A STACKELBERG SIGNALING GAME FOR CO-OPERATIVE RENDEZVOUS IN UNCERTAIN ENVIRONMENTS IN A SEARCH-AND-RESCUE CONTEXT

BY

VIJETH HEBBAR

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Aerospace Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2020

Urbana, Illinois

Adviser:

Professor Cédric Langbort

Abstract

Search and rescue (SAR) operations are challenging in the absence of a medium of communication between the rescuers and the rescuee. Natural signaling, grounded in rationality, can play a decisive role in achieving rapid and effective mitigation in such rescue scenarios. In this work, we model a particular rescue scenario as a modified asymmetric rendezvous game where limited communication capabilities are present between the two players. The scenario can be modelled as a co-operative Stackelberg Game where the rescuer acts as a leader in signaling his intent to the rescuee.

We present an efficient approach to obtain the optimal signaling policy, as well as its robust counterpart, when the topology of the rescue environment is unknown. We also analyse the sensitivity of the optimal signaling policy to the velocities of the two players as a further motivation for the robust solution. We observe that a completely robust approach in designing the signaling policy can lead to highly conservative solutions. To address this conservativeness, we then introduce a stochastic nature on the unknown topology and provide a signaling policy which probabilistic performance guarantees.

Table of Contents

List of Abbreviations

UAV	Unmanned Aerial Vehicle
SAR	Search and Rescue
HRI	Human Robot Interaction
DAG	Directed Acyclic Graph
FTG	Finite Time Guarantee
MC	Monte-Carlo

Chapter 1

Introduction

Unmanned Aerial Vehicle (UAV) usage in Search and Rescue (SAR) applications has been extensively studied in recent years. The primary challenge that is addressed by these UAVs in SAR scenarios is to quickly sweep large swaths of area with the goal of finding the rescuees and in certain situations, providing relief in the form of air-drops. In the absence of any communication, this problem is akin to a 'hide-and-seek' game of one player finding another in a known environment in the minimum time possible. Alpern and Gal (2003) discussed and studied various strategies for such co-operative rendezvous games between non-communicating players. These strategies take actions with the aim of minimising the expected time until rendezvous and assume complete absence of communication between the players during the game. On the other hand, having complete communication between the two players allows them to plan for a fixed rendezvous point and meet there.

We look at the more realistic situation arising between these two extremes, wherein there is limited communication between the UAV and the rescuee. In our work, we will look at the specific application of such limited communication capabilities in establishing some mode of co-ordination between the rescuee and the rescuer. A natural question when considering this scenario - What constitutes an 'intuitive signal' between the rescuee and the rescuer? It would be unreasonable to expect that the two agents in the scenario have a pre-established set of communication protocol that can be used to communicate effectively. For an answer to this question, we turn to a topic of rising interest in the Human Robot Interaction (HRI) community - 'legibility' and 'predictability' of robotic motion.

There has been increased interest in studying intent-expression and legibility of robotic motion in recent years. Dragan, Lee, et al. (2013) and Dragan, Bauman, et al. (2015) studied legible motion for robotic arms in settings involving human-robot collaboration. Szafir et al. (2014) studied the communicative ability of a UAV using modified trajectories, while in their later work (Szafir et al. (2015)) they looked at a more explicit medium of communication, using lights to convey directionality. Both these works illustrate the limited signaling capabilities that UAVs can exploit in the absence of formal communication channels.

In our work, we will not delve into the specifics of how such limited communication capabilities are realized but make reasonable assumptions on their existence. Specifically, we will assume there are certain 'targets goals' in our search and rescue topology that can be indicated using our signaling mechanism. We wish to see how such signaling capabilities can be exploited to influence the rescuee into taking certain actions which can mutually benefit both the agents. We proceed by formalizing the interaction between the rescuee and the rescuer as a game.

Game theory has been used extensively to model human-robot interactions in recent years Yua et al. (2018); Li et al. (2016). The signaling ability of the leader in a Stackelberg type game has been studied and exploited in applications like market structure (Etro (2013)) and security (Tambe (2011) and Rabinovich et al. (2015)). We show that an interaction between a human and an autonomous agent can also be modelled and studied in a similar framework. The autonomous agent (rescuer) acts as the leader and sends out a signal to the rescuee. It is assumed that the rescuer has ex-ante knowledge that the rescuee is observing this signal. Based on the received signal, the rescuee interprets the goal of the rescuer and takes an action, say, walk to the goal interpreted from the signal. Our work seeks to arrive at a signaling policy that the rescuer can implement to achieve its goal. To better motivate this framework we present below an illustration of a rescue scenario.

1.1 Motivating Illustration

Henceforth in our work, we will use the terms 'rescuee' and 'human' interchangeably. Likewise 'rescuer' and 'UAV' are also used interchangeably. As a very simple example, consider a hilly-terrain (Fig. 1.1) with two plains (red circles) to the east and west of the rescuee's initial location (blue circle). Assume that the rescuer believes that rescuee is aware of these two locations as well. A fixed-wing UAV flying in from an initial location (green circle) in the south can signal either of these locations as its intended target through its motion. The rescuee is initially at a location inaccessible to the UAV and the latter wishes to influence the rescuee to move to an alternate accessible location.

A key assumption we make is that the rescuer believes that the signal is observed and interpreted correctly by the rescuee; a reasonable assumption at that as the rescuee might expect the UAV to require a flat patch to land and can interpret the signal as an indication of the chosen landing spot. This assumption is key for the problem to be analyzed as a Stackelberg game. Note that it is not in-fact necessary for the UAV to reach these plains to land, but these plains are simply the 'target goals' we exploit to forward our signal. The UAV may choose some other accessible point along the path taken by the rescuee to rendezvous.

Both the rescuer and the rescuee are assumed to have constant velocities over the terrain. In moving across obstacles (like, hills and clouds) the players incur an increased path cost and thus, take more time to traverse. The constant velocity assumption allows us to work with the path cost and the travel time interchangeably. The rescuee will seek to reach the target in minimum time, or equivalently, minimise its path cost to the goal. The rescuer will try to minimise both the path cost for the rescuee and its own path cost to the point of rendezvous.

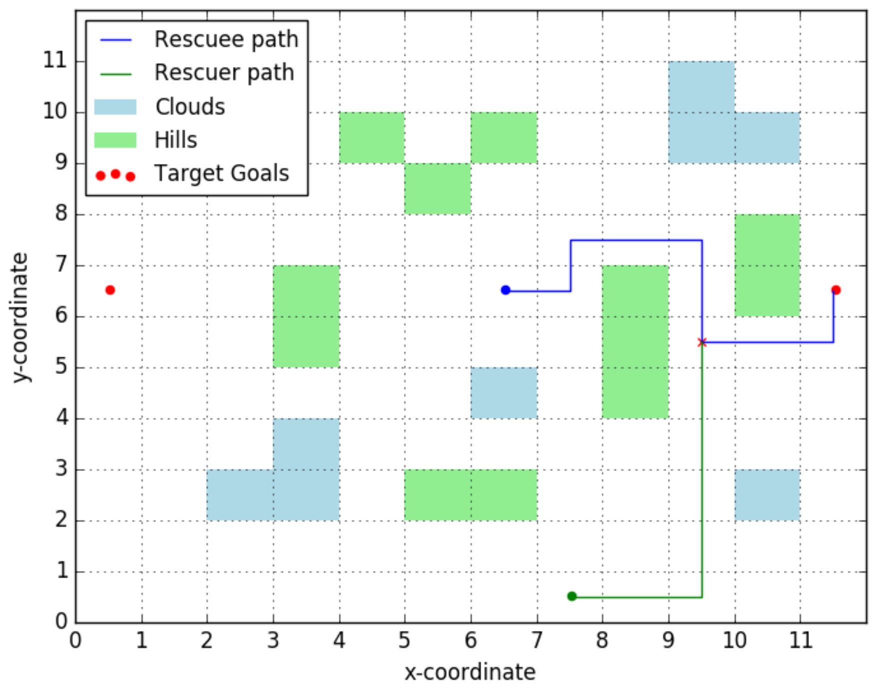


Figure 1.1: Rescue area topology. The two 'plain' regions (red dots) are used as target goals. Rescuee takes optimal path (blue) to perceived target of the rescuer. The rescuer picks an optimal rendezvous point (red cross) to meet the rescuee. Clouds (blue shading) act as obstacles to the UAV and hills (green shading) act as obstacles to the rescuee.

In other words, given a signal, the rescuee interprets the goal that the UAV is flying to and plans the shortest path to that goal. The rescuer, modelled as a rational player, assumes the human is going to behave as expected (i.e. take the shortest path to the perceived goal) and plans out its path to intercept the human's path. There are multiple implicit assumptions we have made in the game as presented above.

Assumption 1 *The ability of the rescuee to compute the shortest path relies on his knowledge of the precise terrain in the region under consideration. We make this assumption on his knowledge of the topology around him.*

Assumption 2 *The formulation presented above also relies on the rescuee's ability to precisely compute these shortest paths given the region topology. We assume that the rescuee possesses such abilities.*

Assumption 3 *We assume that the topology surrounding the rescuee is common knowledge to both the rescuee and the rescuer.*

Assumption 4 *Finally, the rescuer is assumed to have complete knowledge of the velocities of both the human and itself.*

Given these assumptions, we seek to obtain the optimal signaling policy that the rescuer should employ to achieve his goals. These are very strong assumptions and in the latter part of our work, we will relax these and obtain signaling policies that do not rely on such a fairly restricted and ideal world view.

1.2 Roadmap Ahead

We are specifically interested in answering the following questions

1. How can we arrive at the signaling policy that the rescuer should employ to minimise its cost?
2. How sensitive is our approach of finding the optimal signaling policy to changes in velocity of the players and changes in the environment topology? This translates to relaxing the assumptions 3 and 4 above.
3. How do we account for such uncertainty in designing a better approach to find the optimal signaling policy?

In Chapter 2 we will formalize our problem as a Stackelberg game. We will also present the baseline optimal signaling policy to be implemented under the assumptions 1,2,3 and 4 listed in the previous section. Chapter 3 then illustrates the sensitivity of the approach in Section 2 to the velocity of the rescuer. It also presents the challenges faced in designing the signaling policy when the travel costs over the rescue topology are uncertain. Chapter 4 provides one possible answer to the final question we posed above. In doing so, we present a novel algorithm to find feasible points for the rescuer to rendezvous with the rescuee despite the uncertainty in knowing the rescuee's path. The robust counterpart of the signaling policy accounts for any unknown but bounded uncertainty in the path travel costs over the rescue topology. Chapter 5 then highlights some issues with the approach designed in Chapter 4 and presents a modified approach to mitigate the drawbacks of the latter approach. In doing so, we will treat the path travel costs over the rescue topology as stochastic random variables in Chapter 5.

Chapter 2

Optimal signaling Policy

We assume a discretized terrain (e.g grid) for the rendezvous problem in this work. Equivalently, we can study the problem as defined over an undirected finite graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Path costs for travel over an edge between nodes i and j for the rescuee and the rescuer are defined as edge weights w_{ij}^r and w_{ij}^R respectively. Nodes v_r and v_R denote the initial position of the rescuee and rescuer respectively. The rescuer can send messages m from a finite non-empty support $\mathcal{M}$ and v_m corresponds to the goal indicated by message m.

Let $\mathcal{P}$ denote the set of all paths on the graph. $\mathcal{P}_{i \to j}$ denotes the set of all paths starting from node i and terminating at node j. ϕ_r and ϕ_R are real-valued functions defined on $\mathcal{P}$ that give the path cost for any path, for the rescuee and rescuer respectively.

2.1 Rescuee Policy

The rescuer, acting as the leader in the Stackleberg game, sends out a message $m \in \mathcal{M}$ to the rescuee. The rescuee then acting as the follower, observes this message and seeks to minimize

$$U_r(m, P) = \phi_r(P) \tag{2.1}$$

over paths $P \in \mathcal{P}_{v_r \to v_m}$. This optimization problem is simply the shortest path problem on an undirected graph. Dantzig (1963) gave a natural linear program formulation for the shortest path problem. Minimising (2.1) is equivalent to solving the linear network flow problem,

$$\min_{x_{ij} \geq 0} \sum_{ij \in \mathcal{E}} w_{ij}^r x_{ij} \tag{2.2}$$

$$\text{S.T. } \forall i \quad \sum_j x_{ij} - \sum_j x_{ji} = \begin{cases} 1 & i = v_r \\ -1 & i = v_m \\ 0 & \text{otherwise.} \end{cases} \tag{2.3}$$

x_{ij} here can be intuitively seen as an indicator variable for whether the edge (i, j) is a part of the shortest path. The constraints in (2.3) is a node-wise constraint and balances the inflow and outflow at every non-terminal edge. At the source node (v_r) the net outflow is 1, indicating that there is no edge of the shortest path going into the source. Likewise at the terminal node (v_m) the net inflow is 1 indicating that no edge of the shortest path exits this node.

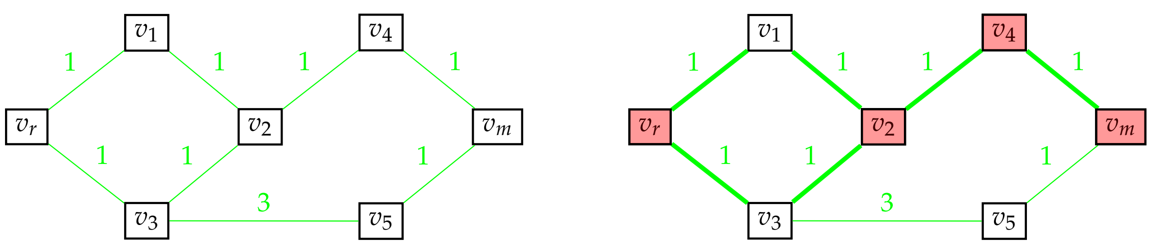
When the edge weights are known with certainty the linear program in (2.2) can be solved using the simplex method. The same problem may also be solved using the Dijkstra's Algorithm presented by Dijkstra (1959). Note that the minimizer to (2.2) needn't be a unique path.

Lemma 1 *The directed subgraph constructed from the set of all shortest paths between two nodes forms a DAG. In particular the one obtained from the set of minimizers to (2.2) forms a DAG. Additionally, every path in the sub-graph is a shortest path between the source and the sink node in the original graph.*

Definition 1 *Let $\mathcal{G}_m = (\mathcal{V}_m, \mathcal{E}_m)$ denote the directed sub-graph obtained as the minimizer to (2.2). We can define the candidate rendezvous points set $\mathcal{X}_m$ as,*

$$\mathcal{X}_m = \{v \in \mathcal{V}_m | v \in P \quad \forall P \in \mathcal{P}^m_{v_r \to v_m}\}$$

where $\mathcal{P}^m_{v_r \to v_m}$ denotes the set of path between v_r and v_m in the directed graph $\mathcal{G}_m$.



(a) An example graph

(b) The candidate rendezvous points set for this graph.

Figure 2.1: As an illustration, consider the graph in (a). For the edge-weights given we have two shortest paths from v_r to v_m, one along $v_r - v_1 - v_2 - v_4 - v_m$ and one along $v_r - v_3 - v_2 - v_4 - v_m$. The set of points that lie on every shortest path is highlighted in red in (b). Thus for this graph $\mathcal{X}_m = \{v_r, v_2, v_4, v_m\}$.

By definition, irrespective of the actual shortest path taken by the human, he will necessarily pass through every point in the candidate rendezvous points set. As the name suggests, this is the set of points that the rescuer will consider as potential points to rendezvous with the human. We make an additional assumption on the behaviour of the human.

Assumption 5 *The rescuee chooses one path at random from the paths in $\mathcal{G}_m$ to move towards the indicated target*

v_m. Unless intercepted by the rescuer at any point in the path, the rescuee stops only once he reaches the indicated target v_m and continues to wait there.

Claim 1 *$\mathcal{X}_m$ is non-empty and finite.*

By definition, $v_m, v_r \in \mathcal{X}_m$ and $\mathcal{X}_m$ is a subset of a finite set $\mathcal{V}$. By virtue of non-emptiness there always exists a point where the rescuer can potentially rendezvous with the rescuee. As a result, $\mathcal{X}_m$ can be considered by the rescuer as the set it is creating for itself by sending message m. We will distinguish v_m as the *terminal rendezvous point*.

2.2 Rescuer Optimal Policy

The rescuer must take its action with the best interests of the rescuee in mind. At the same time, it must also ensure it is passing through regions with low path cost (for example, in ensuring flight path in a relatively safe environment). Accordingly, we define the cost function for the rescuer as,

$$U_R(m, v_x, P_R, P_r) = k_1\phi_R(P_R^x) + k_2\phi_r(P_r^x) \tag{2.4}$$

where $v_x \in \mathcal{X}_m$ is the rendezvous point, $P_R^x \in \mathcal{P}_{v_R \to v_x}$ and $P_r^x \in \mathcal{P}_{v_r \to v_x}$. k_1 and k_2 are tunable parameters that determine the relative importance of the path cost to rescuee and the path cost to rescuer when the rescuer is seeking to optimize its total cost. If k_1 is set very high, then the rescuer tries to minimise the distance it travels and places more priority on trying to get the human to travel to a favourable rendezvous point. If k_2 is set high, then the rescuer disregards the costs it faces in trying to get to the rescuee as quickly as possible. In minimizing (2.4), the rescuer picks both the message m to be sent and the rendezvous point v_x. The chosen optimal v_x and the corresponding paths P_r^x and P_R^x must satisfy the constraint

$$\left(\frac{\phi_R(P_R^x)}{V_R} - \frac{\phi_r(P_r^x)}{V_r}\right)\mathbb{1}_{v_x \neq v_m} \leq 0 \tag{2.5}$$

where V_R and V_r are the constant velocities of the rescuer and rescuee respectively. The first two terms in the left hand side of (2.5) can be interpreted as the time taken by the rescuer and rescuee respectively, to reach the chosen rendezvous node v_x . The third term in the constraint is an indicator variable that takes the value 1 if the chosen rendezvous node is not terminal and 0 if it is. This constraint indicates that the rescuer must reach the rendezvous point before the rescuee, for any point that is not the terminal rendezvous point. It can be observed that this constraint is in line with our Assumption 5 in allowing for a successful rendezvous.

Defining the ratio of velocities $\frac{V_R}{V_r} \triangleq k_v$ we can re-write the optimisation problem to be solved by the the rescuer as,

$$\min_{m \in \mathcal{M}} \min_{v_x \in \mathcal{X}_m} k_1 \phi_R^*(v_x) + k_2 \phi_r^*(v_x) \tag{2.6}$$

$$\text{S.T.} \;\; (\phi_R^*(v_x) - k_v \phi_r^*(v_x)) \mathbb{1}_{v_x \neq v_m} \leq 0 \tag{2.7}$$

$$\text{Where, } \phi_R^*(v_x) \triangleq \min_{P \in \mathcal{P}_{v_R \to v_x}} \phi_R(P) \tag{2.8}$$

$$\phi_r^*(v_x) \triangleq \min_{P \in \mathcal{P}_{v_r \to v_x}} \phi_r(P) \tag{2.9}$$

Equation (2.9) arises from our assumption that the rescuee takes shortest paths to the indicated goal and by Principle of Optimality, also takes the shortest path to any $v_x \in \mathcal{X}_m$. Both (2.9) and (2.8) are once again the shortest path problems on a graph and we can solve their equivalent linear problem formulations instead. For any rendezvous point v_x in the candidate rendezvous point's set $\mathcal{X}_m$ we can re-write (2.9) as the equivalent linear program (LP),

$$\min_{x_{ij} \geq 0} \sum_{ij \in \mathcal{E}} w_{ij}^r x_{ij} \tag{2.10}$$

$$\text{S.T. } \forall i \quad \sum_j x_{ij} - \sum_j x_{ji} = \begin{cases} 1 & i = v_r \\ -1 & i = v_x \\ 0 & \text{otherwise} \end{cases} \tag{2.11}$$

and re-write (2.8) as the equivalent LP,

$$\min_{x_{ij} \geq 0} \sum_{ij \in \mathcal{E}} w_{ij}^R x_{ij} \tag{2.12}$$

$$\text{S.T. } \forall i \quad \sum_j x_{ij} - \sum_j x_{ji} = \begin{cases} 1 & i = v_R \\ -1 & i = v_x \\ 0 & \text{otherwise} \end{cases} \tag{2.13}$$

As indicated in Section 2.1 we can solve the linear programs described in (2.10) and (2.12) using either simplex methods or by implementing the Dijkstra's Algorithm (DA). Having solved the optimisation in (2.10) and (2.12) for each node in $\mathcal{X}_m$, the constrained optimisation in (2.6) can be performed by a search over the finite non-empty sets $\mathcal{M}$ and $\mathcal{X}_m$.

Having obtained the optimal signaling policy for the baseline case we will now do away with some of the assumptions we made in the introducing the problem statement. Specifically, in the next section we will look at the sensitivity of our signaling policy to uncertainties in the velocities of the rescuer and the rescuee. We also briefly introduce the issues faced if there is uncertainty in the rescuers knowledge of the topology faced by the rescuee.

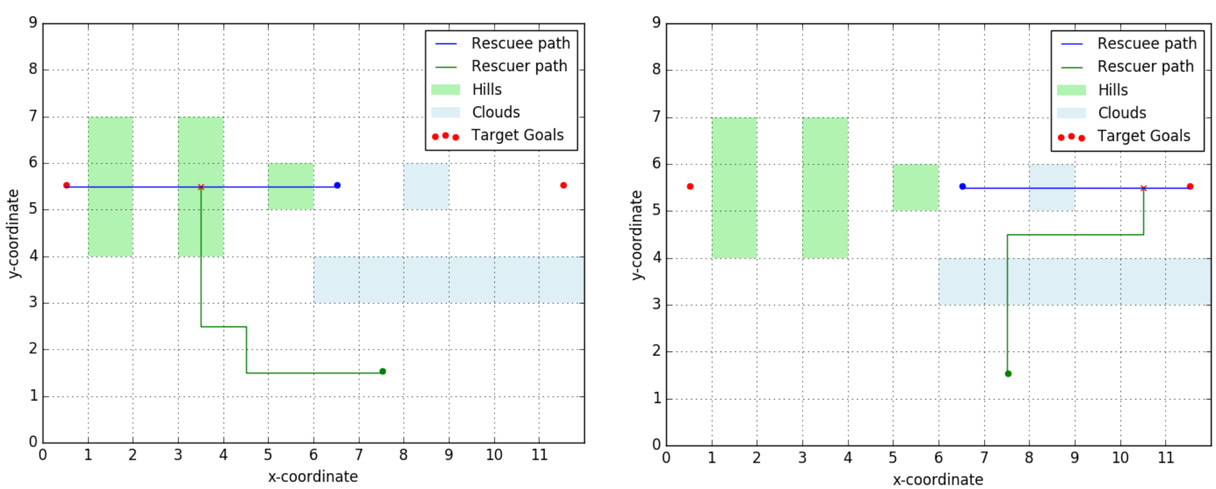
Chapter 3

Sensitivity to Uncertainties

In this chapter, we will consider the effect of removing some of the assumptions we made in introducing the problem statement. Specifically, Section 3.1 will look at the change in the optimal signal generated by the policy designed in the Chapter 2 with change in velocities of agents when Assumption 4 is discarded. Then in Section 3.2 we will look briefly at the effect of discarding Assumption 3 from Chapter 1.

3.1 Sensitivity to Velocity

The effect of uncertainty in velocity can be encapsulated in uncertainty in the parameter k_v. Recall that k_v is simply the ratio of velocities of rescuer and rescuee. Consider the topology as illustrated in Fig. 3.1a and Fig. 3.1b. The illustration assumes the existence of just two messages $\mathcal{M} = \{L, R\}$.



(a) Rendezvous trajectories when $k_v = 1.6$

(b) Rendezvous trajectories when $k_v = 1.9$

Figure 3.1: Illustrating the sensitivity of the optimal signal to parameter k_v. With an increase in the velocity ration from 1.6 to 1.9 we observe that the optimal signal outputted by our policy switches from L to R.

Table 3.1 presents the optimal signal m_{opt} and the cost U_R in sending that signal for various values of the velocity ratio k_v for the topology given in Fig. 3.1a and Fig. 3.1b. The signals have been obtained using the policy presented in the previous chapter. We can interpret an increase in k_v as either an increase in speed of the the rescuer or decrease in the speed of the rescuee.

k_v	m_{opt}	U_R	v_x
1.3	R	16	$(5, 11)$
1.6	L	15	$(5, 3)$
1.9	R	14	$(5, 10)$
2.5	L	10	$(5, 5)$
3.1	R	8	$(5, 7)$

Table 3.1: Variation of m_{opt}, U_R and the rendezvous point v_x with increasing k_v. v_x denotes the position of a grid square using the coordinates of its bottom left corner.

We see that the optimal signal to be sent switches multiple times with an increase in k_v. This sensitivity can be explained as follows. Without loss of generality we can assume that rescuee velocity is constant and rescuer velocity is increasing with k_v. Hills (green shading) take a longer time for rescuee to traverse, and thus, give more time for the rescuer to rendezvous with him there. But once the rescuee has traversed the hill and is passing through a region of low cost he quickly passes through it, getting out of the range of the rescuer quickly. As the velocity of the rescuer increases, it can reach any point on the path of the rescuee quicker and reduce the cost U_R by performing an earlier rendezvous. For a small grid size like ours we obtained 4 switches. For the locations of the players and the targets as illustrated in Fig. 3.1a we can show that for every $M \in \mathbb{Z}^+$, we can find some minimum dimension N for the grid ($N \times N$) and some topology over the grid such that the number of switches is greater than M. Increasing the dimension of the grid can be interpreted equivalently as increasing the resolution of the grid over the layout in Fig. 3.1a.

Claim 2 *For any M in $\mathbb{N}$, there exists a grid of size $N \times N$ (with N scaling linearly in M) and a choice of target goals, obstacle and starting points such that the optimal signal to be sent switches M times with change in velocities of the players.*

We will provide a sketch as an illustration to justify this claim. We wish to show that there exists *some topological layout* over our grid with *some initial positions* of the rescuee and the rescuer such that we can get a very large number of switches in the optimal signal with increase in velocity of the rescuer. We will consider the layout of the players and the target goals as described in 3.2. We now need to construct a family of topological layouts that will give us the sensitive behaviour we want.

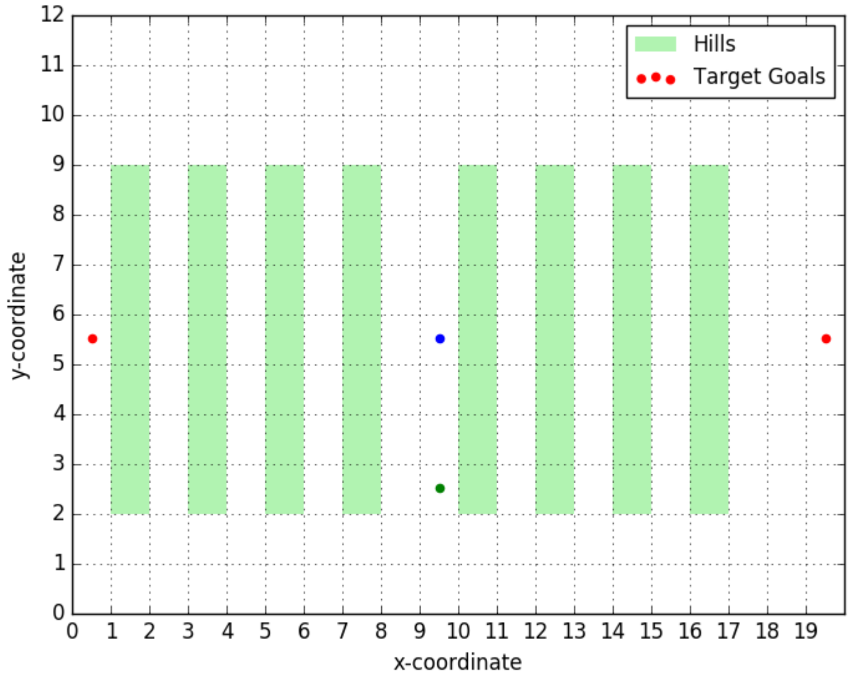


Figure 3.2: A possible terrain to obtain multiple switches. The 'hill' regions consist of edges having thrice the cost to traverse compared to other regions. The initial location of the rescuer and the rescuee is given by the green and blue circles respectively.

We will restrict our search to the layouts where the shortest path from the rescuee's initial location to each target goal is unique. Assuming just two target goals $\{L, R\}$, the set of candidate rendezvous points can be represented as a graph (Fig. 3.3).

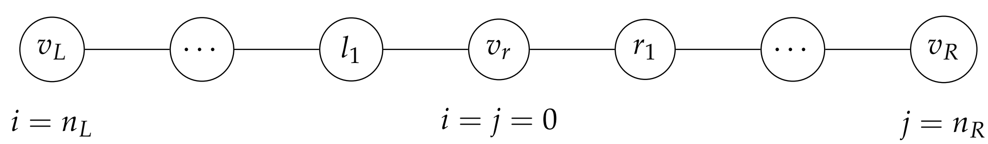


Figure 3.3: Shortest paths for the rescuee to both target goals.

Let nodes on the rescuee's path going to the target goal L be indexed by i and let n_L be the number of nodes in the graph representation of the path. Likewise j and n_R denote the index and the number of nodes on the path going to R. $\{l_i\}$ and $\{r_j\}$ gives the set of nodes on each of the paths. Then $l_0 = r_0 = v_r$, $l_{n_L} = v_L$ and $r_{n_R} = v_R$. Let $\phi_r(v)$ and $\phi_R(v)$ denote the path cost for the rescuee and the rescuer respectively to go to node v from their initial position.

We will now construct a sequence of path costs for both the rescuee and the rescuer to both the goals. We will justify the realizability of such path costs later, but for now assume they are indeed realizable. By construction, the sequence of path costs over the nodes are given as,

$$\phi_R(l_i) = \phi_R(r_j) = i + 4 \tag{3.1a}$$

$$\phi_r(l_i) = \begin{cases} 2(i-1) & i \text{ is even} \\ 2i - 1 & i \text{ is odd} \end{cases} \tag{3.1b}$$

$$\phi_r(r_j) = \begin{cases} 2j - 1 & j \text{ is even} \\ 2(j-1) & j \text{ is odd, } j \neq 1 \\ 1 & j = 1 \end{cases} \tag{3.1c}$$

Assuming the constants k_1 and k_2 in the cost function for the rescuer to be both identity. The total cost U_R of rendezvous at node r_j and l_i to the rescuer is obtained as,

$$U_R(l_i) = \begin{cases} 3i + 2 & i \text{ is even} \\ 3i + 3 & i \text{ is odd} \end{cases} \tag{3.2a}$$

$$U_R(r_j) = \begin{cases} 3j + 2 & j \text{ is odd} \\ 3j + 3 & j \text{ is even} \end{cases} \tag{3.2b}$$

We will now take a look at the velocities at which we can reach each node. Since rescuer is constrained to reach any node v before the rescuee for a successful rendezvous, we must have the velocity ratio k_v satisfying,

$$k_v \geq \frac{\phi_R(v)}{\phi_r(v)}$$

Without loss of generality we will assume the rescuee's velocity to be a constant identity. Then, the threshold velocity for the rescuer to reach any node v is simply,

$$V_R(v) = \frac{\phi_R(v)}{\phi_r(v)} \tag{3.3}$$

Using (3.1) we can compute the threshold velocity for each node and some algebraic manipulation (refer

Appendix A.2) leads us to the following inequality for any even i,

$$V_R(l_i) > V_R(r_i) > V_R(r_{i+1}) > V_R(l_{i+1}) > V_R(l_{i+2}) \quad i > 5 \tag{3.4a}$$

$$V_R(l_i) > V_R(r_{i+1}) > V_R(r_i) > V_R(l_{i+2}) > V_R(l_{i+1}) \quad \text{otherwise} \tag{3.4b}$$

From (3.2) we make the following observation; when $i = j$ and i is even, $U_R(l_i) \leq U_R(r_i)$ and when j is odd, $U_R(r_i) \leq U_R(l_i)$. Thus, when the velocity increases from $V_R(r_i)$ to $V_R(l_i)$ for an even i the best signal to be sent switches from R to L. Likewise for an odd i when the velocity increases from $V_R(l_i)$ to $V_R(r_i)$ the the best signal to be sent switches from L to R.

We see that the number of switches can be made equal to $\min\{n_R, n_L\}$. If the grid is made large enough and the target goals are far enough then we can have an arbitrarily large number of switches. To see this we need to reconsider the question of realizability of the path costs described in (3.1). It can be seen that such a path cost sequence is actually realized up-to an index of $n_R = n_L = 7$ if each of the 'hill' regions are made to have a path cost of 3 while the other regions have a cost of 1. Correspondingly, we can achieve 7 switches as described above. The grid size we considered was 20×12.

In general, for the initial positions of rescuee and the rescuer and the target goal locations we considered, the grid size N required for M switches is obtained as,

$$N \geq 2M + 4$$

We showed that small changes in the ratio of the velocity of the players can strongly affect the outcome of the optimal signaling policy. In our scenario, it might not always be possible for the rescuer to know the exact velocity of the rescuee. Thus, there is a need for a signaling policy that is robust to uncertainty of velocity of players.

3.2 Uncertainty in edge-weights $\{w^r_{ij}\}$ and $\{w^R_{ij}\}$

Several difficulties arise if the rescuee's edge weights w^r_{ij} are not known to the rescuer with certainty. First, the rescuer cannot determine the rescuee's exact set of shortest paths and, a fortiori, the candidate rendezvous points sets $\mathcal{X}_m$'s. This, in turn, affects the rescuer's ability to determine if and where a rendezvous can occur. In addition, even if it knew for sure that a given node is visited by the rescuee, the rescuer would be uncertain as to the cost of the path taken by the rescuee, thus making it challenging to evaluate its own actions according to (2.6,2.7).

In order to address these issues, in the following chapter, we first introduce the notion of robust candidate rendezvous set, which contains nodes that the rescuee will always traverse and, as we prove, can be computed efficiently by the rescuer. Next, we introduce robust counterparts to (2.6,2.7) which allow the rescuer to compute the optimal message in the presence of uncertainty in the rescuee's weights.

Chapter 4

Robust Optimal signaling Policy

From now on, we assume that the edge weights w^r_{ij} and w^R_{ij} are unknown but bounded: $\underline{w}^r_{ij} \leq w^r_{ij} \leq \overline{w}^r_{ij}$ and $\underline{w}^R_{ij} \leq w^R_{ij} \leq \overline{w}^R_{ij}$ $\forall$ $ij \in \mathcal{E}$. We designate the cartesian product $\Pi_{ij\in\mathcal{E}}[\underline{w}^r_{ij}, \overline{w}^r_{ij}]$ as Ω^r and $\Pi_{ij\in\mathcal{E}}[\underline{w}^R_{ij}, \overline{w}^R_{ij}]$ as Ω^R and any arbitrary element from this set is denoted by w^r and w^R respectively.

We make an observation that in the analysis presented in Chapter 2 the edge weights only show up when we seek to find the shortest paths over the graph. In success critical problems such as our rescue scenario, we wish to be completely risk-averse. A natural step forward is then to consider a robust optimal approach in designing our signaling policy.

We presented the linear program formulation of the shortest path problem in Chapter 2. The same problem can also be presented as an integer programming problem, with each $x^r_{ij}, x^R_{ij} \in \{0, 1\}$ (Dantzig (1963)). Efficient ways to compute the robust discrete optimal solutions to this formulation were presented by Bertsimas and Sim (2003) assuming an upper bound on the number of edge-weights that are uncertain. We will work with the more general (and simpler) scenario where we assume all edge-weights are uncertain. In our work, we use the notions of a robust counterpart to an optimisation problem as presented by Ben-Tal et al. (2009).

Consider an optimisation problem given by

$$\min_x f(x, w) \tag{4.1}$$

$$s.t\ g(x, v) < 0 \tag{4.2}$$

Where, $(v, w) \in \mathcal{U}$ are uncertain constants from an uncertainty set $\mathcal{U}$. Then, motivated by Ben-Tal et al. (2009) we have the following notions.

Definition 2 *An uncertain optimisation problem is the collection,*

$$O_{\mathcal{U}} = \big\{ \min_x\ f(x, w) \quad s.t\ g(x, v) < 0 \big\}_{(v,w)\in\mathcal{U}} \tag{4.3}$$

Definition 3 *A vector x is a* robust feasible solution *to $O_{\mathcal{U}}$, if it satisfies all realizations of the constraints from the uncertainty set, that is*

$$g(x,v) < 0 \quad \forall (v,w) \in \mathcal{U} \tag{4.4}$$

Definition 4 *The Robust Counterpart of the uncertain optimisation problem $O_{\mathcal{U}}$ is the optimization problem,*

$$min_x \{ \sup_{(w,v)\in\mathcal{U}} f(x,w) : g(x,v) < 0 \quad \forall (w,v) \in \mathcal{U} \} \tag{4.5}$$

where we are minimising over all the robust feasible solutions.

Formally we can pose our question as, "What signaling policy should the rescuer adopt to incur optimal costs while guaranteeing a successful rescue?".

4.1 Robust Optimal Candidate Rendezvous Set

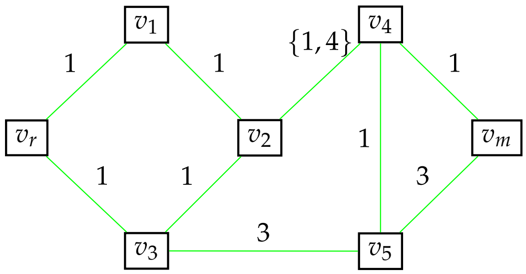
We make the assumption that the rescuee seeks the shortest path in a certain environment with a realization $w^r_{ij} = \hat{w}^r_{ij}$ as the edge weights over the graph $\mathcal{G}$. The realised path of the rescuee can then be obtained by solving the optimisation in (2.1). The rescuer does not a priori know these realized edge weights ($\hat{w}^r_{ij}$) and is faced with finding the set of candidate rendezvous points $\mathcal{X}_m$ in a graph with unknown edge-weights. We define the set of *robust candidate rendezvous points*.

In Chapter 2, we easily obtained $\mathcal{X}_m$ as a finite set from the finite graph $\mathcal{G}_m$. But obtaining $\hat{\mathcal{X}}_m$ using (4.6) involves an uncountable intersections over finite sets. We seek to find the "set of points which lie in the shortest path for all possible combination of edge-weights". Fortunately, it is possible to state the following,

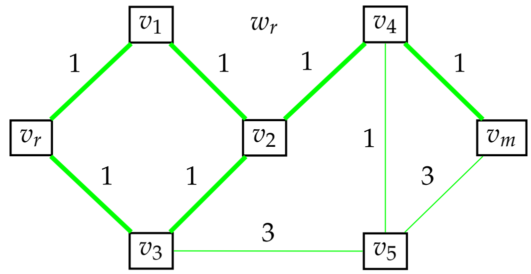
Proposition 1 *Algorithm 1 presented ahead terminates, computes the set $\hat{\mathcal{X}}_m$ and runs in $O(|\mathcal{V}|^3)$.*

Definition 5 *Let $\mathcal{X}^w_m$ be the set of candidate rendezvous points for the set of edge weights $\{w^r_{ij}\}$ as defined in Definition 1. The robust candidate rendezvous set is obtained as,*

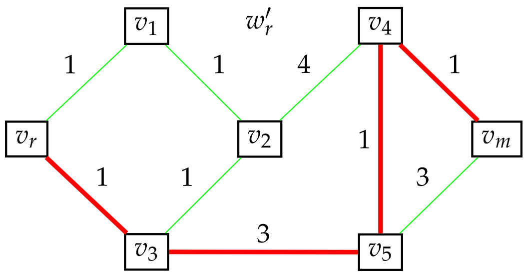
$$\hat{\mathcal{X}}_m = \bigcap_{\{w^r_{ij}\}\in\Omega^r} \mathcal{X}^w_m \tag{4.6}$$



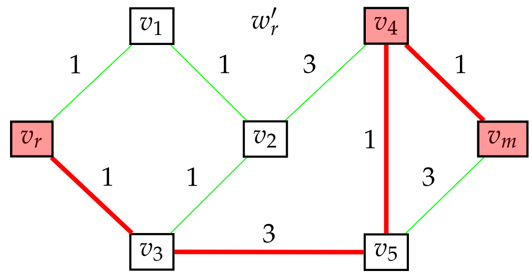
(a) Edge weight over the edge (v_2, v_4) can be either 1 or 4.



(b) Shortest paths for the realised edge-weights w_r.



(c) Shortest path for the realised edge-weights w'_r.



(d) Robust candidate rendezvous points are highlighted.

Figure 4.1: (a) shows the graph in consideration. We see that the edge weights over all edges except one are constants. (b) and (c) show the shortest paths over the graph for two different realizations of edge-weights. (d) highlights the set of robust candidate rendezvous points. So we have $\hat{\mathcal{X}}_m = \{v_r, v_4, v_m\}$

Algorithm 1: Algorithm to find the robust candidate rendezvous points

---

Result: Obtain $\hat{\mathcal{X}}_m$

Set edge weights $\{w_{ij}^{r1}\} = \{\underline{w}_{ij}^r\}$;

Find the graph of shortest path $\mathcal{G}_m^{*1} = (\mathcal{V}_m^{*1}, \mathcal{E}_m^{*1})$ and the corresponding set of paths $\{P^*\}^1$;

Initialize $\hat{\mathcal{X}}_m^1 = \{v : v \in P \; \forall P \in \{P^*\}^1$;

Set $\mathcal{F}^1 = \{ij : w_{ij}^{r1} = \underline{w}_{ij}^r, ij \in \mathcal{G}_m^{*1}\}$;

Set $k = 1$;

while $\mathcal{F}^k \neq \varnothing$ *AND* $\hat{\mathcal{X}}_m^k \neq \{v_r, v_m\}$ **do**

  Set $w_{ij}^{rk} = \overline{w}_{ij}^r \; \forall \, (ij) \in \mathcal{F}^k$ and $w_{ij}^{rk} = w_{ij}^{r(k-1)} \; \forall \, (ij) \in \mathcal{E} / \mathcal{F}^k$;

  Find the graph of shortest path $\mathcal{G}_m^{*k+1}$ and paths $\{P^*\}^{k+1}$ with new weights ;

  $\hat{\mathcal{X}}_m^{k+1} = \hat{\mathcal{X}}_m^k \bigcap \{v : v \in P \; \forall P \in \{P^*\}^{k+1}\}$;

  Update $\mathcal{F}^{k+1} = \{ij : w_{ij}^r(k+1) = \underline{w}_{ij}^r, ij \in \mathcal{E}_m^{*k+1}\}$;

  $k = k + 1$

end

return $\hat{\mathcal{X}}_m^k$

A proof of correctness for this algorithm is provided in Appendix B. In an undirected graph with positive edge-weights, all equivalent shortest paths P^* can be obtained using a minor modification of the Dijkstra's algorithm. If the implementation of Dijkstra's Algorithm runs in $\mathcal{O}(|\mathcal{V}|^2)$, then Algorithm 1 presented ahead runs in $\mathcal{O}(|\mathcal{V}|^3)$. This polynomial time-complexity for Algorithm 1 preserves the efficiency of our approach in finding the robust optimal signaling policy. The computational efficiency of the algorithm allows the rescuer to compute the set of robust candidate rendezvous points on board in real time.

While we assume that the signal sent by the rescuer is correctly interpreted by the rescuee and that the action taken by the rescuee is indeed according to the best response strategy, it may of-course not be the case. In such scenarios the ability to quickly compute the optimal strategy again allows for 'corrective' signals to be sent using the same signaling mechanism. Such repetitive signaling for planning, while not addressed in our current work, is certainly an interesting direction of consideration for future work.

4.2 Robust optimal signaling policy

We make an assumption on the ratio of velocities k_v for the remainder of our work.

Assumption 6 *Let*

$$w^R_{\max} \triangleq \max_{ij} \overline{w}^R_{ij}$$

$$w^r_{\min} \triangleq \min_{ij} \underline{w}^r_{ij}$$

Then, we assume a lower bound on the ratio of velocities as,

$$k_v \triangleq \frac{V_R}{V_r} \geq \frac{w^R_{\max}}{w^r_{\min}} \tag{4.7}$$

This assumption formalizes the notion that rescuer (UAV) can move faster on any part of the terrain than the rescuee. It is worth noting that this assumption alone does not guarantee the existence of non-terminal rendezvous point. It merely implies that on any given path on $\mathcal{G}$, the rescuer takes less time than the rescuee. Thus, even if the rescuee was substantially closer to the target goal than the rescuer, the larger speed of the rescuer may still not help it reach some intermediate node on the rescuee's path.

Definition 6 *For any two nodes v_m, x_n in a path $P \in \mathcal{P}_{v_r \to v_m}$ we define a partial ordering '$\leq$' as,*

$$v_m \leq v_n \quad \textit{if} \quad \phi^*_r(v_m) \leq \phi^*_r(v_n)$$

with ϕ^*_r defined in (2.9).

Without loss of generality, we can list all the nodes in $\hat{\mathcal{X}}_m$ in increasing order as, $v_r = v_{x,1} \leq v_{x,2} \leq \cdots \leq v_{x,L} = v_m$, where $L = |\hat{\mathcal{X}}_m|$. Assumption 6 leads us to,

Claim 3 *For any m, n such that $1 \leq m < n \leq L$ and for any realization $\{w_{ij}^r\} \in \Omega_r$, $\{w_{ij}^R\} \in \Omega_R$ we have,*

$$\phi_R^*(v_{x,m}) - k_v \phi_r^*(v_{x,m}) \leq 0 \implies \phi_R^*(v_{x,n}) - k_v \phi_r^*(v_{x,n}) \leq 0$$

with ϕ_R^* and ϕ_r^* given by (2.8) and (2.9) respectively.

The proof for this claim can be found in Appendix A.3. This statement shows that if a node is a robust candidate rendezvous point then all nodes in $\hat{\mathcal{X}}_m$ succeeding (ordered by Definition 6) this point are also candidate rendezvous points. We draw the straight forward inference from Claim 3,

Corollary 1 *If there exists atleast one robust candidate rendezvous point then necessarily the terminal rendezvous point v_m is also a robust candidate rendezvous point.*

We are now equipped to analyse the problem of finding the optimal signaling policy when faced with uncertain path costs. In doing so we can first break our problem into two cases.

I There always exists atleast one robust candidate rendezvous point for all possible edge-weights

II No robust candidate rendezvous point for some $\{w_{ij}^R\}, \{w_{ij}^r\}$

By Corollary 1 it suffices to check whether v_m is a robust candidate rendezvous point to verify which of the two cases we are in.

4.3 Case I: At-least one robust candidate rendezvous point

In this case, for any $\{w_{ij}^r\} \in \Omega_r$ and $\{w_{ij}^R\} \in \Omega_R$ the constraint in (2.7) simplifies to

$$\phi_R^*(v_x) - k_v \phi_r^*(v_x) \leq 0 \tag{4.8}$$

Then as a direct consequence of the Definition 3 we can make the following proposition.

Proposition 2 *Any robust feasible rendezvous point v_x for (4.8) satisfies,*

$$\phi_{R,\max}^*(v_x) - k_v \phi_{r,\min}^*(v_x) \leq 0 \tag{4.9}$$

Where,

$$\phi^*_{R,\max}(v_x) = \min_{x_{ij} \geq 0} \sum_{ij \in \mathcal{E}} \overline{w}^R_{ij} x_{ij} \tag{4.10}$$

$$S.T.\ \forall i \quad \sum_j x_{ij} - \sum_j x_{ji} = \begin{cases} 1 & i = v_R \\ -1 & i = v_x \\ 0 & otherwise \end{cases}$$

$$\phi^*_{r,\min}(v_x) = \min_{x_{ij} \geq 0} \sum_{ij \in \mathcal{E}} \underline{w}^r_{ij} x_{ij} \tag{4.11}$$

$$S.T.\ \forall i \quad \sum_j x_{ij} - \sum_j x_{ji} = \begin{cases} 1 & i = v_r \\ -1 & i = v_x \\ 0 & otherwise \end{cases}$$

The robust counterpart to the optimisation problem presented in (2.6) is then given by,

$$\min_{m \in \mathcal{M}} \min_{v_x \in \mathcal{X}_m} k_1 \phi^*_{R,\max}(v_x) + k_2 \phi^*_{r,\max}(v_x) \tag{4.12}$$

subject to (4.9), where, $\phi^*_{r,\max}(v_x)$ can be obtained by replacing $\underline{w}^r_{ij}$ in (4.11) with $\overline{w}^r_{ij}$. □

Proof: It is easy to see that the proposition noted above is a direct consequence of the definitions of robust counterpart to an optimization problem as provided by Ben-Tal et al. (2009). Motivated by Definition 4, the robust counterpart to the uncertain optimisation problem presented in (2.6) is given by,

$$\min_{m \in \mathcal{M}} \min_{v_x \in \mathcal{X}_m} \max_{w^R_{ij}, w^R_{ij}} k_1 \phi^*_R(v_x) + k_2 \phi^*_r(v_x) \tag{4.13a}$$

$$\text{S.T.} \max_{w^R_{ij}, w^R_{ij}} \phi^*_R(v_x) - k_v \phi^*_r(v_x) \leq 0 \tag{4.13b}$$

We can make the following observations.

$$\phi^*_R(v_x) - k_v \phi^*_r(v_x) \leq \phi^*_{R,\max}(v_x) - k_v \phi^*_{r,\min}(v_x)$$

And the equality holds when each $w^r_{ij} = \underline{w}^r_{ij}$ and $w^R_{ij} = \overline{w}^r_{ij}$. Thus, if a solution satisfies (4.9) then it satisfies (4.13b) for all values of w^r_{ij} and w^r_{ij}. In other words, such a solution is robust feasible by Definition 3.

By similar reasoning we can see that $\max_{w^R_{ij}, w^R_{ij}} \ k_1 \phi^*_R(v_x) + k_2 \phi^*_r(v_x)$ is attained for $w^r_{ij} = \overline{w}^r_{ij}$ and $w^R_{ij} =$

$\overline{w}^r_{ij}$. Thus, the robust counterpart to (4.13) is given by (4.12) subject to (4.9). □

Explained in simple words, (4.9) formalizes the notion that the potential nodes we will consider for the purpose of rendezvous can always be reached by the rescuer before the rescuee irrespective of the edge-weights over the graph. Among these nodes the minimization in (4.12), translates to finding the node with the best worst-case cost of successful rendezvous.

4.4 Case II: No robust candidate rendezvous point

By Assumption 5, we know that the rescuee travels to the target goal (v_m) and waits there. In the scenario where we have no robust candidate rendezvous point, the only way to guarantee a successful rendezvous is by meeting the rescuee at the target goal node v_m. Then for such a message the rescuer will only consider the cost to rendezvous at the target goal. If there exist no candidate rendezvous points for any message m, the robust counterpart to (2.6) is simply

$$\min_{m \in \mathcal{M}} \min_{v_x \in \mathcal{X}_m} k_1 \phi^*_{R,\max}(v_m) + k_2 \phi^*_{r,\max}(v_m) \tag{4.14}$$

Where $\phi^*_{R,\max}(v_m)$ and $\phi^*_{r,\max}(v_m)$ is obtained as we did for case I in Equations 4.11 and 4.10.

4.5 Robustness to Velocity Variation

We will treat the variation in velocity of the rescuer and the rescuee as variations of the parameter k_v. We assume that k_v is unknown but bounded and takes values over a range $[\underline{k}_v, \overline{k}_v]$. The objective function in (2.6) is unaffected by the value of k_v. The constraint (2.5) is affine in k_v. The robust counterpart to the optimisation in (2.6) subject to (2.5) is obtained as,

$$\min_{m \in \mathcal{M}} \min_{v_x \in \mathcal{X}_m} k_1 \phi^*_R(v_x) + k_2 \phi^*_r(v_x) \tag{4.15}$$

$$\text{S.T. } (\phi^*_R(v_x) - \underline{k}_v \phi^*_r(v_x)) \mathbb{1}_{v_x \neq v_m} \leq 0 \tag{4.16}$$

with ϕ^*_R and ϕ^*_r are given by (2.8) and (2.9) respectively. The edge weights $\{w^r_{ij}\}$ and $\{w^R_{ij}\}$ are assumed fixed and known above. The robustification of the optimisation in Chapter 2 with respect to the edge weights and the parameter k_v can be done independently. For the subsequent robustification of the constraint in (4.16) with respect to edge-weight uncertainty, we only need the $\underline{k}_v$ to satisfy the constraint in Assumption 6.

4.6 Example

In the preceding sections, we formalized an approach to arrive at the robust optimal signaling policy for a Stackelberg rendezvous game. In closing this chapter we will provide an illustrative example of the designed signaling policy in practice.

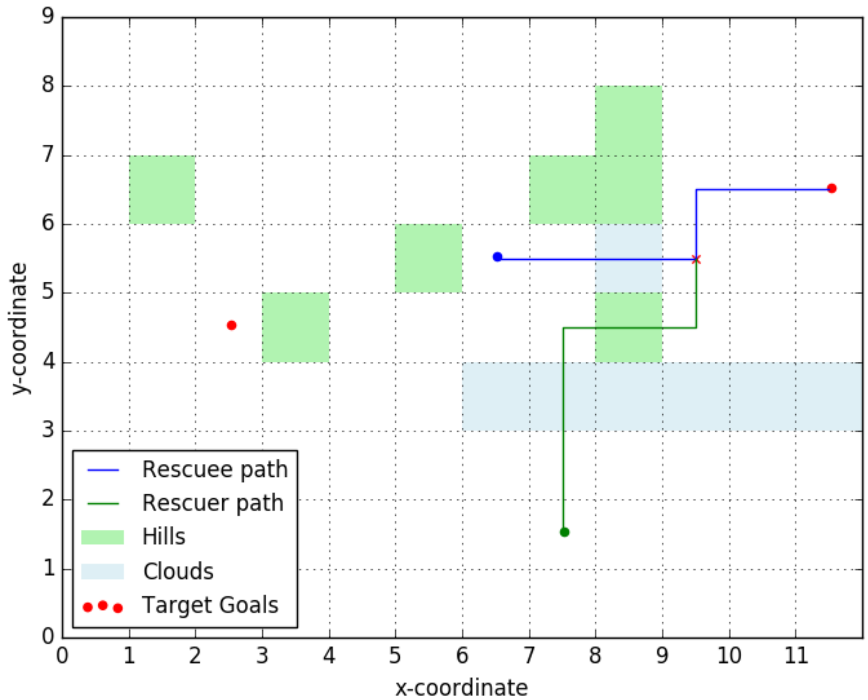


Figure 4.2: Clouds (blue shading) act as 'high edge weight' regions to the UAV and hills (green shading) act as 'high edge weight' regions to the rescuee. The optimal signal sent out was indicating the right target goal.

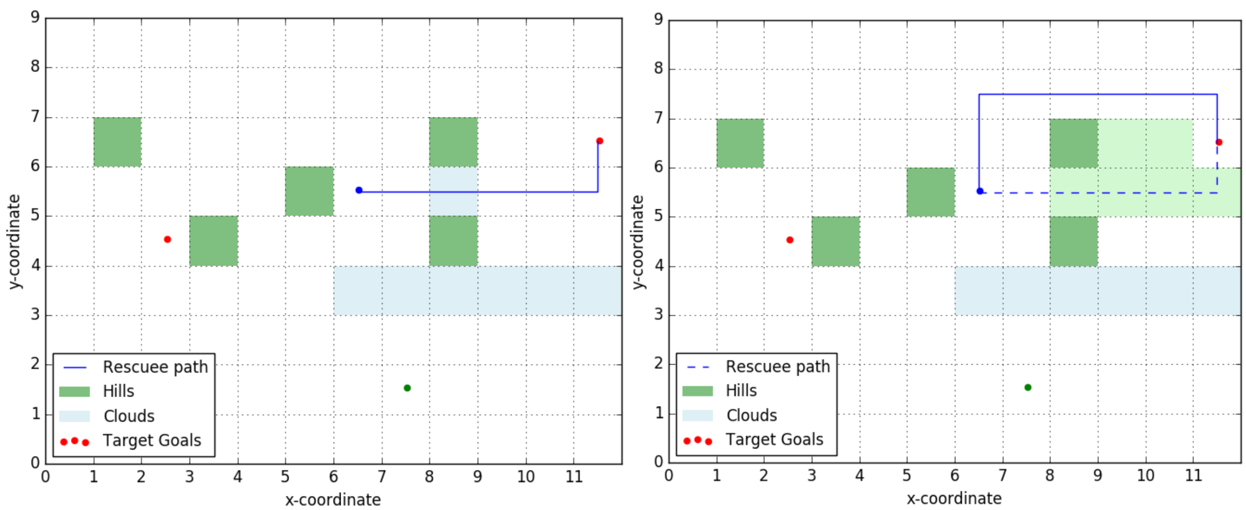
We present the results of a simple simulation on a carefully designed rescue topology to highlight some features of our approach. The parameters chosen are $k_1 = 1$,$k_2 = 1$ and $k_v = 3.1$. All edge weights w^r_{ij} and w^R_{ij} are unknown to the rescuer and are assumed to be uniformly distributed random variables. Each edge in the graph can be of one of two types - 'High weight edges', where each edge weight is a random variable supported over $[2.5, 3]$ and 'low weighted edges', each supported over $[1, 1.5]$. It can be verified that Assumption 6 holds for these set of edge weights and the velocity ratio k_v. The topology over the rescue terrain as well as the path's travelled by the rescuee and the rescuer are presented in Fig. 4.2.

It can be observed that although the left target goal was spatially closer to both the rescuee and the rescuer, the rescuer, implementing the robust optimal signaling policy, signals the rescuee to go towards the right goal. By indicating the right$(m = R)$ target as the intended goal, the rescuer creates the set $\mathcal{X}_R$, containing grid squares $(7, 6)$, $(8, 5)$, $(9, 5)$ and $(11, 6)$, for itself. Note that we denote the position of a grid square using the coordinates of its bottom left corner. The rescuer can now pick one among these candidate rendezvous points for a successful rendezvous (subject to their feasibility). If the rescuer were to signal left($m = L$), there would be no such non-terminal points in the set $\mathcal{X}_L$. This availability of robust candidate rendezvous points encourages the rescuer to signal going right as the rescuer's cost to reach some of the points in $\mathcal{X}_R$ is lower than the cost to reach v_L. In this particular scenario, the rescuer chooses the point $(9, 5)$ for a successful rendezvous.

Chapter 5

Stochastic Optimal signaling Policy

In the preceding chapter, we considered a robust approach in finding an optimal signaling policy for our rescue scenario. Such an approach trades optimality for robustness by designing a policy that only maximises the rewards in the worst-case scenario. It may often be the case that the probability of the worst cases scenario being realized is very small. And it may also be the case that optimal action in the worst cases scenario is strongly sub-optimal in many of the other scenario. It is not very hard to construct a scenario that achieves such sub-optimality in our framework.



(a) Path cost to the rescuee is 7.

(b) Path cost to rescuee is 9 along both indicated paths.

Figure 5.1: The two figures illustrate two realisations of the stochastic topology over the grid. The dark green hilly regions have a cost-to-traverse of 3 for the rescuee. All other grid squares have a cost to traverse varying uniformly over $[1, 1.5]$ for the rescuee. In the first graph, all these grids squares have a realized cost of 1, while in the second figure the light green shaded grid squares have a cost to traverse of 1.5. The other grid squares remain unchanged in the latter.

Consider the two images presented in Figure 5.1. In the first image, the shortest path for the rescuee always passes through the grid squares with bottom left corners at $(7, 5)$, $(8, 5)$ and $(9, 5)$. When looking at the candidate rendezvous set for the realisation of edge-weights as presented in the Figure 5.1a, we see that it contains these three squares when the signal sent out by the rescuer is to go right $(m = R)$. It can also be verified that there are no non-terminal candidates rendezvous points if the rescuer signals the rescuee to go left ($m = L$). If we implemented the non-robust optimal signaling policy for this scenario we would get the result that the rescuer signals the rescuee to go right and the final rendezvous would occur at the grid square $(9, 5)$. The cost to the rescuer would be 15 for a rendezvous at this point. (Assuming cloudy grid square have a cost to traverse of 2 for the rescuer and all other squares have a cost to traverse of 1.)

Now consider another realization of edge-weights as presented in Figure 5.1b. Under this scenario, by signaling right ($m = R$) the (non-robust) candidate rendezvous points set generated for the rescuer consists only of the terminal node v_m and the initial location of the rescuee v_r. When introducing the rescue problem scenario, we made an assumption on v_r being inaccessible to the rescuer and so the rescuer cannot hope to rendezvous with the rescuee at this location. The reason for this lack of candidate rendezvous points is that there are two shortest paths with no common nodes as shown in the figure. So, if we were to run the robust optimal signaling policy over this graph, then the optimal signal to be sent would be $m = L$. The optimal cost to the rescuer would be 17.

Now, it can be noted that for there to be an alternate shortest path not passing through $(7, 5)$, $(8, 5)$ and $(9, 5)$, all the light green squares in Figure 5.1b must necessarily take maximum edge-weights. Since the cost to traverse over these squares are uniformly distributed random variables, the probability of such a realization is 0. We could in-fact have still sent the signal $m = R$, as we did for the scenario in Figure 5.1a and guaranteed a successful rendezvous at (9, 5) with an optimal cost of 16.

This illustration highlights the drawback of the robust optimal signaling policy we designed in the previous chapter. The robust approach can, in certain scenarios, be very conservative and thus, strongly sub-optimal. We may often have additional distributional information over the edge-weights and the robust optimal signaling policy makes no use of this available information. In this chapter we present an approach to reduce the over-conservativeness of the robust approach.

In arriving at the robust optimal signaling policy, the first challenge we faced was in finding the candidate rendezvous points set. We then sought to answer the question, 'What is the set of nodes that will lie on all shortest paths, for every possible edge-weight over a graph?'. Then, the natural question we can ask now is, 'What is the set of nodes that will lie on all shortest paths with a high probability, when the edge-weights vary stochastically?'

5.1 Literature Review: Shortest Path on a Graphs with Random Edge-weights

On the topic of stochastic shortest route problem, Dantzig (1963) replaced the edge weights with their expected values and solved the resulting shortest path problem with certain weights. The problem with this approach is that there exists a finite, and often large, probability that the resulting shortest path is strongly sub-optimal. With a high probability the shortest path thus computed may not in-fact be the shortest. Dantzig (1963) also presents an alternate consideration by choosing the path which minimises a weighted sum of the path cost and variance (risk) in the path cost. This is closer in spirit to what we seek, as it seeks to find the path which has not only a low expected cost but also remains close to this expected cost when edge-weights are varied. But this path may still become sub-optimal with a high probability as there may be other paths with lower cost to traverse in many cases. In our cases a failure in meeting at the correct point, or to correctly obtain what point the human is going to pass through means we have missed him altogether and now have to find him at the next feasible point. The methods above tell us what path the agent who has to traverse a graph should take if he has no way of knowing the surrounding paths. But in our scenario, we assume that the rescuee knows his precise path costs and it is the rescuer who is unaware of the path costs faced by the rescuee.

There are also some methods which take into account probabilistic results on the shortest path. Frank (1969) proposes the following condition for path optimality: For a specified k, consider the path that maximizes the probability of realizing a weight less than k as the optimal path. Another approach taken by Sigal et al. (1980) is to find the path with the greatest probability of realizing the least weight. This latter approach is the closest to what we seek. We can extend the result to sequentially find N paths that have the highest probabilities of realizing the least weight. Formally, these paths are solutions to

$$P^* = \underset{P \in \mathcal{P}_{S \to G}}{\arg\max} \prod_{P' \in \mathcal{P}_{S \to G}/\{P\}} \mathbb{P}\{L(P) \leq L(P')\}$$

The path obtained above is the shortest path with the probability obtained as the objective function above evaluated at P^*. But this approach can be computationally expensive as its evaluation requires us to iterate over all possible paths in $\mathcal{P}_{S \to G}$. Since the number of paths on a graph is exponential in the number of nodes in the graph this computation cannot be done in polynomial time. The usual approach in evaluating the probabilities of the form presented above is through Monte Carlo simulations using the known probability distribution. Motivated by this idea we will design a Monte-Carlo simulation based approach to find our the candidate rendezvous points set.

5.2 Monte-Carlo Approach

To obtain a complete stochastically optimal policy we have three considerations to make.

1. We first need to obtain the the set of stochastic candidate rendezvous points i.e. the set of nodes that the rescuee will pass through with a high probability.

2. Next, for any candidate rendezvous point we need to check feasibility of rendezvous. In other words, we need to evaluate whether the rescuer can indeed reach this point before the rescuee does and thus allow for a successful rendezvous. We may choose to pose this question in a stochastic setting as well. The question to be asked then is, 'Among the candidate rendezvous points, which points can the rescuer reach before the rescuee with a high probability?'.

3. The third challenge we face is in the computation of the objective function when the edge-weights are stochastic.

All these issues can be mitigated by an complete Monte-Carlo method approach to the problem. What we mean by this is that, we can sample over the distribution of edge-weights for both the rescuee and the rescuer and evaluate the optimal signaling policy as presented in Chapter 2. Having obtained the optimal signal to every sampled edge-weight vector, we can simply pick the signal with the highest empirical probability of being optimal.

We can make an observation here that addressing the first challenge posed above requires us to compute only one set of shortest paths for each sampled edge-weight. We can then compute the empirical probability of any node lying on shortest paths by taking a large number of samples. This will be explored more in Section 5.2.1. For both the second and the third challenge, we need to compute shortest paths multiple times; once each for every node in the candidate rendezvous set obtained for a given edge-weight sample. As the size of the grid increases, the size of the candidate rendezvous set grows linearly with the dimensions of the grid. Computing these large number of shortest paths for a large number of sampled edge-weights can prove very challenging for on-board computing. As a solution to this, we will present a hybrid approach which combines the reduced conservativeness of the stochastic approach while maintaining the reduced computational complexity of the robust approach from Chapter 4. The second and the third problem posed above will be solved using a robust approach we present in Section 5.3.

As a first step, we address the problem of finding the set of stochastic candidate rendezvous points. Monte-Carlo based simulations for this requires us to treat the edge-weights over the graph as stochastic. For different values of these edge-weights, we can then compute the shortest paths and find the set of points that lie on these shortest paths. We then seek to find the set of points which lie on some such shortest path with high probability.

Going forward, we assume that the edge weights w^r_{ij} and w^R_{ij} are stochastic bounded random variables: $\underline{w}^r_{ij} \leq w^r_{ij} \leq \overline{w}^r_{ij}$ and $\underline{w}^R_{ij} \leq w^R_{ij} \leq \overline{w}^R_{ij} \; \forall \; ij \in \mathcal{E}$. As before, we designate the Cartesian product $\Pi_{ij \in \mathcal{E}}[\underline{w}^r_{ij}, \overline{w}^r_{ij}]$ as Ω^r and $\Pi_{ij \in \mathcal{E}}[\underline{w}^R_{ij}, \overline{w}^R_{ij}]$ as Ω^R. We can then consider the random vectors $w^r = \{w^r_{ij}\}_{ij \in \mathcal{S}}$ and $w^R = \{w^R_{ij}\}_{ij \in \mathcal{S}}$ over the supports Ω^r and Ω^R respectively.

We will not make any assumptions on the independence of these edge-weights. On the contrary, it is quite likely that edge-weights on edges in a neighbourhood are correlated. On a realistic terrain it is likely that hills would occur together as a range. Similarly, it is likely that there exists a valley between two ranges of hills. These correlations can be captured as a joint probability distribution over the edge-weights. Let F denote the joint probability of the edge-weights over the graph. We will refer to probabilities using notation $\mathbb{P}$ and P will be reserved to denote paths as before.

5.2.1 Obtaining the candidate rendezvous set

Definition 7 *Let $P^*_m(w^r)$ denote the shortest path taken by the rescuee to the target goal v_m when the realised edge-weights are w^r. We then define a candidacy index for each node as $J : \mathcal{V} \to \mathbb{R}$ as,*

$$J(v) = \mathbb{P}(v \in P^*_m(w^r))$$

We will evaluate this probability using Monte-Carlo (MC) methods. MC methods use random samplings from a known distribution to numerically compute various statistical quantities. We can use a similar approach in computing the candidacy index as defined in Definition 7.

A challenge to be addressed in computing the candidacy indices is the presence of multiple shortest paths for a given vector of edge-weights w^r. When the shortest path is unique $\mathbb{P}(v \in P^*_m(w^r_i))$ is easier to compute, but we need to make some additional considerations if there are multiple shortest paths for some w^r. In Chapter 2 and 4, we defined the candidate rendezvous points such that every point in this set lies on every shortest path when there are multiple shortest paths. In the stochastic approach, we will address this issue by assuming that when there are multiple shortest paths, each path is chosen at random with a uniform equal probability. Then, given the edge-weights, the shortest path taken by the rescuee is a uniform random variable over the set of all shortest paths. For example, if there exist two paths with equal shortest path costs from v_r to v_m, then each will be chosen in one sampling with a probability of 0.5.

To capture this additional stochasticity in the choice of shortest path we will treat $P^*_m(w^r)$ as a random variable in itself. We first make the observation that any probability can be written out as an expectation over an indicator variable as,

$$J(v) = \mathbb{P}(v \in P^*_m(w^r)) = \mathbb{E}_{w^r, P^*_m}[\mathbb{1}_{v \in P^*_m(w^r)}]$$

where the expectation is taken with respect to both the distribution F over the edge-weights w^r and the uniform distribution over the choice of shortest path P^*_m. We then present the following

Definition 8 *Let $\mathcal{P}^*_m(w^r)$ be the set of all shortest paths from v_r to v_m when the edge-weights over the graph is w^r. We define the candidacy index estimator*

$$g(v, w^r) = \frac{|P : v \in P, P \in \mathcal{P}^*_m(w^r)|}{|\mathcal{P}^*_m(w^r)|}.$$

We then make the following claim,

Claim 4

$$\mathbb{E}_{w^r}[g(v, w^r)] = J(v)$$

Proof for this claim is presented in Appendix A.4. Inspired by Monte-Carlo methods, we can then approximate the candidacy index as,

$$J(v) \approx \tilde{J}(v) \triangleq \frac{1}{K} \sum_{i=1}^{K} g(v, w^r_i)$$

where K is a free parameter determining the number of samples we take to approximate $J(v)$ and w^r_i is the i^{th} sampled edge-weight vector. We can then define our stochastic candidate rendezvous set using this definition of candidacy index,

Definition 9 *For some $\epsilon > 0$, we define the stochastic candidate rendezvous points set as,*

$$\tilde{\mathcal{X}}^{\epsilon}_m = \{v \in \mathcal{V} : \tilde{J}(v) > 1 - \epsilon\}$$

Now to find this stochastic candidate rendezvous set we need to compute the candidacy indices for all nodes. We know that the shortest paths can be computed using Dijkstra's algorithm in $O(m + n \log n)$. The primary computational complexity in obtaining the candidacy index then arises from the need for a large number of samples. For a grid of size N, we have the number of edges in the order of $4N$. This would correspond to generating $4N$ edge-weight samples in each sampling run. This problem can be alleviated by making a useful observation from the proof of Algorithm 1 presented in Appendix B. Proposition 5 tells us that at the termination of Algorithm 1, the graph G^k we obtain contains all possible shortest paths in the graph G. Since this algorithm is polynomial time, we can efficiently obtain the subgraph which necessarily contain all shortest paths. We then only need to pick edge-weights from the distribution F for edges in this subgraph to compute the candidacy index for nodes. We also know then that the candidacy index for all nodes outside this sub-graph is zero.

The question now remains; For a given ϵ, what is the minimum number of samples K to be picked such that the candidacy index estimate $\tilde{J}(v)$ is a 'good' approximation of the true candidacy index $J(v)$. Requiring a small number of samples here is key in our ability to compute the final optimal signaling policy on board the UAV.

5.2.2 Finite Time Guarantees for Monte-Carlo Method

In this section, we seek to present a bound on the number of samples required to compute the estimate for the candidacy index for each node with reasonable accuracy. This first requires us to show that the function

$$\tilde{J}(v) = \frac{1}{K}\sum_{i=1}^{K} g(v, w_i^r)$$

is measurable. Here w_i^r is the i^{th} sampled edge-weight vector. It suffices to show that the function, $g(v, w^r)$ is measurable for every w^r. The arguments for measurability of this function are presented in Appendix C. Having established that the candidacy index is measurable, we will set out to find the probabilistic bounds we seek.

We will first make an assumption that each edge-weight vector w^r is picked i.i.d from its distribution F over Ω_r. By Law of Large Numbers, we know that as the number of samples, K, increases the estimate of candidacy index $J(v)$ converges to the true candidacy index in probability $J(v)$ as

$$\tilde{J}(v) \xrightarrow{p} \mathbb{E}[g(v, w^r)] = J(v) \quad \forall v \in \mathcal{V}. \quad \text{(By Claim 4).}$$

Then for any $\epsilon > 0$ we have,

$$P\bigg((w_1^r, \ldots, w_K^r) \in \Omega^{rK} : |\tilde{J}(v) - J(v)| > \epsilon\bigg) \to 0.$$

In more rigorous terms, for every $\delta > 0$ we can then find a $K \in \mathbb{N}$ such that for every $k > K$ we have,

$$P\bigg((w_1^r, \ldots, w_k^r) \in \Omega^{rk} : |\tilde{J}(v) - J(v)| > \epsilon\bigg) < \delta. \tag{5.1}$$

We now will attempt to establish a relation between the number of samples required K and the upper-bound on the probability δ. To this end, we make use of the following

Theorem 1 ***(McDiarmid's Inequality)*** *Let $X^n = (X_1, X_2, \ldots, X_n) \in \mathcal{X}^n$ be independent random variables and let $f : \mathcal{X}^n \to \mathbb{R}$ be a finite difference mapping i.e. f satisfies,*

$$|f(x_1, \ldots, x_i, \ldots, x_n) - f(x_1, \ldots, x_1', \ldots, x_n)| \leq c_i \quad \forall x_i, x_i', x_1 \ldots x_n$$

then

$$\mathbb{P}(f - \mathbb{E}[f] \geq \epsilon) \leq \exp \frac{-2t^2}{\sum_{i=1}^n c_i^2}$$

Through some abuse of notation we will explicitly list the dependence of $\tilde{J}$ on the sampled edge-weights $w_1^r, \ldots, w_K^r$ as,

$$\tilde{J}(v, w_1^r, \ldots, w_K^r) = \frac{1}{K} \sum_{i=1}^K g(v, w_i^r)$$

We can now verify the finite difference property of this function,

$$\begin{aligned} |\tilde{J}(v, w_1^r, \ldots, w_i^r, \ldots, w_K^r) - \tilde{J}(v, w_1^r, \ldots, w_i^{r\prime}, \ldots, w_K^r)| &= \frac{1}{K} |g(v, w_i^r) - g(v, w_i^{r\prime})| \\ &\leq \frac{1}{K} \end{aligned}$$

because $0 \leq g(v, w_i^r) \leq 1$ from the definition of g. Then as a direct application of McDiarmid's inequality we have,

$$\mathbb{P}(\tilde{J}(v) - J(v) \geq \epsilon) \leq exp(-2\epsilon^2 K).$$

Setting $K > -\frac{\log \delta}{2\epsilon^2}$ we have the result from Equation 5.1. We have presented a lower bound on the number of samples required to approximate the candidacy index up to a desired confidence level. We are now equipped to compute the stochastic candidate rendezvous set and can now move onto presenting the optimal signaling policy using this set.

5.3 Stochastic Optimal Signaling Policy

In the preceding section we computed the stochastic candidate rendezvous set i.e. 'the set of points that lie on some shortest path with a high probability when the edge-weights are varied stochastically.' In the beginning of section 5.2 we indicated that after employing Monte-Carlo methods to obtain the candidate rendezvous sets, we will use the robust approach developed in Chapter 4 to design the complete stochastic optimal signalling policy. To do this, we simply replace the robust candidate rendezvous points set in Proposition 2 with the stochastic candidate rendezvous points set we obtained here.

We then propose the following optimisation to be solved by the rescuer to arrive at the optimal signal to be sent when non-terminal feasible points exist.

Proposition 3 *Any stochastic feasible rendezvous point $v_x \in \tilde{\mathcal{X}}_m$ for (4.8) satisfies,*

$$\phi^*_{R,\max}(v_x) - k_v \phi^*_{r,\min}(v_x) \leq 0 \tag{5.2}$$

Where, $\phi^*_{R,\max}(v_x)$ and $\phi^*_{r,\min}(v_x)$ are obtained as 4.10 *and* 4.11 *respectively. The stochastic robust counterpart to the optimisation problem presented in* (2.6) *is then given by*

$$\min_{m \in \mathcal{M}} \min_{v_x \in \tilde{\mathcal{X}}_m} k_1 \phi^*_{R,\max}(v_x) + k_2 \phi^*_{r,\max}(v_x) \tag{5.3}$$

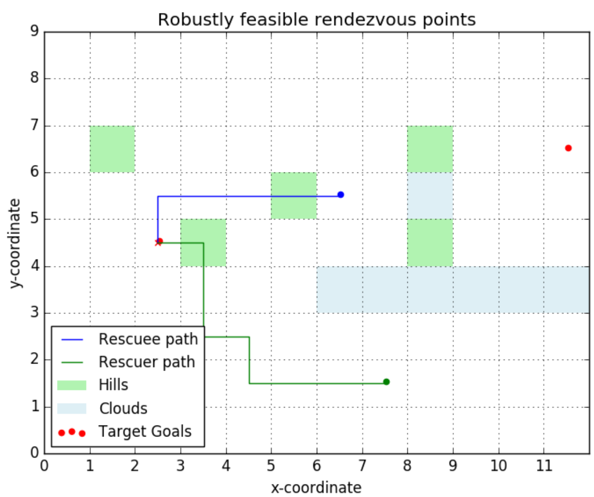
subject to (4.9), where, $\phi^*_{r,\max}(v_x)$ can be obtained by replacing $\underline{w}^r_{ij}$ in (4.11) with $\overline{w}^r_{ij}$. □

Once again in the case where there are no feasible rendezvous points the optimal signal to be sent can be arrived at by comparing the cost to rendezvous at the terminal goals. In the following section we will present simulations results which implement this stochastic optimal signaling policy.

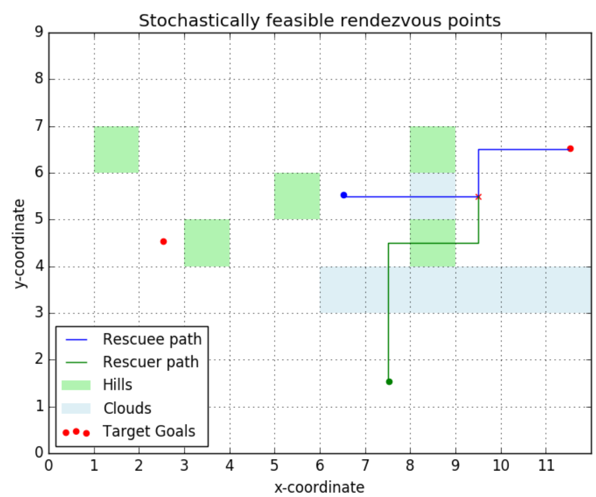
5.4 Simulation results

5.4.1 Stochastic Optimal signaling Policy in Action

Figure 5.2 presents a topography with two different approaches to computing the candidate rendezvous points set. Figure 5.2a implements the fully robust solution to the rendezvous problem as presented in Chapter 4. For the rescuee the cost-to-traverse over hills (green shading) is a uniform random variable supported over $[2.5, 3]$ and cost-to-traverse other regions is a uniform random variable over $[1, 1.5]$. Likewise, for the rescuer, the cost-to-traverse over clouds (blue shading) is a uniform random variable supported over $[2.5, 3]$ and cost-to-traverse other regions is a uniform random variable over $[1, 1.5]$. For the given topography, it can be checked that the robust candidate rendezvous points set can be obtained as $\hat{X}_R = \{x_H, x_R\}$ and $\hat{X}_L = \{x_H, x_L\}$ in signaling $m = R$ and $m = L$ respectively. Since, x_L is spatially closer to both the rescuee and the rescuer we end up with $m = L$ as the optimally signal to be sent. The result rendezvous is illustrated in Figure 5.2a.



(a) Robust candidate rendezvous points.



(b) Candidate rendezvous points exist with a high probability.

Figure 5.2: Over the same topographical layout we observe that the optimal signal to be sent changes with the robustness criteria we choose.

As indicated in the previous section we only introduce stochasticity based relaxation of robustness in arriving at the candidate rendezvous points set. Whether or not these points are feasible is still checked in a conservative robust manner. Considering Figure 5.2b, we see that the path taken by the rescuee passes 'South' of the 'hill' at $[6, 8]$ (referenced by bottom left corner). But it is not hard to create a case where the shortest path to the right target goal passes 'North' of this hill. We created such a case in Figure 5.1 in the introduction of this chapter. We also saw that the likelihood of such a case being realized is 0. Agreeing with this analytic result, from Monte Carlo based computations, we observe that such a path 'North' of the hill is the shortest path with a small probability. In other words, with a high probability (we picked a threshold of 0.9), when edge-weights w^r are picked from a distribution defined for this topography, the shortest paths passes between the two 'hills' at $[6,8]$ and $[4,8]$. Thus, with a high probability the optimal message to be sent is $m = R$ as the rendezvous point (cross) picked in Figure 5.2b is closer than going to x_L for both the rescuee and the rescuer.

5.4.2 Simulation based validation of Monte-Carlo FTG results

In Section 5.2.2, we claimed that the number of iterations n required in the Monte Carlo approach to ensure that our empirical estimate of the expectation $\tilde{J}(v)$ to converge to the true expectation $J(v)$ with a high probability satisfies,

$$\mathbb{P}(\tilde{J}(v) - J(v) \geq \epsilon) \leq exp(-2\epsilon^2 n) < \delta \tag{5.4}$$

We arrived at this statistical bound using the McDiarmid's inequality. Now, we also wish to see how good this bound is in practice. For the purpose of this section we will assume that at most one edge has uncertain edge weights. The scenario we will consider is presented in Figure 5.3.

In Figure 5.3, when the dark green square has a cost-to-traverse of below 2 ,the shortest path to the right goal for the rescuee passes through this square. For values of cost-to-traverse between $[2,4]$ the shortest path no longer passes through this dark green square. Since the cost-to-traverse over this region varies uniformly between $[1,4]$ there is a probability of $\frac{1}{3}$ that this the shortest path traverse this region. We will refer to this square hence forth as a node using the coordinates of its bottom-left corner $[7,3]$. The candidacy index for this node $[7,3]$ is then $\frac{1}{3}$ as the probability of a shortest path through this node is $\frac{1}{3}$. We can now verify both the convergence of the estimated candidacy index to the true index and also the number of samples required to achieve the convergence with reasonable probability.

We are interested in two things. First, we want to see how quickly the empirical estimate $\hat{J}(v)$ converges to the true $J(v)$ with increasing number of iterations. The Hoeffding-like bound obtained from McDiarmid's inequality gives us a probabilistic guarantee, but not a deterministic guarantee, on how quickly the estimate converges to the true value. It only tells us that, with a high-probability, the estimate will converge to the true value when the number of iterations are greater than a given n. Figure 5.4 shows the evolution of the estimate $\hat{J}(v)$ with increase in number of iterations.

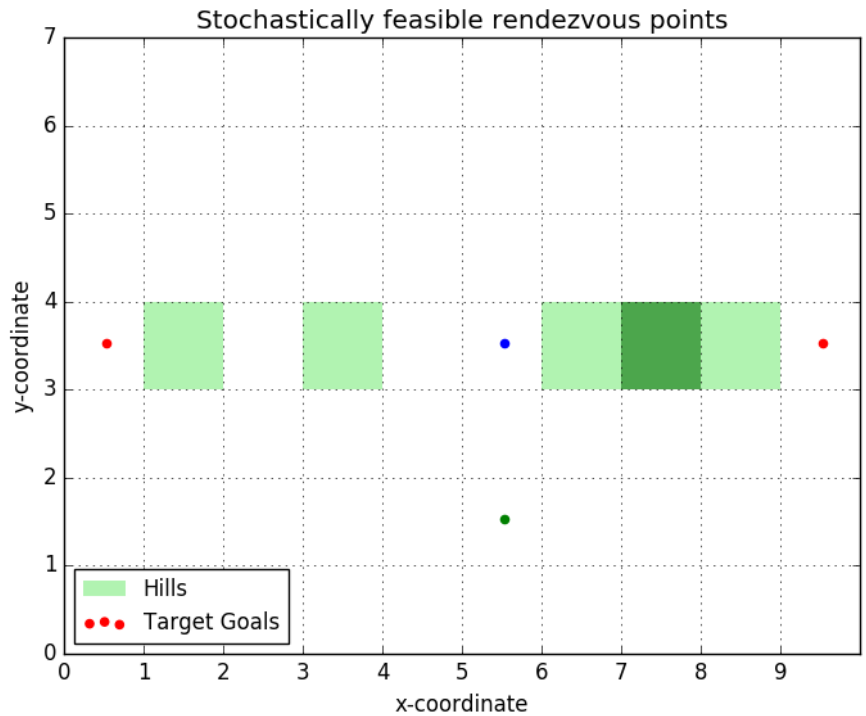


Figure 5.3: The weight to travel over the dark green grid square with the bottom left corner at $(7, 3)$ takes edge weights in the range $w \in [1, 4]$. All light green grid squares have cost-to-traverse of 1.5. All uncolored squares have a cost to traverse of 1.

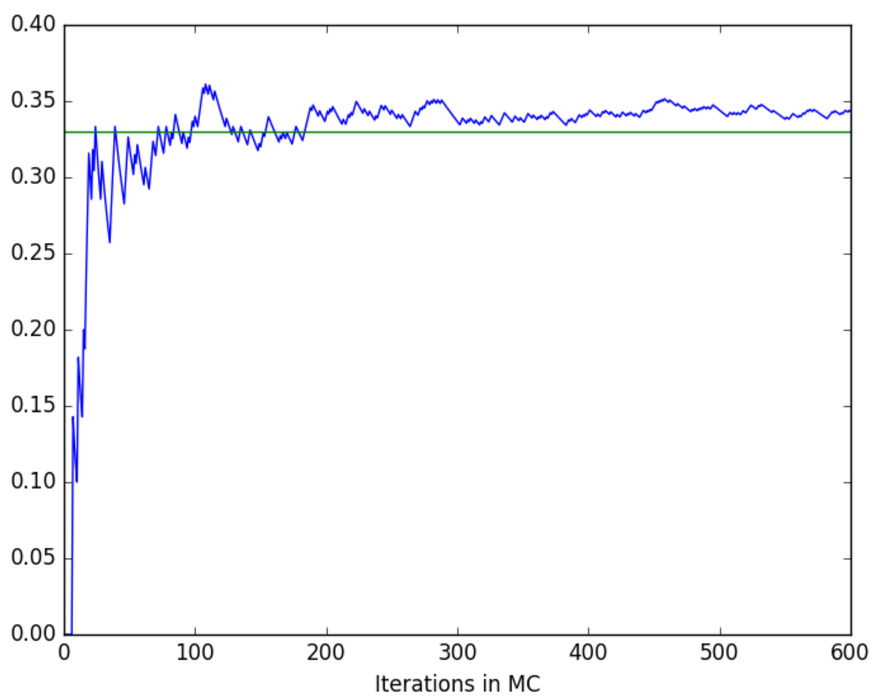


Figure 5.4: The estimate $\hat{J}(v)$ evolution with increasing number of iteration. We see convergence close to the true value $J(v) = 0.33$

Figure 5.5 shows the final estimate $\hat{J}(v)$ after multiple full runs of the MC algorithm. We wish to see in how many runs the final estimate $\hat{J}(v)$ after n iterations falls outside the ϵ ball around the true expectation $J(v)$. For $\delta = 0.05$ with $\epsilon = 0.05$ we see that we need to have $N \approx 600$ to satisfy the inequality in Eq. 5.4. We can see that the convergence in Figure 5.4 happens in much fewer iterations than what we predicted using the McDiarmid's inequality. This is to be expected as performance guarantees based on McDiarmid's inequalities give us very loose bounds. And from Figure 5.5 we see that we exit the $\epsilon = 0.05$ ball around the true mean $J(v) = 0.33$ only once in 100 runs, i.e with an empirical probability of 0.01.

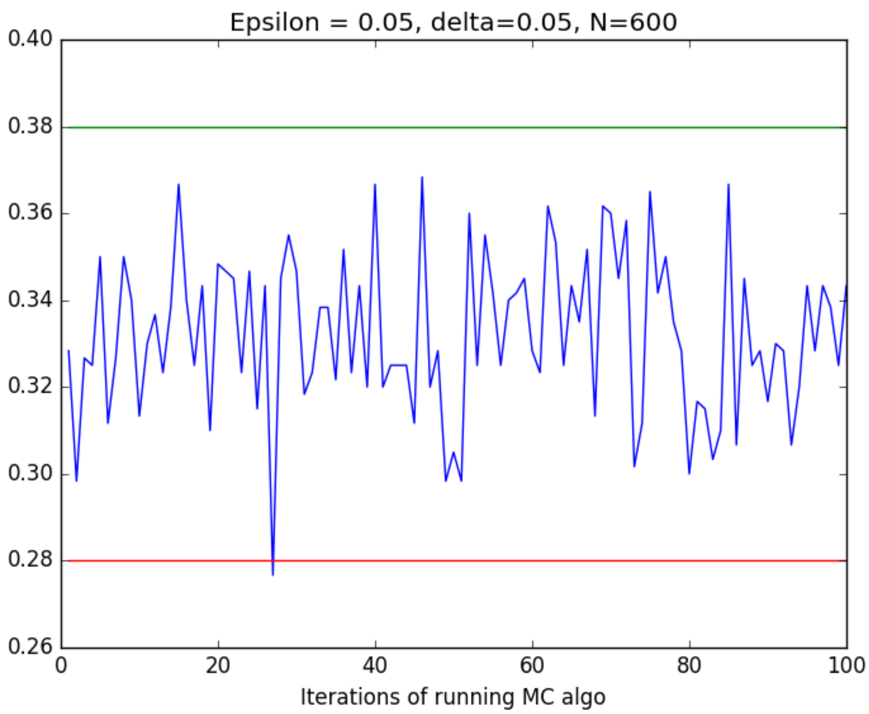


Figure 5.5: The estimate $\hat{J}(v)$ at the end $N = 600$ iterations run 100 times.

From these simulations, we can conclude that not only is the the stochastic approach effective in reducing the conservativeness of the robust optimal signaling policy, but also that we can obtain the stochastic optimal signal using a reasonable number of samples.

Chapter 6

Conclusions and Future Work

We can now take a step back and take a broader view on the work presented so far. We began our work with a desire to exploit implicit signaling capabilities present in robots. But, our work is in no way limited to implementation in these autonomous agents. It is not hard to imagine that the entire rescue scenario we presented could be played out between a human rescuee and a human rescuer flying in a disaster relief aircraft. The signaling methods we assume a UAV to posses are also present in a human controlled aircraft, be it taking elaborate banked turns to indicate intent to turn or the use of flashing lights to indicate directions. While we did not explicitly design the trajectories or signaling methods we mention in passing here, we justified their existence by citing studies into legible and predictable robotic motion from the HRI community. While these studies seek only to convey a robot's intent to humans we go a step further and exploit these intent expression capabilities to influence human behaviour.

A natural paradigm to study this 'influencing' behaviour of autonomous agents is as a Stackelberg game played between the rescuer and the human. The rescuer takes the role of the leader and send out a message to the human who is the follower. The utility of the message to the rescuer lies solely in the fact that the human responds to the sent message by taking a particular action. We initially set out to find an optimal signaling scheme assuming the fact that the game being played out was completely deterministic given a message. This assumption implied that the rescuer could predict the human's precise best response to every message it chose to send and then the choice of message to be sent was simply obtained as a finite space search.

The assumption we made on rescuer possessing perfect knowledge of the rescuees action is a strongly unrealistic one. There are multiple issues that can crop up in the event of our assumption being violated. Our assumption about the rescuer's action is two fold. Firstly, it assumes the perfect knowledge of the state of the world which factors into the human's decision making process. Any ambiguity in the state of the world immediately translates to an ambiguity in the rescuer's knowledge of the human's action. The second assumption we implicitly made is that given the state of the world, the rescuer has precise knowledge of the human's best response to each message it sends. It may often be the case that the human's best response is not deterministic given the state of the world and the message. We will leave the discussion of this second issue to later and focus on the first one for now.

We presented multiple approaches to address the issue of the rescuer having insufficient knowledge of the state of the world. Initially, we took a very conservative approach and sought the messaging policy which minimised the cost in the worst-case scenario of state of the world. With this approach we were guaranteed to have achieve our objective with an upper bound on the incurred cost. In arriving at a robust solution to our problem, we also presented a novel algorithm to guarantee successful rendezvous between two non-communicating players traversing a graph with uncertain edge-weights. It is easy to find applications for this algorithm in other fields like operations research and network science. As an example, consider a variant of the last mile problem where the supply vehicle and the deliver vehicle have to meet over a road network with uncertain traffic congestion.

The robust approach was well suited to our scenario, as, in a success critical scenario like rescue operations we could to guarantee a successful rescue. But at the same time, it may also be the case that a similar scenario was time-critical in addition to being success-critical. In an effort to trade some robustness for optimality, we sought to look at solutions that exploit additional information about the state of the world in arriving at the optimal signal. In particular, we assumed some distributional knowledge over the state of the world and employed Monte-Carlo methods to arrive at a statistical solution to the optimal signaling policy. We also presented some performance guarantees on the number of samples required to achieve a successful rescue with a high confidence.

We will now return to the issue of finding the optimal signal to be sent by the rescuer, in the case where the rescuee's action is not a deterministic function of the state of the world and the message. The rescuee, being a human, cannot be expected to have precise computational capabilities to compute the best response given the state of the world. While there is a subtle difference between choosing a sub-optimal response to a message because of error or lack of computing ability and choosing a sub-optimal response by randomizing over set of available responses, we can model the former scenario as a case of the latter. We can then once again study this problem as a Stackelberg game where the rescuee is playing mixed strategies. While this idea hasn't been addressed in the work presented thus far, we intend to study this problem in more detail in the future. Specifically, we will enlarge the rescuee's response set from the set of shortest paths to the set of almost shortest paths. Byers and Waterman (1984) presented the first approach to finding almost shortest paths and later Eppstein (1999) presented a more efficient approach to arrive at the same set of almost shortest paths. The idea behind using almost shortest paths is that, while a human cannot compute precise path costs, their intuitive guess for which is path is shortest isn't very far from the true shortest path as shown by Zhu and Levinson (2015).

In the work we presented thus far, we only considered a single signal being sent by the rescuer. We also did not analyse the scenario where the rescuer fails to meet the rescuee after signaling. While this possibility is non-existent when there is no uncertainty in edge-weights or when there is uncertainty and the rescuer implements the robust optimal signaling policy, there is always a small probability of failure in implementing the stochastic optimal signaling policy. We can then consider a multi-stage Stackelberg game wherein the rescuer signals its intent to the rescuee multiple times. This is another avenue for future work.

At the end of the day, any model of human behaviour is subject to question. As a validation of the efficacy of our signaling policy, it would it interesting to see the likelihood of its success in an experimental setting. We plan to carry out some experimental work involving human subjects to evaluate the performance of our signaling policy.

Bibliography

Alpern, S. and S. Gal (2003). *The Theory of Search Games and Rendezvous*. Springer: Boston, MA.

Ben-Tal, A., L. El Ghaoui, and A. Nemirovski (2009). *Robust Optimization*. Princeton University Press: Princeton, New Jersey.

Bertsimas, D. and M. Sim (2003). "Robust discrete optimization and network flows". In: *Mathematical Programming* 98, pp. 49–71.

Byers, T. H. and M. S. Waterman (1984). "Determining All Optimal and Near-Optimal Solutions when Solving Shortest Path Problems by Dynamic Programming". In: *Operations Research* 32 (6), pp. 1381–1384.

Dantzig, G. B. (1963). *Linear Programming and Extensions*. Princeton University Press: Princeton, New Jersey.

Dijkstra, E.W. (1959). "A note on two problems in connexion with graphs". In: *Numerische Mathematik*, pp. 269–271.

Dragan, A. D., S. Bauman, et al. (2015). "Effects of Robot Motion on Human-Robot Collaboration". In: *10th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 51–58.

Dragan, A. D., K.C.T. Lee, and S. S. Srinivasa (2013). "Legibility and Predictability of Robot Motion". In: *10th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 301–308.

Eppstein, D. (Feb. 1999). "Finding the k Shortest Paths". In: *SIAM J. Comput.* 28(2), pp. 652–673. DOI: 10.1137/S0097539795290477.

Etro, F. (2013). "Stackelberg, Heinrich von: Market Structure and Equilibrium". In: *Journal of Economics* 109, pp. 89–92.

Frank, H. (1969). "Shortest Paths in Probabilistic Graphs". In: *Operations Research* 17 (4), pp. 583–599.

Li, Y. et al. (2016). "A Framework of Human–Robot Coordination Based on Game Theory and Policy Iteration". In: *IEEE Transactions on Robotics* 32, pp. 1408–1418.

Rabinovich, Z. et al. (2015). "Information disclosure as a means to security". In: *International Conference on Autonomous Agents and Multiagent Systems* 2, pp. 645–653.

Sigal, C. E., A. A. B. Pritsker, and J. J. Solberg (1980). "The Stochastic Shortest Route Problem". In: *Operations Research* 28 (5), pp. 1122–1129.

Szafir, D., B. Mutlu, and T. Fong (2014). "Communication of Intent in Assistive Free Flyers". In: *ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 358–365.

Szafir, D., B. Mutlu, and T. Fong (2015). "Communicating Directionality in Flying Robots". In: *ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 19–26.

Tambe, M. (2011). *Security and Game Theory: Algorithms, Deployed Systems, Lessons Learned.* Cambridge University Press: New York, NY, USA.

Yua, H., H. E. Tsengb, and R. Langaria (2018). "A human-like game theory-based controller for automatic lane changing". In: *Transportation Research Part C* 88, pp. 140–158.

Zhu, S. and D. Levinson (2015). "Do people use the shortest path? An empirical test of Wardrop's first principle." In: *PLoS One* 10 (8), pp. 1381–1384. DOI: `10.1371/journal.pone.0134322`.

Appendix A

Proofs for Results

A.1 Proof for Lemma 1

Every node in the obtained graph lies on the shortest path from the source node to the sink node. By principle of optimality, the shortest path from each node to the sink node can also be found within the same graph. Now consider a potential function over nodes, with the value of the potential function at a node being the shortest path length from that node to the sink node. We can arrange these nodes then in decreasing (non-strict) order of potentials with the sink node having potential 0 and the source node having potential equal to the shortest path length between the source and the sink.

Since all edge weights are positive, any edges in shortest path between the source and the sink can only connect nodes with higher potential to nodes with (strictly) lower potential. We have effectively constructed a topological ordering for nodes in the given graph. Thus, we know that the graph constructed by taking the nodes and edges from all shortest paths is DAG.

Since every edge in the graph obtained above forms a part of some shortest path(s), necessarily the cost to traverse each edge is equal to the difference between potentials of the nodes connected by it. Then every path consisting of such edges is a shortest paths.

A.2 Justifying the inequality 3.4

The threshold velocities for each node can be written out as,

$$V_R(l_i) = \begin{cases} \frac{i+4}{2(i-1)} & i \text{ is even} \\ \frac{i+4}{2i-1} & i \text{ is odd} \end{cases}$$

$$V_R(r_i) = \begin{cases} \frac{i+4}{2i-1} & i \text{ is even} \\ \frac{i+4}{2(i-1)} & i \text{ is odd} \end{cases}$$

We can observe right away that for an even i,

$$\frac{i+4}{2(i-1)} > \frac{i+4}{2i-1} (\because\ 2i-1 > 2i-2)$$
$$\implies V_R(l_i) > V_R(r_i)$$

Similarly, we get $V_R(r_{i+1}) > V_R(l_{i+1})$. Now we wish to show $V_R(r_i) > V_R(r_{i+1})$.

$$\begin{aligned} V_R(r_i) - V_R(r_{i+1}) &= \frac{i+4}{2i-1} - \frac{i+5}{2i} \\ &= \frac{5-i}{2i(2i-1)} < 0 \quad \forall i \geq 5 \end{aligned}$$

Similarly we can show that $V_R(l_{i+1}) > V_R(l_{i+2}) \quad \forall i \geq 4$.

A.3 Proof for Claim 3

Let $\xi_R^*(v_a, v_b) = \arg\min_{P \in \mathcal{P}_{v_a \to v_b}} \phi_R(P)$ denote the shortest path on the graph between any two nodes v_a and v_b for the rescuer. Likewise $\xi_r^*(v_a, v_b)$ denotes the shortest path for the rescuee over the graph between the two nodes.

$$\begin{aligned} \frac{\phi_R^*(v_{x,n})}{\phi_r^*(v_{x,n})} &= \frac{\phi_R(\xi_R^*(v_R, v_{x,n}))}{\phi_r(\xi_r^*(v_r, v_{x,n}))} \\ &\leq \frac{\phi_R(\xi_R^*(v_R, v_{x,m})) + \phi_R(\xi_R^*(v_{x,m}, v_{x,n}))}{\phi_r(\xi_r^*(v_r, v_{x,m})) + \phi_r(\xi_r^*(v_{x,m}, v_{x,n}))} \end{aligned}$$

Since $v_{x,m}$ also lies on the shortest path for the rescuee we have $\phi_r(\xi_r^*(v_r, v_{x,n})) = \phi_r(\xi_r^*(v_r, v_{x,m})) + \phi_r(\xi_r^*(v_{x,m}, v_{x,n}))$. Since $\xi_R^*(v_R, v_{x,n})$ is the shortest path for the rescuer to $v_{x,n}$ we have the triangle inequality $\phi_R(\xi_R^*(v_R, v_{x,n})) \leq \phi_R(\xi_R^*(v_R, v_{x,m})) + \phi_R(\xi_R^*(v_{x,m}, v_{x,n}))$.

$$\begin{aligned} \text{Hence, } \frac{\phi_R^*(v_{x,n})}{\phi_r^*(v_{x,n})} &\leq \frac{\phi_R(\xi_R^*(v_R, v_{x,m})) + \phi_R(\xi_r^*(v_{x,m}, v_{x,n}))}{\phi_r(\xi_r^*(v_r, v_{x,m})) + \phi_r(\xi_r^*(v_{x,m}, v_{x,n}))} \quad (\because \xi_R^* \quad \text{is the shortest path}) \\ &\leq \frac{\phi_R(\xi_R^*(v_R, v_{x,m})) + w_{\max}^R(m-n)}{\phi_r(\xi_r^*(v_r, v_{x,m})) + w_{\min}^r(m-n)} \quad (\because w_{ij}^R \leq w_{\max}^R, w_{ij}^r \geq w_{\min}^r) \\ &\leq \frac{k_v \phi_r(\xi^*(v_r, v_{x,m})) + k_v w_{\min}^r(n-m)}{\phi_r(\xi^*(v_r, v_{x,m})) + w_{\min}^r(n-m)} \quad \text{(Assumption 6)} \\ &= k_v \end{aligned}$$

□

A.4 Proof for Claim 4

We wish to show that $\mathbb{E}_{w^r}[g(v, w^r)] = J(v)$. We will drop the w^r arguments for shortest path taken by the rescuee $P_m^*(w^r)$ for the sake of brevity. By the definition of candidacy index we have,

$$\begin{aligned} J(v) &= \mathbb{E}_{w^r, P_m^*}\left[\mathbb{1}_{v \in P_m^*}\right] \\ &= \mathbb{E}_{w^r, P_m^*}\left[\sum_{P \in \mathcal{P}_m^*(w^r)} \mathbb{1}_{v \in P} \cdot \mathbb{1}_{P = P_m^*}\right] \end{aligned}$$

where $\mathcal{P}_m^*(w_r)$ denotes the set of shortest paths from v_r to v_m for edge-weight w_r. Then by tower property of conditional expectation we have,

$$J(v) = \mathbb{E}_{w^r}\left[\sum_{P \in \mathcal{P}_m^*(w^r)} \mathbb{1}_{v \in P} \cdot \mathbb{E}_{P_m^*}[\mathbb{1}_{P = P_m^*} | w_r]\right].$$

We made the assumption that when there are multiple shortest paths, one is chosen a random with a uniform probability over the set of all shortest paths. Given w^r the set of shortest paths is a deterministic set. So we have,

$$\begin{aligned} J(v) &= \mathbb{E}_{w^r}\left[\sum_{P \in \mathcal{P}_m^*(w^r)} \mathbb{1}_{v \in P} \frac{1}{|\mathcal{P}_m^*(w_r)|}\right] \\ &= \mathbb{E}_{w^r}[g(v, w^r)] \end{aligned}$$

□

Appendix B

Proof for Algorithm 1

Definition 10 *(Graph Union) For two graphs $\mathcal{G}_1 = (\mathcal{V}_1, \mathcal{E}_1)$ and $\mathcal{G}_2 = (\mathcal{V}_2, \mathcal{E}_2)$, the graph union is obtained as the new graph $\mathcal{G}_1 \cup \mathcal{G}_2 = (\mathcal{V}_1 \cup \mathcal{V}_2, \mathcal{E}_1 \cup \mathcal{E}_2)$*

Definition 11 *(Graph Compliment) Let $\mathcal{H} = (\mathcal{V}_H, \mathcal{E}_H)$ be a sub-graph of $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, then we will define the graph compliment of $\mathcal{H}$ with respect to $\mathcal{G}$ as,*

$$\mathcal{G}/\mathcal{H} = (\mathcal{V}, \mathcal{E}/\mathcal{E}_H)$$

We will begin by re-introducing some of the notations used in presenting the algorithm. In doing so, we will drop the subscripts m and superscript r for increased readability. $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ denotes the graph representation of the rendezvous topology. Then, sub-graph $\mathcal{G}^{*k} = (\mathcal{V}^{*k}, \mathcal{E}^{*k})$ denotes the acyclic digraph containing the shortest paths in the k^{th} iteration of the algorithm. We can make the claim of acyclicity since all edge-weights in our graph $\mathcal{G}$ are assumed positive. We define a new digraph $\mathcal{G}^k = (\mathcal{V}^k, \mathcal{E}^k)$ obtained as a graph union in each iteration as,

$$\mathcal{G}^k = \bigcup_{i=1}^{k} \mathcal{G}^{*k} \tag{B.1}$$

$\{w_{ij}^k\}$ denotes the set of edge weights at the k^{th} iteration of the algorithm. Each edge-weight in $\{w_{ij}^k\}$ can be obtained as,

$$w_{ij}^k = \begin{cases} \overline{w}_{ij} & \text{if } (ij) \in \mathcal{E}^{k-1} \\ \underline{w}_{ij} & \text{if } (ij) \in \mathcal{E}/\mathcal{E}^{k-1} \end{cases} \tag{B.2}$$

Claim 5 *Algorithm 1 return a set of nodes $\hat{\mathcal{X}} \subseteq \mathcal{V}$ such that every shortest path in the graph $\mathcal{G}$ for any set of edge-weights $\{w_{ij}\}$ passes through every node in $\hat{\mathcal{X}}$.*

In proving the claim 5, we first present two propositions.

Proposition 4 *Consider the graph $\mathcal{G}^k$ and let $w_{ij}^{k'}$ be any set of edge-weights on $\mathcal{G}^k$ satisfying the property*

$$w_{ij}^{k'} = \begin{cases} w'_{ij} & \text{if } (ij) \in \mathcal{E}^{k-1} \text{ and } \underline{w}_{ij} \leq w'_{ij} \leq \overline{w}_{ij} \\ \underline{w}_{ij} & \text{if } (ij) \in \mathcal{E}^k / \mathcal{E}^{k-1} \end{cases}$$

for some set of w'_{ij}s. Then, every shortest path on such a graph $\mathcal{G}^k$ passes through every node in $\hat{\mathcal{X}}^k$.

We defer the proof of this proposition to later.

Proposition 5 *If $\mathcal{G}^{*k+1}$ is a subgraph of $\mathcal{G}^k$, then edges in $\mathcal{G}/\mathcal{G}^k$ are never a part of the shortest path over $\mathcal{G}$. In particular, change in edge-weights over edges in $\mathcal{G}/\mathcal{G}^k$ has no effect on the shortest path in $\mathcal{G}$.*

Proof: Recall that $\mathcal{G}^{*k+1}$ is obtained as the set of shortest paths when the edge-weights are $\{w_{ij}^{k+1}\}$. In this scenario, all edges in $\mathcal{G}^k$ have maximum edge weight and all edges in $\mathcal{G}/\mathcal{G}^k$ have minimum edge weight. Since, the shortest path lies entirely in $\mathcal{G}^{*k+1}$, and thus in $\mathcal{G}^k$, any path that exits the graph $\mathcal{G}^k$ is necessarily longer than the shortest path. Further, any changes in edge-weights in $\mathcal{G}/\mathcal{G}^k$ will only increase the weight of such a path. Effectively the edges in $\mathcal{G}/\mathcal{G}^k$ play no role in determining the shortest path for any value of edge-weights. Thus, all shortest paths in $\mathcal{G}$ are restricted to the subgraph $\mathcal{G}^k$. □

Proof for Claim 5: We defined the set $\mathcal{F}^k$ as,

$$\mathcal{F}^k = \{ij : w_{ij}^k = \underline{w}_{ij}, ij \in \mathcal{E}^{*k}\}$$

If the termination criteria $\mathcal{F}^k = \varnothing$ is satisfied then all edges in $\mathcal{G}^{*k}$ are present in $\mathcal{G}^{k-1}$. By Proposition 5 all shortest paths lie in $\mathcal{G}^{k-1}$ for any edge-weights over edges in $\mathcal{G}/\mathcal{G}^{k-1}$. Additionally, since $\mathcal{G}^{*k}$ is a subgraph of $\mathcal{G}^{k-1}$, we have $\mathcal{G}^{k-1} = \mathcal{G}^k$. Thus, all shortest paths lie in $\mathcal{G}^k$ for any edge-weights over edges in $\mathcal{G}/\mathcal{G}^k$.

By Proposition 4 we saw that all shortest paths in $\mathcal{G}^k$ pass through all nodes in $\hat{\mathcal{X}}^k$ for any value of edge weight in $\mathcal{G}^{k-1}$. We saw above that at termination $\mathcal{G}^{k-1} = \mathcal{G}^k$, so equivalently all shortest paths in $\mathcal{G}^k$ pass through all nodes in $\hat{\mathcal{X}}^k$ for any value of edge weight in $\mathcal{G}^k$. Since at termination, all shortest paths in $\mathcal{G}$ lie entirely in $\mathcal{G}^k$ we have our result. □.

Before we prove Proposition 4 we present an additional Lemma we will use in the proof. v_r is the initial node of the rescuee and v_m is the target node indicated by the rescuer's signal.

Lemma 2 *Any node $v \in \hat{\mathcal{X}}^k$ divides the graph $\mathcal{G}^k$ into two subgraphs $\mathcal{G}_1^k$ and $\mathcal{G}_2^k$ with $v_r \in \mathcal{V}_1^k$ and $v_m \in \mathcal{V}_2^k$, such that v is the only common node, i.e. $\mathcal{V}_1^k \cap \mathcal{V}_2^k = \{v\}$.*

Proof: Let v' be another node that is common to both sub-graphs $\mathcal{G}_1^k$ and $\mathcal{G}_2^k$. Recall that every path in the graph $\mathcal{G}^{*k}$ is a shortest path from v_r to v_m in $\mathcal{G}$ with edge-weights $\{w_{ij}^k\}$. Since $\mathcal{G}^k = \bigcup_{i=1}^k \mathcal{G}^{*k}$, every node in $\mathcal{G}^k$ must be a part of a shortest path for some set of edge-weights. Thus, we can find atleast one path $\xi'_{v_r \to v_m}$ that passes through v' such that it forms the shortest path for some set of edge-weight $\{w_{ij}^l\}$ for some $l \leq k$.

Since, $\mathcal{G}^k$ is an acyclic digraph with all paths originating from v_r and terminating at v_m, we cannot have any path that travels from $\mathcal{G}_2^k$ to $\mathcal{G}_1^k$. Thus, the path $\xi'_{v_r \to v_m}$ containing node v' cannot also contain v.

We showed that $\xi'_{v_r \to v_m}$ is a shortest path on the graph $\mathcal{G}$ for some value of edge-weights $\{w_{ij}^l\}$ and that doesn't pass through v. But, such a v cannot lie in $\hat{\mathcal{X}}$ by definition. Thus, by contradiction we have shown we cannot have another node v' common to both graphs $\mathcal{G}_1^k$ and $\mathcal{G}_2^k$. □

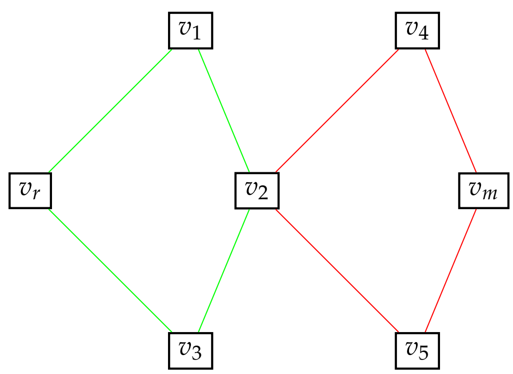


Figure B.1: A possible representation of $\mathcal{G}^1$. All edge weights are minimum. $\hat{\mathcal{X}}^1$ would contain $\{v_r, v_2, v_m\}$. v_2 here connects the two sub-graphs red and green.-

As a direct result from Lemma 2 we have,

Corollary 2

Proof for Proposition 4: From the algorithm we see that any shortest paths over the graph $\mathcal{G}^k$ with edge-weights $\{w_{ij}^k\}$ necessarily passes through every points in $\hat{\mathcal{X}}^k$. We wish to show the same holds true for edge-weights $\{w_{ij}^{k'}\}$. For these edge-weights let us assume there exists a shortest path $\xi'_{v_r \to v_m}$ that does not lie entirely in $\mathcal{G}^k$. By Corollary 2, if we show that such a shortest path exiting $\mathcal{G}^k$ can't exist then we have completed the proof.

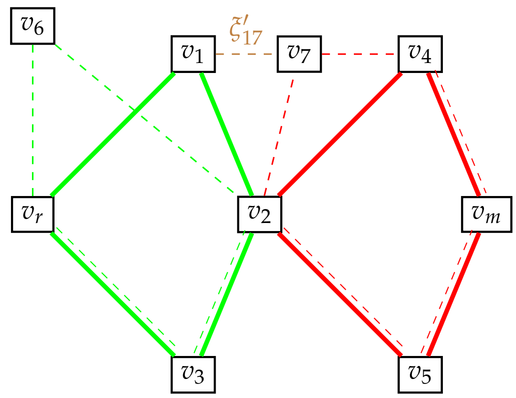


Figure B.2: A possible representation of $\mathcal{G}^k$. All thick edge weights are maximum and all thin edges are minimum weight. $\hat{\mathcal{X}}^k$ would contain $\{v_r, v_2, v_m\}$. v_2 here connects the two sub-graphs red and green. The dashed lines indicate edges in $\mathcal{G}^{*k}$ sub-graph. We want to show that ξ'_{17} can't exist for any edge-weights $\{w_{ij}^{k'}\}$

Let s and t denote the node where the path ξ' leaves and rejoins the graph $\mathcal{G}^k$. We know that it must leave and rejoin as both the start (v_r) and the end (v_m) are a part of the graph. It may leave and return to the sub-graph $\mathcal{G}^k$ multiple times but for the purpose of this proof we can without loss of generality assume it does so just once each. This assumption is justified at the end of this proof. Now, let $\xi'_{s\to t}$ denote the slice of the path that is outside $\mathcal{G}^{k-1}$. We can find a path between s and t entirely in the graph $\mathcal{G}^{k-1}$ as well and denote such a path as $\xi^*_{s\to t}$. Since, ξ' is the shortest path with edge-weights $\{w_{ij}^{k'}\}$ we have,

$$\phi_{w'}(\xi'_{s\to t}) \leq \phi_{w'}(\xi^*_{s\to t}) \tag{B.3}$$

Where $\phi_{w'}(\xi)$ gives the path cost of path ξ with weights $\{w_{ij}^{k'}\}$. Now, increasing the weights in the graph $\mathcal{G}^{k-1}$ to go from the set $\{w_{ij}^{k'}\}$ to $\{w_{ij}^{k}\}$ will still maintain the inequality (B.3), as the left hand side is not affected by the change in costs of edges in the $\mathcal{G}^{k-1}$ and the right hand side is increasing with $\{w_{ij}\}$.

$$\phi_{w^k}(\xi'_{s\to t}) \leq \phi_{w^k}(\xi^*_{s\to t}) \tag{B.4}$$

Where $\phi_{w^k}(\xi)$ gives the path cost of path ξ with weights $\{w_{ij}^{k}\}$. But, (B.4) implies that there exists a shorter path outside graph $\mathcal{G}^k$ (and thus outside $\mathcal{G}^{*k}$) which is not possible. Thus, any shortest path over edge-weights $\{w_{ij}^{k'}\}$ must lie in the graph $\mathcal{G}^k$. Specifically, by Corollary 2 it must pass through all nodes $v \in \hat{\mathcal{X}}^k$.

In closing this proof we make a comment on the assumption made on ξ' above, that it exits the graph $\mathcal{G}^{k-1}$ at-most once. If it does exit and enter multiple times we can define $\xi'_{s\to t}$ as a collection of splices

$\{\zeta'_{s_i \to t_i} : i \in [K]\}$ where K denotes the number of splices of ζ outside $\mathcal{G}^{k-1}$. We can consider a corresponding collection $\zeta^*_{s \to t} = \{\zeta^*_{s_i \to t_i} : i \in [K]\}$ of splices within the graph $\mathcal{G}^{k-1}$ and the same proof holds with minor changes in vocabulary used. $\square$

Appendix C

Measurability of Candidacy Index for each Node

Consider an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with positive stochastic-edge weights $w_{ij} \in \Omega_{ij}$ over each edge $ij \in \mathcal{E}$. Define $\Omega = \prod_{ij \in \mathcal{E}} \Omega_{ij}$. An element in Ω is a vector $w \in \mathbb{R}^{|\mathcal{E}|}$ with edge weights over each edge. Let x_{ij} be an indicator variable over an edge belonging to a certain path P under consideration. Then the set of all paths between two nodes v^S and v^T in the graph can be parametrized as the set of vectors of the form $x \in \{0,1\}^{|\mathcal{E}|}$ satisfying some additional constraints. We will redefine this set $\mathcal{P}_{v^S \to v^T}$ as follows,

$$\mathcal{P}_{v^S \to v^T} = \left\{ x : x = [x_{ij}], ij \in \mathcal{E}, \sum_{j \neq i} x_{ij} - \sum_{j \neq i} x_{ji} = \begin{cases} 1 & i = v^S \\ -1 & i = v^T \\ 0 & else \end{cases} \right\}.$$

We can now define the path length function $f : \mathcal{P}_{v^S \to v^T} \times \Omega \mapsto \mathbb{R}$ between any two points v^S and v^T as,

$$f(x, w) = \sum_{ij \in \mathcal{E}} w_{ij} x_{ij}. \tag{C.1}$$

Note that the vector x completely defines a unique path between the source (v^S) and the target (v^T) nodes. Given the edge weights over the graph $\mathcal{G}$, the path length is a deterministic function of the path x. For every path x, f is a continuous function in w. In fact, it is linear in w for a fixed x. Thus, the path length function is a measurable function from Ω to $\mathbb{R}$ for each vector x. For any pair of measurable functions f_1 and f_2, the point-wise minimum of the two functions $min(f_1, f_2)$ is also measurable. We can then make the claim that the shortest path function $f^* : \Omega \mapsto \mathbb{R}$ defined as,

$$f^*(w) = \min_{x \in \mathcal{P}_{v^S \to v^T}} f(x, w)$$

is also measurable because the point-wise infimum of finite or countably infinite measurable functions is measurable.

We showed above that the shortest path length function is measurable. But we are more interested in describing the set of edges (or equivalently nodes) that form a part of the shortest path. For a pair of measurable functions f_1 and f_2 that map to $\mathbb{R}$, the set of points in the domain S.T. $f_1 \leq f_2$ is measurable. We can now define a measurable indicator function $\mathbb{1}_x : \Omega \to \{0, 1\}$ for each path $x \in \mathcal{P}_{v^S \to v^T}$ as,

$$\mathbb{1}_x(w) = \mathbb{1}\left(\bigcap_{y \in \mathcal{P}_{v^S \to v^T}} \{w : f(x, w) \leq f(y, w)\} \right)$$

This function is measurable because indicator function over a measurable set is measurable. It is easy to see that function $\mathbb{1}_x$ takes value 1 for w if x is a shortest path for the edge-weights w.

It is easy to see that the function we defined as $g(v, w)$ where v is the node and w is the edge-weights over the graph can be easily constructed using a linear combination of functions $\mathbb{1}_x$ as defined above. To show this first we define

$$\mathcal{X}(w) = \sum_{x \in \mathcal{P}_{v^S \to v^T}} x \mathbb{1}_x(w).$$

$\mathcal{X}(w)$ is a vector with each element being the number of shortest paths on which an edge lies given edge-weight w. This vector has a one-one correspondence with a vector which determines how many shortest paths a particular node of the graph lies on. To retrieve the number of shortest paths each node lies on we will use a simple linear transformation. Consider an $|\mathcal{V}| \times |\mathcal{E}|$ matrix A with the ij^{th} entry being 1 if the i^{th} node has the j^{th} edge as a outgoing edge emanating from it. All other entries in the matrix are 0. Then, it can be seen that

$$\mathcal{Y}(w) = A\mathcal{X}(w)$$

is a vector with $|\mathcal{V}|$ entries, with the k^{th} entry being the number of shortest path the the k^{th} node lies on. Since, every entry of $\mathcal{Y}$ is obtained as a linear combinations of measurable functions $\mathbb{1}_x$ we know that the $\mathcal{Y}$ is a measurable function. Then, when $\mathbb{1}_x(w) \neq 0$ for atleast one $x \in \mathcal{P}_{v^S \to v^T}$ we can obtain $g(v, w)$ as the v^{th} entry in,

$$\frac{\mathcal{Y}(w)}{\sum_{x \in \mathcal{P}_{v^S \to v^T}} \mathbb{1}_x(w)}$$

It can be seen that $g(v, w)$ is a measurable function over Ω for every node v. Thus, we have justified the validity of taking probabilities and expectations over this function of random variables.