E, \Delta_T \rangle$, with $V, X_b, R_d, R_c, \Delta_E, \Delta_T$ as discussed, $E$ the set of edges, $C$ the set of costs for each edge, and $B$ the Bernoulli random variables for failure in traversing each edge. We require as solution some sequence $V^* = \{v_1, v_2, ..., v_n\}$ of vertices in $V$ and full assignments $T = \{t_1, t_2, ..., t_m\}$ to $X_b$. Specifically:

**Problem 3.** (*Chance-constrained travelling salesman with temporal constraints*) Find $X_b, V^*$ so that

$$V^* = \arg \min_V f(V)$$

subject to

- $v$ appears at least once in $V^*$, for every $v \in \hat{V}$
- $E$ is the sequence of all $e_i \in E$ connecting $v_i, v_{i+1}$ for all $v_i, v_{i+1} \in V$
- $f(V) = \sum_{e_i \in E} c_{e_i}$
- $P(\cup_{e_i \in E} \{b_{e_i} = 0\}) < \Delta_E$
- $P(\cup_{e_i \in E} \{x_{c,i} - x_{c,j} \notin A_c\}) < \Delta_T$, for our choice of $X_b, V$ and set of random durations $R_d$

Related problems have been attacked in the literature. Consider the vehicle routing with time windows problem [15], in which a team of agents must visit multiple locations during specified time windows, and minimise the total distance travelled. The problem is also known as the travelling salesman with time windows when only
one agent is available. Apart from being relevant to transportation systems as suggested, there exists applications in scientific surveys with autonomous vehicles. For autonomous underwater vehicle (AUV) missions such as those outlined in [45], plans must be made a priori. For time-varying environments, the vehicle must visit the sites of interest during specific time windows in order to gather data.

However, the deterministic vehicle routing with time windows problem does not account for stochasticity inherent in real world applications. For example, the time of traversal for vehicles between locations may be stochastic, the duration of interesting physical phenomena at each location may be stochastic, and there is a non-zero probability of component failure during the traversal of the map. These can not be naturally fitted into the vehicle routing with time windows framework.

While there exist prior literature on vehicle routing with stochastic travel times [25], the problem encoding does not allow stochastic timing constraints. Thus, to the best of our knowledge no literature exists which addresses the problem in the generality desired. Related problems and solution methods are summarised in Table 3.1.

Our chance-constrained travelling salesman problem with temporal constraints is thus difficult to approach. This is also partly because of the chance constraint on temporal consistency, the satisfaction of which is also dependent on our choice of $V^*$. For example consider a simple tour from Start to two locations, V1 and V2, with a wait at each location of at least 120 minutes, as seen in Figure 3-1. The order in which the locations are visited give rise to the two different pSTNs.

A subset of the uncertain durations in the temporal problem thus result from vehicle traversals. Further, the topology of the temporal problems also change depending on the path chosen. Hence, the probability of temporal consistency can not
Figure 3-1: The topology of the pSTN changes depending on the path taken. In this example, the vehicle traverses the roadmap in a), leaving from Start to visit V1 and V2, with a 120 wait at each. Depending on whether we visit V1 or V2 first, the pSTN is comprised of different uncertain durations, as seen in b) and c).
be checked before the path is determined.

This motivates a reformulation of the original problem into two subproblems. This allows us to adopt a generate and test procedure. Candidate paths are chosen with an upper level generator, based on that outlined in Chapter 3. The pSTP implied by each candidate path is then checked for its probability of temporal consistency. If a schedule can be found for the candidate pSTP, then the schedule is returned with the path as a solution. Otherwise, the candidate path generator finds the next best candidate solution. I examine the partition of the problem more closely in the following section.

3.4 Two-stage solution

Note from above that the underlying pSTN of the temporal problem depends on the solution of the spatial problem, due to the inclusion of probabilistic durations reflecting traversal. However, the temporal assignments to the controllable events do not affect path planning. This motivates a decomposition of the problem into two stages, the Spatial problem and the Temporal problem below.

Given the underlying roadmap, I define the spatial problem:

Problem 4 (Spatial problem). Given \( \langle V, \hat{V}, E, C, B, \Delta_E \rangle \) find \( V^* \),

\[
V^* = \arg \min_V f(V)
\]

subject to

- \( v \) appears at least once in \( V^* \), for every \( v \in \hat{V} \)
- \( E \) is the sequence of all \( e_i \in E \) connecting \( v_i, v_{i+1} \) for all \( v_i, v_{i+1} \in V \)
- \( f(V) = \sum_{e_i \in E} c_{e_i} \)
- \( P(\bigcup_{e_i \in E} \{b_{e_i} = 0\}) < \Delta_E \)
The spatial problem captures the chance-constrained path planning aspect of the problem. The agent chooses the path which visits the goal locations, minimising an objective function, while remaining safe with respect to the cumulated risk along traversal.

Similarly, I define the temporal problem:

**Problem 5** (Temporal problem). Given \((V^*, R_d, R_c, \Delta_T)\) find \(X_b\), subject to

- \(X_e\) constructed from \(V^*\) and set of random durations \(R_d\)
- \(P(\bigcup_{c \in C_e} \{x_{c,i} - x_{c,j} \notin A_c\}) < \Delta_T\)

For each candidate path, we infer the implied temporal constraints, and combine these with the temporal constraints specified by the user. This specifies the pSTP for each candidate path, the corresponding temporal problem.

With the above decomposition of the problem we can propose Algorithm 1 as a method to return an optimal solution to Problem 3.

**Algorithm 1:** Two stage solution to Problem 3

**Input:** \((V, \hat{V}, E, C, B, X_b, R_d, R_c, A_E, \Delta_T)\)

**Output:** \(X_b, V\)

1. Construct Problem 4 with \((V, \hat{V}, E, C, B, A_E, \Delta_T)\);

2. while true do

3. \(V \leftarrow \) next best solution to Problem 4;

4. Construct Problem 3 with \((V, R_d, R_c, \Delta_T)\) ;

5. if \(\exists X_b \) solution to Problem 5 then

6. \[\] return \(X_b, V\);

7. if no other solutions to Problem 4 exist then

8. \[\] return Failure;

**Theorem 2.** Assuming soundness and completeness of solution methods to Problem 4 and 5, Algorithm 1 returns an optimal solution to Problem 3 if a solution exists. Otherwise, algorithm returns failure.
Proof. Note first that, assuming non-zero probability of edge failure for every edge, the number of solutions to the spatial component, Problem 4, is finite, given the chance-constraint. Hence Algorithm 1 always returns.

Suppose no solution exists. Then, there are two cases: either the set of solutions to Problem 4 is empty, in which case failure is trivially returned; or for no solution of Problem 4 does there exist a Problem 5 which can be solved, in which case the algorithm will exhaust all solutions to Problem 4 and return failure. Hence if no solution exists, algorithm returns failure.

Suppose a solution exists and some $X_b, V$ is returned. Given the algorithm, if $X_b, V$ do not constitute an optimal solution, then either: a) $V$ is not a solution to Problem 4; b) $X_b$ is not a solution to Problem 5 constructed with $V$; or c) $\exists X'_b, V'$ solution such that $f(V') < f(V')$. Cases a) and b) are false by construction of the algorithm. For case c), if there exist some $V'$ better than $V$, then it would have been considered by the algorithm before $V$, in which case $V'$ was not returned because no solution could be found for Problem 5 constructed with $V'$.

Given the above result, we now need to find solution methods for Problems 4 and 5, which I will refer to as respectively the spatial and temporal subproblems. I now develop the relationship between the algorithms to be described in this thesis and the spatial and temporal subproblems. This development allows us to construct the architecture for solving the chance-constrained probabilistic travelling salesman with temporal constraints. I present here only a high level overview of the algorithms. For details of the spatial subproblem, please refer to Chapter 4. For details of the temporal subproblem, please refer to Chapter 5.

Note first that the spatial subproblem is as formulated in Chapter 4. Candidate paths (Line 3 of Algorithm 1) may be found by calling the chance-constrained roadmap planner described in Algorithm 2. Similarly, the temporal problem is as formulated in Chapter 5. For the pSTPs generated with each candidate path, we check the probability of temporal consistency (Line 5 of Algorithm 1) with the probabilistic temporal consistency checker described in Algorithm 3.

I thus examine two subproblems in order to solve the full problem. The archi-
Figure 3-2: An overview of the architecture for solution of the full problem, as described in Algorithm 1. The first stage generator uses the chance-constrained roadmap planner to enumerate candidate paths in best first order. The second stage test attempts to find a schedule to the pSTP using the probabilistic scheduler.

architecture is given in Figure 3-2. Intuitively, the planner is given the roadmap, the list of goals, the starting location, a set of user defined temporal constraints, and the chance-constraints on safe traversal and temporal consistency.

The first stage generator, the chance-constrained roadmap planner, searches over the set of solutions subject to the chance constraint on traversal. Whenever a path is found which visits all of the goal locations, the implied temporal constraints due to traversal are added to the set of temporal constraints specified by the user to specify a pSTP, the pSTP corresponding to the candidate path.

We then check for a schedule to the pSTP subject to the chance constraint on temporal consistency. If a schedule is found, the schedule is returned with the candidate path as a solution to the full problem. If no solution is found, the scheduler signals the roadmap planner to continue the search, and the procedure is returned for the next best candidate path.
3.5 Conclusion

In this chapter, I have motivated and stated the chance-constrained path-planning problem with temporal constraints. A previous model predictive control approach has been shown to be unsuitable with respect to the mission length spatial subproblem, because its solutions require decisions and constraints to be checked for each time step of the mission. Instead, I have briefly outlined the reasons for a roadmap approach to path planning. Further, by noting the dependence of timing constraints on the path planning aspect of the problem, I have devised a two stage generate and test architecture for the full problem, and outlined the roles of the spatial and temporal components.

In the next two chapters, I will develop in detail the chance-constrained roadmap planner and the chance-constrained scheduler. I will then consider the interactions and analyse the numerical performance of the approach in solving the full problem.
Chapter 4

Chance-constrained roadmap planning

In this Chapter I consider the subproblem of robust path planning under uncertainty. Recall that I reduce the chance-constrained path planning problem with probabilistic durations into a chance-constrained path planner which generates candidate solutions, and a chance-constrained scheduler which verifies the probabilistic consistency of the solutions with respect to temporal constraints. I motivate, define and provide a roadmap solution to the chance-constrained path planning problem as a necessary step to development of the candidate generating algorithm.

4.1 Problem overview

The path planning component is developed in support of the overall problem. As the generator in a generate and test approach to a chance-constrained problem, the component must support risk sensitive path planning. Further, the planning algorithm must be able to work over mission-length horizons, whether the actual execution time is measured in minutes or days. These requirements guide my development of the chance-constrained roadmap path planner.

Deterministic assumptions are commonly made in planning, whether in potential field based methods [49], graph-searched based [16], or even in the more recent
sampling-based path planning literature [26, 19]. Such algorithms assume deterministic evolution of the vehicle state, in terms of pose and vehicle configuration. Further, such algorithms assume that there are areas which are strictly out of bounds for the vehicle, for example obstacle regions in which the vehicle is expected to fail. However, these approaches do not capture the stochastic nature inherent in vehicle motion, due to environmental conditions or failure of actuation components.

A body of work in the literature addresses the issue of path planning under uncertainty. For example, much of the early work in Markov Decision Processes (MDPs) and their partially-observable probabilistic extensions (POMDPs) has had an emphasis on robotic path planning, for example [41] and [11] respectively. Whereas the MDP and POMDP frameworks concentrated on maximising the expected utility of navigation, prior work has also examined path planning as utility maximisation with a bounded probability of failure [35], in what is known as chance-constrained path planning.

The chance-constrained methods have the following advantages. Firstly, the chance-constrained planner considers path planning as constrained optimisation, and so naturally lends itself to planning in the continuous domain. This is important because the control actions applied and the pose of the vehicle at any one time are often continuous in real-world scenarios. Secondly, the chance-constrained methods allow specifications of acceptable amounts of risk. Since the cost of a vehicle loss is essentially infinite, classical decision theoretic methods for balancing expected gain against expected loss do not work very well. Rather than trying to balance the possibility of failure against possible reward using a utility function, the chance-constrained method allows the user to specify an acceptable risk and proceed to devise a plan maximising return.

In this work I develop a path planner using the chance-constrained paradigm. In observation missions, the disincentive for failure, either due to loss of the vehicle or equipment, is extremely high. Thus, guarantees on the probability of safe execution would be desirable. Current state of the art algorithms include a family of discrete time model predictive control (MPC) methods [4, 6, 32, 35] which reformulate the
problem into constrained linear programs or mixed-integer linear programs. However, as such planners require time discretisation and make decisions for each time step, they scale exponentially with time. Though MPC methods are valuable for real time control planning, they are thus not appropriate for planning over the time scales necessary for observation mission. A resource-constrained roadmap based planner is thus developed in this work.

The key insight for the roadmap-based path planner is that we should distribute risk over the traversal of the map. Prior methods distribute risk over each time step as a particular way of dividing the risk over the traversals. In my formulation, I consider graphs in which vertices represent points on the map and edges represent feasible paths between them. There exists standard methods for estimating the risks incurred on the traversal of each edge, for example by the probability evaluation in the chance-constrained model predictive control works with standard Normal distributions, or lower bounds when only the mean and covariance are known [17], based on factors like the environment encountered and distance to obstacles. In this way, I am able to reduce a general chance-constrained path planning problem to chance-constrained roadmap planning. By searching over the graph rather than searching over controls for every time step in the mission, the algorithm is able to work over extended time scales, while guaranteeing the probability of success.

The current state of the art in roadmap path planning focuses on sampling based approaches. However, due to the development of such algorithms from rapidly-exploring random trees (RRTs), the focus is on single destination, single source path planning [28, 8]. In contrast, real world scenarios, such as those described in WHOI oceanography missions [45, 23] require measurements to be taken at multiple locations, and hence require single source, multiple destination planning capabilities. I thus focus my discussion on the multiple destination problem in this thesis.

4.1.1 Overview of the approach

I approach the problem as follows. I argue for a statement of the problem as a resource-constrained graph search, where the resource constrained is the risk accu-
Figure 4-1: A possible graph generated from the map. The light yellow region represents the feasible region, the blue blocks represent obstacles. The vertices in the graph are shown as small ovals, labelled 1 to 5, along with the start node S and the goal node G. The cost and risk associated with the traversal of each edge is indicated as (cost, risk).

I formally outline the equivalent resource-constrained problem, and present the solution algorithm of the reformulated problem. The equivalent problem requires a path comprising edges in the roadmap visiting a set of given goals in any order. I base my solution on the $A^*$ algorithm with multiple destinations, a well studied and efficient solution method for roadmap path planning. I make two extensions to the algorithm. I utilise an additive constraint representing the budget of risk, representing the probability of failure accumulated over the course of the mission. I also utilise a heuristic for guiding the chance-constrained $A^*$.

As the goal is to provide a chance-constrained path planner which generates candidate paths for missions, I shall evaluate the appropriateness of the devised algorithm for the motivating applications. I provide results regarding the scalability of the approach obtained through numerical experiments on problems inspired by real world AUV scenarios.
4.2 Roadmap for chance-constrained path planning

In addressing the spatial aspect of the robust mission planning problem, we assume that we have some graphical representation of the map, with undirected graph $G(E, V)$, where the vertices $V$ represents locations in the map, and edges $E$ represents feasible paths between locations. Further, we assume that we have a set of locations, $\hat{V} \subset V$ in the map which must be visited.

In addition, there exists a finite set $C$ of positive reals, representing the costs of traversal along an edge. For each edge $e$, the cost of traversal is given as $c_e$, for $c_e \in C$. Further we have $B$ set of random variables with Bernoulli distribution, for the success or failure of the vehicle while traversing along edges. Failure while traversing edge $e$ is represented by the outcome $b_e = 0$, for $b_e \in B$, while success if denoted by $b_e = 1$.

Suppose the user has specified that the risk of failure must be less than $\Delta_E$, the chance-constraint. The roadmap version of the chance-constrained roadmap problem can thus be stated:

**Problem 6** (Chance-constrained roadmap planning). Given $\langle V, \hat{V}, E, C, B, \Delta_E \rangle$ find $V^*$ a sequence of vertices to visit,

$$V^* = \arg \min_V f(V)$$

subject to

- $v$ appears at least once in $V^*$, for every $v \in \hat{V}$
- $E$ is the sequence of all $e_i \in E$ connecting $v_i, v_{i+1}$ for all $v_i, v_{i+1} \in V$
- $f(V) = \sum_{e_i \in E} c_{e_i}$
- $P(\bigcup_{e_i \in E} \{b_{e_i} = 0\}) < \Delta_E$

Given a graph defined with cost and risk functions, we require a path defined as a sequence of vertices $V$ containing all vertices in the goal list. Consecutive vertices must be connected in the roadmap. Further, the path must minimise the cumulative
cost of traversal, while making sure the probability of failure at any point during the traversal is less than $\Delta_E$ a user specified bound.

The focus of this thesis is not on the extraction of a roadmap from vehicle characteristics and the map. The interested reader is directed to the early efforts involving Voronoi diagrams [10, 42], or the more recent probabilistic roadmap [22, 21] and its derivatives, for example the rapidly-exploring random graph [20]. Given that a manoeuvre between two vertices in the graph is selected, we know the control inputs at each time step required to accomplish the traversal of the edge. Costs and risks of traversal can thus be constructed by evaluating the risk and cost of the control inputs with methods outlined in [32, 35, 27, 8].

In the following section, I consider how the stated problem may be remapped to a familiar resource-constrained traveling salesman problem.

### 4.3 Mapping to resource-constrained traveling salesman

Consider again the roadmap problem formulation in 6. Disregarding the chance constraint on probabilistic edge failure, we have the well-studied travelling salesman problem. However, the chance constraint is problematic. The original statement involves the evaluation of a probability over a disjunctive space. However, we can convert the problem to a conjunctive form by noting that

$$P \left( \bigcup_{e_i \in \mathcal{E}} \{b_{e_i} = 0\} \right) = 1 - P \left( \bigcap_{e_i \in \mathcal{E}} \{b_{e_i} = 1\} \right)$$

so the equivalent chance-constraint would require

$$P \left( \bigcap_{e_i \in \mathcal{E}} \{b_{e_i} = 1\} \right) > 1 - \Delta_E$$

However, the evaluation may still be difficult. In particular, allowing for correlated elements in $B$, we may need to wait for a complete path to be found to evaluate the
chance-constraint. This means we are not able to check consistency as we perform our search, so that we are unable to eliminate unpromising partial paths. We would prefer a measure of risk which we can calculate incrementally. Ideally, we would be able to calculate the risk as a cumulative sum, allowing us to make use of heuristics from literature in graph search.

We make the simplifying assumption that the distributions for edge failures are independent. This is a reasonable assumption in the context of single vehicle traversals. For example, for underwater surveys, failure typically comes about because of the vehicle hitting the bottom due to sudden changes in the depth of the sea floor. The probability that this will occur when traversing one segment of the survey path should be independent of the probability of failure at another path segment.

Note that the distributions for failure along traversal may be similar between two edges which are close together in the state space, because they are operating over similar bottom depth. Two edges close together may thus have similar probabilities of failure during traversal. However, this does not mean they are correlated. We demonstrate the distinction with the following thought experiment. We start by first noting that the risk of traversal is a characteristic of the path, and is independent of whether we are performing single vehicle planning or planning for two vehicles: if the vehicles have the same specifications, the probability of failure while traversing the same edge for each vehicle is the same. Now, consider if we sent two identical vehicles on traversals along two edges, $e_1$ and $e_2$, which are close together in space. If the failure probabilities are correlated, then if one vehicle fails along its traversal, the probability of the other vehicle would be affected as well. In the independent case, the failure of one vehicle has no bearing on whether the other vehicle fails.

Using the independence assumption, and taking the natural logarithm of both side of the chance constraint, we have an equivalent constraint $\sum_{e_i \in E} \ln P\left(\{b_{e_i} = 1\}\right) > \ln (1 - \Delta_{\mathcal{E}})$.

Assuming that the distributions are known, such that $p_{e_i} = P\left(\{b_{e_i} = 1\}\right)$, we have Problem 2 is equivalent to Problem 7.
Problem 7 (Equivalent spatial problem). Given \((V, \hat{V}, E, C, B, \Delta_E)\) find \(V^*\),

\[
V^* = \arg\min_v f(V)
\]

subject to

- \(v\) appears at least once in \(V^*\), for every \(v \in \hat{V}\)
- \(E\) is the sequence of all \(e_i \in E\) connecting \(v_i, v_{i+1}\) for all \(v_i, v_{i+1} \in V\)
- \(f(V) = \sum_{e_i \in E} c_{e_i}\)
- \(-\sum_{e_i \in E} \ln p_{e_i} < -\ln (1 - \Delta_E)\)

We recognise the statement of the chance constraint is now in the familiar form of a cumulative resource with an upper bound constraint. The problem is thus one of a travelling salesman with a resource constraint. We can leverage existing literature to generate candidate solutions using heuristics.

### 4.4 Resource-constrained search

In this section, I present a solution algorithm for the resource constrained travelling salesman. Inspired by the success of admissible heuristics in reducing the search space, I derive an admissible heuristic to guide the search over the roadmap. The result is a simple extension of A* to roadmap path planning with resource constraints, denoted risk sensitive A*.

Note first that the essence of the problem lies in minimising the cumulative cost, while making sure the accumulated risk is less than the bound \(\Delta_E\). However, through the use of an admissible heuristic, one which under estimates the cost to go, A* combines cost-so-far and cost-to-go to reorder the sequence by which partial paths are processed. I extend this to checking consistency with respect to the resource constraint, to provide lower bound estimates of the resource-to-go. The algorithm will thus be aware of partial paths guaranteed to exceed the risk bound, and discard such partial paths.
Suppose we can access an lower bound on the cost-to-go between two vertices. We may then use the admissible $h_{max}$ heuristic. Intuitively, given a set of goals and given a current vertex, the cost to go is at least the maximum cost-to-go to any one vertex. We thus require a method for lower bounding the pairwise cost-to-go for vertices. Note that additive heuristics are not admissible. For example, the cost-to-go from one vertex to a pair of goals may not be the sum of the cost-to-go to each goal, because one of the goals may have been visited en route to the other goal.

Note that we can lower bound the cost-to-go between any two vertices by running an all pairs shortest path (APSP) algorithm. The cost-to-go is a lower bound in resource-constrained graph search, because the APSP algorithm gives only the shortest path, and ignores consistency with respect to resource constraints.

We may make a similar argument for the heuristic of the risk-to-go, by considering risk as an alternative utility. Again, we may use the $h_{max}$ heuristic [7], maximising over a set of minimum risk-to-go from any vertex to any set of goal vertices. The minimum risk-to-go is lower bounded by running APSP over the graph with risk as utility. The resulting estimates are lower bounds, because the algorithm must optimise the cost of the path, and thus may choose to take riskier paths where appropriate.

We can thus use $h_{max}$ as an admissible heuristics for risk sensitive A*. A expand function, based on the original A* expand, is outlined in Function expandH. Correspondingly, we order the partial paths in the queue from those with the smallest estimated total cost, $CostMin$. Further, when the estimated lower bound on the total risk required, $RiskMin$, exceeds the risk bound, we discard the partial path. The modified algorithm is given in Algorithm 2.

The algorithm initialises by placing the start node as a partial path in the queue, with the list of all goals as the set of unreached goals. Each partial path maintains a list of all the vertices and edges visited on the partial path, with the latest at the head of the lists. At each iteration, the first partial path in the queue, ordered by the sum of cost-to-go and the cost-so-far, is checked. If the sum of the risk-to-go and the risk-so-far is less than the risk bound, and the list of unreached goals is empty, then the partial path is returned, and the path is backtracked to obtain the sequence of
Algorithm 2: Heuristic chance-constrained roadmap

```plaintext
input : start, goals, \( R_{\text{max}} \)
output: path
1 start \( \leftarrow \) initH (start,goals);
2 solved \( \leftarrow 0; \)
3 Q \( \leftarrow \) start;
4 while Q \( \neq \emptyset \) do
5 \( x \leftarrow \) pop(Q);
6 if \( x.RiskMin < R_{\text{max}} \) then
7 if \( x.Remaining = \emptyset \) then
8 \( solved \leftarrow 1; \)
9 break;
10 \( y \leftarrow \) expandH(\( x \));
11 Enqueue \( y \) into Q by ascending \( y.CostMin; \)
12 if solved = 1 then
13 \( \) path = backtrack(\( x \));
```

vertices and edges on the path. If the list of goals is not empty, and the lower bound on the total risk required until all goals are reached is less than the chance constraint, the partial path is expanded with expandH. If the partial path is not expected to yield a path to all the goals, while remaining consistent with the chance constraint, it is simply discarded.

Procedure initH(\( v \), goals)

```plaintext
input : \( v \), goals
output: \( x \)
1 \( x.Path \leftarrow \{ v \}; \)
2 \( x.Cost \leftarrow 0; \)
3 \( x.Risk \leftarrow 0; \)
4 \( x.CostMin \leftarrow 0; \)
5 \( x.RiskMin \leftarrow 0; \)
6 \( x.Remaining \leftarrow \) goals;
7 \( x.Head \leftarrow v; \)
```

In expandH, a partial path not containing all the goals is expanded. New instances of partial paths are made, with vertices reachable from the current head which have not been visited added as the heads of the new partial paths. For each new partial path, the cost-so-far and the risk-so-far are updated, and the minimum cost and risk
Function expandH(x)

input : x
output: y

1 for each edge \{x.Head, v\} such that v \notin x.Path do
2 \quad x'.Path \leftarrow \{v, x.Path\};
3 \quad x'.Head \leftarrow v;
4 \quad if x'.Head \in x'.Remaining then
5 \quad \quad x'.Remaining \leftarrow x'.Remaining\{x'.Head\};
6 \quad x'.Cost \leftarrow x.Cost + c(x.Head, v);
7 \quad x'.Risk \leftarrow x.Risk + r(x.Head, v);
8 \quad x'.CostMin \leftarrow x'.Cost + \arg\max_{v' \in x'.Remaining} CostToGo(v, v');
9 \quad x'.RiskMin \leftarrow x'.Risk + \arg\max_{v' \in x'.Remaining} RiskToGo(v, v');
10 \quad y \leftarrow y \cup x';

<table>
<thead>
<tr>
<th>Vertex</th>
<th>Cost to go</th>
<th>Risk to go</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>4.5</td>
<td>0.08</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>0.075</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0.055</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>0.095</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>0.125</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>0.03</td>
</tr>
<tr>
<td>G</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 4.1: The risk and cost heuristics for each vertex in the example problem.

of any solutions potentially extended from the new partial are calculated.

For a more in depth walkthrough of the algorithm, we consider the problem in Figure 4-1. Suppose we want to travel from S to G with risk less than 0.1. We can find the minimum cost-to-go and risk-to-go of each vertex using standard algorithms. These are listed in Table 4.1.

Initially, the partial path with only the start vertex is placed into the queue. The elements in queue at the end of iterations of the algorithm are given in Table 4.2.

4.5 Results

The performance of the algorithm was tested with simulated vehicle traversal problems defined over randomly generated road maps, inspired by a travelling AUV sce-
Table 4.2: Elements in the queue at the start of iterations of the risk aware graph search with heuristics. Note that the same plan was produced, but the maximum size of the queue and the number of iterations required was much smaller.

I consider the difficulty of solution with varying number of vertices in the roadmap, under varying chance-constraints. I analyse this difficulty in terms of computation time, as well as the number of solutions found.

The number of vertices in the generated roadmaps ranged from 5 to 25, as the typical number of available waypoints in an AUV mission. I generated 1680 roadmaps in total, 80 for each number of vertices. In the roadmaps, the vertices were labeled 0, 1, ..., n - 1 where n was the number of vertices in the roadmap.

In each generated roadmap, the undirected edges were generated as follows. Edges
were first inserted between consecutive vertices, to ensure that the roadmap was connected. Edges between non-consecutive vertices were randomly created. For each edge in the resulting roadmap, the cost incurred during traversal is sampled uniformly from the interval $[0, 1]$, and the risk along each edge was sampled uniformly from the range $[0, 0.05]$.

For each roadmap, problems were generated by requiring the traversals to 4 different goal locations from vertex 0, with varying chance-constraints along traversal. The goals were randomly chosen from among the set of vertices with labels in $1, \ldots, n - 1$, where $n$ is the number of vertices in the set.

The problems were solved using Algorithm 2 implemented using Allegro Lisp on a single core 3.4GHz Ubuntu virtual machine. For each problem, the solver was asked to provide optimal paths, with chance-constraints 0.05, 0.1, 0.2 and 0.4 respectively. The run times for each problem were restricted to 15 seconds.

The difficulty of the problems, in terms of results found, was demonstrated by the proportion of results found. The proportion of problems solved is given in Figure 4-2. In general, the lower the allowed risk, the less likely that a solution will be found. Very few paths are traversed before the total amount of risk becomes greater than that allowed. Similarly, with a greater number of vertices, a smaller proportion of the problems will be solvable. This is as expected, as the vehicle must potentially travel over a greater number of edges to reach all of its goals. This means that the same chance-constraint must be divided amongst a greater number of traversals, and hence feasibility becomes less likely.

Note however that the proportion of solutions derived with the 0.4 chance constraint is always lower than that of the 0.2 chance constraint. In fact, as the size of the roadmap increases, the number of solutions found with the 0.2 and 0.4 chance constraint both dip below that of the 0.1 chance constraint. This is because a large number of potential partial paths are pruned in problems with the lower chance-constraints, due to the use of the heuristics. This means the algorithm spends less time processing unpromising partial paths. For problems with higher chance constraints, more time is necessary to work through the list, and hence failure to find
solutions due to time out is more likely. This can be seen in Figure 4-3.

As expected, time outs also become more frequent with increasing number of vertices in the roadmap, again because there is a larger number of vertices to enqueue at each call to the function expandH. A higher number of vertices overall means a higher number of outgoing edges at each vertex, hence a higher branching factor during the search for a solution.

In general, the higher the number of nodes in the roadmap, and the higher the chance-constraint, the more time intensive the solution procedure will be. The time results varying chance constraints and size of the roadmap are given in Figure 4-4. Again, the higher branching factors, either because the higher chance constraint means fewer partial paths are discarded, or because of a larger roadmap, leads to higher processing times.

We thus note that the major contributions to computational time come from increased branching factor, due to a loose chance constraint or increasing number of vertices. However, while tight chance constraints allow quick return times, the actual number of solutions decreases dramatically, except when the problem is sufficiently large for time outs to occur.
Figure 4-3: The proportion of problems for which solution timed out for the chance-constrained travelling salesman problem, with 4 goals, varying chance-constraints and the size of the roadmap.

Figure 4-4: The times of solution of the chance-constrained travelling salesman problem, with 4 goals, varying chance-constraints and the size of the roadmap.
4.6 Conclusion

It is desirable for autonomous systems to restrict the probability of failure during traversal, either through entering dangerous regions, or through the execution of more risky manoeuvres. In this Chapter, I have shown how the problem of chance-constrained traversal may be mapped to that of a resource-constrained travelling salesman problem. I have proposed a solution technique which will solve such problems, and empirically validated the difficulty of solutions on roadmaps of size appropriate to typical AUV missions. The time scales at which solutions were derived were appropriate to the application, and by analysis of the numerical results I have shown the effectiveness of the heuristic in terms of reducing the branching factor of the problems. It remains now to provide an efficient method for solving risk aware scheduling, which will be explored in the next chapter.
Chapter 5

Probabilistic Simple Temporal Problem

In this chapter, I consider the problem of scheduling with probabilistic durations. Robust scheduling is essential to many autonomous systems and logistics tasks. Probabilistic formalisms quantify the risk of schedule failure, which is essential for mission critical applications. Probabilistic methods for solving temporal problems exist that attempt to minimize the probability of schedule failure. These methods are overly conservative, resulting in a loss in schedule utility. Chance constrained formalism address this problem by imposing bounds on risk, while maximizing utility subject to these risk bounds.

In this chapter I present the probabilistic Simple Temporal Network (pSTN), a probabilistic formalism for representing temporal problems with bounded risk and a utility over event timing. We introduce a constrained optimisation algorithm for pSTNs that achieves compactness and efficiency through a problem encoding in terms of a parameterised STNU and its reformulation as a parameterised STN. I demonstrate through a car sharing application that our chance-constrained approach runs in the same time as the previous probabilistic approach, yields solutions with utility improvements of at least 5% over previous arts, while guaranteeing operation within the specified risk bound.

I proceed in this chapter with definitions of temporal constraints and uncertain du-
rations, leading to a definition of the pSTN. I then formulate the chance-constrained pSTP (cc-pSTP). By reviewing the reductions provided in [48], I enumerate the constraints used in the nonlinear optimisation encoding of the cc-pSTP. I prove the soundness of the method, and provide empirical results to demonstrate the effectiveness of the algorithm.

5.1 Preliminaries

On a Woods Hole Oceanographic Institute mission, a vehicle may be required to sample a methane plume occurring at randomly distributed times of the day. Alternatively, when scheduling for a car sharing network, the uncertain durations of traversal through traffic should be considered when determining pickup times for cars. These are highly relevant examples of real world planning with durations outside the control of the agent. It is thus crucial for field deployable systems to deal robustly with uncertainty in timing when scheduling activities.

Temporal planning has been extensively studied in the operations research and artificial intelligence communities, with early works such as [3, 1, 44]. A Simple Temporal Network (STN) [13] is a variant of this earlier representational work that strikes an effective balance between tractability, expressiveness and simplicity of formulation. While there exists efficient methods for solving corresponding Simple Temporal Problems (STPs), such problems describe only deterministic applications, unable to capture uncertainty in timing for events.

The Simple Temporal Network with Uncertainty (STNU) extends the STN [48], introducing set bounded uncertainty. Events may be controllable, such that the timing can be scheduled. Events may also be uncontrollable, in which case there exists an unknown but set bounded difference between the timing of the uncontrollable event and the timing of an controllable event.

The STNU may be extended, e.g. [43], through probabilistic representation of the uncertainty. In a probabilistic extension, information regarding the distribution of uncontrollable events allows planning for outcomes which are more likely. However,
the probabilistic formulation should be revisited for two reasons:

1. The existing formulation takes a risk minimisation approach, leading to conservative solutions. In contrast, in real-world applications, it is common to accept a bound on the probability of failure and choose actions maximising an objective function; and

2. The existing formulation disallows constraints between two uncontrollable time-points. This is overly restrictive, for example in missions where a traversal with uncertain duration is required to observe natural phenomena with uncertain timing.

We introduce a chance-constrained probabilistic STP (cc-pSTP). Rather than requiring plans which would be consistent for all outcomes of the probabilistic durations, we optimise schedules for which the probability of failure can be bounded. By revisiting the probabilistic extension, we provide a solution method for the cc-pSTP, while addressing problems involving constraints between uncertain events. We present theoretical results for the soundness of the solution, as well as empirical results for scalability, and improved utility.

5.2 Definitions

We first define the constraints and variables which occur in a probabilistic STN (pSTN). Following conventions [48], we define the elements in a pSTN. The definition is measure-theoretic to allow discussion of stochasticity without worrying about the details of the distributions. In particular, the discussion is applicable both to uncontrollable durations with independent distributions, eg. the traversal time of a vehicle and the occurrence of natural phenomena, and those with joint distributions, eg. two vehicles traveling along the same route at the same time.

Definition 3. (pSTN constraints/variables)

- The activated time-points $b_i \in \mathbb{R}$ are those assigned by the agent;
• The received time-points $e_i \in \mathbb{R}$ are those assigned by the external world;

• A free constraint $c_{xy}$ (Free) is a constraint of type $(y - x) \in [l_{xy}, u_{xy}]$, where $x, y$ are time points; and

• Let $(\Omega, \mathcal{F}, P)$ be a probability space with sample space $\Omega$, $\sigma$-algebra $\mathcal{F}$ and measure $P$. An uncertain duration (uDn) $d_{xy} : \Omega \to \mathbb{R}$ is a random variable describing the difference $(y - x) = d_{xy}(\omega)$, where $y$ is a received time point and $x$ is an activated time point.

For ease of visualisation when demonstrating examples, we use the convention set out in Figure 5-1. Note that consistency with respect to a Free constraint is dependent on the outcomes of uDns in addition to the choice of activated time-points. The problem is thus initially a game against an uncooperative environment.

Given the definitions above, we now define the probabilistic STN:

**Definition 4. (Probabilistic STN)**

$N^+ = (X_b, X_e, R_c, R_d)$ defines a pSTN, with

- $X_b = [b_1, ..., b_B]$ the vector of $B \in \mathbb{N}$ activated time-points;

- $X_e = [e_1, ..., e_E]$ the vector of $E \in \mathbb{N}$ received time-points;

- $R_c = \{c_{ij1}, ..., c_{ijc}\}$ the set of $C \in \mathbb{N}$ Free constraints; and

- $R_d = [d_{ij1}, ..., d_{ijc}]$ the vector of $G \in \mathbb{N}$ uDns;

The formulation above is similar to that of [43]. The difference lies in the treatment of stochasticity. Previously, the received time points were random variables, with distributions conditioned on the activated time points. Instead, we characterise stochasticity by treating the probabilistic durations as random variables. By posing
the problem in terms of probabilistic durations, we are able to address problems in which there exist Frees between two received time-points. This was not possible with previous methods, even though such problems are prevalent. As an example, a pSTN of an oceanographic mission is given:

**Example 2.** (Example pSTN) An autonomous underwater vehicle (AUV) is tasked with taking water samples after an underwater volcanic eruption. Relative to the Start of Day at 8am, the timing of the eruption is estimated to be normally distributed, with mean 60 minutes and standard deviation 5 minutes. The vehicle departs from base to the volcano, and the traversal duration is normally distributed with mean 20 minutes and standard deviation 2 minutes. The vehicle must arrive after the eruption, but no more than 120 minutes after the event, as the chemicals diffuse in water.

The problem is encoded as a pSTN $N^+$, with:

- $X_b = [SoD, dep_{home}]$, $SoD$ the start of the day at 8am, and $dep$ the meeting time;
- $X_e = [arr, erupt]$, $arr$ the time of arrival at site, and $erupt$ the eruption time;
- $R_c = \{c_{erupt,arr}\}$, $c_{erupt,arr}$ requiring $arr - erupt \in [0,120]$ such that the vehicle arrives in time;
- $R_d = \{d_{dep,arr},d_{SoD,erupt}\}$, $d_{dep,arr} \sim N(20,4)$ the traversal duration, and $d_{SoD,erupt} \sim N(60,25)$ the eruption time relative to start of day;

We have defined the pSTN, a formalism useful for expressing a problem with probabilistic durations and temporal constraints. We proceed to develop how we may make use of the features offered by the pSTN.
For example, returning to the motivating problem of car sharing, we may also wish to schedule such that all reservations with the vehicle are completed as early as possible, so that maintenance could be performed at a depot. However, consider the effect of one late handover of the vehicle between reservations. The next user must adjust for the loss in allocated time, or risk returning the car late as well. A late return, a very specific instance of temporal inconsistency, may thus be considered a failure. We must thus schedule for the reservations to be completed as early as possible, while avoiding late handovers of the vehicle.

Note that, due to the possibly unbounded range of outcomes, we may not be able to schedule such that temporal consistency is guaranteed for all outcomes. The approach in [43] minimises the risk of temporal inconsistency. However, such an approach would not consider the time at which the vehicle is returned to the depot. Instead, we restrict the probability of late returns, and optimise the return time of the vehicle.

In general, we would wish to optimise the schedule with respect to an objective function, subject to bounds on the probability of temporal inconsistency: we try to make the best schedule, tolerating rare cases in which the schedule can not be met. This motivates a way to measure the riskiness of any schedule with respect to temporal constraints. We first define assignments to activated time points and the outcomes of uDns, so that we may discuss the idea of risk of temporal inconsistency rigorously.

**Definition 5. (Schedule and observations)** A schedule \( S_A \in \mathbb{R}^{|A|} \) is an assignment to a subset of activated time-points \( \{b_a\}_{a \in A} \in X_b, A \subseteq \{1, \ldots, B\} \). It is complete if \( A = \{1, \ldots, B\} \), and partial otherwise. We say a controllable event \( b_i \) has been scheduled by \( S_A \) if \( i \in A \).

Consider a sample \( \omega \) drawn from the joint sample space \( \Omega \). An observation \( O_A(\omega) = \{d_{ia,j_a}(\omega)\}_{a \in A}, A \subseteq \{1, \ldots, G\} \) is a tuple of uDn values for that sample. We say the observation is complete if \( A = \{1, \ldots, G\} \), and partial otherwise. We say a uDn \( d_{ia,j_a} \) has been observed in \( O_A \) if \( a \in A \).

Notice that, given a complete schedule and a complete observation, we can deter-
mine the value of every element in $X_e$ by the definition of uDns (since $e_j = d_{ij}(\omega)+b_i$). We can then check for consistency with respect to the set of Frees using $X_t = [X_b, X_e]$, a vector combining $X_b$ and $X_e$. Recall the definitions in Chapter 2. We see that with a correct choice of the state space and the constraints, we have well-defined failure and risk.

We show below that the outcome for the timing of all events can be described by a measurable function. This is important, as it allows us to reuse the general proofs in Chapter 2 to define the risk of temporal inconsistency.

**Lemma 1.** Let the $X_t$ be the vector $[X_b, X_e]$. There exists a measurable function $f : \mathbb{R}^B \times \mathbb{R}^G \rightarrow \mathbb{R}^B \times \mathbb{R}^G$, such that $X = f(X_b, R_d)$.

Further, the Frees can be written as tuples $\langle f_{cy}, A_{xy} \rangle$, requiring $f_{cy}(X) \in A_{xy}$, where $f_{cy}$ measurable, and $A_{xy}$ a measurable set.

**Proof.** Note that $X_b$ is a vector of controllable variables, the values of which can be assigned from $\mathbb{R}^B$, and that $R_d$ is a set of random variables with values in $\mathbb{R}^G$. Then we know that there exists $f : \mathbb{R}^B \times \mathbb{R}^G \rightarrow \mathbb{R}^B \times \mathbb{R}^G$, such that $X = f(X_b, R_d)$.

Each state variable is either just one of the controllable variables, or the sum of one of the controllable variables with one of the random variables. Note that $f$ is also measurable because it is linear.

Let the state $X$, the times of all controllable and uncontrollable events, be some element in $\mathbb{R}^B \times \mathbb{R}^G$. Then, let $R_c$ be the set of constraints, such that each $c_{xy} \in R_c$ is a tuple $\langle f_{cy}, A_{xy} \rangle$, where $f_{cy}(X) = y - x$ and $A_{xy}$ is the interval $[l_{xy}, u_{xy}]$. Each $f_{cy}$ is measurable, since it is just a linear function, and each $A_{ij}$ is measurable since it is a closed interval.

I reiterate that the above is important. The property inherited from STPs, that of linear constraints on the timing of events, with closed intervals, allows us to define the risk of any offline schedule not meeting free constraints. We can proceed to define risk of schedules as:

**Definition 6.** ($pSTP$ schedule risk) Consider $pSTN N^+ = (X_b, X_e, R_c, R_d)$. 

59
The risk of $S$ with respect to some subset of the constraints $A_m \subseteq R_c$ is:

$$r_{A_m}(S) = P(\Omega_{A_m,S})$$

where $\Omega_{A_m,S} \subseteq \Omega$ is the subset of the sample space such that for every $c_{ij} \in A_m$, if $\omega \in \Omega$ and $x_j(\omega) - x_i(\omega) \notin [l_{ij}, u_{ij}]$, then $\omega \in \Omega_{A_m,S}$.

Intuitively, for any schedule of times which can be chosen by the user, the risk of the schedule is defined as the proportion of outcomes of the probabilistic durations which result in at least one free constraint being violated.

Since I have shown that the pSTN is a form of the general problem defined in Chapter 2 satisfying assumptions, we can use the definitions of failure and risk as previously defined with assurances that the probability theoretic details are in order. Thus we can be assured that it makes sense to talk about how much risk is incurred by adopting a schedule.

**Theorem 3.** The above definition of risk for pSTN is well defined.

**Proof.** From Lemma 1, we know that:

- The full set of timings for events can be determined as a measurable function of some controllable variables, and random variables;

- The constraints are defined with measurable functions, mapping measurable sets.

These satisfy the assumptions made in Section 2.2. As the above definition of risk mirrors 2, we may use Theorem 1 to show that our notion of risk for pSTP is well defined. 

Having thus defined the pSTN elements, as well as the risk associated with any schedule, we proceed to discuss our problem formulation for chance-constrained scheduling.
5.3 Problem statement

In this section, I define the problem of a chance constrained probabilistic STP (CC-pSTP). Intuitively, we wish to perform scheduling such that an objective function is maximised while the probability of temporal inconsistency is bound above by some user defined value. This approach allows us to have some quantitative guarantee on the success of any schedule, despite uncertainty in timing. The guarantee on the probability of success is important in applications where there is a large penalty on failure that is difficult to encode as an objective function.

We define chance-constrained pSTP (cc-pSTP), a problem formulation making use of the pSTN representation. In solving a cc-pSTP, we wish to find a single schedule which optimises the timing of events with respect to an objective function, while bounding the probability of temporal inconsistency. Example functions can be found in optimal scheduling, e.g. deterministic [24] and under set-bounded uncertainty [38].

Definition 7. (Chance-constrained pSTP) We define a cc-pSTP as follows:

Given:
- \( N^+ = (X_b, X_e, R_c, R_d) \), a pSTN;
- \( \Delta_T \in [0, 1] \), an upper bound on the risk of failure, for the set \( R_c \) of Frees; and
- \( V : \mathbb{R}^B \rightarrow \mathbb{R} \), an objective function dependent on assignments to \( X_b \).

Find:
- \( S_B^* \in \mathbb{R}^B \), a schedule of \( X_b \) minimising \( V \);

Subject to:
- \( r_{R_c}(S_B^*) < \Delta_T \), the probability of inconsistency bounded by \( \Delta_T \).

An example cc-pSTP for the AUV scenario is given below:

Example 3. (Example cc-pSTP) Recall Example 2, with \( N^+ \) as before. The departure should be as early as possible, to ensure availability for later missions. We thus set
Figure 5-2: An example demonstrating the challenge involved in evaluating the risk of failure. For any schedule, we can characterise the subset of the sample space for which the schedule is inconsistent, but evaluating the measure of the subset is non-trivial.

**objective V = dep, to minimise the time of departure. However, it should neither arrive early nor be more than 2 hours late. The scientist has required the schedule to satisfy both constraints in at least 99% of cases. Thus we set Δ_T to be 0.01. This specifies a cc-pSTP for the water sampling mission.**

In general, we may use arbitrary distributions for the uDns, as long as we are able to evaluate the cumulative distribution functions (cdfs) for each uDn. We do not require information on the joint distribution. This will be elaborated when we discuss the solution method. In this paper, we assumed continuously differentiable cdfs (e.g. normal, exponential etc.), allowing the use of commercial solvers.

The key challenge in solving the above problem involves the evaluation of \( r_{R_e} \), the risk of failure given each complete schedule. In particular, to evaluate \( r_{R_e}(S) \) for complete schedule \( S \), we need to both find the subset in the joint sample space for which \( S \) fails, and then evaluate the measure of the subset. We demonstrate the challenge with a simple example.

**Example 4.** Suppose we have the pSTN in Figure 5-2. By inspection, the measure of the set for which the temporal constraints are feasible should be independent of the choice of the only controllable variable, the timing of the start event does not impact the probability of failure.

Suppose we assign start = 0, so that the timing of the received time points take on the values of the uncertain durations. Figure 5-3 shows the range of times for the received time points which result in temporal consistency. Note that depending on the distributions.
Figure 5-3: The plot shows part of the region in the space of the received time points which results in temporal consistency.

We may note that we can write:

\[ P(\Omega_{R,c,s}) = \int_{\Omega} 1_{\Omega_{R,c,s}} d\mu \]

where \( 1_{\Omega_{R,c,s}} \) is the indicator function for the subset \( \Omega_{R,c,s} \). This may be used to sketch out a Monte Carlo method for determining the size of this measure, by generating samples from the joint sample space, then evaluating the indicator function by checking consistency. By using the strong law of large numbers, we can get probabilistic bounds on error in approximation as a function of the number of samples required.

However, we note that the MC method for evaluating the measure of the subset is reliant on the number of samples. Further, it does not provide a lower bound of the true probability of success, but converges to the true probability in the limit. This motivates us to consider the concept of strong controllability to pose an approximate problem with tractable solutions, where any solution to the approximation would be a solution to the CC-pSTP.

We have thus formulated a problem involving probabilistic durations, in which we attempt to schedule events to optimise an objective function while providing proba-
bilistic guarantees on the satisfaction of free constraints. We show how the cc-pSTP may be solved in the next section.

5.4 Solution of cc-pSTPs

Given the problem statement in the previous section, I explain the solution method for cc-pSTPs. Intuitively, we choose the most probable set of outcomes of the uDns, and schedule activated time points such that the timing constraints are satisfied for any combination of outcomes in this restricted set. The key insight behind our approach is that by distributing the allowable risk amongst the uncertain durations uDns, we can set-bound the outcomes. This allows us to convert a pSTN into a simple temporal network with uncertainty (STNU), a well studied structure. We can then make use of results for STNUs to reframe a cc-pSTP as a convex constrained optimisation problem solvable with standard optimisers.

5.4.1 Approach

We outline and motivate our approach before developing the details of our algorithm. We begin by noting that a cc-pSTP is an instance of a stochastic optimisation problem. A family of methods have been developed that reformulate the stochastic problem to a deterministic problem, by converting the stochastic constraints into deterministic constraints.

For example, optimising controls for stochastic dynamical systems involves bounding the extent of deviations from the mean, either by distributing the risk evenly [46] or by optimising the distribution of risk [35]. The process results in bounds on the state of the system, leading to reformulations as deterministic constraint optimisation problems.

We choose to draw inspiration from this literature because these methods can make probabilistic guarantees. Alternatives include sampling-based methods in the same vein as the Pegasus POMDP method [31], using particles to simulate the effects of disturbances and initial uncertainty with applications in robotics [5]. However,
the assessment of risk in such methods only \textit{converges in the limit} as the number of particles increase - they can not guarantee bounds on the probability of failure.

We further note that the temporal reasoning community has developed an analogous set bounded approach that leverages the structure of simple temporal problems, the STNU.

The STNU is similar to the pSTN. Uncertainty is represented with set-bounded \textbf{contingent durations} \((Ctg) g_{xy} \in [l_{xy}, u_{xy}] \). These are differentiated from Frees because they are \textit{variables} with values \textit{assigned by the environment}. Ctg describe the relationship of \((y - x)\), where \(y\) is a received time point and \(x\) is an activated time point. A STNU is defined as \(N = (X_b, X_e, R_c, R_g)\) with \(X_b, X_e, R_c\) as in pSTN and \(R_g\) the set of Ctg's such that \(R_g = \{g_{i1j1}, \ldots, g_{iGjG}\}\) for some \(G \in \mathbb{N}\).

A rich set of methods exist for offline, online and incremental solutions to STNU controllability, e.g. \([48, 29, 12]\) as well as the disjunctive extensions \([36, 47]\). By mapping underlying pSTNs into STNUs, following the paradigm in robust stochastic programming, we leverage the literature on offline robust scheduling of STNUs. We do so by distributing the allowable risk. This allows us to adjust the bounds on the uDns for better solutions.

\subsection{Definition of strong controllability}

In our approach, we make use of the idea of \textit{strong controllability} for STNUs \([48]\). Informally, if a STNU is strongly controllable, then there exists an schedule which is consistent for all outcomes of the contingent durations. Formally:

\textbf{Definition 8 (Strong Controllability).} Consider STNU \(N = (X_b, X_e, R_c, R_g)\). Let \(\Omega_g = [l_{i1j1}, u_{i1j1}] \times [l_{i2j2}, u_{i2j2}] \times \ldots \times [l_{iGjG}, u_{iGjG}]\), the space of possible values for the set of contingent durations Ctg's in \(R_g\).

We say that \(N\) is \textbf{strongly controllable} if there exists a schedule \(S_B\) such that for any \(\omega_g \in \Omega_g\), all constraints in \(R_c\) are satisfied.

Intuitively, this means that a STPU is strongly controllable if there exists a schedule, so that, for any combination of outcomes of the uncontrollable durations, the set
of free constraints are satisfied. There is thus at least one schedule which will always "work".

I adopt strong controllability as a step towards a conservative solution to our pSTP, partly because the definition makes it easy to prove the soundness of our method, and partly because the check for strong controllability is in the form of a proof by construction. I now proceed to outline the reasoning behind the checking algorithm.

5.4.3 Strong controllability as constraint satisfaction

The check for strong controllability outlined in [48] is a proof by construction: the algorithm attempts to construct a schedule which will work for all possible values for the uncertain durations. This is especially powerful, as we will not only be able to determine whether the problem is strongly controllable, but also obtain a schedule which will work under all circumstances if the problem is strongly controllable. The key insight which allowed this is the fact that we can replace constraints involving Ctgs with Frees. This turns the STPU into a STP, for which polynomial solution methods are known. Before outlining the rationale, we may walk through an example.

Example 5. Consider the STPU as seen in Figure 5-4. We have two activated time points, and one received time point. The received time point \( t_\omega \) occurs in the interval \([c, d]\) after the execution of \( t_0 \), and we must schedule \( t_1 \) and \( t_0 \) such that \( t_1 \) occurs between \([a, b]\) before \( t_\omega \).

Let us consider the constraints between \( t_0 \) and \( t_1 \), and see if we can obtain some constraint \( t_0 - t_1 \in [a', b'] \). As a first step, observe that \( t_\omega \) may be written \( t_0 + \omega \),
where $\omega$ is some value between $[c, d]$ assigned by the environment. Then, we can write the constraint between $t_\omega$ as

$$t_0 + \omega - t_1 \in [a, b]$$

Consider first the lower bound, that is, $t_0 + \omega - t_1 \geq a$. Note that, this is guaranteed for all values of $\omega \in [c, d]$ iff $t_0 - t_1 \geq a - c$. Consider now the upper bound, $t_0 + \omega - t_1 \leq b$, or equivalently $t_1 - t_0 - \omega \geq -b$. Note that this is guaranteed iff $t_1 - t_0 \geq -b - d$. We have thus found a set of constraints equivalent to those in the original problem, and we may replace the original constraints with the new constraint, taking away the Ctg. This conveniently leaves us with a STP, shown on the right in Figure 5-4.

The above demonstrates how we may reduce away the Ctg constraints in an STPU for strong controllability. The process can be generalised, such that any Frees to a received time point may be replaced by Frees between activated durations. The new constraints allow us to reason without dealing directly with the received time points. The problem is reduced from a game against an uncooperative environment to a standard constrained optimisation problem solvable with readily available packages.

We repeat the general reductions from [48] to develop our algorithm. Consider STNU $\mathcal{N} = (X_b, X_e, R_c, R_g)$. Let $R^- \subseteq R_c$ be the set of Frees involving a received time point. These are the only constraints which depend on variables not controlled by the agent, and thus need to be reframed.

We consider the lower and upper bounds separately and first perform our analysis for lower bounds. For each lower bound $c \in R^-_c$, we obtain one of three cases in Figure 5-5:

**Case 1** (only the end timepoint is a received time point) $t_i + \omega_i - t_j \geq a$

**Case 2** (only the start timepoint is a received time point) $t_i - t_j - \omega_j \geq a$

**Case 3** (both start and end are received time points) $t_i + \omega_i - t_j - \omega_j \geq a$

where $\omega_i$ is the duration of the Ctg starting at activated time point $t_i$ and bounded by $[l_i, u_i]$, with $\omega_j$ similarly defined.
In each case, we replace the original Free with a new Free:

**Case 1** \( t_i - t_j \geq a - l_i \)

**Case 2** \( t_i - t_j \geq a + u_j \)

**Case 3** \( t_i - t_j \geq a - l_i + u_j \)

The corresponding reductions for the upper bounds can be written similarly by multiplying both sides by -1.

In this way, a STPU may be rewritten with only Free links between activated time points. We may discard the Ctgs because the constraints are already encoded by the reformulation outlined above. We may thus obtain a STP with only the Free constraints and the activated time points. A consistent solution to the STP will be a schedule valid under any combination of outcomes for the Ctg, because the constraints derived from the Ctgs require consistency for all possible outcomes. More formally, let \( \hat{R}_c \) be the collection of Frees obtained from the reduction

**Theorem 4.** If there exists a schedule \( S_B \) satisfying all constraints in \( (R_c \setminus \hat{R}_c) \cup \hat{R}_c^c \), then the STNU is strongly controllable. Further, given any combination of outcomes for \( R_g \) the set of contingent durations, \( S_B \) is consistent with respect to all elements in \( R_c \) the set of Free constraints.

I state this result without proof, since it is not the main focus of this thesis. The interested reader may be referred to the more detailed exploration in [48]. However,
the above allows robust scheduling. If the uncertain durations are set-bounded such that the corresponding STNU is strongly controllable, then we can schedule the activated time points to be temporally consistent for every possible outcome of the uncertain durations.

Note now that consistency checking for STP is a well-understood process. Consistency may be checked by looking at the STP as a directed distance graph, and checking negative cycles using an all pair shortest path (APSP) algorithm. As there exists polynomial APSP algorithms such as Floyd-Warshall, strong controllability checking is polynomial.

5.4.4 Approximate CC-pSTP

While the above definition and the check by construction were proposed for STPU, the ideas leading to strong controllability are important to a CC-pSTP. Intuition can be gained by considering a toy CC-pSTP with \( N^+ = (X_b, X_e, R_c, R_d) \), required to have probability of being inconsistent being at most \( \Delta_T \in [0, 1] \).

Suppose we can find a set of bounds \( R_g \) on uncontrollable durations such that the STPU \( N = (X_b, X_e, R_c, R_g) \) is strongly controllable, and the probability of uDns falling outside \( R_g \) is at most \( \Delta_T \). That is,

\[
P \left( \{ \omega : d_{ij}^k (\omega) \notin g_{ij} \forall k \} \right) \leq \Delta_T
\]

for \( d_{ij}^k \) the uDns, and \( g_{ij} \) the bounds of the corresponding Ctgs.

Any schedule \( S \) which is a strong controllability solution with respect to \( N \) is a solution of the original CC-pSTP, since \( S \) is consistent for at least a subset of uDn values in \( R^G \) which occurs with probability at least \( 1 - \Delta_T \). The probability of failure would thus be less than \( \Delta_T \).

Given this intuition, we it is clear that we may define an approximate cc-pSTP.

**Definition 9. (Approximate CC-pSTP)**

*Given:*

- \( pSTP \ N^+ = (X_b, X_e, R_c, R_d) \); and
• Upper bound on the risk of failure, $\Delta_T \in [0,1]$

Find:

• complete schedule $S^*$;

• Set of Ctg $R_g$;

Subject to:

• $\mathcal{N} = (X_b, X_e, R_c, R_g)$ strongly controllable and $S^*$ a solution; and

• $\mu(\Omega_g) \leq \Delta_T$, where $\Omega_g = \{\omega \in \Omega | R_d(\omega) \notin A_g\}$ is the subset of the sample space for which the uDns are not in $A_g$ the product set of the intervals in $R_g$;

The key to this approach is the approximations of the subsets of the sample space which result in failure. Recall that the subsets of the sample space are pre-images of the outcomes of the uncertain durations. Again, the sets of outcomes of the uncertain durations which are inconsistent with a static schedule may be non-trivial. In our case, we approximate these with product sets of rectilinear regions for the outcomes of uncertain durations. The probability of the outcome being within these defined regions may be calculated using the cumulative density functions of the uncertain durations.

We will provide motivating examples for which, under some assumptions, there exists tractable solutions to the approximate problem. However, we first show that the approximate CC-pSTP yields solutions which are valid for the exact CC-pSTP, and demonstrate that the solutions may be conservative.

5.4.5 Soundness and incompleteness

We prove that a schedule $S^*$ obtained from solving the approximate CC-pSTP is also a solution to the exact CC-pSTP. The intuition behind the approach is as follows:

1. We restrict the uDns to bounded intervals, making them Ctg in a new STPU;

2. Suppose the STPU strongly controllable and $S^*$ is a valid schedule;

70
3. Suppose further that the subset has measure greater than $1 - \Delta_T$;

Then the schedule is consistent with probability at least $1 - \Delta_T$, hence the probability of failure is less than $\Delta_T$.

**Theorem 5.** Suppose the schedule $S^*$ and set of Ctg $R_g$ are the solutions to an approximate CC-pSTP with $N^* = (X_b, X_e, R_c, R_d)$ and chance constraint $\Delta_T \in [0, 1]$ for $R_c$, as given in Definition 9. Then $S^*$ is a solution to the corresponding exact CC-pSTPU as given in Definition 7.

**Proof.** Consider each chance constraint $\Delta_T$ for the Frees $R_c$. Let

$$\Omega_{R_c,S^*} = \{\omega \in \Omega | \exists c_{ij} \in R_c \text{ s.t. } x_j(\omega) - x_i(\omega) \notin [l_{ij}, u_{ij}]\}$$

and suppose $\omega \in \Omega_{R_c,S^*}$ (i.e. $R_d(\omega)$ in combination with $S^*$ does not satisfy $R_c$).

Since $S^*$ and set of Ctg $R_g$ were the solutions for the approximate problem, we have $S^*$ valid with respect to all samples $\{\omega'\}$ where $d_{ij}(\omega')$ satisfies $g_{ij} \forall d_{ij} \in R_d$.

Hence, $\omega$ is also an element of $\Omega_g$ the set of samples which give random durations outside intervals specified by $R_g$. As this is true for any $\omega$ thus defined,

$$\Omega_{R_c,S^*} \subseteq \Omega_g$$

Further, since $R_g$ is a solution to the problem in Definition 9, we have

$$\Delta_T \geq \mu(\Omega_g) \geq \mu(\Omega_{R_c,S^*})$$

Noting that $\mu(\Omega_{R_c,S^*})$ is exactly $\tau_{Rc}(S^*)$ the risk of schedule $S^*$, we thus have

$$\tau_{Rc}(S^*) < \Delta_T$$

as required by the original problem.

Note that in general we have introduced conservatism, as $\mu(\Omega_g) \geq \mu(\Omega_{R_c,S^*})$. The actual risk of the returned schedule is less than the probability of uDn values falling outside the returned intervals. The approximation is thus an incomplete solution, in that although a solution $S^*$ may exist with $\mu(\Omega_{R_c,S^*}) \geq \Delta_T$, $\mu(\Omega_g) \geq \Delta_T$ may not
be true. That is, a schedule which is consistent with the required probability may exist, but there may not be a way to bound the intervals to cover enough probability mass.

While we have shown that the solving the approximate problem yields a feasible solution to the exact problem, we have not yet described a solution method for the approximate problem. These are described in the next section, where we show that we can write the problem as a constraint satisfaction problem with convex constraints. We can thus use existing tools, such as SNOPT [18], to find a feasible solution.

5.4.6 Approximate solution as constraint satisfaction

Recall that strong controllability may be checked by writing it as a constraint satisfaction problem. We may thus leverage this work if we were able to map a pSTP to a STPU.

Recall that in a cc-pSTP, we have an upper bound for the probability of temporal inconsistency. We distribute this allowable risk over the set of uDns. For each uDn, we then consider a subset of its possible outcomes, according to the risk allocated, turning each uDn into a corresponding Ctg. This gives us a STNU to check for strong controllability.

For an intuition, start by considering the uDns in the pSTP. For any uDn $d$, we choose a set-bound for the outcome of the uDn, and treat it as a contingent constraint $g$. For example, consider uDn $d \sim N(2, 1)$. With 95% probability, the outcome lies in interval $[0, 4]$. If we found assignments to activated time points which be consistent for all outcomes in $[0, 4]$, we know the temporal constraints will be satisfied in 95% of cases if $d$ is the only uDn in the pSTN.

We may thus derive a system of constraints for a cc-pSTP as in Algorithm 3.

The first for-loop of Algorithm 3 adds two decision variables denoting lower and upper bounds for every uDn, and notes the cdf $F_{d_{xy}}$ associated with each uDn. The cdfs are used to calculate the probability mass lost by restricting the outcome of the uDns. Note that both the cdf evaluating the probability mass discarded by the lower bound and the complement cdf evaluating that discarded by the upper bound are
Algorithm 3: Approximating cc-pSTP

input : \( N^+ = (X_b, X_e, R_c, R_d) \)
output: \( X_r \) bounds on \( R_d \), \( F \) chance-constraint function, and \( \hat{R}_c \) reductions

1. \( X_r \leftarrow \emptyset, \hat{R}_c \leftarrow \emptyset \);
2. for each \( d_{xy} \in R_d \) do
   3. \( X_r \leftarrow [X_r; l_{xy}; u_{xy}] \);
   4. \( F \leftarrow [F; F_{d_{xy}}; 1 - F_{d_{xy}}] \);
5. for each \( c_{xy} \in R_c \) do
   6. \( a, b \leftarrow \) lower and upper bounds for \( c_{xy} \) respectively;
   7. if \( x \in X_e \land y \in X_e \) then
      8. Let \( d_{tx}, d_{ty} \) be uDns ending in \( x \) and \( y \), \([l_x, u_x] \) and \([l_y, u_y] \) corresponding bounds on uDns;
      9. \( \hat{R}_c \leftarrow \hat{R}_c \cup \{t_j - t_i + l_y - u_x \geq a\} \);
     10. \( \hat{R}_c \leftarrow \hat{R}_c \cup \{-t_j + t_i - u_y + l_x \geq -b\} \);
   11. else if \( y \in X_e \) then
      12. Let \( d_{ty} \) be uDn ending in \( y \), \([l_y, u_y] \) corresponding bounds on uDns;
      13. \( \hat{R}_c \leftarrow \hat{R}_c \cup \{t_j - x + l_y \geq a\} \);
      14. \( \hat{R}_c \leftarrow \hat{R}_c \cup \{-t_j + x - u_y \geq -b\} \);
   15. else if \( x \in X_e \) then
      16. Let \( d_{tx} \) be uDn ending in \( x \), \([l_x, u_x] \) corresponding bounds on uDns;
      17. \( \hat{R}_c \leftarrow \hat{R}_c \cup \{y - t_i - u_x \geq a\} \);
      18. \( \hat{R}_c \leftarrow \hat{R}_c \cup \{-y + t_i + l_x \geq -b\} \);

recorded.

In the second for-loop, the algorithm applies reductions to the uDns in the pSTN. The reductions are based on those for strong controllability. However, instead of fixed lower and upper bounds, the reductions performed allow the lower and upper bounds to be decided by the solver.

The reductions are similar to those proposed in [43]. The key innovation is accounting for free constraints between two received time points, disallowed in previous work. This innovation follows from our reformulation of the pSTN. By representing stochasticity with uncertain durations, we have a natural mapping from the STNU structure, allowing us to transcribe Case 3 in a probabilistic context.

We can now define the approximate cc-pSTP as constraint satisfaction with the new decision variables and constraints from Algorithm 3.
Definition 10. (Approximate cc-pSTP as constraint satisfaction) We solve an approximate cc-pSTP as a constraint satisfaction problem with:

Given:

- $N^+, \Delta_T \in [0, 1], \text{ and } V \text{ as in Definition 7}; \text{ and}$
- $X_r, F, \text{ and } \hat{R}_c$ from Algorithm 3

Find:

- $S_B^* \in \mathbb{R}^B$, schedule to $X_b$ minimising $V$; \text{ and}
- $LU \in \mathbb{R}^{2G}$ lower and upper bound values on uDns

Subject to:

- $\sum_{F_i \in F} F_i(LU_i) < \Delta_T, F_i \text{ and } LU_i \text{ the } i^{th} \text{ entries in } F \text{ and } LU \text{ respectively}; \text{ and}$
- $S_B^*$ and $LU$ satisfying constraints $(R_c \setminus R_c^-) \cup \hat{R}_c^-, R_c^- \text{ subset in } R_c \text{ involving received times}$

In the approximate cc-pSTP, we deal with the difficulty in determining the risk of a schedule present in the original cc-pSTP, by introducing the reductions from Algorithm 3. In addition to a schedule to the activated time points, we are required to choose the lower and upper bounds for the uDns. The choice of the schedule and the bounds must satisfy two sets of constraints:

- The chance constraint, ensuring the choice of bounds for uDns do not eliminate too much probability mass. By summing the probability mass beyond the bounds for each uDn, and applying the union bound, we bound the probability of any uDn evaluating to some value beyond the chosen bounds.

- The reduction constraints, enforcing strong controllability when uDn are restricted to set-bounded intervals. The constraints ensure that the bounds for uDns and the solution schedule is chosen together such that, for any outcome of uDns in the restricted intervals, the chosen schedule will be valid with respect to the requirement constraints.
Table 5.1: Solutions found for different parameters

<table>
<thead>
<tr>
<th>Risk</th>
<th>Even distribution solutions</th>
<th>cc-pSTP solutions</th>
<th>cc-pSTP solutions</th>
</tr>
</thead>
<tbody>
<tr>
<td>10%</td>
<td>0</td>
<td>0</td>
<td>143</td>
</tr>
<tr>
<td>20%</td>
<td>0</td>
<td>0</td>
<td>146</td>
</tr>
<tr>
<td>40%</td>
<td>28</td>
<td>0</td>
<td>151</td>
</tr>
<tr>
<td>Total number of Scenarios</td>
<td>428</td>
<td>230</td>
<td>165</td>
</tr>
</tbody>
</table>

Example 6. *(Approximate solution to cc-pSTP)* For Example 3, we find bounds for \(d_{dep,arr}\) and \(d_{SoD,erupt}\) as well as an assignment to dep such that the Free constraint is satisfied. By encoding the problem as above and solving with SNOPT [18], the bounds are respectively [14.421, 29.747] and [36.282, 72.196], such that the total probability mass discarded is 1%. The departure is accordingly scheduled for 57.775 minutes after the start of day.

We note that Boole’s inequality bounds the probability of failure, and thus some conservatism is introduced. The solutions are thus not guaranteed to be optimal. Even so, we show empirically that the approach still produces significant improvements in utility over the existing state of the art.

Note that the approximate cc-pSTP is in a form solvable with commercially available nonlinear solvers. The reductions comprise the majority of the constraints, and are linear in the decision variables \(S_B\) and \(LU\). The only possible source of nonlinearity is the chance-constraint. In our experiments, we used the SNOPT nonlinear solver [18], as it employs a sequential quadratic technique designed for nonlinear optimisation with a large number of linear constraints.

5.5 Experimental results

We benchmark our algorithms on scenarios inspired by car sharing, similar to Zipcars [9]. In each scenario we attempt to schedule for a 6 hour period, with the number of
cars ranging from 1 to 20, each with up to 5 users. For each user, up to three goal locations were generated based on a simplified open source map of Boston.

A pSTN was generated for each scenario. The traversal activities were modelled as normally distributed uncertain durations, with the means of uDns determined by length and speed limits of the roads taken, and standard deviations at 5% of the mean. A total of 1800 pSTNs were generated. To allow comparison to prior art, we ensured that no Free constraints existed between received time-points, as such problems are not handled by previous methods.

For each pSTN, we constructed three cc-pSTPs, with chance-constraints 10%, 20% and 40%. We want to complete all activities as soon as possible, and thus set the time of the last reservation as the objective function in each case.

The cc-pSTPs were solved by evenly distributing the risk, by our risk allocation method, and by the risk minimising method [43] for comparison. Solutions were obtained with SNOPT [18], a nonlinear optimisation solver designed for problems with a large number of linear constraints. Table 5.1 summarises the difficulty of finding solutions for different numbers of activities.

The proportion of solutions found drops off dramatically as the number of constraints increase. This is as expected, as the same amount of risk must be shared amongst a higher number of activities. There is thus a greater space of outcomes which must be covered.

Note that even distribution of risk results in almost no solutions. The risk minimisation method has the largest number of solutions: if a solution exists, it will be found regardless of the risk of the solution. The risk allocation methods find a comparable number of solutions because there is flexibility in how the uncertain durations are bounded, restricted only by the chance-constraint.

The soundness of solutions with respect to the chance-constraints were tested. For each, 50000 samples of the joint outcomes of the uDns were tested for consistency with Free constraints, given the assignments to activated times. Table 5.2 summarises results for the chance-constrained methods and the risk minimisation method. Note that the chance-constrained solutions were correct, whereas the variance of the risk
Table 5.2: Empirical verification of correctness of solution.

<table>
<thead>
<tr>
<th>Method</th>
<th>$P(\text{Success}) \ (\pm 1 - \sigma)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>10% cc-pSTP</td>
<td>0.9012 ± 0.0018</td>
</tr>
<tr>
<td>20% cc-pSTP</td>
<td>0.8059 ± 0.0051</td>
</tr>
<tr>
<td>40% cc-pSTP</td>
<td>0.6250 ± 0.0198</td>
</tr>
<tr>
<td>Min. Risk</td>
<td>0.9372 ± 0.1801</td>
</tr>
</tbody>
</table>

Figure 5-6: Computation time as a function of the number of activities in the scenarios.

minimisation method means no guarantees on the probability of success can be provided for its solutions.

The flip side of robustness is conservatism. For scenarios where solutions are found via the chance-constrained methods and the risk minimisation methods, we compare the utility of solution. On average, the 10%, 20%, and 40% cc-pSTP methods resulted in last activated time point occurring respectively 5.37%, 6.58% and 6.82% earlier than the risk-minimisation methods. These represent significant savings over the risk-minimisation method, which is too conservative in achieving robustness.

Tests were also performed on scalability, with results summarised in Figure 5-6. The runtimes for risk allocation cc-pSTP tend to blow up with the increasing number of constraints, although even problems with over 200 activities took less than 90 seconds with a 2.4GHz processor. The risk minimisation method is slightly faster than the cc-pSTP method, although the outliers take significantly longer.
The empirical validation confirms the soundness of the methods with respect to the chance constraint. Further, the results show that the methods do not blow up in time for relatively complicated problems. Lastly, it confirms that, by accepting varying levels of risk, we can derive better solutions than purely risk-averse behaviour.

5.6 Contributions

The ability to schedule a mission robust to uncertainty in the timing of events is crucial to schedulers which are deployable in the real world. While early efforts are able to perform scheduling, explicitly accounting for set bounded uncertainty of durations, this is not sufficient. In particular, there may be no way to finitely bound the uncertainty about the outcome of the durations. As an alternative, I proposed to schedule, not for all possible outcomes of the uncertain durations, but for only the most likely, based on probabilistic distributions of uncontrollable durations.

In this section, I made the following contributions to address the problem:

- I defined a formalism which allows characterisation of durations with probabilistic distributions;
- I proved that the formalism allows a well-defined measure theoretic notion of risk;
- I showed that while a full solution may be difficult, there exists an approximate solution based on Strong Controllability;
- I implemented the approximate solution as a constraint satisfaction problem, and benchmarked the performance.

We now have two chance-constrained algorithms, separately planning for the spatial and temporal aspects of the observation mission. In the next chapter, we return to the formulation of the joint spatial-temporal problem, and show how our algorithms may be combined to provide solutions scalable to real-world scenarios.
Chapter 6

Chance-constrained path-planning with temporal constraints: the results

Having obtained a high level overview of the approach in Chapter 3, and developed the generate and test components in Chapters 4 and 5 respectively, we are now in a position to consider the detailed workings of the architecture.

In the next section, I demonstrate the architecture with an example problem. I then follow up with a look at how the two components interact by analysing results from numerical experiments, concentrating on the time of solutions, and the difficulty of solution in terms of solutions found.

6.1 Example solution of overall problem

I illustrate the approach with the example problem in Figure 6-1. The basic temporal relations are given in Figure 6-2. Specifically, we wish to arrive at location 2 to make some observation of some physical phenomenon, which occurs at some time after the start of our mission with normal distribution, mean 10, covariance 0.25. We would like to arrive up to 10 time units before the phenomenon occurs, and leave after the observation. We would also like to visit locations 3 and 4, staying at 4 for at least 2
Edge 2-5 has $c = 1, p = 0.03$

$\begin{align*}
1 & \quad c = 6, p = 0.05 \\
2 & \quad c = 3, p = 0.01 \\
3 & \quad c = 5, p = 0.03 \\
4 & \quad c = 7, p = 0.005 \\
5 & \quad c = 4, p = 0.005 \\
6 & \quad c = 4, p = 0.005 \\
7 & \quad c = 11, p = 0.035
\end{align*}$

Figure 6-1: Example survey mission. The vehicle starts at location 1, and must visit 2, 3, and 4, subject to some temporal constraints. Edges are labeled with $c$ the cost of traversal and $p$ the probability of failure.

Figure 6-2: Temporal relationships which must always hold. The dashed lines indicate probabilistic durations, the solid lines represent constraints.
For our example, we assume that the traversal time along each edge is normally distributed, with standard deviation 5% of the mean. We let the cost of traversal along each edge be the expected time of traversal. Hence in our problem, we try to minimise the expected time of traversal. The chance constraints were $\Delta_E = 0.2$ and $\Delta_T = 0.1$.

The first candidate generated by the stage 1 was the path $\{1, 4, 6, 2, 7, 3\}$, with cost 19. The candidate path and corresponding pSTN is shown in Figure 6-3. The second stage checker was unable to find a full assignment to the controllable variables $dep_1, dep_2, dep_3, dep_4$ which satisfied the chance-constraint, because the probability of arriving at 2 before the $obs_2$ occurred was low.

The next candidate generated was path $\{1, 5, 2, 6, 4, 3\}$, with cost 20, summarised in Figure 6-4. The second stage was able to schedule the corresponding temporal problem with $dep_1, dep_2, dep_3, dep_4$ at 0, 10.9, 18.4, and 30.5 respectively.
6.2 Interaction of spatial and temporal problems

In this section, I analyse how the size of the spatial and temporal problems affects the difficulty of solution. Specifically, I consider how solution time and number of solutions change with the number of vertices in the roadmap, and the number of temporal constraints specified in the overall problem.

The roadmaps for each problem were generated as in the evaluations for Chapter 4. The vertices in the generated roadmaps were in the range \{5,10,15,25\}, as the typical number of available waypoints in an AUV mission. I generated 320 roadmaps in total, 80 for each number of vertices. In the roadmaps, the vertices were labeled \(0,1,\ldots,n-1\) where \(n\) was the number of vertices in the roadmap.

In each generated roadmap, the edges were again inserted between consecutive vertices, and edges between non-consecutive vertices were randomly created. For each edge in the resulting roadmap, the cost incurred during traversal is sampled uniformly from the interval \([0,1]\), and the risk along each edge was sampled uniformly from the
range $[0, 0.05]$. For each roadmap, problems were generated by requiring the traversals to 4 different goal locations from vertex 0, with varying chance-constraints along traversal. The goals were randomly chosen from among the set of vertices with labels in $1, \ldots, n - 1$, for $n$ the number of vertices in the set.

For each problem, I also generated a set of temporal constraints in addition to the roadmap. The number of temporal constraints were chosen from the set $\{1, \ldots, 4\}$. The Free constraints imposed limits on the difference between departure times from each goal location. For example, a generated constraint may ask the departure from vertex 1 be no later than 0.75 hours after departure from vertex 0. In addition, for each goal, I imposed the Free constraint that arrival at the goal must precede departure. For example, let the arrival time at $g$ be $arr_g$ and the departure be $dep_g$, then I require $dep_g - arr_g \in [0, \infty)$. The uncertain traversals were added into the temporal problem based on the candidate paths. For example, a partial path from goal vertex 1 to goal vertex 2, traversing along 2 edges with durations distributed according to $N(10, 0.5)$ and $N(11, 0.75)$ would have the distribution $N(21, 1.25)$ for the uncertain duration $arr_2 - dep_1$.

The problems were solved using Algorithm 1 implemented using Allegro Lisp on a single core 3.4GHz Ubuntu virtual machine. For each problem, the solver was asked to provide optimal paths, with chance-constraint 0.15. The run times for each problem were again restricted to 15 seconds.

Unsurprisingly, the time of solution increased as the number of vertices increased, and as the number of temporal constraints increased, as shown in Figure 6-5. The planner must search for candidate solutions over a larger space, hence requiring a longer time to arrive at any candidate solution. In addition, increased number of temporal constraints would be more difficult to satisfy, as they reduce the set of possible solutions based on the pSTPs each candidate solution generates. Thus, the algorithm would need to generate a greater number of solutions before one satisfying all temporal constraints is found.

Consider now the proportion of solution found, varying the number of vertices and
Figure 6-5: The solution times for the chance-constrained travelling salesman problem, with 4 goals, varying the number of temporal constraints and the size of the roadmap.

Figure 6-6: The proportion of problems for which solutions were found within the time limit, for the chance-constrained travelling salesman problem, with 4 goals, varying the number of temporal constraints and the size of the roadmap.
temporal constraints, summarised in Figure 6-6. Again, the difficulty of solutions generally increase with the number of constraints and the number of vertices, for the reasons outlined above. Interestingly, there was an increase in the proportion of solutions found for problems with 3 or 4 temporal constraints, as the number of vertices grew from 5 to 10. This is because a greater number of alternative paths were available, and hence there was a greater flexibility in choosing paths. This made meeting temporal constraint easier than if the number of possible candidate paths were small.

6.3 Conclusion

In this Chapter, I have provided a walkthrough example of how the overall problem may be solved, given the candidate path generator in Chapter 4, and the probabilistic scheduler in Chapter 5. I have also performed numerical experiments looking at how the number of temporal constraints and the number of roadmap vertices in a problem affect the time of solution, and the proportion of solutions found.
Chapter 7

Conclusion

My thesis has been motivated by the idea that when working with autonomous systems, humans should not be expected to adjust goals and specifications include margins of error. Rather, this should be handled by the planning and scheduling capabilities of the autonomous system. Focusing on such applications as routing advisory systems and oceanographic surveys conducted by WHOI, I have proposed the chance-constrained travelling salesman problem with probabilistic temporal constraints (cc-TSP-PTC).

Using a general notion of risk as the probability of failure, I have motivated and provided a rigorous definition of the cc-TSP-PTC, solvable via a generate and test framework. In particular, I generate solutions to the chance-constrained roadmap path planning problem and test the solution for probabilistic temporal consistency with a chance-constrained scheduler.

In support of research into the overall problem, I have developed a solution to the chance-constrained travelling salesman problem, by mapping to a resource-constrained travelling salesman problem. The roadmap planning algorithm optimises with respect to one cumulative cost, while providing solutions meeting an upper bound on a separate resource.

I have further developed a solution to the chance-constrained scheduling problem building on work on static scheduling for uncertain but set-bounded durations, and numerically tested the properties of the scheduler with respect to scalability and
conservatism involved.

7.1 Future work

While I have made progress into the problem of chance-constrained path-planning and scheduling, further research may be conducted.

The existence of a roadmap is one of the most fundamental assumptions made in the current approach. While I have discussed methods for producing the roadmap, further research should consider the possibility of an incremental chance-constrained multiple destination path planner. In my discussion of the roadmap approach, I have highlighted the weakness in RRT-based methods due to their inability to plan for multiple goals. The advantages of speed and any-time property is highly desirable, and justifies consideration.

The problem of chance-constrained scheduling is a rich topic going forward. While the current algorithm was based off static assignments, algorithms described in [39, 30, 29] derive dynamic scheduling policies for problems with set-bounded uncertainty. We may leverage off such literature to provide chance-constrained policies for scheduling, which would give us greater flexibility as we react to situations as they arise.

We may also increase the efficiency of the generate and test algorithm. In search, conflict-directed methods such as that outlined in [50] has reduced the number of operations required by directing search away from unpromising areas. By discovering conflicts while reasoning over infeasible candidates, the algorithm is able to mark out subsets of the search space. In our case, we would require a method of reasoning over infeasible candidates for the pSTP to produce conflict which we can apply to roadmap planning. The process behind this reasoning is not obvious at this point, and will require further investigation.
Bibliography


